สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องรายงานผลการออกรางวัลลอตเตอรี่ควบคุมด้วยไมโครคอนโทรลเลอร์

LOTTO REPORT BY MICROCONTROLLER CONTROL

T 0 5 0 1 1 1

โดย

นายดำรงค์ชัย พรหมมา

นายธีระพงษ์ บุญหล้า

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50111
วัน,เดือน,ปี 2 1 เม.ย. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรายงานผลการออกรางวัลลอตเตอรี่ควบคุมด้วยไมโครคอนโทรลเลอร์

LOTTO REPORT BY MICROCONTROLLER CONTROL

โดย

นายดำรงค์ชัย พรหมมา 43015067

นายธีระพงษ์ บุญหล้า 43015070

อาจารย์ที่ปรึกษา

รศ.สมยศ จุณณะปิยะ

ดร.พิเชฐ ม่วงนวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **เครื่องรายงานผลการออกรางวัลลอตเตอรี่ควบคุมด้วยไมโครคอนโทรลเลอร์**

LOTTO REPORT BY MICROCONTROLLER CONTROL

ผู้จัดทำ

1. **นายดำรงค์ชัย พรหมมา** 43015067
2. นายธีระพงษ์ บุญหล้า 43015070

..... อาจารย์ที่ปรึกษา

(รศ.สมยศ จุณณะปิยะ)

..... อาจารย์ที่ปรึกษา

(ดร.พิเชฐ ม่วงนวล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรายงานผลการออกรางวัลลอตเตอรี่ควบคุมด้วยไมโครคอนโทรลเลอร์

LOTTO REPORT BY MICROCONTROLLER CONTROL

โดย นาย ดำรงค์ชัย พรหมมา 43015067

นาย ธีระพงษ์ บุญหล้า 43015070

อาจารย์ที่ปรึกษา รศ.สมยศ จุณณะปิยะ

ดร.พิเชฐ ม่วงนวล

บทคัดย่อ

โครงงานฉบับนี้เสนอการพัฒนาเครื่องรายงานผลรางวัลลอตเตอรี่ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์โดยเครื่องจะถูกป้อนข้อมูลรางวัลเก็บไว้ในส่วนเก็บข้อมูลรางวัล เมื่อผู้ที่ต้องการตรวจสอบต้องการทราบผลก็จะโทรศัพท์ เข้ามายังศูนย์จากนั้นทำการป้อนเลขหมายลอตเตอรี่ เครื่องก็จะแสดงผลทางเสียงที่มีการโปรแกรมไว้ ให้ผู้ตรวจสอบทราบว่าหมายเลขดังกล่าวถูกรางวัลหรือไม่

การออกแบบจะใช้ไมโครคอนโทรลเลอร์ ร่วมกับโปรแกรมวิชวลเบสิกในไมโครคอมพิวเตอร์ ทำให้สะดวกในการใช้งานและประหยัดค่าใช้จ่าย

ABSTRACT

This project presents the development of Lotto Report by Microcontroller control. The lotto program is edit to data memory. When users have calling, the machine is announce to user has a reward or not.

The designed of project is use low cost microcontroller and Visual Basic Program. To help convenient for use and save money.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 ความสำคัญและที่มา	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ผลที่คาดว่าจะได้รับ	1
1.4 ขอบเขตของโครงงาน	1
บทที่ 2 หลักการที่เกี่ยวข้อง	
2.1 บทนำ	2
2.2 ไมโครคอนโทรลเลอร์ 89C51	2
2.3 อินเตอร์รัพท์	8
2.4 การควบคุมการอินเตอร์รัพท์	10
2.5 การรีเซต	11
2.6 การโปรแกรม 89C51	12
2.7 ขั้นตอนการโปรแกรม 89C51	12
2.8 วงจรโมโนสเตเบิลโดยใช้ไอซีไทม์เมอร์ 555	13
2.9 ไอซีบันทึกเสียง ISD 2590	19
2.10 ไอซีถอดรหัสความถี่โทรศัพท์	24
2.11 โทรศัพท์	28
2.12 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51	30
2.13 การสื่อสารข้อมูลแบบอะซิงโครนัส	31
2.14 โปรแกรม Visual Basic	35
บทที่ 3 ขั้นตอนการดำเนินงาน	
3.1 บทนำ	40
3.2 ศึกษาข้อมูล	41
3.3 วงจรเปลี่ยนสัญญาณกระดิ่ง (Ringing) เป็นสัญญาณดิจิตอล (Digital)	41
3.4 ออกแบบลายวงจรพิมพ์	46
3.5 ออกแบบโปรแกรม	46
3.6 ทดลองและวิเคราะห์การทำงาน	46
3.7 ประกอบวงจรบนลายวงจร	47
3.8 ลงกล่อง	47
บทที่ 4 การทดลองและผลการทดลอง	
4.1 การทดลอง	48
4.1.1 วงจรถอดรหัสความถี่โทรศัพท์ โดยไอซี MT 8870	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 วงจรตรวจจับสัญญาณกระดิ่ง 48
4.1.3 การทดลองส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์ 49
4.2 ผลการทดลอง 50
4.2.1 ผลการทดลองวงจรถอดรหัสความถี่โทรศัพท์ โดยไอซี MT 8870 50
4.2.2 ผลการทดลองวงจรตรวจจับสัญญาณกระดิ่ง 51
4.2.3 ผลการทดลองการทดลองส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์ 52
บทที่ 5 บทวิจารณ์และบทสรุป
5.1 บทวิจารณ์ 53
5.2 บทสรุป 53
5.3 แนวทางการพัฒนา 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่

2.1 แสดงการจัดตำแหน่งขาต่างๆของไมโครคอนโทรลเลอร์เบอร์ 89C51 3

2.2 หน่วยความจำสำหรับเก็บโปรแกรม 7

2.3 หน่วยความจำสำหรับเก็บข้อมูล 7

2.4 แสดงการกำหนดบิตในรีจิสเตอร์ของIEและIP 9

2.5 แสดงการต่อใช้งานขณะ โปรแกรมเข้าไปใน 89C51 13

2.6 แสดงการทดสอบการโปรรแกรมใน 89C51 13

2.7 แสดงองค์ประกอบภายในไอซี 555 14

2.8 แสดงตำแหน่งขาของไอซี 555 15

2.9 แสดงการทำงานของกราวด์โหลดและซัพพลายโหลดขณะต่อเข้ากับวงจรไอซี 555 16

2.10 การทำงานของขาดิสชาร์ต 16

2.11 แสดงการทำงานของไอซี 555 โดยเมื่อป้อนแรงดันอินพุท Ei และวัดสัญญาณที่เอาท์พุทที่ขา 3 หรือ Vo3 เมื่อเทียบกับ EI 17

2.12 แสดงการทำงานของวงจรโมโนสเตเบิล 18

2.13 ลักษณะการจัดขาของ ISD 2590 19

2.14 บล็อกไดอะแกรมภายในของ ISD 2590 20

2.15 แสดงลายละเอียดขาของ MT 8870 24

2.16 แสดงโครงสร้างภายในของ MT8870 25

2.17 แสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาการ์ดไทม์พร้อมวิธีคำนวณ 27

2.18 ไทม์มิ่งไดอะแกรมของ MT 8870 27

2.19 แสดงการต่อวงจรภาคอินพุท 27

2.20 แสดงวงจรผลิตความถี่ 28

2.21 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส 31

2.22 แสดงรายละเอียดการทำงานแต่ละบิต 32

2.23 แสดงรายละเอียดการออกแบบ Database 36

3.1 วงจรเปลี่ยนสัญญาณ Ringing เป็นสัญญาณ Digital 42

3.2 วงจรสัญญาณ DTMF เป็นสัญญาณ Digital 42

3.3 วงจรเสียงตอบรับและวงจรบันทึกเสียง 43

3.4 วงจรควบคุมระบบ 45

3.6 วงจรเพาเวอร์ซัพพลาย 45

4.1 วงจรถอดรหัส DTMF 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 วงจรตรวจจับสัญญาณกระดิ่ง 50
4.3 สัญญาณที่วัดได้จากขา 5 ของไอซี 4N25 51
4.4 สัญญาณที่วัดได้จากขา 3 ของไอซี 555 51
4.5 ผลที่ได้จากการทดลองส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์ 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

2.1 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต P3.X 5

2.2 แอดเดรสของรีจิสเตอร์ต่างๆ 6

2.3 แอดเดรสของการอินเตอร์รัพท์ 11

2.4 ค่าข้อมูลของรีจิสเตอร์ต่างๆ 12

2.5 แสดงการเซตค่าสัญญาณขณะทำการโปรแกรม 13

2.6 สถานะการทำงานของไอซี 555 17

2.7 คุณสมบัติทางไฟฟ้าบางอย่างที่แตกต่างกันของไอซีตระกูล ISD 25XX 21

2.8 คุณสมบัติทางไฟฟ้าของ ISD 2590 21

2.9 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ 26

2.10 ความถี่ DTMF กับปุ่มที่กด 29

4.1 แสดงค่าที่ได้จากการถอดรหัส 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญแผนภาพ

แผนภาพที่

3.1 ขั้นตอนการทำงาน 41

3.2 การบันทึกและตอบเสียง 44

3.3 โฟล์วชาร์ตหลักที่ใช้ในการทำโครงงาน 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ความสำคัญและที่มา

1.1 ความสำคัญและที่มา

ปัจจุบันโทรศัพท์เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์ ในการที่จะสนองความต้องการและอำนวยความสะดวกในรูปแบบต่างๆอย่างมาก เพื่อที่จะลดปัญหาต่างๆที่เกิดขึ้นในชีวิตประจำวัน เช่น การเดินทาง และความยุ่งยากในการตรวจสอบข้อมูลต่างๆ จึงได้มีการสร้างเครื่องรายงานผลการออกรางวัลลอตเตอรี่ทางโทรศัพท์ควบคุมด้วยไมโครคอนโทรลเลอร์ขึ้น เป็นเครื่องที่ใช้รายงานผลการออกรางวัลลอตเตอรี่ทางโทรศัพท์ควบคุมด้วยไมโครคอนโทรลเลอร์ อีกทั้งยังสามารถใช้เป็นต้นแบบในการประยุกต์ไปสร้างเครื่องรายงานข้อมูลและข่าวสารอื่นๆได้อีกด้วย เช่น รายงานข้อมูลคนไข้ในโรงพยาบาล รายงานผลกีฬา เป็นต้น

1.2 วัตถุประสงค์

เพื่อสามารถสร้างเครื่องรายงานผลการออกรางวัลลอตเตอรี่ทางโทรศัพท์ควบคุมด้วยไมโครคอนโทรลเลอร์

1.3 ผลที่คาดว่าจะได้รับ

ผลที่คาดว่าจะได้รับในการสร้างเครื่องรายงานผลการออกรางวัลลอตเตอรี่ทางโทรศัพท์ควบคุมด้วยไมโครคอนโทรลเลอร์ มีดังนี้ คือ

1. สามารถรายงานผลการออกรางวัลลอตเตอรี่ผ่านทางโทรศัพท์
2. สามารถรายงาน ข่าวสาร หรือข้อมูลต่างๆผ่านทางโทรศัพท์
3. ช่วยลดปัญหาการเดินทางและประหยัดเวลา
4. ช่วยลดปัญหาความยุ่งยากในการตรวจสอบข้อมูล
5. สามารถเข้าใจถึงการทำงานของไมโครคอนโทรลเลอร์
6. สามารถป้อนเก็บข้อมูล ปรับเปลี่ยนและเพิ่มเติมได้ ตามฐานข้อมูลที่ใช้

1.4 ขอบเขตของโครงงาน

เมื่อโครงงานนี้สำเร็จเรียบร้อยแล้วจะต้องสามารถใช้เป็นเครื่องรายงานผลการออกรางวัลลอตเตอรี่ได้โดยสามารถป้อนเก็บข้อมูลและรายงานผลกลับมายังผู้ที่ต้องการตรวจผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการที่เกี่ยวข้อง

2.1 บทนำ

หลักการทำโครงงานนี้ได้มีการศึกษาค้นคว้าข้อมูลจากหลายแหล่ง มีทั้งหลักการที่เรียนแล้วแต่รายละเอียดยังไม่พอต้องศึกษาเพิ่มเติม เช่น หลักการของไอซีไมโครคอนโทรลเลอร์ตระกูล 8051 ซึ่งเป็นไอซีประมวลผลที่มีความเหมาะสมที่จะนำมาใช้ในโครงงาน หลักการเชื่อมต่อระหว่าง Computer กับ MCS-51 ผ่านทางพอร์ตอนุกรม RS-232 การเขียนโปรแกรม Visual Basic และหลักการที่ใช้ในโครงงานนี้ มีหลักการที่เกี่ยวข้องอยู่ทั้งหมดด้วยกัน ดังนี้

2.2 ไมโครคอนโทรลเลอร์ 89C51

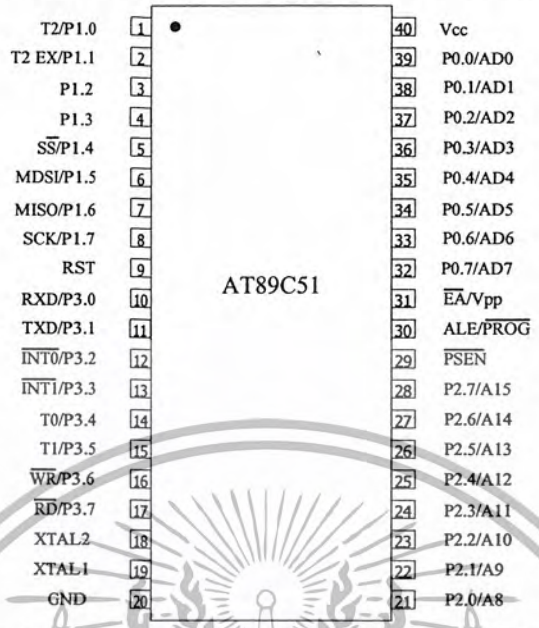
2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์เบอร์ 89C51

1. เป็นไมโครคอนโทรลเลอร์ขนาด 8 bit ต้องการแหล่งจ่ายไฟ +5 V ชุดเดียว
2. มีวงจรออสซิลเลเตอร์(Oscillator)และวงจรผลิตสัญญาณนาฬิกา(Clock Oscillator) ภายในไอซี
3. มีขาสัญญาณอินพุต/ เอาต์พุตจำนวน 32 bit
4. สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก(External Program Memory โดยอ้างแอดเดรสได้ถึง 64 Kbyte
5. มีหน่วยความจำโปรแกรมภายในตัว(Flash Memory)ขนาด 8 Kbyte
6. มีหน่วยความจำข้อมูลภายใน(Data Memory)ตัวขนาด 256 byte
7. หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น
8. มีไทม์เมอร์/เคาน์เตอร์ (Timer/Counters)ขนาด 16 bit จำนวน 2 ชุด
9. การอินเตอร์รัพท์ (Interrupt)สามารถทำได้จาก 6 แหล่งกำเนิด โดยการอินเตอร์รัพท์ยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ
10. มีพอร์ตสื่อสารอนุกรมภายในของมันตัวเองซึ่งทำงานเป็นแบบฟูลดูเพล็กซ์(Full Duplex)
11. มีชุดคำสั่งในการคำนวณทางคณิตศาสตร์ทางตรรกศาสตร์ และยังใช้คำสั่งเดียวกันกับไมโครคอนโทรลเลอร์ตระกูล MCS-51
12. คำสั่งโดยส่วนใหญ่ใช้เวลาทำงานแต่ละคำสั่งเพียง 1 ไมโครวินาที (μSec) เมื่อใช้คริสตอล (Crystal) ความถี่ 12 MHz
13. สามารถใช้เครื่องคอมพิวเตอร์ PC ทำการลบและเขียนโปรแกรมได้โดยตรง
14. สามารถโปรแกรมทับในลักษณะซ้ำเดิมได้นับพันครั้งเหมือนกับการโปรแกรมในลักษณะเดียวกันกับ EPROM ที่รู้จักกันดี โดยการโปรแกรมสามารถโปรแกรมได้ในลักษณะ "In – System Programming" บนบอร์ดคอนโทรลที่เราออกแบบขึ้นมาได้ หรือจะโปรแกรมโดยใช้เครื่องโปรแกรมทั่วไป เช่น Pro – 100 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ลักษณะภายนอกของไมโครคอนโทรลเลอร์เบอร์ 89C51

ไมโครคอนโทรลเลอร์เบอร์ 89C51 จะมีตำแหน่งขาพื้นฐาน ดังแสดงในภาพที่ 2.1



ภาพที่ 2.1 แสดงการจัดตำแหน่งขาต่างๆ ของไมโครคอนโทรเลอร์เบอร์ 89C51

2.2.3 หน้าที่การใช้งานของแต่ละขา

- ขา Vcc เป็นขาป้อนแรงดันไฟเลี้ยง +5 Volt
- ขา GND เป็นขากราวด์ (ground) สำหรับต่อกับกราวนด์ของระบบ
- ขารีเซต (RST) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์โดยการรีเซตต้องคงสถานะเป็น 1 อย่างน้อยนาน 2 แมชชีนไซเกิล ในขณะที่ออสซิลเลเตอร์ยังทำงานตามปกติ
- ขา ALE/PROG เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลตช์ (Latch) ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากขานี้ยังทำหน้าที่เป็นอินพุตรับพัลส์ในการรับโปรแกรม (program pulse input) ในส่วนของหน่วยความจำแบบแฟลช (Flash memory) สำหรับไมโครคอนโทรลเลอร์เบอร์ AT89C51 ที่มีหน่วยความจำโปรแกรมภายในเป็นแบบแฟลช (Flash memory)
- ขา PSEN (Program Store Enable)ทำหน้าที่เป็นสัญญาณสโตรบเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอกเมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณ สโตรบจำนวน 2 ครั้งในแต่ละแมชชีนไซเกิล แต่ในขณะที่ติดต่อกับหน่วยความจำภายนอกจะไม่มีการส่งสัญญาณสโตรบแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา $\overline{EA}$/VPP (External Access enable / VPP) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอกโดยถ้ามีสถานะเป็น 0 จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0000 –1FFFH อย่างไรก็ตามถ้าบิตป้องกัน(security bit)ในหน่วยความจำแบบแฟลช (Flash memory) ถูกโปรแกรมไว้ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (Vpp) ขนาด 12 โวลท์ เพื่อใช้ในระหว่างการโปรแกรม
- ขา X-TAL1 และขา X-TAL2 เป็นขาที่จะทำงานในลักษณะอินพุตและเอาต์พุตของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟเออร์ (Inverting Oscillator Amplifier) สำหรับใช้ต่อร่วมกับคริสตอลภายนอก
- ขาพอร์ต 0 (Port 0) มี 8 ขา ได้แก่ขา P0.0 – P0.7 เป็นขาพอร์ตอินพุต/เอาต์พุต 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสถานะปล่อยลอย ซึ่งในสถานะนี้เองที่สามารถนำมาใช้เป็นลักษณะพอร์ตอินพุตอิมพีแดนซ์สูงได้ นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยที่หน้าที่การกำหนดตำแหน่งแอดเดรสไบต์ต่ำ (A0 – A7)ซึ่งจะใช้งานเป็นแบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 bit นั้นก็คือที่ Data 0-7(D0-D7)
- ขาพอร์ต 1 (Port1) มี 8 ขา ได้แก่ ขาP1.0 –P1.7 เป็นขาพอร์ตอินพุต/เอาท์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต
- ขาพอร์ต 2 (Port 2) มี 8 ขา ได้แก่ขา P2.0 – P2.7 เป็นขาอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วยโดยทำหน้าที่การกำหนดตำแหน่งแอดเดรสไบต์สูง
- ขาพอร์ต 3 (Port 3) มี 8 ขา ได้แก่ขา P3.0 – P3.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมักถูกใช้งานในหน้าที่พิเศษต่างๆ ดังแสดงในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต	หน้าที่พิเศษ
P3.0	RXD(serial input port)
P3.1	TXD(serial output port)
P3.2	INT0(external interrupt 0)
P3.3	INT1(external interrupt 1)
P3.4	T0(Timer 0 external input)
P3.5	T1(Timer 1 external input)
P3.6	WR(external data memory write strobe)
P3.7	RD(external data memory read strobe)

ตารางที่ 2.1 แสดงหน้าที่พิเศษของแต่ละขาของพอร์ต P3.X

2.2.4 รีจิสเตอร์

1. รีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์ใช้งานทั่วไป มีไว้สำหรับให้ผู้เขียนโปรแกรมสามารถนำข้อมูลไปพักไว้ชั่วคราวหรือใช้งานชั่วคราวได้ตามต้องการ ซึ่งรีจิสเตอร์ใช้งานทั่วไปนี้มีอยู่ด้วยกัน 8 ตัวถูกจัดให้อยู่รวมกันและมีให้เลือกใช้ถึง 4 แบงค์ (bank) นั่นคือมี รีจิสเตอร์ใช้งานทั่วไปจำนวน 32 ตัว ให้ใช้งานเพียงแต่เลือกใช้รีจิสเตอร์ R0-R7 ในแบงค์ใดแบงค์หนึ่ง ซึ่งจะถูกกำหนดจากบิต RS0,RS1ในรีจิสเตอร์หน้าที่พิเศษ PSW ดังนั้นการเลือกใช้จึงเลือกได้เพียงแบงค์เดียวในการใช้งานในขณะนั้น อย่างไรก็ตามค่าข้อมูลที่เก็บไว้ในรีจิสเตอร์แบงค์ใดก็ตามที่มีชื่อเดียวกันแต่อยู่คนละแบงค์กันไม่มีผลซึ่งกันและกัน ทำให้ผู้เขียนโปรแกรมใช้งานรีจิสเตอร์ทั่วไปนี้ได้ทั้ง 32 ตัว อย่างเต็มที่และไม่ยุ่งยากในการเขียนโปรแกรม

2 รีจิสเตอร์ทำหน้าที่พิเศษ (SFR)

รีจิสเตอร์หน้าที่พิเศษมีบทบาทอย่างมากในการควบคุมการทำงานของไมโครคอนโทรลเลอร์และทำให้การเขียนโปรแกรมสามารถทำได้สะดวกมากขึ้น รีจิสเตอร์หน้าที่พิเศษทำหน้าที่สำคัญคือ ควบคุมการทำงานในส่วนต่างๆภายในไมโครคอนโทรลเลอร์และทำหน้าที่แสดงสถานการณ์ทำงาน ซึ่งในรีจิสเตอร์หน้าที่พิเศษบางตัวยังสามารถเข้าถึงได้ในระดับบิต (bit addressable) การจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์หน้าที่พิเศษต่างๆ การทำงานของ 89C51 จะไม่ใช้แรมภายใน 128ไบท์ ตำแหน่งที่ 00H ถึง 7FH ซึ่งเป็นกลุ่มของรีจิสเตอร์พิเศษภายใน (SFR) ซึ่งอาจทำเป็นแอดเดรสเหมือนกับแรมภายในโดยใช้แอดเดรส 80H ถึง FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NAME	FUNCTION	INTETNAL RAM ADDRESS (HEX)
A	Accuculator	0E0
B	Arithmetic	0F0
DPH	Adressing external memory	83
DRS	Adressing external memory	82
IE	Interupt emable control	0A8
IP	Interupt priority	0B8
P0	Input/Output port latct	80
P1	Input/Output port latct	90
P2	Input/Output port latct	A0
P3	Input/Output port latct	0B0
PCON	Power control	87
PSW	Program status word	0D0
SCON	Serial port control	98
SNUF	Serial port data buffer	99
SP	Stack pointer	81
TMOD	Timer/counter mode control	89
TCON	Timer/counter control	88
TL0	Timer 0 low byte	8A
TH0	Timer 0 high byte	8C
TL1	Timer 1 low byte	8B
TH1	Timer 1 high byte	8D

ตารางที่ 2.2 แอดเดรสของรีจิสเตอร์ต่างๆ

2.2.5 หน่วยความจำภายในของไมโครคอนโทรลเลอร์เบอร์ 89C51

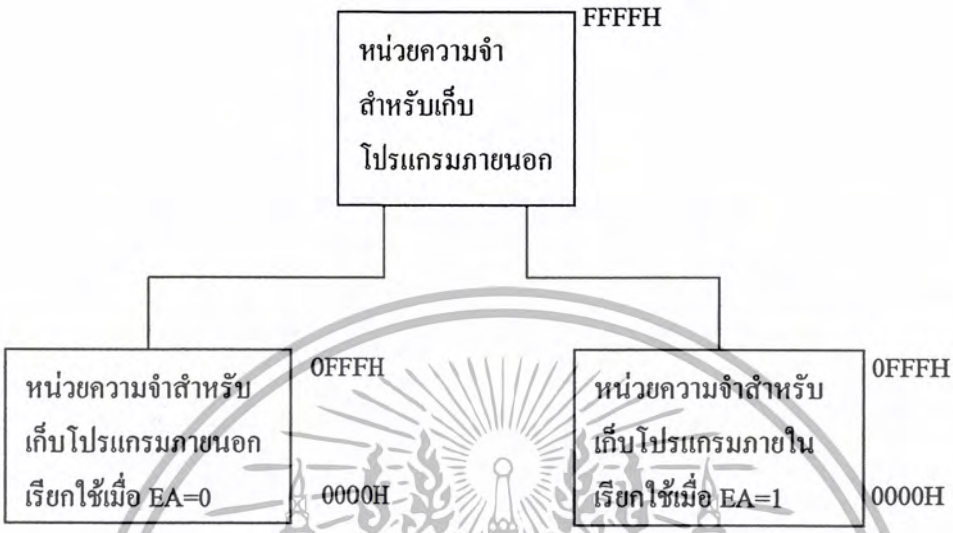
ในไมโครคอนโทรลเลอร์เบอร์ 89C51 แบ่งชนิดหรือหน้าที่ของหน่วยความจำลักษณะการใช้งานออกเป็น 2 ส่วนคือ หน่วยความจำโปรแกรม (Program Memory) และหน่วยความจำข้อมูล

1. หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมจะใช้สำหรับเก็บโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ ซึ่งจะเป็นหน่วยความจำแบบแฟลช (Flash Memory) ซึ่งสามารถแบ่งออกได้เป็น 2 ส่วน คือ หน่วยความจำโปรแกรมภายในและหน่วยความจำโปรแกรมภายนอก หน่วยความจำโปรแกรมภายในถูกเลือกใช้งานถ้าขาสัญญาณ EA มีค่าเป็น 1 โดยจะถูกใช้งานในช่วงแอดเดรส 0000-1FFFH นอกเหนือจากช่วงแอดเดรสนี้จะใช้หน่วยความจำโปรแกรมภายนอกทั้งหมด ในกรณีตรงกันข้ามถ้าขาสัญญาณ EA มีค่าเป็น 0 ในช่วงแอดเดรส 0000-1FFFH จะถูกใช้จากหน่วยความจำภายนอก หรือกล่าวได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าขาสัญญาณ EA มีค่าเป็น 0 จะเป็นการเลือกใช้หน่วยความจำโปรแกรมภายนอกทั้งหมดตลอดช่วงแอดเดรส ดังแสดงในภาพที่ 2.2



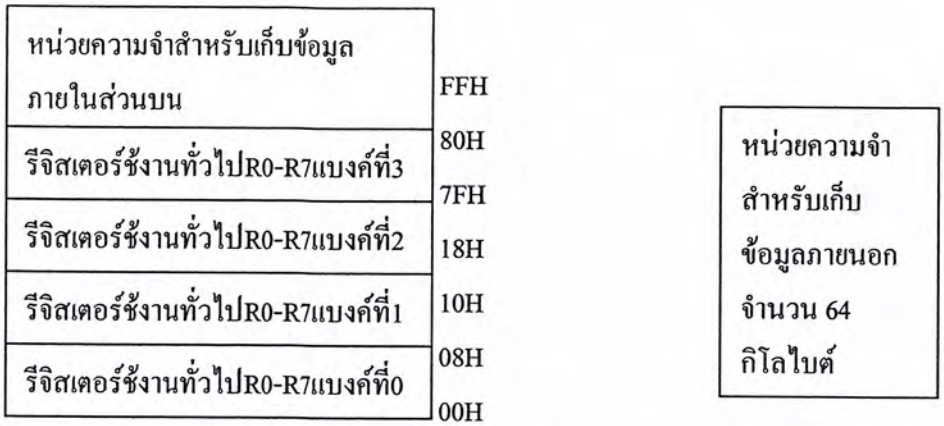
ภาพที่ 2.2 หน่วยความจำสำหรับเก็บโปรแกรม

2. หน่วยความจำข้อมูล

สำหรับหน่วยความจำข้อมูลจะใช้สำหรับเก็บข้อมูลหรือค่าตัวแปรต่างๆจากการทำงานของโปรแกรมซึ่งในเบอร์ 89C51จะมีหน่วยความจำในส่วนนี้เท่ากับ 256 ไบท์ ซึ่งจะเป็นหน่วยความจำแบบ RAM หน่วยความจำข้อมูลสามารถแบ่งออกได้เป็น 2 ส่วน คือ หน่วยความจำข้อมูลภายในและหน่วยความจำข้อมูลภายนอก สำหรับหน่วยความจำข้อมูลภายในยังแบ่งออกเป็น 2 ส่วนย่อย คือ ส่วนที่ใช้เก็บข้อมูลทั่วไปและส่วนที่ใช้เป็นรีจิสเตอร์หน้าที่พิเศษหรือ SFR (Special Function Register) โดยส่วนที่ใช้เก็บข้อมูลทั่วไปจะถูกใช้สำหรับเก็บข้อมูลหรือค่าตัวแปรต่างๆ จากการทำงานของโปรแกรม ส่วนรีจิสเตอร์ควบคุมการทำงานและบอกสถานการณ์ทำงานของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เบอร์ 89C51จะมีหน่วยความจำข้อมูลภายในขนาด 256 ไบท์ ดังแสดงในภาพที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 หน่วยความจำสำหรับเก็บข้อมูล

2.3 อินเทอร์รัพท์

โปรแกรมคอมพิวเตอร์มีเพียง 2 ทางเท่านั้นที่จะหาเงื่อนไขที่เป็นจริงของวงจรภายในและภายนอก วิธีแรก คือ ใช้คำสั่งซอฟท์แวร์กระโดดบนสภาวะแฟลกและพอร์ตพิน วิธีที่สอง คือ สัญญาณฮาร์ดแวร์ที่เรียกว่าอินเทอร์รัพซึ่งทำให้โปรแกรมไปเรียกซับรูทีน(Subroutine)เทคนิคของซอฟท์แวร์ใช้เวลาของโปรเซสเซอร์เมื่อถูกทำหน้าที่เมื่อต้องการใช้โปรแกรม Application เกือบทั้งหมดของไมโครคอนโทรลเลอร์เกี่ยวข้องกับสถานการณ์ที่เร็วพอจะควบคุมภาวะแวดล้อมที่สร้างเหตุการณ์

อินเทอร์รัพท์อาจสร้างโดยการทำงานของชิพภายในหรือโดยแหล่งกำเนิดภายนอก อินเทอร์รัพท์บางตัวทำให้ 8951 ทำฮาร์ดแวร์ให้เรียกซับรูทีนที่มีตำแหน่งที่กำหนดมาก่อน (โดยผู้ออกแบบ 8951) ในโปรแกรมเมมโมรี่มีอินเทอร์รัพท์ 5 ชนิดใน 8951 ซึ่งจะถูกสร้างโดยอัตโนมัติโดยการทำงานภายในของไทม์เมอร์แฟลก 0 ไทม์เมอร์แฟลก 1 และอินเทอร์รัพท์อนุกรม (RI หรือ TI) อินเทอร์รัพท์อีก 2 ตัวถูกทริกโดยสัญญาณภายนอก โดยวงจรที่ต่อขากับ INT0 และ INT1 (ขาของพอร์ต P3.2 และ P3.3)

(MSB) (LSB)

EA	X	ET2	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
EA	IE.7	disables all interrupt, if EA=0, no interrupt will be Acknowledged, if EA=1, each interrupt source is Individually enabled or disabled by setting or cleaning its Enable bit.
	IE.6	reserved
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt, if ET2=0, the Timer 2 interrupt is disabled.
ES	IE.4	enables or disables the Serial port interrupt. if ES=0, the Serial port interrupt is disabled.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET1	IE.3	enables or disables the Timer 1 overflow interrupt, if ET1=0, the Timer 1 interrupt is disabled.
EX1	IE.2	enables or disables External interrupt 1, if EX1=0, external interrupt 1 is disabled.
ET0	IE.1	enables or disables the Timer 0 overflow interrupt, if ETO=0, the Timer 0 interrupt is disabled.
EX0	IE.0	enables or disables External interrupt 0, if EX0=0, External interrupt 0 is disabled.

(MSB) (LSB)

PCT	X	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCT	IP.7	83C154/C1540D only Priority interrupt circuit control bit. The priority register contents are valid and priority assigned interrupts can be processed when this bit is "0" When the bit is "1" priority interrupt circuit is stopped, and interrupts can be controlled be the interrupt enable register (IE).
	IP.6	reserved
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2=1 programs it to the higher priority level.
PS	IP.4	defines the Serial Port interrupt priority level. PS=1 Programs it to the higher priority level.
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1=1 programs it to the higher priority level.
PTX	IP.2	defines the External interrupt 1 priority level. PX1=1 Programs it to higher priority level.
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0=1 programs it to the higher priority level.
PX0	IP.0	defines the External interrupt 2 priority level. PX0=1 Programs it to the higher priority level.

ภาพที่ 2.4 แสดงการกำหนดบิทในรีจิสเตอร์ของ IE และ IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timer Control Register (TCON)

โปรแกรมสามารถสกัดกั้นอินเทอร์รัพท์ทั้งหมดหรือบางตัวจากโปรแกรมโดยการเซตหรือเคลียร์บิทในรีจิสเตอร์เหล่านี้ IE,IP แสดงดังภาพที่ 2.4 หลังจากอินเทอร์รัพท์ถูกขจัดการโดยซับรูทีน ซึ่งทำโดยโปรแกรมเมอร์ในโปรแกรมเมมโมรี่ โปรแกรมอินเทอร์รัพท์ต้องทำงานที่คำสั่งที่เกิดอินเทอร์รัพท์ โปรแกรมถูกทำโดยเก็บค่า PC ไปบนสแตรคในแรมก่อนเปลี่ยน PC เป็นแอดเดรสของอินเทอร์รัพท์ในรอม ค่า PC จะได้จากแสตคหลังจากคำสั่ง RETI ที่ได้ทำแล้วที่ส่วนท้ายของซับรูทีน

2.3.1 อินเทอร์รัพท์แฟลกไทม์เมอร์

เมื่อไทม์เมอร์/เคาน์เตอร์เกิดโอเวอร์โฟลว์ผลคือ ไทม์เมอร์แฟลก (TF0, TF1) จะเซตเป็น 1 แฟลกถูกเคลียร์เป็น 0 เมื่ออินเทอร์รัพท์ทำให้โปรแกรมเรียกซับรูทีนของไทม์เมอร์ในเมมโมรี่

2.3.2 อินเทอร์รัพท์พอร์ตข้อมูลอนุกรม

เมื่อรับข้อมูลแล้วบิทอินเทอร์รัพท์ (RI) ต้องเซตเป็น 1 ใน SCON ข้อมูลส่งเรียบร้อยแล้ว บิทอินเทอร์รัพท์ (TI) ต้องเซตใน SCON และจะนำมา OR กัน เพื่อหาตัวอินเทอร์รัพท์แก่โปรเซสเซอร์ ซึ่งเป็นการอินเทอร์รัพท์พอร์ตข้อมูลอนุกรม บิทเหล่านี้จะไม่ถูกเคลียร์เมื่อการเรียกโปรแกรมอินเทอร์รัพท์ถูกทำโดยโปรเซสเซอร์ที่จัดการสื่อสารข้อมูลอนุกรมต้องรีเซต RI หรือ TI เพื่อจำทำข้อมูลถัดไป

2.3.3 อินเทอร์รัพท์ภายนอก

ขา INT0 และ INT1 จะใช้โดยวงจรภายนอก การอินพุทบนขาเหล่านี้ต้องเซตอินเทอร์รัพท์แฟลก IE0 และ IE1 ใน SCON เป็น 1 โดยวิธี 2 วิธี แฟลก IEX อาจเซตเมื่อขา INTX เป็น 0 หรือแฟลกอาจเซตเมื่อมีการเปลี่ยนจาก 1 เป็น 0 บนขา INTX บิท IT0,IT1 จะโปรแกรมขาสำหรับการอินเทอร์รัพท์ที่การเปลี่ยนลอจิกเมื่อเซตเป็น 1

แฟลก IEX จะรีเซตเมื่อการอินเทอร์รัพท์ที่การเปลี่ยนลอจิกพบโดยโปรแกรม และเข้าถึงซับรูทีนนี้ เป็นผลของนักออกแบบระบบ โปรแกรมจึงต้องเป็น 1 ก่อนที่ RETI จะถูกจัดการความผิดพลาดในเรื่องนี้จะทำให้เกิดการอินเทอร์รัพท์ทันทีหลังจาก RETI ด้วยแหล่งอินเทอร์รัพท์เดียวกัน

2.4 การควบคุมการอินเทอร์รัพท์

โปรแกรมต้องสามารถยับยั้งการอินเทอร์รัพท์ทั้งหมดหรือบางตัวเพื่อให้งานที่สำคัญทำจนเสร็จ รีจิสเตอร์ IE จะเก็บค่าบิทที่โปรแกรมได้ เพื่อให้ทำอินเทอร์รัพท์แล้วแหล่งอินเทอร์รัพท์แต่ละแห่งอาจถูกเลือกหรือไม่ก็ได้ บ่อยครั้งที่ต้องการอินเทอร์รัพที่สำคัญกว่าโดยทันทีบิทของรีจิสเตอร์ IP อาจเซตโดยโปรแกรมเพื่อกำหนดความสำคัญของแต่ละแหล่งอินเทอร์รัพท์ เพื่อให้อินเทอร์รัพท์ที่สำคัญกว่าเกิดเมื่อการอินเทอร์รัพท์เกิดพร้อมกันตั้งแต่ 2 แหล่งขึ้นไป

2.4.1 ความสามารถ/ไม่สามารถอินเทอร์รัพท์

บิทในรีจิสเตอร์ IE ต้องเซตเป็น 1 ถ้าแหล่งอินเทอร์รัพท์ถูกใช้ และเซตเป็น 0 เมื่อใช้ บิท EN เป็นบิทที่สำคัญสามารถ/ไม่สามารถอินเทอร์รัพท์ทุกแหล่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ความสำคัญของการอินเทอร์รัพท์

บิทใน IP ถูกกำหนดถ้าอินเทอร์รัพท์ต้องการลำดับสำคัญบิท จะเซตเป็น 1 เมื่อเป็นการอินเทอร์รัพท์ที่สำคัญสูงและเป็น 0 เมื่ออินเทอร์รัพท์มีความสำคัญต่ำอินเทอร์รัพท์ที่สำคัญไม่อาจถูกอินเทอร์รัพท์โดยตัวที่มีความสำคัญกว่าจนกว่าตัวที่สำคัญสูงกว่าจะทำเสร็จ

ถ้าอินเทอร์รัพท์ 2 ตัว ที่มีความสำคัญเท่ากันเกิดพร้อมกันจะมีการทำตามลำดับดังนี้

1. IP0
2. TF0
3. IE1
4. TE1
5. Serial = TI

การอินเทอร์รัพท์อนุกรมสามารถให้ความสำดับที่สำคัญกว่าทำก่อนโดยเซตบิทใน IP เป็น 0 และบิทอื่นๆ เป็น 0

2.4.3 จุดหมายการอินเทอร์รัพท์

แต่ละอินเทอร์รัพท์ทำให้โปรแกรมทำฮาร์ดแวร์ให้เรียกแอดเดรสที่ตรวจสอบแล้วในโปรแกรมเมมโมรี่นี้เป็นสิ่งที่โปรแกรมต้องวางรูทีนในแอดเดรสที่เกิดอินเทอร์รัพท์อินเทอร์รัพท์จะเก็บค่าใน PC เมื่อทำการอินเทอร์รัพจะเก็บค่าในแสตคและเรียกตำแหน่งที่ต้องการ ดังตารางที่ 2.3

คำสั่ง RETI จะอยู่ท้ายของรูทีนและรีเซตลอจิกของอินเทอร์รัพท์ เพื่อให้อินเทอร์รัพท์อีกตัวสามารถรับการอินเทอร์รัพท์ได้ อินเทอร์รัพท์อาจเกิดขึ้นแต่ไม่ถูกทำเนื่องจากสภาวะสกัดกั้นมีอยู่ตลอดการอินเทอร์รัพท์ นี้เป็นความต้องการเบื้องต้นในการใช้บริการ อินเทอร์รัพท์ INTX ทำที่ลอจิก 0

INTERRUPT	ADDRESS(HEX)
IE0	0003
TE0	000B
IE1	0013
TF1	001B
Serial	0023

ตารางที่ 2.3 แอดเดรสของการอินเทอร์รัพท์

2.5 การรีเซต

การรีเซต สามารถมองเป็นการอินเทอร์รัพท์สูงสุดเพราะโปรแกรมไม่สามารถสกัดกั้นได้การอินเทอร์รัพท์ชนิดนี้บ่อยครั้งจะเรียกว่า "Non maskable" เนื่องจากไม่มีบิทใดจะสามารถหยุดมันได้ซึ่งไม่เหมือนกับการอินเทอร์รัพท์แบบอื่น PC จะไม่เก็บค่าโปรแกรมครั้งสุด รีเซตเป็นคำสั่งที่สมบูรณ์ในการกระโดดไปที่ 0000H และเริ่มทำงานที่จุดนี้ เมื่อลอจิก 1 ให้ที่ขา RST ไมโครคอนโทรลเลอร์ 89C51 จะอยู่ในภาวะรีเซตหลังจากนั้นขา RST เป็น 0 รีจิสเตอร์ภายในไมโครคอนโทรลเลอร์ จะมีค่าดังตารางนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTER	VALUE (HEX)
PC	0000
DPTR	0000
A	00
B	00
SP	07
SPW	00
P3-0	FF
IP	XXX00000b
IE	0XX00000b
TCON	00
TMOD	00
TH0	00
THL	00
TH1	00
TL1	00
SCON	00
SBUF	XX
PCON	0XXXXXXXb

ตารางที่ 2.4 ค่าข้อมูลของรีจิสเตอร์ต่างๆ

2.6 การโปรแกรม 89C51

โดยปกติแล้วไมโครคอนโทรลเลอร์นี้จะอยู่ในสภาวะว่าง (Contect = FFH) และพร้อมที่จะได้รับการโปรแกรม ซึ่งสามารถโปรแกรมได้ทั้งแบบใช้แรงดันค่าสูง (+ 12V) และในแบบแรงดันต่ำ (ที่ระดับแรงดัน Vcc นั่นคือ + 5V) กรณีโปรแกรมในโหมดแรงดันไฟค่าสูง 12V สามารถทำได้โดยการใช้เครื่องโปรแกรมทั่วไป ที่สามารถจะโปรแกรมลงใน EPROM ได้ โดยใช้ซอฟต์แวร์ที่สามารถโปรแกรม 89C51 ส่วนการโปรแกรมแบบค่าแรงดันต่ำนั้นทำให้สามารถโปรแกรมได้ในแบบ In–System Programming

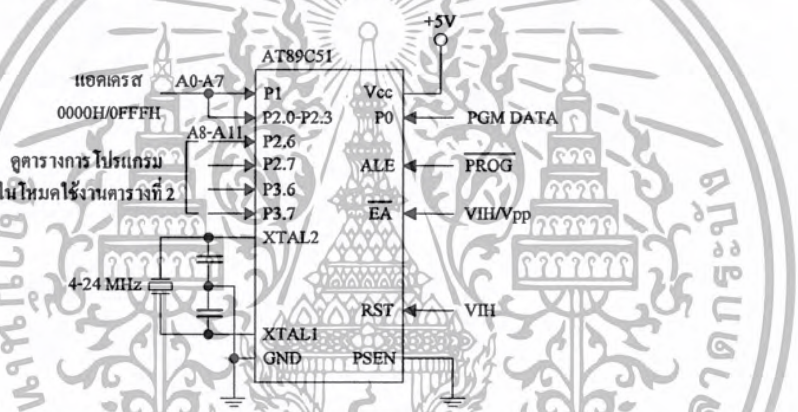
2.7 ขั้นตอนการโปรแกรม

ก่อนโปรแกรม 89C51 ไลน์ แอดเดรสข้อมูลและสัญญาณควบคุมจะต้องกำหนดขึ้นตามภาพที่ 2.5 และจะต่อขาใช้งานและสัญญาณเข้าไปยัง 89C51 ตามภาพที่ 2.6 และการตรวจสอบการโปรแกรมต่อตามตารางที่ 2.5

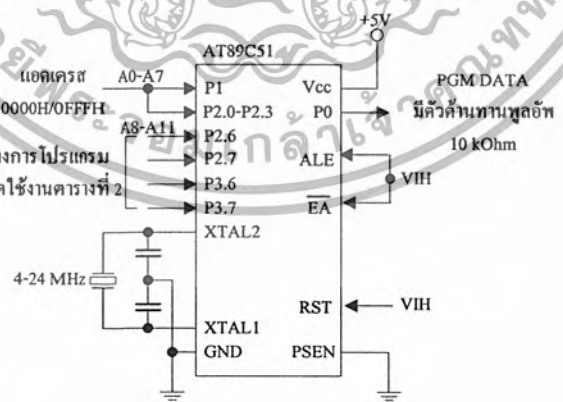
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาต่อใช้งาน / โหมดการทำงาน	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{pp}	P2.6	P2.7	P3.6	P3.7
เขียนรหัสข้อมูล	H	L		H/12 (1)	L	H	H	H
อ่านรหัสข้อมูล	H	L	H	H	L	L	H	H
การลบข้อมูลในชิป	H	L	(2)	H/12	H	L	L	L
กำหนดไบต์การอ่าน	H	L		H	L	L	L	L

ตารางที่ 2.5 แสดงการเซตค่าสัญญาณขณะทำการโปรแกรม



ภาพที่ 2.5 แสดงการต่อใช้งานขณะโปรแกรมเข้าไปใน 89C51



ภาพที่ 2.6 แสดงการทดสอบการโปรแกรมใน AT89C51

จากนั้นจะเริ่มทำตามขั้นตอนดังนี้

1. ป้อนค่าตำแหน่งแอดเดรสบนไลน์แอดเดรส
2. ป้อนไบท์ข้อมูลเข้าทางไลน์ข้อมูล
3. กำหนดค่าสัญญาณควบคุมที่ถูกต้องเข้าทางขา P2.6, P3.6, P3.7

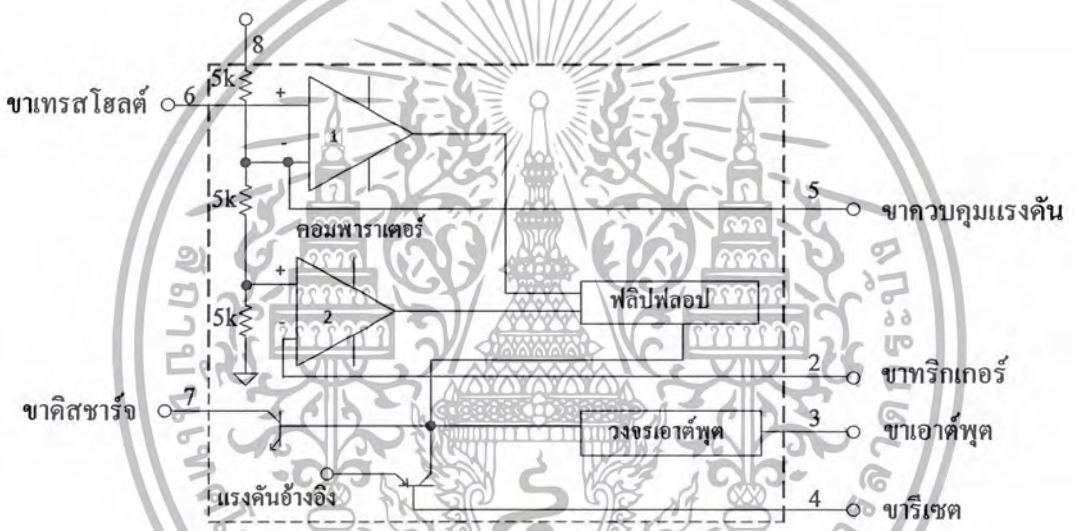
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กำหนดค่าแรงดันป้อนให้ขา EA/Vpp ไว้ที่ +5 ในกรณีที่ค่าแรงดันสูง

5. ป้อนพัลส์ที่ขา ALE/PROG เมื่อจะโปรแกรม 1 ไบท์เข้าไปในหน่วยความจำแฟลช จากนั้นวงรอบการเขียนข้อมูลจะเกิดขึ้นเองตามมา โดยในระยะเวลาใน 1 รอบ จะไม่เกิน 1.5 มิลลิวินาที เสร็จแล้วเริ่มทำขั้นตอนที่ 1 ถึง 5 โดยการเปลี่ยนแอดเดรสแล้วข้อมูลชุดถัดไปจนครบทั้งหมดหรือ ได้รับ Object File ที่ต้องการ

2.8 วงจรโมโนสเตเบิ้ลโดยใช้ไอซีไทม์เมอร์ 555

วงจรรวมตัวหนึ่งที่เป็นที่นิยมมาก มีการใช้งานสะดวก นำไปประยุกต์ใช้กับวงจรต่างๆ ได้ง่าย นั่นคือ ไอซี 555 มีการทำงานอยู่ 2 ประเภทคือ ประเภทการสร้างสัญญาณตลอดเวลาหรือที่เรียกว่า อะสเตเบิลมัลติไวเบรเตอร์ และประเภทสร้างสัญญาณเพียง 1 ลูกคลื่นหรือที่เรียกว่า วันชอตมัลติไวเบรเตอร์ หรือ โมโนสเตเบิลมัลติไวเบรเตอร์



ภาพที่ 2.7 แสดงองค์ประกอบภายในไอซี 555

การทำงานของวงจรประเภทอะสเตเบิลของไอซี 555 จะสังเกตเห็นได้ว่าสัญญาณแรงดัน ทางเอาท์พุทจะเปลี่ยนค่าแรงดันสูง ต่ำ เช่นนี้ไปตลอดคาบสัญญาณไม่มีหยุด ช่วงเวลาของ 1 คาบสัญญาณทางเอาท์พุตนี้จะหาได้จากค่าตัวต้านทาน และตัวเก็บประจุที่นำมาต่อกับไอซี 555 ข้อสังเกตที่ไม่ควรลืมก็คือ ค่าแรงดันเอาท์พุตในขณะที่มีค่าต่ำสุดจะมีประมาณ 0.1 โวลท์

2.8.1 หน้าที่ของขาไอซี 555

1. ขากราวด์และขาไฟเลี้ยงจากรูปร่างของไอซี 555 เป็นแบบตัวถัง มีตัวถังให้เลือกอยู่ 2 ชนิด คือ ชนิด TO-99 และ DIP แบบ 8 ขา ดังภาพที่ 2.8 ซึ่งจะเห็นขาที่ 1 จะเป็นกราวด์ขาที่ 8 เป็นไฟเลี้ยง โดยไฟเลี้ยงนี้จะสามารถป้อนค่าได้ตั้งแต่ +5 ถึง + 8 โวลท์ ซึ่งนั่นหมายความว่า เราสามารถประยุกต์ใช้ไอซี 555 เข้ากับวงจรดิจิตอลคอมพิวเตอร์ทั่วไป จนถึงวงจรสำคัญต่างๆ หรือวงจรที่ใช้ภายในรถยนต์ก็ได้ด้วยเช่นกัน ตัวไอซีวงจรภายในต้องการกระแสไฟเลี้ยงประมาณ 0.7 มิลลิแอมป์ต่อโวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(เช่นต้องการ 10 มิลลิแอมป์สำหรับไฟเลี้ยง +15 โวลท์ เป็นต้น) กำลังสูญเสียภายใน (ซึ่งมักจะกลายเป็นความร้อน) จะที่ค่าสูงสุดเพียง 600 มิลลิวัตต์

2. ขาเอาท์พุตคือขาที่ 3 ดังภาพที่ 2.8 ที่ขาเอาท์พุตหรือขาที่ 3 เป็นไปได้ทั้งตัวดึงกระแสเข้า (Sink current)และตัวผลักกระแสออก(Source current)โดยเมื่อสัญญาณที่เอาท์พุต มีค่าต่ำไฟเลี้ยงจากวงจรภายนอกก็จะไหลเข้าสู่ตัวไอซี 555 โดยขณะนั้นซับพลายโหลด(floating supply load)ซึ่งเสมือนถูกต่อลอยอยู่จะเริ่มทำงานโดยทำหน้าที่เป็นตัวผ่านกระแสจากไฟเลี้ยงภายนอก เข้าสู่ตัวไอซี และเมื่อสัญญาณที่เอาท์พุทมีค่าสูงตัวไอซี 555 ก็จะทำตัวเป็นผู้ให้กระแสโดยปล่อยกระแสให้ไหลออกไปสู่ กราวด์โหลด (grounded load) หรือโหลดภายนอกที่ต่ออยู่กับขานั่นเอง ข้อสังเกตุ ประการหนึ่งก็คือ การทำงานปกติ ณ เวลาหนึ่งๆ จะเป็นการสวิทช์หรือเลือกใช้ระหว่างซับพลายโหลดและกราวด์โหลดอย่างใดอย่างหนึ่งเสมอ และไม่ค่อยจะมีวงจรประยุกต์ใดที่ใช้กราวด์และซับพลายโหลดพร้อมกันในเวลาเดียวกัน

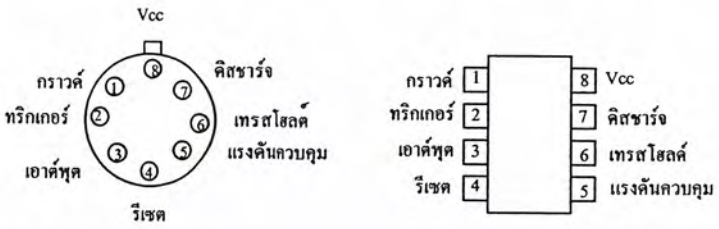
ข้อสังเกตอีกประการหนึ่ง กระแสที่ดึงเข้าจ่ายออกจากไอซีจะมีค่าตามคู่มือประมาณ 200 มิลลิแอมป์ แต่เวลาใช้จริงจะมีค่าประมาณ 40 มิลลิแอมป์ โดยขณะที่ขาเอาท์พุทมีสภาวะค่าสูง จะได้ค่าแรงดันทางเอาท์พุตต่ำกว่าไฟเลี้ยงที่จ่ายให้ไอซีประมาณ 0.5 โวลท์ และสภาวะที่เอาท์พุทมีค่าต่ำจะให้ค่าแรงดันที่ขาเอาท์ประมาณ 0.1 โวลท์ เมื่อเทียบกับกราวด์โดยคงกระแสเข้าประมาณ 25 มิลลิแอมป์

3. ขารีเซตคือขาที่ 4 ซึ่งทำให้ 555 สามารถยกเลิกการทำงาน หรือสั่งทำงานซ้ำต่อสัญญาณคำสั่งที่มาจากวงจรทริกเกอร์จากขาอินพุท (trigger input)ในขณะที่ไม่ได้ใช้ขานี้ควรที่จะต่อเข้ากับไฟเลี้ยงบวก และถ้าเมื่อใดที่ขารีเซตนี้ต่อกับกราวด์หรือมีระดับแรงดันน้อยกว่า 0.4 โวลท์ ทั้งขาที่ 3 และ 7 (ขาอินพุทและขาดิสชาร์ต) ก็จะมีค่าแรงดันต่ำทันที

4. ขาดิสชาร์ตคือขาที่ 7 หน้าที่ของขานี้จะเหมือนกับชื่อของขา กล่าวคือเมื่อสัญญาณเอาท์พุทมีค่าสภาวะต่ำ ขานี้ก็จะใช้เพื่อการดิสชาร์ตกระแสของตัวเก็บประจุที่ต่อที่ขา 7 (ตัวเก็บประจุจะทำหน้าที่เกี่ยวกับการหน่วงเวลา) และเมื่อสัญญาณเอาท์พุตมีสภาวะค่าสูง ขานี้ก็จะทำหน้าที่เปิดวงจร และอนุญาตให้เก็บประจุ เกิดการชาร์ตกระแสจากแหล่งจ่ายไฟภายนอกเข้าตัวเก็บประจุ จนกระทั่งแรงดันของตัวเก็บประจุมีค่าที่กำหนดไว้

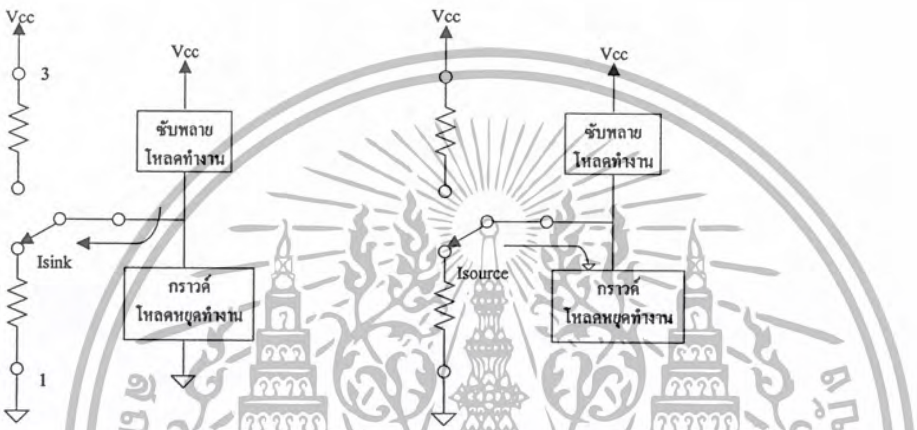
5. ขาควบคุมแรงดันคือขาที่ 5 ตัวเก็บประจุที่ทำหน้าที่เป็นฟิลเตอร์ขนาด 0.01 ไมโครฟาหรัดมักจะถูกนำมาต่อที่ขานี้กับกราวด์ ตัวเก็บประจุที่นำมาต่อที่ขานี้ จะทำหน้าที่บายพาส หรือส่งสัญญาณรบกวนและแรงดันกระเพื่อมที่มาจากแหล่งจ่ายไฟ(ripple voltage)ลงกราวด์ไป ทั้งนี้ทำให้แรงดันเทรสโฮลด์มีค่าถูกต้องหรือใกล้เคียงกับค่าที่เราออกแบบมากที่สุด สำหรับ คำว่าแรงดันเทรสโฮลด์นี้อาจแปลให้เข้าใจง่ายๆ ว่าแรงดัน ณ ค่าหนึ่งที่เริ่มทำให้เกิดการเปลี่ยนแปลงสถานะของเอาท์พุท นอกจากนี้ขาควบคุมแรงดันนี้ยังสามารถนำมาใช้เพื่อเปลี่ยนแปลงระดับแรงดันที่ขาเทรสโฮลด์และขาทริกเกอร์ได้อีกด้วย ยกตัวอย่างเช่น ถ้าเรานำตัวต้านทานขนาด 10 กิโลโอห์มมาต่อระหว่างขา 5 และขา 8 (จะเสมือนต่อต้านทานขนานจากเดิม 2/3 Vcc ไปเป็น 0.75 Vcc และแรงดันทริกเกอร์เปลี่ยนไปเป็น 0.375 Vcc จุดนี้เองที่ทำให้เราสามารถกำหนดสัดส่วนของเทรสโฮลด์ใหม่ได้ (แต่เดิมจะเป็น 2/3 และ 1/3 ของ Vcc)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตัวถังโลหะ(มองด้านบน) ตัวถังแบบดิพ(มองด้านข้าง)

ภาพที่ 2.8 แสดงตำแหน่งขาของไอซี 555



ในขณะที่เอาต์พุต Low ในขณะที่เอาต์พุต High

ภาพที่ 2.9 แสดง การทำงานของกราวด์โหลด และซับพลายโหลด ขณะที่ต่อเข้ากับวงจรไอซี 555

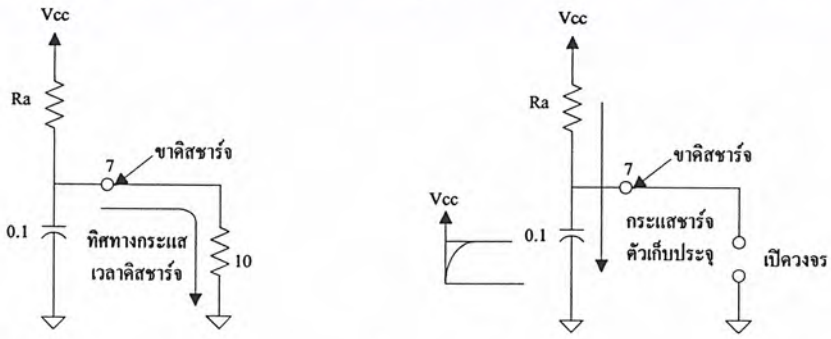
6. ขาทริกเกอร์คือขาที่ 2 และเทรสโฮลด์คือขาที่ 6 ไอซี 555 มีสถานการณ์ทำงาน 2 สถานะและมีสถานการณ์จำอยู่ 2 สถานะ (รวมเป็น 4 สถานะ) โดยสถานะต่างๆ เหล่านี้จะขึ้นอยู่กับขาทั้งสอง ก็คือ ขาทริกเกอร์ (ขาที่ 2) และขาเทรสโฮลด์ (ขาที่ 6) สัญญาณอินพุตที่ป้อนให้กับขาทริกเกอร์จะถูกเปรียบเทียบกับคอมพาราเตอร์หมายเลข 2 ในรูป 2.8 ซึ่งถูกกำหนดให้เปรียบเทียบที่ค่าเทรสโฮลด์ค่าต่ำหรือ V_{LT} (lower threshold voltage) ซึ่งจะมีค่าเท่ากับ Vcc/3 สำหรับสัญญาณอินพุทที่ป้อนให้กับเปรียบเทียบกับคอมพาราเตอร์หมายเลข 1 ซึ่งถูกกำหนดให้เปรียบเทียบที่ค่าเทรสโฮลด์ค่าสูง (เราอาจจะเรียกค่าแรงดันค่าสูงที่เริ่มทำให้เกิดการเปลี่ยนแปลงต่อสถานะเอาท์พุท ซึ่งเป็นชื่อที่เข้าใจง่ายแต่ก็ยาวมาก ดังนั้นจึงของใช้การทับศัพท์แทน) ซึ่งจะมีค่าเท่ากับ 2 Vcc/3

สัญญาณอินพุตที่ป้อนแต่ละครั้งค่าแรงดันที่ป้อนนั้นเป็นไปได้ที่จะมีค่าแรงดันมากกว่าหรือน้อยกว่า หรือเท่ากับค่าเทรสโฮลด์ทั้งสอง อันจะทำให้เกิดสถานะ 4 สถานะขึ้นมาดังที่เราได้กล่าวมาแล้ว โดยเมื่ออยู่ในการทำงานแบบ A แรงดันอินพุตที่ป้อนเข้ามามีค่าต่ำกว่าแรงดันอ้างอิง ทั้งขาทริกเกอร์และขาเทรสโฮลด์ ดังนั้นแรงดันที่ขาเอาท์พุทของไอซี 555 ก็จะให้ค่าสูง ในขณะที่การทำงานแบบ D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าฉลาดกระบัง

(ดังตาราง 2.6) แรงดันอินพุทที่ป้อนเข้ามามีค่าสูงกว่าแรงดันอ้างอิงทั้งสอง แรงดันที่ขาเอาท์พุตก็จะให้ค่าต่ำ

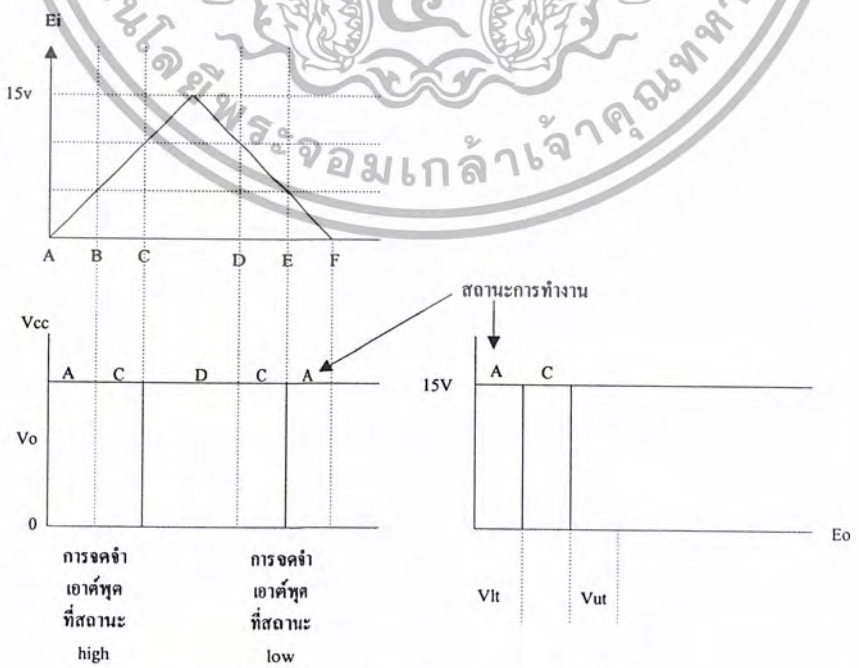


ภาพที่ 2.10 การทำงานของขาดิสชาร์ต

สถานะการทำงาน	ขาทริกเกอร์ 2	ขาเทรสโฮลด์	ผลลัพธ์	
			ขาเอาต์พุต 3	ขาดิชชาร์จ 7
A	ต่ำกว่า V_{LT}	ต่ำกว่า V_{UT}	High	ปิดวงจร
B	ต่ำกว่า V_{LT}	สูงกว่า V_{UT}	จำสถานะเดิมต่อไป	
C	สูงกว่า V_{UT}	ต่ำกว่า V_{UT}	จำสถานะเดิมต่อไป	
D	สูงกว่า V_{UT}	สูงกว่า V_{UT}	Low	กราวด์

ตารางที่ 2.6 สถานะการทำงานของไอซี 555 : $V_{UT} = 2(V_{CC})/3, V_{LT} = V_{CC}/3$, High = V_{CC}, Low หรือ กราวด์ = 0

ข้อสังเกตอีกประการ เมื่อเราป้อนอินพุทค่าแรงดันต่ำเราจะได้ค่าแรงดันเอาท์พุทออกมาเป็นค่าสูง ในทางกลับกันเมื่อเราป้อนอินพุทค่าแรงดันสูง เราจะได้เอาท์พุตออกมาเป็นค่าแรงดันต่ำตรงนี้อาจทำให้เราไอซี 555 เสมือนเป็นตัวอินเวอร์เตอร์ หรือตัวกลับสัญญาณ



ภาพที่ 2.11 สถานะการทำงานของไอซี 555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50111

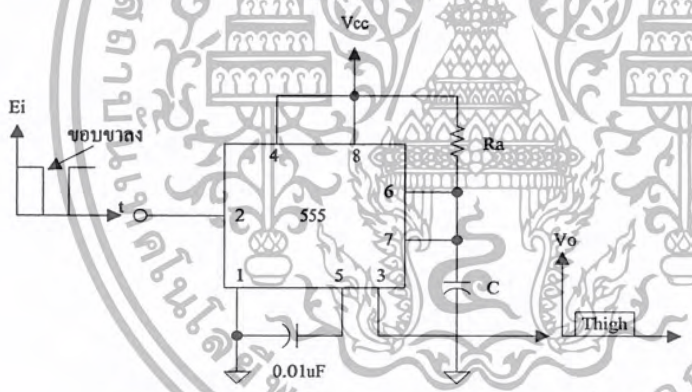
พิจารณาจากตารางที่ 2.6 สำหรับกรณีการทำงานแบบ B และ C นั้นจะเป็นสถานะชนิดที่ไอซี 555 ยังคงจดจำสถานะเดิมอยู่ โดยเมื่อแรงดันที่ป้อนเข้ามาทั้งสองลดลงจากค่า V_{UT} หรือสถานะ C วงจรจะยังคงมีการจดจำค่าสถานะเดิมเอาไว้อยู่เพื่อให้ง่ายต่อความเข้าใจมากขึ้น

ข้อสังเกต การที่เกิดสถานะจะจำขึ้นในสถานะ Bและ C นี้จะทำให้เกิดปรากฏการณ์ที่เรียกว่า ฮิสเตอร์ริซิสลูป (hysteresis loop)ขึ้น และการที่วงจรสามารถจดจำสถานะนี้ได้ แสดงว่าวงจรนี้มีการแปลงหน่วยความจำด้วย ข้อสังเกตอีกประการหนึ่งคือ ในขณะที่วงจรกำลังอยู่ในสถานะจดจำนี้เราจะไม่มีทางรู้สถานะต่อไปของเอาท์พุทได้เลย นอกจากเราจะรู้ว่าสถานเดิมก่อนเข้าสู่สภาวะจดจำนี้ ซึ่งก็ดูได้จากแรงดันเอาท์พุทปัจจุบันนั่นเอง

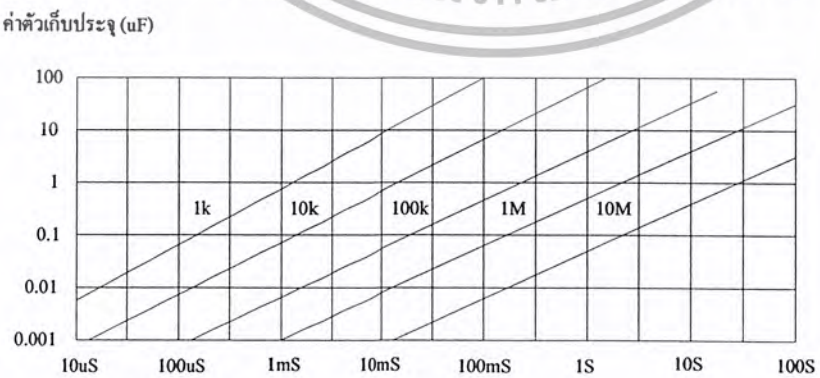
2.8.2 หลักการทำงานของวงจรโมโนสเตเบิ้ล

พิจารณาภาพที่ 2.12 (ก) แสดงวงจรในลักษณะใช้สัญญาณเพียง 1 ลูกคลื่น หรือวงจรโมโนสเตเบิ้ล (แปลว่าวงจรที่ให้ลูกคลื่นลูกเดียว แล้วกลับสู่สถานการทำงานปกติคงที่เหมือนเดิม) เมื่อเราป้อนพัลส์ขอบขาลบให้กับขาที่ 2 ของไอซี 555 จะได้เอาท์พุทมีค่าสูงที่ขา 7 จะยกเลิกการลัดวงจรตัวเก็บประจุ C ก็เริ่มทำการเก็บประจุโดยจะมีแรงดันเพิ่มขึ้นจาก 0 ไปเป็น 2/3 Vcc ซึ่งจะทำให้ตัว คอมพาราเตอร์ที่ 1 ดังรูป 2.12 ทำให้เอาท์พุทเปลี่ยนจากแรงดันสูงกลับสู่แรงดันต่ำเหมือนเดิม ดังภาพที่ 2.12 (ก) สามารถคำนวณเวลาของลูกคลื่นได้ดังนี้

$$t_{high} = 1.1\, R_A\, C$$



(ก) ในการออกแบบนี้ ค่า Pulse หรือ Ei ต้องมีค่าแรงดันสูงกว่า 2/3 Vcc



(ข) กราฟช่วยการออกแบบ

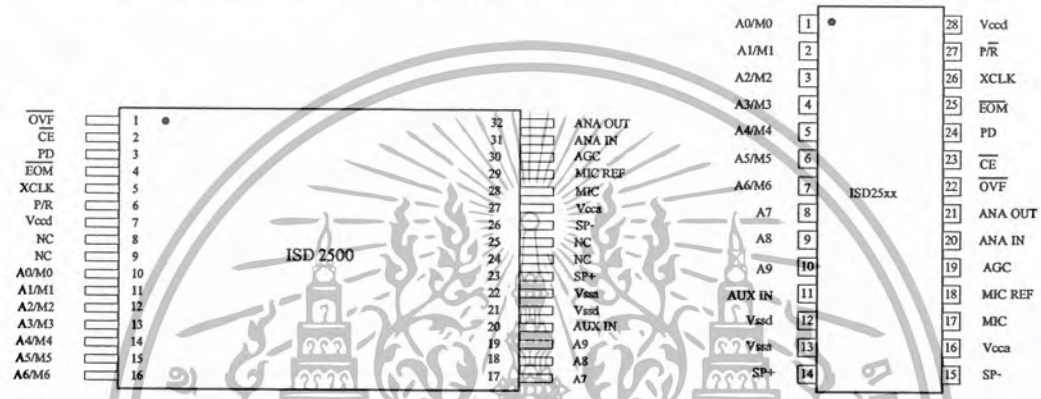
ภาพที่ 2.12 แสดงการทำงานของวงจรโมโนสเตเบิ้ล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.12 (ข) เกิดจากการวาดกราฟการทำงานของวงจรรูป (ก) เมื่อมีการเปลี่ยนค่า R_A และ C ซึ่งกราฟจะมีการประโยชน์กับผู้ที่ต้องการการออกแบบแต่ไม่ชอบการคำนวณอย่างมาก

2.9 ไอซีบันทึกเสียง ISD 2590 (ตระกูล 25XX)

ไอซีบันทึกเสียงมีหลายตระกูล มีการพัฒนาหลายๆ ด้านไม่ว่าจะเป็นการเพิ่มระยะเวลาในการบันทึกให้มากขึ้นการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล เป็นต้น ต่อไปนี้เราจะมาดูรายละเอียดบางตระกูล ไอซีบันทึกเสียงมีอยู่หลายเบอร์ด้วยกัน ได้แก่ ไอซีตระกูล 12XX และ 14XX ซึ่งแตกต่างของไอซี ตระกูล 25XX กับตระกูล 12XX และ 14XX ที่เห็นได้ชัดคือ ระยะเวลาในการบันทึกนานกว่ามาก และไม่ต้องใช้อุปกรณ์ที่เป็นภาคขยายเสียงต่อรวมภายนอกสามารถขับลำโพงได้โดยตรง



ภาพที่ 2.13 ลักษณะการจัดขาของ ISD 2590

การป้อนสัญญาณเสียงจะใช้ไมโครโฟนไดนามิคไมโครโฟน หรือ คอนแดนเซอร์ไมโครโฟนก็ได้ ลักษณะตัวถังของ ISD 2590 ดังภาพที่ 2.13 เป็นตัวถังแบบ DLP/SOIC และแบบ TSOP สำหรับการใช้งานในแบบทั่วไปแล้วตัวถังบรรจุแบบ DLP/SOIC น่าจะใช้งานง่านกว่าแบบ TSOP

2.9.1 คุณสมบัติของ ISD 2590

คุณสมบัติของ ISD 2590 มีหลายประการด้วยกัน คือ

1. สามารถบันทึก และเล่นกลับได้อย่างสะดวก และง่ายดาย
2. ไม่ต้องใช้อุปกรณ์ประเภทอื่นๆ ประกอบร่วมภายนอก
3. ประสิทธิภาพในการบันทึกและเล่นกลับที่ให้เสียงได้เหมือนต้นกำเนิดเสียง
4. สามารถควบคุมการบันทึกและเล่นกลับ ด้วยสวิตช์หรือจากการควบคุมด้วยไมโครคอนโทรลเลอร์
5. ระยะเวลาในการบันทึกค่อนข้างนานคือ 90 วินาที
6. สามารถต่อคาสเคดได้โดยตรง เพื่อเพิ่มระยะเวลาในการบันทึกให้มากขึ้น
7. ปิดการทำงานโดยอัตโนมัติ เมื่อไม่มีการบันทึกหรือ เล่นกลับนานเกินไป
8. สามารถเก็บความจำได้นานมาก โดยไม่ต่อแบตเตอรี่ (สามารถเก็บได้นานเป็นสิบๆ ปี)
9. วงรอบของการบันทึกสูงถึง 100,000 ครั้ง

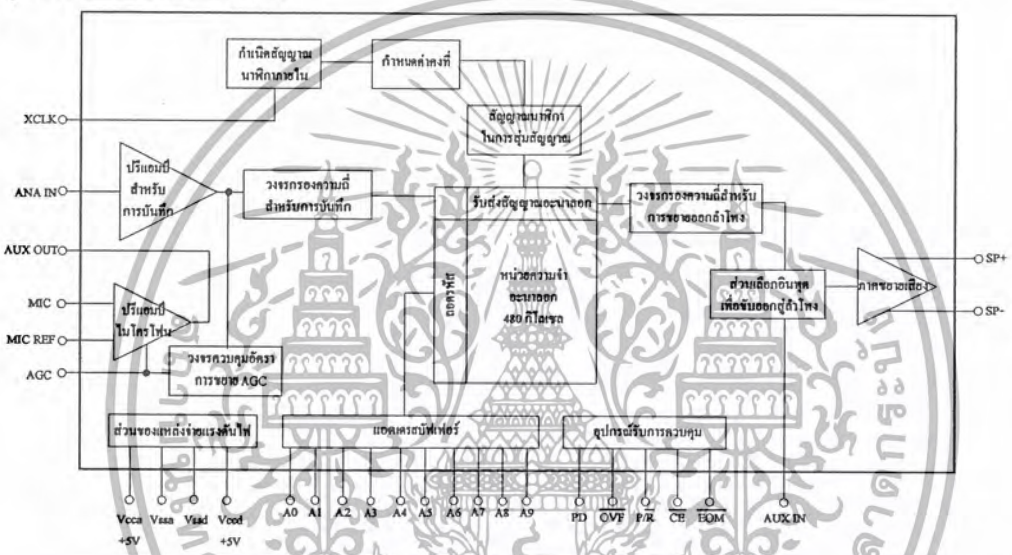
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. มีวงจรกำเนิดสัญญาณนาฬิกาภายในตัว

11.สามารถโปรแกรมควบคุมการเล่นกลับได้เพียงอย่างเดียว

รูปแบบการใช้งานตั้งแต่วงจรขยายสัญญาณจากไมโครโฟน จนถึงหน่วยจัดเก็บข้อมูลที่ทำการบันทึกและขับข้อมูลออกลำโพงก็ถูกรวบรวมไว้ในไอซีเพียงตัวเดียว นั้นคือ ไอซีบันทึกเสียงตระกูล ISD 25XX

ส่วนในโหมดการบันทึกจะจัดเก็บข้อมูลต่างๆไว้ในหน่วยความจำที่เป็นเซลไม่ต้องการแรงดันสำรองเพื่อรักษาข้อมูลไม่ให้หาย (non-volatile memory cells)สัญญาณเสียงที่อยู่ในรูปของสัญญาณอนาลอกจะถูกบันทึกไว้ในหน่วยความจำโดยตรง โดยอาศัยเทคโนโลยี DAST (Direct Analog Storage Technology) และการจัดเก็บหน่วยความจำก็จะจัดเก็บใส่ลักษณะที่เป็นสัญญาณอนาลอกอยู่เช่นเดิม จึงทำให้การเล่นกลับสามารถให้สัญญาณเสียงที่เหมือนกับต้นกำเนิดเสียงมากเพราะไม่มีกระบวนการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอลเข้ามาเกี่ยวข้อง



ภาพที่ 2.14 บล็อกไดอะแกรมภายในไอซี

พิจารณาในภาพที่ 2.14 เป็นภาพที่แสดงบล็อกไดอะแกรมภายในของไอซี ISD 2590 เมื่อพิจารณาดูบล็อกไดอะแกรมแล้วก็มีลักษณะคล้ายคลึงกับตระกูล ISD 12XX / 14XX มาก หากแต่มีความแตกต่างกันในส่วนของบล็อกแอดเดรสบัฟเฟอร์ และบล็อคส่วนรับการควบคุม นอกจากนั้นยังมีบล็อกมัลติเพล็กเซอร์สัญญาณอินพุทของเพาเวอร์แอมป์ภายในตัวไอซี เพื่อทำการเลือกที่จะขยายสัญญาณที่ถูกบันทึกไว้ หรือขยายสัญญาณจากภายนอกที่ขา AUX/IN ทั้งหมดเป็นข้อแตกต่างของ ISD 2590 ที่ไม่เหมือนกับ ISD 12XX / 14XX นอกจากนั้นอัตราการทำงานของไอซีตระกูล ISD 12XX / 14XX ก็แตกต่าง ดังจะแสดงข้อมูลทางการบันทึกสัญญาณของไอซีในตระกูลไว้ในตารางที่ 2.7

2.9.2 หลักการทำงานของไอซีบันทึกเสียงเบื้องต้น

หลักการทำงานเบื้องต้นของไอซีนั้นต้องทำความเข้าใจ หรือทราบรายละเอียดคุณสมบัติต่างๆทางเทคนิคของไอซีตระกูลนี้เสียก่อน รายละเอียดแสดงในตารางที่ 2.7 คุณสมบัติทางไฟฟ้าบางอย่างที่แตกต่างกันของไอซีตระกูล ISD 2590 รายละเอียดในตารางนี้มีความสำคัญมากต่อการใช้งาน เพราะจะทำให้การทำงานของวงจรถูกต้อง อีกทั้งสร้างความปลอดภัยและมั่นใจในการทำงานเบื้องต้น อีกทั้งสามารถประยุกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานต่างๆ ได้โดยอาศัยจากข้อมูลเบื้องต้นนี้ เพราะฉะนั้นเราจะต้องศึกษาและพิจารณาข้อมูลเหล่านี้ให้ดีๆ ถ้าเกิดปัญหาจากการประกอบวงจร เราก็สามารถที่จะแก้ไขปัญหานี้ได้อย่างถูกต้อง

เบอร์ไอซี	ระยะเวลาบันทึก	การสุ่มสัญญาณทางอินพุต	ความถี่ที่ผ่านวงจรกรอง	ความถี่สัญญาณภายใน
ISD2545	45 วินาที	10.6 กิโลเฮิรตซ์	4.5 กิโลเฮิรตซ์	1,365.3 กิโลเฮิรตซ์
ISD2560	60 วินาที	8.0 กิโลเฮิรตซ์	3.4 กิโลเฮิรตซ์	1,024 กิโลเฮิรตซ์
ISD2575	75 วินาที	6.4 กิโลเฮิรตซ์	2.7 กิโลเฮิรตซ์	819.2 กิโลเฮิรตซ์
ISD2590	90 วินาที	5.33 กิโลเฮิรตซ์	2.3 กิโลเฮิรตซ์	682.7 กิโลเฮิรตซ์

ตารางที่ 2.7 คุณสมบัติทางไฟฟ้าบางอย่างที่แตกต่างกันของไอซีตระกูล ISD 25XX

พารามิเตอร์	สัญลักษณ์	ค่า	หน่วย
แรงดันอินพุตด้านต่ำ "0"	V_L	0.8	โวลต์
แรงดันอินพุตด้านสูง "1"	V_H	2	โวลต์
แรงดันอินพุตด้านต่ำ	V_{CL}	0.4	โวลต์
แรงดันอินพุตด้านสูง	V_{CH}	V_{CC} -0.4	โวลต์
แรงดันอินพุตด้านสูงที่ขา OVF	V_{CH1}	2.4	โวลต์
แรงดันอินพุตด้านสูงที่ขา EOM	V_{CH2}	V_{CC} -1.0	โวลต์
กระแสของแรงดันไฟเลี้ยงที่ Vcc = 5 โวลต์	I_{CC}	25	มิลลิแอมป์
กระแสขณะสแตนด์บายที่V cc = 5 โวลต์	I_{SB}	1.10	ไมโครแอมป์
กระแสรั่วไหลทางอินพุต	I_L	± 1	ไมโครแอมป์
อิมพีแดนซ์โหลดเอาต์พุต	R_{EC}	16	โอห์ม
ความต้านทานอินพุตของปรีแอมป์ไมโครโฟน	R_{MIC}	10	กิโลโอห์ม
ความต้านทานอินพุตของขาอินพุตภายนอก	R_{AUX}	10	กิโลโอห์ม
ความต้านทานอินพุตของขาอินพุตอะนาลอก	$R_{ANA\ IN}$	3	กิโลโอห์ม
อัตราการขยายของปรีแอมป์ 1	Rpre1	24	เดซิเบล
อัตราการขยายของปรีแอมป์ 2	$A_{PRE\ 2}$	5	เดซิเบล
อัตราการขยายของขา AUX (สัญญาณภายนอก)	A_{AUX}	1	โวลต์ต่อโวลต์
อัตราการขยายของภาคขยายเอาต์พุตลำโพง	A_{APP}	22	เดซิเบล
ความต้านทานเอาต์พุตของขา AGC	R_{AGC}	5	กิโลโอห์ม
แรงดันไฟเลี้ยงตัวไอซีทั้งหมด	V_{CC}	5.7	โวลต์
อุณหภูมิขณะทำงาน	T_S	-65 – 150	องศาเซลเซียส

ตารางที่ 2.8 คุณสมบัติทางไฟฟ้าของ ISD 2590

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Adress / Mode Input (A0 – A9/M6) ขา 1-10 ขาแอดเดรสแลโหมดอินพุทจะมีสองฟังก์ชั่นที่ขึ้นอยู่กับระดับของสอง MSB (ขา A8,A9) ของแอดเดรส ถ้าแอดเดรสใดแอดเดรสหนึ่งของสอง MSB เป็น "0" อินพุทก็จะปรากฏที่แอดเดรสบิททั้งหมดและใช้เป็นแอดเดรสเริ่มต้นสำหรับวงรอบการบรรทุกและเล่นกลับ และขาแอดเดรสจะเกิดการแลทช์โดยขอบขาลงของพัลส์ที่ขา CE และถ้า MSBs มีสถานะเป็น "1" ขาแอดเดรส/โหมดอินพุท จะมาขึ้นที่โหมดบิททั้งหมดและเกิดการแลทช์เมื่อพัลส์ของขาลงปรากฏที่ CE

2. Auxiliary Input (AUX IN) ขา 11 จะเป็นขารับอินพุทจากภายนอกซึ่งเป็นมัลติเพล็กซ์สัญญาณผ่านออกไปทางเอาท์พุทของวงจรขยายภายในและขับออกสู่ขาเอาท์พุทลำโพง โดยขั้นตอนการทำงานนี้จะเกิดขึ้นเมื่อขา CE มีสถานะเป็น "1" วงจรรอบของการเล่นกลับก็จะสิ้นสุด ถ้ามีการต่อคาสเคด ISD 2590 กันหลายๆ ตัว ขา AUX IN จะถูกต่อเข้ากับสัญญาณเล่นกลับที่ออกมาจากขาเอาท์พุทลำโพงของตัวก่อนหน้านี้

3. Ground Input (Vssa , Vssd) ขา 12 และ 13 โดยคุณสมบัติของไอซีดิจิตอล ขากราวด์ทั้งสองนี้จะถูกต่อและปิดไว้ภายในตัวถังบรรจุของไอซี การใช้งานกราวด์ทั้งสองนี้จะเลือกต่อกับกราวด์ของเพาเวอร์ซัพพลายในส่วนที่มีค่าอิมพิแดนซ์ต่ำ เพื่อไม่ต้องการให้เกิดแรงดันที่แตกต่างกันระหว่างกราวด์ทั้งสอง

4. Speaker Output (Sp+ , Sp-) ขา 14 และ 15 เป็นขาเอาท์พุทต่อออกลำโพงในตระกูล ISD 25XX นี้จะมีวงจรขับสัญญาณความแตกต่างออกสู่ลำโพง ซึ่งประกอบอยู่ในตัวไอซีเรียบร้อยแล้ว โดยมีความสามารถในการขับลำโพงเอาท์พุทได้ 50 มิลลิวัตต์ ที่โหลดลำโพง 16 โอห์ม ขาต่อออกลำโพงทั้งสองนี้จะใช้ต่อขนานกันโดยตรงเด็ดขาดเมื่อต้องถูกใช้ต่อคาสเคดกันหลายๆ ตัว และไม่เหมาะในการต่อลำโพงขนานกันทางเอาท์พุทหลายตัว โดยเฉพาะในบางครั้งขาเอาท์พุทลำโพงสามารถต่อคาสเคดกับไอซีอีกตัวได้โดยตรง เพราะมีตัวเก็บประจุคับปลิ้งอยู่ภายในแล้ว

5. Voltage Input (Vcca , Vccd) ขา 16 และขา 28 เป็นแรงดันที่จะต้องแยกกันต่างหากระหว่างขารับแรงดันของวงจรอะนาลอกและวงจรดิจิตอล ที่ประกอบอยู่ภายในตัวได้แล้ว ขารับแรงดันต้องการไฟเลี้ยง +5 โวลท์ และต้องเป็นแรงดันไฟเลี้ยงที่มีสัญญาณรบกวนต่ำมาก

6. Microphone Input (MIC) ขา 17 จะรับสัญญาณอินพุทที่ผ่านเข้ามายังไมโครโฟนแล้วส่งผ่านสัญญาณเข้าสู่วงจรปรีแอมป์ที่ประกอบอยู่ภายในตัวไอซี ภายในประกอบด้วยวงจรควบคุมอัตราการขยายอัตโนมัติ (AGC) โดยวงจรนี้จะทำหน้าที่ควบคุมอัตราการขยายของวงจรปรีแอมป์ให้มีอัตราการขยายอยู่ในช่วง -15 ถึง 24 เดซิเบล ไมโครโฟนภายนอกจะถูกคับปลิ้งผ่านตัวเก็บประจุภายนอกในลักษณะอนุกรมกับขา 17 นี้ ค่าความจุของตัวเก็บประจุคับปลิ้งจะกำหนดค่าโดยคำนึงถึงค่าความต้านทาน 10 กิโลโอห์มที่ต่ออยู่ภายในกับขา 17 ของไอซีเพื่อเกิดการคัทออฟที่ความถี่ต่ำ

7. Microphone Reference Input (MIC REF) ขา 18 นี้เข้ากับกราวด์อนาลอก (VSSa)โดยมีตัวเก็บประจุต่ออนุกรมอยู่ก่อน เพื่อทำหน้าที่กำจัดสัญญาณรบกวนทางอินพุทขา 17 และเพื่อให้เกิดการชดเชยทางด้านสัญญาณรบกวนให้ดีกว่า 10 เดซิเบล

8. Automatic Gain Control Input (AGC) ขา 19 เป็นขาควบคุมเพื่อควบคุมการขยายของปรีแอมป์ไมโครโฟนทางด้านไดนามิคเพื่อให้เกิดความเหมาะสมกับระดับสัญญาณที่มีย่านความกว้างมากของสัญญาณทางอินพุทจากไมโครโฟน และเพื่อให้ระดับสัญญาณที่ทำการบันทึกมีความผิดเพี้ยนน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา AGC นี้จะต้องต่อร่วมกับอุปกรณ์ RC เพื่อกำหนดค่าเวลาคงที่โดยมีค่าความต้านทานภายใน 5 กิโลโอห์ม และจะต่อร่วมกับตัวเก็บประจุภายนอกตัวหนึ่ง ผ่านลงกราวด์อนาลอก ค่าที่เหมาะสม บางครั้งกำหนดไว้ที่ค่าความต้านทาน 470 กิโลโอห์ม และตัวเก็บประจุ 4.7 ไมโครฟาหรัด

9. Analog Input (ANA IN) ขา 20 จะรับสัญญาณที่ผ่านวงจรปรีแอมป์ออกมาทางขา 21 โดยผ่านตัวเก็บประจุคับปลิ้งภายนอกคับปลิ้งสัญญาณเข้าที่ขา 20 นี้ เพื่อผ่านสัญญาณเข้าไปทำการบันทึกไว้ใน ตัวไอซี ตัวเก็บประจุคับปลิ้งภายนอกนี้จะต้องสัมพันธ์กัน

10. Analog Output (ANA OUT) ขา 21 เป็นขาเอาท์พุทของวงจรปรีแอมป์ขยายสัญญาณจากไมโครโฟนที่ได้รับการควบคุมอัตราการขยายจากวงจร AGC ภายในแล้ว

11. Overflow Output (OVF) ขา 22 สัญญาณพัลส์ "O" จะปรากฏออกมาทางขาเอาท์พุทนี้เพื่อเป็นการแสดงว่าสิ้นสุดการเล่นกลับหรือหน่วยความจำภายในตัวไอซีถูกอ่านออกมาหมดแล้ว และจะแสดงเป็นสภาวะการหยุดเล่นกลับพัลส์เอาท์พุทจากขา OVF ให้กับขา CE อินพุท จนกว่าขา PD จะได้รับพัลส์เพื่อทำการรีเซตและเริ่มวงรอบการเล่นกลับใหม่อีกครั้ง พัลส์ที่ขา OVF นี้สามารถใช้เริ่มต้นทำงานของ ISD 2590 ในตัวถัดไปได้ เมื่อถูกต่อคาสเคดกันอยู่หลายตัว

12. Chip Enable Input (CE) ขา 23 จะต้องได้รับสัญญาณพัลส์ "O" เพื่อทำให้เกิดการเปลี่ยนแปลงระหว่างการเล่นกลับและการบันทึก ที่ขาแอดเดรสอินพุทและขา P/R อินพุทจะถูกแลทช์จากพัลส์ขอบขาลงของพัลส์ที่ขา CE

13. Power Down Input (PD) ขา 24 ในขณะที่ไม่มีการบันทึกหรือเล่นกลับที่ขา PD จะมีสภาวะเป็น "1" ก็จะเป็นการรักษาระดับการสิ้นเปลืองกำลังงานในระดับต่ำมากๆ แต่เมื่อขาใช้กับงานทางด้านโทรศัพท์ทั่วไป OVF มีสภาวะเป็น "0" ที่แสดงถึงการเล่นกลับสิ้นสุดลงปรากฏขึ้น ขา PD ปกติเป็น "1" อยู่ในขณะนั้น ก็จะถูกรีเซตและเริ่มกระบวนการบันทึกหรือเล่นกลับใหม่อีกครั้ง

14. End – of – Message / RUN Output (EOM) ขา 25 เป็นส่วนของอุปกรณ์ non- volatile ภายในตัวไอซีที่ใช้กำหนดหรือระบุการสิ้นสุดของการเก็บข้อมูลที่ทำการบันทึก ขา EOM นี้จะให้เอาท์พุทออกเป็น "0" เมื่อข้อมูลที่บันทึกอยู่ถูกเล่นกลับออกมาหมดแล้ว

15. External Clock Input (XCLK) ขา 26 เป็นขารับสัญญาณนาฬิกาภายนอกเพื่อกำหนดค่าความถี่สัญญาณนาฬิกาในการสุ่มสัญญาณ แต่โดยปกติได้ระบุว่าสัญญาณนาฬิกาการสุ่มสัญญาณถูกกำหนดๆ ไว้ภายในแล้ว ซึ่งจะไม่ขึ้นกับอุณหภูมิภายนอกหรือย่านแรงดันไฟเลี้ยงที่ไม่คงที่ การใช้งานปกติแล้วจะต่อ ขา 25 นี้เข้ากับกราวด์ของไฟเลี้ยง

16. Play /Record Input (P/R) ขา 27 เมื่อขาอินพุทควบคุมการเล่นกลับและบันทึกได้รับพัลส์ "1" จะเป็นวงรอบของการเล่นกลับ และถ้าเป็น "0" จะเป็นการเลือกวงรอบการบันทึก ถ้าหากได้รับพัลส์ที่ขอบขาลงของขา CE จะเป็นการแลทช์อินพุทที่ขา P/R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 ไอซีถอดรหัสความถี่โทรศัพท์

วงจรถอดรหัสเป็นวงจรที่จำเป็นมากๆ ในที่นี้เราจะมาทำความเข้าใจกับคำว่าถอดรหัสความถี่โทรศัพท์ หมายถึง การแปลงสัญญาณความถี่ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม (ชนิด Tone หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิตอลซึ่งไอซีดังกล่าว คือ MT8870 ใช้แปลงรหัสความถี่โทรศัพท์ให้เป็นเลขมาตรฐานสองขนาด 4 บิท

จากที่ผ่านๆมาการออกแบบวงจรถอดรหัสความถี่ของโทรศัพท์มักใช้ไอซีจำพวกเฟสล็อกลูป ซึ่งสร้างปัญหามากมาย ไม่ว่าเรื่องของความถี่ที่เปลี่ยนไป การปรับแต่งวงจร ขนาดของวงจรที่ใหญ่ เพราะต้องใช้ไอซีจำนวนมาก และปัจจุบันได้มีการใช้วงจรรวมหรือไอซีมาประกอบเป็นวงจรรวม ทำให้ลดปัญหาต่างๆ ที่เกิดขึ้นจากวงจรเดิมและทำให้การทำงานของวงจรมีประสิทธิภาพมากขึ้นด้วย

คุณสมบัติของ MT 8870

1.เป็นตัวรับและถอดรหัสความถี่ (DTME receiver)

2.กินไฟน้อยใช้ไฟเลี้ยงระดับกับ TTL

3.สามารถติดตั้งอัตราขยายภายในตัวไอซีได้

4.สามารถปรับการ์ดไทม์ (Guard time) ได้

5.เป็นไอซีคุณภาพสูง

การนำ MT 8870 ไปใช้งาน

1.นำไปใช้งานด้านรีโมทคอนโทรลทางไกล

2.เครื่องป้องกันโทรศัพท์

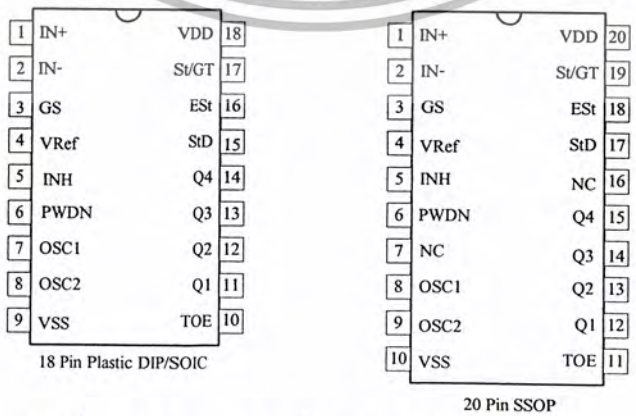
3.เครื่องกันขโมย

4.การควบคุมอุปกรณ์ผ่านทางโทรศัพท์

5.เครื่องสอบถามทางโทรศัพท์

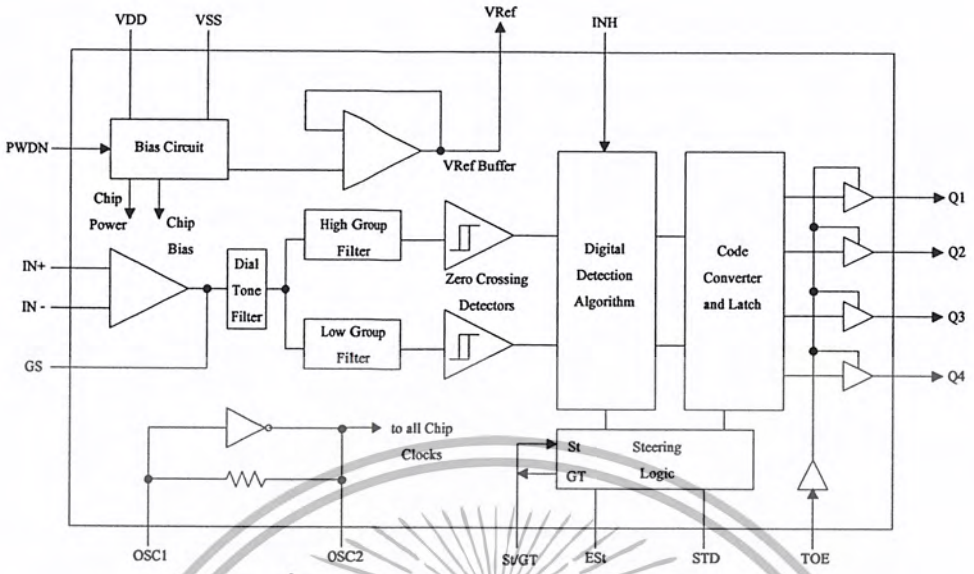
2.10.1 โครงสร้างของ MT 8870

โครงสร้างภายในของ MT 8870 ประกอบด้วยวงจรกรองความถี่ และวงจรถอดรหัสฟังก์ชั่นทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO2_CMOS



ภาพที่ 2.15 แสดงรายละเอียดขาของ MT 8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 แสดงโครงสร้างภายในของ MT 8870

รายละเอียดที่น่าสนใจอีกอย่างหนึ่งคือ ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิทช์คาปาซิเตอร์ฟิลเตอร์สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอล เพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิท และเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุทเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอกอีกเพียงเล็กน้อย ก็จะได้อัตราการขยายตามต้องการ ส่วนวงจรด้านเอาท์พุทเป็นวงจรแลทช์ 3 สถานะ ดังนั้นเพื่อความเข้าใจที่ดีขึ้นโปรดพิจารณาตามภาพที่ 2.15 แสดงขาของ MT 8870 และภาพที่ 2.16 แสดงโครงสร้างภายในของ MT 8870

ฟังก์ชั่นการทำงานภายใน MT 8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

1. ภาคกรองความถี่ (filter section)
2. ภาคถอดรหัส (decoder section)
3. ภาคตรวจสอบสัญญาณ (steering circuit)
4. ภาคขยายสัญญาณความแตกต่าง (differential input)
5. ภาคกำเนิดความถี่ (oscillator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F_{LOW}	F_{HIGH}	N2O	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	1
941	1209	.	H	1	0	1	0
941	1477	#	H	1	1	0	1
697	1633	A	H	1	1	0	0
770	1633	B	H	1	1	1	1
852	1633	C	H	1	1	1	0
941	1633	D	H	0	0	0	1
-	-	ANY	L	Z	Z	Z	Z

ตารางที่ 2.9 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ

1. ภาคกรองสัญญาณความถี่

การทำงานของส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือช่วงความถี่ต่ำ และช่วงความถี่สูง โดยใช้วงจรกรองความถี่อันดับ 6 ชนิดสวิทช์คาปาซิเตอร์ (six – order switched capacitor band filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วงคือช่วงความถี่สูงและช่วงความถี่ต่ำ

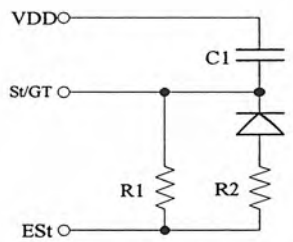
2. ภาคถอดรหัส

ในส่วนของความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิตอล และมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นที่เข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา (early steering) ก็จะแอคทีฟสำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆ นั้นแสดงในตารางที่ 2.9

3. ภาคตรวจสอบสัญญาณ

สัญญาณก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาท์พุท จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ซึ่งต้องกดปุ่มให้ มีความถี่ออกมาเป็นช่วงเวลาพอควร มิฉะนั้น วงจรส่วนนี้จะไม่รับโดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอกสัญญาณที่ขา Est จะเป็น “high” นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$t_{GTP} = (R_P C_1) \ln\left[\frac{V_{DD}}{(V_{DD} - V_{TSt})}\right]$$

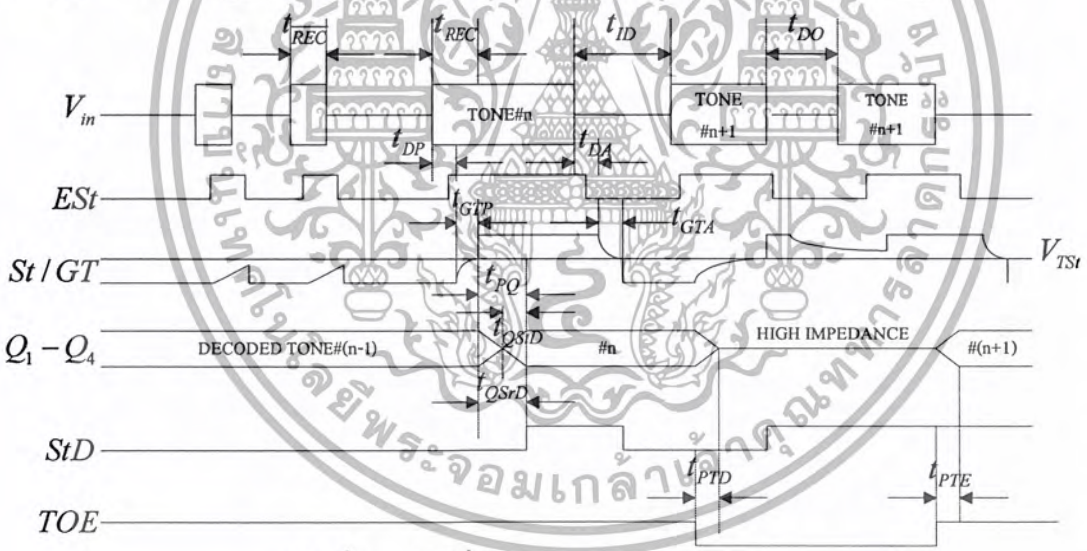
$$t_{GTA} = (R_1 C_1) \ln\left[\frac{V_{DD}}{V_{TSt}}\right]$$

$$R_P = \left(\frac{R_1 R_2}{R_1 + R_2}\right)$$

ภาพที่ 2.17 แสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาการ์ดไทม์ พร้อมวิธีการคำนวณ

จากภาพที่ 2.17 เมื่อขา Est เป็น "high" ทำให้ Vc สูงขึ้น ตัวเก็บประจุ C จะคายประจุทำให้แรงดัน Vc สูงขึ้นจนถึงค่าเทรสโฮลด์ วงจรถอดรหัส จึงถอดรหัสออกมาเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงานดูได้จากไทม์มิ่งไดอะแกรม ในภาพที่ 2.18

พิจารณาคำว่า การ์ดไทม์ นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่ตั้งไว้ จึงจะได้รับการยอมรับว่าสัญญาณนั้นถูกต้องหรือพูดได้ว่า เวลาที่ตั้งไว้โดย RC ก็คือการ์ดไทม์ นั่นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าเวลาที่ตั้งไว้จึงสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่าก็จะ ไม่มีการถอดรหัสเป็นตัวเลขออกไป

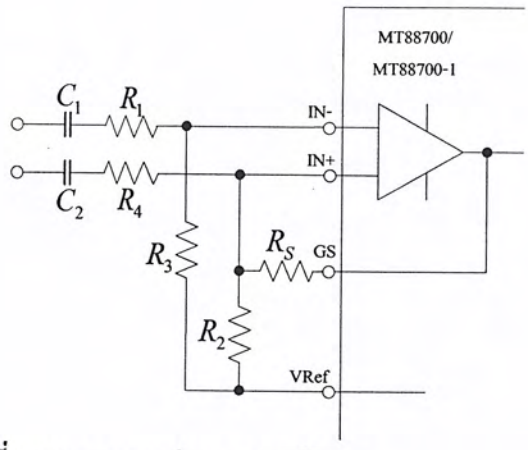


ภาพที่ 2.18 ไทม์มิ่งไดอะแกรมของ MT 8870

4.ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุทของ MT 8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายได้โดยต่อวงจรภายนอกเพิ่มเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.19 แสดงการต่อวงจรภาคอินพุต

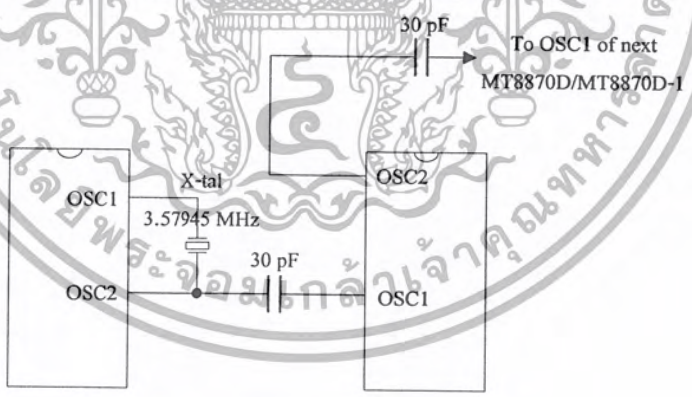
จากภาพที่ 2.19 แสดงการต่อวงจรภายนอกเข้ากับอินพุท ซึ่งสามารถคำนวณอัตราการขยายความแตกต่างและอินพุท อิมพิแดนซ์ได้ดังนี้

$$\text{อัตราการขยาย } (A_v \text{diff}) = \frac{R1}{R5}$$

$$\text{อินพุทอิมพิเดนซ์ } (Z_{in} \text{ diff}) = 2\sqrt{R_1^2 + \left(\frac{1}{\omega C}\right)^2}$$

5. ภาคกำเนิดความถี่

ในภาคนี้ภายในไอซีจะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อคริสตอลขนาด 3.5794 M_{HZ} ก็สามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงดังภาพที่ 2.20



ภาพที่ 2.20 แสดงวงจรผลิตความถี่

2.11 โทรศัพท์

ส่วนประกอบหลักของเครื่องโทรศัพท์แบ่งออกได้ 3 ส่วนดังนี้

1.ส่วนที่รับส่งสัญญาณเสียงพูด (Speech Transmission)

2.ส่วนที่กำเนิดสัญญาณ โค๊ดเลขหมายของผู้ถูกเรียก

3.ส่วนที่รับสัญญาณเรียกจากชุมสาย (Electronic Ringer)

2.11.1 ส่วนรับส่งสัญญาณเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้ประกอบด้วยเสียงพูด (Transmitter) หูฟัง(Receiver) และวงจรรับฟัง(Transmis circuit)

2.11.2 ส่วนที่ทำหน้าที่ส่งโค๊ดเลขหมายของผู้เรียก

1. แบบ Pulse dialing เป็นวิธีการส่งโค๊ดของเลขหมายโดยวิธีการตัดต่อเส้นทางเดินของไฟกระแสตรงในสายชั่วขณะสั้นๆ ส่งเลขหมาย จะใช้ 10-20 pps (pulse per sec) ข้อเสียของการส่งเลขหมายแบบนี้ก็คือใช้เวลาในการส่งเลขหมายนาน ค่าเฉลี่ย 1.5 วินาที / เลขหมาย และยังต้องมีการปฏิบัติบำรุงรักษาเป็นระยะ เช่น การทำความสะอาดหน้าสัมผัส การปรับเทียบอัตราความเร็วในการส่งเลขหมาย

2. แบบกดปุ่ม (Push button)สัญญาณโค๊ดเลขหมายของผู้เรียก ถูกส่งในรูปสัญญาณ 2 ความถี่หรือเรียกว่า DTMF (Dual Tone Multifrequency) ซึ่งให้ข้อดีในแง่ลดเวลาในการส่งเลขหมายลงเฉลี่ย แล้วจะเหลือเพียง 0.7 วินาที/ เลขหมาย ความผิดพลาดของโค๊ดมีโอกาสน้อยมาก เพราะมีการตรวจสอบความถี่ 2 ชุดพร้อมๆ กัน นอกจากนั้นยังสามารถให้จำนวนโค๊ดได้มาก เช่นเพิ่มโค๊ด * (star) และ # (square) ซึ่งใช้ในการบริการพิเศษต่างๆ เช่นบริการเลขหมายย่อย ความถี่ที่เกิดจะแบ่งออกเป็น 2 กลุ่ม คือกลุ่มความถี่สูงตั้งแต่ความถี่ 1205 Hz. เป็นต้นไปและกลุ่มความถี่ต่ำตั้งแต่ 942 Hz. ลงมา เมื่อถูกกดปุ่มหนึ่งปุ่มความถี่ 2 ความถี่จากกลุ่มความถี่สูงและกลุ่มความถี่ต่ำจะถูกส่งไปในสาย ค่าของความถี่ทั้งสองจะขึ้นอยู่กับปุ่มที่กดตามตารางที่ 2.10 เช่น เมื่อทำการกดปุ่มหมายเลข 5 ความถี่ 770 Hz. และ 1336 Hz ถูกส่งออกไปพร้อมๆ กัน

	Frequency Hz	
Button	Group	Group
1	697	1209
2	697	1336
3	697	1477
4	770	1209
5	770	1336
6	770	1477
7	825	1209
8	825	1336
9	825	1477
.	941	1209
0	941	1336
	941	1477

ตารางที่ 2.10 ความถี่ DTMF กับปุ่มที่กด

3. ส่วนที่รับสัญญาณเรียกแบบกระดิ่ง เมื่อผู้เช่าเป็นฝ่ายถูกเรียก (B subscriber) สัญญาณเรียก (Ringing signal) จะถูกส่งออกมาจากชุมสายเป็นแรงไฟฟ้า AC ความถี่ 25 Hz. ค่าแรงดันอยู่ในช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

70-90 V_{rms} โดยรวมมากับไฟ DC -48 V, ในสภาพที่ผู้เช่าวางหูปกติ วงจรกระดิ่งจะถูกต่อกับสายโดยผ่าน Condenser ค่าประมาณ 1 uF ซึ่งจะทำหน้าที่กันไฟ DC ไม่ให้ผ่านวงจรกระดิ่งแต่แรงไฟ AC จะผ่านไปยังขดลวดของกระดิ่งไฟฟ้าได้วงจรของกระดิ่งไฟฟ้า สัญญาณกระดิ่งเป็นสัญญาณที่ส่งเป็นช่วง คือจะส่งสัญญาณ 1 วินาที และหยุด 4 วินาที กระแสที่ไหลผ่านขดลวดจะทำให้สนามแม่เหล็กที่ขั้วมากหรือน้อยขึ้นอยู่กับทิศทางกระแสขั้ว S ของแกนและขั้ว N ของ Armature เกิดจากการเหนี่ยวนำของแท่งแม่เหล็กถาวรที่อยู่ตรงกลาง

เมื่อผู้เช่าทำการยกหูรับ Hook switch จะทำการต่อวงจรเสียงพูดมีค่าความต้านทานทาง DC ต่ำ กลยุทธ์ระแสที่ไหลในสายจะทำให้ชุมสายทราบว่าผู้เช่ายกหู วงจรส่งสัญญาณกระดิ่งจะถูกตัดออก และชุมสายจะทำการต่อเส้นทางเสียงพูดไปยังผู้เรียกให้พูดคุยกันได้

2.11.3 สัญญาณที่รับส่งระหว่างผู้เช่าและชุมสาย (Subscriber Signaling)

2.11.3.1 สัญญาณที่ส่งจากบ้านผู้เช่าไปยังชุมสาย

1. ON HOOK หมายถึงสภาพผู้เช่าวางหู หรือสภาพว่าง (Idel) ลักษณะขอวงจรจะเป็น Open loop high impedance

2. OFF HOOK สภาพผู้เช่ายกหู สายจะมีสภาพ Closes loop low impedance

3. Dialing ผู้เช่าทำการหมุนเลขหมายเครื่องแบบ Rotary dial สัญญาณจะเป็นแบบ Pulsing ค่า Impedance จะสูง, ต่ำ สลับกันตามโค๊ดที่หมุน ในเครื่องแบบกดปุ่มจะมี สัญญาณ DTMF ส่งออกไป

2.11.3.2 สัญญาณที่ส่งจากชุมสาย

1. Dialing Tone เป็นสัญญาณที่บอกให้ทราบว่า ขณะนี้อุปกรณ์ที่ชุมสายพร้อมที่จะรับโค๊ดการหมุนเลขหมายจากผู้เรียก ให้ผู้เรียกทำการส่งเลขหมายได้ สัญญาณ Dial Tone นี้เป็นสัญญาณต่อเนื่องความถี่ 425 Hz. Modulated ด้วย 50 HZ. ผู้เช่าจะได้ยินโทนนี้เมื่อทำการยกหูโทรศัพท์เพื่อทำการเรียก

2 Busy Tone เป็นสัญญาณที่ส่งมาบอกให้ทราบว่าอุปกรณ์ไม่ว่าง เช่นผู้เช่ายกหูแล้วได้ยินเสียงนี้ แทนที่จะได้ยิน Dial Tone แสดงว่าอุปกรณ์ในชุมสายไม่ว่าง แต่ถ้าได้ยินเสียงนี้หลังจากหมุนหมายเลขไปแล้วแสดงว่าผู้เช่าฝ่ายถูกเรียก (B Subscriber) ไม่ว่างหรืออุปกรณ์สำหรับต่อออกไปยังชุมสายอื่นไม่ว่างในกรณีที่ผู้ถูกเรียกอยู่ต่างชุมสายสัญญาณที่ส่งเป็นสัญญาณที่ขาดตอนเป็นช่วงๆ ส่ง 0.5 วินาที หยุด 0.5 วินาที ความถี่ของสัญญาณ 425 Hz. Sine wave

3. Ringing Tone เป็นสัญญาณที่ผู้เรียกได้ยินหลังจากหมุนเลขหมายครบแล้ว ทราบว่า การต่อกระทำได้สำเร็จ ขณะนี้ชุมสายได้ส่งสัญญาณเรียก (Ringing Signal) ไปยังผู้ถูกเรียกสัญญาณใช้ความถี่ 425 Hz. Sine wave โดยจะส่ง 1 วินาที หยุด 4 วินาที

4. Ringing Signal หรือสัญญาณเรียก เป็นสัญญาณที่ส่งไปยังผู้เช่าฝ่ายถูกเรียกซึ่งจะได้ยินเป็นเสียงกระดิ่ง หรือโทน ขึ้นอยู่กับวงจรที่ใช้สัญญาณเป็น Sine wave 25 Hz. ค่าแรงดัน 70-90 V_{rms} ช่วงการส่งเช่นเดียวกับ Ringing Tone คือส่ง 1 วินาที หยุด 4 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สัญญาณโทนอื่นๆ และเสียงตอบจากเครื่องบันทึกเสียง (Anounvenent machine) สัญญาณโทนอื่นๆ เช่น No Tone (Number unobtainable tone) บอกให้ทราบว่าเลขหมายนี้ได้เปลี่ยนไปแล้ว เป็นต้น

2.12 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ (วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรสื่อสารที่สามารถและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณพอร์ต 3 คือ ขา P3.0 เป็นขารับสัญญาณข้อมูลเข้า หรือ RxD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TxD โดยใช้มาตรฐานของ RS-232 แต่ในปัจจุบันสามารถติดต่อได้ในมาตรฐาน RS-442 หรือ RS-485 ได้แล้ว โดยใช้ไอซีพิเศษที่ทำหน้าที่แปลงสัญญาณการสื่อสารดังกล่าว

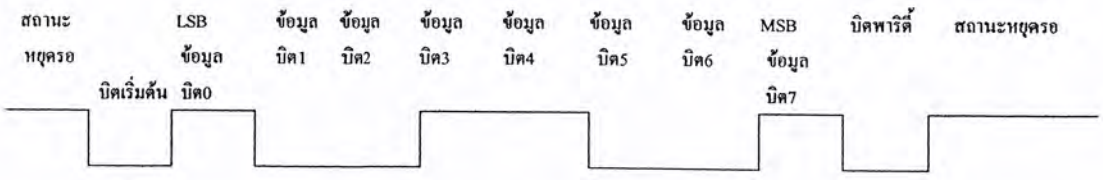
2.13การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูล โดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยแต่จะใช้การกำหนดค่าอัตราเร็วในการรับส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตรานี้ว่า อัตราบอด หรือ บอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที (Bit Per Second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วน ดังนี้ คือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (Stop Bit) มีขนาด 1 บิต

ภาพที่ 2.21 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีการเริ่มต้นข้อมูล ขา DATA จะมีสถานะลอจิก " 1 " เรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก " 0 " ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น (Start Bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ซึ่งข้อมูลที่ต้องการส่งมีจำนวน 8 Bit จากนั้นตามด้วย บิตพาริตี้ (Parity Bit) ซึ่งใช้ในการตรวจสอบข้อผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ จะส่ง คือ บิตปิดท้าย หรือ บิตหยุด (Stop Bit) โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก " 1 " อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต , 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



ภาพที่ 2.21 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส

อัตราเร็วในการับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่า ตั้งแต่ 110 ถึง 19,200 บิตต่อวินาที โดยมีค่ามากขึ้นตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีของคอมพิวเตอร์ เนื่องจากอัตราบอดคือ ค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมุติว่าข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี้บิต มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์ จะมีความยาวกับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 9600 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดได้ให้เป็นแบบคี่ (odd) , แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ พาริตี้คี่หรือพาริตี้คู่แสดงถึงลอจิก 1 ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมพาริตี้ว่ามีจำนวนเป็นเลขคู่หรือคี่ ยกตัวอย่างเช่น ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99 H หรือ 10011001B จะเห็นว่าข้อมู,ในไบต์นี้มีจำนวนลอจิก “ 1 จำนวน 4 ตัว ซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าของพาริตีต้องมีลอจิกเป็น “ 0 แต่ถ้ากำหนดพาริตี้เป็นคี่ ค่าของพาริตี้จะต้องเป็น 1 เพื่อให้ข้อมูล 1 ไบต์รวมทั้งพาริตี้เป็นคี่ พาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter : เป็นอุปกรณ์ที่ใช้ในการรับส่งข้อมูลอนุกรม) ซึ่งทางภาครับต้องกำหนดการตรวจสอบพาริตี้ที่ตรงกันเอาไว้ว่าจะตรวจสอบพาริตี้คู่หรือคี่ โดยการนับจำนวนลอจิก 1 ทั้งหมดรวมทั้งพาริตี้บิตด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่ แต่ค่าตัวเลขในการนับออกมาเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น แต่ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะใช้ไม่ได้ผล สำหรับการตั้งพาริตี้เป็น NONE นั้นทางภาครับและส่งจะไม่มีการตรวจสอบพาริตี้

2.13.1รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังนี้ คือ

1) รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังชั่นพิเศษหรือ SFR มีขนาด 8 บิตแบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (Transmit buffer register) และรับข้อมูล (Receiver Buffer Register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา 3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกจะผ่านทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์

2) รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิตมีรายละเอียดการทำงาน ดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SMO	SM1	SM2	REN	TB8	RB8	TI	RI

ภาพที่ 2.22 แสดงรายละเอียดการทำงานแต่ละบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0-SM1 (Serial Port Mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 มีรายละเอียดดังต่อไปนี้

SM2 : ใช้ในการเอ็นนาเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ (Multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกทีฟ ถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกทีฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception) :ใช้ในการเอ็นนาเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้ปืน "1"

TB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8 : ใช้สำหรับรับข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8คือข้อมูลของบิตหยุด (Stop Bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt Flag) : ใช้ในการแสดงการเกิดการอินเตอร์รัปเมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receiver Interrupt) : ใช้แสดงการเกิดการอินเตอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรมสามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อสามารถรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

2.13.2โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

พอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 สามารถเลือกการทำงานได้ถึง 4 โหมด คือ

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้

การเลือกโหมดทำได้โดยการกำหนดข้อมูลให้แก่บิต SM0 และ SM1 ในรีจิสเตอร์ SCON

1.การทำงานในโหมด 0 ของวงจรพอร์ตอนุกรม

มีไดอะแกรมการทำงานและไดอะแกรมเวลาแสดงได้ดังภาพที่ 2.22 ข้อมูลอนุกรมจะผ่านเข้าและออกทางขา RxD ส่วนขา TxD ทำหน้าที่เป็นสัญญาณนาฬิกาของการเลื่อนข้อมูล (Shift clock) ในโหมดนี้มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนข้อมูลบิต โดยการรับและส่งข้อมูลในบิต LSB ก่อน อัตราในการรับส่งข้อมูลหรืออัตราบอดถูกกำหนดไว้คงที่ ที่ 1/12 ของความถี่สัญญาณนาฬิกา

เริ่มต้นการส่งข้อมูลด้วยการเขียนข้อมูลที่ต้องการส่งมายังรีจิสเตอร์ SBUF สัญญาณเขียนข้อมูล SBUF แอกทีฟเป็น " 1" ที่สเตท 6 เฟส 2 (S6P2) ของแมชชีนไซเกิล ส่งมายังวงจรควบคุมการส่ง (TX control) ทำให้วงจรควบคุมเริ่มต้นส่งข้อมูล สัญญาณ SEND จะแอกตีฟเป็น " 1" ตลอดการส่งข้อมูล

ข้อมูลจากรีจิสเตอร์ SBUF จะถูกเลื่อนออกที่ขา P3.0 หรือขา RxD ครั้งละบิต ตามจังหวะของสัญญาณนาฬิกาที่ส่งออกมาทางขา P3.1 หรือ TxD โดยสัญญาณนาฬิกาของการเลื่อนข้อมูลจะมีขอบขาลงของสัญญาณที่สเตท 3 เฟส และมีขอบขาขึ้นของสัญญาณที่สเตท 6 เฟส 1 ของแต่ละแมชชีนไซเกิลในกระบวนการส่งข้อมูล จนกระทั่งส่งข้อมูลครบ 8 บิตแล้ว บิต TI ในรีจิสเตอร์ SCON จะเกิดการเซต เป็นการแจ้งให้ทราบว่าส่งข้อมูลครบแล้ว หากการอินเตอร์รัพจากพอร์ตอนุกรมได้รับการเอ็นนาเบิลไว้ ก็จะเกิดการอินเตอร์รัพขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล สัญญาณ SEND จะกลายเป็น "0" จนกว่าจะเริ่มต้นกระบวนการรับข้อมูลใหม่

ในกระบวนการรับข้อมูล เริ่มต้นด้วยการเซต REN ให้เป็น "1" และเคลียร์บิต RI ในรีจิสเตอร์ SCON ก่อนที่สเตท 6 เฟส 2 ของแมชชีนไซเกิลถัดไป วงจรควบคุมการรับ (RX control) จะทำการเขียนข้อมูล 11111110 ไปยังชิฟต์รีจิสเตอร์สำหรับรับข้อมูลและทำการแอกทีฟสัญญาณ RECEIVER ให้เป็น "1" ในสัญญาณนาฬิกาลูกถัดไป

เมื่อสัญญาณ RECEIVER แอกทีฟ ก็จะเกิดการส่งสัญญาณนาฬิกาของการเลื่อนข้อมูล (Shift Clock) ขึ้นผ่านทางขา P 3.1 หรือ TxD เพื่อกำหนดการรับข้อมูลครั้งละบิต โดยสัญญาณนาฬิกานี้จะเกิดขึ้นในช่วงสเตท 3 เฟส 1 ถึงสเตท 6 เฟส 1 ของแต่ละแมชชีนไซเกิล การรับข้อมูลเข้ามาทาง P 3.0 หรือ RxD จะเกิดขึ้นที่สเตท 5 เฟส 2 ในแมชชีนไซเกิลเดียวกับสัญญาณนาฬิกาของการเลื่อนข้อมูลจนกระทั่งรับข้อมูลมาครบทั้ง 8 บิต บิต RI จะได้รับการเซตเพื่อแจ้งให้ทราบการเสร็จสิ้นกระบวนการรับข้อมูล หากการอินเตอร์รับจากพอร์ตอนุกรมเกิดการเอ็นนาเบิลไว้ก็จะเกิดการอินเตอร์รับขึ้นในระบบ เมื่อเสร็จสิ้นกระบวนการรับข้อมูล กระบวนการ RECEIVER จะกลายเป็น "0" จนกว่าจะเริ่มต้นรับข้อมูลใหม่

การทำงานในโหมดนี้ของไมโครคอนโทรลเลอร์ MCS-51 จะใช้กับการเชื่อมต่อกับรีจิสเตอร์ภายนอกเพื่อทำการขยายพอร์ตอินพุทและเอาต์พุต แต่ไม่เป็นที่นิยมใช้งานมากนักเนื่องจากในไมโครคอนโทรลเลอร์ MCS-51เองมีพอร์ตอยู่ค่อนข้างมาก และติดต่อกับพอร์ตเหล่านั้นได้ง่ายและเร็วกว่ามาก

2.การทำงานในโหมด 1 ของวงจรพอร์ตอนุกรม

ในโหมดนี้ใช้การรับส่งข้อมูลรวม 10 บิต โดยส่งข้อมูลออกทางขา P 3.1 หรือ TxD และรับข้อมูลเข้าทางขา P 3.0 หรือ RxD ข้อมูลทั้ง 10 บิตประกอบด้วย บิตเริ่มต้น (มีค่าเป็น "0") บิตข้อมูล 8 บิต โดยรับและส่งข้อมูลนบิต LSB ก่อน และบิตหยุดหรือบิตปิดท้าย (มีค่าเป็น "1")ในการรับข้อมูลบิตหยุดจะถูกเก็บไว้ในบิต RB 8 ในรีจิสเตอร์ SCON อัตราบอดในโหมดนี้ได้รับการกำหนดโดยอัตราการเกิดโอเวอร์โฟลว์ของไทเมอร์ 1 ของ AT89C51

กระบวนการส่งข้อมูลเริ่มต้นด้วยการแอกทีฟสัญญาณเขียนข้อมูลมายังรีจิสเตอร์ SBUF ส่งมายังวงจรควบคุมการส่ง (TX control) จากนั้นวงจรควบคุมจะทำการแอกทีฟสัญญาณ SEND ที่สเตท 1 เฟส 1 ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แมชชีนไซเกิลต่อมา โดยสัญญาณ SEND จะเป็น "0" ตลอดการส่งข้อมูล เมื่อสัญญาณ SEND แอกทีฟ จะทำการส่งบิตเริ่มต้นก่อนเป็นบิตแรก โดยมีคาบเวลาของบิตเริ่มต้นเท่ากับ 1 แมชชีนไซเกิล จากนั้นตามด้วยการส่งบิตข้อมูล 8 บิตเรียงลำดับจากบิต LSB โดยข้อมูลที่ทำการส่งถูกเรียกออกมาจากรีจิสเตอร์ Buffer สำหรับการส่งข้อมูล ในทุกๆบิตข้อมูลที่ทำการส่งออกไป จะเกิดสัญญาณพัลซ์ SHIFT ขึ้น เพื่อให้เรียกข้อมูลในแต่ละบิตจากรีจิสเตอร์บัฟเฟอร์ การกำหนดจังหวะการส่งข้อมูลใช้สัญญาณนาฬิกาการส่ง (TX clock) เป็นตัวกำหนด โดยสัญญาณนาฬิกานี้ได้มาจากการหารสัญญาณ TCLK จากไทม์เมอร์ 1 ด้วย 16 หลังจากการส่งบิตข้อมูลก็จะทำการส่งบิตหยุดหรือบิตปิดท้าย 1 บิต ดังนั้นการส่งข้อมูลจะใช้สัญญาณนาฬิกาทั้งหมด 10 ลูกเมื่อทำการส่งข้อมูลครบทั้งหมดแล้ว จะทำการเซตบิต TI ในรีจิสเตอร์ SCON หากการอินเตอร์รับจากพอร์ตอนุกรมได้รับการเอ็นาเบิลไว้ ก็จะเกิดการอินเตอร์รับขึ้นในระบบ หลังจากที่ทำการบริการอินเตอรัปต์หรือส่งข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต TI ก่อนเป็นอันดับแรก เพื่อให้การส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

ด้านการรับข้อมูล จะทำการตรวจจับการเปลี่ยนแปลงระดับลอจิกจาก "1" เป็น "0" ที่ขา RxD โดยใช้อัตราการสุ่มเท่ากับ 1/16 เท่าของอัตราบอด เมื่อตรวจจับพบไทม์เมอร์/เคาน์เตอร์ที่ใช้ในการกำหนดอัตราบอดจะรีเซตและทำการเขียนข้อมูล 1FFH ไปยังชิฟรีจิสเตอร์ ข้อมูลจะเริ่มเดินทางเข้าสู่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ผ่านทางขา RxD ในการตีความว่าบิตที่เข้ามาเป็น "0" หรือ "1" จะใช้การตีความข้างมาก โดยบิตของข้อมูลที่เข้ามาจะแบ่งออกเป็น 16 สเตท การสุ่มข้อมูลจะทำการสุ่มที่สเตทที่ 7 , 8 และ 9 หาก 2 ใน 3 ของการสุ่มพบว่าข้อมูลเป็นลอจิกใด จะตีความข้อมูลในบิตนั้นเป็นตามเสียงข้างมาก ยกตัวอย่าง สุ่มพบลอจิก "1" 2 ใน 3 ครั้ง จะตีความว่าบิตของข้อมูลที่รับได้นั้นเป็น "1"

ลำดับการรับข้อมูลมีลักษณะเดียวกับการส่งข้อมูลคือ เริ่มที่บิตเริ่มต้นก่อน ตามด้วยบิตข้อมูล และบิตปิดท้าย ในทุกๆการรับข้อมูลได้ 1 บิต จะมี Pulse Shift เกิดขึ้น เพื่อทำการเลื่อนข้อมูลเข้าสู่รีจิสเตอร์บัฟเฟอร์การรับข้อมูล การกำหนดจังหวะการรับข้อมูลใช้สัญญาณนาฬิกาการรับข้อมูล (RX clock) หลังจากสัญญาณนาฬิกาลูกสุดท้าย อันหมายถึงการรับสัญญาณข้อมูลได้ครบแล้ว วงจรควบคุมการรับข้อมูลจะทำการส่งข้อมูลจากรีจิสเตอร์บัฟเฟอร์ไปยังรีจิสเตอร์ SBUF และบิต RB8 ในรีจิสเตอร์ SCONโดยข้อมูลในบิตที่ RB8 ก็คือ ข้อมูลของบิตหยุดนั่นเอง พร้อมกันนั้นยังทำการเซตบิต RI ในรีจิสเตอร์ SCON ด้วย หากการอินเตอร์รับจากพอร์ตอนุกรมได้รับการเอ็นาเบิลไว้ ก็จะเกิดการอินเตอร์รับขึ้นในระบบ หลังจากบริการอินเตอร์รับหรือรับข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต RI ก่อน เพื่อให้การรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

การทำงานในโหมดนี้ได้รับความนิยมสูงสุด เนื่องจากมีกระบวนการที่ไม่ซับซ้อนและสามารถทำการรับส่งข้อมูลกับคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ

3.การทำงานในโหมด 2 และ 3 ของวงจรพอร์ตอนุกรม

ในทั้ง 2 โหมดนี้จะใช้รูปแบบข้อมูลรวม 11 บิต ประกอบด้วยบิตเริ่มต้น มีค่าเป็น "0" จำนวน 1 บิต , บิตข้อมูล 8 บิต โดยทำการรับและส่งบิต LSB ก่อน, บิตข้อมูลบิตที่ 9 และบิตปิดท้ายมีค่าเป็น "1" จำนวน 1 บิต ในการส่งข้อมูล ข้อมูลบิตที่ 9 จะนำไปเก็บไว้ที่บิต TB8 ในรีจิสเตอร์ SCON และในการรับข้อมูล ข้อมูลบิตที่ 9 จะนำไปเก็บไว้ที่บิต RB8 ในรีจิสเตอร์ SCON สำหรับอัตราบอดในโหมด 2 จะคงที่โดยเลือกได้ 2 ค่า คือ 1/32 หรือ 1/64 ของความถี่สัญญาณนาฬิกา สำหรับในโหมด 3 อัตราบอดสามารถปรับได้เหมือนกับโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

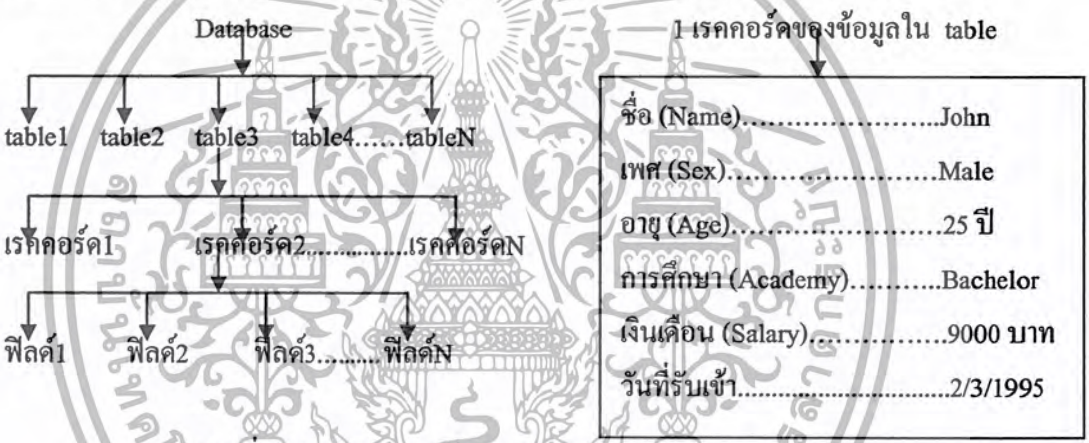
2.14 โปรแกรม Visual Basic

โปรแกรม Visual Basic นี้จะมีความสามารถในการนำมาประยุกต์ใช้ในการพัฒนางานด้านฐานข้อมูล ให้มีประสิทธิภาพมากยิ่งขึ้น และถึงแม้จะไม่เคยศึกษามาก่อนก็สามารถนำมาใช้งานได้ไม่ยาก โดยหลักการสำคัญที่ใช้มี ดังนี้

2.14.1 การใช้โปรแกรม Visual Basic สร้างไฟล์ฐานข้อมูล (Database)

Database คือ กลุ่ม (Collection) ของตาราง (Table) หรือกล่าวได้อีกนัยหนึ่งคือ Database เกิดจากกลุ่มของ table หลายๆ table มารวมกันอยู่ที่เดียวกันนั่นเอง ดังนั้นการสร้าง table จึงถือว่าเป็นพื้นฐานสำคัญของการสร้างหน่วยย่อยภายในไฟล์ Database และเนื่องจาก Database มีหลายๆ table อยู่ภายใน โดยแต่ละ table ก็เป็นกลุ่มของข้อมูลที่แตกต่างกัน ดังนั้นจึงจำเป็นต้องมีวิธีการจัดการข้อมูลต่างๆภายใน table เหล่านี้ให้สอดคล้องกับจุดประสงค์ของผู้ใช้

จะสร้างไฟล์ table ขึ้นมา 1 table เพื่อเก็บข้อมูลพนักงาน 120 คน ในบริษัทแห่งหนึ่ง ถ้าพิจารณาว่าข้อมูลพนักงานแต่ละคนอยู่บนกระดาษ 1 แผ่น ดังนั้นใน table นี้จะมีกระดาษอยู่ทั้งสิ้น 120 แผ่น



ภาพที่ 2.23แสดงรายละเอียดของการออกแบบ Database

นอกจากจะมี table สำหรับเก็บข้อมูลพนักงานแล้ว ยังมี table อื่นๆอีกมากมาย เราเรียกกระดาษหนึ่งแผ่นดังกล่าวนี้ว่า 1 เรคคอร์ด และภายใน 1 เรคคอร์ดนี้จะประกอบด้วยฟิลด์ต่างๆ เรคคอร์ดในตัวอย่างข้างต้นนี้ประกอบด้วยฟิลด์ชื่อ, ฟิลด์เพศ, ฟิลด์อายุ, ฟิลด์การศึกษา, ฟิลด์เงินเดือน และฟิลด์วันที่เริ่มเข้าทำงาน

ในขั้นต้นต้องออกแบบฟิลด์ใน 1 เรคคอร์ดของ table เสียก่อน เพื่อใช้เป็นต้นแบบให้เรคคอร์ดอื่นๆ นำไปใช้สำหรับใส่ข้อมูลพนักงานแต่ละคนลงไป การสร้าง table โดยอาศัยโปรแกรม Visual Data Manager มีรายละเอียดขั้นตอนการทำ ดังนี้

1.เปิดโปรแกรมวิช่วลเบสิก จากนั้นวิช่วลเบสิกจะแสดงหน้าต่าง New Projrct เพื่อให้เลือกชนิดการสร้างไฟล์ ให้คลิก Cancel เพราะไม่ต้องการสร้างๆไฟล์

2.ที่เมนูบาร์ คลิก Add-In

3.คลิกที่ Visual Data Manager...จากนั้นโปรแกรม Visual Data Manager จะถูกรันขึ้นมาทันที โดยมีข้อความ VisData อยู่ที่ส่วนหัวของโปรแกรม

4.คลิกที่ File, New, Microsoft Access, Version 7.0 MDB...(จะแสดงหน้าต่าง Select Microsoft Access Database to Create เพื่อให้ใส่ชื่อไฟล์ Database ที่ต้องการสร้าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. บันทึกไฟล์ Database ที่สร้างขึ้นมาใหม่ในชื่อ .....(ชื่อที่ต้องการสร้าง)โดยอยู่ในไดเรกทอรี่ที่ต้องการ

6. ขั้นตอนต่อไปเป็นการสร้างโครงสร้างและตั้งชื่อ table รวมทั้ง Add Field ให้ครบ โดยหลังจากกดปุ่ม Save ที่ขั้นตอนที่ 5 แล้วจะปรากฏหน้าต่าง 2 หน้าต่าง คือ หน้าต่าง Database Window และหน้าต่าง SQL Statement ให้สังเกตว่าตอนนี้อยู่ในไฟล์ Database ที่เพิ่งสร้างขึ้น

7. คลิกขวาลงบนพื้นที่ว่างใน Database Window (จะเกิด PopupMenu)

8. คลิก New table จะเปิดหน้าต่าง Table Structure ซึ่งทำให้สามารถตั้งชื่อ table ที่ต้องการสร้างและ Add Field เพื่อสร้างโครงสร้างของเรคคอร์ดต้นแบบให้แก่ table

2.14.2 การใช้เมดธอดเพื่อค้นหาค่าที่ต้องการในฟิลด์ที่กำหนดจากทุกเรคคอร์ด

- เมดธอด FindFirst จะใช้ค้นหาค่าที่ต้องการในฟิลด์ที่กำหนด ซึ่งค่านั้นเป็นตัวแรกที่อยู่ในกลุ่มเรคคอร์ดของออบเจ็กต์ Recordset
- เมธอด FindLast จะใช้ค้นหาค่าที่ต้องการในฟิลด์ที่กำหนด ซึ่งค่านั้นเป็นตัวสุดท้ายที่อยู่ในกลุ่มเรคคอร์ดของออบเจ็กต์Recordset
- เมธอด FindNext จะใช้ค้นหาค่าที่ต้องการในฟิลด์ที่กำหนด โดยเริ่มค้นหาที่เรคคอร์ดถัดไปซึ่งอยู่ถัดจากเรคคอร์ดที่ตัวชี้ ชี้อยู่
- เมธอด FindPrevious จะใช้ค้นหาค่าที่ต้องการในฟิลด์ที่กำหนด แต่เป็นตัวที่อยู่ก่อนหน้าตำแหน่งที่ชี้ ชี้อยู่ในขณะนั้น ซึ่งจะทำให้เริ่มค้นหาที่เรคคอร์ดถอยหลังจากเรคคอร์ดตัวที่ชี้ในปัจจุบัน

2 14.3 การเขียนโปรแกรม Visual Basic ติดต่อสื่อสารทาง Serial Port

สามารถทำได้โดยใช้ VB Control ที่ชื่อ MSComm โดยที่ต้องกำหนด Custom Control เข้าไปที่เมนู Project--->components แล้วเลือกที่ช่อง MSComm ก็จะปรากฏเป็นรูปไอคอนโทรศัพท์สีเหลืองให้คลิกที่ไอคอนลากมาไว้บนฟอรมใน Project ของโปรแกรม

การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ

1. การติดต่อแบบอินเตอรัพต์ ขบวนการอินเตอร์รัพ ชนิดนี้อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูอยู่เสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่งจะส่งเป็นรหัสแอสกี้เราจะเขียนโปรแกรมอินเตอร์รัพ โดยเมื่อที่ข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2. การติดต่อแบบโพลลิ่ง ในระบบพีซีการโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งผ่านจากคีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลาการทำงานกับข้อมูลที่รับมาจะตรวจสอบที่ความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งมาทางคีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า Wet Poll ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิคการโพลแบบ Round Robin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง คือ OnComm Event ซึ่งใช้ในการติดต่อแบบอินเตอร์รัพ การเขียนโปรแกรมติดต่อ Serial Port แบบธรรมดาจะใช้ comEvent เพียง ComEVReceive,comEVSend

องค์ประกอบในการใช้ MSComm

การตั้งค่าติดต่อกับพอร์ต

- ComPort คือ เราต้องกำหนดหมายเลขพอร์ตที่ใช้ต่อ RS-232(Com1,Com2)รายละเอียดดูในSerial Port Detail
- Setting คือ เราต้องกำหนดอัตรา Baud,Parity,Data(จำนวนบิต),Stop
- HandShaking คือ เราจะกำหนดได้ 4 แบบ 1. ComNone 2.comXonXoff 3.comRST 4.comTRSXonXoff

การใช้ Buffer ในการรับส่งข้อมูล

- InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา
- OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป
- Rthreshold คือ การที่เรากำหนดการเกิด Event-Driven ในการรับข้อมูลเข้ามา
- Stheshold คือ การที่เรากำหนดการเกิด Event-Driven ในการรับข้อมูลออกไป
- Inputlen คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล
- EOFEnable คือ การที่บอกว่าสิ้นสุดของไฟล์(EOF)End of file

ด้านฮาร์ดแวร์

- ParityEnable คือ ค่าของคาแรคเตอร์ที่จะแทนในเมื่อเกิด Parity Error
- NullDiscard คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER
- RSTEnable คือ ทำให้มีสัญญาณ RST (Request to Send)
- DTREnable คือ ทำให้มีสัญญาณ DTR(Data terminal Ready)

การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

1.Property ชื่อ ComPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน

ตัวอย่าง MSComm1.CommPort=1

ในที่นี้จะใช้ Com1 ที่อยู่ด้านหลังเครื่องคอมพิวเตอร์

2.Property ชื่อ Setting คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ว่าอัตราบอดของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ค่าต่างๆ ดังนี้

MSComm1.Setting=Baud (อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช้ใส่N),จำนวนบิตข้อมูล,บิตหยุด

ตัวอย่าง MSComm1.Setting=9600,N,8,1

3. Property ชื่อ InputLent คือ การกำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ใน Buffer

ตัวอย่าง MSComm1.Inpuut=1

4. Property ชื่อ PortOpen คือ จะเปิดพอร์ตใช้งานหรือไม่ ถ้าเปิด=Ture ถ้าปิด=False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง MSComm1.PortOpen=Ture

5. Property ชื่อ Rthreshold คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในBuffer รับข้อมูล(Comport)มันทำให้เกิด CommEvent ใน OnComm Event

ตัวอย่าง MSComm1.Rthreshold=1

จากรายละเอียดที่กล่าวมาเราจะเขียนในโพรซีเดอร์ VB ซึ่งจะไว้ใน Sub Form_Load หรือจะสร้าง Sub ขึ้นมาใหม่ในกรณีที่เรียกใช้ภายหลัง

```
Private Sub From_Load ()
    MSComm1.Setting=9600,N,8,1
    MSComm1.ComPort=1
    MSComm1.InputLen=1
    MSComm1.PortOpen=True
    MSComm1=Rthreshold=1
END Sub
```

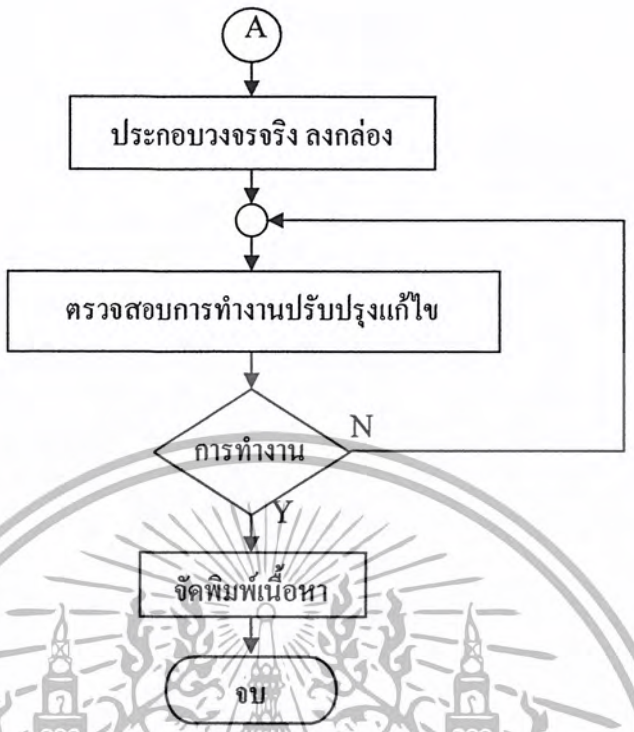
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1บทนำ

ในการจัดทำโครงงานนี้จำเป็นต้องมีการวางแผน จึงจะทำให้การดำเนินงานเป็นไปอย่างราบรื่น อย่างมีรูปแบบ จุดประสงค์เพื่อให้งานสำเร็จลุล่วงไปด้วยดี รูปแบบการเนินงานได้เขียนโฟว์ชาร์ต ให้เห็นลำดับขั้นในการดำเนินงาน มีขั้นตอนต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนภาพที่ 3.1 ขั้นตอนการทำงาน

3.2 ศึกษาข้อมูล

ศึกษาข้อมูลรายละเอียดต่างๆ ที่จะใช้ทำโครงงานนี้ ซึ่งมีทั้งด้านฮาร์ดแวร์ (Hardware) และซอร์ฟแวร์ (Software) ส่วนหลักของโครงงานนี้คือ ไมโครคอนโทรลเลอร์ ใช้เบอร์ 89C51 การสร้างฐานข้อมูลโดยใช้โปรแกรม Visual Basic 6.0 และหลักการเชื่อมต่อกันระหว่าง MCS-51 กับ Computer

3.3 วงจรเปลี่ยนสัญญาณกระดิ่ง (Ringing) เป็นสัญญาณดิจิตอล (Digital)

สัญญาณกระดิ่งมีแรงดันประมาณ 75-90 โวลท์อาร์เอ็มเอส มีความถี่ 50 Hz. มีสัญญาณ 1 วินาที หยุด 4 วินาที จะเปลี่ยนเป็นสัญญาณดิจิตอลระดับแรงดัน 0 V และ 5 V โดยต่อ C1 ค่า 0.1 uF 25 V เพื่อกันไฟกระแสตรง จากนั้นสัญญาณผ่านวงจรเร็คติไฟร์ D1-D4 ให้เลือกเฉพาะกระแสด้านบวก และจำกัดกระแสด้วย R1 ค่า เพื่อส่งให้ Optoisolator ทำงานจะได้สัญญาณที่ขา 5 ระดับแรงดันประมาณที่ 0 V และ 5 V แต่จะได้รับแรงที่ไม่คงที่ดังนั้น จึงสร้างวงจรโมโนสเตเบิลและกำหนดคาบของสัญญาณใหม่ จากคุณสมบัติวงจรโมโนสเตเบิลจะทำงานเมื่อมีสัญญาณขา 2 รับสัญญาณขอบขาลง ระดับสัญญาณขา 3 สัญญาณเอาท์พุท 1 คาบ

$T = 1.1\ R*C$

กำหนด $T = 0.5$ S และ $C = 1$ uF

$R = T / 1.1\ C$

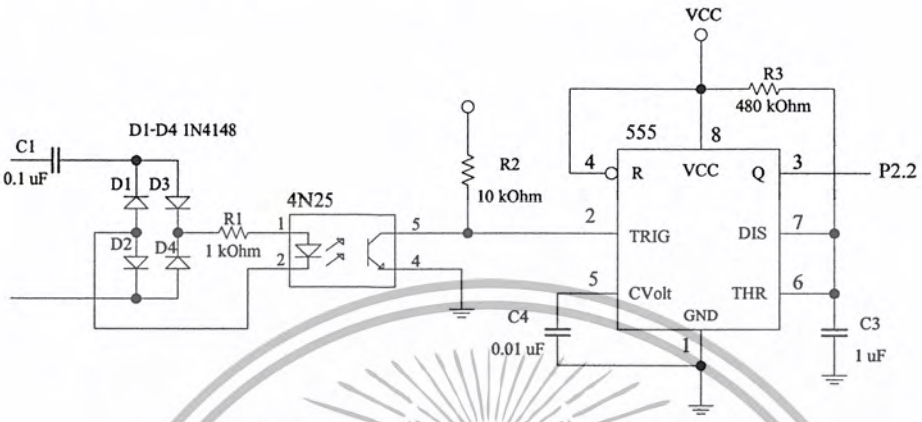
$= 0.5 / 1.1 * 1uF$

$= 454$ กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ R = 480 กิโลโอห์ม

เมื่อสัญญาณตามต้องการผ่าน NOT GATE ก่อนส่งให้ไมโครคอนโทรลเลอร์ที่ P2.4 ซึ่งหน่วงไว้ 2 s เพื่อจะทำการรับสัญญาณกระดิ่งและไมโครคอนโทรลเลอร์จะสั่งให้วงจร RELAY ทำงาน เพื่อวงจรส่วนอื่นจะได้ทำงานต่อ

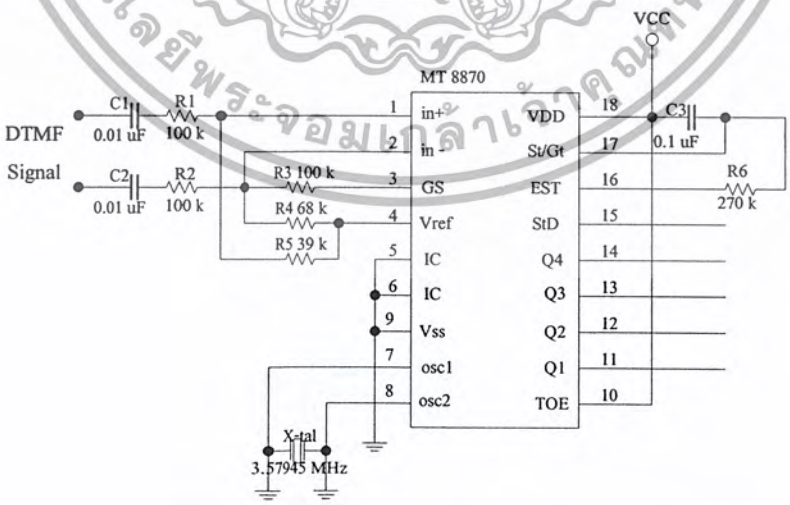


ภาพที่ 3.1 วงจรเปลี่ยนสัญญาณ Ringing เป็นสัญญาณ Digital

ดังนั้นในช่วง 500 mS ที่มีสัญญาณกระดิ่งเข้ามาสัญญาณรบกวนไม่สามารถที่จะเข้ามารบกวนได้ ทำให้สัญญาณ O/P ของไอซี 555 เกิดการเปลี่ยนแปลงได้ เพราะว่า ถ้าเกิด NOISE เข้ามารบกวน ก็จะทำให้สัญญาณที่พอร์ต 2.3 นั้นไม่เกิดการเปลี่ยนแปลง ทำให้การนับสัญญาณกระดิ่งพลาดได้

3.3.1 วงจรเปลี่ยนสัญญาณ DTMF เป็นสัญญาณ Digital

วงจรนี้สามารถเปลี่ยนสัญญาณ DTMF เป็นสัญญาณ Digital ได้ การถอดรหัสความถี่โทรศัพท์ คือ การแปลงสัญญาณความถี่ที่เกิดจากการกดปุ่มเครื่องโทรศัพท์ (ชนิด Tone หรือ DTMF) ให้เป็นระบบตัวเลขทางดิจิตอลขนาด 4 บิต

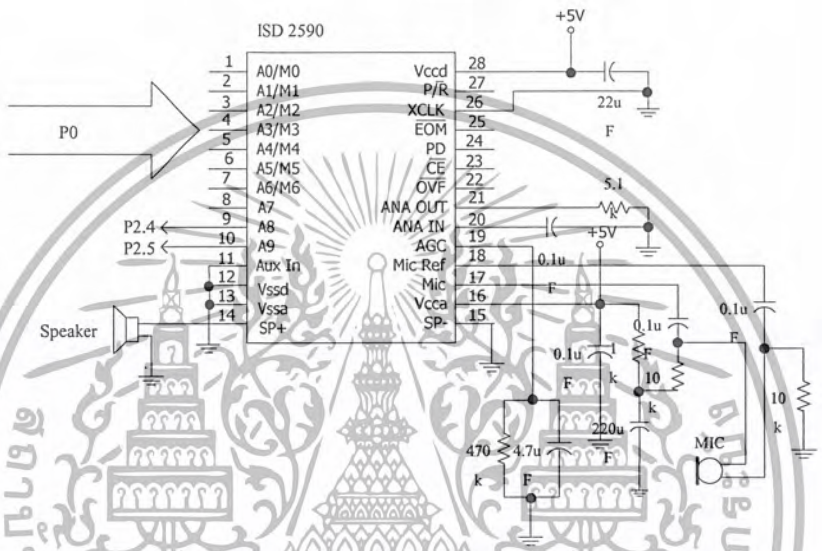


ภาพที่ 3.2 วงจรสัญญาณ DTMF เป็นสัญญาณ Digital

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการกดปุ่มจากโทรศัพท์วงจร DTMF ที่ขา 15 (STD) นั้นจะมีสัญญาณ positive pulse เกิดขึ้น 1 ครั้ง เมื่อมีการกดปุ่ม 1 ปุ่ม ดังนั้นเราจะตรวจจับสัญญาณที่ขานี้เมื่อมีการกดปุ่ม ตัวเลขใดๆ ทำให้ไมโครคอนโทรลเลอร์ P1.4- P1.7 รับสัญญาณดิจิตอล จากขาสัญญาณขา 11- 14 (Q1-Q4) เพื่อที่ไมโครคอนโทรลเลอร์จะได้ทำงานต่อไป

3.3.2วงจรเสียงตอบรับและวงจรบันทึกเสียง



รูปที่ 3.3 วงจรเสียงตอบรับและวงจรบันทึกเสียง

วงจรเสียงตอบรับและวงจรบันทึกเสียงใช้วงจรเดียวกัน จะแตกต่างที่ตัวโปรแกรมนั่นคือ ขา 27 . เมื่อทำการบันทึกจะต่อกราวด์ ถ้าจะตอบเสียงจะให้ขานี้ต่อกับ Vcc ซึ่งขา 24 PD ขา 23 CE จะต่อกราวด์ ในทั้ง 2 กรณี จากไอซี ISD2590 สามารถออกแบบการทำงานได้หลายแบบ ในที่นี้ออกแบบการทำงานของวงจร ที่สัมพันธ์กับไมโครคอนโทรลเลอร์ โดยเริ่มแรกทำการต่อวงจรเสียงตอบรับและวงจรบันทึกเสียง กับวงจรไมโครคอนโทรลเลอร์ แล้วเขียนโปรแกรมสำหรับอัดเสียงลงไมโครคอนโทรลเลอร์ เพื่อควบคุมไอซี 2590 ซึ่งจะกำหนดแอดเดรสตำแหน่งต่างๆ แล้วจะกำหนดระยะเวลาให้เหมาะสม กับระยะเวลาของเสียงที่ต้องการบันทึก ดังเช่น ช่วงแรกเป็นช่วง intro ที่ address กำหนดระยะเวลาที่เหมาะสม จากนั้นบันทึกเสียงค่าตัวเลข ตั้งแต่ 0-9

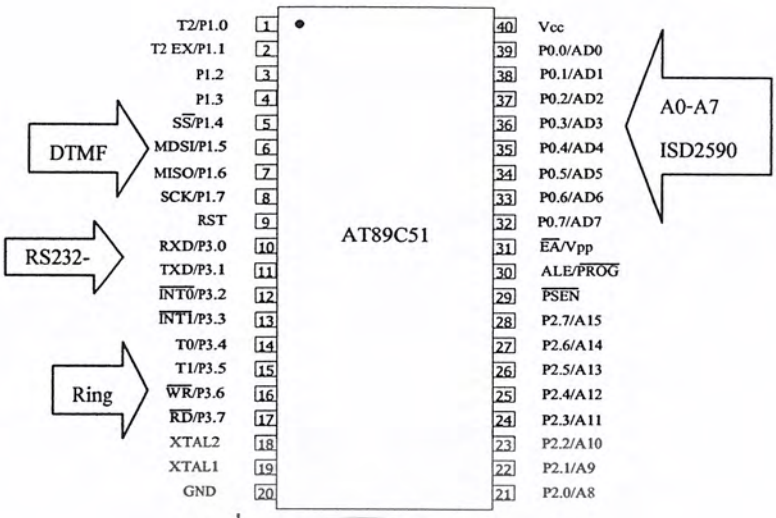
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพที่ 3.2 การบันทึกและตอบเสียง

3.3.4 วงจรควบคุมระบบ

วงจรนี้เปรียบเสมือนสมองคอยสั่งการให้วงจรทั้งหมดทำงานตามต้องการ ทางด้าน Hard Ware ของวงจรส่วนนี้ไม่ยุ่งยากเท่าไร จะยุ่งยากตรงที่ต่อวงจรนี้ให้ทำงานสัมพันธ์กับส่วนต่างๆ ดังนี้ P0.X ต่อกับ ISD2590 ขา CE กับขา PD จะต่อที่ขา P2.6 , P2.7 ตามลำดับ P2.2 ใช้เป็นพอร์ตอินพุท ที่ส่งมาจากไอซี DTMF และไอซี 555 P1.4– P1.7 ต่อกับ Q1 – Q4 ของไอซี DTMF ขา 18 , 19 ต่อกับ คริสตัล 12 MHz. โดยมีคาปาซิเตอร์ 22 pF ต่อร่วมด้วย ขา EA ต่อกับแหล่งจ่ายไฟ Vcc เพื่อใช้หน่วยความจำภายในไมโครคอนโทรลเลอร์ และขารีเซตขา 9 ต่อกับวงจรรีเซต

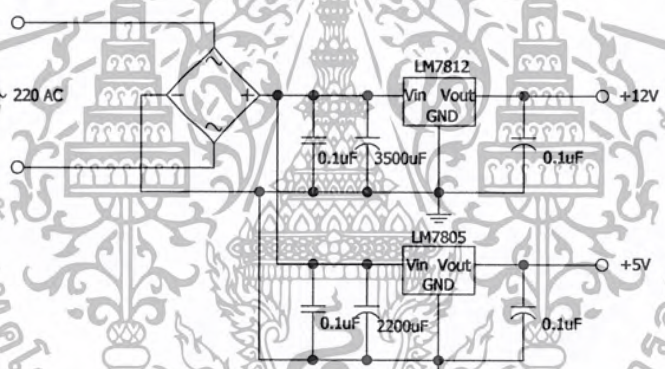
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.4 วงจรควบคุมระบบ

3.3.5 วงจรเพาเวอร์ซัพพลาย

วงจรเพาเวอร์ซัพพลายมีแหล่งจ่าย 2 แหล่งคือ ไฟตรง 12 โวลท์ และ 5 โวลท์



ภาพที่ 3.6 วงจรเพาเวอร์ซับพลาย

ซึ่งแหล่งจ่าย 12 โวลท์ จ่ายให้วงจรรีเลย์กับวงจรขยายขนาด 3 วัตต์ ส่วนแหล่งจ่าย 5 โวลท์ จะนำจ่ายให้วงจรทั้งหมดของโปรเจคนี้ วงจรนี้ประกอบด้วย วงจรบริดจ์เรกติฟายประกอบด้วยไดโอด 4 ตัว เพื่อแปลงให้เป็นไฟตรง กรองไฟตรงที่ได้ให้เรียบโดยคอนเดนเซอร์ และรักษาระดับแรงดันด้วยไอซีเรกูเลเตอร์ 12 โวลท์ และ 5 โวลท์ ที่ขาเอาท์พุทของไอซีเรกูเลเตอร์แต่ละตัวจะต่อตัวเก็บประจุ 0.1 uf แบบแทนทาลัม เพื่อกำจัดสัญญาณรบกวนความถี่สูงที่เกิดจากตัวเรกูเลเตอร์หรือจากไอซีตัวอื่นๆ

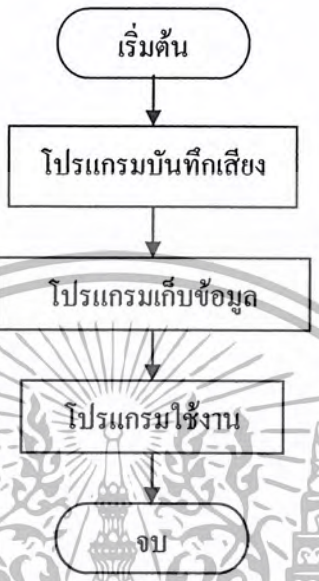
3.4 ออกแบบลายวงจรพิมพ์

เมื่อทดลองวงจรบนโฟรโตบอร์ดจนแน่นอนแล้ว ทำการออกวงจรเขียนลายวงจรและลายพิมพ์ โดยใช้โปรแกรมโพรเทล (Protel) เริ่มด้วยออกแบบลายวงจรก่อน เพราะสามารถทำแบบ Auto route คือ ให้โปรแกรมออกแบบลายพิมพ์เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ออกแบบโปรแกรม

การออกแบบโปรแกรมจะมี 3 ส่วนคือ ออกแบบโปรแกรมบันทึกเสียง ออกแบบโปรแกรมเก็บข้อมูลในหน่วยความจำ และออกแบบโปรแกรมใช้งาน โปรแกรมแต่ละส่วนนั้น จะทำการเขียนโฟล์วชาร์ตเพื่อความสะดวกในการจัดลำดับขั้นตอน และการตรวจสอบ



แผนภาพที่ 3.3 โฟล์วชาร์ตหลักที่ใช้ในการทำโครงงาน

3.6 ทดลองและวิเคราะห์การทำงาน

จะทำการทดลองวงจรทีละส่วนให้ได้การทำงานตามที่ได้ออกแบบไว้ จากนั้นจะนำวงจรทั้งหมดมาต่อรวมกันเพื่อทดสอบการทำงานจริง ซึ่งได้เกิดข้อผิดพลาดหลายๆ อย่าง ดังนั้นจึงตรวจสอบการทำงานของวงจรใหม่ โดยให้วงจรทั้งหมดทำงาน และทำการวัดสัญญาณทีละวงจร วิเคราะห์ผลที่ได้กับการออกแบบเพื่อแก้ไขปรับปรุงให้วงจรทำงานได้ถูกต้องสมบูรณ์ จากนั้นจะทำการเขียนโปรแกรม ซึ่งจะต้องดำเนินตามขั้นตอนเดิม คือ จะเขียนโปรแกรมให้กับวงจรทีละส่วนแล้วทำการตรวจสอบ จนได้ผลเป็นที่น่าพอใจแล้วจึงนำวงจรทั้งหมดมาต่อและทดสอบให้กับโปรแกรมที่เขียนแล้ววัดรูปคลื่นสัญญาณเพื่อวิเคราะห์หาข้อผิดพลาด พร้อมทั้งแก้ไขให้ถูกต้อง

3.7 ประกอบวงจรบนลายวงจร

เป็นอีกขั้นตอนหนึ่งที่จะต้องทำการต่อทีละวงจรแล้วทดสอบให้ได้ผลกับที่ได้ทดลองไว้ ซึ่งบางวงจรผลที่ได้ดีกว่าก็ไม่มีปัญหา แต่บางวงจรผลที่ได้ไม่ตรงกับการทดลองต้องทำการวิเคราะห์แก้ไขปรับปรุงให้ถูกต้อง เมื่อทดลองวงจรทั้งหมดบนลายวงจรแล้ว ขั้นต่อมาจะทดสอบทั้งระบบ ขั้นตอนนี้มีปัญหาต่างๆ มากมายจึงต้องทำการวิเคราะห์หาข้อผิดพลาด และปรับปรุงแก้ไขให้ถูกต้องต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 ลงกล่อง

ขั้นแรกต้องพิจารณากล่องที่จะมาใช้กับโครงงาน จากนั้นต้องทำการออกแบบตำแหน่งการวางอุปกรณ์ต่างๆให้เหมาะสมและสวยงาม เมื่อได้เจาะการวางอุปกรณ์เรียบร้อย ก็ประกอบวงจรทั้งหมดลงกล่อง พร้อมจัดทางเดินสายสัญญาณต่างให้เป็นระเบียบเรียบร้อย จากนั้นจะทำการตรวจทานตำแหน่งอุปกรณ์ต่างๆ อีครั้ง และขั้นสุดท้ายจะเป็นการทดสอบการทำงานของโครงงานอีกครั้ง ถ้าโครงงานไม่สามารถทำงานตามที่ต้องการ ก็จะต้องมาตรวจสอบการทำงานของวงจรแต่ละส่วนอีกครั้ง นั้นคือต้องวิเคราะห์ข้อผิดพลาดพร้อมทั้งแก้ไขให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

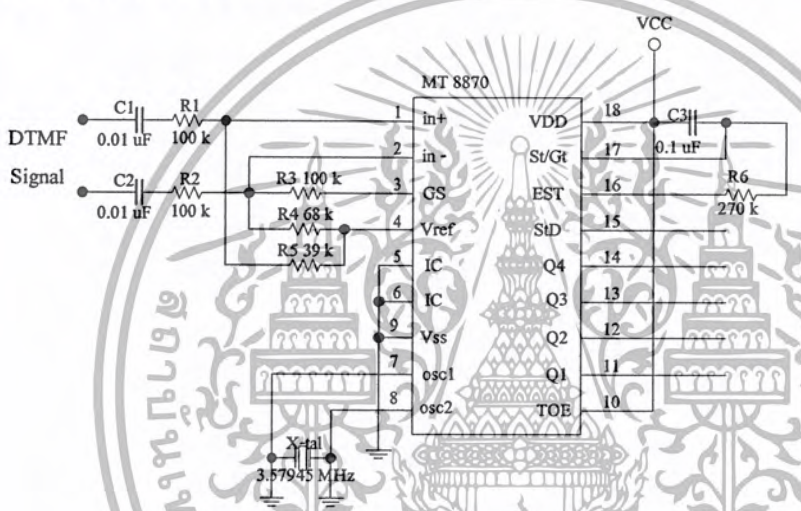
การทดลองและผลการทดลอง

รูปคลื่นและสัญญาณต่างๆสามารถนำมาวิเคราะห์เพื่อนำมาใช้ประโยชน์ตามหลักการ ดังนั้นจึงเป็นส่วนสำคัญอันดับแรกที่ต้องให้ความสนใจเป็นพิเศษ เมื่อทำการทดลองก็สามารถนำมาวิเคราะห์รูปคลื่นสัญญาณเหล่านี้ เพื่อให้ได้ผลงานตามต้องการ โดยมีลำดับการทดลองและผลการทดลอง ดังนี้

4.1 การทดลอง

4.1.1 วงจรถอดรหัสความถี่โทรศัพท์ โดยไอซี MT 8870

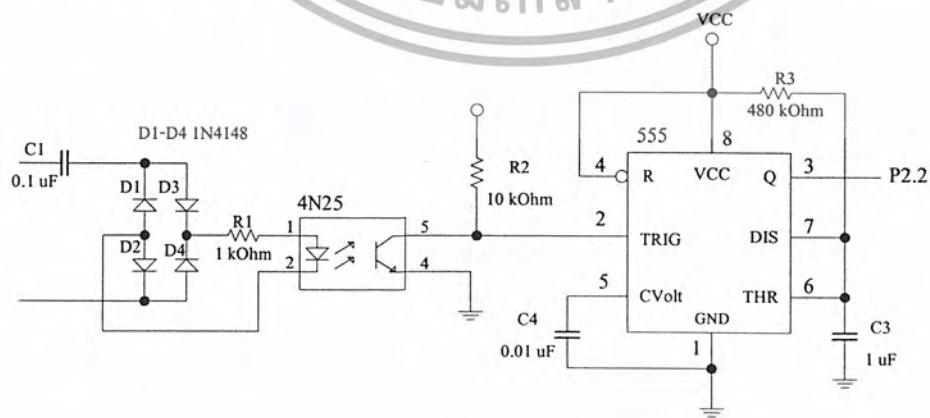
วัตถุประสงค์ของการทดลองวงจรนี้เพื่อให้ทราบว่า สามารถถอดรหัสความถี่มาเป็นเลขฐานสองได้ การทดลองทำได้โดยการต่อ LED 4 ตัว ที่ Q1-Q4 ดังภาพที่ 4.1 จากนั้นต่อสัญญาณโทรศัพท์ให้ที่อินพุทของ MT8870 สังเกตการติด-ดับของ LED



ภาพที่ 4.1 วงจรถอดรหัส DTMF

4.1.2 วงจรตรวจจับสัญญาณกระดิ่ง

วัตถุประสงค์ของวงจรนี้เพื่อที่จะให้เราทราบว่า สัญญาณกระดิ่งที่รับมามีแรงดันประมาณ 75-90 โวลท์อาร์เอ็มเอส มีความถี่ 50 HZ มีสัญญาณ 1 วินาที หยุด 4 วินาที



ภาพที่ 4.2 วงจรตรวจจับสัญญาณกระดิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผ่านวงจรตรวจจับสัญญาณกระดิ่งจะถูกเปลี่ยนเป็นสัญญาณดิจิตอลระดับแรงดัน 0 V และ 5 V โดยการต่อ C1 ค่า 0.1 uF 63 V เพื่อกันไฟกระแสตรง จากนั้นผ่านวงจรเร็กติไฟร์เออร์ D1-D4 ให้ผ่านเฉพาะกระแสด้านบวก และถูกจำกัดกระแสด้วยความต้านทาน R1 ค่า 1 K ส่งให้ Optoisolator ทำงานจะได้สัญญาณที่ขา 5 ระดับแรงดันประมาณ 0 V และ 5 V

4.1.3 การทดลองส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์

- ทำการเขียนโปรแกรมเพื่อส่งข้อมูล ดังนี้

```
                ORG     0000H
                MOV     TMOD,#20H
                MOV     PCON,#00H
                MOV     SCON,#50H
                MOV     TH1,#0FBH
                SETB    TR1
                CLR     RI
                CLR     TI
                ;
                MOV     DPTR,#SERIAL
                ACALL   TX
TX:             CLR     TI
LOOP:           CLR     A
                MOVC    A,@A+DPTR
                INC     DPTR
                CJNE    A,#0FFH,LOOP1
                RET
LOOP1:          MOV     SBUF,A
                JNB     TI,$
                CLR     TI
                ACALL   DELAY
                AJMP    LOOP
DELAY:          MOV     R7,#100
DEL:            MOV     R6,#0E6H
DEL1:           NOP
                NOP
                DJNZ    R6,DEL1
                DJNZ    R7,DEL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
SERIAL:         DB      00AH,00DH
                DB      '/16012545/610103/a',00AH,00DH
                DB      00AH,00DH
                END
```

- ทำการแอสเซมเบอร์ จ่ายไฟให้กับวงจร จากนั้นทำการรันโปรแกรมโดยการกดสวิตช์ RESET
- เปิดโปรแกรม Test Loop ของโปรแกรมตรวจผลรางวัลลอตเตอรี่
- สังเกตการแสดงผลบนหน้าจอคอมพิวเตอร์

4.2 ผลการทดลอง

4.2.1 ผลการทดลองจากข้อ 4.1.1 สามารถให้ค่าออกมาได้ ดังนี้

ความถี่ต่ำ (Hz)	ความถี่สูง (Hz)	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0

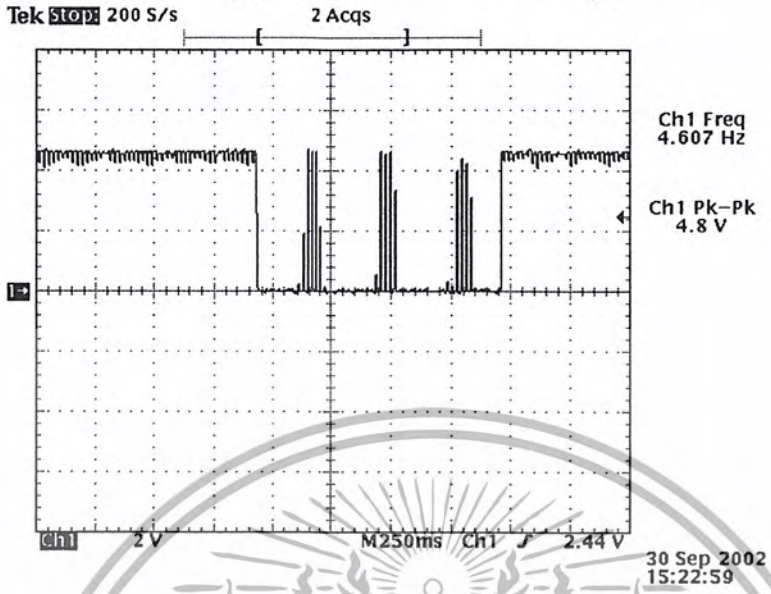
0 คือ LED ติด

1 คือ LED ดับ

ตารางที่ 4.1 แสดงค่าที่ได้จากวงจรถอดรหัส DTMF

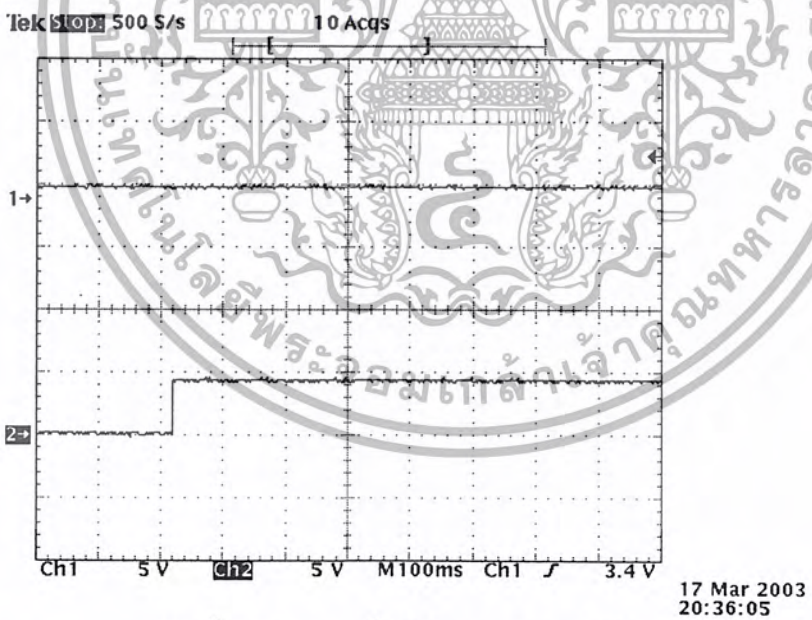
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลองจากข้อ 4.1.2 สัญญาณที่วัดได้จากวงจรตรวจจับสัญญาณกระดิ่ง



ภาพที่ 4.3 สัญญาณที่วัดได้จากขา 5 ของไอซี 4N25

จากรูปจะเห็นว่าสัญญาณที่ส่งให้ Optoisolator ทำงานจะได้สัญญาณที่ขา 5 ระดับแรงดันประมาณที่ 0 V และ 5 V แต่จะได้รับแรงที่ไม่คงที่ดังนั้น จึงสร้างวงจรโมโนสเตเบิลและกำหนดคาบของสัญญาณใหม่

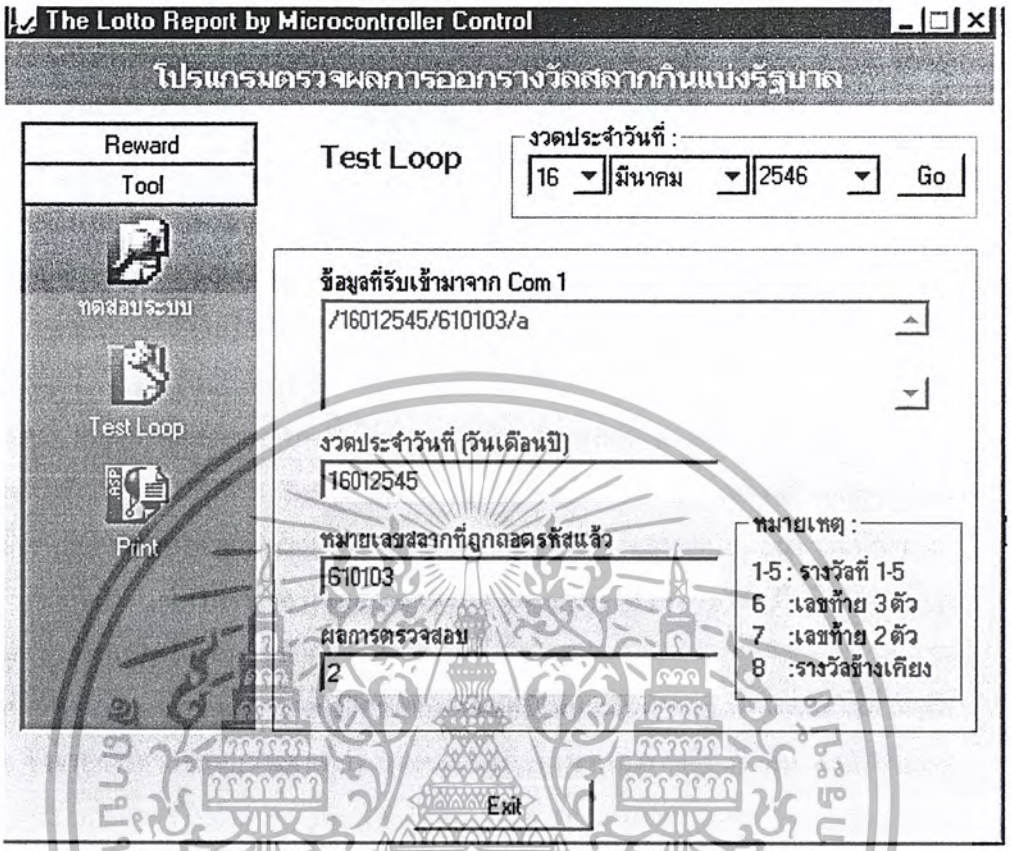


ภาพที่ 4.4 สัญญาณที่วัดได้จากขา 3 ของไอซี 555

จากรูปจะเห็นว่าสัญญาณที่ได้จะเป็นพัลส์ 1 ลูกคลื่นเพื่อส่งไปกระตุ้นให้ไมโครคอนโทรลเลอร์ทำงาน ทั้งนี้เราสามารถกำหนดคาบเวลาของสัญญาณใหม่ได้โดยการใช้ตัวต้านทานแบบปรับค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดลองจากข้อ 4.1.3 การส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์



ภาพที่ 4.5 ผลที่ได้จากการทดลองส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์

อย่างไรก็ตาม หัวใจสำคัญของการสื่อสารข้อมูลผ่านพอร์ตอนุกรมคือ การกำหนดอัตราบอด และรูปแบบของข้อมูลว่า มีจำนวนบิตเริ่มต้น,บิตของข้อมูล,บิตตหยุด หรือว่ามีการตรวจสอบบิตพาริตี้หรือไม่ ถ้าหากข้อกำหนดเหล่านี้ในตัวรับและตัวส่งไม่ตรงกัน จะทำให้การถ่ายทอดข้อมูลเกิดการผิดพลาดได้อย่างง่ายดาย ส่งผลให้การสื่อสารข้อมูลล้มเหลวอย่างสิ้นเชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.ก้องเกียรติ ณ สีมา.ทฤษฎีและการใช้งานไทม์เมอร์ ไอซี 555.สำนักพิมพ์ศูนย์พิมพ์ดวงกมล,2538.

2.กฤษฎา วิศวธีรานนท์.เรียน/เล่น/ใช้ ไอซีดิจิตอล.สำนักพิมพ์ซีเอ็ด,2541.

3.คณะผู้จัดทำบริษัทซีเอ็ด.เข้าใจ/สร้าง/เล่น ไมโครโปรเซสเซอร์ 1-2.สำนักพิมพ์ซีเอ็ด,2542.

4.ธีรวัฒน์ ประกอบผล.การประยุกต์ใช้งานไมโครคอนโทรลเลอร์.สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น),2545.

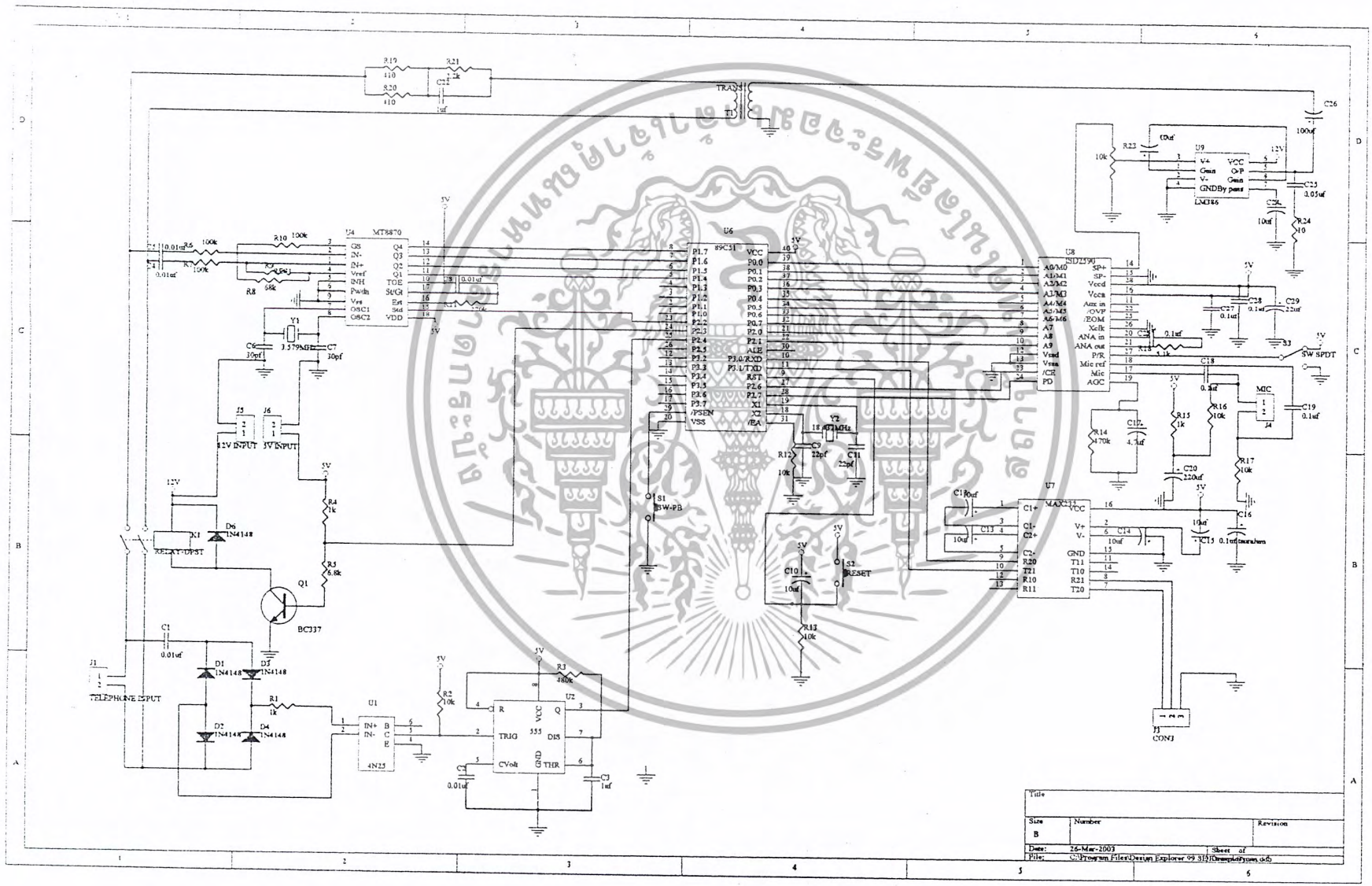
5.ชัยวัฒน์ ลิ้มพรจิตรและวิไลวรพจน์ กรแก้ววัฒนกูล.เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51. สำนักพิมพ์อินโนเวตีฟเอ็กเพอริเม้นต์,2544.

6.สมยศ จุณณะปิยะ.การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51.โรงพิมพ์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2543.

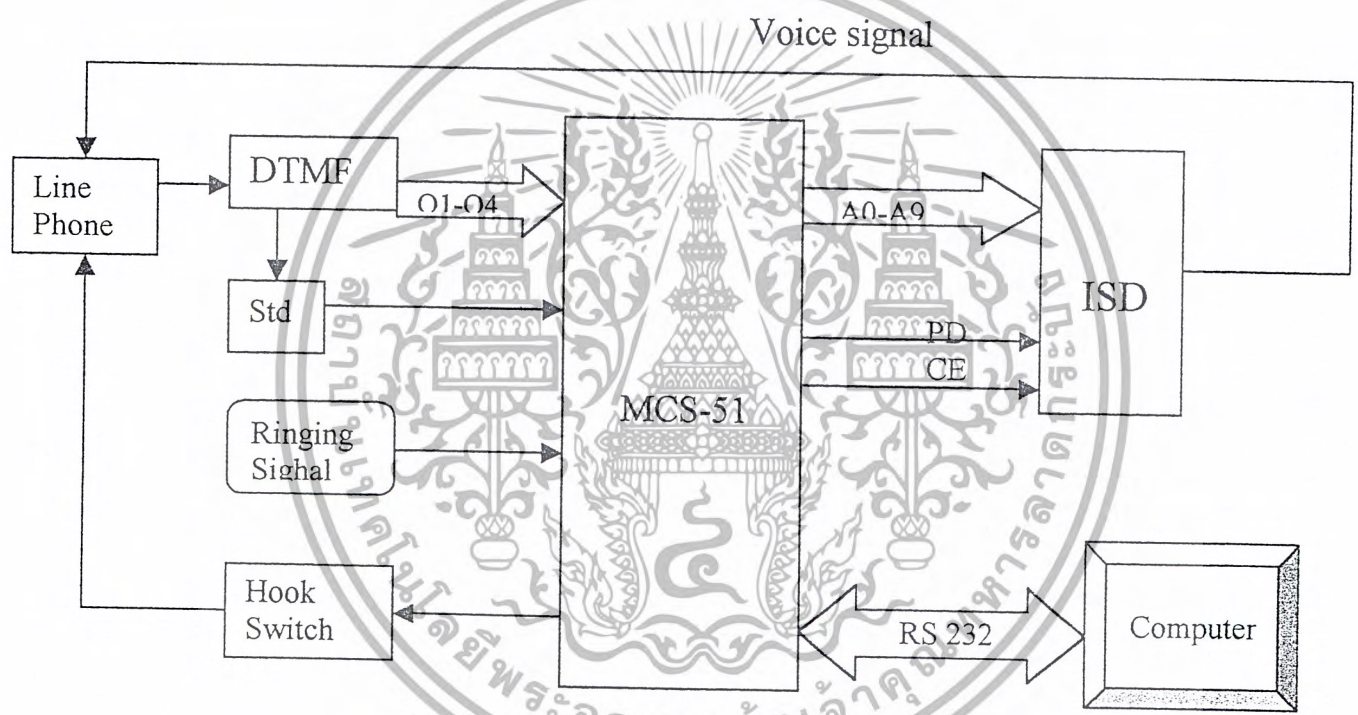
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

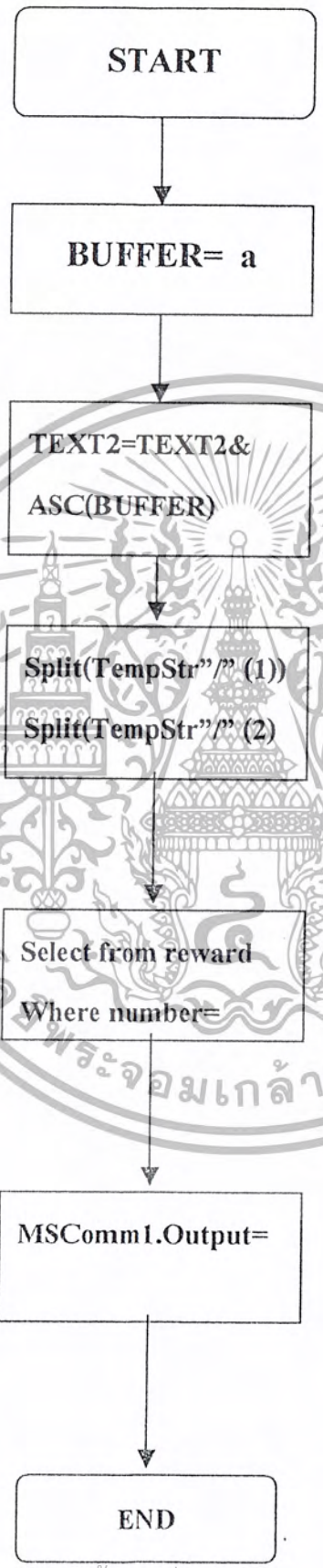
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


TRANS1
T1
U4 MT8870
U6 89C51
U8 ISD2590
U7 MAX232
U9 LM386
U1 4N25
U2 555
MIC
SW SPDT
SW-PB
RESET
RELAY-DPST
TELEPHONE INPUT
CONJ
12V INPUT
5V INPUT
BC337
Title
Size
B
Number
Revision
Date: 26-Mar-2003
Sheet of
File: C:\Program Files\Design Explorer 99 SE\Examples\mm.ddb

BLOCK DIAGRAM



PROTOCAL = /_ _(วัน)_ _(เดือน) _ _ _ _(ปี) / _ _ _ _ _ _(หมายเลข) / a



1 รางวัลที่ 1

2 รางวัลที่ 2

3 รางวัลที่ 3

4 รางวัลที่ 4

5 รางวัลที่ 5

6 รางวัลเลขท้าย 3 ตัว

7 รางวัลเลขท้าย 2 ตัว

8 รางวัลข้างเคียง

0 ไม่ถูกรางวัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

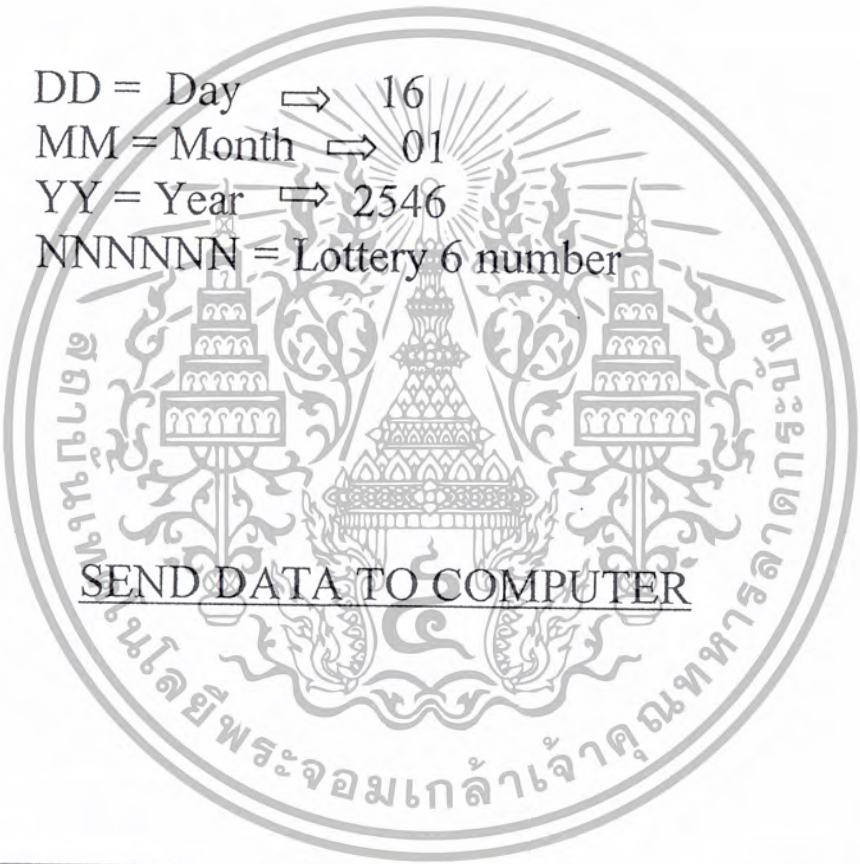
Main Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA FROM PORT 1

MOV A,P1 ⇦ DDMMYYYYNNNNNN

NOTE: DD = Day ⇨ 16
MM = Month ⇨ 01
YY = Year ⇨ 2546
NNNNNN = Lottery 6 number



SEND DATA TO COMPUTER

ASCII ⇨ /DDMMYYYY/NNNNNN/a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim reward As Integer
Dim strSQL As String
Dim ReadBuffer As String
Dim no As Long
Private Sub Combo1_Click()
Grid.RecordSource = ""
Grid.HoldFields
Grid.Rebind

End Sub

Private Sub Combo2_Click()
Grid.RecordSource = ""
Grid.HoldFields
Grid.Rebind
End Sub

Private Sub Combo3_Change()
Grid.RecordSource = ""
Grid.HoldFields
Grid.Rebind
End Sub

Private Sub Combo3_Click()
lotID = Format(CLng(Combo1.Text), "00") & Format(Combo2.ListIndex +
1, "00") & Combo3.Text

Grid.RecordSource = ""
Grid.HoldFields
Grid.Rebind
End Sub

Private Sub Combo3_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
Case 8, 48 To 57
Case Else
    KeyAscii = 0
End Select

End Sub

Private Sub Command1_Click()
Unload Me
End Sub

Private Sub Command2_Click()
' ทดสอบระบบค้นหา
If Text1.Text = "" Then
    Label6.Caption = "กรุณาป้อนหมายเลขสลาก แล้วกดปุ่มทดสอบ"
    Exit Sub
End If

lotID = Format(CLng(Combo1.Text), "00") & Format(Combo2.ListIndex +
1, "00") & Combo3.Text
Dim sqltxt As String
Dim rs As New ADODB.Recordset
' ตรวจสอบรางวัลที่ 1-5 และรางวัลข้างเคียง
sqltxt ="select *from reward where number='"& Text1 & "' and id="& lotID
& "order by price desc"
rs.Open sqltxt, cn, adOpenForwardOnly, adLockReadOnly, adCmdText
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Not rs.EOF Then
    Label6.Caption ="ท่านถูกรางวัลที่" & rs("reward") & " เงินรางวัลมูลค่า " & rs("price") &
" บาท"
    rs.Close
    Exit Sub
End If

rs.Close
' ตรวจสอบรางวัลเลขท้าย 3 ตัว
sqltxt = "select * from reward where number = '" & Right(Text1, 3) &
"' and id=" & lotID & " and reward=6"
rs.Open sqltxt, cn, adOpenForwardOnly, adLockReadOnly, adCmdText
If Not rs.EOF Then
     Label6.Caption = "ท่านถูกรางวัลเลขท้าย 3 ตัว เงินรางวัลมูลค่า " & rs("price") & " บาท"
    rs.Close
    Exit Sub
End If

rs.Close
' ตรวจสอบรางวัลเลขท้าย 2 ตัว
sqltxt = "select * from reward where number = '" & Right(Text1, 2) &
"' and id=" & lotID & " and reward=7"
rs.Open sqltxt, cn, adOpenForwardOnly, adLockReadOnly, adCmdText
If Not rs.EOF Then
     Label6.Caption = "ท่านถูกรางวัลเลขท้าย 2 ตัว เงินรางวัลมูลค่า " & rs("price") & " บาท"
Else
     Label6.Caption = "ท่านไม่ถูกรางวัลสลากกินแบ่งรัฐบาล"
End If
rs.Close



End Sub
Private Sub Command3_Click()
lotID = Format(CLng(Combo1.Text), "00") & Format(Combo2.ListIndex +
1, "00") & Combo3.Text
strSQL = "select * from reward where id=" & lotID & " and reward=" &
reward

Grid.DatabaseName = cn.ConnectionString
Grid.RecordSource = strSQL
Grid.HoldFields
Grid.Rebind
CheckStatus

End Sub

Private Sub Form_Load()
Dim strFile As String

If App.PrevInstance Then
    MsgBox "โปรแกรมนี้ได้ถูกเปิดใช้งานอยู่แล้วครับ", vbInformation
End
End If
Set cn = New ADODB.Connection

strFile = App.Path & "\data.mdb"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OpenDatabase_Proc:
On Error GoTo EH_FormLoad

cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="& strFile &
";Persist Security Info=False"


With Combo1
     .Clear
     .AddItem "1"
     .ItemData(Combo1.NewIndex) = 1
     .AddItem "16"
     .ItemData(Combo1.NewIndex) = 16
     If Format(Date, "dd") < 16 Then
         .ListIndex = 0
     Else
         .ListIndex = 1
     End If
End With
With Combo2
     .Clear
     .AddItem "มกราคม"
     .AddItem "กุมภาพันธ์"
     .AddItem "มีนาคม"
     .AddItem "เมษายน"
     .AddItem "พฤษภาคม"
     .AddItem "มิถุนายน"
     .AddItem "กรกฎาคม"
     .AddItem "สิงหาคม"
     .AddItem "กันยายน"
     .AddItem "ตุลาคม"
     .AddItem "พฤศจิกายน"
 .AddItem "ธันวาคม"
 .ListIndex = Month(Date)-1
End With
Dim i As Integer
With Combo3
 .Clear
    For i = 2540 To 2560
 .AddItem i
 .ItemData(Combo3.NewIndex) = i
    Next
 .Text = Year(Date)+543
End With

reward = 1
Label12.Caption = "รางวัลที่ 1"
Command3_Click


    MSComm1.Settings = "9600,n,8,1"
    MSComm1.InputLen = 1
    MSComm1.RThreshold = 1
    MSComm1.CommPort = 1
    MSComm1.PortOpen = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Exit Sub

EH_FormLoad:
    Select Case Err.Number
    Case -2147467259, 3024, 3044 'Couldn't Find File

    With dlgFile
        .DialogTitle = "เลือกแฟ้มฐานข้อมูล"
        .Filter = "ฐานข้อมูล|*.mdb|All Files (*.*)|*.*"
        .FilterIndex = 0
        .CancelError = False
        .Action = 1

        strFile = .FileName
    End With

    If strFile <> "" Then
        Resume OpenDatabase_Proc
    Else
        End
    End If

Case Else
        MsgBox Err.Number & vbNewLine & Err.Description,
vbCritical
        End
End Select

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As
Integer)
MSComm1.PortOpen = False

End Sub

Private Sub Grid_AfterDelete()
Command3_Click
End Sub

Private Sub Grid_BeforeUpdate(ByVal Cancel As GridEX20.JSRetBoolean)
Grid.Value(1) = lotID
Grid.Value(3) = reward
Grid.Value(6) = Grid.Row

Select Case reward
Case 1
        Grid.Value(5) = 3000000 '
Case 2
        Grid.Value(5) = 100000
Case 3
        Grid.Value(5) = 40000
Case 4
        Grid.Value(5) = 20000
Case 5
        Grid.Value(5) = 10000
Case 6   ' รางวัลเลขท้าย 3 ตัว
        Grid.Value(5) = 2000
Case 7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Grid.Value(5) = 1000
Case 8
            Grid.Value(5) = 50000
End Select
End Sub

Private Sub Grid_RowColChange(ByVal LastRow As Long, ByVal LastCol As
Integer)
CheckStatus
End Sub


Private Sub MSComm1_OnComm()
On Error GoTo com_err
Dim buffer As Variant
Dim TempStr As String
Dim n As String
Dim r As Long

    Select Case MSComm1.CommEvent
    Case comEvReceive
    buffer = MSComm1.Input
    End Select
    '    Text2 = Text2 & Asc(buffer)
    ReadBuffer = ReadBuffer & buffer


    If buffer = "a" Then
        TempStr = ReadBuffer
        If Mid(TempStr, 1, 1) = "/" Then
            r = Split(TempStr, "/")(1)      'ถอดรหัสงวดวันที่
            n = Split(TempStr, "/")(2)      'ถอดรหัสหมายเลขสลาก
        End If
        TempStr = ""

        If Label2.Caption = "Test Loop" Then
            Text2 = ReadBuffer & vbNewLine & Text2
            Text3 = r
            Text4 = n
        End If
        ReadBuffer = ""
        check r, n
    End If
Exit Sub

com_err:
buffer = ""
ReadBuffer = ""
    'MsgBox Err.Number & Err.Description, vbInformation, "Comm Error
"

End Sub
Sub check(ByRef mLotID As Long, ByRef mNumber As String)

Dim sqltxt As String
Dim rs As New ADODB.Recordset
' ตรวจสอบรางวัลที่ 1-5 และรางวัลข้างเคียง
sqltxt = "select * from reward where number='" & mNumber & "' and
id=" & mLotID & " order by price desc"
rs.Open sqltxt, cn, adOpenForwardOnly, adLockReadOnly, adCmdText
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Not rs.EOF Then
     MSComm1.Output = CStr(rs("reward"))
     If Label2.Caption = "Test Loop" Then Text5 = rs("reward")
    rs.Close
    Exit Sub
End If

rs.Close
' ตรวจสอบรางวัลเลขท้าย 3 ตัว
sqltxt = "select * from reward where number = '" & Right(mNumber, 3)
& "' and id=" & mLotID & " and reward=6"
rs.Open sqltxt, cn, adOpenForwardOnly, adLockReadOnly, adCmdText
If Not rs.EOF Then
     MSComm1.Output = CStr(rs("reward"))
     If Label2.Caption = "Test Loop" Then Text5 = rs("reward")
    rs.Close
    Exit Sub
End If

rs.Close
' ตรวจสอบรางวัลเลขท้าย 2 ตัว
sqltxt = "select * from reward where number = '" & Right(mNumber, 2)
& "' and id=" & mLotID & " and reward=7"
rs.Open sqltxt, cn, adOpenForwardOnly, adLockReadOnly, adCmdText
If Not rs.EOF Then
     MSComm1.Output = CStr(7)
     If Label2.Caption = "Test Loop" Then Text5 = 7
Else
     MSComm1.Output = CStr(0)    ' หากไม่ถูกรางวัลใดๆให้ output=0
     If Label2.Caption = "Test Loop" Then Text5 = 0
End If
rs.Close

End Sub

Private Sub SSListBar1_ListItemClick(ByVal ItemClicked As
Listbar.SSListItem)
If SSListBar1.CurrentGroup.Index = 1 Then
    reward = ItemClicked.Index
    Label2.Caption = ItemClicked.Text
    Command3_Click
    Grid.Top = 1440
    Frame2.Top = 8000
    Frame3.Top = 8000
    Select Case ItemClicked.Index
        Case 6
                Grid.Columns(2).Format = "000"
        Case 7
                Grid.Columns(2).Format = "00"
        Case Else
                Grid.Columns(2).Format = "000000"
        End Select
Else
    If ItemClicked.Index = 1 Then
            Label2.Caption = ItemClicked.Text
            Grid.Top = 8000
            Frame2.Top = 1440
            Frame3.Top = 8000
    ElseIf ItemClicked.Index = 2 Then
            Label2.Caption = ItemClicked.Text
            Grid.Top = 8000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Frame2.Top = 8000
            Frame3.Top = 1440
            frmCom2.Show , Me
    ElseIf ItemClicked.Index = 3 Then
    lotID = Format(CLng(Combo1.Text), "00") & Format(Combo2.ListIndex
+ 1, "00") & Combo3.Text

            ActiveReport1.Show

    End If
End If
End Sub
Sub CheckStatus()
Dim status As Boolean
status = False
Select Case reward
Case 1
        If Grid.RowCount = 0 Then status = True
Case 2
        If Grid.RowCount < 5 Then status = True
Case 3
        If Grid.RowCount < 10 Then status = True
Case 4
        If Grid.RowCount < 50 Then status = True
Case 5
        If Grid.RowCount < 100 Then status = True
Case 6
        If Grid.RowCount < 4 Then status = True
Case 7
        If Grid.RowCount < 1 Then status = True
Case 8
        If Grid.RowCount < 2 Then status = True
End Select
            Grid.AllowAddNew = status

End Sub

Private Sub Text1_GotFocus()
Command2.Default = True
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
Select Case KeyAscii
Case 8, 48 To 57
Case Else
    KeyAscii = 0
End Select

End Sub

Private Sub Text1_LostFocus()
Command2.Default = False
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
;*** Project  : The Lotto Report by Microcontroller Control    *****
;*** Programer: Mr. P.Damrongchai                              *****
;*** Complier : Syntronic RAD51                                *****
;*** Date     : March 29,2003                                  *****
;*********************************************************************
;
;
;
;*********************************************************************
;*******           Main Program                                *****
;*********************************************************************
;
PD          BIT     P2.7                    ; Control for ISD
CE          BIT     P2.6                    ; Control for ISD
RING        BIT     P2.4                    ; Ringing signal
HOOK        BIT     P2.3                    ; Hook switch on/off
STD         BIT     P2.2                    ; Std MT8870 DTMF
A9          BIT     P2.1                    ; ISD  A9
A8          BIT     P2.0                    ; ISD  A8
;
;*********************************************************************
;*******           Main Program                                ******
;*********************************************************************
;
                ORG     0000H
                ;
                MOV     PCON,#00H           ; SMOD = 0
                MOV     SCON,#50H           ; Serial(Mode 1),Ren=1
                MOV     TMOD,#20H           ; Timer1(Mode 2)
                MOV     TH1,#0F6H           ; 9600 Baud
                SETB    TR1                 ; Start Timer1
                CLR     RI
                CLR     TI
                ;
INITIAL:
                MOV     P0,#00H             ; A0-A7 for ISD
                MOV     P1,#0FFH            ; Input 4 bits DTMF
                MOV     P2,#0FFH            ; Control Port
                SETB    PD                  ; Stop sound
                CLR     HOOK                ; Disconnect
                ;
LAW:            JB      RING,$              ; Check bit 1
                LCALL   DELAY_1s
                JB      RING,HOOK_ON
                LJMP    LAW
                ;
HOOK_ON:
                LCALL   DELAY_100ms
                SETB    HOOK                ; Connect Line
                LCALL   DELAY_100ms
                LCALL   INTRO               ; Sound Intro
                LCALL   DELAY_100ms
                LCALL   DTMF_16N            ; Receive DTMF
                CJNE    A,#77H,CCC          ; No std
                LJMP    INITIAL
                ;

CCC:            LCALL   DELAY_100ms
                LCALL   SEND                ; Send to PC
                LCALL   DELAY_100ms
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    LCALL   RX                  ; Wait PC
                    LCALL   DELAY_100ms
                    CLR     HOOK                ; Disconnect
                    LJMP    LAW
                    ;


;*********************************************************************
;*****               Store DTMF to RAM                           ****
;*********************************************************************
DTMF_16N:                                       ; DTMF 16 Number
                    LCALL   RENT
                    CJNE    A,#77H,DD0          ; No std
                    LJMP    STOP_DTMF_16N       ; Exit DTMF_16N
DD0:                CJNE    A,#0BH,DM_0         ; Cancle (*)
                    LJMP    DTMF_16N
DM_0:               MOV     50H,A               ; Date 1

                    LCALL   RENT
                    CJNE    A,#77H,DD1
                    LJMP    STOP_DTMF_16N
DD1:                CJNE    A,#0BH,DM_1
                    LJMP    DTMF_16N
DM_1:               MOV     51H,A               ; Date 2

                    LCALL   RENT
                    CJNE    A,#77H,DD2
                    LJMP    STOP_DTMF_16N
DD2:                CJNE    A,#0BH,DM_2
                    LJMP    DTMF_16N
DM_2:               MOV     52H,A               ; Month 1

                    LCALL   RENT
                    CJNE    A,#77H,DD3
                    LJMP    STOP_DTMF_16N
DD3:                CJNE    A,#0BH,DM_3
                    LJMP    DTMF_16N
DM_3:               MOV     53H,A               ; Month 2
                    ;
                    LCALL   RENT
                    CJNE    A,#77H,DD4
                    LJMP    STOP_DTMF_16N
DD4:                CJNE    A,0BH,DM_4
                    LJMP    DTMF_16N
DM_4:               MOV     54H,A               ; Year 1
                    ;
                    LCALL   RENT
                    CJNE    A,#77H,DD5
                    LJMP    STOP_DTMF_16N
DD5:                CJNE    A,#0BH,DM_5
                    LJMP    DTMF_16N
DM_5:               MOV     55H,A               ; Year 2
                    ;
                    LCALL   RENT
                    CJNE    A,#77H,DD6
                    LJMP    STOP_DTMF_16N
DD6:                CJNE    A,#0BH,DM_6
                    LJMP    DTMF_16N
DM_6:               MOV     56H,A               ; Year 3
                    ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   RENT
                CJNE    A,#77H,DD7
                LJMP    STOP_DTMF_16N
DD7:            CJNE    A,#0BH,DM_7
                LJMP    DTMF_16N
DM_7:           MOV     57H,A           ; Year 4
                ;
                LCALL   RENT
                CJNE    A,#77H,DD8
                LJMP    STOP_DTMF_16N
DD8:            CJNE    A,#0BH,DM_8
                LJMP    DTMF_16N
DM_8:           MOV     58H,A           ; Number 1
                ;
                LCALL   RENT
                CJNE    A,#77H,DD9
                LJMP    STOP_DTMF_16N
DD9:            CJNE    A,#0BH,DM_9
                LJMP    DTMF_16N
DM_9:           MOV     59H,A           ; Number 2
                ;
                LCALL   RENT
                CJNE    A,#77H,DD10
                LJMP    STOP_DTMF_16N
DD10:           CJNE    A,#0BH,DM_10
                LJMP    DTMF_16N
DM_10:          MOV     5AH,A           ; Number 3
                ;
                LCALL   RENT
                CJNE    A,#77H,DD11
                LJMP    STOP_DTMF_16N
DD11:           CJNE    A,#0BH,DM_11
                LJMP    DTMF_16N
DM_11:          MOV     5BH,A           ; Number 4
                ;
                LCALL   RENT
                CJNE    A,#77H,DD12
                LJMP    STOP_DTMF_16N
DD12:           CJNE    A,#0BH,DM_12
                LJMP    DTMF_16N
DM_12:          MOV     5CH,A           ; Number 5
                ;
                LCALL   RENT
                CJNE    A,#77H,DD13
                LJMP    STOP_DTMF_16N
DD13:           CJNE    A,#0BH,DM_13
                LJMP    DTMF_16N
DM_13:          MOV     5DH,A           ; Number 6
                ;
STOP_DTMF_16N:  RET
                ;
;*****************************************************************
;*****           Send to Computer                            *****
;*****************************************************************
;
SEND:                                   ; /ddmmyyyy/nnnnnn/a
                MOV     A,#'/'          ; /
                LCALL   SBYTE
                ;
                MOV     A,50H           ; d
                LCALL   NUMBER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,51H           ; d
                LCALL   NUMBER
                ;
                MOV     A,52H           ; m
                LCALL   NUMBER
                MOV     A,53H           ; m
                LCALL   NUMBER
                ;
                MOV     A,54H           ; y
                LCALL   NUMBER
                MOV     A,55H           ; y
                LCALL   NUMBER
                MOV     A,56H           ; y
                LCALL   NUMBER
                MOV     A,57H           ; y
                LCALL   NUMBER
                ;
                MOV     A,#'/'          ; /
                LCALL   SBYTE
                ;
                MOV     A,58H           ; n1
                LCALL   NUMBER
                MOV     A,59H           ; n2
                LCALL   NUMBER
                MOV     A,5AH           ; n3
                LCALL   NUMBER
                MOV     A,5BH           ; n4
                LCALL   NUMBER
                MOV     A,5CH           ; n5
                LCALL   NUMBER
                MOV     A,5DH           ; n6
                LCALL   NUMBER
                ;
                MOV     A,#'/'          ; /
                LCALL   SBYTE
                MOV     A,#'a'          ; a , End of message
                LCALL   SBYTE
                RET
                ;
;*************************************************************
;*****           Converse A to ASCII                     *****
;*************************************************************
;
NUMBER:         CJNE    A,#0AH,N0       ; DTMF 0
                MOV     A,#'0'
                LCALL   SBYTE
N0:             CJNE    A,#01H,N1       ; DTMF 1
                MOV     A,#'1'
                LCALL   SBYTE
N1:             CJNE    A,#02H,N2       ; DTMF 2
                MOV     A,#'2'
                LCALL   SBYTE
N2:             CJNE    A,#03H,N3       ; DTMF 3
                MOV     A,#'3'
                LCALL   SBYTE
N3:             CJNE    A,#04H,N4       ; DTMF 4
                MOV     A,#'4'
                LCALL   SBYTE
N4:             CJNE    A,#05H,N5       ; DTMF 5
                MOV     A,#'5'
                LCALL   SBYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
N5:             CJNE    A,#06H,N6       ; DTMF 6
                MOV     A,#'6'
                LCALL   SBYTE
N6:             CJNE    A,#07H,N7       ; DTMF 7
                MOV     A,#'7'
                LCALL   SBYTE
N7:             CJNE    A,#08H,N8       ; DTMF 8
                MOV     A,#'8'
                LCALL   SBYTE
N8:             CJNE    A,#09H,N9       ; DTMF 9
                MOV     A,#'9'
                LCALL   SBYTE
N9:             RET
                ;
;***************************************************************
;*****           Receive DTMF from port 1                  *****
;***************************************************************
;
RENT:           MOV     P2,#0FFH        ; Input Port
                MOV     R1,#90
CDR:            JB      STD,TOE         ; Check bit 0
                LCALL   DELAY_100ms
                DJNZ    R1,CDR
                MOV     A,#77H          ; Define
                SJMP    STOP_RENT
TOE:            MOV     R7,P1
                JB      STD,$           ; Check bit 1
                MOV     A,R7
                ANL     A,#0F0H
                SWAP    A
STOP_RENT:      RET
                ;
;***************************************************************
;*****           Send to serial port                       *****
;***************************************************************
;
SBYTE:          CLR     TI
                MOV     SBUF,A          ; Send to Serial Port
                JNB     TI,$
                CLR     TI
                LCALL   DELAY_100ms
                RET
                ;
;***************************************************************
;*****           Wait   Computer                           *****
;***************************************************************
;
RX:                                     ; Wait 0 to 8
                JNB     RI,$
                CLR     RI
                MOV     A,SBUF
                LCALL   DELAY_100ms
                ;
                CJNE    A,#'0',RE1      ; No Reward
                LCALL   NO_REWARD
                LJMP    RE9
RE1:
                CJNE    A,#'1',RE2      ; Reward 1
                LCALL   REWARD_1
                LJMP    RE9
RE2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CJNE    A,#'2',RE3      ; Reward 2
                LCALL   REWARD_2
                LJMP    RE9
RE3:
                CJNE    A,#'3',RE4      ; Reward 3
                LCALL   REWARD_3
                LJMP    RE9
RE4:
                CJNE    A,#'4',RE5      ; Reward 4
                LCALL   REWARD_4
                LJMP    RE9
RE5:
                CJNE    A,#'5',RE6      ; Reward 5
                LCALL   REWARD_5
                LJMP    RE9
RE6:
                CJNE    A,#'6',RE7      ; Reward tail 3
                LCALL   REWARD_6
                LJMP    RE9
RE7:
                CJNE    A,#'7',RE8      ; Reward tail 2
                LCALL   REWARD_7
                LJMP    RE9
RE8:
                CJNE    A,#'8',RE9
                LCALL   REWARD_8        ; Reward near 1
RE9:            RET
                ;
;*************************************************************
;*****           Sound  Reward                           *****
;*************************************************************
;
NO_REWARD:                              ; No reward
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#00H
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_1:                               ; Reward 1
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#22H
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_2:                               ; Reward 2
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#43H
                MOV     P0,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_3:                               ; Reward 3
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#64H
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_4:                               ; Reward 4
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#86H
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_5:                               ; Reward 5
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#0A7H
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_6:                               ; Reward tail 3
                CLR     PD
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#0C8H
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_7:
                CLR     PD              ; Reward tail 2
                SETB    CE
                CLR     A8
                CLR     A9
                MOV     A,#0EAH
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
REWARD_8:                               ; Reward near 1
                CLR     PD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB    CE
                CLR     A9
                SETB    A8
                MOV     A,#0BH
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_3s
                RET
INTRO:                                  ; Intro
                CLR     PD
                SETB    CE
                CLR     A9
                SETB    A8
                MOV     A,#2CH
                MOV     P0,A
                CLR     CE
                SETB    CE
                LCALL   DELAY_10s
                RET
                ;
                ;
;***************************************************************
;****            Delay time for program                     ****
;***************************************************************
;
DELAY_1ms:
                MOV     R7,#02          ; Delay 1ms
DEL_01:         MOV     R6,#00H
DEL_02:         SJMP    DEL_03
DEL_03:         SJMP    DEL_04
DEL_04:         SJMP    DEL_05
DEL_05:         SJMP    DEL_06
DEL_06:         SJMP    DEL_07
DEL_07:         DJNZ    R6,DEL_02
                DJNZ    R7,DEL_01
                RET

;***************************************************************
;
DELAY_10ms:
                MOV     R7,#10          ; Delay 10ms
DEL_11:             MOV     R6,#00H
DEL_21:             SJMP    DEL_31
DEL_31:         SJMP    DEL_41
DEL_41:         SJMP    DEL_51
DEL_51:         SJMP    DEL_61
DEL_61:         SJMP    DEL_71
DEL_71:         DJNZ    R6,DEL_21
                DJNZ    R7,DEL_11
                RET
                ;



;***************************************************************
;
DELAY_25ms:
                MOV     R7,#25          ; Delay 25ms
DEL_14:         MOV     R6,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEL_24:             SJMP    DEL_34
DEL_34:             SJMP    DEL_44
DEL_44:             SJMP    DEL_54
DEL_54:             SJMP    DEL_64
DEL_64:             SJMP    DEL_74
DEL_74:             DJNZ    R6,DEL_24
                    DJNZ    R7,DEL_14
                    RET
                    ;
;*********************************************************************
;
DELAY_50ms:
                    MOV     R7,#50          ; Delay 50ms
DEL_15:             MOV     R6,#00H
DEL_25:             SJMP    DEL_35
DEL_35:             SJMP    DEL_45
DEL_45:             SJMP    DEL_55
DEL_55:             SJMP    DEL_65
DEL_65:             SJMP    DEL_75
DEL_75:             DJNZ    R6,DEL_25
                    DJNZ    R7,DEL_15
                    RET
                    ;
;*********************************************************************
;
DELAY_100ms:
                    MOV     R7,#100         ; Delay 100ms
DEL_12:             MOV     R6,#00H
DEL_22:             SJMP    DEL_32
DEL_32:             SJMP    DEL_42
DEL_42:             SJMP    DEL_52
DEL_52:             SJMP    DEL_62
DEL_62:             SJMP    DEL_72
DEL_72:             DJNZ    R6,DEL_22
                    DJNZ    R7,DEL_12
                    RET
                    ;
;*********************************************************************
;
DELAY_250ms:
                    ACALL   DELAY_50ms      ; Delay 250ms
                    ACALL   DELAY_50ms
                    ACALL   DELAY_50ms
                    ACALL   DELAY_50ms
                    ACALL   DELAY_50ms
                    RET
                    ;
;*********************************************************************
;
DELAY_500ms:
                    ACALL   DELAY_250ms     ; Delay 500ms
                    ACALL   DELAY_250ms
                    RET
                    ;
;*********************************************************************
;
DELAY_1s:
                    ACALL   DELAY_500ms     ; Delay 1s
                    ACALL   DELAY_500ms
                    RET
                    ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************
;
DELAY_3s:
                    ACALL   DELAY_1s          ; Delay 3s
                    ACALL   DELAY_1s
                    ACALL   DELAY_1s
                    RET
;*****************************************************************
;
DELAY_5s:
                    ACALL   DELAY_1s          ; Delay 5s
                    ACALL   DELAY_1s
                    ACALL   DELAY_1s
                    ACALL   DELAY_1s
                    ACALL   DELAY_1s
                    RET
                    ;
;*****************************************************************
;
DELAY_10s:
                    ACALL   DELAY_5s          ; Delay 10s
                    ACALL   DELAY_5s
                    RET
                    ;;
                    END
                    ;
;*****************************************************************
;*****               End  of  Program                        *****
;*****************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STAND-ALONE/PARALLEL INTERFACE PRODUCTS

Single-Chip Board Layout Diagrams

EXAMPLE PC BOARD LAYOUT

The printed circuit layout example in this chapter shows an implementation of the circuit shown in Figure 2 under the section "Circuit Examples for ISD1000A and ISD2500 Products." A dip switch (SW1) for addresses and pull-up resistors on the control inputs has been added. Additionally, two new address inputs and a pulse stretcher (U2A) for the overflow output (pin 22) have been added to the circuit so that this PC board will also work with the ISD2500 series. Also, one-shot pulse generator (U2B) adds the ability to generate short duration $\overline{CE}$ pulses for Message Cueing experimentation. The schematic of this board is shown in Figure 4, Single Chip Demo Board.

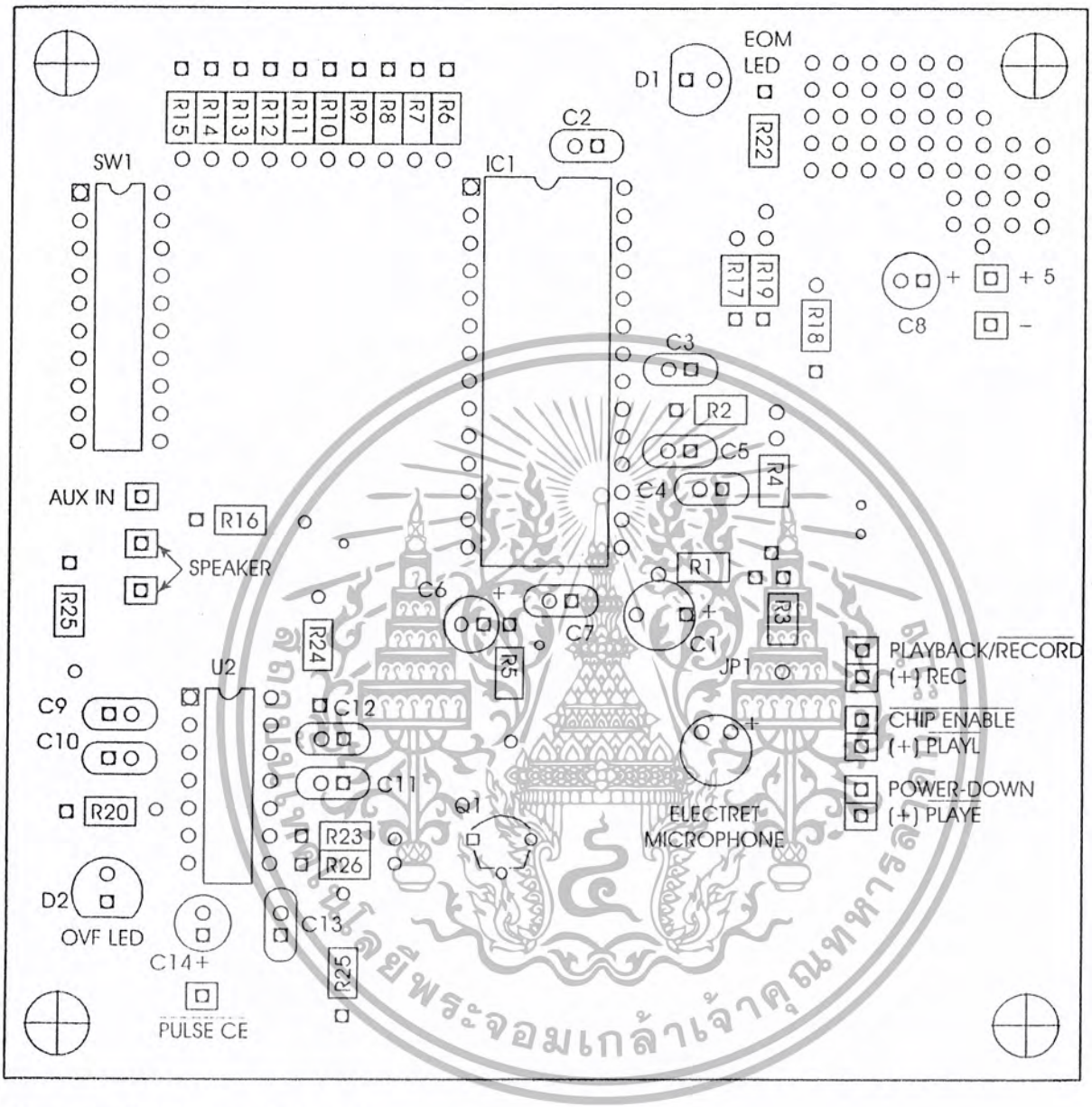
NOTE *The following PC board drawings are not to scale.*

The PC board layout incorporates the recommendations in "Good Audio Design Practices." Power feeds from the right-hand side of the board at the two points labeled "+" and "–". C8 is physically located at the branch of the V_{CC} traces that feed V_{CCD} on the top (going to pin 28) and V_{CCA} on the bottom (going to pin 16).

High-frequency ESR decoupling capacitors C2 and C7 are located adjacent to the supply pins they serve. C2, the decoupling cap for the V_{CCD} supply pin, has its own separate ground trace going directly back to the "–" input of the board. C7 picks up its ground from the major ground plane area on the "bottom" end of the board. The major power traces are greater than 30-mils wide and as direct as possible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 1: Single-Chip Demo Board Silk Screen



NOTE: *(*) = ISD2500 series devices only.*
(+) = ISD1100, ISD1200, and ISD1400 series devices only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2: Single-Chip Demo Board Component Side PC Layout

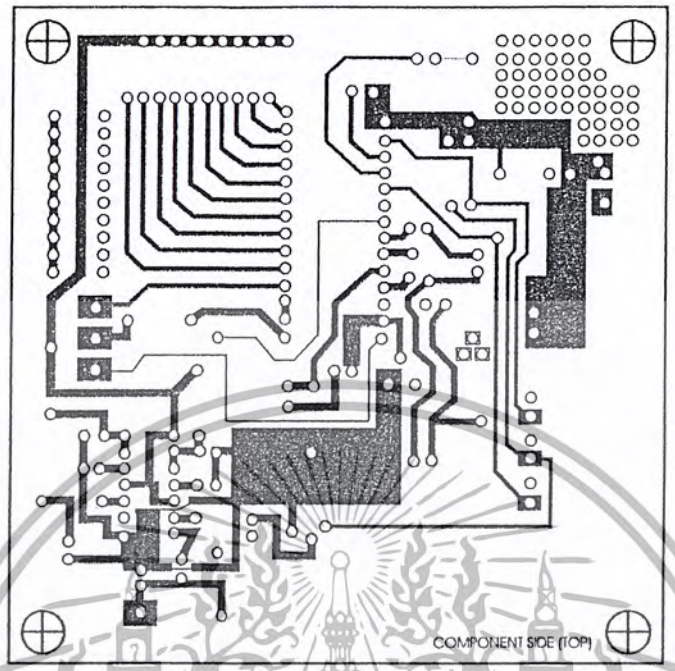


Figure not drawn to scale.

NOTE: *(*) = ISD2500 series devices only.*
(+) = ISD1100, ISD1200, and ISD1400 series devices only.
Figure not drawn to scale.

Figure 3: Single-Chip Demo Board Solder Side PC Layout

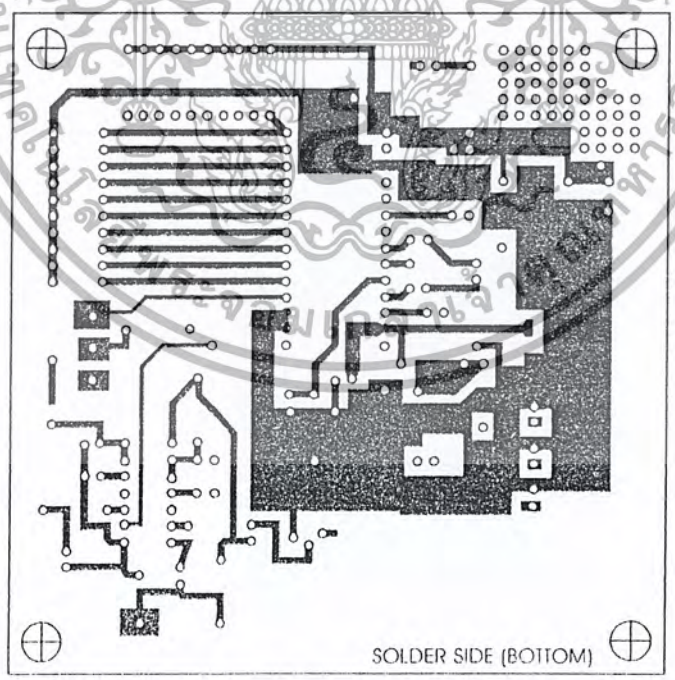


Figure not drawn to scale.

NOTE: *(*) = ISD2500 series devices only.*
(+) = ISD1100, ISD1200, and ISD1400 series devices only.
Figure not drawn to scale.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 4: ISD2500 Single-Chip Demo Board Schematic

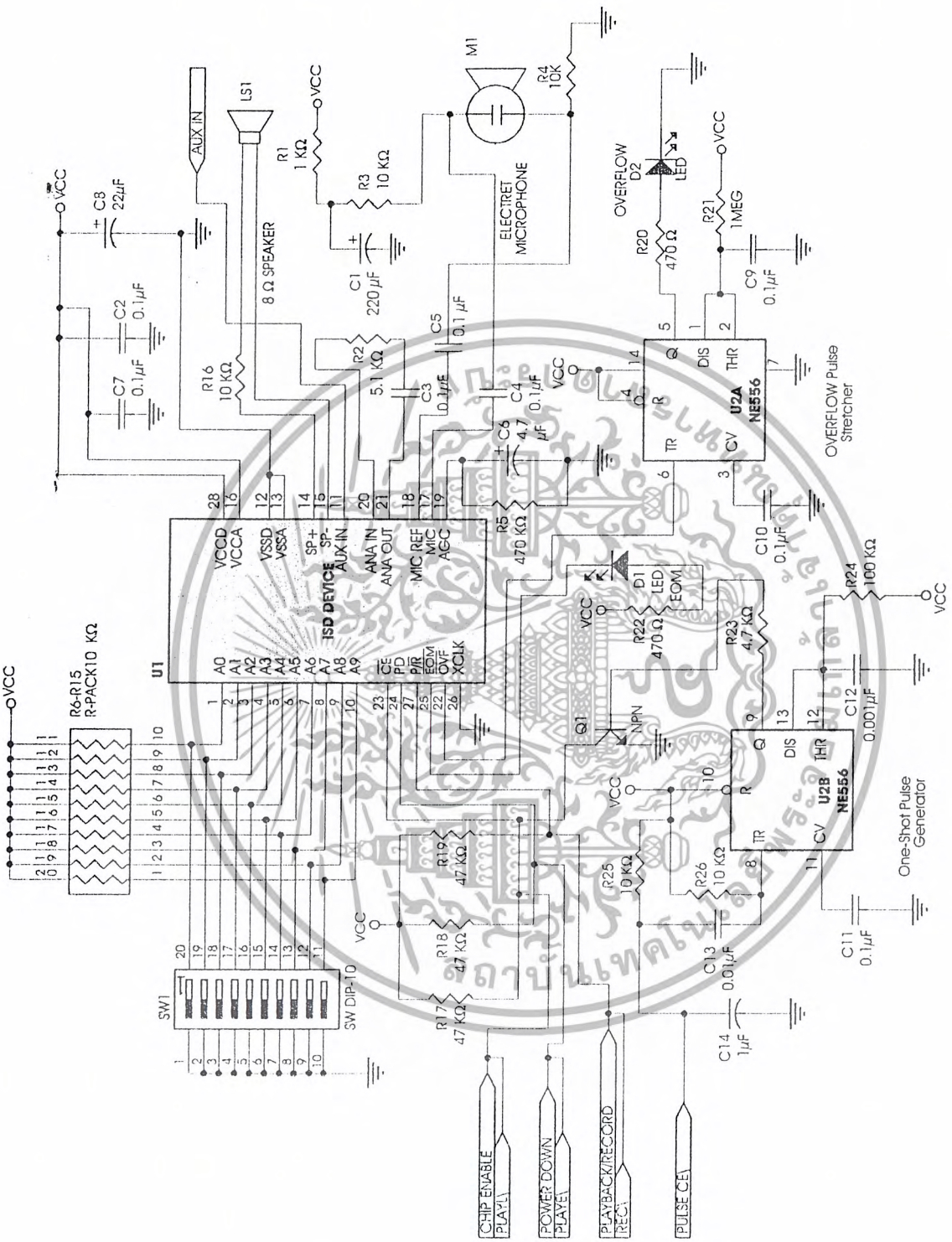


Table 5: ISD2500 Address Space

Dec.	Binary										ISD2560 (Seconds)	ISD2575 (Seconds)	ISD2590 (Seconds)	ISD25120 (Seconds)
	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	1	1	0	0	1	0	5.0	6.25	7.5	10.0
100	0	0	0	1	1	0	0	1	0	0	10.0	12.5	15.0	20.0
250	0	0	1	1	1	1	1	0	1	0	25.0	31.25	37.5	50.0
300	0	1	0	0	1	0	1	1	0	0	30.0	37.5	45.0	60.0
400	0	1	1	0	0	1	0	0	0	0	40.0	50.0	60.0	80.0
500	0	1	1	1	1	1	0	1	0	0	50.0	62.5	75.0	100.0
599	1	0	0	1	0	1	0	1	1	1	59.9	74.875	89.85	119.8
600	1	0	0	1	0	1	1	0	0	0				
through	Unused address space. An ISD2500 device addressed in this region will default to an overflow condition.													
767	1	0	1	1	1	1	1	1	1	1				
768	1	1	0	0	0	0	0	0	0	0				
through	This address space used by the ISD2500 Operational Modes.													
1023	1	1	1	1	1	1	1	1	1	1				

Many product designs use small, inexpensive microcontroller chips and do not have sufficient port pins to handle the ISD device for full addressing. This need not be a serious obstacle. Most of these same applications do not need 100 ms. resolution of the address space. They can operate with less resolution. For each degree of resolution that is not used, one less port pin is needed. The following table (a simple binary count) illustrates this principle using an ISD1016A or ISD1416 as the example.

Table 6 indicates that by grounding the four least significant address bits, A0–A3, one can still address messages with a 1.6 second resolution using only four port or latch outputs. This would provide ten different 1.6-second message segments in an ISD1016A (ten 2-second messages in an ISD1020A) and might make it possible to use a smaller, less expensive microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In the circuit in Figure 8, the output level of the ISD device must be attenuated by resistors R1 and R2 before it can be applied to the inputs of the LM386. This is because the ISD device output leads are designed to drive a speaker directly to 12.5 mW. The voltage swing drives the LM386 into distortion if not reduced. At 5 volts, the value required is approximately 1 MΩ; at 9 volts the resistors can be at 560 KΩ. This is because the input pins of the LM386 have about a 50 KΩ to ground impedance, forming a voltage divider. Using 560 KΩ resistors and putting the 100 KΩ potentiometer R3 across the input pins creates a volume control.

The LM386 data sheet includes various applications circuits, all single ended. To eliminate the "pop" that can occur with the ISD speaker outputs being used single ended, the LM386 is used differentially. This lets the common mode rejection of the LM386 reduce the "pop" considerably. Because the differential connection is DC isolated, the LM386 can be run at any voltage in its allowable range while the ISD device remains at 5 volts. This give the designer some options as far as required power output for the particular application.

In the example circuit in Figure 8, pins 1 and 8 are left open for minimum gain of 20 in the LM386N-1. Then a volume control is provided in the potentiometer (R3) across pins 2 and 3.

Figure 8: LM386 Speaker Driver

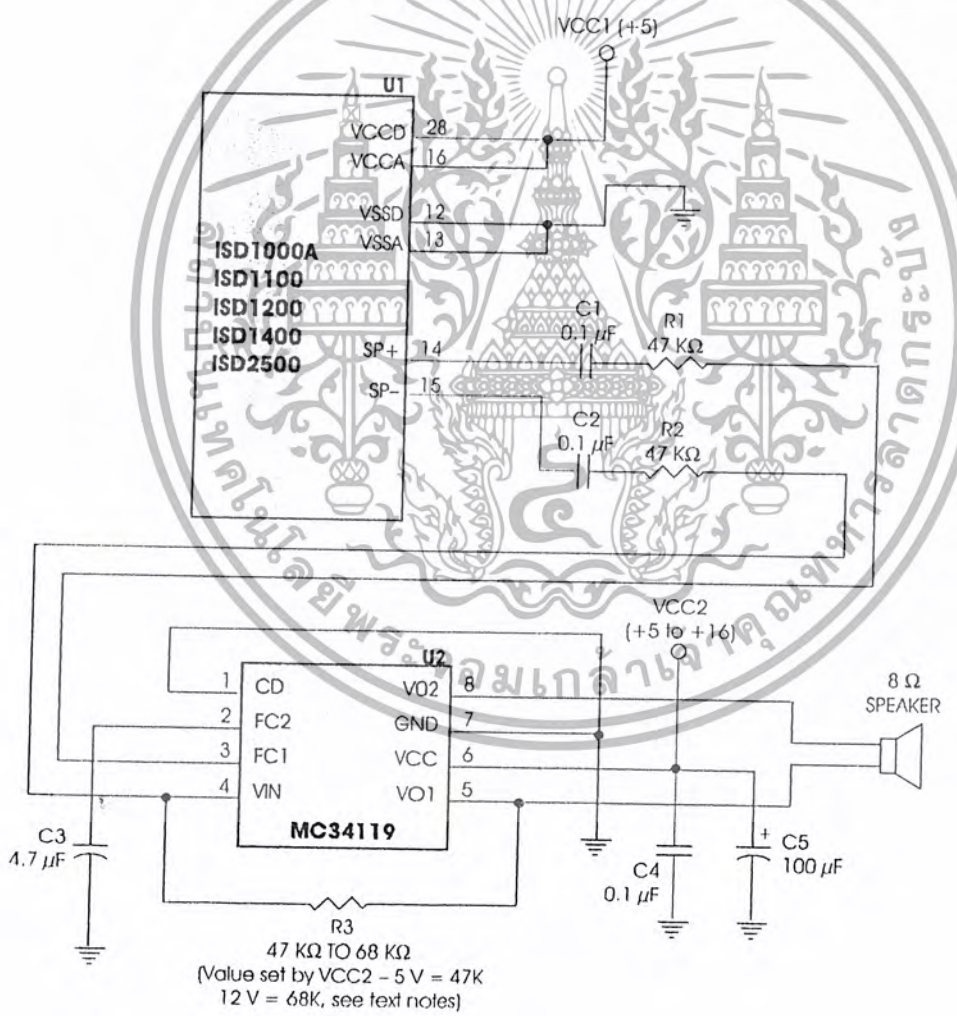
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

USING THE MOTOROLA MC34119 LOW-POWER AUDIO AMPLIFIER

The MC34119 low power audio amplifier integrated circuit, shown in Figure 9, was intended primarily for telephone applications. According to the Motorola data sheet, it operates over a voltage range from 2 to 16 volts and can supply up to 250 mW into a 32 Ω speaker. It can drive speaker loads down to 8 Ω. The MC34119 does not automatically power down. If the CD pin (pin 1) is taken to V_{CC}, however, the device will power down to typically less than 1/2 mA.

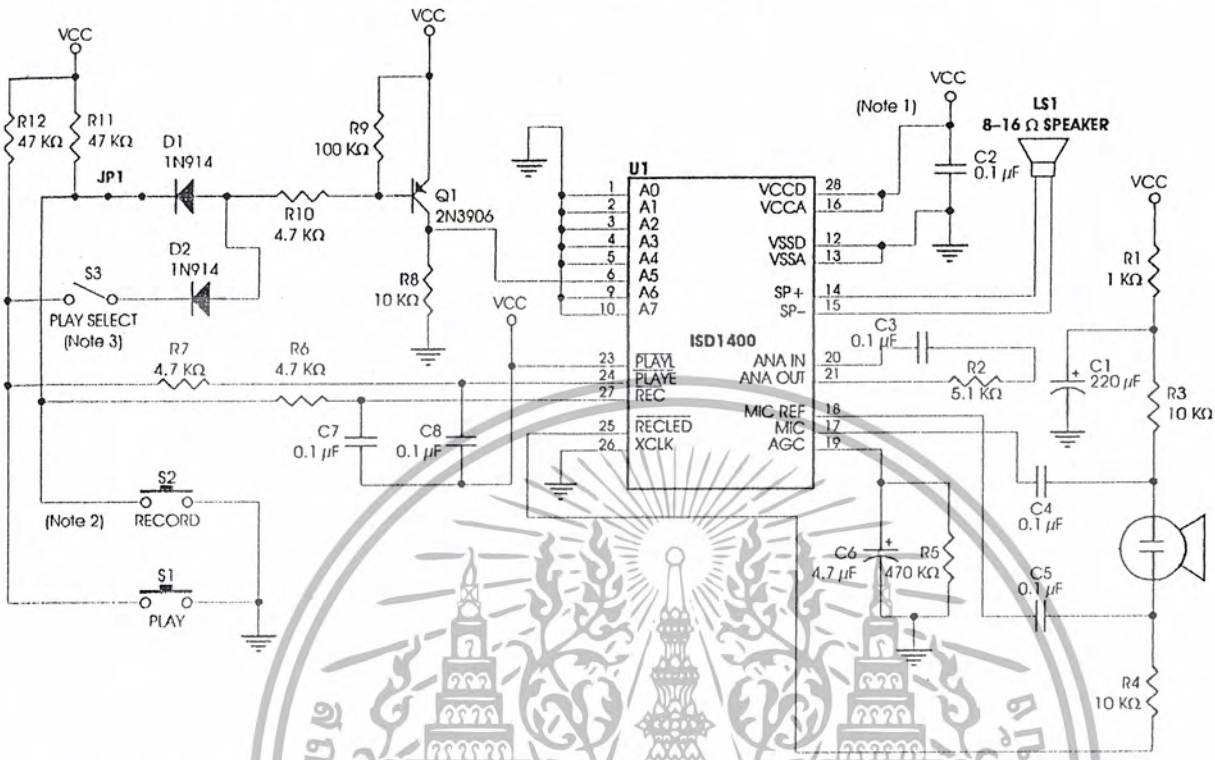
The circuit used with the MC34119 is similar to that used with the NSC device. The FC1 and VIN inputs are driven differentially from the ISD Speaker outputs. Speaker volume is set by adjusting the value of R3 which is a feedback resistor used to set gain in the MC34119. Care should be taken so that the Motorola amplifier's package power dissipation specification is not exceeded. The designer may wish to derive a control signal to drive the CD pin to lower circuit power consumption when not in use.

Figure 9: MC34119 Speaker Driver



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 12: Fixed and Re-recordable Message Segment Circuit Example



1. *V_{CC} = 5 to 6.5 volts.*
2. *Pressing S2 begins recording at 4-second boundary. Pressing S1 starts Playback at 0 or 4 seconds according to switch S3 setting.*
3. *S3 open, Playback begins at 0, S3 closed, Playback begins at 4 seconds.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้