สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครือข่าย พี เอ บี เอกซ์

PABX NETWORKING

โดย

นายสุรชัย เงินสุทธิชัยพร

นายอนุสรณ์ อังสนานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50092
วัน,เดือน,ปี 2 1 เม.ย. 2547

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่าย พี เอ บี เอกซ์

PABX NETWORKING

โดย

นายสุรชัย เงินสุทธิชัยพร 42010400

นายอนุสรณ์ อังสนานนท์ 42010433

อาจารย์ที่ปรึกษา

รศ.ดร. สุวิพล สิทธิชีวภาค

ผศ. เกรียงไกร วงศ์โรจนภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **เครือข่าย พี เอ บี เอกซ์**

PABX NETWORKING

ผู้จัดทำ

1. นายสุรชัย เงินสุทธิชัยพร 42010400

2. นายอนุสรณ์ อังสนานนท์ 42010433

..... อาจารย์ที่ปรึกษา

(รศ.ดร. สุวิพล สิทธิชีวภาค)

..... อาจารย์ที่ปรึกษา

(ผศ. เกรียงไกร วงศ์โรจนภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่าย พี เอ บี เอกซ์

PABX NETWORKING

โดย นาย สุรชัย เงินสุทธิชัยพร 42010400
นาย อนุสรณ์ อังสนานนท์ 42010433

อาจารย์ที่ปรึกษา รศ.ดร. สุวิพล สิทธิชีวภาค
ผศ. เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

โครงงานนี้เป็นการจำลองระบบไอพี เทเลโฟน มาไว้บนเครือข่ายคอมพิวเตอร์โดยมีการเชื่อมต่อผ่าน ไอพี เกตเวย์ ระหว่างเครือข่ายขององค์การโทรศัพท์ กับเครือข่ายคอมพิวเตอร์ซึ่งสามารถรองรับการใช้งานได้หลายคู่สายในกรณีที่มีผู้ใช้พร้อมกันมากกว่าหนึ่งคู่สาย ซึ่งโครงงานนี้แบ่งออกเป็น 2 ส่วนคือ ฮาร์ดแวร์และซอฟท์แวร์ ส่วนของฮาร์ดแวร์ จะใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงานและเชื่อมต่อระหว่างโทรศัพท์ กับคอมพิวเตอร์ ในส่วนของ ซอฟท์แวร์ใช้ในการจัดการเกี่ยวกับการส่งผ่านและรับข้อมูลเสียงผ่านเครือข่ายคอมพิวเตอร์

ABSTRACT

This project is to simulate the IP Telephone system on computer network, by connecting via IP gateway between Public Switch Telephone Network (PSTN) and computer network that support the telephone lines. It can use more than one line at the same time. This project consists of 2 important parts. First part is hardware and the other is software. For hardware part, we use microcontroller to control and interface between computer and telephone. For software part, it is used to control a voice data correctly flow in the computer network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ

1.1 ความเป็นมาของโครงงาน 1

1.2 หน้าที่ภาคต่างๆ ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 วัตถุประสงค์ของโครงงาน 3

บทที่ 2 ทฤษฎีและหลักการ

2.1 ความรู้เบื้องต้นเกี่ยวกับโทรศัพท์ 4

2.1.1 การทำงานของเครื่องโทรศัพท์ 4

2.1.2 ระบบโทรศัพท์แบบส่งสัญญาณความถี่คู่ 6

2.1.3 สัญญาณในการติดต่อกันระหว่างผู้เรียกและผู้รับโทรศัพท์ 8

2.2 ระบบเน็ตเวิร์คคอมพิวเตอร์ 9

2.2.1 OSI โมเดล 9

2.2.2 การติดต่อสื่อสารระหว่างเวิร์กสเตชัน 12

2.2.3 รูปแบบการเชื่อมต่อของระบบ LAN 12

2.2.4 โพรโตคอลมาตรฐานของระบบเครือข่าย 14

2.3 โปรโตคอลในการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ผ่านอินเตอร์เน็ต 18

2.3.1 ความหมายของ TCP/IP 18

2.3.2 ชุดโปรโตคอลทีซีพี/ไอพี 19

2.3.3 แบบอ้างอิง TCP/IP 21

2.3.4 TCP/IP Client - Server 22

2.3.5 ข้อเปรียบเทียบและความสัมพันธ์ระหว่างโปรโตคอล TCP และ UDP 22

2.3.6 พอร์ตของ TCP/IP 23

2.3.7 ซ็อคเกตของ TCP/IP 23

2.3.8 หมายเลข IP Address 24

2.3.8.1 การจัดคลาสเครือข่าย 24

2.3.8.2 ลักษณะสำคัญของแต่ละคลาส 25

2.4 ไมโครคอนโทรลเลอร์ MCS-51 26

2.4.1 กล่าวนำ 26

2.4.2 โครงสร้างของ MCS-51 27

2.4.3 การจัดการหน่วยความจำของ MCS-51 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.4.4 สถาปัตยกรรมของ MCS-51 30

2.4.5 การทำงานของ MCS-51 35

2.4.6 การอินเตอร์รัปต์ 36

2.5 ภาคแปลงรหัสสัญญาณความถี่คู่ 39

2.5.1 บทนำ 39

2.5.2 ภาคกรองสัญญาณความถี่ 40

2.5.3 ภาคถอดรหัส 41

2.5.4 ภาคตรวจสอบสัญญาณ 41

2.5.5 ภาคขยายสัญญาณความแตกต่าง 41

2.5.6 ขั้นตอนการทำงาน 43

2.6 วงจรกำเนิดสัญญาณคู่ความถี่ 43

2.7 การติดต่อสื่อสารผ่านพอร์ตอนุกรม 44

2.7.1 นัลโมเด็ม 46

2.7.2 UART 47

2.7.3 รีจีสเตอร์ของพอร์ตอนุกรม 47

2.7.4 DLAB 48

2.7.5 อุปกรณ์ที่ใช้ในการติดต่อสื่อสารผ่าน RS-232 53

2.8 การเชื่อมโยงทางแสง 55

2.8.1 ทรานซิสเตอร์คัพเปลอร์ 56

2.8.2 ดาร์ลิงตันทรานซิสเตอร์คัพเปลอร์ 57

2.8.3 ออปโต้คัพเปลอร์ที่ใช้สวิทช์สองทิศทางหรือไตรแอค (Triac) 57

2.8.4 การประยุกต์ใช้งานโดยการนำไปใช้ควบคุมโหลด 57

บทที่ 3 การคำนวณและการสร้าง

3.1 วงจรตรวจจับสัญญาณกระดิ่ง 59

3.2 วงจรตรวจสอบสัญญาณความถี่เสียง 60

3.3 วงจรถอดรหัสสัญญาณความถี่คู่ 61

3.4 วงจรกำเนิดสัญญาณคู่ความถี่ 63

3.5 วงจรควบคุมเสียงพูดแบบ 2 ทิศทาง 64

3.5.1 วงจรเชื่อมต่อกับไฟตรง 65

3.5.2 ตัวจ่ายแรงดันคงที่ 66

3.5.3 วงจรขยายสัญญาณจากไมโครโฟน 66

3.5.4 วงจรในการส่งสัญญาณ 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.5.5 วงจรในการรับสัญญาณ 68

3.5.6 วงจรขจัดไซด์โทน 69

3.5.7 การเชื่อมต่อของสัญญาณลอจิก 70

3.6 การออกแบบโปรแกรมที่ตัวไมโครคอนโทรลเลอร์ MCS-51 71

3.6.1 การออกแบบโปรแกรมที่ตัวไมโครคอนโทรลเลอร์ MCS-51 เพื่อติดต่อกับไคลเอ็นท์ 71

3.6.2 การออกแบบโปรแกรมตรวจสอบการติดต่อจากเซิร์ฟเวอร์เพื่อโทรออก 71

3.7 การออกแบบโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอ็นท์หรือเซิร์ฟเวอร์ 74

3.7.1 การออกแบบโปรแกรมไคลเอ็นท์ 74

3.7.2 การออกแบบโปรแกรมเซิร์ฟเวอร์ 74

บทที่ 4 การทดลองและผลการทดลอง

4.1 การทดลองวงจรตรวจสอบสัญญาณกระดิ่ง 80

4.2 การทดลองวงจรตรวจสอบสัญญาณความถี่เสียง 82

4.3 การทดลองวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ 84

4.4 การทดลองวงจรกำเนิดสัญญาณดีทีเอ็มเอฟ 85

4.5 ภาคแสดงผลในส่วนการทดลองโปรแกรม 92

4.5.1 หน้าโปรแกรมขณะเตรียมพร้อมใช้งาน 92

4.5.2 หน้าโปรแกรมการตั้งค่าต่างๆ 92

4.5.3 หน้าโปรแกรมขณะที่ทำหน้าที่เป็นไคลเอ็นท์ 93

4.5.4 หน้าโปรแกรมขณะที่ทำหน้าที่เป็นเซิร์ฟเวอร์ 94

4.6 การทดสอบวัดคุณภาพของสัญญาณเสียงขณะที่สนทนา 95

บทที่ 5 บทวิจารณ์และบทสรุป

5.1 ส่วนสรุปผลของโครงงาน 98

5.2 ส่วนแก้ไขและพัฒนา 98

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 1.1	แสดงการจำลองระบบเครือข่าย พี เอ บี เอกซ์	1
รูปที่ 2.1	แสดงบล็อกไดอะแกรมของเครื่องโทรศัพท์	5
รูปที่ 2.2	แสดงแป้นกดหมายเลขและค่าความถี่ต่างๆ	7
รูปที่ 2.3	แสดงสัญญาณพื้นฐานของโทรศัพท์	7
รูปที่ 2.4	แสดงสัญญาณระหว่างผู้เรียกและผู้รับโทรศัพท์	8
รูปที่ 2.5	แสดงการเชื่อมต่อแบบ Bus	12
รูปที่ 2.6	แสดงการเชื่อมต่อแบบ Star	13
รูปที่ 2.7	แสดงการเชื่อมต่อแบบ Ring	13
รูปที่ 2.8	แสดงประเภทของอินเตอร์เน็ตแบบ 10 Base T	15
รูปที่ 2.9	แสดงประเภทของอินเตอร์เน็ตแบบ 10Base 2	15
รูปที่ 2.10	แสดงประเภทของอินเตอร์เน็ตแบบ 10Base 5	16
รูปที่ 2.11	แสดงหลักการทำงานของโทเคนริง	17
รูปที่ 2.12	แสดงโทโปโลยีเอฟดีดีไอ	17
รูปที่ 2.13	แสดงเอฟทีพีไคลเอ็นต์ – เซิร์ฟเวอร์	22
รูปที่ 2.14	แสดงรูปแบบของไอพีแอดเดรส	24
รูปที่ 2.15	แสดงการแบ่งคลาสเครือข่าย	25
รูปที่ 2.16	แสดงบล็อกไดอะแกรมแสดงโครงสร้างของ MCS-51	27
รูปที่ 2.17	แสดงสถาปัตยกรรมภายในของ MCS-51	30
รูปที่ 2.18	แสดงตำแหน่งขาของ MCS-51	31
รูปที่ 2.19	แสดงสัญญาณต่างๆ ของ MSC-51 เบอร์ 8051	31
รูปที่ 2.20	แสดงการต่อสัญญาณนาฬิกาที่ขา XTAL1 และ XTAL2	32
รูปที่ 2.21	แสดงการใช้สัญญาณนาฬิกาจากภายนอก	32
รูปที่ 2.22	แสดงการต่อสัญญาณรีเซต	33
รูปที่ 2.23	แสดงการสร้างสัญญาณแอดเดรสและสัญญาณข้อมูล	34
รูปที่ 2.24	แสดงลำดับสถานะการทำงานใน MCS-51	36
รูปที่ 2.25	แสดงวงจรตรวจสัญญาณอย่างง่ายและแสดงการกำหนดเวลาการ์ดไทม์	39
รูปที่ 2.26	แสดงโครงสร้างภายในของ MT8870	40
รูปที่ 2.27	แสดงภาคกรองสัญญาณความถี่	41
รูปที่ 2.28	แสดงการต่อวงจรภาคอินพุต	42
รูปที่ 2.29	แสดงการวงจรถอดรหัสความถี่ DTMF	42
รูปที่ 2.30	แสดงวงจรกำเนิดสัญญาณคู่ความถี่	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 2.31 (ก) คอนเนคเตอร์แบบ DB-25 (ข) คอนเนคเตอร์แบบ DB-9 45

รูปที่ 2.32 แสดงการเชื่อมต่อสายของนัลโมเด็ม 47

รูปที่ 2.33 แสดงรูปคลื่นของการส่งสัญญาณแบบอนุกรมด้วยระดับแรงดันของ TTI/CMOS 53

รูปที่ 2.34 แสดงคลื่นของสัญญาณที่รับ-ส่งผ่านพอร์ท RS-232 54

รูปที่ 2.35 แสดงลักษณะขาและวงจรภายในของชิพ RS – 232 54

รูปที่ 2.36 แสดงสัญลักษณ์ของออปโต้คัพเปลอร์ 55

รูปที่ 2.37 (ก) แสดงถึงแสงที่เกิดขึ้นเนื่องจากกระแสไบอัสตรงไหลผ่าน (ข) อิเล็กตรอนส่วนเกินข้ามรอยต่อไปรวมกับโฮลพร้อมกับเปล่งแสงออกมา 56

รูปที่ 2.38 แสดงวงจรที่ใช้ควบคุมกำลังงานไฟฟ้ากระแสสลับ 58

รูปที่ 2.39 แสดงวงจรที่ใช้ควบคุมกำลังไฟฟ้ากระแสสลับที่มีค่าสูง 58

รูปที่ 3.1 แสดงบล็อกไดอะแกรมของฮาร์ดแวร์ 59

รูปที่ 3.2 แสดงวงจรตรวจจับสัญญาณโทรศัพท์ 60

รูปที่ 3.3 แสดงวงจรตรวจสอบสัญญาณความถี่เสียง 61

รูปที่ 3.4 แสดงวงจรถอดรหัสสัญญาณความถี่คู่ 63

รูปที่ 3.5 แสดงวงจรกำเนิดสัญญาณคู่ความถี่ 63

รูปที่ 3.6 แสดงบล็อกไดอะแกรมภายใน MC34114 65

รูปที่ 3.7 แสดงวงจรสมมูลย์ของการอินเตอร์เฟสกับคู่สายโทรศัพท์ 65

รูปที่ 3.8 แสดงบล็อกไดอะแกรมและอุปกรณ์ภายนอก MC34114 67

รูปที่ 3.9 แสดงเส้นทางของสัญญาณทางด้านรับและด้านส่ง 68

รูปที่ 3.10 แสดงผังการทำงานของโปรแกรมตรวจสอบการโทรศัพท์เข้าเพื่อติดต่อกับไคลเอ็นท์ 72

รูปที่ 3.11 แสดงผังการทำงานของโปรแกรมตรวจสอบการติดต่อจากเซิร์ฟเวอร์เพื่อโทรออก 73

รูปที่ 3.12 แสดงผังการทำงานของโปรแกรมไคลเอ็นท์ 75

รูปที่ 3.13 แสดงผังการทำงานของโปรแกรมเซิร์ฟเวอร์ 76

รูปที่ 3.14 แสดงรูปวงจรรวมทั้งหมด 77

รูปที่ 3.15 แสดงภาพวงจรที่ลงอุปกรณ์เรียบร้อยแล้ว 78

รูปที่ 3.16 แสดงภาพรวมของวงจรทั้งหมดที่ลงอุปกรณ์เรียบร้อยแล้ว 78

รูปที่ 3.17 แสดงภาพด้านหน้าของอุปกรณ์ที่พร้อมใช้งานได้ 79

รูปที่ 4.1 แสดงวงจรตรวจจับสัญญาณกระดิ่ง 80

รูปที่ 4.2 แสดงสัญญาณกระดิ่งที่ป้อนเข้ามาที่สายสัญญาณโทรศัพท์ 81

รูปที่ 4.3 แสดงสัญญาณเอาท์พุท (ขา 5) ของ 4N26 เมื่อมีอินพุทเป็นสัญญาณกระดิ่ง 81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.4 แสดงการเปรียบเทียบสัญญาณเอาท์พุท ของ timer 555 (channel 1) กับ สัญญาณเอาท์พุท ของ 4N26 (channel 2) เมื่อมีอินพุทเป็นสัญญาณกระดิ่ง 82

รูปที่ 4.5 แสดงวงจรตรวจจับสัญญาณโทน 82

รูปที่ 4.6 แสดงสัญญาณเอาท์พุท (ขา 8) ของ LM 567 เมื่อมีอินพุทเป็น สัญญาณ Dial Tone 83

รูปที่ 4.7 แสดงสัญญาณเอาท์พุท (ขา 8) ของ LM 567 เมื่อมีอินพุทเป็น สัญญาณ Busy Tone 83

รูปที่ 4.8 แสดงวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ 84

รูปที่ 4.9 แสดงวงจรกำเนิดสัญญาณดีทีเอ็มเอฟ 85

รูปที่ 4.10 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 1 86

รูปที่ 4.11 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 2 86

รูปที่ 4.12 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 3 87

รูปที่ 4.13 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 4 87

รูปที่ 4.14 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 5 88

รูปที่ 4.15 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 6 88

รูปที่ 4.16 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 7 89

รูปที่ 4.17 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 8 89

รูปที่ 4.18 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 9 90

รูปที่ 4.19 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 0 90

รูปที่ 4.20 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข * 91

รูปที่ 4.21 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข # 91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.22 แสดงรูปหน้าโปรแกรมขณะเตรียมพร้อมใช้งาน 92

รูปที่ 4.23 แสดงรูปหน้าการตั้งค่าต่างๆ ให้กับโปรแกรม Visual Basic 93

รูปที่ 4.24 แสดงหน้าต่างที่พร้อมจะรับข้อมูลจาก Microcontroller 93

รูปที่ 4.25 แสดงรูปหน้าโปรแกรมขณะที่ทำหน้าที่เป็นไคลเอ็นท์ 94

รูปที่ 4.26 แสดงรูปหน้าโปรแกรมขณะที่ทำหน้าที่เป็นเซิร์ฟเวอร์ 95

รูปที่ 4.27 แสดงการเชื่อมต่อเครือข่ายเพื่อวัดสัญญาณเสียง 95

รูปที่ 4.28 แสดงสัญญาณอินพุตที่ป้อนให้กับระบบ (channel1) เทียบกับสัญญาณ เอาท์พุตของระบบ (channel2) โดยป้อนสัญญาณซายน์ที่ความถี่ 1 kHz 96

รูปที่ 4.29 แสดงสัญญาณอินพุตที่ป้อนให้กับระบบ (channel1) เทียบกับสัญญาณ เอาท์พุตของระบบ (channel2) โดยป้อนสัญญาณซายน์ที่ความถี่ 2 kHz 97

รูปที่ 4.30 แสดงสัญญาณอินพุตที่ป้อนให้กับระบบ (channel1) เทียบกับสัญญาณ เอาท์พุตของระบบ (channel2) โดยป้อนสัญญาณซายน์ที่ความถี่ 3 kHz 97

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model 10

ตารางที่ 2.2 แสดงสัญญาณต่างๆ ของพอร์ต P3 35

ตารางที่ 2.3 แสดงลำดับความสำคัญของการทำอินเตอร์รัปต์ 37

ตารางที่ 2.4 แสดงตำแหน่งของโปรแกรมอินเตอร์รัปต์แต่ละชนิด 37

ตารางที่ 2.5 แสดงค่าภายในรีจีสเตอร์หลังจากการรีเซต 38

ตารางที่ 2.6 แสดงสัญญาณ DTMF ที่ได้จาก ไอซีเบอร์ 5088 เมื่อป้อนที่ขา D0,D1,D2 และ D3 ด้วยค่าต่างๆ 44

ตารางที่ 2.7 แสดงตำแหน่งขาสัญญาณของพอร์ทอนุกรม 46

ตารางที่ 2.8 แสดงหน้าที่ของขาสัญญาณแต่ละเส้น 46

ตารางที่ 2.9 แสดงตำแหน่งมาตรฐานของพอร์ตอนุกรม 48

ตารางที่ 2.10 แสดงรีจีสเตอร์ของพอร์ตอนุกรม 48

ตารางที่ 2.11 แสดงอัตราบอดและค่าใช้ในการหาร 49

ตารางที่ 2.12 แสดง Interrupt Enable Register 49

ตารางที่ 2.13 แสดง FIFO Control Register 50

ตารางที่ 2.14 แสดง Modem Control Register 50

ตารางที่ 2.15 แสดง Modem Status Register 51

ตารางที่ 2.16 แสดง Interrupt Identification Register 51

ตารางที่ 2.17 แสดง Line Control Register 52

ตารางที่ 2.18 แสดง Line Status Register 53

ตารางที่ 3.1 แสดงการถอดรหัสสัญญาณความถี่คู่ 62

ตารางที่ 3.2 แสดงสัญญาณ DTMF ที่ได้จากไอซีเบอร์TP5088 เมื่อป้อนที่ขา D0,D1,D2 และ D3 ด้วยค่าต่างๆ 64

ตารางที่ 3.3 แสดงความสัมพันธ์ระหว่างโหมดการทำงานกับลอจิกของ MUTE และ MS 70

ตารางที่ 4.1 แสดงผลการทดลองที่ได้จากวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ 84

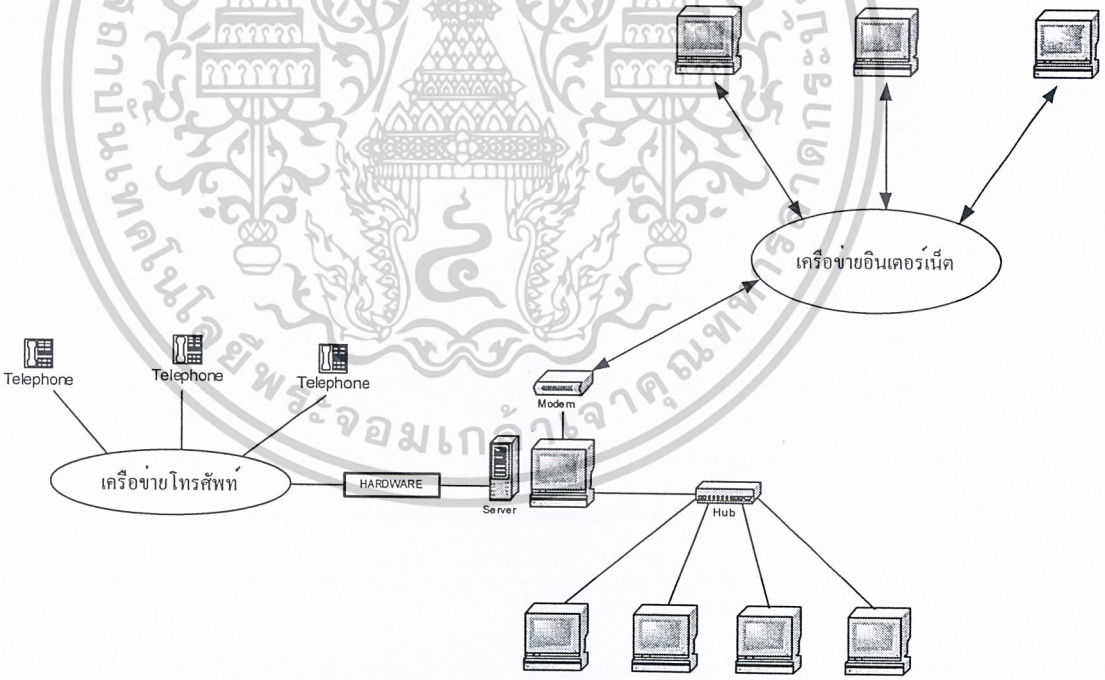
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ในปัจจุบันเครือข่ายคอมพิวเตอร์ มีการพัฒนาอย่างรวดเร็ว มีประสิทธิภาพสูงในการประมวลผล อีกทั้งระบบการสื่อสารสามารถส่งข้อมูลทั้งภาพ และเสียงได้ ทำให้มีการพัฒนาโปรแกรมประยุกต์ขึ้นมาเพื่อรองรับการใช้งาน เครือข่ายคอมพิวเตอร์นอกจากจะช่วยในการทำงานของผู้ใช้คอมพิวเตอร์ให้เป็นไปอย่างมีประสิทธิภาพแล้ว ยังสามารถนำมาประยุกต์ให้ผู้ใช้คอมพิวเตอร์ในเครือข่ายสามารถที่จะพูดคุยกับเครื่องโทรศัพท์ปลายทางได้โดยไม่จำเป็นที่จะต้องมีหมายเลขโทรศัพท์ และเครื่องโทรศัพท์ แต่จะส่งเสียงพูดผ่านทางไมโครโฟน สำหรับหน่วยงานขนาดกลางและเล็กจะทำให้ประหยัดค่าใช้จ่ายได้ ด้วยเหตุนี้จึงเกิดโครงงานนี้ขึ้นมา

สำหรับโครงงานนี้ ระบบ ไอพี เทเลโฟน (IP TELEPHONE) เป็นเทคโนโลยีที่ผนวกเครือข่ายคอมพิวเตอร์เข้ากับเครือข่ายโทรศัพท์พื้นฐาน (PSTN) ทำให้สามารถติดต่อสื่อสารข้อมูลกันในลักษณะของสัญญาณเสียงระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ และระหว่างเครื่องคอมพิวเตอร์กับเครื่องโทรศัพท์



รูปที่ 1.1 แสดงการจำลองระบบเครือข่าย พี เอ บี เอกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 หน้าที่ภาคต่างๆ ของโครงงาน

โครงงานนี้ การออกแบบโครงสร้างและระบบการทำงานสามารถอธิบายหน้าที่ภาคต่างๆ ของโครงงานนี้ ดังต่อไปนี้

- ภาคสถานะของโทรศัพท์ (Line Status) แบ่งได้เป็น ดังนี้
- ภาคควบคุมสาย (Line Control)

ทำหน้าที่รับสัญญาณจากภาคควบคุมระบบมาทำการควบคุมการทำงานของรีเลย์ (Relay) เพื่อเป็นสวิทซ์ตัดต่อระหว่างภาคตรวจจับสัญญาณกระดิ่งโทรศัพท์ (Check Ringing) กับภาคอื่นๆ ของโครงงาน

- ภาคตรวจจับสัญญาณกระดิ่งโทรศัพท์ (Check Ringing)

ทำหน้าที่ตรวจจับสัญญาณกระดิ่งโทรศัพท์ เมื่อมีผู้ใช้บริการเรียกเข้ามายังหมายเลขโทรศัพท์ของโครงงานที่ได้กำหนดไว้แล้วส่งสัญญาณให้ภาคแสดงสถานะสาย (Line Status) ไปให้ไมโครคอนโทรลเลอร์รับทราบว่าจะมีการเริ่มต้นทำงานของระบบ

- ภาคตรวจจับสัญญาณความถี่เสียง (Check Tone)

ทำหน้าที่คอยตรวจจับสัญญาณโทรศัพท์เมื่อมีการวางหูแล้ว ส่งให้ภาคแสดงสถานะสาย (Line Status) ไปแจ้งให้ไมโครคอนโทรลเลอร์รับทราบว่ามีการวางสายแล้วเพื่อต้องการให้รีเลย์ (Relay) กลับสู่สภาวะปกติ

- ภาคถอดรหัสสัญญาณดีทีเอ็มเอฟ (DTMF Decoder)

ทำหน้าที่ถอดรหัสสัญญาณ DTMF ที่รับมาจากคู่สายโทรศัพท์ที่ผู้ใช้บริการกดหมายเลขโทรศัพท์เข้ามาแล้วแปลงเป็นรหัส BCD (Binary Code Decimal) ส่งให้ภาคควบคุมระบบ (MCS-51) รับทราบรหัสคำสั่ง

- ภาคกำเนิดสัญญาณดีทีเอ็มเอฟ (Dial DTMF)

ทำหน้าที่กำเนิดสัญญาณคู่ความถี่ที่สามารถนำมาประยุกต์ใช้งานในโทรศัพท์ที่ถูกควบคุมด้วยไมโครคอนโทรลเลอร์

- ภาคควบคุมระบบ (Control System)

ทำหน้าที่ควบคุมสภาวะการทำงาน การจัดการสัญญาณอินพุต-เอาท์พุตของระบบทั้งหมด โดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยส่งสัญญาณทางพอร์ทอนุกรมไปยังภาคควบคุมในการทำหน้าที่เป็นไคลเอ็นท์ หรือ เซิร์ฟเวอร์

1.3 ขอบเขตของโครงงาน

โครงงานนี้ การออกแบบโครงสร้างสามารถแบ่งได้เป็น 2 ส่วนใหญ่ๆ คือ

1. การออกแบบโครงสร้างทางฮาร์ดแวร์ (Hardware)

ในการจัดทำโครงงานนี้ ได้ดำเนินการออกแบบวงจร และทดลองผลการทำงานของวงจรภาคต่างๆ ดังนี้คือ ภาคควบคุมสาย (Line Control), ภาคตรวจจับสัญญาณกระดิ่งโทรศัพท์ (Check Ringing), ภาคตรวจจับสัญญาณความถี่เสียง (Check Tone), ภาคถอดรหัสสัญญาณดีทีเอ็มเอฟ (DTMF Decoder), และภาคกำเนิดสัญญาณดีทีเอ็มเอฟ (Dial DTMF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การออกแบบโครงสร้างทางซอร์ฟแวร์ (Software)

จะดำเนินการเขียนโปรแกรมด้วยภาษาแอสเซมบลีสำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 เพื่อทำหน้าที่ควบคุมระบบการทำงานทั้งหมดและการเขียนโปรแกรมด้วยภาษา Visual Basic Version 6.0 สำหรับไมโครคอมพิวเตอร์ เพื่อควบคุมในการทำหน้าที่เป็นไคลเอ็นท์ หรือ เซิร์ฟเวอร์

1.4 วัตถุประสงค์ของโครงงาน

วัตถุประสงค์ของโครงงานนี้คือเพื่อศึกษาหาความรู้และประสบการณ์ทางด้านการออกแบบวงจรควบคุมสัญญาณจากวงจรต่างๆ ให้สามารถทำงานร่วมกันได้ ทฤษฎีและขั้นตอนการปฏิบัติงานการทำงานของไมโครคอนโทรลเลอร์การเขียนโปรแกรมด้วยภาษาแอสเซมบลีสำหรับควบคุมไมโครคอนโทรลเลอร์ตระกูล MCS-51 และการเขียนโปรแกรมด้วยภาษา Visual Basic Version 6.0 สำหรับไมโครคอมพิวเตอร์ และสามารถนำโครงงานนี้ไปประยุกต์ใช้งานได้จริง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ความรู้เบื้องต้นเกี่ยวกับโทรศัพท์

เครื่องรับโทรศัพท์เป็นอุปกรณ์เกี่ยวกับการโทรคมนาคมที่ใช้ในการติดต่อสื่อสารได้สะดวกรวดเร็ว และให้ข่าวสารที่ชัดเจนฉับไว ค่าใช้จ่ายถูก จึงเป็นที่นิยมกันอย่างมาก โทรศัพท์ที่เห็นกันอยู่ทั่วๆไป มีอยู่ 2 แบบ คือ แบบหมุน และแบบสัญญาณความถี่คู่ (DTMF) หรือเรียกอีกอย่างหนึ่งว่า ระบบกดปุ่ม ซึ่งระบบโทรศัพท์แบบหมุนเป็นระบบแบบดั้งเดิม ซึ่งในปัจจุบันกำลังจะเลิกใช้งานแล้ว โดยระบบกดปุ่มได้เข้ามาได้รับความนิยมแทน และใช้งานกันมากหน้าที่ของระบบโทรศัพท์ ทั้งสองระบบจะมีลักษณะเหมือนๆ กัน จะต่างกันตรงที่ระบบกดปุ่มจะส่งสัญญาณออกไปเป็นความถี่ที่แตกต่างกันสองความถี่ ส่วนระบบกดปุ่มจะส่งสัญญาณออกไปเป็นความถี่ที่แตกต่างกันสองความถี่ ส่วนระบบกดปุ่มจะส่งสัญญาณเป็นจำนวนพัลส์ หน้าที่หลักๆ ของโทรศัพท์ มีดังนี้

1. เครื่องโทรศัพท์จะทำให้ชุมสายรู้ว่ามีผู้ต้องการใช้โทรศัพท์ เมื่อมีการยกหูโทรศัพท์ขึ้น
2. เครื่องโทรศัพท์จะได้รับสัญญาณหมุน (Dial Tone) ที่ส่งมาจากชุมสายเพื่อบอกให้ผู้ใช้รู้ว่าพร้อมที่จะทำการกดหรือหมุนหมายเลขที่ต้องการติดต่อซึ่งก็คือเสียงที่ได้ยินเมื่อยกหูโทรศัพท์เป็นสัญญาณที่มีความถี่ 400 – 425 Hz ดังต่อเนื่องกันไป
3. เครื่องโทรศัพท์ทำหน้าที่ส่งรหัสหมายเลขที่ผู้เรียก ต้องการจะติดต่อด้วยไปยังชุมสายโทรศัพท์ด้วยการกดปุ่มหมายเลขหรือหมุนหมายเลขที่เราต้องการจะติดต่อ
4. เครื่องโทรศัพท์จะส่งสัญญาณบอกผู้เรียกว่าหมายเลขที่ต้องการจะติดต่อด้วยว่าว่างหรือไม่ ถ้าว่างก็จะส่งสัญญาณกลับ (Ring Back Tone) ที่มีความถี่ 425 Hz โดยจะดัง 1 วินาที และเงียบ 4 วินาที สลับกันไป แต่ถ้าหมายเลขที่ต้องการเรียกไปนั้นไม่ว่างก็จะส่งสัญญาณความถี่ 425 Hz โดยจะดังเป็นช่วงๆ 0.5 วินาที และหยุด 0.5 วินาที
5. เครื่องรับโทรศัพท์ทางด้านส่งจะเปลี่ยนสัญญาณเสียงเป็นสัญญาณไฟฟ้า และทางด้านรับจะเปลี่ยนสัญญาณไฟฟ้ากลับมาเป็นสัญญาณเสียงอีกครั้งหนึ่ง
6. เครื่องรับโทรศัพท์จะส่งสัญญาณเสียงเรียกเมื่อมีผู้เรียกเข้ามายังเครื่องโทรศัพท์ การส่งสัญญาณเสียงเรียกจะเป็นสัญญาณกระดิ่ง หรือสัญญาณลักษณะใดก็ได้ ขึ้นอยู่กับเครื่องโทรศัพท์นั้นๆ โดยทั่วไปแล้ว Ring current มีค่าประมาณ 70 – 90 โวลต์ ความถี่ 12 – 25 Hz
7. เครื่องโทรศัพท์จะส่งสัญญาณไปยังชุมสาย เมื่อเราวางหูโทรศัพท์เพื่อแจ้งให้ทราบว่าสิ้นสุดการใช้งานแล้ว และให้ชุมสายเลิกการติดต่อ

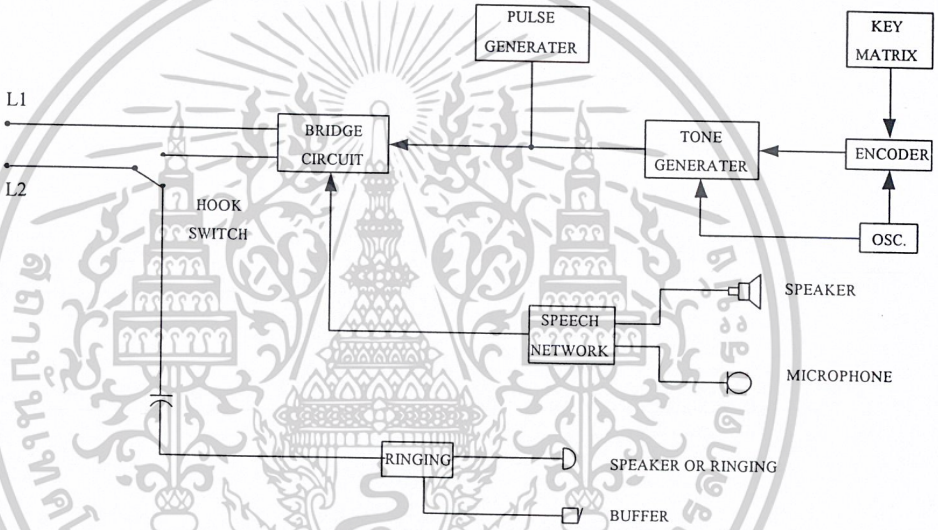
2.1.1 การทำงานของเครื่องโทรศัพท์

ในรูปที่ 2.1 เป็นบล็อกไดอะแกรมของส่วนต่างๆ ที่จำเป็นในโทรศัพท์ โดยการทำงานของเครื่องรับโทรศัพท์อธิบาย ได้ดังนี้

เครื่องโทรศัพท์จะเชื่อมต่อกับชุมสายโทรศัพท์ด้วยสาย L1 และสาย L2 หรือ Local Loop ซึ่งก็คือสายส่ง 2 Wire ลวดตัวนำ 2 เส้นในเส้นลูปที่มีชื่อว่า Tip และRing โดยRingจะต่ออยู่กับสัญญาณไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

48 Vdc และ Tip จะต่อกับกราวนด์ เมื่อมีการยกหูโทรศัพท์ระดับแรงดันระหว่าง Tip – Ring จะมีค่าลดลงเหลือประมาณ 9 โวลต์ วงจรแรกที่เชื่อมต่อระหว่างวงจรในเครื่องโทรศัพท์กับอุปกรณ์ชุมสาย คือ วงจรกำเนิดเสียงเรียก (Bell or Ring) ซึ่งจะส่งสัญญาณเรียกเมื่อมีการติดต่อมาจากผู้อื่น เหตุผลประการสำคัญที่ต้องนำวงจรนี้มาเชื่อมต่อกับอุปกรณ์ชุมสายโดยตรง คือ เมื่อวางหูโทรศัพท์ไว้กับที่วางตามปรกติ Hook Switch จะเปิดวงจรออกทำให้ไม่มีแรงดันจากชุมสาย ผ่านไปยังวงจรส่วนที่อยู่หลัง Hook Switch ดังนั้นถ้าวงจรกำเนิดเสียงเรียกอยู่หลัง Hook Switch ก็จะไม่สามารถสร้างสัญญาณเสียงเรียกได้ในขณะที่มีผู้ติดต่อเข้ามา วงจรเสียงเรียกจึงต่ออยู่กับชุมสายโทรศัพท์โดยตรง เมื่อเราวางหูโทรศัพท์ลงบน Hook Switch เมื่อมีสัญญาณเรียกเข้ามาจากชุมสาย สัญญาณเรียกจะผ่าน Hook Switch เข้าไปยังวงจรกำเนิดเสียงเรียก ทำให้วงจรทำงาน ส่วนที่เป็นลำโพงหรือบัสเซอร์ก็จะดังขึ้น เมื่อเรายกหูโทรศัพท์ก็จะทำให้ Hook Switch ตัดวงจรเสียงเรียกออก และ Hook switch ก็จะต่อกับวงจรเสียงพูดแทน (Speech Network)



รูปที่ 2.1 แสดงบล็อกไดอะแกรมของเครื่องโทรศัพท์

Hook Switch

On – Hook

โดยทั่วไปแล้ว Ringer จะเชื่อมต่อ Tip (+) และ Ring (-) ทำให้สามารถรับสัญญาณ Incoming Call หรือสัญญาณสั่นกระดิ่งซึ่งส่งมาจากชุมสาย ส่วนวงจรที่เหลือ เช่น Dialing , Hybrid, Tx, Rx จะแยกออกจากสายโทรศัพท์ในกรณีโทรศัพท์อยู่ในสถานะวางหู (On Hook) เมื่อ Handset อยู่ในสถานะวางหูจะไม่มีกระแสไหลผ่านเครื่องโทรศัพท์ วงจร Ring จะมี Capacitor ป้องกันแระแสไฟตรงไหลผ่าน

Off – Hook

เมื่อ Handset ถูกยกจะมีสภาวะเป็น Off–Hook ทำให้เกิดกระแสลูป (Loop Current) ไหลแบตเตอรี่ของชุมสาย ผ่านเครื่องโทรศัพท์ และกลับมายังชุมสาย เมื่อมีกระแสไหลผ่านเพียงพอที่จะทำให้ชุมสายทราบได้ว่ามีการยกหูขึ้นที่ทางผู้ใช้ และหลังจากนั้นจะส่งสัญญาณ Dial Tone ไปให้ผู้เรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรกำเนิดเสียงพูดจะทำหน้าที่เปลี่ยนสัญญาณเสียงพูดให้เป็นสัญญาณไฟฟ้า เพื่อส่งไปยังด้านรับฝ่ายตรงกันข้าม และจะทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าที่รับเข้ามาจากฝ่ายตรงข้าม ให้เป็นสัญญาณเสียงพูดวงจร ส่วนนี้จะประกอบไปด้านวงจรที่จะใช้ควบคุมเสียงพูดที่จะย้อนกลับไปยังหูฟังของผู้พูด ให้มีความแรงของสัญญาณพอดี ในขณะที่เราพูดเข้าทางไมโครโฟนเราก็จะได้ยินเสียงของเราเองทางหูฟังด้วย เพื่อให้เราทราบว่าเสียงที่เราพูดนั้นแรงหรือค่อยเท่าไร วงจรในส่วนควบคุมนี้จะเป็นการควบคุมเสียงของเราเองไม่ให้ออกทางหูฟังแรงมากเกินไป เพราะถ้าดังเกินไปจะทำให้รำคาญ และยังกลบเสียงพูดของคู่สนทนาด้วย ขณะเดียวกัน ก็ควบคุมไม่ให้เสียงที่ย้อนกลับมาที่หูฟังเบาเกินไป เพราะจะทำให้ผู้พูดคิดว่าตัวเองพูดโทรศัพท์ค่อย ก็จะทำให้ผู้พูดใช้เสียงดังมากขึ้น ทำให้คู่สนทนาได้ยินเสียงดังเกินไป

วงจรกำเนิดสัญญาณพัลส์ จะทำหน้าที่กำเนิดสัญญาณพัลส์ เพื่อส่งเลขหมายที่เรากดไปให้กับชุมสายโทรศัพท์ที่เป็นระบบพัลส์

วงจรกำเนิดความถี่ จะทำหน้าที่กำเนิดความถี่คู่ เพื่อทำหน้าที่ส่งหมายเลขไปให้ชุมสายโทรศัพท์แบบความถี่คู่หรือที่เรียกว่า DTMF

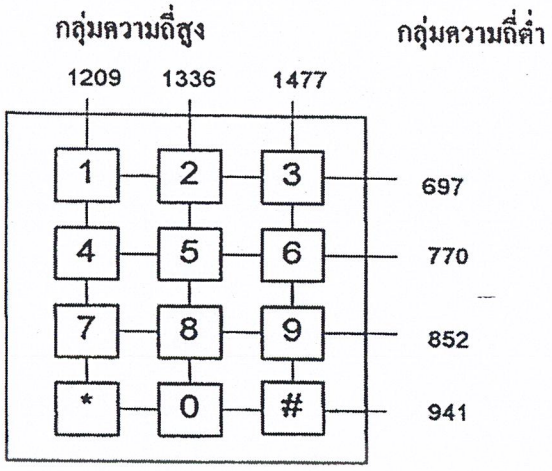
วงจร Hybrid จะทำหน้าที่เป็นตัวเชื่อมต่อคู่สายโทรศัพท์เข้ากับวงจรในส่วนอื่นๆ จะทำหน้าที่เป็นวงจรปรับความสมดุลของอิมพีแดนซ์ คือ ปรับอิมพีแดนซ์ของเครื่องรับโทรศัพท์ ให้สมดุลกับคู่สายโทรศัพท์ ปกติจะมีค่าอิมพีแดนซ์ประมาณ 600 โอห์ม หรืออาจเรียกวงจรนี้ว่า วงจรแปลงกลับไปมาระหว่าง 2 Wire กับ 4 Wire

เมื่อมีการยกหูโทรศัพท์ขึ้น Hook Switch จะปิดวงจร ทำให้มีกระแสไหลจากชุมสายครบวงจรผ่านเครื่องโทรศัพท์ ในขณะเดียวกันกระแสค่าเดียวกันนี้จะไหลผ่านวงจรเชื่อมต่อสายโทรศัพท์ที่ชุมสายด้วยเพื่อที่จะทำให้อุปกรณ์ต่างๆ ในชุมสายพร้อมจะทำการติดต่อกับเครื่องโทรศัพท์ได้ จากนั้นชุมสายก็จะทำการส่งสัญญาณหมุนไปยังผู้ที่ยกหูโทรศัพท์ เพื่อให้ส่งหมายเลขโทรศัพท์ที่ต้องการติดต่อมา

2.1.2 ระบบโทรศัพท์แบบส่งสัญญาณความถี่คู่ (Dual Tone Multi – Frequency type)

เป็นระบบการส่งสัญญาณที่นิยมใช้ในปัจจุบัน ระบบนี้มีชื่อย่อว่า DTMF มีวิธีการส่งหมายเลขโทรศัพท์ของผู้ที่ต้องการติดต่อไปให้กับชุมสายโทรศัพท์ โดยส่งสัญญาณไปด้วยความถี่ 2 ความถี่ที่มอดูเลตกันไปซึ่งจะเป็นตัวแทนของหมายเลขที่กด ความถี่ที่ถูกส่งออกไปจะอยู่ที่ย่านความถี่ของเสียงพูด (0–4 kHz) ซึ่งความถี่ที่ต่อกว่าเป็นความถี่ที่แสดงในแนวนอน และอีกค่าหนึ่งก็จะเป็นความถี่ในแนวตั้ง ซึ่งค่าต่างๆ แสดงได้ในรูปที่ 2.2

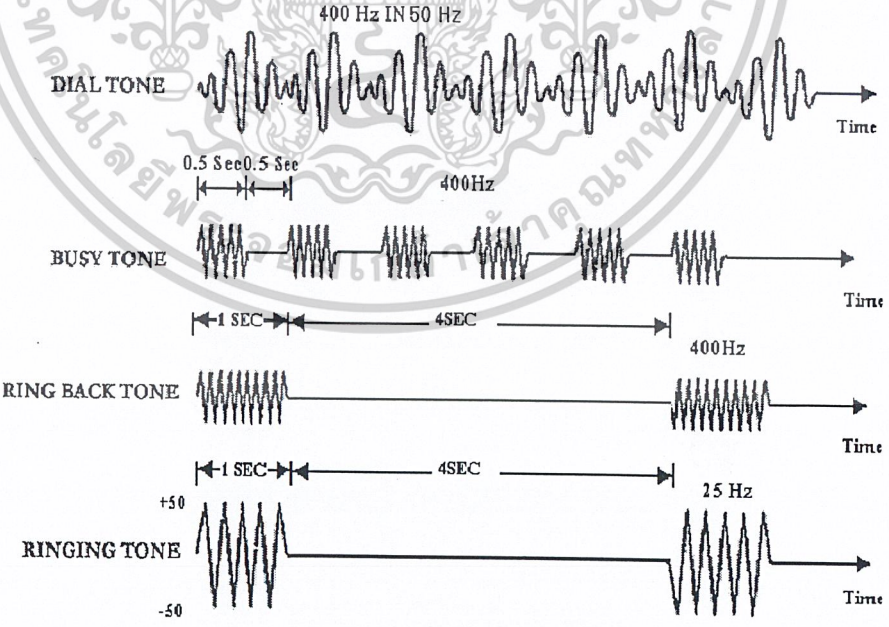
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงแป้นกดหมายเลขและค่าความถี่ต่างๆ

ข้อดีของการส่งสัญญาณแบบ DTMF

1. ลดระยะเวลาในการส่งหมายเลขโทรศัพท์ไปยังชุมสาย
2. สามารถใช้วงจรที่ประกอบด้วยอุปกรณ์โซลิดสเตทได้ ทำให้เกิดความประหยัดและสะดวก
3. ลดอุปกรณ์ประเภทหน่วยความจำภายในชุมสายโทรศัพท์
4. การส่งเลขหมายให้กับชุมสายโทรศัพท์ไม่เกิดความผิดพลาดเพราะใช้ 2 ความถี่ในการส่งทำให้ไม่ผิดพลาดในการกดหมายเลข



รูปที่ 2.3 แสดงสัญญาณพื้นฐานของโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 สัญญาณในการติดต่อกันระหว่างผู้เรียกและผู้รับโทรศัพท์

สัญญาณในการติดต่อกันระหว่างผู้เรียกและผู้รับโทรศัพท์ ลักษณะของสัญญาณดังกล่าวแสดงไว้ในรูป 2.4

ด้านผู้เรียก

1. ขณะที่ไม่ได้มีการยกหูโทรศัพท์ จะมีแรงดันตกคร่อมสายโทรศัพท์เป็นสัญญาณไฟตรง 48 โวลต์

2. เมื่อผู้เรียกยกหูโทรศัพท์ แรงดันจะลดลงเหลือ 8 โวลต์ พร้อมกับจะมีสัญญาณให้หมุน ซึ่งเป็นสัญญาณไฟสลับขนาด 250 mV ความถี่ 425 Hz รวมกับความถี่ 50 Hz ซึ่งเป็นการกดรหัสสัญญาณความถี่แล้วสัญญาณให้หมุนนี้ก็จะหายไป

3. กดรหัส (Code) เบอร์โทรศัพท์ทั้งหมด 7 หลัก รหัสความถี่ที่ส่งจะเป็นสัญญาณผสม 2 ความถี่ เป็นความถี่สูงและต่ำผสมกัน แต่ละหมายเลขจะมี DTMF 1 คู่

4. ขณะที่รอรับสายจะมีสัญญาณตอบกลับสองแบบ เพื่อจะบอกว่าสายว่างหรือไม่ ซึ่งก็คือสัญญาณเรียกกลับ หรือสัญญาณสายไม่ว่าง ตามลำดับ

5. เมื่อมีการรับสายแล้ว สัญญาณจะอยู่ที่ 8 โวลต์ โดยมีการกระเพื่อมตามลักษณะความถี่เสียงความดังของเสียงพูดตามสาย

6. เมื่อวางหูโทรศัพท์เลิกการติดต่อ ขนาดแรงดันจะกลับไปอยู่ที่ 48 โวลต์ดังเดิม

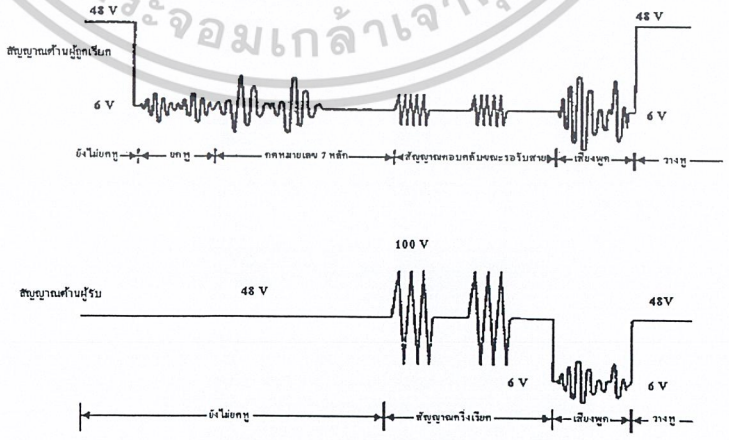
ด้านผู้รับ

1. ขณะที่วางหูจะมีแรงดันไฟตรงตกคร่อมอยู่ 48 โวลต์

2. เมื่อมีสัญญาณกริ่งเรียก จะมีขนาดประมาณ 100 โวลต์ จังหวะ 1 วินาที หยุด 4 วินาที ซึ่งตรงกับสัญญาณเรียกกลับที่เครื่องส่ง

3. จากนั้นเมื่อผู้รับยกหูโทรศัพท์ ขนาดแรงดันไฟตรงจะเหลือ 8 โวลต์ และมีการกระเพื่อมตามขนาดและความถี่ของสัญญาณเสียงพูด

4. เมื่อวางหูโทรศัพท์ขนาดแรงดันไฟฟ้าก็จะกลับไปเป็น 48 โวลต์ดังเดิม



รูปที่ 2.4 แสดงสัญญาณระหว่างผู้เรียกและผู้รับโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ระบบเน็ตเวิร์คคอมพิวเตอร์

การใช้งานเครื่องไมโครคอมพิวเตอร์ในปัจจุบันนี้ มีอยู่ 2 แบบ คือ Stand Alone และระบบ LAN (Local Area Network) ซึ่งใช้งานในแบบ Stand Alone เพียงเครื่องเดียว เหมาะสำหรับนักเรียนหรือนักธุรกิจเล็กๆไม่ใหญ่มาก แต่ถ้าเป็นบริษัทที่มีธุรกิจในระดับกลางขึ้นไปดู จะไม่เหมาะสมซึ่งการต้องการใช้ทรัพยากรร่วมกันปรับปรุงข้อมูลให้ใหม่ และถูกต้องเสมอตลอดจนระบบรักษาความปลอดภัยของข้อมูล เน็ตเวิร์คคงจะเป็นทางเลือกที่ดีที่สุด

ระบบเน็ตเวิร์ค หมายถึง การนำเครื่องไมโครคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไปมาเชื่อมต่อกัน เพื่อทำการแชร์ข้อมูลและใช้ทรัพยากรร่วมกัน เช่น ไฟล์ข้อมูล และเครื่องพิมพ์ ระบบเน็ตเวิร์คสามารถแบ่งออกเป็น 3 ประเภทด้วยกัน คือ

1. LAN (Local Area Network)

ระบบเครือข่ายระดับท้องถิ่น เป็นเน็ตเวิร์คในระยะไม่เกิน 10 กิโลเมตร ไม่ต้องใช้โครงข่ายการสื่อสารขององค์การโทรศัพท์ คือ จะเป็นระบบเน็ตเวิร์คที่อยู่ภายในอาคารเดียวกัน หรือต่างอาคารกันในระยะใกล้ๆ

2. MAN (Metropolitan Area Network)

ระบบเครือข่ายระดับเมืองเป็นเน็ตเวิร์คที่จะต้องใช้การสื่อสารขององค์การโทรศัพท์หรือการสื่อสารแห่งประเทศไทย เพราะเป็นการติดต่อกันในเมือง เช่น มีเครื่องเวิร์กสเตชันอยู่ที่บางนา มีการติดต่อสื่อสารกับเครื่องเวิร์กสเตชันที่ดอนเมือง

3. WAN (Wide Area Network)

ระบบเครือข่ายระดับกว้างไกลหรือเรียกได้ว่าเป็น World Wide ของระบบเน็ตเวิร์ค โดยจะเป็นการติดต่อสื่อสารกัน ในระดับประเทศข้ามทวีปจะต้องใช้มีเดีย (Media) ในการสื่อสารขององค์การโทรศัพท์ หรือการสื่อสารแห่งประเทศไทย (คู่สายโทรศัพท์ dial-up line / คู่สายเช่า leased line / ISDN (Integrated Service Digital Network) สามารถส่งได้ทั้งข้อมูล เสียง และภาพในเวลาเดียวกัน

<u>ประโยชน์ของเน็ตเวิร์ค</u>

ระบบเน็ตเวิร์คเป็นระบบที่มีการใช้ทรัพยากร และใช้ข้อมูลร่วมกัน สามารถแบ่งประโยชน์ออกเป็น 5 ข้อ คือ

1. การใช้โปรแกรม – ข้อมูลร่วมกัน (Shared Applications)
2. การใช้ฮาร์ดแวร์ร่วมกัน (Shared Hardware)
3. การกระจายการประมวลผล (Distributed Processing)
4. การติดต่อสื่อสารแบบรวดเร็ว (Rapid Communication)
5. การติดต่อสื่อสารระหว่างเวิร์กสเตชัน (Inter Connected Station)

2.2.1 OSI โมเดล

องค์การมาตรฐานสากล ISO (International Organization for Standardization) ได้กำหนดมาตรฐานของเครือข่าย โดยจัดแบ่งกิจกรรมของเครือข่ายออกเป็นงานย่อยๆ และกำหนดโมเดลแบ่งเป็นขั้นๆ ตามลำดับเรียกว่า มาตรฐาน OSI (Open System Interconnection) โดยที่จะแบ่งกิจกรรมที่ซับซ้อนใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายออกเป็นงานย่อยๆ ก็จะช่วยในการออกแบบ และการใช้งานเครือข่ายรวมถึงการเชื่อมโยงกันเป็นไปได้ด้วยความสะดวก และมีวิธีการทำงานอยู่ในกรอบเดียวกัน ดังแสดงในตารางที่ 2.1

ระดับชั้น	ชื่อระดับชั้น	ระดับหน้าที่
7	Application	Application-Oriented Layers
6	Presentation	
5	Session	
4	Transport	
3	Network	Network-Dependent Layers
2	Data Link	
1	Physical	

ตารางที่ 2.1 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model

ในแต่ละขั้นของ OSI model จะมีการติดต่อสื่อสารกันเป็นขั้นๆ ตามลำดับลงมา เช่น Application Layer ตามลำดับไปจนถึงขั้นแรกสุด คือ Physical Layer

Application Layer เป็นระดับชั้นบนสุดของโมเดลเป็นส่วนที่จะทำให้การติดต่อระหว่างเครือข่ายกับผู้ใช้เป็นไปได้ตามต้องการ ตัวอย่างแอปพลิเคชั่นของเครือข่าย เช่น ระบบ E-Mail การโอนถ่ายข้อมูล (File Transfer) การขอเข้าใช้ระบบคอมพิวเตอร์ในเครือข่าย เป็นต้น

Presentation Layer เป็นระดับชั้นที่คอยรวบรวมข้อความและแปลรหัสหรือรูปแบบข้อมูลที่ได้รับมาจากระดับชั้นที่ 7 เพื่อทำให้เป็นรูปแบบการสื่อสารแบบเดียวกันเพื่อช่วยลดปัญหาต่างๆ ที่เกิดขึ้นกับผู้ใช้งานในระบบ

Session Layer เป็นระดับชั้นที่จัดการในเรื่องการติดต่อแต่ละครั้ง หรือ session ให้ระบบคอมพิวเตอร์ทั้งสองฝั่ง โดยทำหน้าที่ตั้งแต่เริ่มการติดต่อ ดูแลในการส่งผ่านข้อมูล ในการติดต่อครั้งนั้นๆ เป็นไปได้โดยไม่มีปัญหาจนถึงเลิกการติดต่อเมื่อเสร็จงาน

Transport Layer ทำหน้าที่ควบคุมปริมาณ และรายละเอียดวิธีการรับส่งข้อมูล ให้เป็นไปตามกำหนดที่ตั้งไว้ และจัดการให้การเชื่อมโยงเครือข่ายเป็นไปด้วยความราบรื่น Transport Layer จะเป็นชั้นสุดท้ายที่จัดการเรื่องเส้นทางในการส่งข้อมูล และจัดการตรวจสอบความผิดพลาดของข้อมูล ซึ่งส่วนของ TCP (Transmission Control Protocol) ในโพรโตคอล TCP/IP ทำงานที่ระดับนี้

Network Layer ทำหน้าที่ควบคุมวิธีการส่งผ่านข้อมูลระหว่างเครือข่ายให้ถูกต้อง และเป็นไปตามเส้นทางที่กำหนด โดยจะจัดการส่งผ่าน packet ข้อมูลผ่านอุปกรณ์ต่างๆ ไปยังเครือข่ายย่อยได้อย่างถูกต้องตามที่ต้องการ นอกจากนี้ยังจัดการดูแลเส้นทางในการส่งข้อมูล (Routing table) และกั้นหรือกรอง packet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ส่งไปยังเครือข่ายเดียวกัน ไม่ให้ข้ามไปยังเครือข่ายอื่น ซึ่งจะช่วยลดปริมาณข้อมูลที่จะวิ่งบนเครือข่ายได้ส่วนหนึ่ง โพรโตคอล IP, TCP/IP และ IPX เป็นโพรโตคอลที่ทำงานอยู่ใน layer นี้

Data Link Layer ทำหน้าที่เรียกใช้หรือกำหนดช่องทางในการส่งข้อมูลที่ถูกต้อง เช่น Ethernet, Token ring หรือ FDDI เป็นต้น รวมถึงการลำดับและอัตราการรับส่งข้อมูลหรือ flow control และสถานที่ที่จะส่งข้อมูลไป (address) ทั้งนี้ Data link Layer จะเป็นชั้นแรกที่จัดการแปลงข้อมูลจาก bit ให้เป็น packet โดยจะมีการเพิ่มข้อมูลเพื่อตรวจสอบความถูกต้อง ในกรณีที่ส่งข้อมูลออกไป หรือในกรณีที่อ่านข้อมูลที่เข้ามาก็จะตรวจสอบผ่าน checksum เพื่อดูว่าข้อมูลที่ได้รับมาถูกต้องครบถ้วน และถ้าได้รับ packet ข้อมูลที่ไม่ถูกต้องก็จะไม่เอาข้อมูลนั้นไปใช้งาน และจะบอกให้ต้นทางส่งข้อมูลเดิมมาใหม่

Physical Layer รับผิดชอบดูแลในรายละเอียดในการส่งข้อมูลในด้าน hardware เช่น การควบคุม Network Interface Card การส่งสัญญาณผ่านสายสัญญาณแบบต่าง ๆ การเชื่อมต่อเข้ากับเครือข่ายแบบต่างๆ โดยใช้ Physical Layer จะจัดสร้างสัญญาณทางไฟฟ้า, สัญญาณเสียง หรือสัญญาณที่จำเป็นในการสื่อสารโดยตรงเนื่องจาก Network Layer เป็นชั้นที่โพรโตคอล IP, TCP/IP ทำงานอยู่จะกล่าวโดยละเอียดต่อไป

การใช้โปรแกรมข้อมูลร่วมกัน สามารถจะทำให้ User หลายๆ คนใช้โปรแกรมข้อมูลร่วมกันได้ และเป็นการประหยัดเวลาในการติดตั้งโปรแกรมใช้งานไว้ที่ไฟล์เซิร์ฟเวอร์และเก็บข้อมูลไว้ที่ไฟล์เซิร์ฟเวอร์เพียงตัวเดียว เครื่องเวิร์กสเตชันไม่จำเป็นต้องมีดิสก์ไดร์ฟ หรือฮาร์ดดิสก์ ก็สามารถใช้งานโปรแกรม และข้อมูลจากเครื่องไฟล์เซิร์ฟเวอร์ได้

การใช้ฮาร์ดแวร์ร่วมกัน ระบบเวิร์กเป็นระบบที่สามารถ จะใช้อุปกรณ์จากฮาร์ดแวร์ร่วมกันได้ดี เช่น ระบบเน็ตเวิร์คของท่านมีเครื่องเวิร์กสเตชันอยู่ 12 เครื่อง แต่ละเครื่องมีงานที่ต้องพิมพ์บนเครื่องพิมพ์เลเซอร์ ถ้าไม่มีระบบเน็ตเวิร์คจะต้องใช้งบประมาณในการซื้อเครื่องพิมพ์อย่างน้อย 4–6 เครื่อง แต่สำหรับเน็ตเวิร์คแล้ว สามารถใช้เครื่องพิมพ์เพียง 1–2 เครื่องโดยการทำเป็น Print Server คอยรับงานพิมพ์จากเครื่องทั้ง 12 เครื่อง นอกจากนี้ยังใช้งานโมเด็มและฮาร์ดดิสก์ร่วมกันได้ด้วย

การกระจายการประมวลผล เป็นที่ทราบแล้วว่าโปรแกรมใช้งานต่างๆ เช่น Excel, Word, Data Base ฯลฯ ได้ถูกติดตั้งในฮาร์ดดิสก์ของเครื่องไฟล์เซิร์ฟเวอร์ ส่วนเครื่องเวิร์กสเตชัน จะมีเพียงจอภาพ ซีพียู และคีย์บอร์ด สำหรับทำงานเมื่อ User ของเวิร์กสเตชันเครื่องหนึ่งต้องการ ใช้งานโปรแกรม Word เครื่องไฟล์เซิร์ฟเวอร์จะทำงาน Copy โปรแกรม Word จากฮาร์ดดิสก์ของไฟล์เซิร์ฟเวอร์มาไว้ยังหน่วยความจำ Ram ของเวิร์กสเตชัน เครื่องนั้นต่อจากนี้เป็นหน้าที่ของ ซีพียู ภายในเครื่องเวิร์กสเตชันทำการ Ram โปรแกรม Word ต่อไปเป็นการกระจายการประมวลผลอย่างมีประสิทธิภาพ

การติดต่อสื่อสารแบบรวดเร็ว ด้วยมาตรฐาน IEEE 802.3 สามารถส่งข้อมูลด้วยอัตราความเร็ว 10 เมกะบิตต่อวินาที (โดยใช้หลักการส่งข้อมูลแบบ Base–band) ความเร็วในระดับนี้จะทำให้การปรับปรุง (Update) ข้อมูลเป็นไปอย่างรวดเร็วโดยเฉพาะการนำอิเล็กทรอนิกส์เมล์ (E–mail) มาใช้บนระบบเน็ตเวิร์คปัจจุบันระบบ Ethernet สามารถจะส่งข้อมูลด้วยความเร็ว 100 เมกะบิตต่อนาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การติดต่อสื่อสารระหว่างเวิร์กสเตชัน

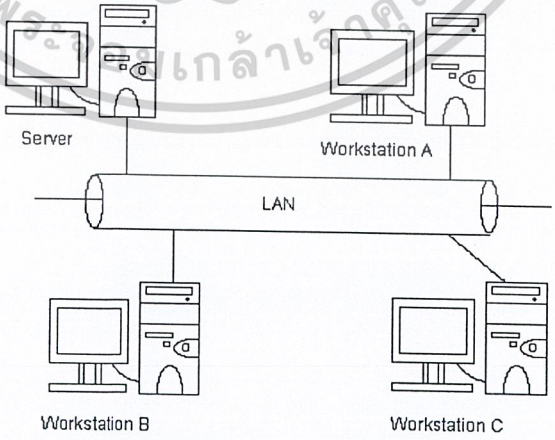
ในขณะที่เครื่องเวิร์กสเตชันแต่ละเครื่องกำลังทำงานอยู่นั้น บังเอิญต้องมีการปิด (Shut Down) เครื่องไฟล์เซิร์ฟเวอร์เพื่อเปลี่ยนฮาร์ดดิสก์หรือ อุปกรณ์อื่นๆ สามารถส่งข้อความไปยังเครื่องเวิร์กสเตชันทุกเครื่องเพื่อเตือนให้เซฟข้อมูลภายใน 10 นาที เช่น Shut Down File Server Within 10 minute. Please save data now. ข้อความเหล่านี้จะปรากฏที่หน้าจอเครื่องเวิร์กสเตชันทุกเครื่องทันที ถ้าเวลาการเซฟน้อยไปจะขอเพิ่มเป็น 20 นาที เครื่องเวิร์กสเตชันสามารถจะส่งข้อความไปบอกไฟล์เซิร์ฟเวอร์ได้ เช่น Please give 20 minute for save data … OK

Peer To Peer เป็นระบบที่เครื่องเวิร์กสเตชันทุกเครื่องบนระบบเน็ตเวิร์คมีฐานะเท่าเทียมกัน กล่าวคือทุกเครื่องสามารถที่จะใช้ไฟล์ข้อมูลในเครื่องอื่นๆได้ และสามารถให้เครื่องอื่นๆมาใช้ไฟล์ข้อมูลของเครื่องตนเองได้เช่นกัน โดยระบบPeer to Peer มีลักษณะการทำงานแบบ Distributed System โดยกระจายทรัพยากรต่างๆไปสู่เครื่องเวิร์กสเตชันอื่นๆ แต่จะมีปัญหาคือ เรื่องของการรักษาความปลอดภัย (Security) เพราะข้อมูลที่เป็นความลับก็จะถูกส่งออกไปสู่เวิร์กสเตชันเครื่องอื่นเช่นกัน ตัวอย่างของโปรแกรมที่มีความสามารถของ Peer to Peer คือ Windows for Workgroup, Personal Netware เป็นต้น

Client/Server เป็นการทำงานแบบ Distributed Processing หรือการประมวลผลแบบกระจาย โดยจะแบ่งกันประมวลผล ระหว่างเครื่องเซิร์ฟเวอร์กับเครื่องเวิร์กสเตชันด้วย และเมื่อใดที่เครื่องเวิร์กสเตชันต้องการผลลัพธ์ของข้อมูลบางส่วน จะมีการเรียกใช้ไปยังเครื่องเซิร์ฟเวอร์นำเฉพาะข้อมูลบางส่วนเท่านั้นส่งกลับมาให้เครื่องเวิร์กสเตชันเพื่อทำการคำนวณข้อมูลนั้นต่อไป

2.2.3 รูปแบบการเชื่อมต่อของระบบ LAN (Topology)

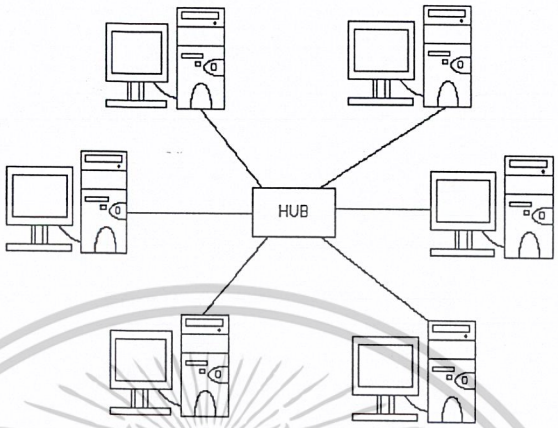
1.) Bus การเชื่อมต่อแบบบัสจะมีสายหลักอยู่ 1 เส้น เครื่องคอมพิวเตอร์ทั้งเซิร์ฟเวอร์และเวิร์กสเตชันทุกเครื่องจะเชื่อมต่อกับสายหลักเส้นนี้ เครื่องคอมพิวเตอร์จะถูกมองเป็น Node เมื่อเครื่องเวิร์กสเตชันเครื่องหนึ่ง (Node A) ต้องการจะส่งข้อมูลให้กับเครื่องอื่น (Node C) จะต้องส่งข้อมูล และแอดเดรสของ Node C ลงไปบนบัสสายหลัก เมื่อเครื่อง Node C ได้รับข้อมูลแล้วก็จะนำข้อมูลไปทำงานทันที ดังรูป



รูปที่ 2.5 แสดงการเชื่อมต่อแบบ Bus

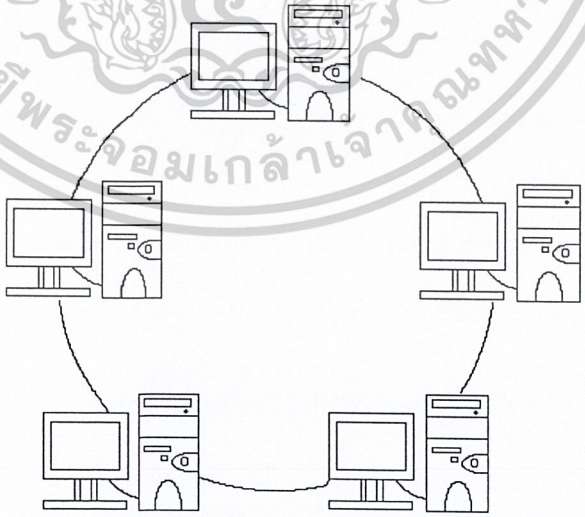
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) Star การเชื่อมต่อแบบสตาร์นี้จะใช้ Hub เป็นศูนย์กลางในการเชื่อมต่อ คือ เครื่องทุกเครื่องจะต่อผ่าน Hub สายเคเบิลที่ใช้ส่วนมากจะเป็น UTP และ Fiber Optic ในการส่งข้อมูล Hub จะเป็นเสมือนตัวทวนสัญญาณ (Repeater) และ Hub บางรุ่นยังสามารถตรวจจับข้อมูล (Data Detection) ต่างๆ เช่น Receive – Sent Data, Judders, Collision Data Short Frames ดังรูป



รูปที่ 2.6 แสดงการเชื่อมต่อแบบ Star

3.) Ring การเชื่อมต่อแบบวงแหวนนี้จะเป็นการเชื่อมต่อจากเครื่องคอมพิวเตอร์หนึ่งไปยังอีกเครื่องหนึ่งจนครบวงจร โดยการส่งข้อมูลจะส่งออกมาที่สายสัญญาณวงแหวนโดยจะเป็นการส่งผ่านจากเครื่องคอมพิวเตอร์หนึ่งไปยังอีกเครื่องหนึ่งจนกว่าจะถึงปลายทาง เครื่องคอมพิวเตอร์ปลายทางก็จะดึงข้อมูลนั้นออกไป จากนั้นเมื่อข้อมูลนี้กลับไปยังต้นทางเครื่องคอมพิวเตอร์ต้นทางก็จะทำการลบข้อมูลทิ้งไปเมื่อทำการตรวจแล้วว่าข้อมูลนี้สามารถส่งถึงปลายทางเรียบร้อยแล้ว แสดงได้ดังรูป



รูปที่ 2.7 แสดงการเชื่อมต่อแบบ Ring

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 โพรโตคอลมาตรฐานของระบบเครือข่าย

หน่วยงานสากลที่มีหน้าที่ในการกำหนดมาตรฐานของการออกแบบผลิตภัณฑ์อิเล็กทรอนิกส์ กำหนดรูปแบบการส่งสัญญาณ ฯลฯ คือ IEEE (Institute of Electrical and Electronics Engineers), EIA (Electronics – Industries Association) โดยจะมีโพรโตคอลอยู่ 3 แบบ ด้วยกัน คือ

1. ARC net
2. Ethernet
3. Token Ring

ARC net เป็นโพรโตคอลที่ออกแบบโดยบริษัท Data Point ประมาณปี 1977 (Attached ARC net Resource Computing network) ใช้หลักการออกแบบ "Transmission Permission" ในการส่งข้อมูลจะมีการกำหนดตำแหน่งแอดเดรสของเครื่องเวิร์กสเตชันลงไปด้วย สามารถจะเชื่อมต่อได้ทั้ง แบบ Bus และ Star มีความเร็วในการส่งผ่านข้อมูลน้อยมากเพียง 2.5 Mbps (2.5 เมกะบิตต่อวินาที) ทำให้ไม่เป็นที่นิยมใช้งาน

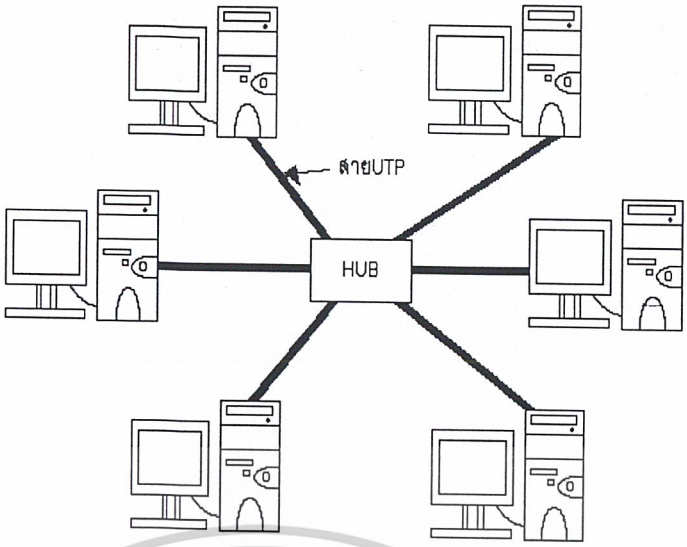
Ethernet เป็นโพรโตคอลที่ออกแบบโดยบริษัท Xerox ประมาณปี 1970 ใช้หลักการทำงานแบบ CSMA/CD (Carrier Sense Multiple Access With Collision Detection) ในการส่งข้อความไปบนสายสัญญาณของระบบเครือข่าย ถ้าหากมีการส่งออกมาพร้อมกันย่อมจะเกิดการชนกัน (Collision) ของสัญญาณทำให้การส่งผ่านข้อมูลต้องหยุดลงทันที CSMA/CD จะใช้วิธีของ Listen-before-transmitting คือ ก่อนจะส่งสัญญาณออกไปจะต้องตรวจสอบว่า ในขณะนั้นมีเครื่องเวิร์กสเตชันเครื่องใดทำการ รับ - ส่ง ข้อมูลบนสายเคเบิลอยู่หรือไม่ ถ้ามีต้องรอจนกว่าสายเคเบิลจะว่าง แล้วส่งข้อมูลออกไปบนสายเคเบิล

ระบบโพรโตคอล Ethernet นั้นเป็นมาตรฐานของ IEEE 802.3 สามารถจะเชื่อมต่อได้ทั้งแบบ Bus และ Star โดยใช้สาย Coaxial หรือ สายทองแดงคู่ตีเกลียว (UTP = Unshielded Twisted Pair) มีความเร็วในการรับ-ส่งข้อมูล 10 Mbps (10 เมกะบิตต่อวินาที) ในปัจจุบันพัฒนาความเร็วเกิน 100 Mbps มีความยาวสูงสุดระหว่างเครื่องเวิร์กสเตชัน 2.8 กิโลเมตร ในการส่งผ่านสัญญาณอิเล็กทรอนิกส์ไปบนสายเคเบิลจะใช้แบบ Manchester-Encoded Digital Base band และกล่าวถึงสัญญาณดิจิตอล 0-1 ในการส่งผ่านไปบนสายเคเบิล Ethernet มีรูปแบบการต่อสายเคเบิล 3 แบบ ด้วยกันคือ

1. 10 Base T
2. 10 Base 2
3. 10 Base 5

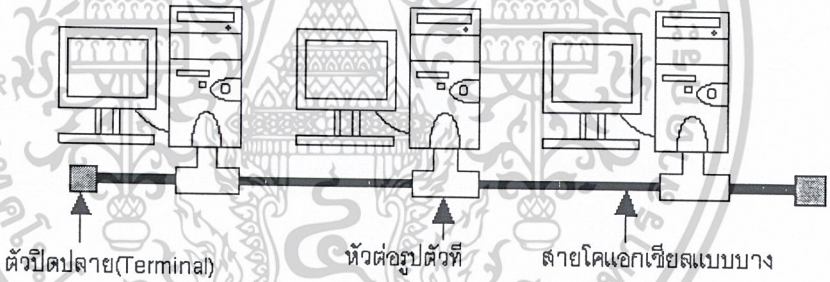
10 Base T เป็นรูปแบบในการต่อสาย Twisted Pair ในการรับ-ส่งข้อมูลที่มีความเร็ว 10Mbps ด้วยสัญญาณแบบ Base band ปัจจุบันจะใช้สาย UTP (Unshielded Twisted Pair) ซึ่งจะมีสายเส้นเล็กๆ ภายใน 8 เส้นตีเกลียวกัน 4 คู่ ด้วยความยาวไม่เกิน 100 เมตร แสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงประเภทของอินเตอร์เน็ตแบบ 10 Base T

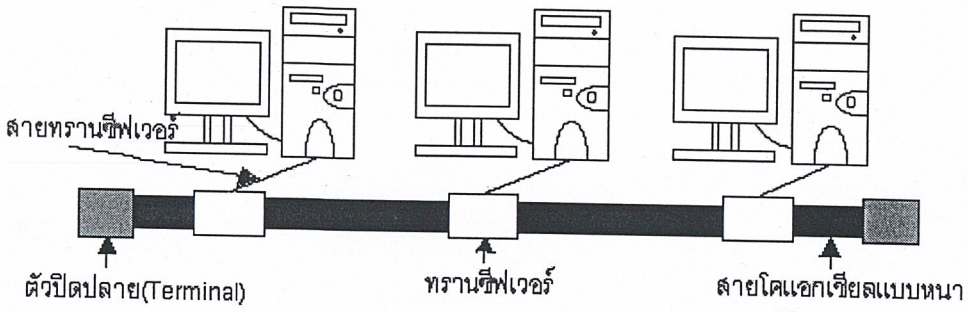
10 Base 2 เป็นรูปแบบในการต่อซึ่งใช้สายโคแอกเชียลแบบบาง (Thin Ethernet) ซึ่งความยาวของสายในเซกเมนต์หนึ่งๆไม่เกิน 185 เมตร แสดงดังรูป



รูปที่ 2.9 แสดงประเภทของอินเตอร์เน็ตแบบ 10Base2

10 Base 5 เป็นรูปแบบในการต่อซึ่งใช้สายโคแอกเชียลแบบหนา (Thick Ethernet) การเชื่อมต่อแต่ละจุดจะมี Transceiver เป็นตัวเชื่อมต่อ และใช้สายเคเบิล AUI เชื่อมระหว่างเครื่องเวิร์กสเตชั่น ซึ่งความยาวของสายในเซกเมนต์หนึ่งๆไม่เกิน 500 เมตร แสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงประเภทของอินเตอร์เน็ตแบบ 10Base5

โดยรหัสขึ้นต้นด้วย10 หมายถึงความเร็วสายสัญญาณ 10 เมกะบิตต่อวินาที คำว่า Base หมายถึงสายสัญญาณชนิด Base band รหัสถัดมาหากเป็นตัวเลขหมายถึงความยาวสายต่อเซกเมนต์ในหน่วย 100เมตร (5 = 500 ,2 แทนค่า 185) หากตัวเป็นอักษรจะหมายถึงชนิดของสายเช่น T หมายถึง Twisted Pair หรือ F คือ Fiber Optic เป็นต้น

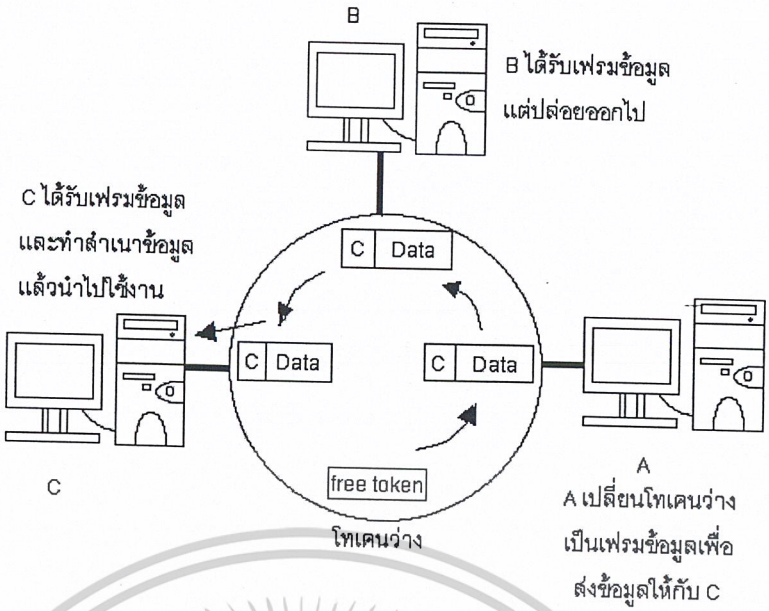
ส่วนมาตรฐานอินเตอร์เน็ตความเร็ว 100 เมกะบิตต่อวินาทีที่นิยมใช้ในปัจจุบันได้แก่ 100 Base TX และ 100 Base FX สำหรับอินเตอร์เน็ตความเร็วสูงแบบกิกะบิตอินเตอร์เน็ตเริ่มแพรหลายมากขึ้น ตัวอย่างในปัจจุบันได้แก่ 1000 Base T ,1000 Base LX และ 1000 Base FX เป็นต้น

ในเครือข่ายอินเตอร์เน็ตที่มีสถานีที่มาต่อเป็นจำนวนมากจะพบว่าการทำงานจะล่าช้าเพราะแต่ละสถานพยายามยึดช่องสัญญาณเพื่อส่งข้อมูลและเกิดการชนกันตลอดเวลาโดยไม่สามารถกำหนดได้ว่าสถานีใดจะได้ใช้สายสัญญาณเมื่อเวลาใด ดังนั้นอินเตอร์เน็ตจึงไม่เหมาะกับการใช้งานในระบบเวลาจริง

โทเคนริง (Token Ring) หรือ IEEE 802.5 เป็นเครือข่ายที่ใช้โทโปโลยีแบบวงแหวนด้วยสายคู่ตีเกลียวหรือเส้นใยนำแสงโดยอัตราการรับส่งข้อมูลของโทเคนริงที่ใช้โดยทั่วไปคือ 4 และ 16 เมกะบิตต่อวินาที ซึ่งหลักการทำงานของโทเคนริงคือ จะมีเฟรมพิเศษที่เรียกว่า โทเคนว่าง (free token) วิ่งวนอยู่สถานีที่ต้องการรับส่งข้อมูลจะรอให้โทเคนว่างเดินทางมาถึงแล้วรับโทเคนว่างมาเปลี่ยนเป็น เฟรมข้อมูล (data frame) โดยใส่ flag แสดงเฟรมข้อมูลและบรรจุแอดเดรสของสถานีต้นทางและปลายทางตลอดจนข้อมูลอื่นๆ จากนั้นสถานีจึงปล่อยเฟรมนี้ออกไป เมื่อทางสถานีปลายทางได้รับเฟรมข้อมูลก็จะทำสำเนาข้อมูลไว้และปล่อยเฟรมข้อมูลกลับมายังสถานีต้นทาง สถานีต้นทางจะตรวจสอบเฟรมข้อมูลหากพบว่าสถานีปลายทางได้เก็บข้อมูลไปแล้วก็จะทำการลบข้อมูลนั้นออกไป และปล่อยโทเคนว่างคืนสู่เครือข่ายให้สถานีอื่นมีโอกาสใช้ต่อไป แสดงได้ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

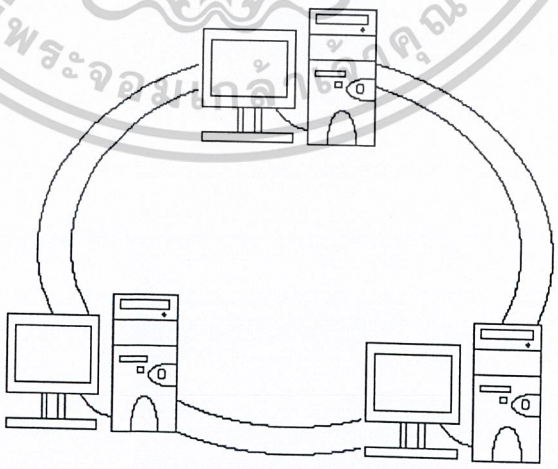


รูปที่ 2.11 แสดงหลักการทำงานของโทเคนริง

โดยกลไกแบบส่งผ่านโทเคนจัดอยู่ในประเภทประเมินเวลาได้ กล่าวคือสามารถคำนวณเวลาสูงสุดที่แต่ละสถานีมีสิทธิ์จับโทเคนริงเพื่อส่งข้อมูลได้ โทเคนริงจึงเหมาะกับระบบที่ต้องการความแน่นอนทางเวลาหรือทำงานแบบเวลาจริง

เอฟดีดีไอ (FDDI:Fiber Distributed Data Interface) มีอัตราการรับส่งข้อมูลที่ 100 เมกะบิตต่อวินาทีจึงมักใช้เป็นแกนหลัก(backbone) ซึ่งเป็นส่วนของเครือข่ายที่จะต้องรองรับภาระการสื่อสารในปริมาณมาก และนิยมใช้กับระบบงานเวลาจริงเช่นเดียวกับโทเคนริงเนื่องจากใช้หลักการเดียวกับโทเคนริง

โทโปโลยีของเอฟดีดีไอเป็นแบบวงแหวนคู่ดังรูป



รูปที่ 2.12 แสดงโทโปโลยีเอฟดีดีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
50092

วงแหวนของเอฟดีดีไอประกอบด้วยวงแหวนหลักเรียกว่า วงแหวนปฐมภูมิ (Primary Ring) และวงแหวนรองเรียกว่า วงแหวนทุติยภูมิ (Secondary Ring) การออกแบบให้มีวงแหวนสองวงเพื่อให้มีเส้นทางสำรอง หากวงแหวนหลักเกิดชำรุดระบบจะเปลี่ยนไปใช้วงแหวนสำรองแทน

2.3 โปรโตคอลในการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ผ่านอินเตอร์เน็ต

อินเตอร์เน็ตเป็นเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นๆ เชื่อมโยงเข้ามาใช้งานได้ แต่ปัญหาที่เกิดในการเชื่อมโยงเครือข่ายเข้าด้วยกันก็คือ เครือข่ายมีความต่างชนิด ต่างยี่ห้อ และระบบปฏิบัติการที่แตกต่างกัน มาตรฐาน TCP/IP เป็นสิ่งที่ถูกนำมาแก้ปัญหานี้ให้เป็นระบบเปิดที่สมบูรณ์แบบที่สามารถเชื่อมต่อเครือข่ายที่ทั่วโลกยอมรับ

ในหัวข้อนี้จะกล่าวถึงลักษณะการทำงานของอินเตอร์เน็ตซึ่งประกอบไปด้วยเรื่องเกี่ยวกับ TCP/IP การกำหนดชื่อ และเลข IP อินเตอร์เน็ตนับได้ว่าเป็นเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นๆ เชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่นๆ อีก แต่ปัญหาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันคือ แต่ละเครื่องใช้ข่ายคอมพิวเตอร์ต่างชนิด ต่างยี่ห้อและต่างระบบปฏิบัติการ มาตรฐานของTCP/IP จึงถูกใช้เป็นกุญแจสำคัญในการแก้ปัญหาเหล่านี้ โดยจะกลายเป็นระบบเปิดที่สมบูรณ์และมีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่พีซี และไม่จำกัดระบบปฏิบัติการที่ใช้ TCP/IP จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟท์แวร์ผลิตออกมาสนับสนุน มากมาย ดังนั้นจึงนับได้ว่า TCP/IP เป็นหัวใจของอินเตอร์เน็ตเลยทีเดียว

2.3.1 ความหมายของ TCP/IP

TCP/IP เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Networking Protocol) จัดทำขึ้นเพื่อใช้เป็นเกณฑ์ให้เครื่องคอมพิวเตอร์ใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตามจะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้ เช่น เครื่องคอมพิวเตอร์ของ Digital Equipment ซึ่งเป็นระบบMiniติดต่อสื่อสารกับ Compaq ซึ่งเป็นเครื่อง PC ได้เมื่อดำเนินการด้วย TCP/IP

TCP/IP เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟท์แวร์ การจัดโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่าง ๆ ที่กำหนดให้ทำเมื่อเกิดความผิดพลาด หรือต้องทำเพื่อป้องกันไม่ให้เกิดความผิดพลาด

TCP/IP เกิดจากการนำข้อกำหนดของรูปแบบต่างๆ กันเข้ามาใช้ร่วมกัน TCP/IP และ TCP/IP ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า TCP/IP ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายเดียวกัน หรือต่างเครือข่ายกันก็ได้ และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในดังข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบเครือข่ายจะใช้ TCP/IP เป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ LAN (Local Area Network) และเครือข่ายบริเวณกว้าง WAN (Wide Area Network) ไม่ได้ใช้งานเฉพาะกับอินเตอร์เน็ตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ชุดโปรโตคอลทีซีพี/ไอพี (TCP/IP protocol suite)

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบด้วยชุดข้อกำหนดรูปแบบต่างๆ ซึ่งแบ่งเป็นกลุ่ม ได้ดังนี้

กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocol)

ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง แบ่งย่อยออกได้เป็น 2 ชนิดคือ

1. TCP (Transmission Control Protocol) ทำหน้าที่นำส่งข้อมูลโดยรับประกันความเชื่อถือ ทีซีพีด้านส่งต้องส่งแพคเกตซ้ำใหม่หากแพคเกตสูญหาย ทีซีพีด้านรับมีหน้าที่จัดแพคเกตให้ถูกต้องตามลำดับและกำจัดแพคเกตที่ซ้ำซ้อน ทีซีพีเป็นโปรโตคอลแบบ connection oriented คือต้องสถาปนาการเชื่อมต่อระหว่างสถานีต้นทางและสถานีปลายทางก่อนการรับส่งข้อมูล

ทีซีพีต้นทางจัดแบบข้อมูลเพื่อส่งให้ไอพีดำเนินการ ทีซีพีปลายทางเมื่อรับแพคเกตจากไอพีก็จะส่งต่อให้โปรโตคอลประยุกต์ไปดำเนินการต่อไป

2. UDP (User Datagram Protocol) เป็นโปรโตคอลแบบ connectionless คือไม่ต้องสถาปนาการเชื่อมต่อระหว่างสถานีต้นทางและสถานีปลายทาง โดยเป็นโปรโตคอลระดับชั้นเดียวกับทีซีพีแต่ว่าไม่มีกลไกรับประกันความเชื่อถือในการรับส่งข้อมูล หากข้อมูลหาย ซ้ำซ้อน หรือลำดับไม่ถูกต้อง ยูดีพีจะปล่อยให้โปรโตคอลที่เรียกใช้ตัวมันจัดการกับปัญหาเอง

ลักษณะเด่นของยูดีพีคือใช้การประมวลผลต่ำกว่าทีซีพี เนื่องจากเฮดเดอร์มีขนาดเล็กและไม่ต้องสถาปนาการเชื่อมต่อ

กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol)

ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูล และถ้ามีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่กลุ่มข้อกำหนดรูปแบบนี้ก็จะแบ่งย่อยข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่ กลุ่มข้อกำหนดรูปแบบนี้จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสมแล้วส่งออกไป เมื่อถึงผู้รับปลายทางกลุ่มข้อกำหนดนี้ก็จะทำหน้าที่ตรงข้าม คือรวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผลกลุ่ม ข้อกำหนดรูปแบบนี้ประกอบด้วย

1. IP (Internet Protocol) เป็นโปรโตคอลแกนของทีซีพี/ไอพี โดยทำหน้าที่กำหนดรูปแบบของแอดเดรสประจำเครื่องเพื่อใช้ในการลำเลียงข้อมูลจากเครื่องต้นทางไปยังเครื่องปลายทาง นอกจากนี้ยังทำหน้าที่เลือกเส้นทางส่งข้อมูล ตลอดจนแบ่งขนาดข้อมูลให้เหมาะสมกับฮาร์ดแวร์ระดับล่าง

2. ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับสถานะของ IP เช่น ข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ในเครือข่าย

3. RIP (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่เหมาะสมที่สุดกับข้อมูลมากที่สุด

4. OSPF (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่ง ที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากเส้นทางที่สั้นที่สุดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address)

ทำหน้าที่พิจารณาที่อยู่ของเครือข่าย และเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องและข้อมูลที่จะไปยังผู้รับปลายทาง โดยไม่ว่าเครือข่ายจะใหญ่โตแค่ไหน หรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตามที่อยู่จะต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบกลุ่มนี้มีดังนี้

1. ARP (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่ เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน

2. DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะในการใช้งานจริงนั้นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบชื่อจัดทำขึ้นเพื่อให้สะดวกในการใช้งานของผู้ใช้

3. RARP (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เช่นเดียวกับ ARP แต่จะทำตรงกันข้ามกับ ARP

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocol)

และสนับสนุนข้อมูลสถานะเพื่อนำไปใช้ในการเลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1. EGP (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะทำการถ่ายโอนข้อมูลเส้นทางกันระหว่าง Gateway กับเครือข่ายภายนอกเพื่อทำการสื่อสาร

2. GGP (Gateway–to-Gateway Protocol) ข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่าง Gateway กับ Gateway

3. IGP (Interior Gateway Protocol) ข้อกำหนดรูปแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

Gateway เป็นอุปกรณ์หรือเครื่องคอมพิวเตอร์ ที่ทำหน้าที่เสมือนประตูสื่อสารเป็นช่องสัญญาณเข้าหรือออกไปยังระบบสื่อสารอื่น หรือภายในเครือข่ายเดียวกันติดต่อกัน

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services)

ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดรูปแบบนี้ประกอบด้วย

1. BOOTP (Bootstrap Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายเริ่มทำงาน ข้อกำหนดรูปแบบนี้จะอ่านโปรแกรมควบคุมการทำงานจากคอมพิวเตอร์ให้บริการ Server computer) มาให้

2. FTP (File Transfer Protocol) ให้บริการถ่ายโอนแฟ้มข้อมูลระหว่างเครื่อง โดยช่วยให้ผู้ใช้เข้าถึงโฮสต์และจำกัดขอบเขตการทำงานเฉพาะที่เกี่ยวข้องกับแฟ้มข้อมูลเช่น ทำสำเนา ลบ หรือสร้างไดเรกทอรี เป็นต้น

3. TFTP (Trivial File Transfer Protocol) ทำหน้าที่เป็นโปรโตคอลถ่ายโอนแฟ้มข้อมูลเช่นเดียวกับเอฟทีพี แต่ทีเอฟทีพีใช้บริการผ่านยูดีพี ซึ่งประโยชน์ของการใช้งานทีเอฟทีพีได้แก่การใช้ใน สถานีไร้ดิสค์ (Diskless Workstation) ซึ่งไม่มีดิสค์ประจำตัว สถานีไร้ดิสค์จะบูทระบบด้วยโปรโตคอล บูทพี และใช้ทีเอฟทีพีเพื่อถ่ายโอนระบบปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. TELNET (Telecommunication Network) เป็นโปรโตคอลสำหรับขอเข้าใช้โฮสต์ระยะไกล หรือ รีโมตล็อกอิน (Remote Login) เทลเน็ตให้บริการเข้าใช้คอมพิวเตอร์ในเครือข่ายโดยเสมือนกับว่ากำลังทำงานอยู่ที่เทอร์มินัลของคอมพิวเตอร์เครื่องนั้น เทลเน็ตเซิร์ฟเวอร์ที่คอมพิวเตอร์ปลายทางจะรอรับการขอใช้บริการจากเทลเน็ตไคลเอนต์ ผู้ขอใช้เทลเน็ตจะต้องมีบัญชีประจำเครื่องที่ให้บริการเทลเน็ต

กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญจัดไว้ให้บนระบบเครือข่ายที่สนใจมีดังนี้

- NFS(Network File System) เป็นโปรโตคอลที่พัฒนาขึ้น เพื่อใช้งานในระบบรีโมตดิสค์ เพื่อให้คอมพิวเตอร์เครื่องหนึ่งสามารถใช้แฟ้มข้อมูลที่เก็บอยู่ในดิสค์ของอีกเครื่องหนึ่งเสมือนว่าเป็นแฟ้มที่อยู่ในดิสก์ของเครื่องนั้น

- NIS (Network Information Services) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Login และ Password

- RPC (Remote Procedure Call) เป็นข้อกำหนดรูปแบบที่อำนวยความสะดวก ให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล

- SNMP (Simple Network Management Protocol) ทำหน้าที่เป็นโปรโตคอลช่วยบริหารระบบ เช่น เก็บรวบรวมข้อมูลการทำงานของอุปกรณ์และคอมพิวเตอร์ในเครือข่าย ตรวจสอบปริมาณข้อมูลที่ไหลเวียน หรือช่วยวิเคราะห์หาข้อผิดปกติในระบบ เป็นต้น

2.3.3 แบบอ้างอิง TCP/IP

เครือข่ายคอมพิวเตอร์มีแบบอ้างอิงที่ใช้เป็นมาตรฐานคือ OSI Model ในขณะที่ TCP/IP มีแบบอ้างอิงเมื่อเทียบกับ OSI Model ดังนี้

- Application Layer ในชั้นนี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้ในเครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูลและอาจกล่าวได้ว่าโปรโตคอล TCP/IP เป็นโปรโตคอลในชั้น Application Layer รวมกับ Presentation Layer ใน OSI model

- Transport Layer ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายเปรียบเทียบได้กับ Session Layer รวมกับ Transport Layer โดยโปรโตคอลจะมีซ็อคเกต (Socket) เป็นจุดปลายในการสื่อสาร โดยซ็อคเกตประกอบด้วยหมายเลขของเครื่องคอมพิวเตอร์และพอร์ทของเครื่องที่ต้องการข้อมูลส่งเข้าไปถึง ในชั้นนี้มีการรองรับให้ถึงที่หมายและลำดับข้อมูลไม่มีการซ้ำซ้อน โดยในชั้นนี้จะมีโปรโตคอลหลัก 2 ตัว คือ TCP และ UDP

- Internet Layer ในชั้นนี้มีการกำหนด Datagram และทำการหาเส้นทางการส่ง การทำงานในชั้นนี้เป็นแบบ Connectionless เนื่องจากไม่มีการเชื่อมต่อระหว่างต้นทางกับปลายทางก่อนโดย Datagram แต่ละตัวสามารถเลือกเส้นทางได้อย่างอิสระ ไม่รับประกันความถูกต้องของข้อมูล

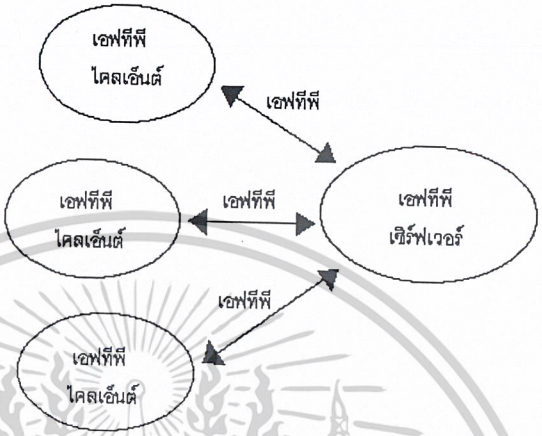
- Network Interface Physical Layer ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็นส่วน ๆ เรียกว่าเฟรม หรือ แพคเกต และส่งข้อมูลออกไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 TCP/IP Client - Server

โปรโตคอลใน TCP/IP อาศัยหลักการทำงานตามแบบ Client-Server ซึ่งมีองค์ประกอบ 3 ส่วนคือ

1. โปรเซสไคลเอ็นต์ซึ่งทำหน้าที่ขอบริการ
2. โปรเซสเซิร์ฟเวอร์ซึ่งทำหน้าที่ให้บริการ
3. โปรโตคอลที่ใช้สื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์



รูปที่ 2.13 แสดงเอฟทีพีไคลเอ็นต์ – เซิร์ฟเวอร์

จากรูปแสดงการเชื่อมโยงระหว่างไคลเอ็นต์ 3 โปรเซส ซึ่งขอใช้บริการถ่ายโอนแฟ้มข้อมูลจากเซิร์ฟเวอร์ผ่านเอฟทีพี โปรเซสไคลเอ็นต์และเซิร์ฟเวอร์ไม่จำเป็นต้องอยู่ต่างเครื่องกัน เนื่องจากระบบปฏิบัติการส่วนใหญ่ที่ใช้ ทีซีพี/ไอพี มักทำงานแบบผู้ใช้หลายคนและพร้อมกันหลายงาน แต่ละ ไคลเอ็นต์จึงอาจเป็นโปรเซสของผู้ใช้ต่างบุคคลที่ทำงานภายในเครื่องเดียวกันและขอบริการไปยังเซิร์ฟเวอร์ต่างเครื่องหรือเครื่องเดียวกันกับไคลเอ็นต์ก็ได้

2.3.5 ข้อเปรียบเทียบและความสัมพันธ์ระหว่างโปรโตคอล TCP และ UDP

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆ ก่อน แล้วจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูลดาต้าแกรม (datagram) ใหม่ให้ต่อเนื่องกันและประกอบกลับเป็นข้อมูลทั้งหมดได้ ซึ่งจะแยกข้อมูลที่ไม่ถูกต้องออก ดังนั้นแอพพลิเคชันหรือโปรเซสใดๆ ที่อาศัยการส่งผ่านข้อมูลด้วยโปรโตคอล TCP จะต้องใช้หน่วยความจำและขนาดช่องสัญญาณ (Bandwidth) มากกว่า UDP

การติดต่อสื่อสารระหว่างกันจะต้องเป็นแบบ connection-oriented คือต้องมีการสร้างการติดต่อกันเป็นเซสชั่น (session) ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมๆ กัน (Full Duplex) เมื่อแอพพลิเคชั่นต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้นโพรเซสเลเยอร์ (Process Layer) ติดต่อไปและมีการสร้างช่องสัญญาณข้อมูลผ่านพอร์ตที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโปรโตคอล TCP ในระหว่างการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับส่งข้อมูลนี้ โปรโตคอล TCP จะเพิ่มขบวนการสอบทานข้อมูลเพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น ซึ่งทำให้ความน่าเชื่อถือของการส่งผ่านข้อมูลโดย โปรโตคอล TCP จะมีมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบมากกว่าในการทำงานเช่นกัน

ส่วนโปรโตคอล UDP โนการทำงานในชั้นการทำงานระหว่างโฮสต์ นอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ก็ยังมีโปรโตคอล UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบที่ทั้ง 2 ด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกันระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับ ข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลแบบนั้นๆ ด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอพพลิเคชั่น หรือ โปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการการตรวจสอบขึ้นมาเอง

2.3.6 พอร์ตของ TCP/IP

เมื่อพิจารณาข้อมูลที่ไอพีลำเลียงถึงสถานีปลายทางและส่งผ่านตามระดับชั้นขึ้นมาถึง Transport Layer ในไอพีเฮดเดอร์จะระบุชนิดของโปรโตคอลว่าต้องการเรียกใช้บริการของทีซีพีหรือยูดีพี ต่อจากนั้นทีซีพีหรือยูดีพีจะส่งข้อมูลไปยังโปรโตคอลระดับบนตามหมายเลขประจำโปรโตคอล หมายเลขที่ใช้ระบุ โปรโตคอลประยุกต์เป็นตัวเลขขนาด 16 บิต ทำให้สามารถมีหมายเลขได้ตั้งแต่1 ถึง 65536 และเรียกว่า พอร์ต (Port) หรือกล่าวได้ว่าพอร์ตเสมือนเป็นแอดเดรสประจำโปรโตคอลในชั้นประยุกต์

TCP/IP ได้สงวนพอร์ตหมายเลข1ถึง1023 ไว้ใช้ประจำโปรโตคอลประยุกต์โดยเรียกหมายเลขพอร์ตนี้ว่า well-known port ซึ่งใช้เป็นมาตรฐานทุกสถานีหรืออุปกรณ์เครือข่าย ตัวอย่างเช่น เซิร์ฟเวอร์ที่ให้บริการเทลเน็ตใช้ทีซีพีพอร์ตเบอร์ 23 , เซิร์ฟเวอร์ที่ให้บริการเอฟทีพีใช้ทีซีพีพอร์ตเบอร์ 21 เป็นต้น

โดยโปรโตคอลประยุกต์หนึ่งๆอาจเรียกใช้บริการได้ทั้งทีซีพีและยูดีพีพร้อมกัน โดยใช้หมายเลขพอร์ตเดียวกันแต่เป็นอิสระต่อกัน เช่น echo มีหมายเลขพอร์ตเท่ากับ 7 ทั้งทีซีพีและยูดีพี หากมีการขอบริการทีซีพีและยูดีพีพร้อมกันที่พอร์ตนี้ โปรเซสที่เกิดขึ้นจะเป็นอิสระต่อกัน

2.3.7 ซ็อคเกตของ TCP/IP

ซ็อคเกต (socket) หรือ ซ็อคเกตแอดเดรส (socket address) หมายถึงคู่ของไอพีแอดเดรสและหมายเลขพอร์ต ทีซีพีและยูดีพีอาศัยซ็อคเกตเป็นตัวแยกแยะโปรเซสต้นทางและปลายทางที่ติดต่อระหว่างกัน หากพิจารณาถึงโปรโตคอลประยุกต์ซึ่งอนุญาตให้บริการได้หลายเซสชั่นพร้อมๆกัน และจำเป็นต้องแยกแยะเซสชั่นที่เกิดขึ้นให้ได้เพื่อให้ข้อมูลรับส่งได้ถูกต้องตามคู่ต้นทางและปลายทาง

โดยเซิร์ฟเวอร์ปลายทางจะใช้หมายเลขพอร์ตตามพอร์ตมาตรฐานประจำแอพลิเคชั่นสำหรับสร้างซ็อคเกต แต่ไคลเอนต์ต้นทางจะเลือกหมายเลขพอร์ตนอกเหนือจากพอร์ตมาตรฐานมาใช้เป็นซ็อคเกตต้นทาง

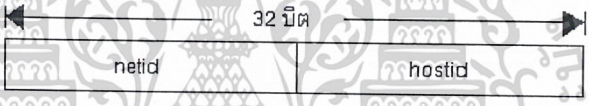
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.8 หมายเลข IP Address

อุปกรณ์ที่เชื่อมเข้าเครือข่ายและสามารถทำงานได้ตามข้อกำหนดของ TCP/IP จะต้องมีแอดเดรสประจำอุปกรณ์นั้น โดยไอพีรุ่นสี่ได้กำหนดให้ใช้ไอพีแอดเดรสขนาด 32 บิต อุปกรณ์ที่เชื่อมกับอินเตอร์เน็ตจะมีไอพีแอดเดรสขนาด 32 บิตประจำอินเตอร์เฟสที่ไม่ซ้ำกัน อุปกรณ์ที่มีหลายอินเตอร์เฟสจะมีไอพีแอดเดรสหลายค่าตามจำนวนอินเตอร์เฟสที่มีอยู่โดยไม่ซ้ำค่ากัน แต่คอมพิวเตอร์หรือโฮสต์ที่พบโดยทั่วไปมักมีเพียงอินเตอร์เฟสเดียวจึงมักเรียกไอพีแอดเดรสว่าเป็นแอดเดรสประจำโฮสต์

แอดเดรสขนาด 32 บิตมีจำนวนแอดเดรสรวมเท่ากับ 4291967296 แต่เมื่อนำมาจัดสรรแล้วจะไม่สามารถใช้งานได้ครบทั้งหมด ไอพีแอดเดรสนิยมเขียนในรูปเลขฐานสิบตามแบบ dotted decimal โดยแบ่งตัวเลข 32 บิตออกเป็น 4 ไบต์ แต่ละไบต์แทนด้วยเลขฐานสิบหนึ่งตัวและคั่นแต่ละไบต์ด้วยเครื่องหมายจุด เช่น แอดเดรส 10011110 01101100 00000010 00000001 จะได้เป็น 158.108.2.1

แอดเดรสขนาด 32 บิตประกอบขึ้นจากหมายเลข 2 ส่วนคือ หมายเลขเครือข่าย (Network Identifier) และ หมายเลขโฮสต์ (Host Identifier) หมายเลขเครือข่ายใช้สำหรับจัดคลาสเครือข่าย ส่วนหมายเลขโฮสต์ใช้ระบุหมาเลขโฮสต์ในเครือข่าย ดังนั้นไอพีแอดเดรสจึงแบ่งได้เป็น2ส่วนดังรูป จำนวนบิตที่ใช้สำหรับหมายเลขเครือข่ายและหมายเลขโฮสต์จะขึ้นกับคลาสที่สังกัด



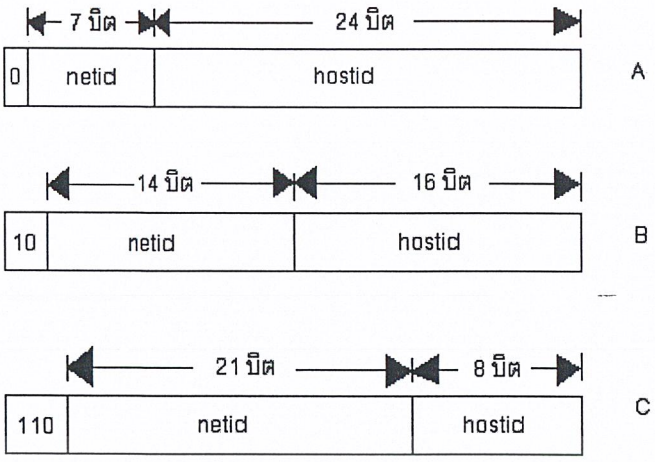
รูปที่ 2.14 แสดงรูปแบบของไอพีแอดเดรส

ในปัจจุบันฟิลด์กำหนดหมายเลขเครือข่ายนิยมเรียกว่า พรีฟิกซ์เครือข่าย (network-prefix) เพราะทุกโฮสต์ในเครือข่ายจะต้องมีพรีฟิกซ์หรือบิตนำหน้าเหมือนกัน

2.3.8.1 การจัดคลาสเครือข่าย

ไอพีแอดเดรสมีการจัดแบ่งออกเป็นกลุ่มหรือคลาส(Class) เครือข่ายที่ใช้งานในปัจจุบันมักสังกัดอยู่ในคลาสใดคลาสหนึ่งคือ คลาส A , B หรือ C การแบ่งคลาสอาศัยจำนวนพรีฟิกซ์เครือข่ายที่แตกต่างกันตามรูป แต่ละคลาสจึงมีจำนวนเครือข่ายในสังกัดและจำนวนโฮสต์ต่อเครือข่ายไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงการแบ่งคลาสเครือข่าย

การจัดแบ่งคลาสโดยใช้พรีฟิกซ์เป็นการผนวกข้อมูลเพื่อใช้ในการเลือกเส้นทาง เช่น หากตรวจสอบพบว่าพรีฟิกซ์ 2 บิตแรกมีค่าเป็น 10 แสดงว่าเป็นแอดเดรสในคลาส B

2.3.8.2 ลักษณะสำคัญของแต่ละคลาส

จำนวนเครือข่ายในแต่ละคลาสและจำนวนโฮสต์สูงสุดที่มีได้สามารถคำนวณได้จากจำนวนบิตที่ใช้งานตามสูตร 2^n เมื่อ n คือจำนวนบิต เช่นในคลาส B มีเลขโฮสต์จำนวน16บิต จึงมีโฮสต์ได้ไม่เกิน 2^{16} บิตซึ่งเท่ากับ 65536 แต่เลขโฮสต์ที่ทุกบิตเป็น 0 และ 1 จะสงวนไว้ใช้งานเฉพาะกรณี จำนวนโฮสต์จึงลดไป 2 โฮสต์ในทุกๆ เครือข่ายหรือมีโฮสต์ได้ไม่เกิน $2^{16}-2 = 65534$ เป็นต้น

<u>คลาส A</u>

เครือข่ายในคลาส A มีบิตซ้ายสุดเป็น 0 และใช้7บิตถัดมากำหนดเลขเครือข่าย ส่วนอีก24บิตเป็นเลขโฮสต์ คลาส A จึงมีจำนวนเครือข่ายได้ 2^7 หรือ 128 ค่า แต่เครือข่าย 0.0.0.0 และ 127.0.0.0 ถูกสงวนไว้เป็นแอดเดรสเฉพาะคือ แอดเดรส 0.0.0.0 ใช้เป็นแอดเดรสกำหนดเส้นทางโดยปริยาย (default route) ส่วนแอดเดรส 127.0.0.0 เป็นแอดเดรสลูปแบ็ค ดังนั้นจำนวนเครือข่ายในคลาส A จึงมีได้ 126 เครือข่าย

แต่ละเครือข่ายในคลาส A จะมีแอดเดรสได้ $2^{24}-2$ เท่ากับ 16777214 โดยเครือข่ายในคลาส A มักใช้งานกับหน่วยงานขนาดใหญ่ที่ต้องการแอดเดรสเป็นจำนวนมาก

<u>คลาส B</u>

เครือข่ายในคลาส B มีบิตเริ่มเป็น 10 และใช้14บิตถัดมาเป็นตัวกำหนดเลขเครือข่าย จำนวนบิตที่กำหนดเลขโฮสต์มีขนาด16 บิต คลาส B จึงมีสมาชิกเครือข่ายได้ $2^{14}-2$ หรือ 16382 ค่า แต่ละเครือข่ายมีเลขโฮสต์ได้ $2^{16}-2$ หรือ 65534 แอดเดรส

เครือข่ายในคลาส B มักจัดสรรให้กับหน่วยงานขนาดกลาง ในปัจจุบันมีเครือข่ายในคลาส B เหลือไม่มากนัก และมักไม่จัดสรรเครือข่ายในคลาสนี้ให้กับผู้ขอจดทะเบียนรายใหม่หากไม่มีความจำเป็นอย่าง แท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส C

เครือข่ายในคลาส C มีบิตเริ่มเป็น 110 และใช้21บิตถัดมาเป็นตัวกำหนดเลขเครือข่าย จำนวนบิตที่กำหนดเลขโฮสต์มีขนาด8บิต คลาส B จึงมีสมาชิกเครือข่ายได้ 2^{21}-2 หรือ 2097150 ค่า แต่ละเครือข่ายมีเลขโฮสต์ได้ 2^{8}-2 หรือ 254 แอดเดรส

จำนวนแอดเดรสที่มีจำกัดเพียง 254 แอดเดรส ทำให้เครือข่ายในคลาสนี้เหมาะกับหน่วยงานขนาดเล็ก

2.4 ไมโครคอนโทรลเลอร์ MCS-51

2.4.1 กล่าวนำ

ไมโครคอนโทรลเลอร์แบบชิพเดียว (Single Chip Microcontroller) คือไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชีพเดียวเหมาะสำหรับควบคุมอุปกรณ์อื่น ๆ โดยอัตโนมัติ โดยผู้ใช้สามารถเขียนโปรแกรมควบคุมได้ตามต้องการ ซึ่งไมโครคอนโทรลเลอร์ชิพเดี่ยวตระกูล 51 หรือ MCS-51 ที่เลือกใช้ในโครงการนี้คือเบอร์ AT89C51 ของบริษัท ATMEL ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีหน่วยความจำโปรแกรม (ROM) ภายในแบบ Flash Memory ขนาด 4 Kbytes ซึ่งเป็นหน่วยความจำที่สามารถเขียนและลบข้อมูลได้ไม่ถึง 1,000 ครั้ง โดยที่ไม่ต้องใช้หน่วยความจำแบบ EPROM ภายนอก และสะดวกต่อการพัฒนาโปรแกรม

คุณสมบัติของไมโครคอนโทรลเลอร์เบอร์ AT89C51

ไมโครคอนโทรลเลอร์ เบอร์ AT89C51 ผลิตโดยบริษัท ATMEL เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ซึ่งมีคุณสมบัติที่สำคัญดังนี้

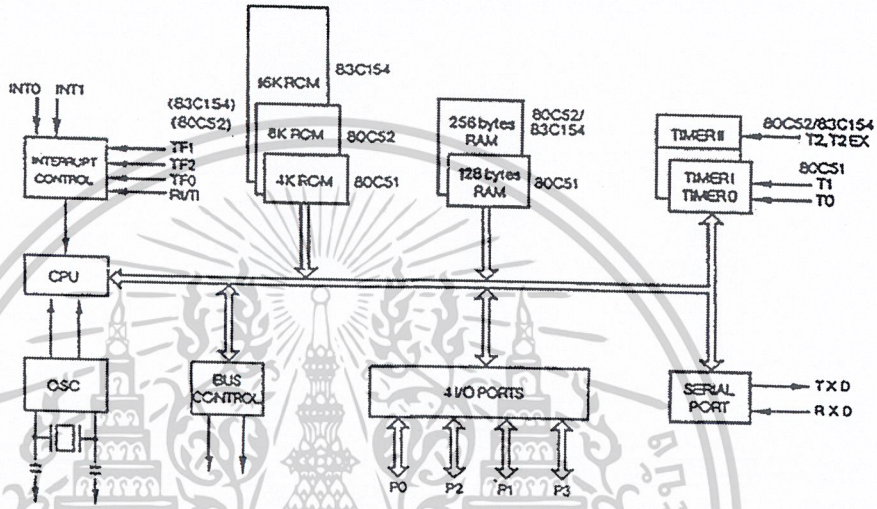
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายใน 4 Kbytes ซึ่งเพียงพอต่อการทำงานของโครงการนี้
- สามารถต่อหน่วยความจำข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรม (RAM) ได้ 64 Kbytes
- สามารถใช้กับหน่วยความจำโปรแกรมภายนอก (Program Memory) ซึ่งเป็นหน่วยความจำซึ่งใช้สำหรับเก็บชุดคำสั่งที่จะทำให้ MCS-51 ทำงานได้สูงสุด
- มีคำสั่งคูณและหารเลขขนาด 8 บิต
- มีการติดต่อสื่อสารข้อมูล แบบอนุกรม (Series) หรือ Universal Asynchronous Receiver Transmitter (UART)
- มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด
- มีการขอขัดจังหวะการทำงานของโปรแกรม (Interrupt)
- ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
- สามารถเลือกการทำงานให้อยู่ในโหมดของ Idle (การทำงานปกติ) และ Power down (การประหยัดพลังงานไฟฟ้า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคุณสมบัติที่กล่าวถึงทำให้ MCS-51 เป็นที่นิยมใช้ในการควบคุมระบบอัตโนมัติ ซึ่งบรรจุไว้ในไอซีวงจรรวมเดี่ยว (Single Chip) ขนาด 40 ขา ดังนั้นจึงสามารถออกแบบให้ระบบมีขนาดเล็กทำให้ ตรวจสอบข้อผิดพลาดได้ง่ายรวมถึงการลดต้นทุนการผลิตหากจะต้องมีการผลิตเป็นจำนวนมาก

2.4.2 โครงสร้างของ MCS-51

โครงสร้างภายในของ MCS-51 ประกอบด้วย เกทประเภทต่าง ๆ เช่น AND OR NOT ซึ่งเกทเหล่านี้ จะถูกออกแบบให้มีหน้าที่การทำงานต่างๆ เช่นวงจรถอดรหัสสร้างสัญญาณนาฬิกาโครงภายในของ MCS-51 จะประกอบด้วยส่วนย่อยๆดังบล็อกไดอะแกรมรูปที่ 2.15



รูปที่ 2.16 แสดงบล็อกไดอะแกรมแสดงโครงสร้างของ MCS-51

จากรูปที่ 2.16 โครงสร้าง MCS–51 ประกอบด้วย 3 ส่วนใหญ่ๆ คือ

1. CPU (Central processing Unit) ซึ่งมีส่วนย่อยอีก แต่ละส่วนมีหน้าที่ต่างๆ กัน ได้แก่ ส่วนสร้างสัญญาณควบคุม (Control Unit) ในการติดต่อกับส่วนอื่นๆ ส่วนประมวลผลทางคณิตศาสตร์ (Arithmetic Unit)

2. หน่วยความจำ (Memory) มีไว้สำหรับจดจำข้อมูล สำหรับนำข้อมูลไปเก็บไว้และอ่านเอาข้อมูลออกมาจากหน่วยความจำ

3. Input /Output Device ทำหน้าที่รับส่งข้อมูลกับอุปกรณ์ภายนอก

2.4.3 การจัดการหน่วยความจำของ MCS-51

หน่วยความจำของ MCS-51 แบ่งออกเป็น 2 แบบ คือ

1. Program Memory เป็นหน่วยความจำสำหรับเก็บโปรแกรมทำหน้าที่เก็บโปรแกรมที่ผู้ใช้เขียนขึ้นเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์โดยหน่วยความจำจะเป็นแบบ ROM มีความจำ 4 Kbyte (ตำแหน่ง 0000H-0FFFH) ในการใช้งาน เราสามารถกำหนดให้ไมโครคอนโทรลเลอร์เลือกใช้โปรแกรมที่เก็บอยู่ภายในตัวไมโครคอนโทรลเลอร์ หรือโปรแกรมที่เก็บอยู่ในหน่วยความจำ (EPROM) ที่อยู่ภายนอกก็ได้ การเลือกการติดต่อทำได้โดยการป้อนสัญญาณควบคุมให้ที่ขา EA (External Access) ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการให้ไมโครคอนโทรลเลอร์ติดต่อกับโปรแกรมที่อยู่ในหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ จะต่อขานี้กับลอจิก 1 หากต้องการให้ไมโครคอนโทรลเลอร์ติดต่อกับโปรแกรมที่เก็บอยู่ในหน่วยความจำภายนอกจะต่อขานี้กับลอจิก 0 การติดต่อกับหน่วยความจำโปรแกรมภายนอกจะติดต่อได้ทั้งหมด 64 Kbyte (ตำแหน่ง 0000H-FFFFH)

ในกรณีที่กำหนดให้ไมโครคอนโทรลเลอร์ติดต่อกับโปรแกรมที่อยู่ในหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์จะติดต่อได้ 4 Kbyte (สำหรับเบอร์ 8051) หากตำแหน่งของโปรแกรม มีค่าเกินกว่าตำแหน่งของหน่วยความจำภายใน (โปรแกรมยาวเกินกว่า 4 Kbyte) ตัวไมโครคอนโทรลเลอร์จะทำการติดต่อกับโปรแกรมที่อยู่ในหน่วยความจำภายนอกอัตโนมัติ

2. Data Memory เป็น หน่วยความจำข้อมูลมีไว้ใช้สำหรับเก็บข้อมูลทั่วไปหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยพื้นฐานแล้วหน่วยความจำข้อมูลจัดเป็นหน่วยความจำแบบ static ดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่เก็บไว้ภายในสูญหาย สำหรับพื้นที่ของหน่วยความจำข้อมูลภายในของ 8051 สามารถมีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็น 2 ลักษณะ ตามตำแหน่งที่ตั้งของหน่วยความจำนั้น คือ หน่วยความจำภายในซึ่งเป็นแรม (RAM) ที่อยู่ภายในตัวไอซีเอง และหน่วยความจำข้อมูลภายนอกซึ่งเป็นการใช้ไอซีหน่วยความจำมาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำเอาไอซีอีพรอมมาต่อเป็นหน่วยความจำโปรแกรม

ในกรณีที่กำหนดให้ไมโครคอนโทรลเลอร์ใช้ที่อยู่ในหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ หน่วยความจำข้อมูลภายในของ 8051 มีจำนวนทั้งหมด 256 ไบต์ โดยสามารถจำแนกออกได้เป็น 2 ลักษณะ คือ พื้นที่เฉพาะสำหรับตัวประมวลผลกลาง ซึ่งจะเรียกว่า รีจีสเตอร์ (Register) และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมที่ผู้ใช้สร้างขึ้นมา หน่วยความจำขนาด 128 ไบต์แรก บริเวณนี้มีตำแหน่งแอดเดรสอยู่ในช่วง 00H-7FH ซึ่งมีการจำแนกออกเป็น 3 ส่วนตามการใช้งาน คือ

บริเวณแอดเดรส 00H-1FH แบ่งออกเป็นกลุ่มข้อมูลจำนวน 8 ไบต์ รวมทั้งหมด 4 กลุ่ม โดยพื้นที่ในแต่ละกลุ่มจะถูกใช้งานในฐานะของรีจีสเตอร์ใช้งานโดยทั่วไป ซึ่งมีชื่อเรียกว่า รีจีสเตอร์ R0-R7

บริเวณแอดเดรส 20H-2FH จำนวน 16 ไบต์ พื้นที่บริเวณนี้เป็นส่วนสำหรับผู้ใช้ซึ่งมีความพิเศษแตกต่างไปจากหน่วยความจำอื่นๆ เนื่องจากผู้ใช้สามารถอ้างถึงหน่วยความจำในบริเวณนี้ได้ในลักษณะไบต์ข้อมูลหรือในลักษณะบิตข้อมูลได้โดยตรง ดังนั้นหากมองในลักษณะบิตข้อมูลแล้วจะมีพื้นที่ตัวแปรบิตให้ใช้ได้มากถึง 128 บิต

บริเวณแอดเดรส 30H-7FH เป็นบริเวณที่ผู้ใช้สามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถอ้างถึงได้ในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

หน่วยความจำขนาด 128 ไบต์ถัดไป เป็นพื้นที่ตั้งแต่บริเวณแอดเดรส 80H-FFH เป็นบริเวณของหน่วยความจำที่มีการใช้งานเฉพาะจาก 8051 เท่านั้น โดยจะนำมาใช้เป็นตำแหน่งของรีจีสเตอร์หน้าที่พิเศษ จำนวน 20 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจีสเตอร์หน้าที่พิเศษ เป็นรีจีสเตอร์สำหรับการควบคุมหน้าที่และการทำงานของพอร์ตทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานในรีจีสเตอร์พิเศษสามารถระบุชื่อหรืออ้างถึงตำแหน่งของรีจีสเตอร์เหล่านั้นได้แก่

- แอกคิวมูเลเตอร์ รีจีสเตอร์ (Accumulator) หรือ ACC เป็นรีจีสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายใน CPU และเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานของ รีจีสเตอร์แอกคิวมูเลเตอร์นี้มีลักษณะการทำงานเช่นเดียวกับแอกคิวมูเลเตอร์ทั่วๆ ไป
- รีจีสเตอร์ B (Register B) เป็นรีจีสเตอร์ที่ใช้สำหรับการทำคำสั่งคูณและหารตัวเลข ในกรณีที่ไม่ใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจีสเตอร์ทั่วๆ ไปได้
- โปรแกรมเคาน์เตอร์ (Program Counter) เป็นรีจีสเตอร์ที่ใช้ชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานภายในโปรแกรมจะเรียกว่า รีจีสเตอร์ PC
- สแต็กพอยน์เตอร์ (Stack pointer) เป็นรีจีสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแต็กสำหรับเก็บข้อมูลของแอกคิวมูเลเตอร์ รีจีสเตอร์ต่างๆรวมทั้งข้อมูลจากโปรแกรม โดยปกติแล้วเมื่อทำการเริ่มต้นระบบใหม่ภายหลังจากการจ่ายไฟฟ้า หรือมีการรีเซตเกิดขึ้น ค่าภายในสแต็กพอยน์เตอร์จะมีค่า 07H ซึ่งเป็นตำแหน่งแอดเดรสรหัสภายในเนื้อที่บริเวณ 128 ไบต์แรกของหน่วยความจำข้อมูลภายใน การใช้งานภายในจะเรียกว่า รีจีสเตอร์ SP
- ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data pointer) เป็นรีจีสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจีสเตอร์ DPTR และสามารถใช้งานแยกเป็นรีจีสเตอร์ขนาด 8 บิต 2 ตัว คือ รีจีสเตอร์ DPH และรีจีสเตอร์ DPL เพื่อเก็บค่าของแอดเดรสของหน่วยความจำที่ต้องใช้ภายในโปรแกรมหรืออาจจะเป็นแอดเดรสของอุปกรณ์ภายนอก ซึ่งกำหนดให้ติดต่อกันโดยใช้ตำแหน่งของหน่วยความจำนั้นภายในโปรแกรม
- โปรแกรมสเตตัสเวิร์ด (PSW) รีจีสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงค์ของรีจีสเตอร์ที่ใช้งานด้วย
- รีจีสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port register) รีจีสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาต์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจีสเตอร์ขนาด 8 บิต สามารถใช้งานได้ในลักษณะของการอินพุต หรือการเอาต์พุตข้อมูลได้ การดำเนินการใดๆที่เกี่ยวข้องกับพอร์ตทั้ง 4 นี้ ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้ พอร์ต P0 และพอร์ต P2 ยังสามารถที่จะใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรสในช่วง 8 บิตล่างของหน่วยความจำ ช่วงเวลาต่อมาจึงนำพอร์ต P0 ไปใช้เป็นบัสในการรับหรือส่งข้อมูลกับอุปกรณ์ภายนอก สำหรับพอร์ต P3 นอกเหนือจากนำไปใช้เป็นพอร์ตอินพุตเอาต์พุตตามปกติแล้วยังนำไปใช้ในฐานะบัสควบคุมเกี่ยวกับการอินเตอร์รัพต์ได้อีกด้วย
- รีจีสเตอร์ SBUF เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการติดต่อสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจีสเตอร์ PCON เป็นรีจีสเตอร์ที่ใช้ในการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโพรเซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราการทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟล็กสภาวะสำหรับการใช้งานทั่วไป (บิต GRO และบิต GRI)

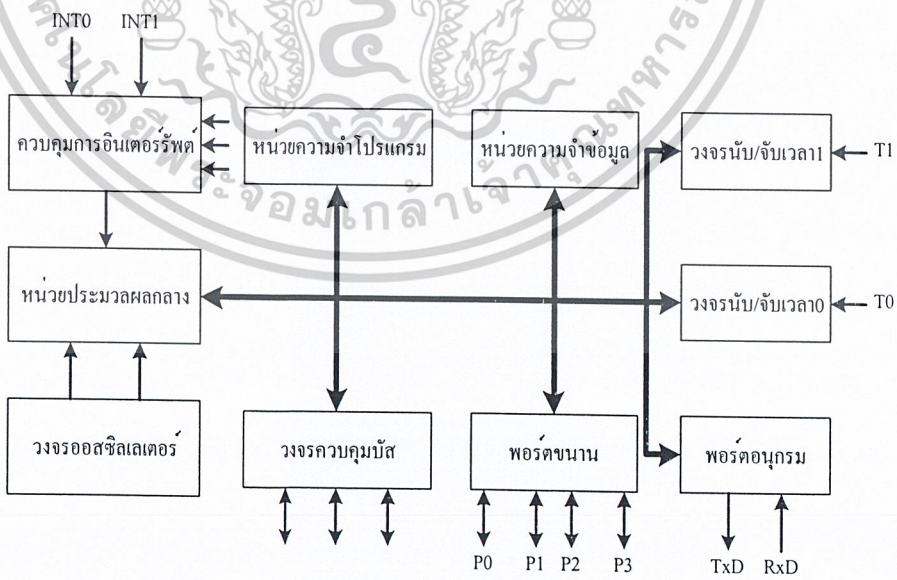
บิต PD (Power down) เป็นการกำหนดให้ลดกำลังไฟฟ้าที่จ่ายให้กับส่วนของโพรเซสเซอร์ภายในลง โดยยังคงมีกำลังไฟฟ้าจ่ายให้กับส่วนหน่วยความจำข้อมูลภายในผ่านทางขาสัญญาณ RST วิธีการนี้มักนำมาใช้ในกรณีที่มีการตรวจสอบการไม่มีกำลังไฟฟ้า (Power failure) โดยวงจรตรวจสอบภายนอกจะต้องมีการอินเตอร์รัพต์เข้ามา เพื่อทำการเก็บข้อมูลที่กำลังประมวลผลอยู่ก่อน และเมื่อมีกระแสไฟฟ้าจ่ายเป็นปกติจึงค่อยนำข้อมูลนั้นมาประมวลผลต่อไป

บิต IDL (Idle mode) เป็นการกำหนดโพรเซสเซอร์ให้หยุดการทำงานชั่วขณะ (Sleep) และจะกลับมาอยู่ในสภาพปกติอีกครั้งเมื่อทำการรีเซตทางฮาร์ดแวร์ หรือมีการอินเตอร์รัพต์อย่างใดอย่างหนึ่งเกิดขึ้น การทำงานในลักษณะนี้สามารถเกิดขึ้นได้เนื่องจากว่าสภาวะการหยุดทำงานชั่วขณะนั้นเป็นเพียงเพื่อห้ามไม่ให้มีสัญญาณนาฬิกาจ่ายให้ส่วนของโพรเซสเซอร์เท่านั้น ส่วนของวงจรการอินเตอร์รัพต์พอร์ตอนุกรมและวงจรจับเวลา ยังคงมีสัญญาณนาฬิกาอยู่เป็นปกติ

- รีจีสเตอร์ IP, IE, TMOD, SCON เป็นกลุ่มของรีจีสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของการอินเตอร์รัพต์ต่างๆ ของ 8051

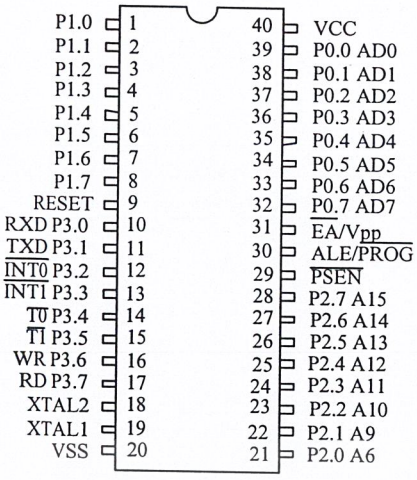
2.4.4 สถาปัตยกรรมของ MCS-51

สถาปัตยกรรมภายในของ MCS-51 ซึ่งจะอธิบายถึงส่วนย่อยๆ ภายใน MCS-51 เพียงชิพเดียว ดังรูปที่ 2.17 และสัญญาณจากภายใน จะต่อออกมาสู่ภายนอกทางขา (Pin) ต่างๆ ของ MCS-51 ที่มีอยู่ 40 ขา ดังรูปที่ 2.18



รูปที่ 2.17 แสดงสถาปัตยกรรมภายในของ MCS-51

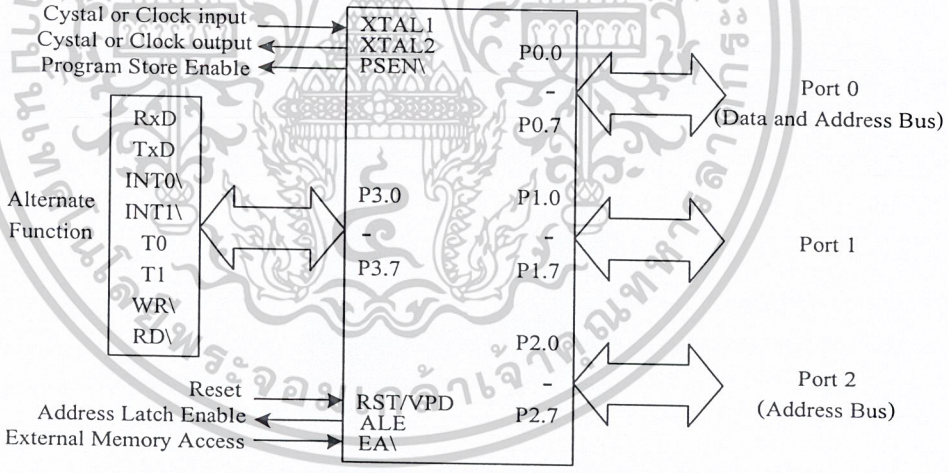
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงตำแหน่งขาของ MCS-51

สัญญาณต่างๆ ของไมโครคอนโทรลเลอร์ MSC-51

ไมโครคอนโทรลเลอร์ในตระกูล MSC-51 เป็นไอซีขนาด 40 ขาซึ่งมีสัญญาณต่างๆแสดงในรูปที่ 2.19



รูปที่ 2.19 แสดงสัญญาณต่างๆ ของ MSC-51 เบอร์ 8051

สัญญาณต่างๆ ของไมโครคอนโทรลเลอร์ MSC-51 สามารถจำแนกตามการทำงานเป็น 3 กลุ่มคือ

- กลุ่มสัญญาณตำแหน่ง เป็นตัวกำหนดตำแหน่งของหน่วยความจำ
- กลุ่มสัญญาณควบคุม เป็นสัญญาณควบคุมการทำงานของไมโครคอนโทรลเลอร์
- กลุ่มสัญญาณข้อมูลเป็นทางผ่านของข้อมูลระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่และการใช้งานของสัญญาณต่างๆ เป็นดังนี้

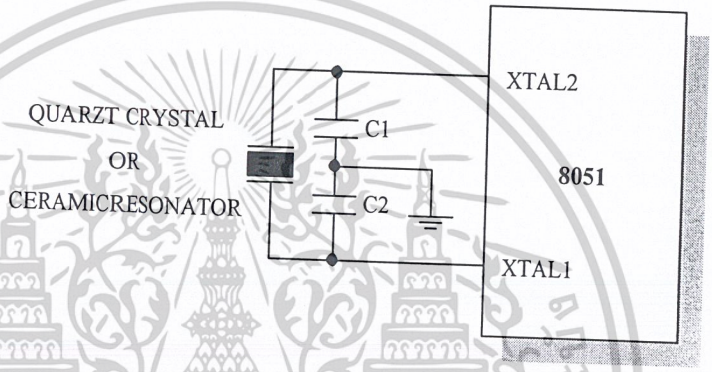
- VCC สำหรับต่อกับไฟเลี้ยง 5 โวลต์

- VSS สำหรับต่อกับกราวนด์

- XTAL1เป็นอินพุตของภาคขยายสัญญาณแบบอินเวิร์ตของวงจรผลิตสัญญาณนาฬิกา

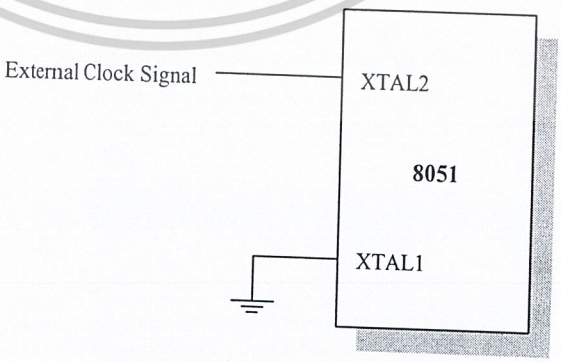
- XTAL2เป็นเอาต์พุตของภาคขยายสัญญาณแบบอินเวิร์ตของวงจรผลิตสัญญาณนาฬิกา

การต่อใช้งานของขา XTAL1 และ XTAL2 เพื่อสร้างวงจรผลิตสัญญาณให้กับ ไมโครคอนโทรลเลอร์โดยใช้คริสตอลสามารถทำได้ดังรูปที่ 2.20



รูปที่ 2.20 แสดงการต่อสัญญาณนาฬิกาที่ขา XTAL1 และ XTAL2

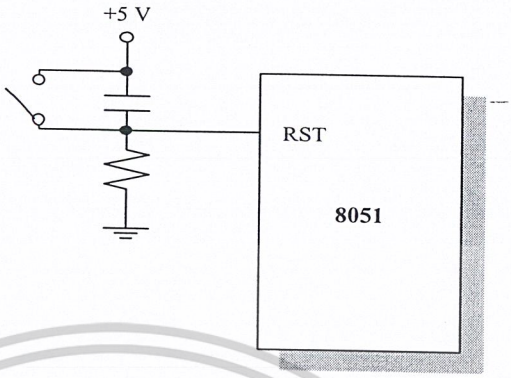
หากต้องการใช้สัญญาณนาฬิกาจากภายนอกจะต่อ XTAL1 ลงกราวนด์และต่อสัญญาณนาฬิกาจากภายนอกเข้าที่ขา XTAL2 ดังรูปที่ 2.21



รูปที่ 2.21 แสดงการใช้สัญญาณนาฬิกาจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RST สัญญาณรีเซต ไมโครคอนโทรลเลอร์จะถูกรีเซตเมื่อสัญญาณที่ขานี้มีค่าเป็นลอจิก 1 นานไม่ต่ำกว่า 2 แมชชีนไซเกิล การต่อขารีเซตจะเป็นดังรูปที่ 2.22



รูปที่ 2.22 แสดงการต่อสัญญาณรีเซต

- ALE/PROG (Address Latch Enable) เป็นสัญญาณเอาต์พุต ซึ่งไมโครคอนโทรลเลอร์จะส่งออกไปเป็นพัลส์เพื่อแลทช์ค่าตำแหน่งไบต์ต่ำที่อยู่ที่พอร์ต P0 ในขณะติดต่อกับหน่วยความจำภายนอก สัญญาณนี้จะถูกส่งออกไปด้วยอัตราคงที่คือ 1/6 เท่าของความถี่สัญญาณนาฬิกาซึ่งสามารถนำไปใช้เป็นสัญญาณนาฬิกาให้กับอุปกรณ์ภายนอกได้ สัญญาณพัลส์นี้จะถูกข้ามไป 1 พัลส์เมื่อมีการติดต่อกับหน่วยความจำข้อมูลภายนอก (External Data Memory) และสัญญาณนี้จะใช้เป็นอินพุตเพื่อควบคุมการโปรแกรม PROM ภายในตัวไมโครคอนโทรลเลอร์ด้วย

- PSEN (Program Store Enable) เป็นเอาต์พุต สำหรับส่งสัญญาณสโตรป (พัลส์ต่ำ) เพื่ออ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก (External Program Memory) เมื่อซีพียูอ่านรหัสคำสั่งจากหน่วยความจำภายนอก จะส่งสัญญาณสโตรปออกมา 2 ครั้งใน 1 แมชชีนไซเกิลแต่สัญญาณสโตรปทั้ง 2 ครั้งจะถูกข้ามไปหากเป็นช่วงที่ซีพียูติดต่อกับ External Data Memory

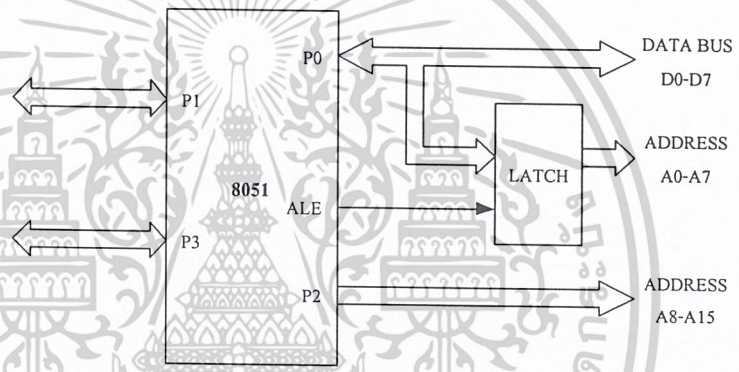
- EA (External Access) เป็นสัญญาณอินพุต ใช้สำหรับควบคุมให้ไมโครคอนโทรลเลอร์เลือกติดต่อกับโปรแกรมที่เก็บอยู่ในหน่วยความจำโปรแกรมภายใน หรือโปรแกรมที่เก็บอยู่ในหน่วยความจำภายนอกตัวไมโครคอนโทรลเลอร์ หากให้ค่าลอจิก 1 ที่ขานี้จะเป็นการเลือกใช้หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ หากต้องการให้ซีพียูติดต่อกับหน่วยความจำโปรแกรมภายนอกต้องต่อสัญญาณเข้ากับลอจิก 0 หรือ VSS ถึงแม้ว่าเบอร์ 8031 ไม่มี EPROM ภายในก็ต้องต่อขานี้ลงกราวด์ด้วย ในกรณีของการโปรแกรม ROM ภายในไมโครคอนโทรลเลอร์จะต่อขานี้เข้ากับไฟ 21 V ถ้าเป็น 8751 AH แต่หากเป็น 8751 BH ต้องต่อกับ 12.75 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PORT 0 เป็นพอร์ตอินพุตเอาต์พุต 2 ทางแบบ Open drain ขนาด 8 บิต P0.1-P0.7 เมื่อใช้เป็นเอาต์พุตสามารถต่อกับไอซี TTL ตระกูล LS ได้ 8 ตัว เมื่อต้องการใช้งานเป็นอินพุตต้องส่งค่าลอจิก 1 ออกไปที่พอร์ตก่อนเพื่อทำให้ลอยซึ่งจะเป็นอิมพีแดนซ์สูง

พอร์ต P0 จะทำงานอีกหน้าที่หนึ่งคือ เป็นมัลติเพล็กซ์ของสัญญาณตำแหน่งด้านต่ำ และสัญญาณข้อมูลในการติดต่อกับหน่วยความจำภายนอก การทำงานในลักษณะนี้จะใช้การพูลอัพ (Pull up) จากภายในที่สามารถจ่ายกระแสให้กับอินพุตของ TTL ได้ 8 ตัว นอกจาก 2 หน้าที่ดังกล่าวแล้ว พอร์ต P0 ยังใช้เป็นตัวรับข้อมูลในช่วงการโปรแกรม EPROM และเป็นตัวส่งข้อมูลออกมาในช่วงการตรวจสอบโปรแกรมภายใน ROM หรือ EPROM ซึ่งจะต้องใช้พูลอัพจากภายนอกในขณะทำการตรวจสอบโปรแกรม

การสร้างสัญญาณตำแหน่ง A0-A7 และสัญญาณข้อมูล D0-D7 ทำโดยใช้อุปกรณ์แลทช์ข้อมูลดังแสดงในรูปที่ 2.23



รูปที่ 2.23 แสดงการสร้างสัญญาณแอดเดรสและสัญญาณข้อมูล

- PORT 1 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุตจะต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้เป็นอินพุตต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปที่พอร์ตนี้ก่อน เพื่อทำให้เกิดพูลอัพภายใน เมื่อสัญญาณอินพุตเป็น 0 เข้ามาจะทำให้ พอร์ตจ่ายกระแสออกเนื่องจากมีการพูลอัพอยู่ภายใน นอกจากนี้พอร์ต P1 ยังทำหน้าที่รับตำแหน่งด้านต่ำในช่วงของการโปรแกรม EPROM และช่วงการตรวจสอบโปรแกรมใน ROM หรือ EPROM อีกด้วย สำหรับเบอร์ 8032 AH และ 8052 AH ขา P1.0 และ P1.1 จะทำหน้าที่เป็น T2 และ T2EX อีกหนึ่งหน้าที่

- PORT 2 เป็นพอร์ตอินพุต/เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุตจะต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุต ต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปที่พอร์ตก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุตเข้ามาเป็น 0 จะทำให้พอร์ต P2 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน ในระหว่างการติดต่อกับโปรแกรมภายนอกหรือการติดต่อกับหน่วยความจำข้อมูลภายนอกที่มีการอ้างตำแหน่งแบบ 16 บิต พอร์ต P2 จะส่งตำแหน่งไบต์สูงออกไป ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในลักษณะนี้จะมีการพูลอัพภายในอยู่ ในช่วงของการติดต่อกับหน่วยความจำข้อมูลภายนอกที่ใช้การอ้างตำแหน่งแบบ 8 บิต (คำสั่ง MOVX @Ri) สัญญาณที่ขาของพอร์ต P2 จะมีค่าเท่ากับรีจีสเตอร์ P2 ที่อยู่ใน SFR นอกจากนี้พอร์ต P2 ยังทำหน้าที่รับตำแหน่งไบต์สูง ในช่วงของการโปรแกรม EPROM และการตรวจสอบโปรแกรมใน ROM และ EPROM อีกด้วย

- PORT 3 เป็นพอร์ตอินพุต/เอาต์พุต 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุตสามารถต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุตเข้ามาเป็น 0 จะทำให้พอร์ต P3 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน นอกจากนี้ พอร์ต P3 ยังทำหน้าที่เป็นสัญญาณอื่นๆ อีกดังนี้

พอร์ต	สัญญาณ	หน้าที่
P3.0	RXD	อินพุตของพอร์ตอนุกรม
P3.1	TXD	เอาต์พุตของพอร์ตอนุกรม
P3.2	INT0	สัญญาณอินเตอร์รัพต์จากภายนอกตัวที่ 0
P3.3	INT	สัญญาณอินเตอร์รัพต์จากภายนอกตัวที่ 1
P3.4	T0	อินพุตจากภายนอกของตัวตั้งเวลา 0
P3.5	T1	อินพุตจากภายนอกของตัวตั้งเวลา 1
P3.6	WR	สัญญาณการเขียนข้อมูลออกไปภายนอก
P3.7	RD	สัญญาณการอ่านข้อมูลจากภายนอกเข้ามา

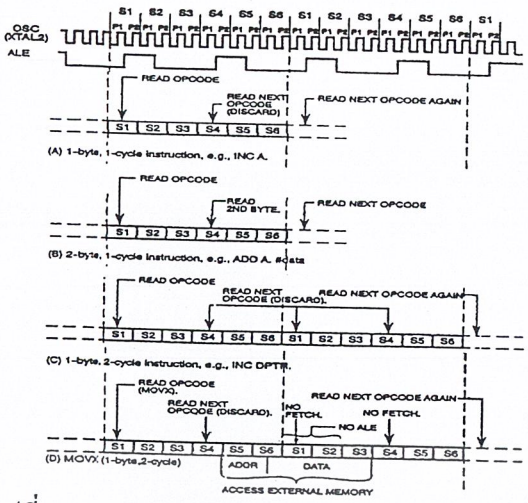
ตารางที่ 2.2 แสดงสัญญาณต่างๆ ของพอร์ต P3

เมื่อต้องการใช้งานพอร์ต P3 ให้ทำหน้าที่เป็นสัญญาณต่างๆ จะต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปแลทช์ที่พอร์ต P3 ก่อนเพื่อให้เกิดการพูลอัพภายใน หากเรากำหนดให้มีค่าลอจิก 0 จะทำให้สัญญาณที่ขาต่างๆ มีค่าเป็น 0 ตลอดเวลา

2.4.5 การทำงานของ MCS-51

การทำงานของ MCS-51 ผู้ใช้จะต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลขฐาน 2 ที่เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งอาจประกอบด้วย 1,2 หรือ 3 ไบต์ ก็ได้ มากระทำตามคำสั่งนั้นโดยจะเริ่มจากการทำงานภายใน MCS-51 เองแล้วช่วงต่อไปจะเป็นช่วงการทำงานตามคำสั่ง (Execute Cycle) ซึ่งการทำงานดังกล่าว จะขึ้นอยู่กับสัญญาณควบคุมที่สร้างจากวงจร Oscillator ทำให้การทำงานดังกล่าว จะขึ้นอยู่กับสัญญาณควบคุมที่สร้างจากวงจร Oscillator ทำให้การทำงานต่างๆ เป็นไปตามลำดับ ที่ผู้ผลิตได้ออกแบบไว้ดังรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 แสดงลำดับสถานะการทำงานใน MCS-51

จากรูปที่ 2.24 ในหนึ่งแมชชีนไซเคิล คือ ช่วงเวลาตั้งแต่ S1 ถึง S6 ซึ่งจะใช้เวลา 12 คาบของสัญญาณออสซิลเลเตอร์ หากใช้ออสซิลเลเตอร์ความถี่ 12 MHz จะได้ว่า 1 แมชชีนไซเคิลใช้เวลา 1 ไมโครวินาที ดังนั้นการทำงานใน 1 คำสั่ง ต่ำสุดจะกินเวลาเพียง 1 ไมโครวินาที รูปที่ 2.24 (A) แสดงการทำงานของคำสั่ง INCA ซึ่งเป็นคำสั่ง 1 ไบต์ ทำงานเสร็จภายใน 1 แมชชีนไซเคิล รูปที่ 2.24 (B) แสดงการทำงานของคำสั่ง ADDA # data ซึ่งเป็นคำสั่ง 2 ไบต์ แต่ทำงานเสร็จภายใน 1 แมชชีนไซเคิล รูปที่ 2.24 (C) แสดงการทำงานของคำสั่ง INC DPTR 1 ไบต์ แต่ทำงานเสร็จภายใน 2 แมชชีนไซเคิล รูปที่ 2.24 (D) แสดงการทำงานของคำสั่ง MOVX ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จภายใน 2 แมชชีนไซเคิล

2.4.6 การอินเตอร์รัปต์

การอินเตอร์รัปต์ของ MCS-51 สามารถรับสัญญาณอินเตอร์รัปต์ที่เกิดขึ้นได้อย่างน้อย 5 ชนิด ได้แก่

- อินเตอร์รัปต์จากภายนอก 0 (External Interrupt 0)
- อินเตอร์รัปต์จากภายนอก 1 (External Interrupt 1)
- อินเตอร์รัปต์ของไทม์เมอร์ 0 (Timer Flag Interrupt 0)
- อินเตอร์รัปต์ของไทม์เมอร์ 1 (Timer Flag Interrupt 1)
- อินเตอร์รัปต์ของพอร์ตสื่อสารอนุกรม (Serial Port Interrupt)

การอินเตอร์รัปต์แต่ละชนิดที่ MCS-51 สามารถรับได้แบ่งกลุ่มดังนี้

1. External Interrupt เป็นอินเตอร์รัปต์ที่เกิดขึ้นจากภายนอก MCS-51 มี 2 ชนิด คือ External Interrupt 0 และ External Interrupt 1 โดยการต่อขา 12 (INT 0) และขา 12 (INT 1) ตามลำดับซึ่งสัญญาณอินเตอร์รัปต์ชนิดนี้ จะทำงานเพื่อสัญญาณนอกที่ส่งเข้ามามีสถานะลอจิกเป็น 0

2. Timer Flag Interrupt อินเตอร์รัปต์ของกลุ่มนี้ ประกอบด้วย Timer Flag Interrupt 0 และ Timer Flag Interrupt 1 เป็นอินเตอร์รัปต์ที่เกิดขึ้นจากภายในตัวเอง จะเกิดขึ้นโดยบิต TFO หรือ TFI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกเซตเมื่อ ไทม์เมอร์ 0 หรือ ไทม์เมอร์ 1 เกิด overflow ขึ้นทำให้เกิดสัญญาณอินเตอร์รัปต์เข้าไปขัดจังหวะการทำงานของโปรแกรม ที่ทำงานอยู่ขณะนั้น ให้ทำงานอื่นแทน

3. Serial port Interrupt เป็นอินเตอร์รัปต์ที่เกิดขึ้นจากภายในตัวชิพเองสัญญาณอินเตอร์รัปต์ที่เกิดขึ้น ได้มาจากบิต TI หรือ RI บิตที่ควบคุมการอินเตอร์รัปต์ ทั้งสองนี้จะไม่ถูกเคลียร์โดยฮาร์ดแวร์ใน MCS-51 เมื่อซีพียูไปทำงานในโปรแกรมบริการอินเตอร์รัปต์เพราะการเกิดอินเตอร์รัปต์ของพอร์ตสื่อสารอนุกรม อาจเกิดจากบิต RI หรือ TI ก็ได้ ดังนั้นโปรแกรมในส่วนอินเตอร์รัปต์จะต้องตรวจสอบเองว่าสัญญาณอินเตอร์รัปต์ที่เกิดขึ้นได้มาจากบิด RI หรือ TI ทั้งสองอย่างจะถูกเคลียร์โดยซอฟท์แวร์เท่านั้น

การทำงานเกี่ยวกับการอินเตอร์รัปต์นั้น จะมี Special Function Register ที่เกี่ยวข้องโดยตรง 2 ตัว ได้แก่ IE (Interrupt Enable) และ IP (Interrupt Priority) รีจีสเตอร์ IEสามารถควบคุมการอินเตอร์รัปต์ใน MCS-51 ได้ทั้งหมดโดยใช้บิต EA ให้มีค่าเป็น 0 สัญญาณอินเตอร์รัปต์ทุกชนิด ที่เกิดขึ้นจะไม่สามารถอินเตอร์รัปต์ MCS-51 ได้สำหรับรีจีสเตอร์ IP นั้นเราสามารถจัดลำดับความสำคัญของการทำอินเตอร์รัปต์ได้โดยเซทบิดเป็น 1 สำหรับอินเตอร์รัปต์ที่มีความสำคัญสูง และเซตบิดให้เป็น 1 สำหรับอินเตอร์รัปต์ ที่มีความสำคัญต่ำ แต่ถ้าไม่ได้จัดลำดับความสำคัญการทำอินเตอร์รัปต์จะเป็นไปตามตารางที่ 2.3

INTERRUPT	PRIORITY LEVEL
1. IE0	High Priority
2. TF0	↓
3. IE1	↓
4. TF1	↓
5. Serial	Low Priority

ตารางที่ 2.3 แสดงลำดับความสำคัญของการทำอินเตอร์รัปต์

สำหรับการทำอินเตอร์รัปต์นั้น ตัว MCS-51 ได้กำหนดแอดแดรสหน่วยความจำของโปรแกรมบริการ ของแต่ละอินเตอร์รัปต์ ดังตารางที่ 2.4

INTERRUPT	ADDRESS (HEX)
1. IE0	0003
2. TF0	000B
3. IE1	0013
4. TF1	001B
5. Serial	0023

ตารางที่ 2.4 แสดงตำแหน่งของโปรแกรมอินเตอร์รัปต์แต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรีเซต (Reset)

การรีเซต สามารถมองเป็นการอินเตอร์รัพต์สูงสุด เพราะโปรแกรมไม่สามารถสกัดกั้นได้ การอินเตอร์รัพต์ชนิดนี้บ่อยครั้งเราเรียกว่า Nonmaskable เนื่องจากไม่มีบิตตัวใดที่สามารถหยุดมันได้ ซึ่งไม่เหมือนกับการอินเตอร์รัพต์แบบอื่น PC จะไม่เก็บค่าโปรแกรมครั้งหลังสุด รีเซตเป็นคำสั่งที่สมบูรณ์ในการกระโดดไปที่ 0000H และเริ่มทำงานที่จุดนี้

เมื่อลอจิก 1 ใช้ที่ขา RST 8051 จะอยู่ในสภาวะรีเซต หลังจากขา RST เป็น 0 รีจีสเตอร์ภายในจะมีค่าดังต่อไปนี้

REGISTER	VALUE HEX
PC	0000
DPTR	0000
A	00
B	00
SP	07
SPW	00
P0-3	FF
IP	XXX00000B
IE	0XX00000B
TCON	00
TMOD	00
TH0	00
TL0	00
TH1	00
TL1	00
SCON	00
SBUF	XX
PCON	0XXXXXXXB

ตารางที่ 2.5 แสดงค่าภายในรีจีสเตอร์หลังจากการรีเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ภาคแปลงรหัสสัญญาณความถี่คู่ (DTMF Decoder)

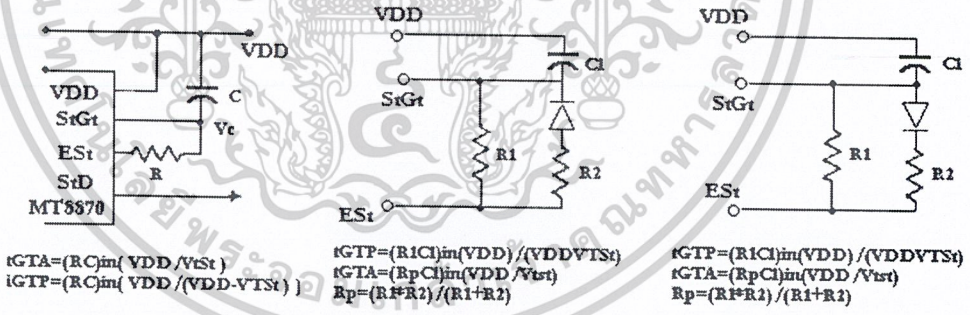
2.5.1 บทนำ

สำหรับการถอดรหัสสัญญาณโทรศัพท์ เราเลือกใช้ไอซีสำเร็จรูปเบอร์ MT 8870 ในการถอดรหัสสัญญาณ DTMF เนื่องจากเป็นไอซีที่หาได้ง่าย และมีประสิทธิภาพดี เป็นที่นิยมใช้แพร่หลาย

รายละเอียดของไอซีเบอร์ MT 8870 :

- คุณสมบัติของ MT8870 เป็นตัวรับและถอดรหัสความถี่ DTMF (DTMF Receiver & Decoder)
  - เป็นตัวรับและถอดรหัสความถี่โทรศัพท์
  - กินไฟน้อยใช้ไฟเลี้ยงระดับเดียวกับ TTL
  - สามารถปรับการ์ดไทม์ (Guard Time) ได้
  - สามารถตั้งอัตราขยายภายในตัวไอซีได้
  - เป็นไอซีคุณภาพสูง

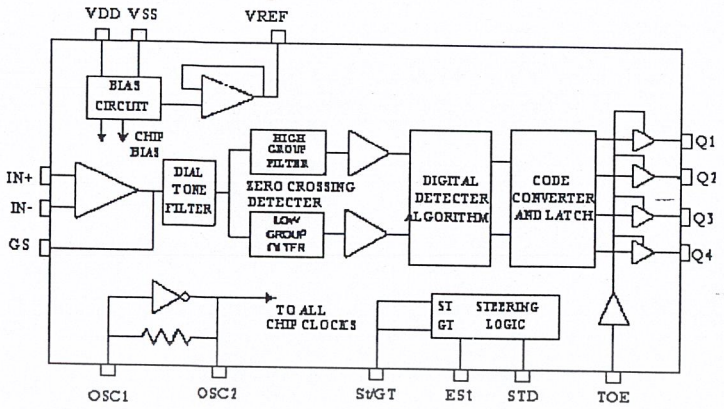
สำหรับการ์ดไทม์นั้น หมายถึง คาบเวลาของความถี่เข้ามา ซึ่งจะต้องนานเท่ากันหรือมากกว่าช่วงเวลาที่เราตั้งไว้ จึงจะได้รับการยอมรับว่า สัญญาณที่ได้รับมาถูกต้อง หรือว่าเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั่นเอง เมื่อสัญญาณที่เข้ามานานเท่ากันหรือมากกว่าเวลาที่ตั้งไว้ จึงสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่ที่เข้ามาสั้นกว่าเวลาที่ตั้งที่ไว้ ก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลาและการคำนวณดูได้จากรูปที่ 2.25



รูปที่ 2.25 แสดงวงจรตรวจสัญญาณอย่างง่ายและแสดงการกำหนดเวลาการ์ดไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โครงสร้างของ MT8870



รูปที่ 2.26 แสดงโครงสร้างภายในของ MT8870

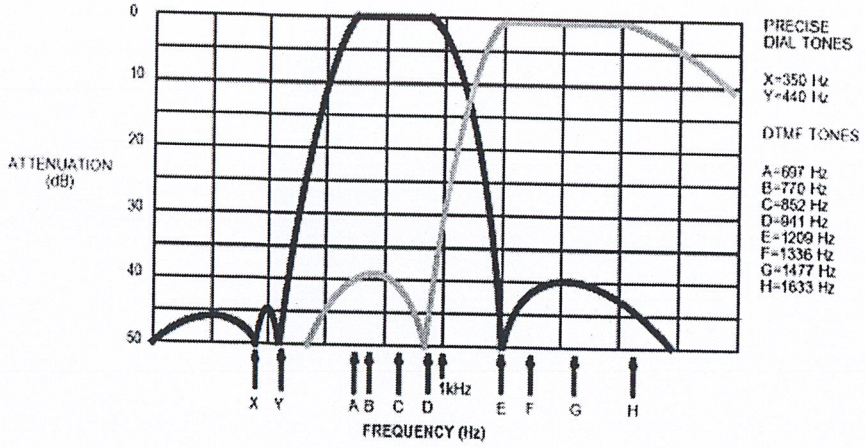
โครงสร้างภายในของ MT8870 ประกอบไปด้วย วงจรความถี่และวงจรถอดรหัสฟังก์ชั่นทางดิจิตอลเป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO^2-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิทซ์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอลเพื่อตรวจจับ และถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐาน 2 ขนาด 4 บิตและเช็คช่วงเวลาที่สัญญาณเข้ามาส่วนภาคอินพุตเป็นแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่อกับอุปกรณ์ภายนอกเอาท์พุตเป็นวงจรแลทช์ 3 สถานะ ภายในโครงสร้างของ MT8870 จะประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

- ภาคกรองสัญญาณความถี่ (Filter Section)
- ภาคถอดรหัส (Decoder Section)
- ภาคตรวจสอบสัญญาณ (Steering Circuit)
- ภาคกำเนิดความถี่ (Oscillator)
- ภาคขยายสัญญาณความแตกต่าง (Differential Input)

2.5.2 ภาคกรองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือช่วงความถี่สูง และช่วงความถี่ต่ำ โดยใช้วงจรกรองความถี่ผ่านอันดับ 6 ชนิดสวิทซ์คาปาซิเตอร์ (Six-order Switched Capacitor Band Pass Fitter) ซึ่งความถี่แยกได้มี 2 ช่วง คือช่วงความถี่สูง และช่วงความถี่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 แสดงภาคกรองสัญญาณความถี่

2.5.3 ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้ว จะผ่านเข้ามาวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิตอลและมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่ในช่วงที่ DTMF ต้องการหรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสาน เพื่อตรวจสอบว่าความถี่นั้นถูกต้องสัญญาณที่ขา Est (Early steering) หรือ ขา 16 ก็จะแอกทีฟ

2.5.4 ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาท์พุท จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มโทรศัพท์ให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรส่วนนี้ จะได้รับโดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอกสัญญาณขา Est จะเป็น “Hing” นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามาทำให้ VC สูงขึ้นตัวเก็บประจุ C จะคายประจุทำให้แรงดัน VC สูงขึ้นเท่ากับค่าเทรชโฮลต์ วงจรถอดรหัสจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต รายละเอียดในการทำงานดูได้จากแผนภูมิเวลาหรือไทม์มิ่งไดอะแกรม (Timing Diagram) จะเข้าใจได้ง่ายสำหรับการ์ดไทม์ (Guard Time) นั้นหมายถึงช่วงคาบเวลาของความถี่เข้ามาซึ่งจะต้องนานเท่ากับหรือนานกว่าช่วงเวลาที่เราตั้งไว้ จึงจะได้รับการยอมรับสัญญาณความถี่นั้นถูกต้อง หรือพูดได้ว่าเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั่นเอง ถ้าสัญญาณเข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป

2.5.5 ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุทของ MT8870 เป็นภาคขยายออปแอมป์ ที่สามารถปรับตัวอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้าไปดังรูปที่ 2.28 ซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุต และอิมพีแดนซ์ได้ดังนี้

$$R_3 = \frac{R_2 \times R_5}{R_2 + R_5}$$

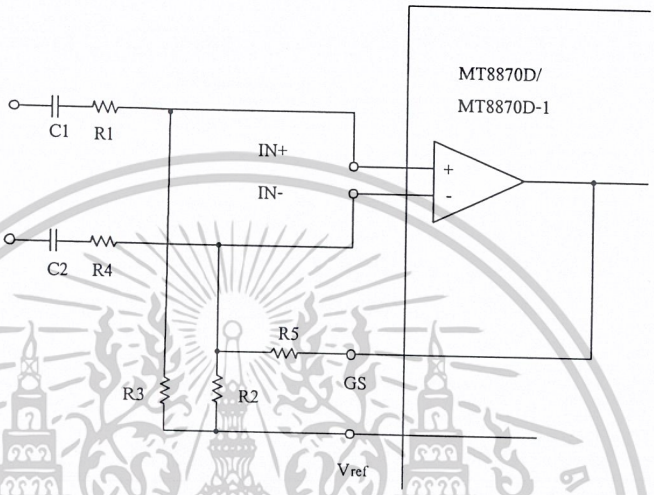
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราขยาย

$$A_{V_{dif}} = \frac{R_5}{R_1}$$

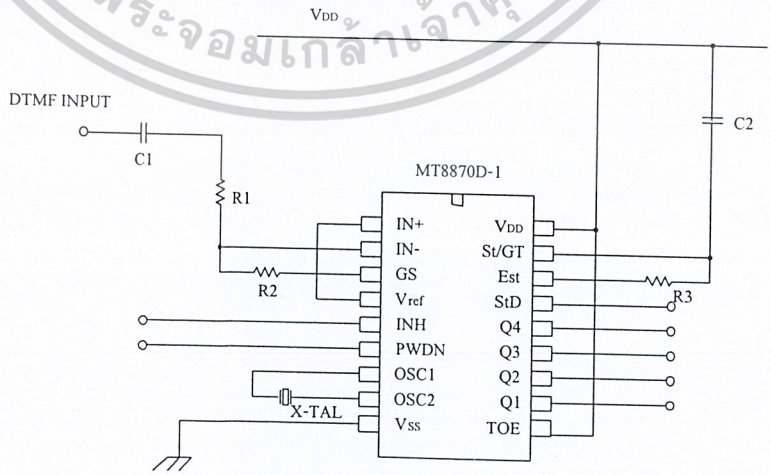
อินพุทอิมพีแดนซ์

$$\left(Z_{I_{dif}} = 2\sqrt{R_1^2 + \left(\frac{1}{wc}\right)^2} \right)$$



รูปที่ 2.28 แสดงการต่อวงจรภาคอินพุต

สำหรับวงจรการต่อใช้งานก็กำหนดค่าต่างๆ มาโดยเฉพาะการทำงานสำหรับวงจรการถอดรหัส DTMF นั้นเมื่อผู้เรียกกดหมายเลข หลังจากได้รับการตอบรับเรียบร้อยสัญญาณความถี่สองความถี่ที่ผสมกันมา จะเข้าสู่วงจรถอดรหัส DTMF เบอร์ MT8870 ถอดรหัสแล้วจะได้เอาพุทออกมาเป็น 4 บิต แล้วส่งไปส่วนควบคุมต่อไปดังรูปที่ 2.29



รูปที่ 2.29 แสดงการวงจรถอดรหัสความถี่ DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.6 ขั้นตอนการทำงาน

เพื่อให้ทราบถึงรายละเอียดต่าง ๆ ในการทำงานและหน้าที่ของเขาต่างๆ ของ MT8870 ในการต่อเชื่อมเข้าเป็นระบบเชื่อมโยงกับส่วนอื่น ๆ สามารถที่จะอธิบายด้วยแผนภูมิเวลาข้างล่างนี้

A – ตรวจพบว่าความถี่เข้ามา แต่คาบเวลาไม่ถูกต้องเอาท์พุทไม่เปลี่ยน

B – ความถี่ #n ถูกตรวจพบและมีคาบเวลาไม่ถูกต้องความถี่ถูกถอดรหัส และแลทซ์ไว้ที่เอาท์พุท

C – ช่วงความถี่ #n ช่วงห่างถูกต้องเอาท์พุทยังคงแลทซ์อยู่ จนกว่าจะได้รับความถี่ที่ถูกต้อง

D - เอาท์พุทเปลี่ยนเป็น High Impedance

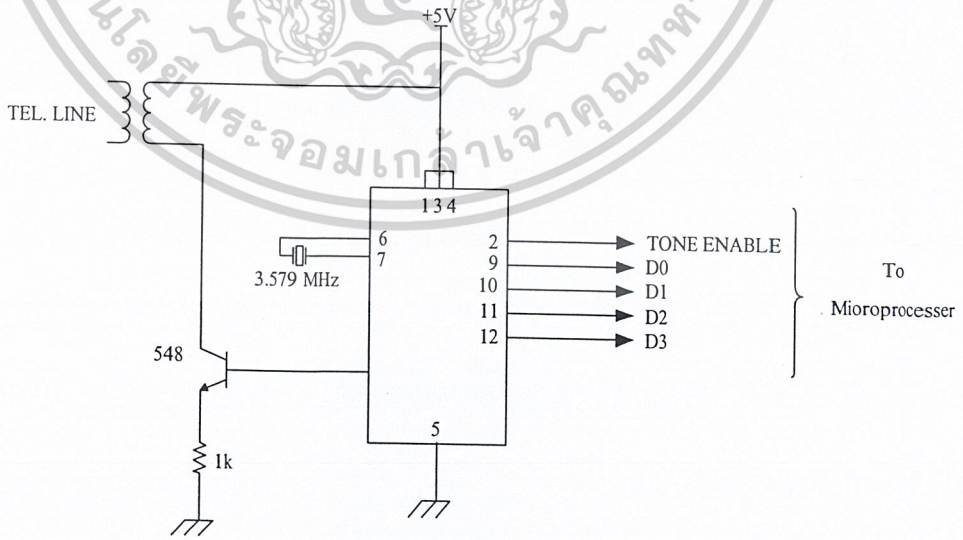
E – ความถี่ #n +1 ถูกตรวจพบคาบเวลาถูกถอดรหัสและแลทซ์ไว้

F - ความถี่ #n +1 หายไปช่วงห่างไม่ถูกต้อง เอาท์พุทยังแลทซ์อยู่

G - จบความถี่ #n +1 ช่วงห่างถูกต้องเอาท์พุทยังแลทซ์อยู่จนถึงความถี่ใหม่ที่ต้องเข้ามา

2.6 วงจรกำเนิดสัญญาณคู่ความถี่ (Dial DTMF)

ไอซีที่นำใช้ทำหน้าที่กำเนิดสัญญาณคู่ความถี่ใช้ไอซีเบอร์ TP5088 ที่สามารถนำมาประยุกต์ใช้ในงานโทรศัพท์ที่ถูกควบคุมด้วยไมโครโปรเซสเซอร์ โดยไม่จำเป็นต้องผ่านคีย์บอร์ด ในช่วงที่ขาโทนเอนนาเบิล (TONE ENABLE) อยู่ในสภาวะแรงดันไฟต่ำ จะทำให้วงจรออสซิลเลเตอร์ไม่ทำงานและจะไม่รับข้อมูลอินพุทที่เข้ามา ขณะที่มีการเปลี่ยนแปลงสภาวะจากแรงดันไฟต่ำไปแรงดันไฟสูงที่ขาโทนเอนนาเบิล ข้อมูลจะถูกป้อนเข้าไปในไอซีและสัญญาณคู่ความถี่ จะถูกสร้างจากความถี่ DTMF มาตรฐาน ซึ่งจะมีกลุ่มความถี่ต่ำและกลุ่มความถี่สูง



รูปที่ 2.30 แสดงวงจรกำเนิดสัญญาณคู่ความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำเนิดสัญญาณคู่ความถี่ใช้ไมโครโปรเซสเซอร์ควบคุมทางขาอินพุทและขาโทนเอนนาเบิล โดยจะป้อนข้อมูลเข้าทางขา D0-D3 ก่อน แต่ว่า TP5088 จะยังไม่ทำงานและจะไม่รับข้อมูลอินพุทที่เข้ามา

เพราะขาโทนเอนนาเบิลอยู่ในสภาวะแรงดันไฟต่ำ จากนั้นจะเปลี่ยนแปลงสภาวะจากแรงดันไฟต่ำไปแรงดันไฟสูงที่ขาโทนเอนนาเบิล เพื่อให้มีการรับข้อมูลทางขา D0-D3 และสร้างสัญญาณคู่ความถี่ออกไปตามข้อมูลอินพุท D0-D3 ตามตาราง

หมายเลข	Low Frequency	High Frequency	D3	D2	D1	D0
1	697	1209	0	0	0	1
2	697	1336	0	0	1	0
3	697	1477	0	0	1	1
4	770	1209	0	1	0	0
5	770	1336	0	1	0	1
6	770	1477	0	1	1	0
7	852	1209	0	1	1	1
8	852	1336	1	0	0	0
9	852	1477	1	0	0	1
0	941	1336	1	0	1	0

ตารางที่ 2.6 แสดงสัญญาณ DTMF ที่ได้จาก ไอซีเบอร์ 5088
เมื่อป้อนที่ขา D0,D1,D2 และ D3 ด้วยค่าต่างๆ

2.7 การติดต่อสื่อสารผ่านพอร์ตอนุกรม

การติดต่อสื่อสารผ่านพอร์ตอนุกรมนั้นมีข้อยุ่งยากมากกว่าการใช้พอร์ตขนานสำหรับอุปกรณ์ทั่วๆ ไปแล้วจะต้องการข้อมูลในการประมวลผลแบบขนาน ดังนั้นจึงเพิ่มอุปกรณ์ที่ทำหน้าที่ในการแปลงข้อมูลแบบอนุกรมมาเป็นข้อมูลแบบขนาน อุปกรณ์นี้คือ UART และซอฟท์แวร์ที่ใช้ควบคุมการทำงานก็จะยุ่งยากกว่าการควบคุมผ่านพอร์ตมาตรฐานแบบขนาน แต่การสื่อสารโดยใช้พอร์ตอนุกรมนั้นก็มีข้อดีอยู่ไม่น้อยเช่นกันอาทิเช่น

1. การติดต่อสื่อสารผ่านพอร์ตอนุกรมสามารถใช้สายได้ยาวกว่าการติดต่อสื่อสารพอร์ตขนาน โดยที่พอร์ตอนุกรมจะใช้แรงดันในช่วง –3 ถึง –25 V แทนลอจิก 0 และใช้แรงดันในช่วง +3 ถึง +25 V แทนลอจิก 1 ในขนานที่พอร์ตขนานจะใช้แรงดัน 0 V แทนลอจิก 0 และ +5 V แทนลอจิก 1 ดังนั้นจะเห็นว่า ช่วงการสวิงแรงดันของพอร์ตอนุกรมจะมีค่าประมาณ 50 V ส่วนพอร์ตขนานจะมีช่วงสวิง 5 V เท่านั้น ซึ่งจะเห็นได้อย่างชัดเจนว่า หากมีการสูญเสียในสายแล้วการสื่อสารผ่านพอร์ตอนุกรมจะสามารถส่งข้อมูลไปได้ไกลกว่าอย่างแน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้จำนวนสายน้อยกว่าการส่งข้อมูลแบบขนานในกรณีที่อุปกรณ์อยู่ห่างจากเครื่องคอมพิวเตอร์มากๆ ย่อมจะสะดวกและประหยัดกว่า หากจะเดินสายเพียง 3 เส้น ซึ่งเป็นลักษณะการสร้างของนัลโมเด็ม (Null modem) เมื่อเทียบกับการเดินสายจำนวน 19 หรือ 25 เส้น ในการพอร์ตขนาน

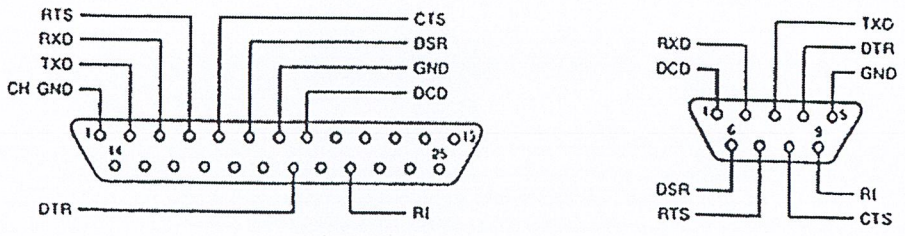
3. ปัจจุบันอุปกรณ์ที่ใช้แสงอินฟราเรด(Infrared) ได้รับความนิยมมากขึ้น ซึ่งจะเห็นได้จากอุปกรณ์ประเภทสมุดบันทึกอิเล็กทรอนิกส์ เครื่องคอมพิวเตอร์แบบโน้ตบุค ฯลฯ จะมีการติดต่อสื่อสารโดยใช้อินฟราเรดร่วมอยู่ด้วยและแน่นอนว่าการใช้อินฟราเรดก็จะต้องใช้การติดต่อผ่านพอร์ตอนุกรม เนื่องจากความไม่สะดวกอย่างยิ่งในการที่จะส่งข้อมูลแบบขนานด้วยอินฟราเรด

ลักษณะทางฮาร์ดแวร์

อุปกรณ์ที่ใช้สื่อสารแบบอนุกรมสามารถแยกออกได้เป็น 2 ประเภทคือ DCE (Data communication Equipment) อุปกรณ์เหล่านี้ได้แก่ โมเด็ม TA อะแดปเตอร์ , พล็อตเตอร์ ฯลฯ และ DTE (Data Terminal Equipment) ซึ่งก็คือ คอมพิวเตอร์นั้นเอง ข้อกำหนดทางไฟฟ้าของพอร์ตอนุกรมได้ถูกกำหนดเป็นมาตรฐานโดย EIA (Electronic Industry association) หรือ RS-232 ซึ่งประกอบไปด้วยสิ่งต่างๆ เหล่านี้

1. ช่วงไม่มีข้อมูล (space) หรือลอจิก 0 ต้องมีแรงดันอยู่ในช่วง –3 และ -25 V
2. ช่วงข้อมูล (mark) หรือลอจิก 1 ต้องมีแรงดันอยู่ในช่วง +3 และ +25 V
3. แรงดันในช่วง –3 ถึง +3 V ไม่มีการนิยามไว้
4. แรงดันในขณะเปิดวงจรต้องมีค่าไม่เกิน 25V
5. กระแสขณะช็อตวงจรมีค่าไม่เกิน 500 mA

ข้อมูลเหล่านี้ยังไม่ใช้ข้อกำหนดที่ครอบคลุมมาตรฐานของ RS-232 ทั้งหมดมาตรฐานของ RS-232 นั้นนอกจากจะมีคุณสมบัติดังที่ได้กล่าวมาแล้ว ยังจะต้องประกอบด้วยค่าคาปาซิแตนซ์ของสายอัตราบอดสูงสุด ฯลฯ ซึ่งรายละเอียดต่างๆ ได้ดูจากมาตรฐานของ EIARS-232-C ตามมาตรฐานของ RS-232 กำหนดอัตราบอดไว้สูงสุดที่ 20,000 BPS ซึ่งค่อนข้างจะช้าเกินไปสำหรับมาตรฐานในปัจจุบันในช่วงหลังได้มีการกำหนดมาตรฐาน RS-232 ขึ้นและยังคงใช้กันอยู่ในปัจจุบัน พอร์ตอนุกรมนี้จะมีอยู่ด้วยกัน 2 ขนาด คือ คอนเนคเตอร์แบบ D-type ตัวผู้ขนาด 25 pin รูปที่ 2.31 (ก) และคอนเนคเตอร์แบบ D-type ตัวผู้เช่นกันขนาด 9 pin รูปที่ 2.31 (ข) ซึ่งคอนเนคเตอร์ทั้ง 2 แบบนี้จะติดอยู่ที่ด้านหลังของเครื่องคอมพิวเตอร์



รูปที่ 2.31 (ก) คอนเนคเตอร์แบบ DB-25 (ข) คอนเนคเตอร์แบบ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D-type 25 Pin NO	D-type 25 Pin NO	Abbreviation	Full Name
Pin 2	pin 3	TD	Transmit Data
Pin 3	pin 2	RD	Receive Data
Pin 4	pin 7	RTS	Request To Send
Pin 5	pin 8	CTS	Clear To Send
Pin 6	pin 6	DSR	Data Set Ready
Pin 7	pin 5	SG	Signal Ground
Pin 8	pin 1	CD	Carrier Detect
Pin 20	pin 4	DTR	Data Terminal Ready
Pin 22	pin 9	RI	Ring Indicator

ตารางที่ 2.7 แสดงตำแหน่งขาสัญญาณของพอร์ทอนุกรม

Abbreviation	Full Name	FUNCTION
TD	Transmit Data	- Serial Data Output (TXD)
RD	Receive Data	- Serial Data Input (RXD)
CTS	Clear To Send	- This line indication that the modem is Ready to exchange data
DCD	Data Carrier Detect	- When the modem detect a carrier from the modem at the other end of the phone line this line becomes active
DSR	Data Set Ready	- This tell the UART that the modem is ready establish link
DTR	Data Terminal Ready	- This is the opposite to DSR This tell the MODEM that the UART is ready to link
RTS	Request To Send	- This line inform the MODEM that the UART is ready to exchange data
RI	Ring Indicator	- Goes active when modem detect ringing signal from the PSTN

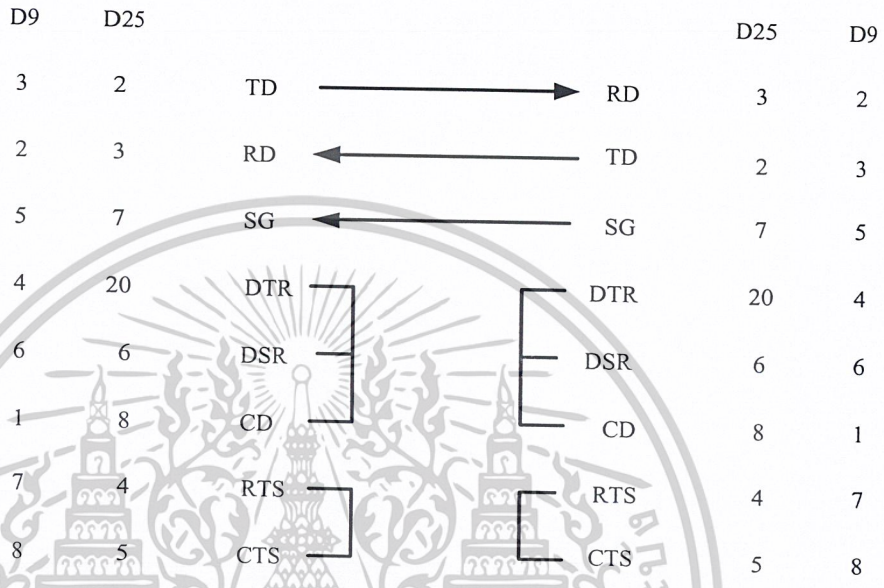
ตารางที่ 2.8 แสดงหน้าที่ของขาสัญญาณแต่ละเส้น

2.7.1 นัลโมเด็ม (Null Modem)

นัลโมเด็มใช้เพื่อเชื่อมต่อ DTE สองเครื่องเข้าด้วยกัน ซึ่งสามารถนำไปใช้ในการส่งผ่านข้อมูลระหว่างเครื่องคอมพิวเตอร์ด้วยโปรโตคอลแบบ Z modem หรือ Z modem รูปที่ 2.32 แสดงการเชื่อมต่อสายของนัลโมเด็มลักษณะการต่อสายของโมเด็มนั้นจะใช้สายเพียง 3 เส้น (TD, RD และSG) โดยที่สาย TD ของคอมพิวเตอร์เครื่องหนึ่งจะต่อไปยังสาย RD ของคอมพิวเตอร์อีกเครื่องหนึ่ง และในทางกลับกัน RD ของคอมพิวเตอร์เครื่องหนึ่งจะต่อกลับไปยังสาย TD ของคอมพิวเตอร์อีกเครื่องหนึ่ง ส่วนสาย SG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของคอมพิวเตอร์ทั้งสองเครื่องจะถูกต่อเข้าด้วยกัน การทำ loop back นั้นจะนำสัญญาณ DTR ป้อนเข้ากับขา DSR และ CD ของคอมพิวเตอร์ทั้งสองเครื่อง เนื่องจากเมื่อสัญญาณ DTR แอกทีฟจะทำให้ DTR และ CD แอกทีฟทันทีทำให้คอมพิวเตอร์เข้าใจว่าโมเด็มพร้อมที่จะทำงานแล้ว ส่วนการต่อสาย RTS ต่อเข้ากับ CTS ของคอมพิวเตอร์ทั้งสองเครื่องนั้น เพื่อต้องการให้คอมพิวเตอร์ติดต่อสื่อสารกันด้วยความเร็วที่เท่ากัน



รูปที่ 2.32 แสดงการเชื่อมต่อสายของนัลโมเด็ม

2.7.2 UART (8250 and Compatibles)

UART นั้นหมายถึง Universal asynchronous Receiver/Transmitter ในชุดของ 8250 จะประกอบด้วย 16450, 16550, 16650 และ 16750 เบอร์ของ UART เหล่านี้มักพบมากในคอมพิวเตอร์ทั่วๆ ไป ในระดับแรงดันสำหรับ UART จะเป็นระดับ TTL ดังนั้นจึงต้องอาศัย RS-232 Level Converters เพื่อเปลี่ยนระดับแรงดันของ RS-232 ให้กลับมาเป็นระดับแรงดันของ RS-232 ให้กลับมาเป็นระดับแรงดันของ TTL (0-5 V)

2.7.3 รีจีสเตอร์ของพอร์ตอนุกรม

ตำแหน่งพอร์ตและ IRQ ของพอร์ตอนุกรมนั้นมีลักษณะและการกำหนดและการอ่านตำแหน่งจาก BIOS คล้ายๆ กับพอร์ตขนานในตารางที่ 2.9 เป็นการแสดงตำแหน่งมาตรฐานของพอร์ตอนุกรมและ IRO ที่ใช้สำหรับตำแหน่งของพอร์ตแต่ละเบอร์ ส่วนตารางที่ 2.10 แสดงรีจีสเตอร์ทั้งหมดของ UART สำหรับเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name	Address	IRQ
Com 1	3F8	4
Com 2	2F8	3
Com 3	3E8	4
Com 4	2E8	3

ตารางที่ 2.9 แสดงตำแหน่งมาตรฐานของพอร์ตอนุกรม

สำหรับตำแหน่งที่เก็บไว้ใน BIOS คือ 0000;0400h โดยแต่ละตำแหน่งจะใช้พื้นที่ในการเก็บ 2 ไบต์

Base Address	DLAB	Read / Write	Abr.	Register Name
+ 0	= 0	Write	-	Transmitter Holding Buffer
	= 0	Read	-	Receive Buffer
	= 1	Read / Write	-	Divisor Latch Low Byte
+ 1	= 0	Read / Write	IER	Interrupt Enable Register
	= 1	Read / Write	-	Divisor Latch High Byte
+ 2	-	Read	IIR	Interrupt Identification Register
	-	Write	FCR	FIFO Control Register
+ 3	-	Read / Write	LCR	Line Control Register
+ 4	-	Read / Write	MCR	Modem Control Register
+ 5	-	Read	LSR	Line Status Register
+ 6	-	Read	MSR	Modem Status Register
+ 7	-	Read / Write	-	Scratch Register

ตารางที่ 2.10 แสดงรีจีสเตอร์ของพอร์ตอนุกรม

2.7.4 DLAB

ในตารางที่ 2.11 จะเห็นว่ามีคอลัมน์ DLAB ซึ่งเมื่อมีการเปลี่ยนแปลง ค่าจาก 0 เป็นหนึ่ง จะทำให้ชื่อรีจีสเตอร์เปลี่ยนตามไปด้วยนั้น เนื่องจาก UART มีรีจีสเตอร์ทั้งหมด 12 ตัว แต่มีพอร์ตเพียง 8 พอร์ทเท่านั้น DLAB หมายถึง Divisor Latch Access Bit เมื่อกำหนดให้ DLAB ค่าเป็น 1 ผ่านทาง Control Register จะทำให้สามารถกำหนดความเร็วในการในการติดต่อสื่อสารได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Speed (PBS)	Divisor (Dec)	Divisor Latch Low Byte	Divisor Latch High Byte
50	2304	90h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

ตารางที่ 2.11 แสดงอัตราบอดและค่าใช้ในการหาร

Interrupt Enable Register (IER)

Bit	Note
Bit 7	Reserved
Bit 6	Reserved
Bit 5	Enable Low Power Model (16750)
Bit 4	Enable Sleep Mode (16750)
Bit 3	Enable Modem Status Interrupt
Bit 2	Enable Receive Line Status Interrupt
Bit 1	Enable Transmitter Holding Register Empty Interrupt
Bit 0	Enable Receive Data Available Interrupt

ตารางที่ 2.12 แสดง Interrupt Enable Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

First In / First Out Control Register (FCR)

Bit	Note		
Bit 6 and Bit 7	Bit 7	Bit 6	Interrupt Trigger Level
	0	0	1 Byte
	0	1	4 Byte
	1	0	8 Byte
	1	1	14 Byte
Bit 5	Enable 64 Byte FIFO (16750 only)		
Bit 4	Reserved		
Bit 3	DMA Mode Select Chang Status of RXRDY & TXRDX Pins form Mode 1 to Mode 0		
Bit 2	Clear Transmit FIFO		
Bit 1	Clear Receive FIFO		
Bit 0	Enable FIFO		

ตารางที่ 2.13 แสดง FIFO Control Register

Modem Control Register (MCR)

Bit	Notes
Bit 7	Reserved
Bit 6	Reserved
Bit 5	Auto flow Control Enable (16750 only)
Bit 4	Loop Back Mode
Bit 3	Aux. Output 2
Bit 2	Aux. Output 1
Bit 1	Force Request to Send
Bit 0	Force Data Terminal Ready

ตารางที่ 2.14 แสดง Modem Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Modem Status Register (MSR)

Bit	Notes
Bit 7	Carrier Detect
Bit 6	Ring Indicator
Bit 5	Data Set Ready
Bit 4	Clear to Send
Bit 3	Delta Data Carrier
Bit 2	Trailing End Ring Indicator
Bit 1	Delta Data Set Ready
Bit 0	Delta Clear to Send

ตารางที่ 2.15 แสดง Modem Status Register

Interrupt Identification Register

Bit			Note
Bit 6 And Bit 7	Bit 6	Bit 7	
	0	0	No FIFO
	0	1	FIFO Enable but Unstable
	1	1	FIFO Enable
Bit 5			64 Byte FIFO Enable (16750 only)
Bit 4			Reserved
Bit 3	0		Reserved
	1		16550 Time-out Interrupt Pending
Bit 1 And Bit 2	Bit 2	Bit 1	
	0	0	Modem Interrupt
	0	1	Transmitter Holding Register Interrupt
	1	0	Received Data Available Interrupt
	1	1	Received Line Status Interrupt
Bit 0	0		Interrupt Pending
	1		No Interrupt Pending

ตารางที่ 2.16 แสดง Interrupt Identification Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line Control Register (LCR)

Bit	Notes			
Bit 7	1	Divisor Latch Access Bit		
	0	Access to Receive buffer. Transmitter Buffer & Interrupt Enable Register		
Bit 6	Set Break Enable			
Bit 3, 4 and 5	Bit 5	Bit 4	Bit 3	Parity Select
	X	x	0	No Parity
	0	0	1	Odd Parity
	0	1	1	Even Parity
	1	0	1	High Parity (Sticky)
	1	1	1	Low Parity (Sticky)
Bit 2	Length of Stop Bit			
	0	One Stop bit		
	1	2 Stop bits for words of Length 6, 7, 8 or 1, 5 Stop Bits for Word Lengths of 5 Bits		
Bit 1, 0	Bit 1	Bit 0	Word Length	
	0	0	5 Bit	
	0	1	6 Bit	
	1	0	7 Bit	
	1	1	8 Bit	

ตารางที่ 2.17 แสดง Line Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line Status Register (LSR)

Bit	Notes
Bit 7	Error in Received FIFO
Bit 6	Empty Data Holding Register
Bit 5	Empty Transmitter Holding Register
Bit 4	Break Interrupt
Bit 3	Framing Error
Bit 2	Party Error
Bit 1	Overrun Error
Bit 0	Data Ready

ตารางที่ 2.18 แสดง Line Status Register

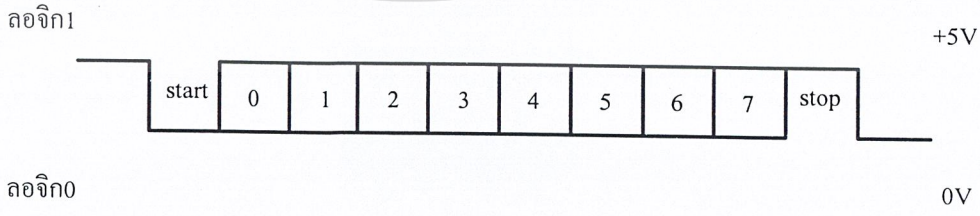
Scratch Register

สำหรับรีจีสเตอร์ตัวนี้ไม่ได้นำมาใช้สำหรับการสื่อสาร แต่จะใช้เป็นตัวเก็บข้อมูล UART ที่ใช้ก็มี 8250/8250B หรือ 8250A/16450 ซึ่งในปัจจุบันไม่นิยมใช้แล้ว โดยเฉพาะ 8250/8250B นั้นไม่ได้ออกแบบมาสำหรับเครื่อง AT และไม่สามารถกำหนดความเร็วของบัสได้

2.7.5 อุปกรณ์ที่ใช้ในการติดต่อสื่อสารผ่าน RS-232

RS-232 Waveforms

จากในตอนต้นเราได้ทำความรู้จักกับการสื่อสารโดยใช้ RS-232 มาแล้ว ซึ่งทำให้เราทราบว่าการสื่อสารโดยใช้ RS-232 นั้นเป็นการสื่อสารแบบอะซิงโครนัส (Asynchronous) นั่นคือการสื่อสารไม่จำเป็นต้องมีการส่งสัญญาณนาฬิกา (clock) ไปกับข้อมูลด้วยซึ่งแต่ละเวิร์ดของข้อมูลจะใช้บิดเริ่มต้น(start bit) ในการซิงโครนัสข้อมูล โดยอาศัยสัญญาณนาฬิกาที่แต่ละด้านของการส่งข้อมูล

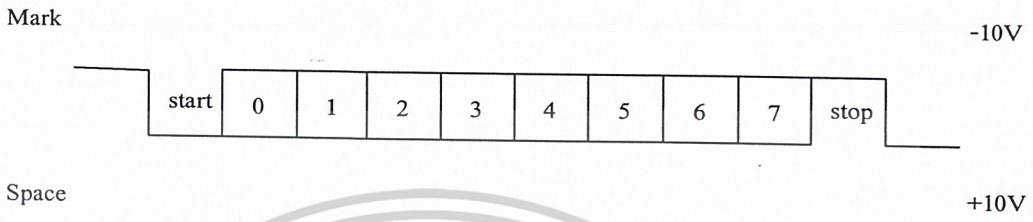


รูปที่ 2.33 แสดงรูปคลื่นของการส่งสัญญาณแบบอนุกรมด้วยระดับแรงดันของ TTI/CMOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดอะแกรมในรูปข้างบนแสดงรูปคลื่นจาก UART เมื่อใช้รูปแบบของการส่งแบบ 8IN ซึ่ง 8INนี้หมายถึง ประกอบด้วยข้อมูล 8 บิต ไม่มีพาริตี้และ 1 บิตหยุด (stop) การส่งข้อมูลและการส่งบิตที่มีนัยสำคัญน้อยที่สุด (LSB) ก่อนส่วนที่มีนัยสำคัญมากที่สุด (MSB) จะอยู่ลำดับสุดท้าย

ในรูปที่ 2.34 แสดงรูปคลื่นของการรับและส่งข้อมูลผ่านพอร์ต RS-232 ซึ่งระดับของแรงดันจะแตกต่างจากในรูปที่ 2.33 แต่มีลักษณะข้อมูลเหมือนกัน

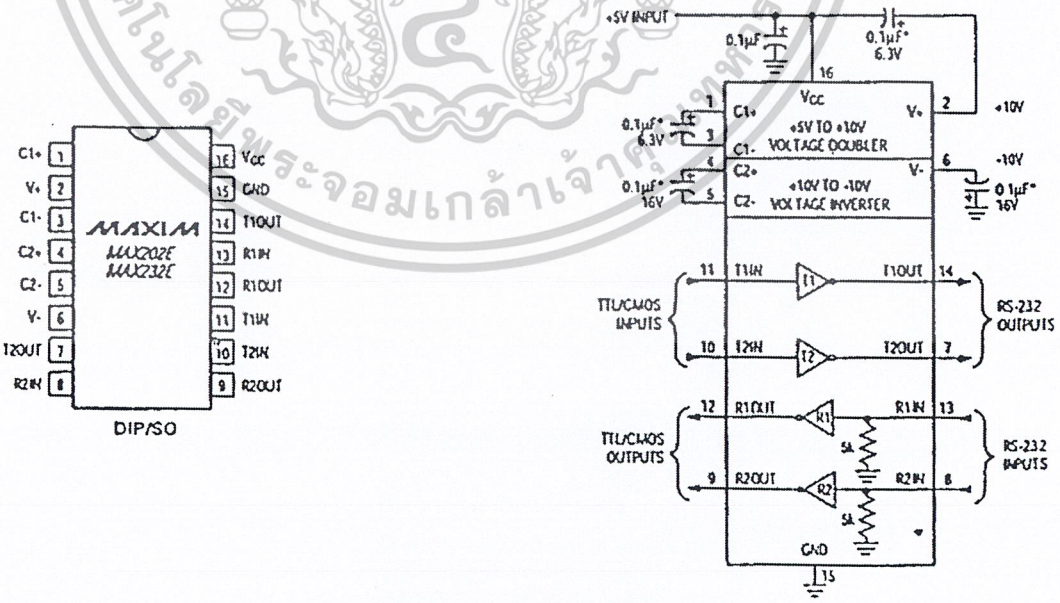


รูปที่ 2.34 แสดงคลื่นของสัญญาณที่รับ-ส่งผ่านพอร์ท RS-232

RS-232 Level converters

อุปกรณ์ดิจิตอลโดยส่วนใหญ่จะต้องการระดับแรงดัน TTL และ CMOS ดังนั้นขั้นตอนแรกในการต่ออุปกรณ์เหล่านี้เข้ากับ RS-232 จึงต้องแปลงระดับแรงดันของ RS-232 ไปเป็นระดับแรงดัน ซึ่งสามารถทำได้โดยการใช้ RS-232 Level converters

ในรูปที่ 2.30 แสดงลักษณะขาและวงจรภายในของชิพ MAX ซึ่งจะสร้างระดับแรงดัน +10V และ -10V จากแหล่งจ่าย +5V ในไอซีนี้จะประกอบด้วยตัวรับและตัวส่งอย่างละ 2 ชุด



รูปที่ 2.35 แสดงลักษณะขาและวงจรภายในของชิพ RS - 232

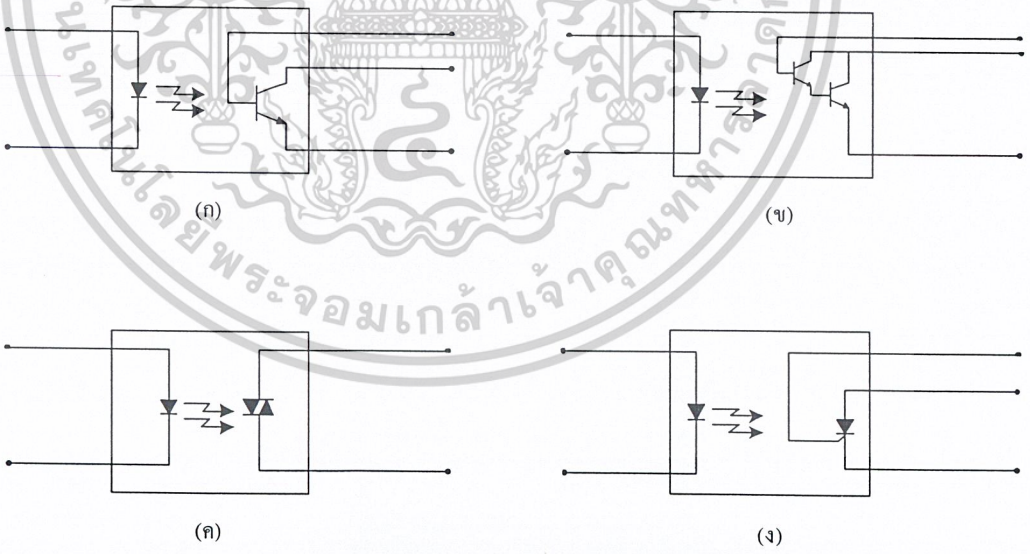
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การเชื่อมโยงทางแสง

อุปกรณ์เชื่อมโยงทางแสง (Optocoupler) หรือตัวแยกโดยใช้แสง เป็นอุปกรณ์ที่มีคุณสมบัติในการไอโซเลท ทำให้สามารถนำมาใช้ในการเชื่อมโยงสัญญาณต่างๆ ของวงจรที่มีกราวด์ต่างกัน สามารถป้องกันการรบกวนซึ่งกันและกันระหว่างอินพุทกับเอาท์พุตได้อย่างเด็ดขาด ซึ่งการคับปลิ้งด้วยวิธีอื่นๆ จะทำไม่ได้ จึงได้นำเอาออปโต้คัพเปลอร์มาประยุกต์ใช้งาน ในวงจรเพื่อประสิทธิภาพการทำงานและความน่าเชื่อถือของวงจร

ออปโต้คัพเปลอร์ เป็นอุปกรณ์เดี่ยวที่ประกอบด้วยแหล่งกำเนิดแสงและตัวตรวจรับแสง โดยที่ทั้งสองส่วนนี้จะแยกจากกันและกัน มีฉนวนที่โปร่งใส เช่น กระจกชิ้นบางๆ คั่นกลาง และชิ้นส่วนทั้งหมดจะถูกบรรจุอยู่ในตัวถังทึบแสง รูปร่างภายนอกมีอยู่หลายแบบ แต่ที่พบเห็นบ่อยๆ ส่วนมากจะมีตัวถังเป็นแบบดิพ (DIP : Dual in–line package) เหมือนไอซี แต่มี 6 ขา แหล่งกำเนิดแสงส่วนจะใช้ไดโอดเปล่งแสงอินฟราเรด (RED : Infrared Emitter Diode) ทำจากสารแกลเลียมอาร์เซไนด์ (GaAs) ส่วนตัวตรวจรับหรืออุปกรณ์ภาคเอาท์พุทนั้น อาจจะเป็นไฟโต้ทรานซิสเตอร์ โฟโต้ดาร์ลิงตันสวิทช์สองทิศทางซึ่งทำงานเมื่อมีแสงมากระตุ้น และ SCR ที่ถูกกระตุ้นด้วยแสงเป็นต้น รูปที่ 2.36 แสดงสัญลักษณ์ของวงจรชนิดต่างๆ ดังที่กล่าวมาแล้วข้างต้น ถึงแม้จะมีหลายชนิดมากกว่านี้ แต่ที่แสดงให้เห็นรูปเป็นที่พบเห็นกันบ่อยๆ

ออปโต้คัพเปลอร์หรือออปเตอร์ไอโซเลทเตอร์ ได้รับการออกแบบไว้ให้ทำการป้องกัน อุปกรณ์อิเล็กทรอนิคส์ไม่ให้ได้รับแรงดันกระชากสูงๆ หรือคุ้มครองระดับนอยส์ต่ำๆ ซึ่งเป็นเหตุให้เกิดเอาท์พุทไม่ถูกต้องหรือทำให้เกิดคลื่นผิดพลาดขึ้นมา



รูปที่ 2.36 แสดงสัญลักษณ์ของออปโต้คัพเปลอร์

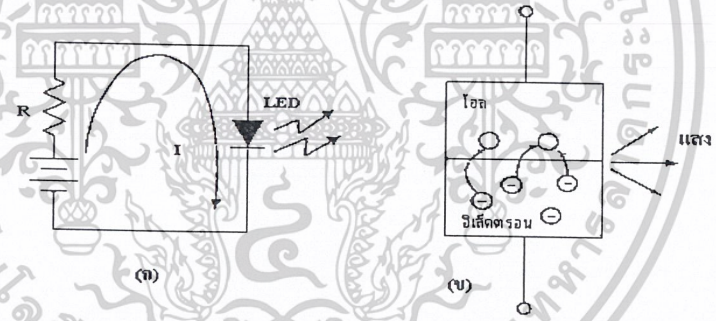
(ก) แบบมีเอาท์พุทเป็น Opto-Transistor (ข) แบบมีเอาท์พุทเป็น Opto-Darington Transistor

(ค) แบบมีเอาท์พุทเป็น Opto-Diac (ง) แบบมีเอาท์พุทเป็น Opto-SCR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออปโต้คัพเปลอร์เป็นอุปกรณ์ที่ทำให้สามารถเชื่อมต่อกับอุปกรณ์ตัวอื่นๆที่มีระดับลอจิกแตกต่าง ในออปโต้คัพเปลอร์ สัญญาณอินพุตจะถูกเปลี่ยนเป็นพลังงานแสง เพราะมี LED ที่อยู่ภายใน พลังงานจึงถูกส่งไปยังโฟโต้ดีเทคเตอร์ ดังนั้นมันจึงงานตรงกับพลังงานของแสงที่ได้จาก LED และสเปคตามอัตราส่วนการส่งผ่านกระแส (Current–Transfer Ratio ; CTR) กับ Isolation Voltage เป็นอัตราส่วนระหว่างกระแสอินพุตต่อกระแสเอาท์พุตซึ่งเป็นการวัดความสามารถของออปโต้คัพเปลอร์ในเรื่องความสามารถให้สัญญาณอินพุตถูกส่งไปยังเอาท์พุต อย่างมีประสิทธิภาพขึ้นอยู่กับ ประสิทธิภาพของ IRED ช่องว่างระหว่างชิ้นส่วนทางอินพุตและเอาท์พุต รวมทั้งพื้นที่ความไว และอัตราขยายของตัวตรวจรับ สำหรับ Isolation Voltage ของออปโต้คัพเปลอร์ คือ ปริมาณแรงดันที่ออปโต้คัพเปลอร์สามารถทำงานได้อย่างปลอดภัย

เมื่อมีกระแสไหลผ่าน IRED ของออปโต้คัพเปลอร์ ในลักษณะไบอัสตรงจนมีอิเล็กตรอน ส่วนเกินกระโดดข้ามรอยต่อไปรวมกับโฮล ในขณะเดียวกันก็ได้ปล่อยพลังงานโฟตอน หรือแสงออกมา ดังรูปที่ 2.37 แสงที่ได้รับเป็นแสงอินฟราเรด เพราะสารกึ่งตัวนำทำด้วยสารสารแกลเลียมอาร์เซไนด์ ตัวแปรอินพุททางด้านไฟฟ้ากระแสตรง เป็นตัวกำหนดตัวแปรทางด้านไฟฟ้ากระแสตรง เป็นตัวกำหนดตัวแปรทางด้านไฟฟ้าของไดโอดเปล่งแสงอินฟราเรด ได้แก่กระแสของไดโอดเมื่อได้รับไบอัสตรง (I_F) แรงดันตกคร่อมไดโอดเมื่อได้รับไบอัสตรง (V_F) และแรงดันสูงสุดที่ทนได้ เมื่อได้รับไบอัสกลับ



รูปที่ 2.37 (ก) แสดงถึงแสงที่เกิดขึ้นเนื่องจากกระแสไบอัสตรงไหลผ่าน
(ข) อิเล็กตรอนส่วนเกินข้ามรอยต่อไปรวมกับโฮลพร้อมกับเปล่งแสงออกมา

เนื่องจากตัวแปรเอาท์พุททางด้านไฟฟ้ากระแสตรงและตัวแปรส่งถ่าย (Transfer Parameter) นั้นจะแตกต่างกัน โดยขึ้นอยู่กับชนิดของชิ้นส่วนที่เป็นตัวตรวจจับ ที่ใช้ในออปโต้คัพเปลอร์ ซึ่งมีรายละเอียดแตกต่างกันขึ้นอยู่กับตัวตรวจรับนั้นๆ ตัวอย่างเช่น

2.8.1 ทรานซิสเตอร์คัพเปลอร์ (Transistor coupler)

อุปกรณ์ประเภทนี้ได้รับความนิยมมากที่สุด มีความไวระดับกลาง และมีราคาถูก ตรงจุดเชื่อมต่อ (Junction) ภายในระหว่างคอลเลคเตอร์-เบส ของทรานซิสเตอร์สามารถเอาสายมาต่อข้างนอกให้ทำหน้าที่โฟโต้ไดโอด ซึ่งมีความเร็วในการทำงานสูงยิ่งไปกว่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 ดาร์ลิงตันทรานซิสเตอร์คัพเปลอร์ (Darlington Transistor Coupler)

อุปกรณ์ประเภทนี้ให้อัตราส่วนการส่งกระแส หรือมีเกนการขยายสูงให้กระแสเอาท์พุทเพิ่มขึ้น ซึ่งจะได้แกนขยายสูงเป็น 10 เท่า แต่ความเร็วในการทำงานจะช้ากว่า 10 เท่า ของการใช้งานทรานซิสเตอร์ตัวเดียว

ออปโต้แบบทรานซิสเตอร์คัพเปลอร์ และแบบดาร์ลิงตันทรานซิสเตอร์คัพเปลอร์นั้น มีหลักการทำงานเหมือนกัน รอยต่อระหว่างขาคอลเลคเตอร์กับขาเบสถูกทำให้กว้างขึ้น แสงที่ตกกระทบรอยต่อจะทำให้เกิดคู่ของอิเล็กตรอนและโฮลขึ้นมาเกิดการนำกระแสได้ ตัวแปรสำหรับออปโต้ทรานซิสเตอร์คัพเปลอร์ และแบบดาร์ลิงทรานซิสเตอร์คัพเปลอร์ มีดังนี้

I_C : เป็นกระแสสูงสุดที่ไหลต่อเนื่องผ่านขาคอลเลคเตอร์ (เอาท์พุท)

$V_{(BR)CBO}$: เป็นแรงดันพังทลายสูงสุดจากขาคอลเลคเตอร์ไปยังขาเบส

$V_{(BR)CEO}$: เป็นแรงดันพังทลายสูงสุดจากขาคอลเลคเตอร์ไปยังขาอิมิตเตอร์

$V_{(BR)ECO}$: เป็นแรงดันพังทลายสูงสุดจากขาอิมิตเตอร์ไปยังขาคอลเลคเตอร์

$CTR_{(n)}$: เป็นอัตราส่วน (เป็นเปอร์เซ็นต์) ต่ำสุดระหว่างกระแสเอาท์พุทของคอลเลคเตอร์สูงสุดต่อกระแสไดโอดที่ค่า V_{cc} และ I_f ที่กำหนด

$V_{CE(sat)}$: เป็นแรงดันอิ่มตัวระหว่างขาคอลเลคเตอร์และขาอิมิตเตอร์

2.8.3 ออปโต้คัพเปลอร์ที่ใช้สวิทซ์สองทิศทางหรือไตรแอค (Triac)

ทำงานเมื่อมีแสงมากระตุ้นเป็นภาคเอาท์พุท ถูกออกแบบมาสำหรับใช้งานซึ่งต้องการการแยกการทริก หรือกระตุ้นตัวไตรแอค การแยกการสวิทซ์ทางด้านไฟฟ้าสลับที่มีปริมาณกระแสต่ำและการแยกกันทางไฟฟ้าที่มีค่าสูง อุปกรณ์ชนิดนี้มีตัวแปรที่สำคัญคือ

$I_{T(RMS)}$: เป็นกระแสค่า RMS สูงสุดขณะอยู่ในสถานะที่ทำงาน

V_{DRM} : เป็นแรงดันซ้ำๆ ระหว่างขั้วเอาท์พุทเมื่ออยู่ในสถานะหยุดทำงาน

V_{TM} : เป็นแรงดันยอดสูงสุด (Peak Voltage) เมื่ออยู่ในสถานะที่ทำงาน

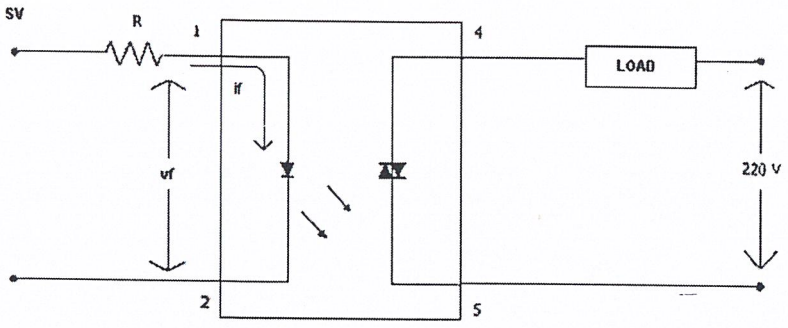
I_{FT} : เป็นค่ากระแสกระตุ้นไดโอดเปล่งแสงอินฟราเรดสูงสุดซึ่งต้องการใช้เพื่อคงสถานะให้เอาท์พุทค้างไว้ (Latch)

I_H : เป็นค่ากระแสยึด (Holding Current) ซึ่งต้องการสำหรับเอาท์พุท เพื่อที่จะคงสถานะค้างเอาไว้ให้ได้

2.8.4 การประยุกต์ใช้งานโดยการนำไปใช้ควบคุมโหลด

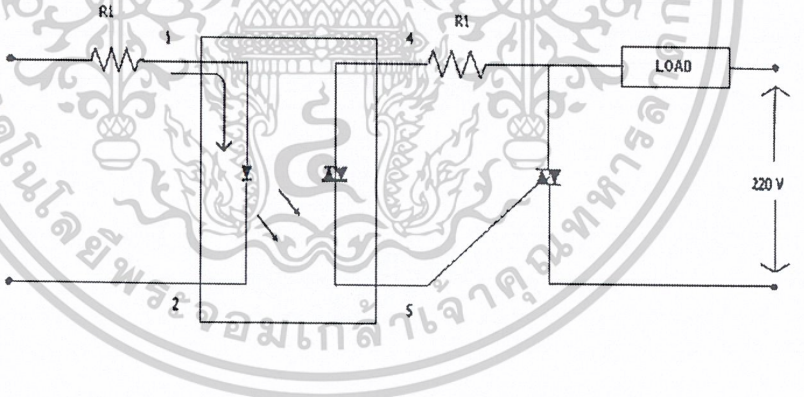
การนำเอาออปโต้แบบไตรแอคคัพเปลอร์มาใช้ควบคุมโหลดที่เป็นไฟสลับ 220 โวลท์ แทนการใช้รีเลย์ และการควบคุมปราศจากข้อยุ่งยากเหมือนกับวงจรที่ออกแบบโดยใช้รีเลย์ ดังนั้นจะกล่าวถึงเฉพาะการนำเอาออปโต้แบบไตรแอคคัพเปลอร์มาประยุกต์ใช้งานเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 แสดงวงจรที่ใช้ควบคุมกำลังงานไฟฟ้ากระแสสลับ

จากรูปที่ 2.38 แสดงการใช้ MOC 3020 ในการเปิด-ปิด กระแสผ่านโหลดที่ต้องการกำลังงานไฟฟ้ากระแสสลับเพียงเล็กน้อย เมื่อเอาท์พุทจากลอจิกเกตมีค่าลอจิกเป็น 0 กระแสจะไหลผ่านไดโอดเปล่งแสงไปกระตุ้นไดโอด ให้นำกระแสไฟฟ้าสลับ และเมื่อเอาท์พุทของลอจิกเกทซึ่งป้อนเข้าสู่ออปโต้มีค่าลอจิกเป็น 1 จะทำให้ไม่มีกระแสไหลผ่านไดโอดอินฟราเรด จะทำให้ไดแอคหยุดนำกระแส จากรูปที่ 2.38 จะใช้ได้กับโหลดที่ใช้กระแสน้อยๆ เท่านั้น เพราะไดแอคขนาดเล็กสามารถทนกระแสได้น้อย ซึ่งน้อยเกินไปที่จะใช้ควบคุมอุปกรณ์ไฟฟ้าได้ แต่ก็เหมาะสมที่จะนำมาสร้างทริกเกอร์ไดแอคกำลังงานสูงนอกวงจร เพื่อควบคุมอุปกรณ์ไฟฟ้าที่ต้องการกระแสสูงๆ ได้

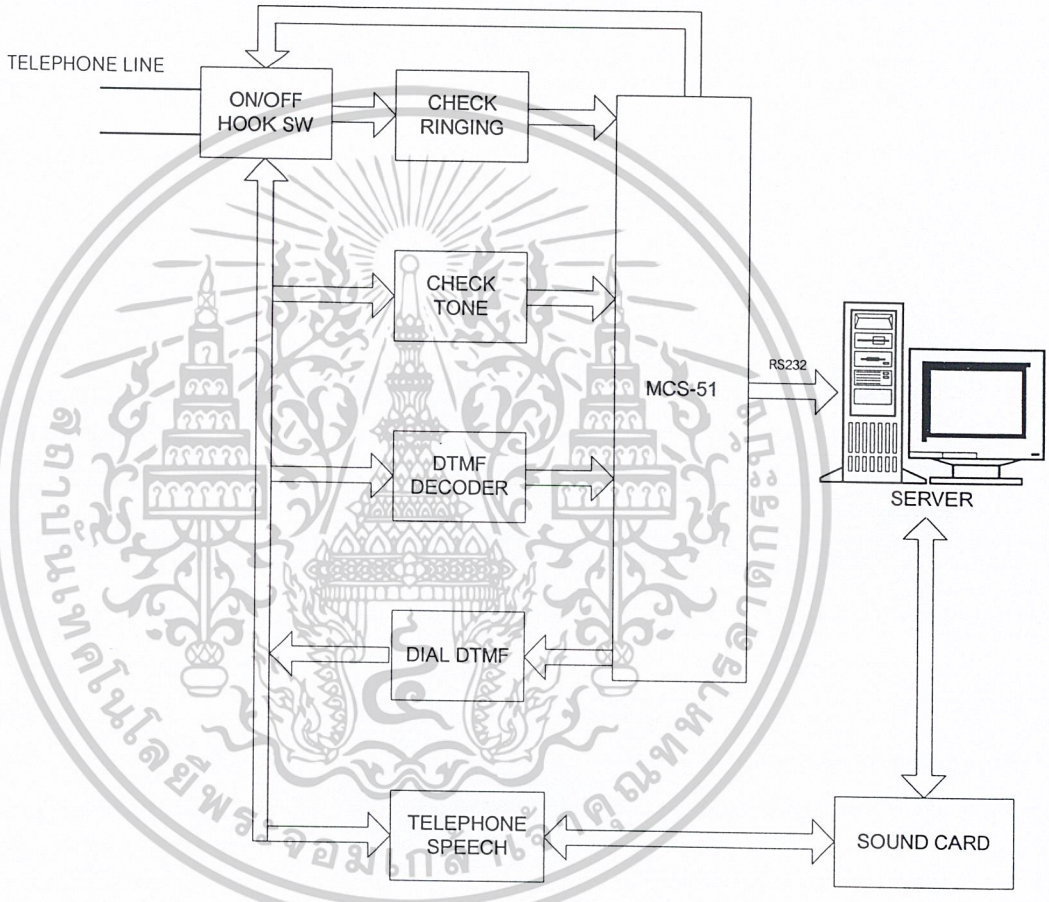


รูปที่ 2.39 แสดงวงจรที่ใช้ควบคุมกำลังไฟฟ้ากระแสสลับที่มีค่าสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

ในโครงงานนี้จะแบ่งเป็น 2 ส่วนหลักๆ คือ ส่วนทางด้านฮาร์ดแวร์ (Hardware) และส่วนของซอฟท์แวร์ (Software) โดยทางด้านฮาร์ดแวร์จะเป็นส่วนของการสื่อสารข้อมูลเสียงระหว่างสายโทรศัพท์กับการ์ดเสียง (Sound card) ของคอมพิวเตอร์ และเป็นส่วนวงจรเชื่อมต่อ (Interface) กับคอมพิวเตอร์เพื่อใช้การควบคุมการทำงาน ดังรูปที่ 3.1

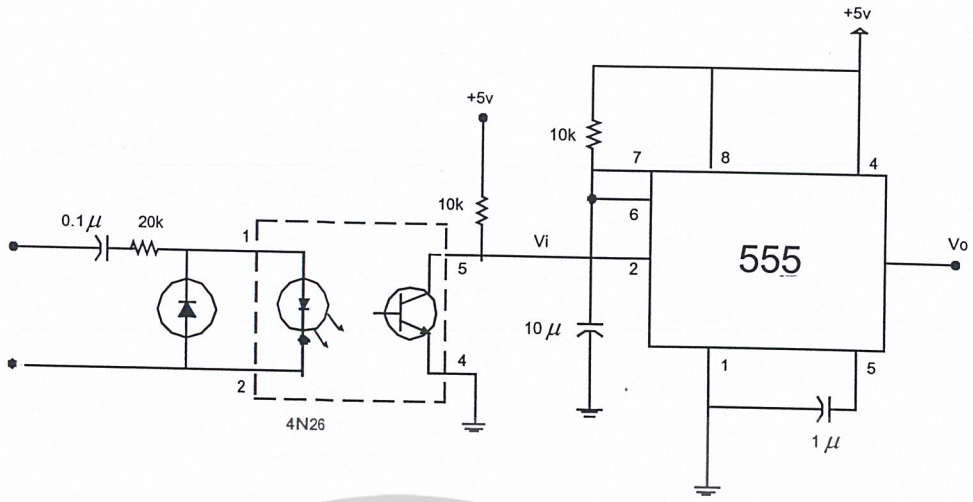


รูปที่ 3.1 แสดงบล็อกไดอะแกรมของฮาร์ดแวร์

3.1 วงจรตรวจจับสัญญาณกระดิ่ง (Check Ringing)

ในการทำงานเมื่อมีสัญญาณโทรศัพท์จะต้องมีการตรวจจับสัญญาณกระดิ่ง เพื่อให้วงจรทราบว่ามีการโทรศัพท์เข้ามาในระบบ ซึ่งจะทำให้หน่วยประมวลผลสั่งให้ทำการต่อวงจรเมื่อมีการเรียกเข้ามาเท่ากับจำนวนสัญญาณกระดิ่งที่ตั้งไว้ ซึ่งเปรียบเสมือนการยกหูโทรศัพท์ โดยวงจรที่ใช้แสดงได้ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงวงจรตรวจจับสัญญาณกระดิ่ง

การทำงานของวงจรสามารถอธิบายได้ดังนี้

เนื่องจากสัญญาณกระดิ่งเป็นสัญญาณกระแสสลับและมีขนาดค่อนข้างสูง เราจึงนำสัญญาณนี้มาใช้โดยตรงไม่ได้ จึงต้องทำการแปลงสัญญาณให้มีขนาดต่ำลง เพื่อให้วงจรอื่นปลอดภัย

สัญญาณกระดิ่งจะถูกลดทอนลงเมื่อนำมาผ่านตัวเก็บประจุ และตัวต้านทานที่ต่อตอนต้นของวงจร จากนั้นจะผ่านไดโอด เพื่อให้เฉพาะสัญญาณด้านบวกเท่านั้นที่ผ่านได้ และสัญญาณที่ผ่านไดโอดจะไปเข้าที่ไอซี 4N26 ซึ่งเป็น Optocoupler ทำหน้าที่แยกกราวนด์ของสัญญาณกระดิ่งออกจากกราวนด์ของระบบเอาท์พุทที่ได้จากไอซี 4N26 จะเป็นพัลส์ที่มีความถี่เท่ากับสัญญาณกระดิ่ง เพื่อที่ให้มีความเที่ยงตรงมากขึ้น จึงนำไปผ่านโมโนสเตเบิล โดยใช้ไอซี Timer 555 เพื่อสร้างพัลส์ขึ้นมาใหม่ โดยความกว้างของพัลส์สามารถกำหนดได้จากค่าพัลส์สามารถกำหนดได้จากค่า RC ที่ต่อกับไอซี

โดย

$$T = 1.1R_AC$$

เอาท์พุทที่ได้จากไอซี Timer 555 จะเป็นลอจิก 0 เมื่อไม่มีสัญญาณกระดิ่ง และเป็นลอจิก 1 เมื่อมีสัญญาณกระดิ่งสลับกันไปทำให้การตรวจจับสัญญาณโทรศัพท์สะดวกและถูกต้องขึ้น

$$T = 1.1R_AC$$

$$= 1.1 \times 100k \times 10\mu = 1.1$$

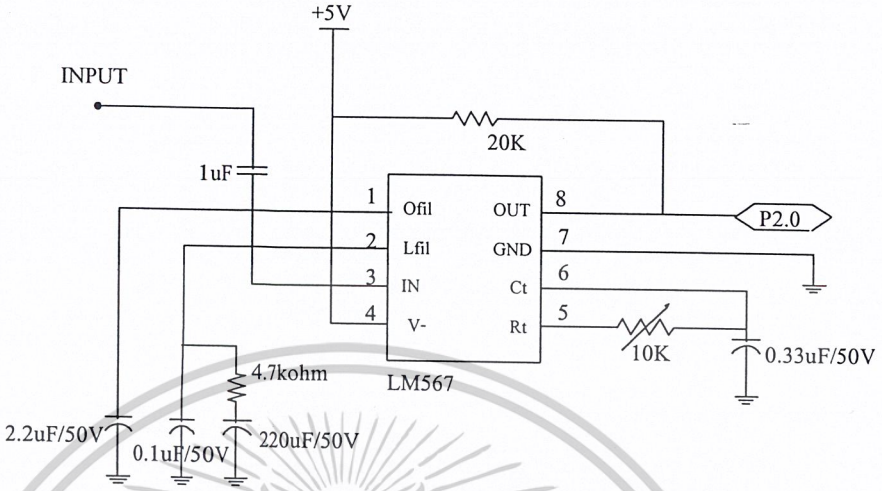
3.2 วงจรตรวจสอบสัญญาณความถี่เสียง (Check Tone)

วงจรตรวจสอบสัญญาณความถี่เสียง ทำหน้าที่ตรวจสอบสัญญาณโทนภายในคู่สาย เพื่อบอกให้ทราบว่าเป็นสัญญาณให้หมุนเลขหมาย สัญญาณไม่ว่างหรือสัญญาณตอบกลับ

จากรูปที่ 3.2 ไอซี LM567 เป็นหัวใจหลักของวงจรนี้ ทำหน้าที่เป็นวงจรโทน – ดีโค้ดเดอร์ ซึ่งจะทำงานในลักษณะตรงข้ามกับสัญญาณความถี่เสียงที่เข้ามา คือ ถ้าไม่มีสัญญาณความถี่เสียงเข้ามาทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตขา 3 และผลที่เอาต์พุตที่ขา 8 จะมีสภาวะเป็น 1 แต่ถ้ามีสัญญาณความถี่เสียงเข้ามาทางด้านอินพุต ก็จะทำให้ทางด้านเอาต์พุตมีค่าสภาวะ 0



รูปที่ 3.3 แสดงวงจรตรวจสอบสัญญาณความถี่เสียง

เมื่อมีสัญญาณความถี่เสียงเข้ามา C6 จะทำหน้าที่คัปปลิ้งสัญญาณความถี่เสียงไปยังขา 3 ของไอซี LM567 และผ่านเข้าไปยังภาคเฟสล็อกลูปที่อยู่ภายในตัวไอซี โดยมี VR1, R5 และ C5 เป็นตัวกำหนดความถี่ของวงจร และ C4 ทำหน้าที่กรองความถี่แบบความถี่ต่ำผ่าน และในขณะเดียวกันสัญญาณความถี่เสียงอีกส่วนหนึ่งจะแยกไปเข้ายังภาคแยกเฟส 90 องศา (Quadrature Phase Detector) ที่อยู่ภายในตัวไอซี ทำการเปรียบเทียบความถี่และเฟสของสัญญาณอินพุต จากนั้นส่งสัญญาณเอาต์พุตไปทำการแยก และส่งออกเป็นสภาวะ 0 ทางเอาต์พุตที่ ขา 8

3.3 วงจรถอดรหัสสัญญาณความถี่คู่ (DTMF Decoder)

เนื่องจากว่าสัญญาณที่ได้จากการกดหมายเลขแต่ละตัวบนหน้าปัดโทรศัพท์ จะออกมาเป็นความถี่คลื่นรูปซายน์ตามลักษณะของโทรศัพท์แบบกดปุ่ม ในการนำไปใช้งานจริงจำเป็นต้องเปลี่ยนความถี่คลื่นรูปซายน์ให้เป็นสัญญาณดิจิตอลเสียก่อน ก็คือจะต้องผ่านสัญญาณคลื่นรูปซายน์เข้าไปยังภาค DTMF Decoder ซึ่งจะนำมาใช้ในวงจรใช้งาน คือเมื่อมีผู้เรียกเข้ามาทางภาครับจะทำการรับคู่สาย แล้วภาค Automatic Answer Sound ส่งสัญญาณตอบรับผู้เรียกเข้ามา เพื่อให้ทำการกดเลขหมายของคู่สายย่อย เมื่อการกดเลขหมายของรู่สายย่อยที่ได้ออกมาเป็นสัญญาณ DTMF ก็จะผ่านภาค DTMF Decoder เพื่อที่แปลงสัญญาณคลื่นรูปซายน์ให้เป็นสัญญาณดิจิตอลเพื่อที่จะส่งให้ภาค Control ทำการวิเคราะห์ต่อเช็คคู่สายย่อยต่อไป ในวงจรใช้งานจริงจะใช้ไอซีเบอร์ MT8870 ทำหน้าที่ถอดรหัสความถี่ของโทรศัพท์แบบกดปุ่ม ให้เป็นตัวเลข BCD ได้ดังแสดงในตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

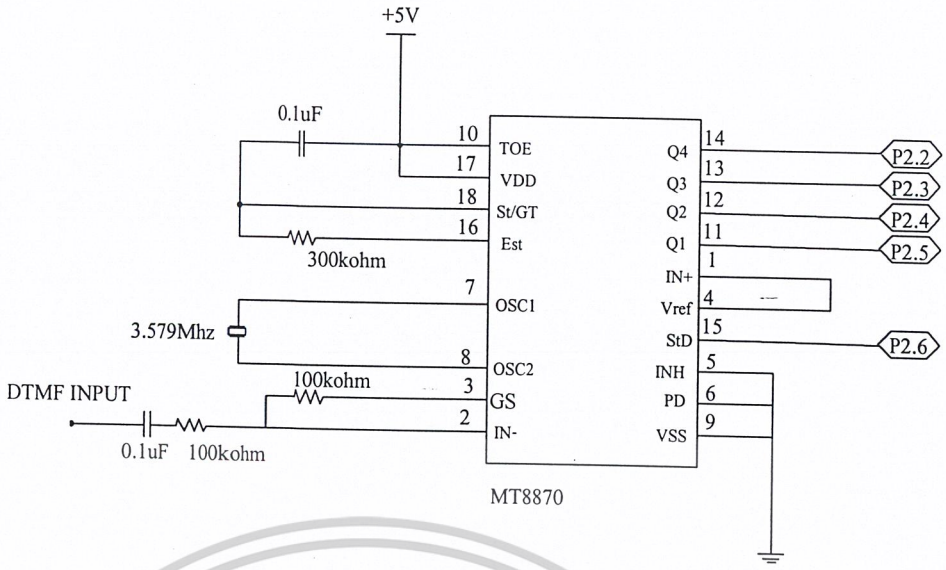
หมายเลขที่กด	BCD CODE	หมายเลขที่กด	BCD CODE
1	0001	7	0111
2	0010	8	1000
3	0011	9	1001
4	0100	0	1010
5	0101	*	1011
6	0110	#	1100

ตารางที่ 3.1 แสดงการถอดรหัสสัญญาณความถี่คู่

โดยส่วนที่สำคัญของวงจรนี้จะอยู่ที่ไอซีเบอร์ MT8870 ซึ่งเป็นไอซีที่ทำหน้าที่ในการถอดรหัสทางความถี่ที่ถูกส่งเข้ามาทางคู่สายโทรศัพท์สัญญาณความถี่จะถูกส่งออกมาจากผู้ที่โทรเข้ามาโดยการกดปุ่มที่เครื่องโทรศัพท์ของผู้ที่โทรเข้ามา เช่นในกรณีที่กดปุ่ม 1 ก็จะเป็นการผสมความถี่ 697 เฮิรตซ์ กับ 1209 เฮิรตซ์ ความถี่ที่เกิดจากการกดปุ่มนี้จะถูกส่งผ่านมาเข้าตามคู่สายโทรศัพท์ที่ C1 ซึ่งจะทำหน้าที่คับปลิ้งสัญญาณความถี่ผ่านไปยังขา 2 ของไอซีเบอร์ MT8870 ซึ่งไอซีเบอร์ MT8870 เป็นไอซีสำเร็จรูปที่สามารถถอดรหัสความถี่ได้เลยโดยเอาท์พุทที่ได้จะอยู่ในรูปของรหัสของเลขฐานสองจะมีทั้งหมด 16 ค่า และขาที่แสดงสภาวะการสโตรป ในวงจรนี้ได้ใช้ LED สีแดง 4 หลอด เป็นตัวแสดงสภาวะการถอดรหัสความถี่ให้อยู่ในรูปรหัสของเลขฐานสองและ LED สีแดง 4 หลอด เป็นตัวแสดงสภาวะการสโตรปรหัสเลขฐานสองที่ได้จากการถอดรหัสความถี่ซึ่งรหัสความถี่ที่ถอดได้แล้วจะถูกส่งต่อไปยังแผงวงจรควบคุมเพื่อที่จะได้สั่งงานให้วงจรส่วนอื่นๆ ทำงานต่อไป

ในส่วนของความหมายของขาไอซีเบอร์ MT8870 อุปกรณ์ที่นำมาต่อร่วมกับไอซีเบอร์ MT8870 เช่น แท่งคริสตอล ใช้ทำหน้าที่ในการกำเนิดความถี่ 3.8 เมกะเฮิรตซ์ ให้แก่ไอซีเบอร์ MT8870 เพื่อใช้ในการถอดรหัสความถี่ ในการปรับขนาดและแรงดันที่มีขนาดของแรงดันสูงให้ต่ำลงและส่งผ่าน หรือกั้นสัญญาณที่ไม่ต้องการเพื่อทำให้วงจรมีเสถียรภาพดีขึ้น และมีความแน่นอนในการทำงาน

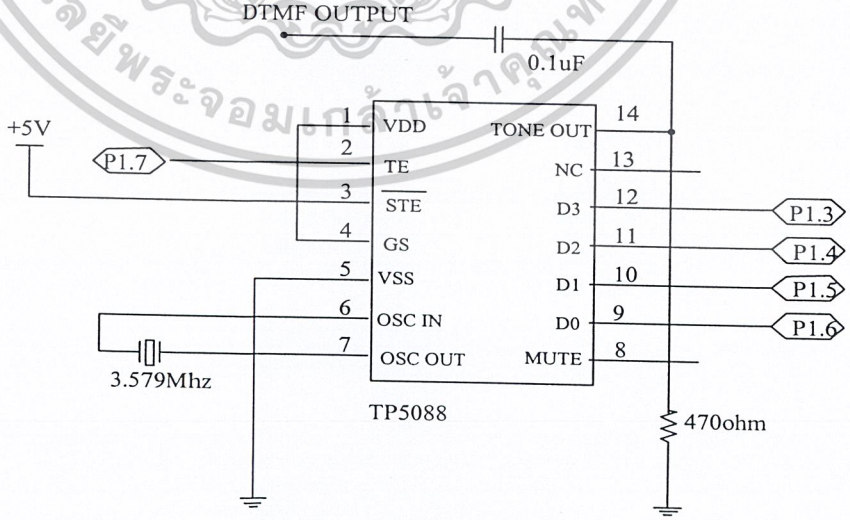
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงวงจรถอดรหัสสัญญาณความถี่คู่

3.4 วงจรกำเนิดสัญญาณคู่ความถี่ (Dial DTMF)

ไอซีที่ทำหน้าที่กำเนิดสัญญาณคู่ความถี่โดยใช้ไอซีเบอร์ TP5088 ที่สามารถนำมาประยุกต์ใช้งานในโทรศัพท์ ที่ถูกควบคุมด้วยไมโครคอนโทรลเลอร์ โดยไม่จำเป็นต้องผ่านคีย์บอร์ดในช่วงที่ขาโทนเอนาเบิล (Tone Enable) อยู่ในสภาวะแรงดันไฟต่ำจะทำให้วงจรออสซิลเลเตอร์ไม่ทำงาน และจะไม่รับข้อมูลที่ อินพุตที่เข้ามาขณะที่มีการเปลี่ยนแปลงสภาวะจากแรงดันไฟต่ำแรงดันไฟสูง ที่ขาเอนาเบิล ข้อมูลจะถูกป้อนเข้าไปในไอซี และสัญญาณคู่ความถี่ จะถูกสร้างจาก DTMF มาตรฐาน ซึ่งมีกลุ่มความถี่ต่ำ และความถี่สูง



รูปที่ 3.5 แสดงวงจรกำเนิดสัญญาณคู่ความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำเนิดสัญญาณคู่ความถี่ใช้ไมโครคอนโทรลเลอร์ควบคุมทางขาอินพุตและขาโทนเอนาเบิล โดยจะป้อนข้อมูลเข้ามาขา D0 – D3 ก่อน แต่ว่า TP 5088 จะยังไม่ทำงานและจะไม่รับข้อมูลอินพุตที่เข้ามา เพราะขาโทนเอนาเบิลอยู่ในสภาวะที่แรงดันไฟต่ำ จากนั้นจะเปลี่ยนจากสภาวะแรงดันไฟต่ำเป็นแรงดันไฟสูงที่ขาโทนเอนาเบิล เพื่อให้มีการรับข้อมูลทางขา D0 – D3 และสร้างสัญญาณคู่ความถี่ออกไปตามสัญญาณอินพุท D0 – D3 ตามตารางที่ 3.2

หมายเลข	Low Frequency	High Frequency	D3	D2	D1	D0
1	697	1209	0	0	0	1
2	697	1336	0	0	1	0
3	697	1477	0	0	1	1
4	770	1209	0	1	0	0
5	770	1336	0	1	0	1
6	770	1477	0	1	1	0
7	852	1209	0	1	1	1
8	852	1336	1	0	0	0
9	852	1477	1	0	0	1
0	941	1336	1	0	1	0

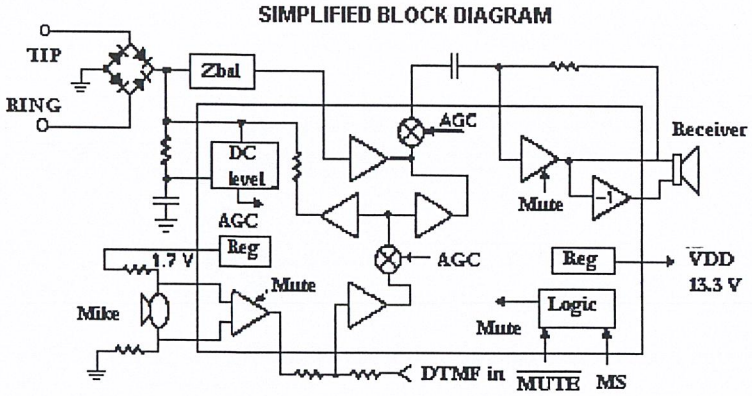
ตารางที่ 3.2 แสดงสัญญาณ DTMF ที่ได้จากไอซีเบอร์ TP5088
เมื่อป้อนที่ขา D0, D1, D2 และ D3 ด้วยค่าต่างๆ

3.5 วงจรควบคุมเสียงพูดแบบ 2 ทิศทาง (Two Way Speech Circuit)

วงจรควบคุมเสียงพูดแบบ 2 ทิศทาง (Two Way Speech Circuit) เป็นอีกส่วนหนึ่งภายในเครื่องโทรศัพท์ ที่จัดว่ามีความสำคัญต่อการทำงานของตัวเครื่องโทรศัพท์ เพราะเป็นส่วนที่จะต้องทำงานเกี่ยวกับสัญญาณพูดที่เราพูดผ่านไมโครโฟน หรือสัญญาณที่ได้ยินจากคู่สนทนา ข้อสำคัญของการออกแบบวงจรนี้ คือ การแมตช์ซิ่งอิมพีแดนซ์ของสายส่งสัญญาณจากชุมสายกับอิมพีแดนซ์ของวงจร ซึ่งจะต้องมีความใกล้เคียงกันมากที่สุด เพื่อประสิทธิภาพในการส่งสัญญาณ

ไอซี MC34114 ประกอบด้วยวงจรควบคุมเสียงพูดที่มีวงจรไฮบริดจ์ (วงจรแยกระบบสายส่งจาก 2 Wire เป็น 4 Wire) วงจรเชื่อมต่อกระแสไฟตรงที่ต่ออยู่กับสายทิปกับริง สามารถปรับแต่งอัตราขยายสัญญาณของด้านส่ง ด้านรับ และไซด์โทน (Side Tone : การที่เสียงของผู้พูดสามารถได้ยินในส่วนของผู้ฟังเพื่อให้ทราบได้ว่าเราควรจะพูดดังค่อยขนาดไหนในการติดต่อ) มีส่วนวงจรชดเชยผลอันเนื่องมาจากความยาวของสายว่างสัญญาณ (Line Length Compensation) ที่อัตราขยายเปลี่ยนแปลงตามกระแสภายในลูปรวมทั้งวงจร

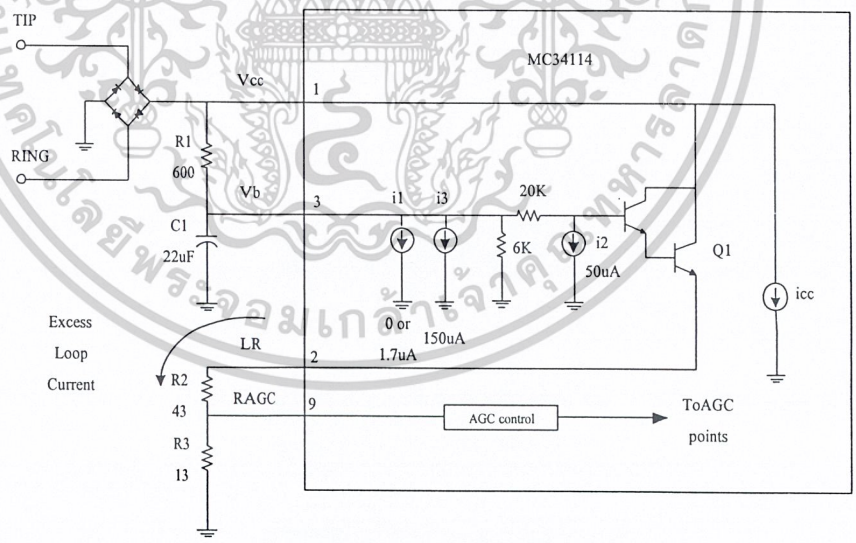
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงบล็อกไดอะแกรมภายใน MC34114

3.5.1 วงจรเชื่อมต่อกับไฟตรง

วงจรเชื่อมต่อกับไฟตรง (ขา 1, 2, 3) จะกำหนดคุณสมบัติของไฟตรงจากกระแสในลูป จากรูปที่ 3.6 ระดับแรงดันไฟตรงที่ V_{CC} ถูกจำกัดโดยยกระดับแรงดันของขา 1 กับ ขา 2 บวกระดับแรงดันไฟตรงคร่อม R_2 และ R_3 ไอซี MC34114 ต้องการ I_{CC} เป็นกระแสไบอัสภายใน ซึ่งปกติมีค่าประมาณ 10 มิลลิแอมป์ เราสามารถที่จะลดกระแส I_{CC} หากจำเป็นโดยการเพิ่มค่า R_{12}



รูปที่ 3.7 แสดงวงจรสมมูลย์ของการอินเตอร์เฟสกับคู่สายโทรศัพท์

ในระหว่างการพูดและการส่งสัญญาณพัลส์ ตัวกำเนิดกระแส I_1 ไม่ทำงานการยกระดับแรงดันจะตกลงไป เนื่องมาจากขา B และขา E ของทรานซิสเตอร์ Q_1 (ประมาณ 1.4 โวลต์) คร่อมความต้านทาน 20 กิโลโอห์ม และแรงดันตกคร่อม R_1 ซึ่งทำให้ V_{CC} จะเปลี่ยนแปลงตั้งแต่ 0.15 ไปจนถึงประมาณ 1.0 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกระแสลูปที่มาจากทิปกับริงมีค่ามากเกินกว่า I_{CC} จะต้องการกระแสที่เกินจะไหลผ่าน Q_1 และ R_2 เพื่อให้เป็นไปตามคุณสมบัติของความสัมพันธ์ระหว่างกระแสและแรงดัน

ในการส่งสัญญาณแบบโทน แหล่งจ่ายกระแส I_1 ทำงาน ทำให้กระแสไหลบผ่าน R_1 เพิ่มขึ้น 1.7 มิลลิแอมป์ ยกระดับแรงดันขึ้นไปอีกประมาณ 1.0 โวลต์ (เมื่อ R_1 มีค่า 600 โอห์ม) คุณสมบัติพิเศษนี้เป็นการประกันได้ว่า เมื่อกระแสลูปมีค่าน้อย จะมีแรงดันที่ V_{CC} มากพอสำหรับสัญญาณ DTMF และแหล่งจ่ายไฟ V_{DD} จะสามารถจ่ายแรงดันที่เพียงพอไปให้ส่วนที่แป้นกดสัญญาณภายนอก กระแส I_{CC} ในการทำงานแบบนี้จะเพิ่มขึ้นไปประมาณ 1.3 มิลลิแอมป์

ความต้านทาง R_1 ใช้ได้ตั้งแต่ 100 ไปจนถึง 1800 โอห์ม ถ้าใช้ค่าที่มากเกินไป กระแสที่ไหลไปยัง V_R จะมีค่าไม่เพียงพอ แต่ถาน้อยเกินไป การกรองที่ V_R จะไม่เป็นผล ถึงแม้ว่าจะมีการเพิ่ม C_1 ก็ตาม (สัญญาณเสียงจะถูกกรองโดย V_R)

แรงดันตกคร่อม R_3 เป็นตัวควบคุมการทำงานของ AGC (เป็นส่วนที่ชดเชยอันเนื่องมาจากผลของความยาวสายส่ง) เมื่อความต้านทานที่ตกคร่อม R_{AGC} เพิ่มขึ้นประมาณ 0.4 โวลต์ ไปเป็น 1.2 โวลต์ ส่วนควบคุมการทำงานของ AGC จะเปลี่ยนแปลงอัตราการขยายของ AGC ตั้งแต่ 1.0 ไปจนถึง 0.5 (ซึ่งจะลดอัตราการขยายของส่วนรับและส่งไปประมาณ 6 dB)

ค่าของ R_2 และ R_3 สามารถที่จะเปลี่ยนแปลงได้ เมื่อมีการเพิ่มเติมวงจรที่กระแสลูป อาทิเช่น ไมโครคอนโทรลเลอร์ต่างๆ หรือเพื่อเปลี่ยนแปลงจุดเริ่มต้นการทำงานของวงจร AGC หากไม่มีการใช้งาน AGC ควรจะต่อขา 9 เข้ากับกราวนด์ เพื่อให้ได้อัตราการขยายสูงสุดหรือเข้ากับ V_R เพื่ออัตราขยายต่ำสุด

3.5.2 ตัวจ่ายแรงดันคงที่

ไอซี MC34114 มีตัวจ่ายแรงดันคงที่ 2 ตัว เพื่อจ่ายแรงดันให้แก่วงจรภายในและภายนอก ตัวจ่ายแรงดันคงที่ V_R จ่ายแรงดัน 1.7 โวลต์ ที่แรงดันสูงสุด 500 ไมโครแอมป์ ซึ่งผลที่ได้นี้จะนำไปไบอัสขา 10 (TX1) และไบอัสไมโครโฟน โดยปกติ V_R มีค่าน้อยกว่า V_{CC} ประมาณ 0.3 โวลต์ เมื่อ - V_{CC} มีค่าน้อยกว่า 2.0 โวลต์

ตัวจ่ายแรงดันคงที่ V_{DD} จ่ายแรงดัน 3.3 โวลต์ที่กระแสสูงสุด 1.0 มิลลิแอมป์ ในขณะใช้พูดแบบปกติ และกระแสสูงสุด 2.5 มิลลิแอมป์ ในการส่งสัญญาณแบบพัลส์หรือโทน ปกติเราใช้ V_{DD} ในการจ่ายพลังงานให้แก่แป้นกดที่อยู่ภายนอก รวมทั้งวงจรอื่นที่ต่ออยู่ด้วยกัน ปกติ V_{DD} จะมีค่าน้อยกว่า V_{CC} ประมาณ 0.5 โวลต์

V_{DD} เป็นตัวจ่ายกระแสคงที่แบบขนาน ซึ่งจะเปลี่ยนไปเป็นค่าความต้านทานสูง โดยหน่วยความจำของวงจรแป้นกด เมื่อ V_{CC} มีค่าเป็น 0 กระแสรั่วไหลจะมีค่า 0.02 ไมโครแอมป์ เมื่อป้อนค่าแรงดันไม่เกิน 6 โวลต์ เข้าที่ V_{DD} โดยที่ขา 17 เปิดวงจร หรือต่อกับ V_{DD} หากขา 17 ต่อลงกราวนด์ กระแสหลายร้อยไมโครแอมป์จะไหลเข้า V_{DD} 1และไหลลงกราวนด์ที่ขา 17

3.5.3 วงจรขยายสัญญาณจากไมโครโฟน

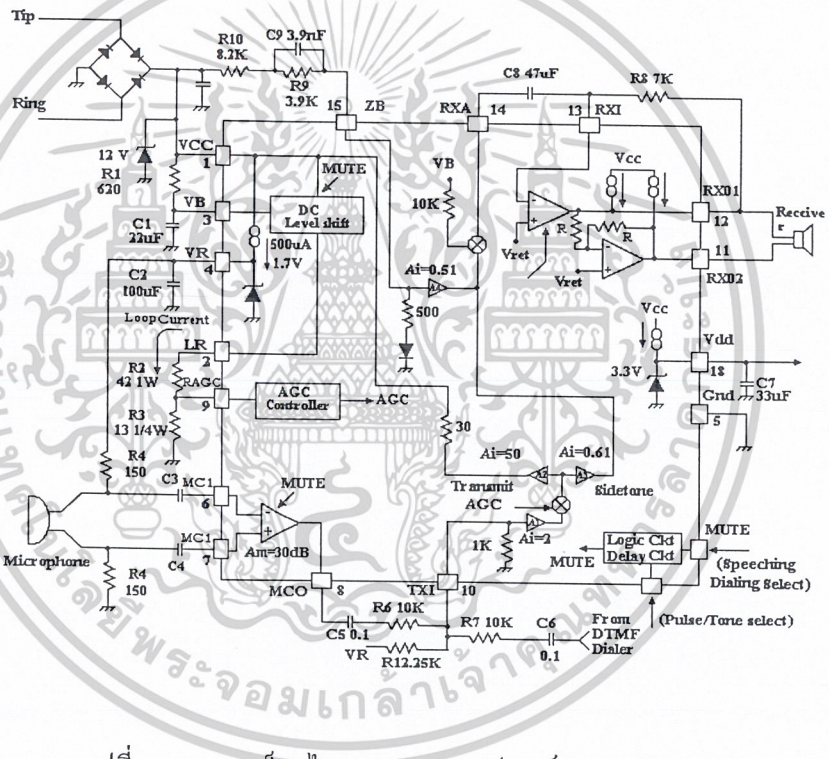
วงจรขายสัญญาณจากไมโครโฟน (ขา 6, 7, 8) มีสัญญาณเข้าแบบผลต่าง (Differential) สัญญาณออกแบบซิงเกิลเอนด์ และอัตราการขยายภายในคงที่ +30 dB เอาต์พุตตรงเฟสกับ MC2 และกลับเฟสกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ MC1 มีความต้านทาน 20 กิโลโอห์ม และแมตช์ซึ่งเป็นอย่างดีเพื่อ CMRR (Common Mode Rejection) ที่2สูงประมาณ 26 dB เพื่อที่จะมีการขจัดสัญญาณจากการเหนี่ยวนำจากสายนำสัญญาณที่ไม่ต้องการ (CMRR มีค่าสูง) ไมโครโฟนจึงมีการไบอัสจากความต้านทานที่มีค่าเท่ากัน

เอาต์พุต (MC0) มีแรงดันไบอัสตรงอยู่ประมาณ 1.1 โวลต์ (เมื่อ V_{CC} มีค่ามากกว่าประมาณ 3.0 โวลต์) มีอัตราการแกว่งประมาณ 2.0 โวลต์ (แกว่ง 500 มิลลิแอมป์ เมื่อ V_{CC} มีค่า 1.2 โวลต์) เอาต์พุตอิมพีแดนซ์มีค่าประมาณ 270 โอห์ม และมีกระแสสูงสุดประมาณ 160 ไมโครแอมป์ ที่ 5% ของ THD (Total Harmonic Distortion)

เมื่อ MC34114 อยู่ในระหว่างการส่งสัญญาณหมุน วงจรขยายไมโครโฟนจะถูกลดกำลังการส่งลงไปประมาณ 70 dB (300 – 400 กิโลเฮิรตซ์) ซึ่งเพียงพอในการหยุดการทำงานของไมโครโฟนระดับ8 แรงดันไฟตรงที่ MC0 มีค่าประมาณ 80 มิลลิโวลต์ เมื่อถูกลดกำลังส่งลง



รูปที่ 3.8 แสดงบล็อกไดอะแกรมและอุปกรณ์ภายนอก MC34114

3.5.4 วงจรในการส่งสัญญาณ

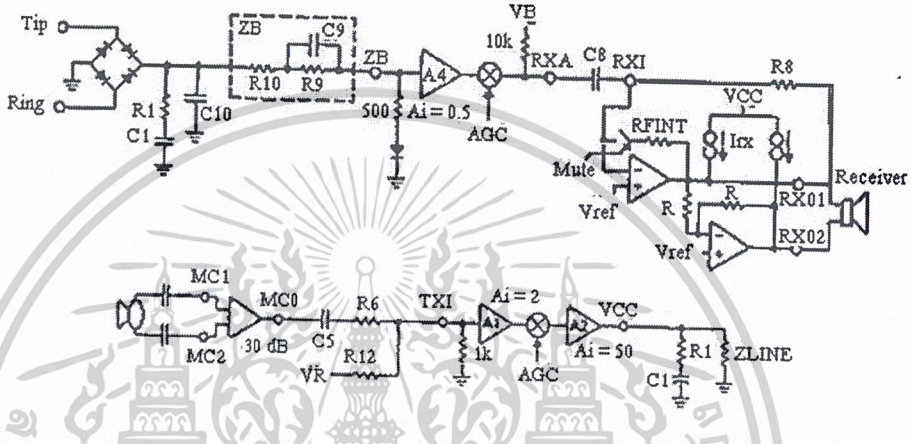
วงจรที่ใช้ในการส่งสัญญาณออกไปมีอุปกรณ์ดังแสดงในรูปเอาต์พุตที่ MC0 ถูกเปลี่ยนไปเป็นกระแสเข้า TX1 โดย C_5, R_6 และความต้านทานภายในของ TX1 1 กิโลโอห์ม A_1 และ A_2 เป็นอุปกรณ์ขยายกระแสที่มีอัตราขยายรวมกันเป็น 100 AGC ที่เข้ามามีค่าเป็น 1 เมื่อมีกระแสลูปน้อยและลดลงเป็น 0.5 เมื่อกระแสลูปมีค่ามากขึ้น ดังนั้น จะทำให้อัตราขยายจาก TX1 ไปจนถึง V_{CC} มีค่าตั้งแต่ 100 ถึง 50 เป็นผลทำให้กระแส V_{CC} กระทำต่อ R_1 และอิมพีแดนซ์ของสายส่ง (ประมาณ 600 โอห์ม) ก่อให้เกิดแรงดันที่ V_{CC} และเช่นกันที่ขั้ว ทิปและริง ระดับแรงดันระหว่างขา MC1 – MC2 และขั้วทิปและริงมีค่าตามสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_{TX} = \frac{\left(A_m \times 100 \times AGC \times \frac{R_1}{Z_{LINE}} \right)}{(R_6 + 1000)}$$

เมื่อ A_m เป็นอัตราขยายของอุปกรณ์ขยายไมโครโฟน (31.3 V/V) ที่กระแสลูปมีค่าน้อย ๆ G_{RX} เป็น 84 V/V (38.5 dB) และมีค่าเป็น 42 V/V (32.5 dB) ที่กระแสลูปมีค่ามากๆ

สัญญาณที่ V_{CC} กลับเฟสกันกับสัญญาณที่ TX1 แต่มีเฟสเดียวกับที่ MC1



รูปที่ 3.9 แสดงเส้นทางของสัญญาณทางด้านรับและด้านส่ง

3.5.5 วงจรในการรับสัญญาณ

วงจรที่ใช้รับสัญญาณเข้ามามีอุปกรณ์ดังแสดงในรูปที่ 3.8 ซึ่งโดยปกติมีค่า 600 โอห์ม จะเป็นตัวกำหนดจุดสิ้นสุดของสายส่ง (Return Loss) ของสัญญาณที่ส่งเข้ามาจากขั้วทิปและริง สัญญาณที่ได้รับจะสร้างกระแสไฟสลับผ่าน ZB Network (Balance Impedance Network) และความต้านทาน 500 โอห์ม ที่ขา ZB A_4 จะลดกระแสลงครึ่งหนึ่งแล้วส่งต่อไปให้ AGC แล้วผ่าน C_8 ไปยัง RX1 (จุดรวมอัตราการขยาย ซึ่งถ้า C_8 มีค่ามาก RXA จะเปรียบเป็นกราวนด์เสมือนและไม่มีกระแสสลับไหลผ่านความต้านทานภายใน 10 กิโลโอห์ม) แรงดันที่ขา RX01 ถูกกำหนดโดยกระแส C_8 และความต้านทานป้อนกลับ R_8 ออปแอมป์ตัวที่สอง (ที่ขา RX02) มีการกำหนดไว้แล้วว่าให้มีการขยายแบบกลับขั้ว และมีการขยายเป็น 1 (Inverting Unity Gain) อัตราการขยายแรงดันจากขั้วทิปและริงไปยัง RX01 – RX02 มีค่าตามสมการต่อไปนี้

$$G_{RX} = \frac{(R_Z \times AGC)}{ZB + 500}$$

เมื่อ

$$ZB = R_{10} + (R_9 // C_9) = R_{10} + R_9$$

เมื่อใช้ค่าอุปกรณ์ตามรูป อัตราการขยายจะมีค่าประมาณ 0.495 V/V (-6.1 dB) เมื่อกระแสในลูปมีค่าน้อย และอัตราการขยายกลายเป็น 0.25 V/V (-12 dB) เมื่อกระแสไหลในลูปสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ MC34114 อยู่ในระหว่างการส่งสัญญาณเลขออก (MUTE มีค่าเป็น 0) อัตราการขยายของวงจรภาครับจะลดลงด้วย เพราะมีค่า R_{FINT} ที่มีค่า 1.0 กิโลโอห์มขนานกับ R_g อัตราการลดลงของสัญญาณจะมีค่าดังสมการต่อไปนี้

$$G_{REM} = 20 \times log\left(\frac{(R_g + R_{FINT})}{R_{FINT}}\right)$$

เมื่อขา MUTE กลับไปสู่สภาวะ 1 อีกครั้งจะมีการหน่วงเวลาประมาณ 11 msec ก่อนที่ความต้านทาน จะถูกทำให้กลับไปเป็นสภาวะเดิม เพราะเหตุที่ว่าจะได้ป้องกันสัญญาณทรานเซียนส์อันเนื่องมา จากการส่งสัญญาณพัลส์ อันเป็นเหตุให้เกิดเสียงคลิกขึ้นที่หูฟัง

แรงดันไบอัสที่ขา RX1, RX01 และ RX02 มีค่าประมาณ 0.65 โวลต์ กระแสไบอัสที่ขา RX1 มีค่าประมาณ 50 นาโนแอมป์ แรงดันสูงสุดที่ RX01 และ RX02 อยู่ในเทอมของความต้านทานของหูฟังและกระแส I_{RX} โดยค่านี้หาได้จากสมการ

$$I_{RX} = \frac{(V_R \times 50 \times AGC)}{(R_{12} + 100)}$$

3.5.6 วงจรขจัดไซด์โทน

การขจัดไซด์โทนสามารถทำได้โดยการนำเอาตัวขยายกระแส A_3 มาสร้างสัญญาณที่คล้ายคลึงกับสัญญาณทางด้านส่ง แล้วนำมาขจัดไซด์โทนที่ผ่านมาทางด้าน ZB และ A_4 เพื่อที่จะได้การขจัดสัญญาณที่สมบูรณ์ (ไม่มีกระแสสลับออกมาทาง RXA) จำเป็นที่จะต้องให้ ZB มีค่าตามสมการ

$$ZB = \left(40 \times \left(R_1 // Z_{LINE}\right) - 500\right)$$

ซึ่ง ZB เป็นวงจรที่ประกอบขึ้นด้วย R9, R10 และ C9 และ ZB เป็นความต้านทานทาง AC ของสายส่ง อุปกรณ์ที่มีปฏิกิริยาตอบสนองต่อความต้านทานของสายส่ง สามารถชดเชยได้ด้วยการใช้วงจร ZB ที่มีปฏิกิริยาตอบสนองที่เปรียบเทียบกันได้ ในรูปที่ 3.8 C9 จะเป็นตัวชดเชยการเลื่อนของเฟสอันเนื่องมาจากสายส่ง

เนื่องจากปกติในสายส่งสัญญาณที่เชื่อมต่ออยู่ระหว่าง ชุมสายกับเครื่องรับโทรศัพท์ จะมีค่าความต้านทาน ตัวเก็บประจุ และขดลวดเหนี่ยวนำอยู่ โดยเฉลี่ยแล้วทุกๆ ระยะ 1 ไมล์ ที่เพิ่มขึ้นของสายส่งจะเสมือนว่ามีตัวเก็บประจุค่าประมาณ 0.07 μF ต่อคร่อมอยู่ระหว่างสายส่ง และมีความต้านทาน 42 โอห์ม กับขดลวดเหนี่ยวนำ 1 mH ต่ออนุกรมกันอยู่ ดังนั้นจึงต้องมีวงจรที่สามารถรับรู้ค่าความผิดพลาดเหล่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.7 การเชื่อมต่อของสัญญาณลอจิก

ขาอินพุตลอจิก 2 ขา ของ MC34114 ถูกใช้ในการเปลี่ยนแปลงโหมดการทำงานดังตารางที่ 3.3 ต่อไปนี้

Mute	Ms	Mode
High	X	Speech
Low	High	Pulse Dialing
Low	Low	Tone Dialing

ตารางที่ 3.3 แสดงความสัมพันธ์ระหว่างโหมดการทำงานกับลอจิกของ MUTE และ MS

ค่าลอจิก 1 ของขา MUTE มีค่าระหว่าง V_{DD} - 0.5 จนถึง V_{DD} ส่วนค่าลอจิก 0 ของขา MUTE มีค่าระหว่าง 0–1 โวลต์ การเปลี่ยนแปลงลอจิกต้องมากกว่าเทรสโฮลต์ 2.3 โวลต์ เมื่อขา MUTE เปลี่ยนไปเป็นค่า 0 หรือขา MS เกิดการเปลี่ยนแปลงลอจิก การเปลี่ยนแปลงภายในวงจรจะเกิดขึ้นภายใน 10 μsec แต่ถ้าขา MUTE เปลี่ยนไปเป็นค่า 1 จะเปลี่ยนแปลงหลังจากมีการหน่วง 11 msec เนื่องมาจากมีการป้องกันสัญญาณทรานเซียนส์ ที่เกิดขึ้นจากสัญญาณพัลส์อันจะทำให้ได้ยินเสียงคลิกที่หูฟัง

ขา MS จะทำงานเมื่อขา MUTE มีลอจิก 0 หน้าที่ที่แท้จริงของ MS ก็คือ การใช้ค่าแรงดันเลเวลวิฟท์แก่ V_{CC} และ LR ในการส่งสัญญาณแบบโทน ค่าลอจิก 0 มีค่าระหว่าง 0 – 0.3 โวลต์ ค่าลอจิก 1 มีค่าระหว่าง 2 - V_{DD} โวลต์ ถ้าเทรสโฮลต์มีค่า 0.75 โวลต์ เมื่อไม่มีการเลือกการทำงานระหว่างการส่งแบบพัลส์ หรือโทนให้ต่อลงกราวนด์ หรือ V_{DD} ห้ามไม่ให้ปล่อยลอยไว้เป็นอันขาด

เมื่ออยู่ในสภาวะออนฮุก และมีแรงดันไม่เกิน 6 โวลต์ ต่ออยู่กับ MUTE กระแสรั่วไหล 0.02 ไมโครแอมป์จะไหลเข้าขา MUTE และ V_{DD} แรงดันเท่ากัน แต่ถ้าแรงดันมีไม่เท่ากันแล้วกระแสจะไหลผ่านตัวต้านทานภายใน และไดโอด หากมีแหล่งจ่ายไฟเพื่อคงหน่วยความจำของวงจรที่เป็นกดต่ออยู่และปรากฏว่ามีแรงดันของแหล่งจ่าย เพื่อคงหน่วยความจำของวงจรเป็นกดที่ V_{DD} ขา MUTE จะต้องอยู่กับ V_{DD} หรือกราวนด์มิฉะนั้นแล้วกระแส 100–200 ไมโครแอมป์ จะไหลผ่าน V_{DD} และออกทางขา MUTE

V_{CC} มีค่าเท่ากับ 0 และมีแรงดันไม่เกิน 6 โวลต์ ต่ออยู่ที่ขา MS จะมีกระแสรั่วไหลเกิดขึ้น 0.01 ไมโครแอมป์ ตลอดเวลาที่ขา MUTE ปล่อยลอยหรือต่อกับ V_{DD} หากขา MUTE ต่อลงกราวนด์จะมีความต้านทาน 3.5 กิโลโอห์มเกิดขึ้นระหว่าง MS และ MUTE

หาก V_{CC} < 1.5 โวลต์ ขา MUTE จะไม่ทำงานเป็นเหตุให้ MC34114 อยู่ในการทำงานโหมดสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบในส่วนของซอฟท์แวร์ (Software)

การออกแบบในส่วนของซอฟท์แวร์ (Software) จะมีอยู่ 2 ส่วน คือ โปรแกรมที่ตัวไมโครคอนโทรลเลอร์ MCS-51 และโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอ็นท์หรือเซิร์ฟเวอร์

3.6 การออกแบบโปรแกรมที่ตัวไมโครคอนโทรลเลอร์ MCS-51

การออกแบบโปรแกรมเราจะใช้หลักการของการโพลลิ่ง (Polling) โปรแกรมเพื่อทำการตรวจเช็คติดต่อ โดยจะมีส่วนของโปรแกรมหลัก และโปรแกรมย่อย ซึ่งโปรแกรมย่อยมี 2 ส่วนด้วยกัน คือ โปรแกรมตรวจสอบการโทรศัพท์เข้าเพื่อติดต่อกับไคลเอ็นท์ และโปรแกรมตรวจสอบการติดต่อจากเซิร์ฟเวอร์เพื่อโทรออก ดังนี้

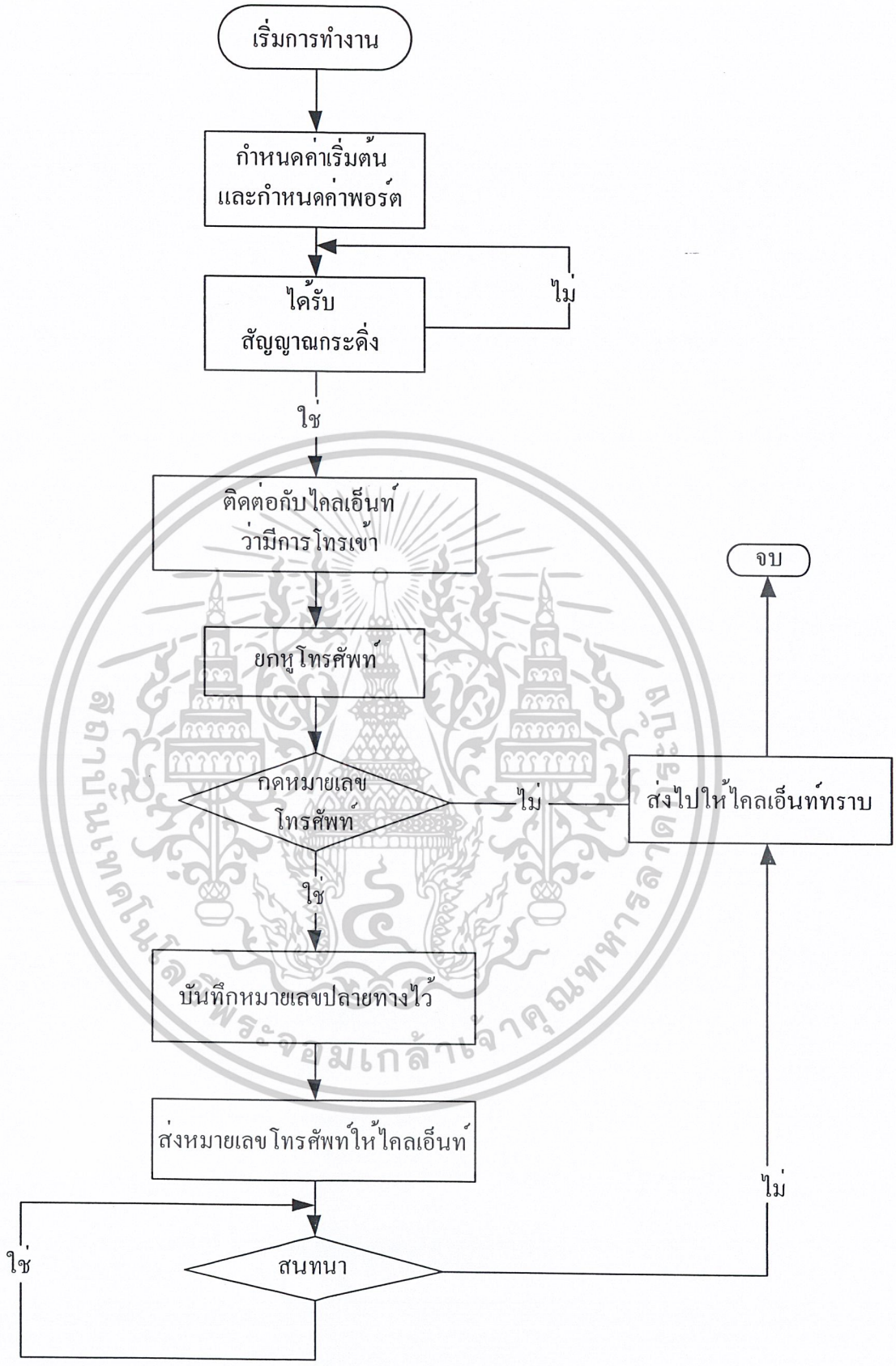
3.6.1 การออกแบบโปรแกรมตรวจสอบการโทรศัพท์เข้าเพื่อติดต่อกับไคลเอ็นท์

เมื่อโปรแกรมถูกเรียกจากโปรแกรมหลักก็จะทำการตรวจสอบการโทรศัพท์เข้า โดยจะทำการตรวจสอบสัญญาณกระดิ่ง ถ้าไม่มีการโทรเข้าก็จะจบโปรแกรมย่อยนี้ แต่ถ้ามีการโทรศัพท์เข้ามาหรือมีสัญญาณกระดิ่งเข้ามาก็จะส่งไปให้ไคลเอ็นท์ทราบ หลังจากนั้นก็จะยกหูโทรศัพท์ จากนั้นก็จะรับหมายเลขจากผู้ที่โทรศัพท์เข้ามา ซึ่งในส่วนนี้ก็จะมีการตรวจเช็คว่าที่การกดเลขหมายหรือไม่ ถ้าไม่มีการกดก็จะจบโปรแกรมย่อยนี้ แต่ถ้ามีการกดเลขหมายเลขโทรศัพท์ก็จะรับหมายเลขเข้ามา ส่งไปให้กับ ไคลเอ็นท์ หลังจากนั้นส่วนของโปรแกรมย่อยนี้ก็จะตรวจเช็คการสนทนาว่าสนทนากันหรือไม่จนกว่าจะหยุดสนทนา ก็จะส่งไปให้ไคลเอ็นท์ทราบแล้วก็จะจบโปรแกรมย่อยนี้ ซึ่งแสดงไว้ในผังการทำงาน ดังรูปที่ 3.10

3.6.2 การออกแบบโปรแกรมตรวจสอบการติดต่อจากเซิร์ฟเวอร์เพื่อโทรออก

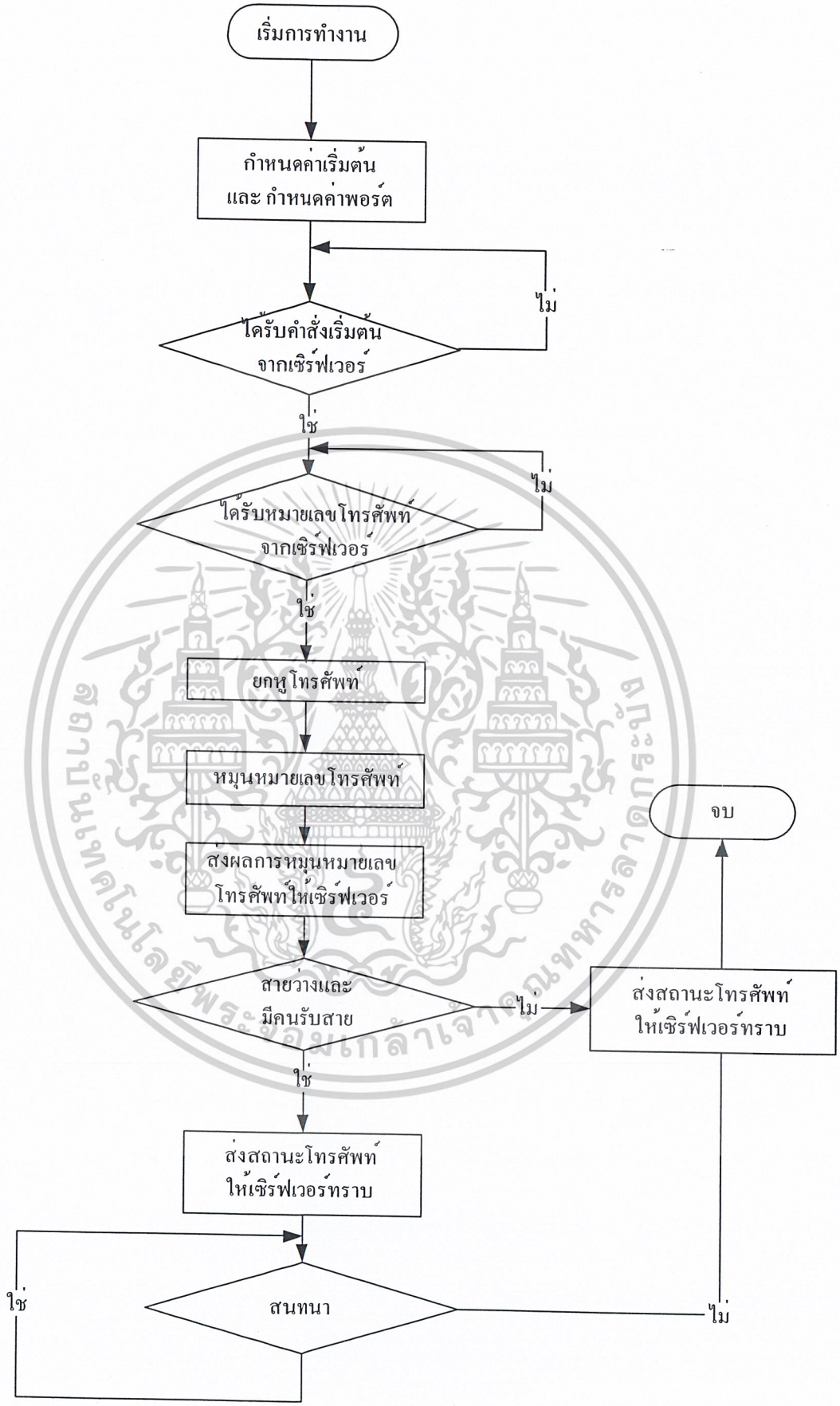
เมื่อโปรแกรมถูกเรียกจากโปรแกรมหลัก ก็จะทำการตรวจสอบการติดต่อจากเซิร์ฟเวอร์ ถ้าไม่มีการติดต่อจากเซิร์ฟเวอร์ก็จะจบโปรแกรมย่อยนี้ แต่ถ้ามีการติดต่อจากเซิร์ฟเวอร์ ก็จะรอรับหมายเลขโทรศัพท์จากเซิร์ฟเวอร์ เมื่อได้รับหมายเลขโทรศัพท์จากเซิร์ฟเวอร์ แล้วก็จะทำการยกหูโทรศัพท์ จากนั้นจะทำการหมุนหมายเลขโทรศัพท์ หลังจากนั้นก็จะตรวจเช็คว่าสายว่างหรือไม่ ถ้าสายไม่ว่างก็ส่งไปให้เซิร์ฟเวอร์ทราบแล้วจบโปรแกรมย่อยนี้ แต่ถ้าสายว่างก็สามารถสนทนากันได้ หลังจากนั้นก็จะตรวจเช็คการสนทนาว่าสนทนากันหรือไม่ จนกว่าจะหยุดสนทนา ก็จะส่งไปให้เซิร์ฟเวอร์ทราบ แล้วก็จะจบโปรแกรมย่อยนี้ ซึ่งแสดงไว้ในผังการทำงานดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงผังการทำงานของโปรแกรมตรวจสอบการโทรศัพท์เข้าเพื่อติดต่อกับไคลเอ็นท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงผังการทำงานของโปรแกรมตรวจสอบการติดต่อจากเซิร์ฟเวอร์เพื่อโทรออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การออกแบบโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอ็นท์หรือเซิร์ฟเวอร์

โปรแกรมที่ออกแบบในโครงงานนี้ สร้างขึ้นโดยใช้ Visual Basic 6 เป็นเครื่องมือในการสร้าง ซึ่งแต่ละโปรแกรมที่สร้างขึ้นโดย Visual Basic 6 จะเรียกว่าโปรเจค (Project) โดยในหนึ่งโปรเจคจะประกอบด้วยไฟล์ สกุล .frm ที่ทำหน้าที่เก็บข้อมูลส่วนที่เป็นหน้าต่างของโปรเจคนั้น และไฟล์สกุล .VBP ทำหน้าที่เป็นไฟล์หลักในการทำงานของโปรเจค นอกจากนี้ในหนึ่งโปรเจค อาจจะประกอบด้วยไฟล์สกุล .bas ที่เป็นโมดุล และไฟล์อื่นๆ ที่ช่วยการทำงานของโปรเจค

การออกแบบโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอ็นท์หรือเซิร์ฟเวอร์ โดยการส่งข้อมูลระหว่างไคลเอ็นท์และเซิร์ฟเวอร์ จะใช้โปรโตคอล TCP/IP โดยจะใช้พอร์ตหมายเลข 5678 เป็นพอร์ตส่งข้อความควบคุม ส่วนพอร์ตหมายเลข 1000 และ 2000 เป็นพอร์ตที่ใช้ส่งข้อมูลเสียง

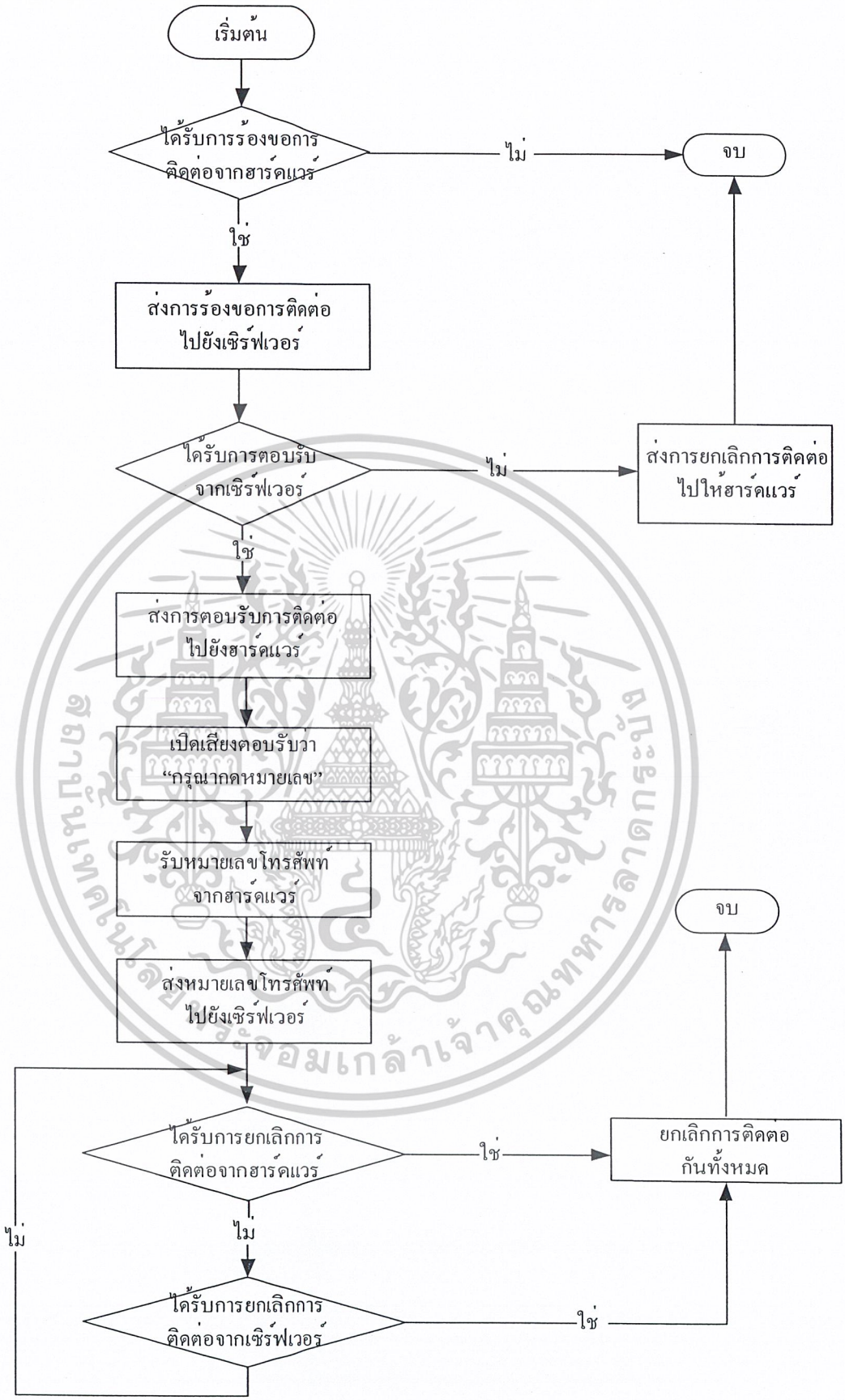
3.7.1 การออกแบบโปรแกรมไคลเอ็นท์

โปรแกรมไคลเอ็นท์จะถูกเรียกก็ต่อเมื่อได้รับการร้องขอการติดต่อจากฮาร์ดแวร์ หลังจากนั้นโปรแกรมไคลเอ็นท์ส่งการร้องขอการติดต่อไปยังเซิร์ฟเวอร์ แล้วก็รอรับการตอบรับจากเซิร์ฟเวอร์ ถ้าไม่ได้ก็จะยกเลิกการติดต่อไปยังฮาร์ดแวร์ แต่ถ้าได้รับการตอบรับ ก็จะส่งการตอบรับการติดต่อไปยังฮาร์ดแวร์ซึ่งในกระบวนการนี้ จะทำการต่อพอร์ตส่งข้อความควบคุม และพอร์ตเสียงทั้งสองพอร์ตที่แอดเดรสของเซิร์ฟเวอร์ หลังจากนั้นก็เปิดไฟล์เสียงตอบรับว่า “กรุณากดหมายเลขปลายทาง” ถัดมาก็จะรับเลขหมายโทรศัพท์จากฮาร์ดแวร์ส่งไปยังเซิร์ฟเวอร์อีกทีหนึ่ง จากนั้นก็จะทำการตรวจเช็คว่าได้รับการยกเลิกการติดต่อจากฮาร์ดแวร์หรือจากเซิร์ฟเวอร์หรือไม่ ถ้าไม่ได้รับการยกเลิกการติดต่อแสดงว่ายังคงสนทนากันอยู่ แต่ถ้าได้รับการยกเลิกการติดต่อก็จะยกเลิกการติดต่อทั้งหมดก็คือยกเลิกการต่อพอร์ตทั้งหมดที่ต่อไว้ สุดท้ายก็จะจบโปรแกรมไคลเอ็นท์นี้ ดังแสดงผังการทำงานดังรูปที่ 3.12

3.7.2 การออกแบบโปรแกรมเซิร์ฟเวอร์

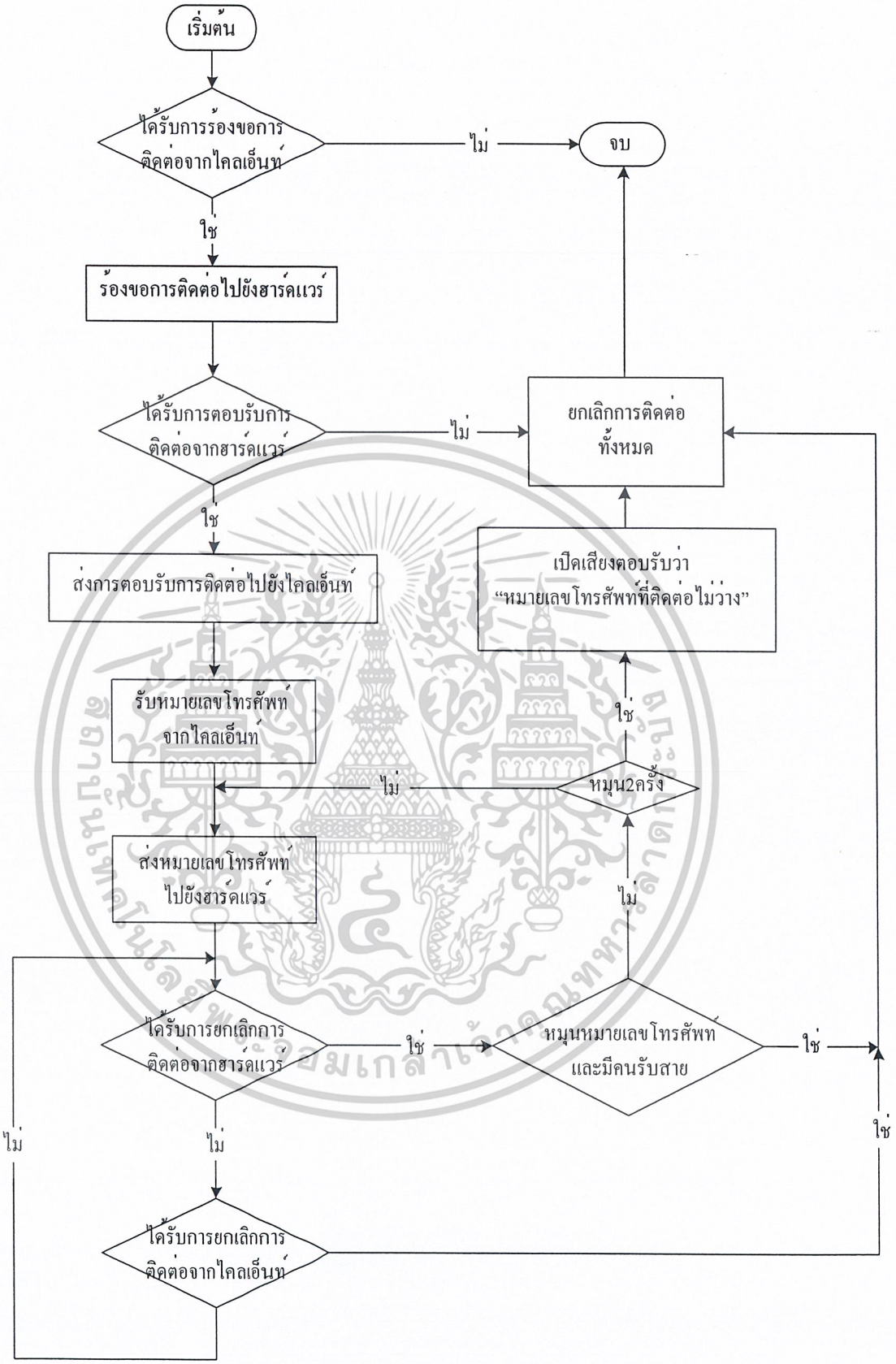
โปรแกรมเซิร์ฟเวอร์จะถูกเรียกก็ต่อเมื่อได้รับการร้องขอการติดต่อจากไคลเอ็นท์ หลังจากนั้นโปรแกรมเซิร์ฟเวอร์ส่งการร้องขอการติดต่อไปยังฮาร์ดแวร์ แล้วก็จะรอการตอบรับจากฮาร์ดแวร์ ถ้าไม่ได้รับการตอบรับก็จะยกเลิกการติดต่อไปยังไคลเอ็นท์ แต่ถ้าได้รับการตอบรับจากฮาร์ดแวร์ก็ส่งการตอบรับไปยังไคลเอ็นท์ และทำการต่อพอร์ตส่งข้อความควบคุมและพอร์ตเสียงทั้งสองพอร์ต หลังจากนั้นก็รับหมายเลขโทรศัพท์ จากไคลเอ็นท์แล้วส่งไปยังฮาร์ดแวร์อีกทีหนึ่ง ถัดมาก็ทำการตรวจเช็คการยกเลิกจากฮาร์ดแวร์ซึ่งในกระบวนการนี้จะมีการตรวจจับเวลาในการโทร เพื่อที่ตรวจเช็คว่าไม่มีคนรับสายหรือว่าสายไม่ว่างหรือไม่ ถ้าไม่มีคนรับสายหรือว่าสายไม่ว่างก็จะโทรอีกครั้งหนึ่ง และถ้ายังไม่สามารถต่อสายได้อีกเปิดไฟล์เสียงตอบรับว่า “หมายเลขโทรศัพท์ที่ติดต่อไม่ว่าง” และยกเลิกการติดต่อทั้งหมด แต่ถ้าสามารถโทรได้แล้วสามารถสนทนากันได้ก็ตรวจเช็คการยกเลิกการติดต่อจากฮาร์ดแวร์หรือจากไคลเอ็นท์จนกว่าจะได้รับการยกเลิกการติดต่อก็จะยกเลิกการติดต่อทั้งหมด ดังแสดงผังการทำงานดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



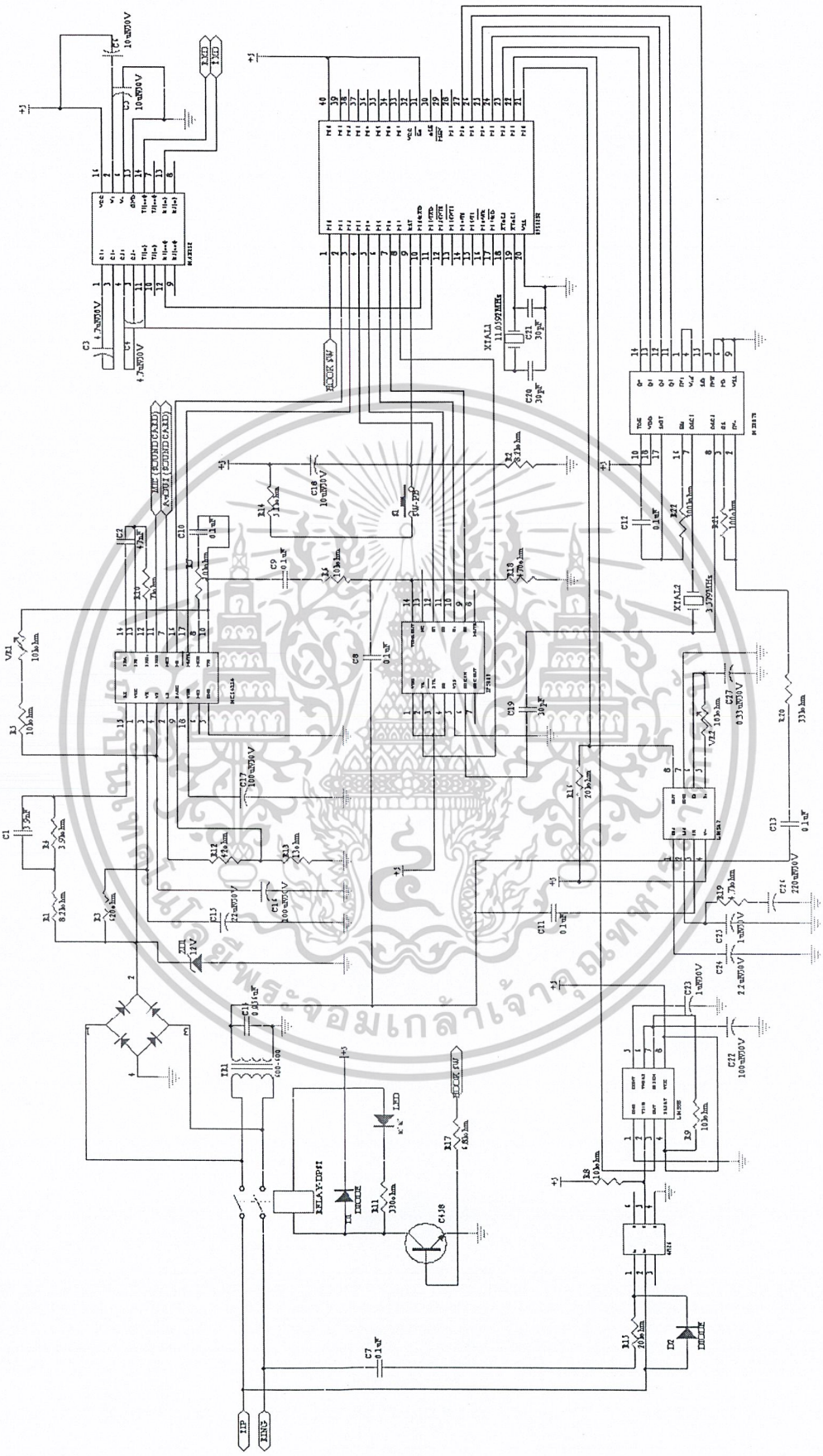
รูปที่ 3.12 แสดงผังการทำงานของโปรแกรมไคลเอ็นท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



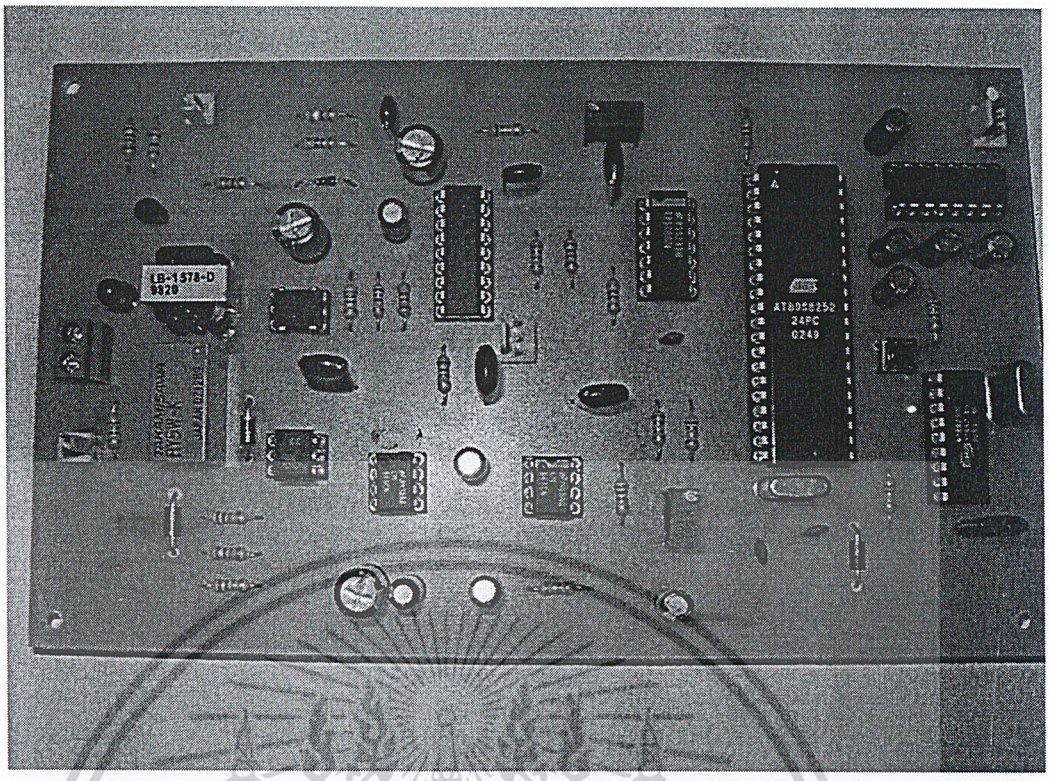
รูปที่ 3.13 แสดงผังการทำงานของโปรแกรมเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

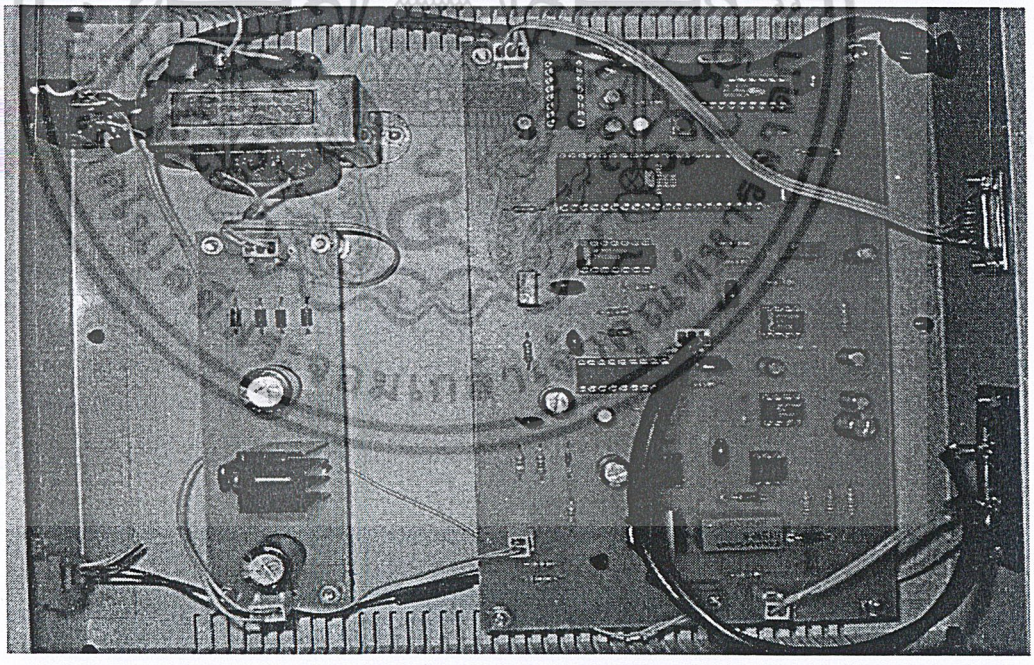


รูปที่ 3.14 แสดงวงจรรวมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

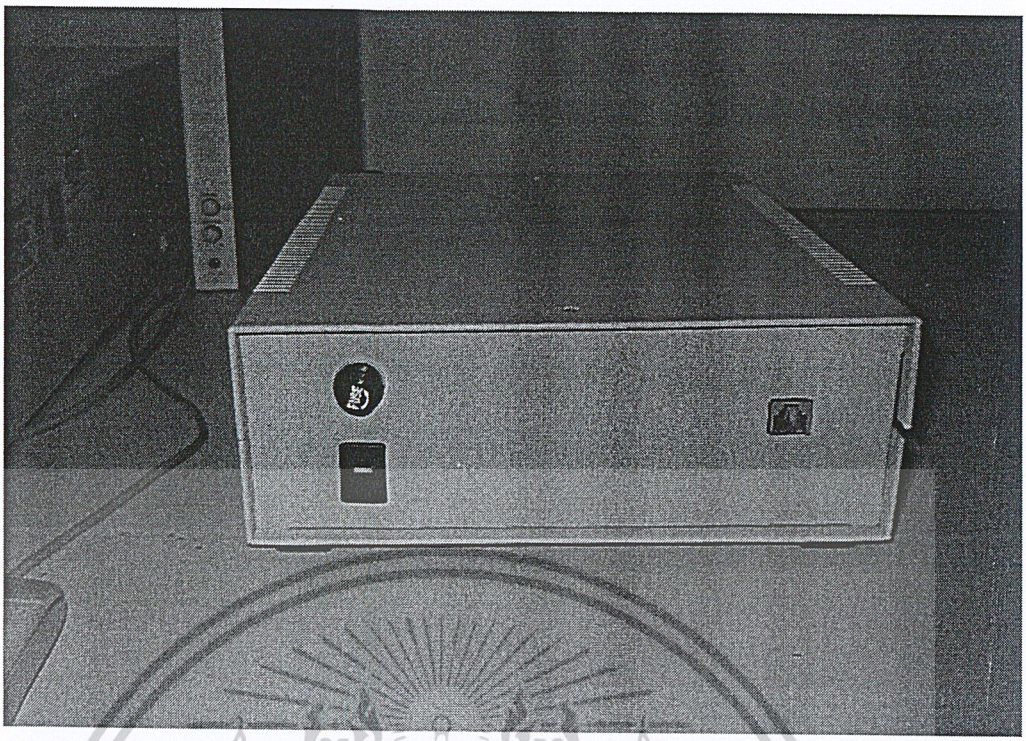


รูปที่ 3.15 แสดงภาพวงจรที่ลงอุปกรณ์เรียบร้อยแล้ว



รูปที่ 3.16 แสดงภาพรวมของวงจรทั้งหมดที่ลงอุปกรณ์เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงภาพด้านหน้าของอุปกรณ์ที่พร้อมใช้งานได้

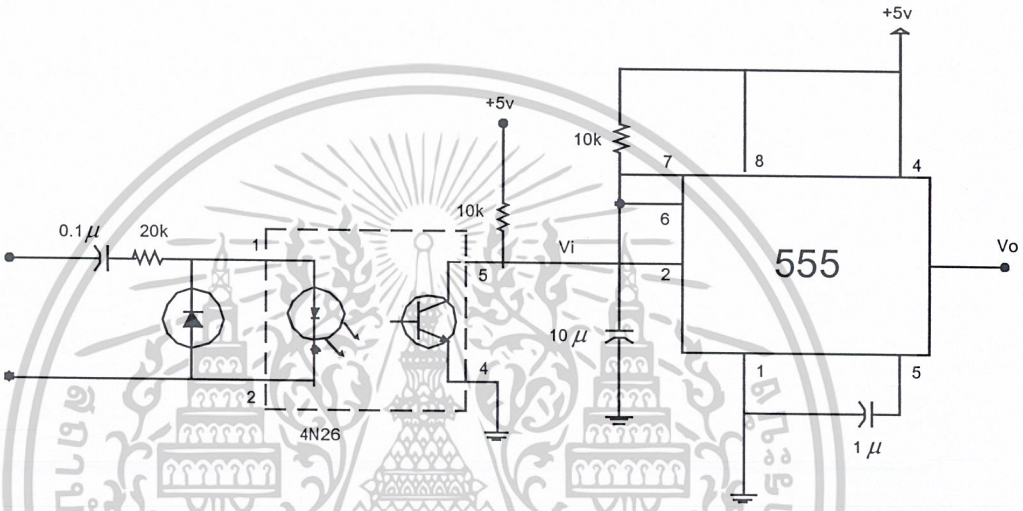
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองวงจรตรวจสอบสัญญาณกระดิ่ง (Check Ringing)

1. ทำการต่อวงจรตามรูปที่ 4.1
2. ป้อนสัญญาณกระดิ่งที่สายสัญญาณโทรศัพท์ ดังรูปที่ 4.2 และวัดรูปสัญญาณที่ Vi (เอาท์พุทของ 4N26) และ Vo (เอาท์พุทของ timer 555)

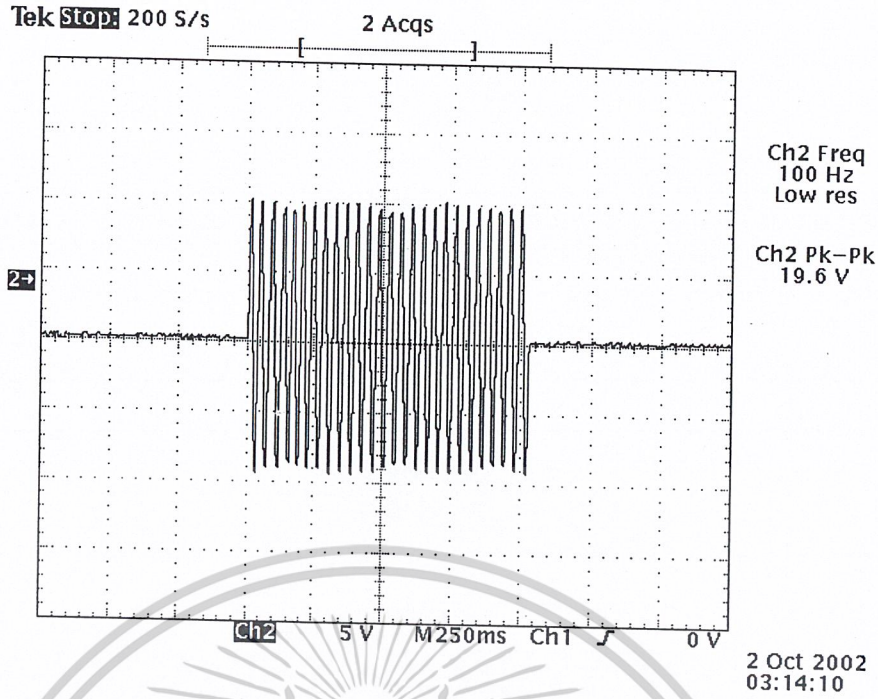


รูปที่ 4.1 แสดงวงจรตรวจจับสัญญาณกระดิ่ง

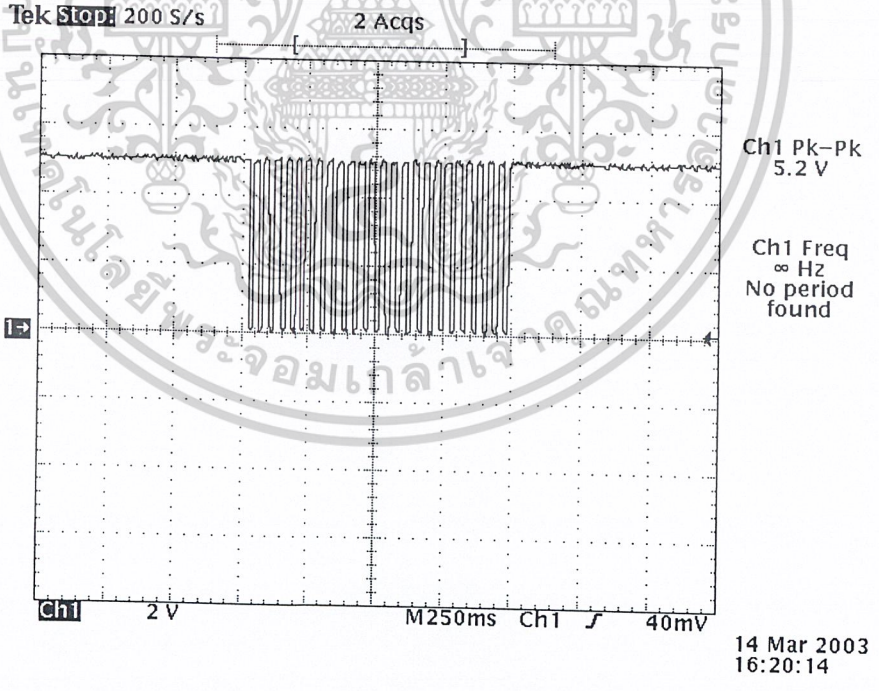
ผลการทดลอง

เมื่อมีสัญญาณกระดิ่งเข้ามาดังรูปที่ 4.2 เราจะทำการจับสัญญาณเอาท์พุทของ 4N26 และ timer 555 ที่ตำแหน่ง Vi และ Vo ดังรูปที่ 4.3 และ รูปที่ 4.4 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

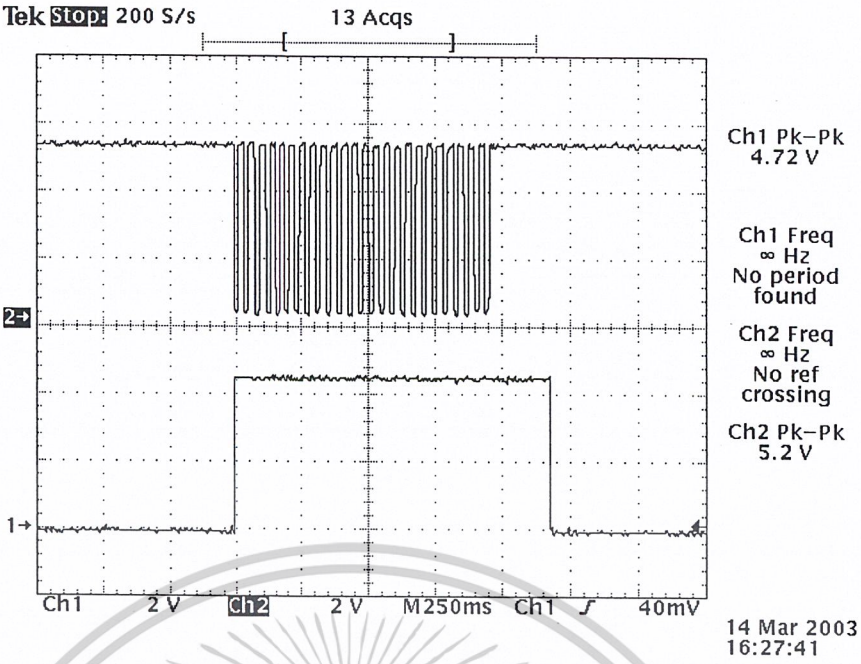


รูปที่ 4.2 แสดงสัญญาณกระดิ่งที่ป้อนเข้ามาที่สายสัญญาณโทรศัพท์



รูปที่ 4.3 แสดงสัญญาณเอาท์พุท (ขา 5) ของ 4N26 เมื่อมีอินพุทเป็นสัญญาณกระดิ่ง

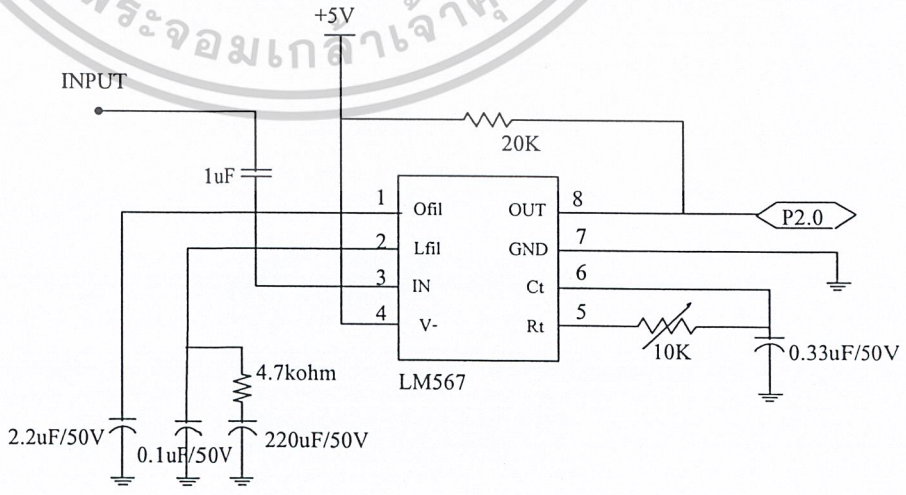
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงการเปรียบเทียบสัญญาณเอาท์พุท ของ timer 555 (channel 1) กับสัญญาณเอาท์พุท ของ 4N26 (channel 2) เมื่อมีอินพุทเป็นสัญญาณกระดิ่ง

4.2 การทดลองวงจรตรวจสอบสัญญาณความถี่เสียง (Check Tone)

1. ทำการต่อวงจรตรวจจับสัญญาณความถี่เสียง ดังรูปที่ 4.5
2. ป้อนสัญญาณ Busy Tone ที่สายสัญญาณโทรศัพท์และวัดรูปสัญญาณที่เอาท์พุท (ขา 8) ของ LM567
3. ป้อนสัญญาณ Dial Tone ที่สายสัญญาณโทรศัพท์และวัดรูปสัญญาณที่เอาท์พุท (ขา 8) ของ LM567

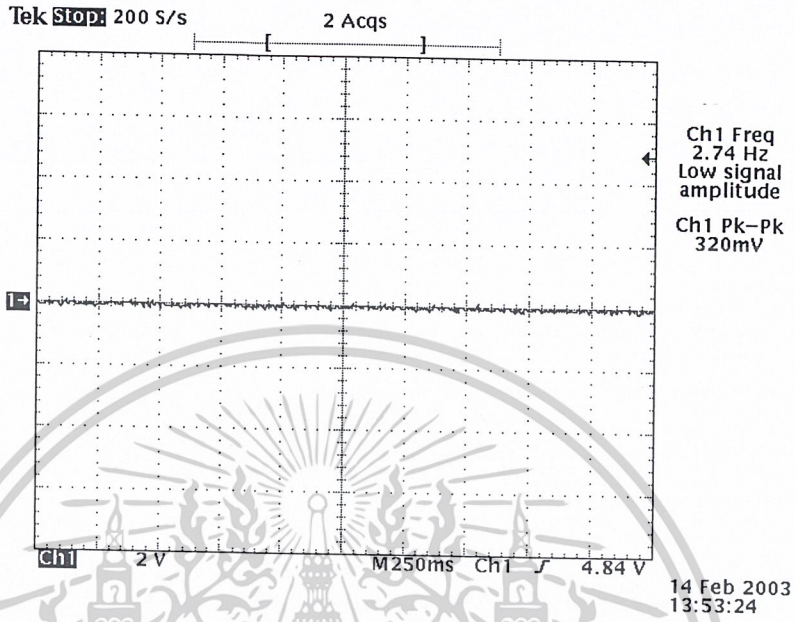


รูปที่ 4.5 แสดงวงจรตรวจจับสัญญาณความถี่เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

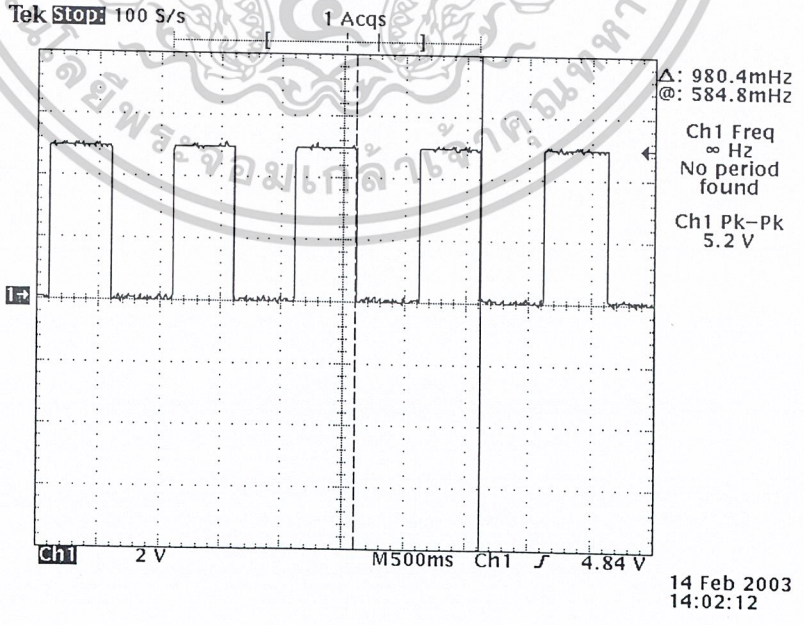
ผลการทดลอง

เมื่อมีสัญญาณ Dial Tone เข้ามาที่อินพุท เราจะทำการจับสัญญาณที่เอาท์พุท (ขา 8) ของ LM567 จะได้สัญญาณที่มีระดับ 0 โวลท์ ดังรูปที่ 4.6



รูปที่ 4.6 แสดงสัญญาณเอาท์พุท (ขา 8) ของ LM 567 เมื่อมีอินพุทเป็นสัญญาณ Dial Tone

เมื่อมีสัญญาณ Busy Tone เข้ามาที่อินพุท เราจะทำการจับสัญญาณที่เอาท์พุท (ขา8) ของ LM567 จะได้สัญญาณ ดังรูปที่ 4.7

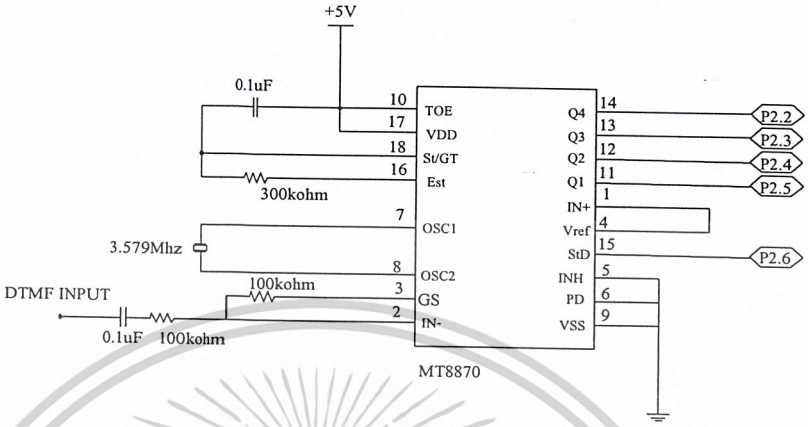


รูปที่ 4.7 แสดงสัญญาณเอาท์พุท (ขา 8) ของ LM 567 เมื่อมีอินพุทเป็นสัญญาณ Busy Tone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ (DTMF Decoder)

1. ทำการต่อวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ ดังรูปที่ 4.8
2. ป้อนสัญญาณดีทีเอ็มเอฟเข้าที่อินพุทของ MT8870 และทำการวัดที่ขา Q4, Q3, Q2 และ Q1



รูปที่ 4.8 แสดงวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ

ผลการทดลอง

เมื่อป้อนสัญญาณดีทีเอ็มเอฟเข้ามาที่อินพุท เราจะทำการจับสัญญาณที่เอาท์พุท (ขา 11, 12, 13 และ 14) ของ MT8870 จะได้ผลการทดลอง ดังตารางที่ 4.1

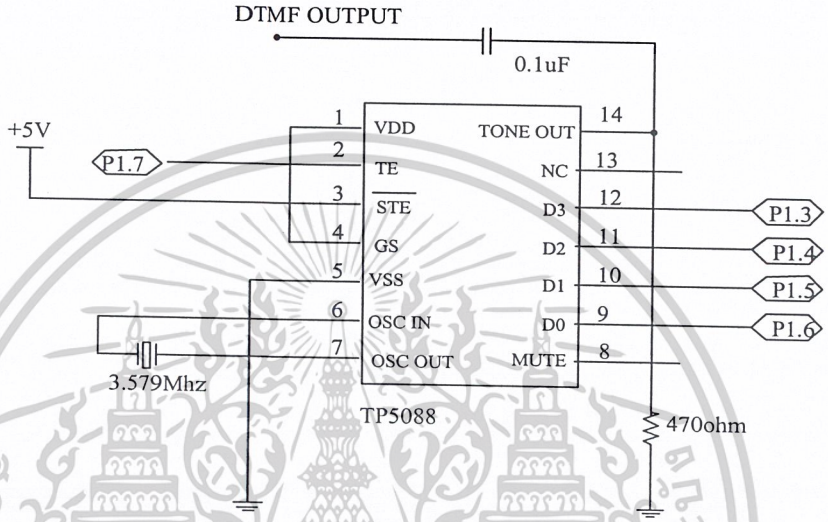
หมายเลข	BINARY CODE (Q4, Q3, Q2, Q1)
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
0	1010
*	1011
#	1100

ตารางที่ 4.1 แสดงผลการทดลองที่ได้จากวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองวงจรกำเนิดสัญญาณดีทีเอ็มเอฟ (Dial DTMF)

1. ทำการต่อวงจรกำเนิดสัญญาณดีทีเอ็มเอฟ ดังรูปที่ 4.9
2. ป้อนสัญญาณแรงดันเข้าที่ขา TE ของ TP5088 ด้วยแรงดัน +5 โวลท์ โดยกำหนดบิตให้สร้างสัญญาณหมายเลข 1 โดยให้ D3 = 0 โวลท์, D2 = 0 โวลท์ , D1 = 0 โวลท์ และ D0 = +5 โวลท์ และวัดสัญญาณที่เอาท์พุท (ขา14) ของ TP5088 แล้วทำการกำหนดบิตให้แต่ละหมายเลข และวัดสัญญาณที่เอาท์พุท (ขา14) ของ TP5088

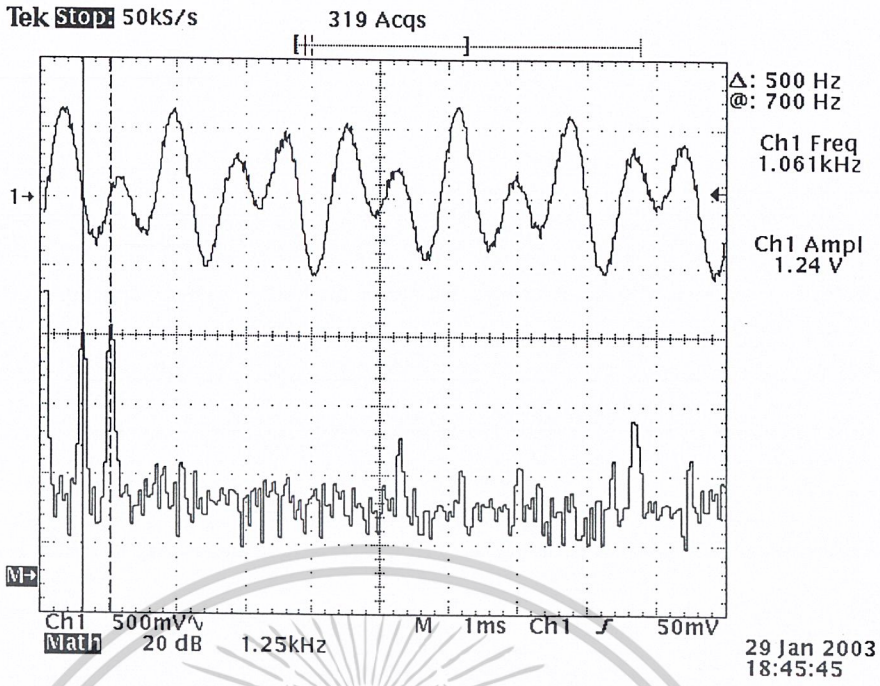


รูปที่ 4.9 แสดงวงจรกำเนิดสัญญาณดีทีเอ็มเอฟ

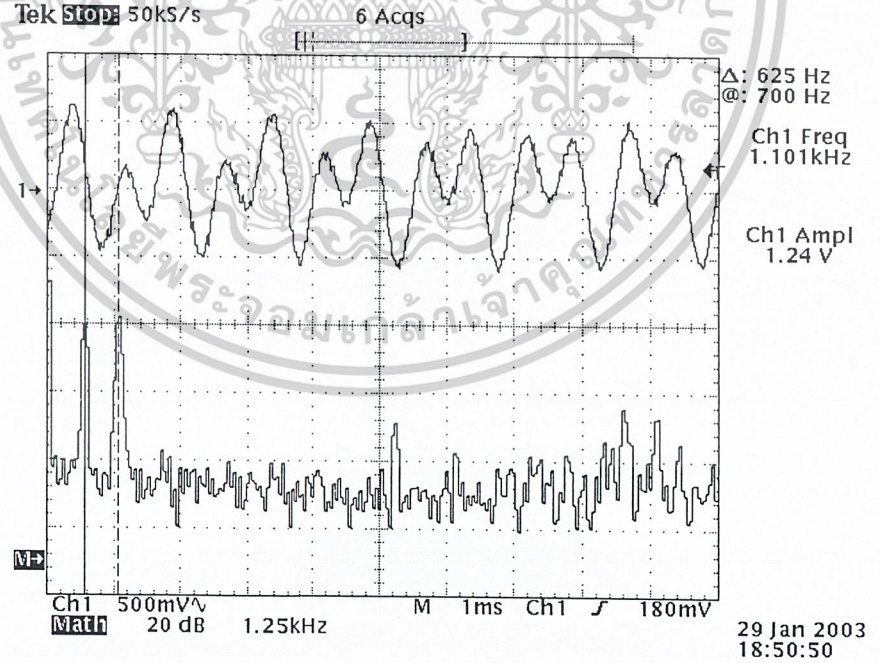
ผลการทดลอง

เมื่อป้อนสัญญาณแรงดันให้สร้างสัญญาณหมายเลขแต่ละหมายเลขเข้ามาที่อินพุท เราจะทำการจับสัญญาณที่เอาท์พุท (ขา 14) ของ MT8870 ซึ่งมีลักษณะสัญญาณของแต่ละหมายเลขที่ได้จะเป็นสัญญาณที่มี 2 ความถี่รวมกันอยู่ ดังผลการทดลองที่ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

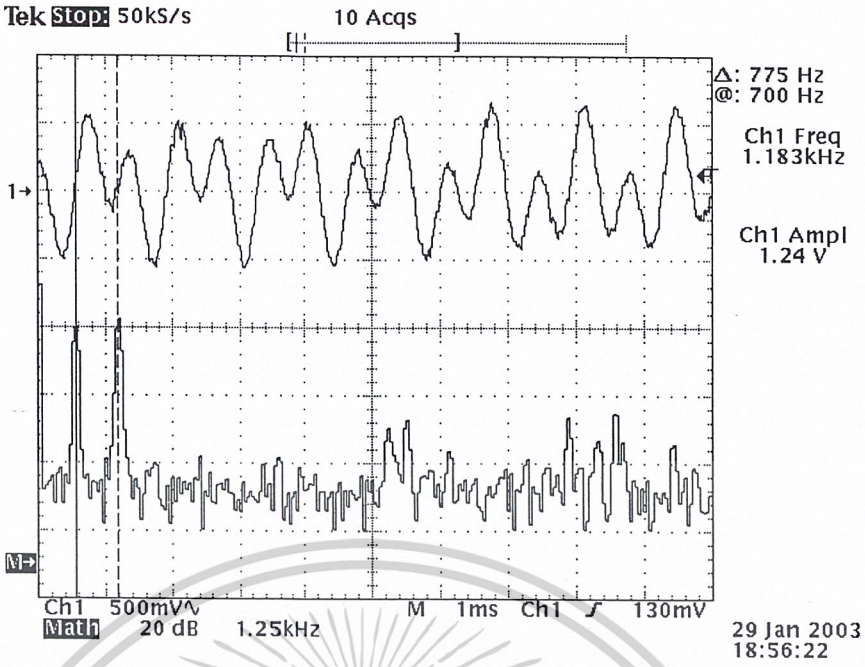


รูปที่ 4.10 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 1

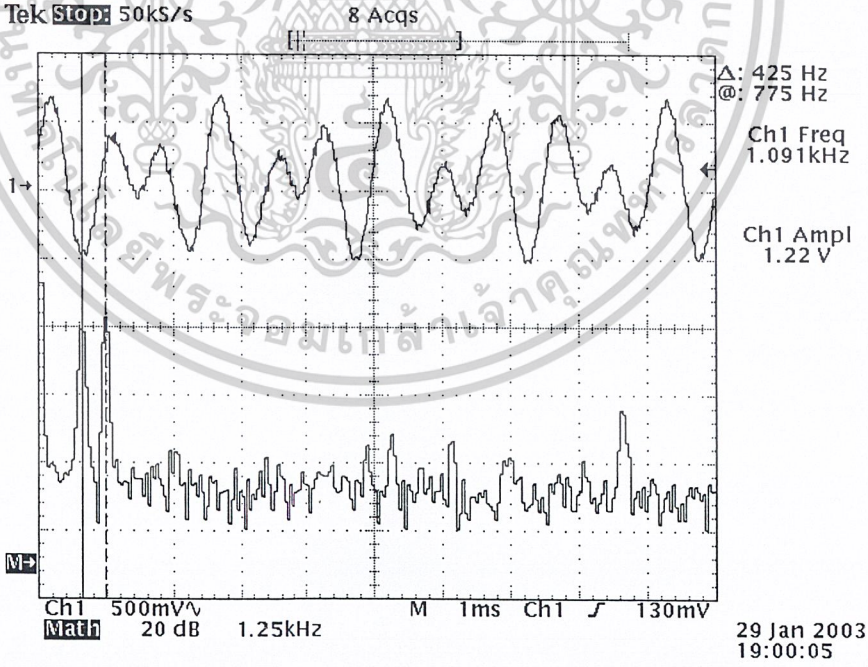


รูปที่ 4.11 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

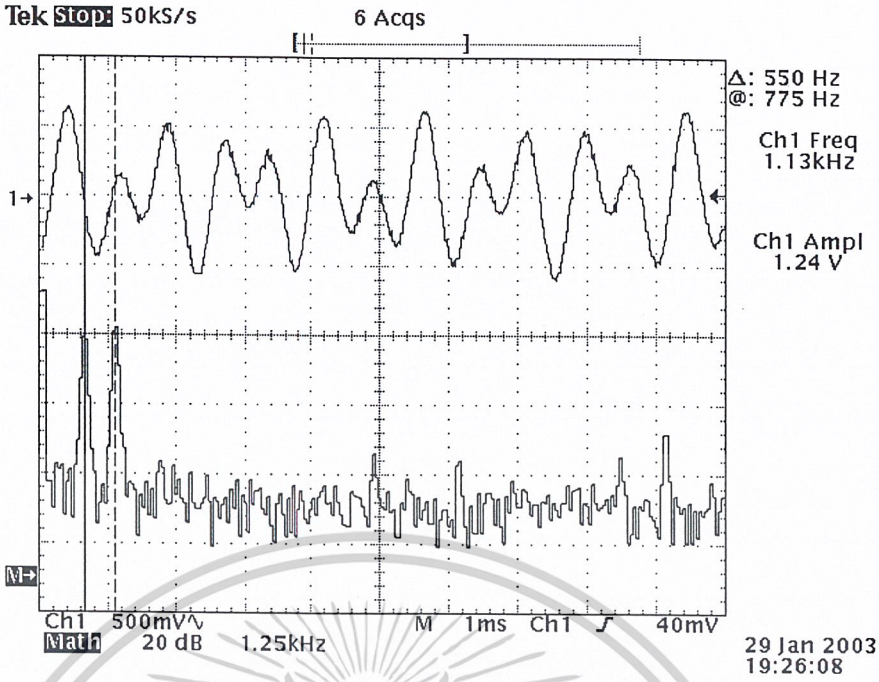


รูปที่ 4.12 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 3

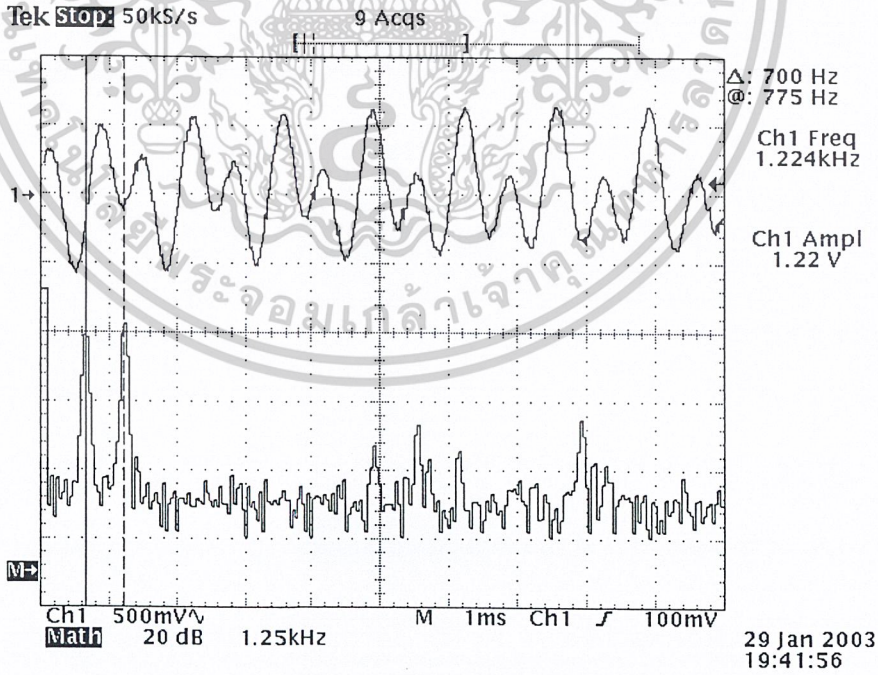


รูปที่ 4.13 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

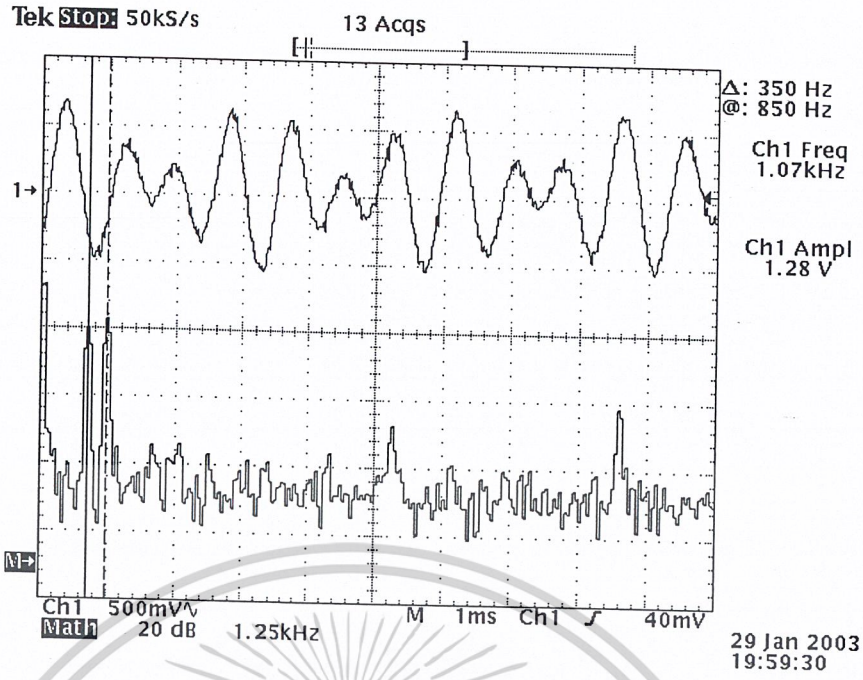


รูปที่ 4.14 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 5

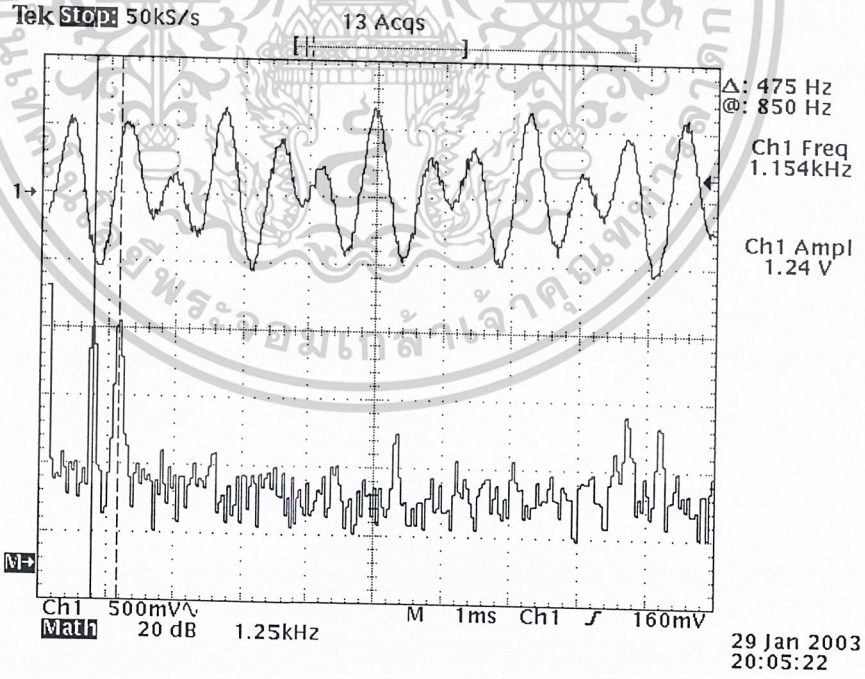


รูปที่ 4.15 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

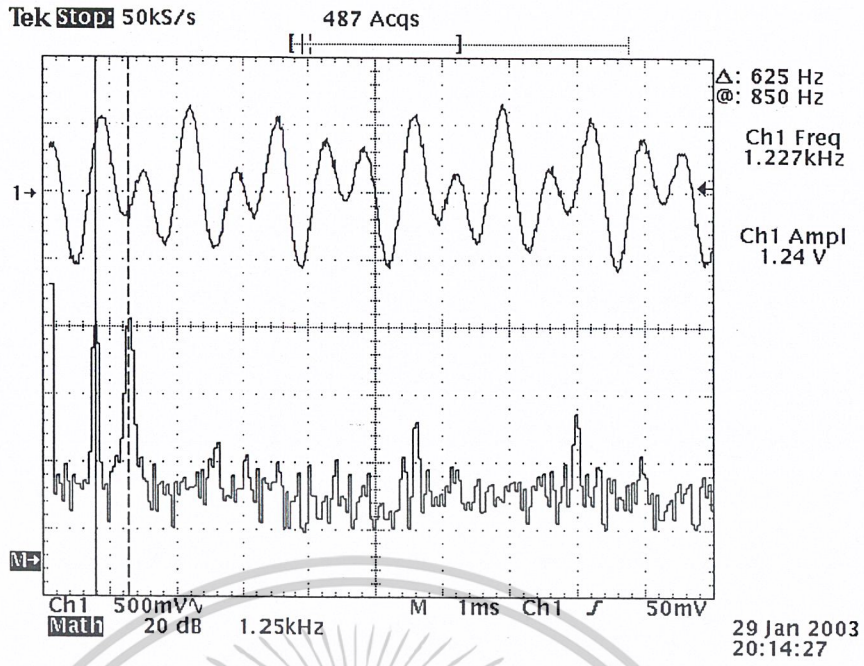


รูปที่ 4.16 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 7

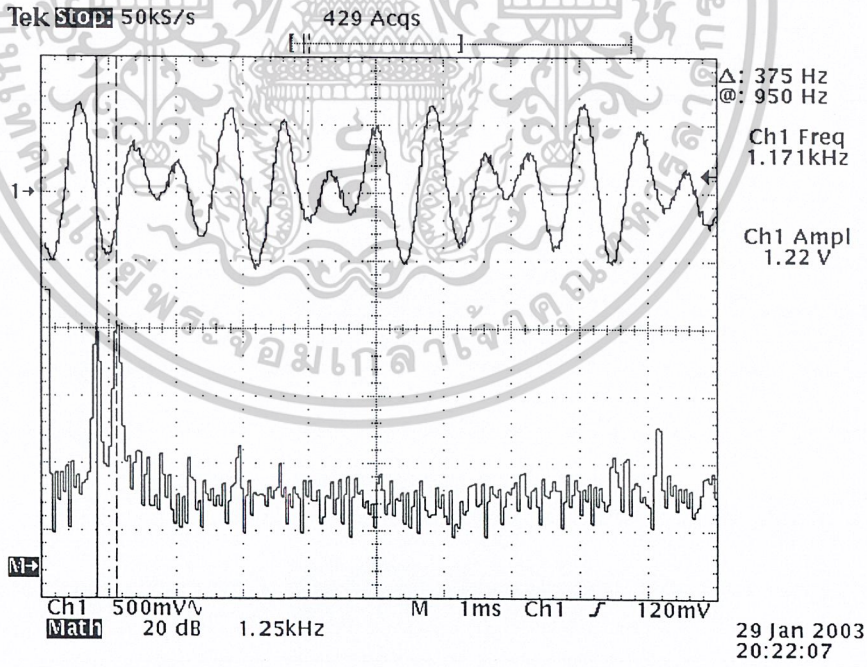


รูปที่ 4.17 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

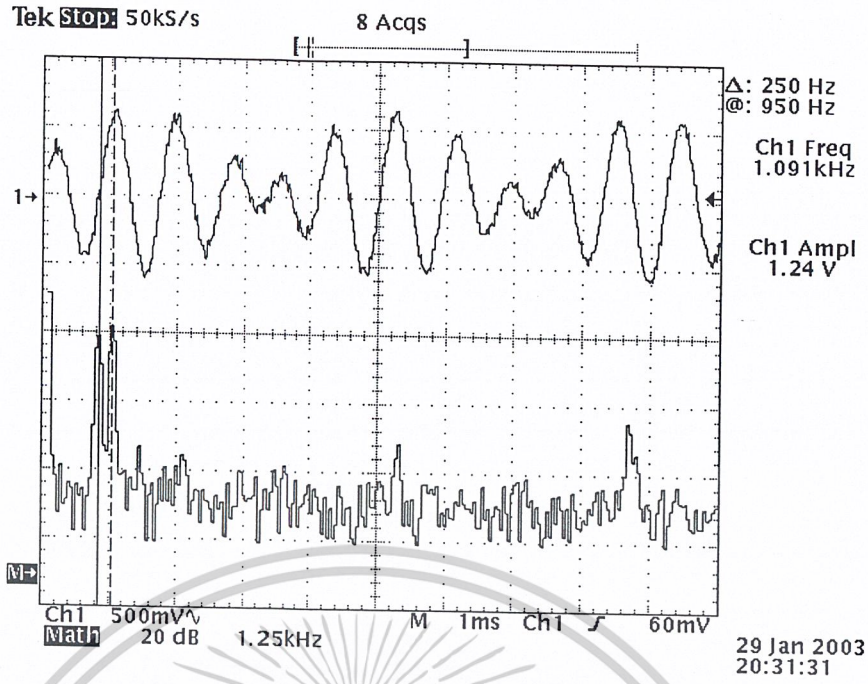


รูปที่ 4.18 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 9

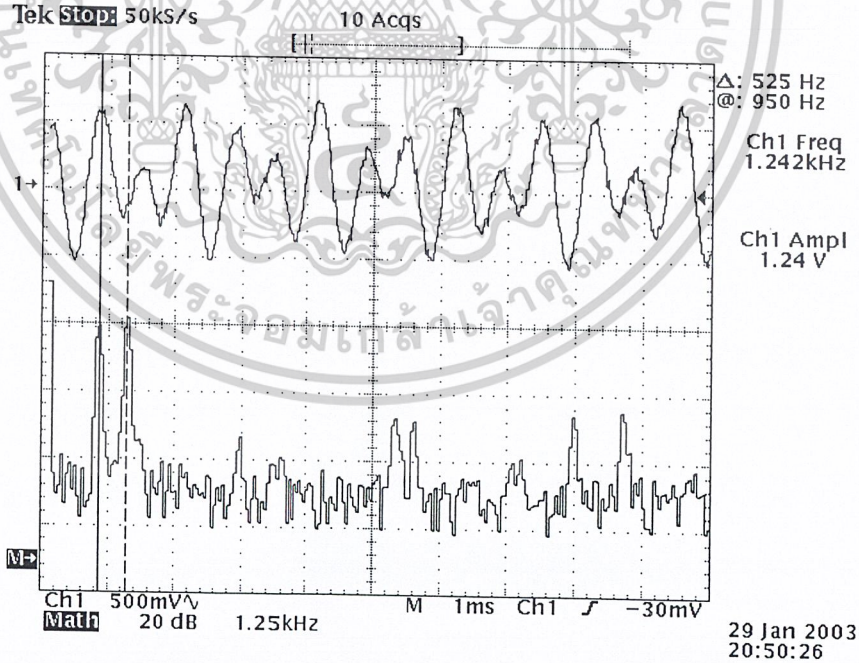


รูปที่ 4.19 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข *



รูปที่ 4.21 แสดงสัญญาณดีทีเอ็มเอฟที่เอาท์พุท (ขา 14) ของ TP5088 เมื่อเป็นสัญญาณหมายเลข #

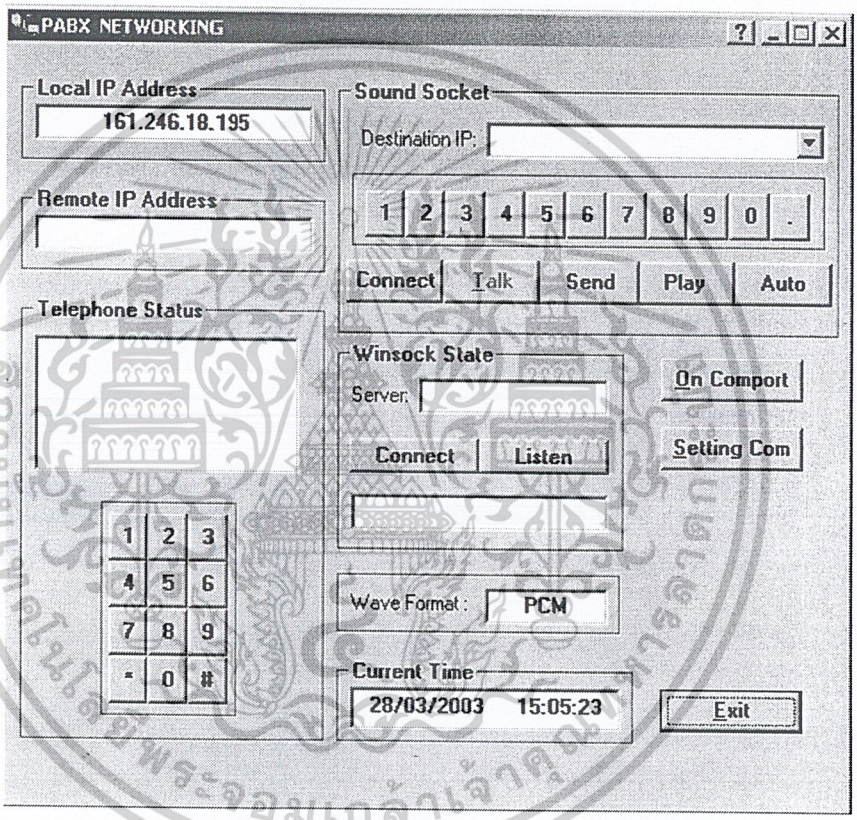
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ภาคแสดงผลในส่วนการทดลองโปรแกรม

ในโครงงานนี้จะแสดงผลด้วยโปรแกรม Visual Basic โดยจะรับค่ามาจาก Microcontroller (MCS-51) ซึ่งการแสดงผลจะเป็นดังนี้

4.5.1 หน้าโปรแกรมขณะเตรียมพร้อมใช้งาน

ในหน้าโปรแกรมนี้ เป็นหน้าโปรแกรมที่เตรียมพร้อมการใช้งาน ซึ่งทำหน้าที่เป็นเซิร์ฟเวอร์อยู่ตลอดเวลาทั้งต้นทางและปลายทางก่อนที่ได้รับการร้องขอการติดต่อจากฮาร์ดแวร์ โดยหลังจากที่ได้รับการร้องขอการติดต่อจากฮาร์ดแวร์แล้วหน้าโปรแกรมนี้ จะเปลี่ยนหน้าที่เป็นไคลเอ็นท์แทนเพื่อที่จะเรียกเซิร์ฟเวอร์ปลายทางอีกเครื่องหนึ่ง ดังรูปที่ 4.22



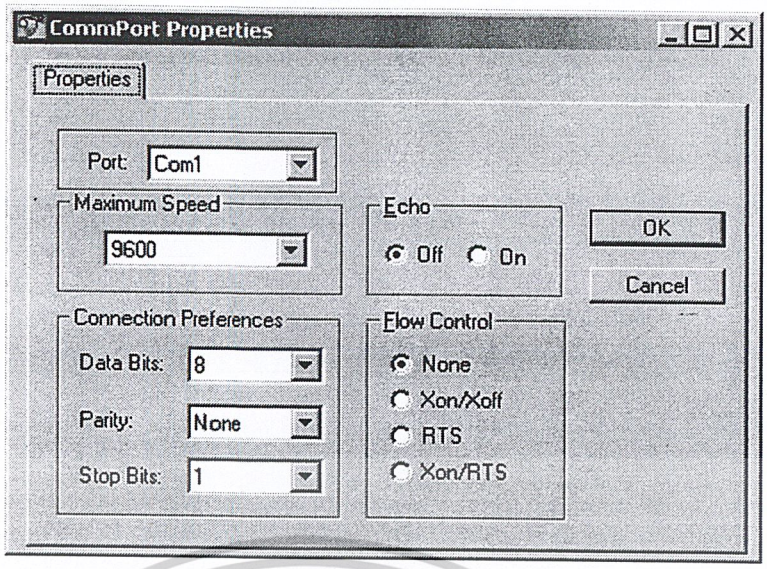
รูปที่ 4.22 แสดงรูปหน้าโปรแกรมขณะเตรียมพร้อมใช้งาน

4.5.2 หน้าโปรแกรมการตั้งค่าต่างๆ

ในหน้าโปรแกรมการตั้งค่าคุณสมบัติต่างๆ นี้ จะมีหน้าโปรแกรม ซึ่งเป็นส่วนสำคัญที่ต้องตั้งค่าคุณสมบัติต่างๆ ให้ถูกต้อง ดังต่อไปนี้

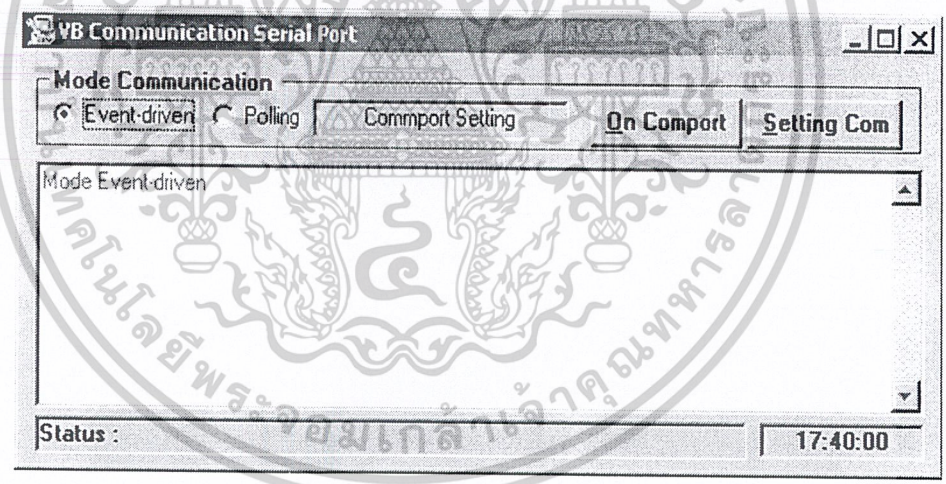
การตั้งค่าต่างๆ ในการรับค่าที่ได้รับมาจาก Microcontroller (MCS-51) นั้นให้ไปตั้งค่าที่ Setting Com ซึ่งค่าต่างๆ เป็นดังรูปที่ 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 แสดงรูปหน้าการตั้งค่าต่างๆ ให้กับโปรแกรม Visual Basic

ต่อมาก็จะตั้งค่าให้โปรแกรมพร้อมที่จะรับค่าที่ส่งมาจาก Microcontroller (MCS-51) โดยการตั้งไปที่ On Comport และ Communication Mode ให้ตั้งไปที่ Event-driven ดังรูปที่ 4.24

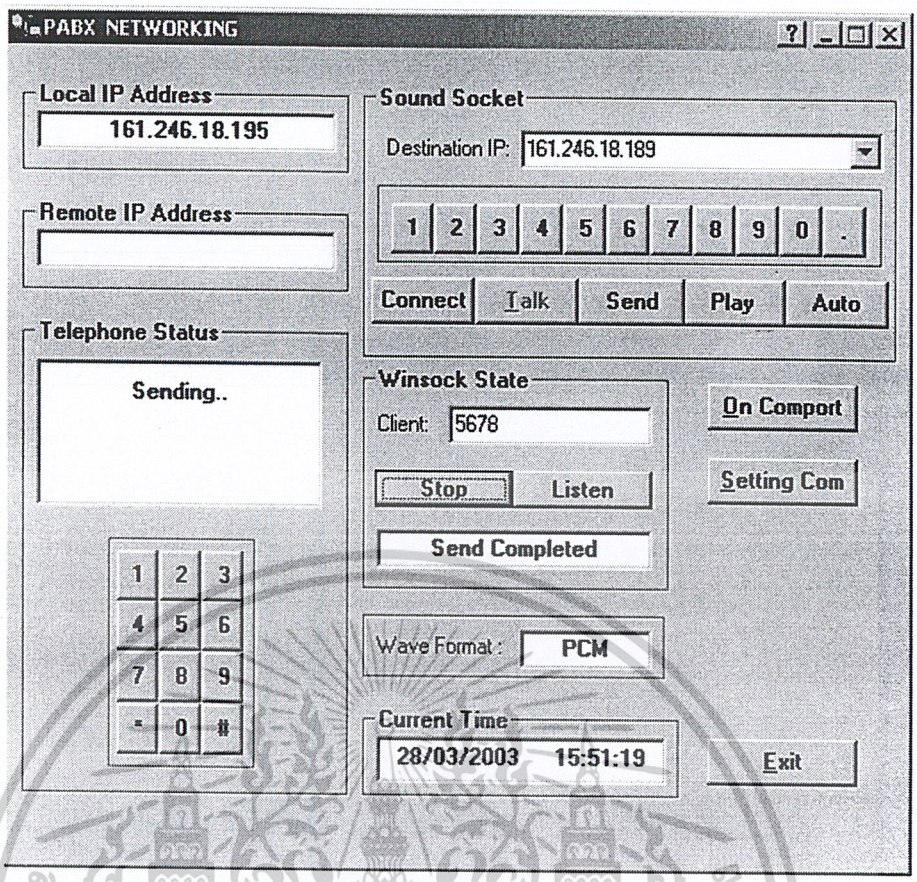


รูปที่ 4.24 แสดงหน้าต่างที่พร้อมจะรับข้อมูลจาก Microcontroller

4.5.3 หน้าโปรแกรมขณะที่ทำหน้าที่เป็นไคลเอ็นท์

ในหน้าโปรแกรมนี้ เป็นหน้าโปรแกรมที่ทำหน้าที่เป็นไคลเอ็นท์ ซึ่งเมื่อเริ่มให้โปรแกรมทำหน้าที่เป็นไคลเอ็นท์ทำงานต้องป้อนหมายเลขโทรศัพท์ดังรูปที่ 4.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

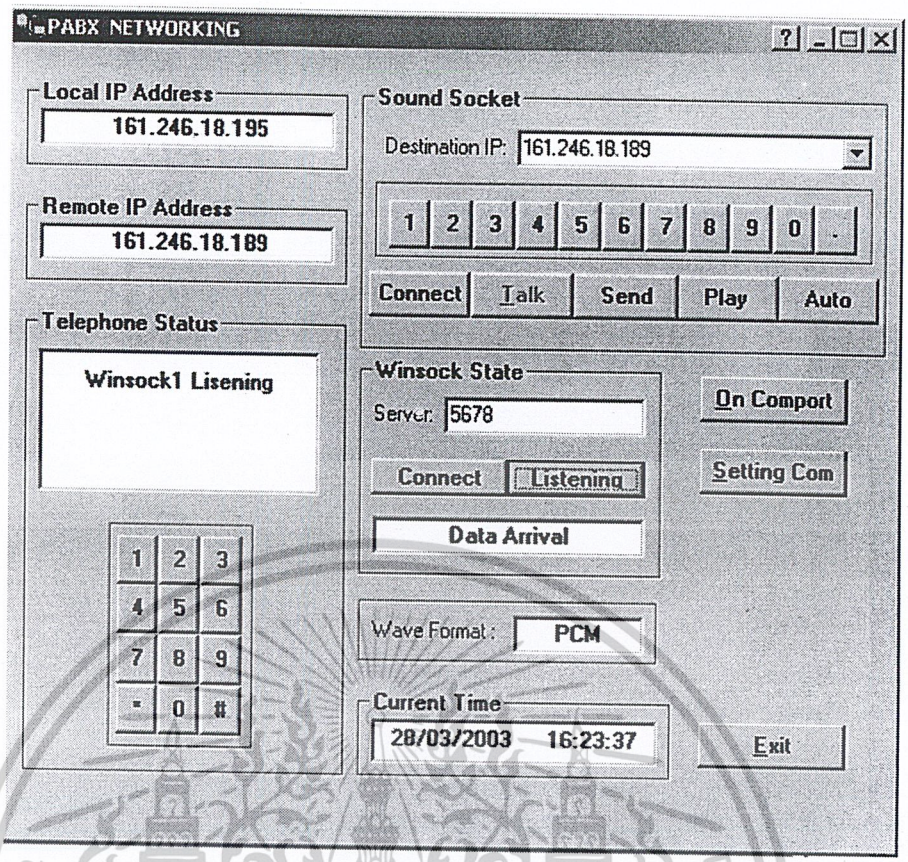


รูปที่ 4.25 แสดงรูปหน้าโปรแกรมขณะที่ทำหน้าที่เป็นไคลเอ็นท์

4.5.4 หน้าโปรแกรมขณะที่ทำหน้าที่เป็นเซิร์ฟเวอร์

ในหน้าโปรแกรมนี้ เป็นหน้าโปรแกรมที่ทำหน้าที่เป็นเซิร์ฟเวอร์ ซึ่งเมื่อเริ่มให้โปรแกรมทำหน้าที่เป็นเซิร์ฟเวอร์ทำงานต้องถูกเรียกจากไคลเอ็นท์ แล้วก็รับหมายเลขโทรศัพท์มาส่งไปยังฮาร์ดแวร์ ดังรูปที่ 4.26

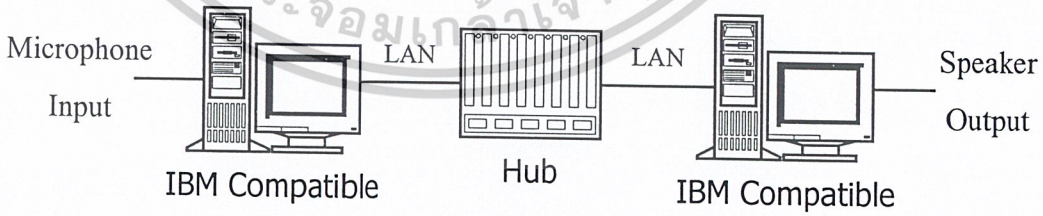
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 แสดงรูปหน้าโปรแกรมขณะที่ทำหน้าที่เป็นเซิร์ฟเวอร์

4.6 การทดสอบวัดคุณภาพของสัญญาณเสียงขณะที่สนทนา

หลังจากที่ทำการต่อพอร์ตเสียงแล้วด้วยโปรแกรม จากนั้นก็ทำการป้อนสัญญาณไซน์ความถี่ 1 kHz, 2kHz และ 3kHz เข้าที่ไมโครโฟนของ Sound Card แล้วทำการวัดสัญญาณที่ Speaker ของ Sound Card ดังรูปที่ 4.27

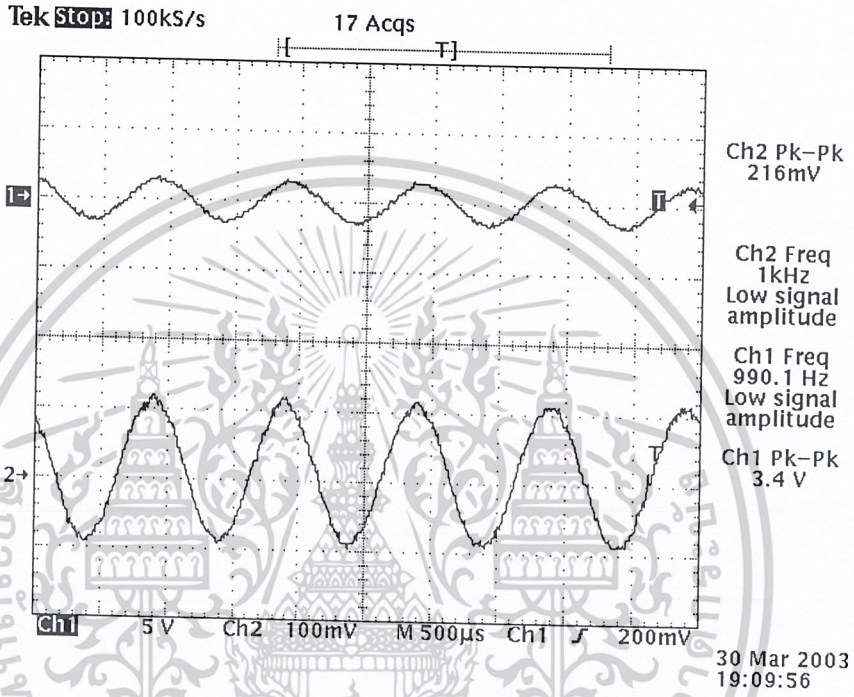


รูปที่ 4.27 แสดงการเชื่อมต่อเครือข่ายเพื่อวัดสัญญาณเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

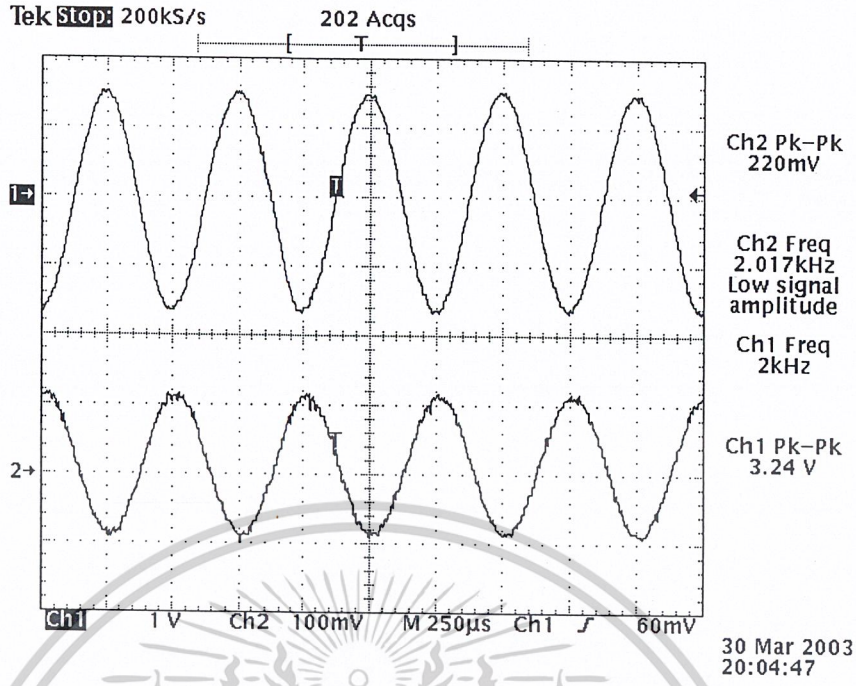
ผลการทดลอง

เมื่อป้อนสัญญาณไซน์ โดยความถี่ที่ป้อนนั้นจะอยู่ในช่วงความถี่ ที่ใช้พูดในระบบโทรศัพท์ ซึ่งความถี่ที่ทำการทดลองป้อนคือ 1kHz, 2kHz, และ 3kHz เข้าที่ไมโครโฟนของ Sound Card แล้วทำการวัดสัญญาณที่ Speaker ของ Sound Card จะได้ ผลการทดลองคุณภาพของสัญญาณเสียงที่วัดได้จะเห็นว่าจะมีสัญญาณรบกวนผสมเข้ามาดังรูปที่ 4.28, 4.29 และ 4.30 ตามลำดับ

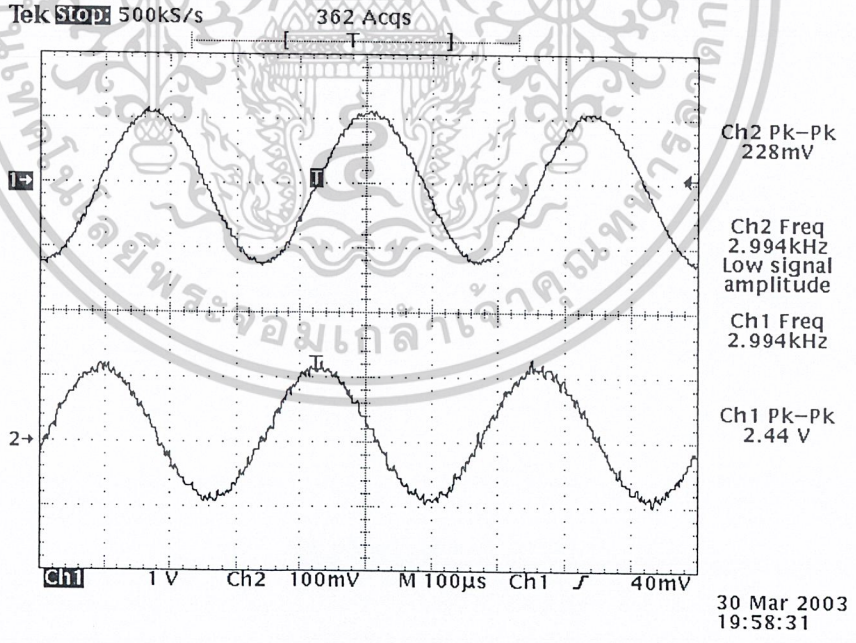


รูปที่ 4.28 แสดงสัญญาณอินพุตที่ป้อนให้กับระบบ (channel1) เทียบกับสัญญาณเอาท์พุตของระบบ (channel2) โดยป้อนสัญญาณซายน์ที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 แสดงสัญญาณอินพุตที่ป้อนให้กับระบบ (channel1)
เทียบกับสัญญาณเอาท์พุตของระบบ (channel2) โดยป้อนสัญญาณซายน์ที่ความถี่ 2 kHz



รูปที่ 4.30 แสดงสัญญาณอินพุตที่ป้อนให้กับระบบ (channel1)
เทียบกับสัญญาณเอาท์พุตของระบบ (channel2) โดยป้อนสัญญาณซายน์ที่ความถี่ 3 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

5.1 ส่วนสรุปผลของโครงงาน

ในส่วนของโครงงานนี้ได้แบ่งการทำงานออกเป็น 2 ส่วน คือ

1. ส่วนของฮาร์ดแวร์ ได้ทำการทดลองต่อวงจรต่างๆ ได้แก่ ภาคตรวจจับสัญญาณกระดิ่ง (Check Ringing), ภาคตรวจจับสัญญาณความถี่เสียง (Check Tone), ภาคถอดรหัสสัญญาณดีทีเอ็มเอฟ (DTMF Decoder) และภาคกำเนิดสัญญาณดีทีเอ็มเอฟ (Dial DTMF) ซึ่งมีปัญหาที่เกิดขึ้นดังนี้

- การตรวจสอบของวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟบางครั้งก็ตรวจจับไม่ได้ ดังนั้นจึงแก้ไขโดยเปลี่ยนค่าความต้านทาน อินพุทของวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ ให้มีค่าน้อยลงเพื่อปรับอัตราขยายของวงจร ผลที่ได้คือการตรวจจับของวงจร ถอดรหัสสัญญาณดีทีเอ็มเอฟได้ดีขึ้น
- วงจรตรวจจับสัญญาณกระดิ่ง ตอนแรกไม่ได้สัญญาณพัลส์ตามต้องการ ดังนั้นจึงแก้ไขโดยเพิ่มค่าตัวเก็บประจุให้เป็นได้ค่าที่เหมาะสม ทำให้ได้สัญญาณพัลส์ตามต้องการ
- ในส่วนของวงจรอื่นๆ สามารถทำงานได้ปกติ

2. ส่วนซอฟท์แวร์ ได้แบ่งออกเป็น 2 ส่วนคือ ส่วนของไมโครคอนโทรลเลอร์ที่ใช้ควบคุมการทำงานของวงจรของระบบ และส่วนของโปรแกรม Visual Basic 6 ซึ่งควบคุมในการทำหน้าที่เป็นไคลเอ็นท์ หรือ เซิร์ฟเวอร์ ซึ่งมีปัญหาที่เกิดขึ้นดังนี้

- โปรแกรมที่ตัวฮาร์ดแวร์สามารถทำงานได้ดีพอสมควร แต่ก็มีบางครั้งที่ทำงานผิดพลาดบ้าง
- โปรแกรมในส่วนของการควบคุมในการที่ทำหน้าที่เป็นไคลเอ็นท์หรือเซิร์ฟเวอร์ บางครั้งก็เกิดปัญหา โดยไม่สามารถทำงานตามผังการทำงานของโปรแกรม สาเหตุนั้นเพราะว่าภาษาโปรแกรมที่ใช้ในการเขียนโปรแกรมคือ Visual Basic Version 6.0 ซึ่งหลักการของภาษาเป็นแบบ Object Oriented Programming โดยจะเกิดตามเหตุการณ์ที่ได้เขียนโค๊ดไว้จาก Object ใด Object หนึ่ง ซึ่งถ้าเกิดเหตุการณ์ขึ้นในลักษณะคล้ายกันโปรแกรมก็จะทำงานผิดพลาดได้

การทดสอบการทำงานของโครงงานเมื่อสามารถพูดคุยกันได้แล้ว โดยทดสอบสนทนาโดยไม่นำเสียงผ่านทางฮาร์ดแวร์ก็คือ สนทนาโดยใช้ไมโครโฟนกับลำโพงที่ตัวคอมพิวเตอร์ สามารถที่จะสนทนาได้ชัดเจนดีแต่มีการหน่วงของสัญญาณเสียงไป แต่เมื่อนำเสียงผ่านทางฮาร์ดแวร์และก็หมุนหมายเลขโทรศัพท์ สนทนากับผู้รับสายที่ถูกเรียก เสียงที่สนทนาไม่ค่อยชัดเจนนัก ซึ่งเกิดจากวงจรขยายเสียงสองทิศทางขยายเสียงไม่ค่อยดีนัก

5.2 ส่วนแก้ไขและพัฒนา

โครงงานนี้จะมีปัญหาที่เกิดขึ้นหลายอย่างด้วยกัน คือ การทำงานของโปรแกรมควบคุมในการที่ทำหน้าที่เป็นไคลเอ็นหรือเซิร์ฟเวอร์, ความชัดเจนของเสียงที่สนทนาไม่ค่อยชัดเจนนัก, และการหน่วงของสัญญาณเสียง ดังนั้นปัญหาที่เกิดขึ้นจะเป็นแนวทางในการพัฒนาโครงงานให้ดีขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

October 1991

TP5088 DTMF Generator for Binary Data

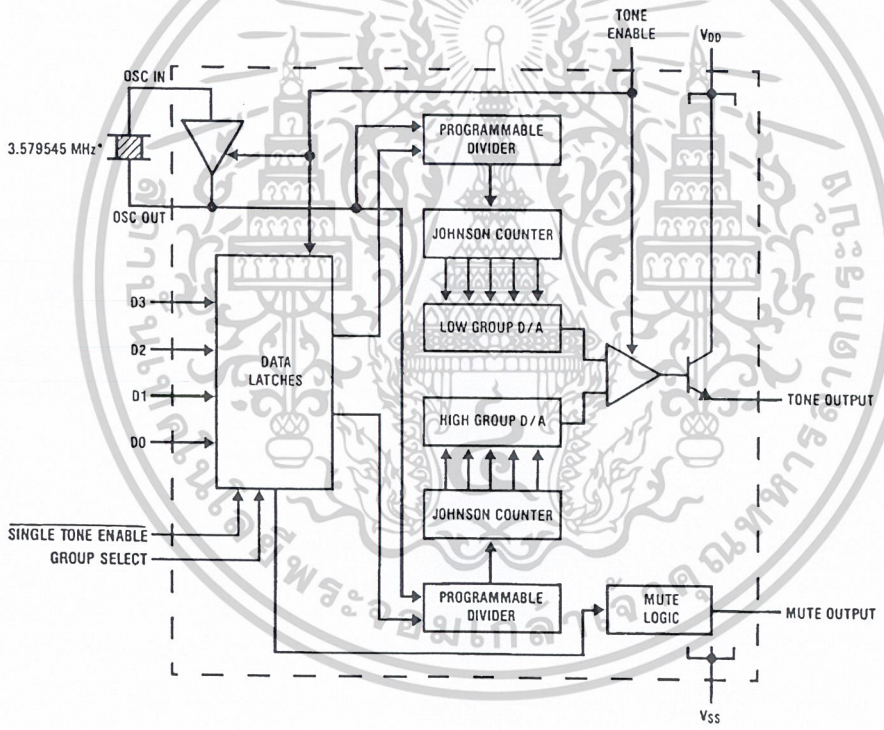
General Description

This CMOS device provides low cost tone-dialing capability in microprocessor-controlled telephone applications. 4-bit binary data is decoded directly, without the need for conversion to simulated keyboard inputs required by standard DTMF generators. With the TONE ENABLE input low, the oscillator is inhibited and the device is in a low power idle mode. On the low-to-high transition of TONE ENABLE, data is latched into the device and the selected tone pair from the standard DTMF frequencies is generated. An open-drain N-channel transistor provides a MUTE output during tone generation.

Features

- Direct microprocessor interface
- Binary data inputs with latches
- Generates 16 standard tone pairs
- On-chip 3.579545 MHz crystal-controlled oscillator
- Better than 0.64% frequency accuracy
- High group pre-emphasis
- Low harmonic distortion
- MUTE output interfaces to speech network
- Low power idle mode
- 3.5V–8V operation

Block Diagram



TL/H/5004–1

*Crystal Specification: Parallel Resonant 3.579545 MHz, $R_S \leq 150\Omega$, L = 100 mH, C_0 = 5 pF, C_1 = 0.02 pF.

© 1995 National Semiconductor Corporation TL/H/5004

RRD-B30M115/Printed in U. S. A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage ($V_{DD} - V_{SS}$)	12V
$\overline{MUTE}$ Voltage	12V
Maximum Voltage at Any Other Pin	V_{DD} + 0.3V to V_{SS} − 0.3V
Operating Temperature, T_A	−30°C to +70°C
Storage Temperature	−55°C to +150°C
Maximum Power Dissipation	500 mW

Electrical Characteristics

Unless otherwise noted, limits printed in **BOLD** characters are guaranteed for V_{DD} = 3.5V to 8V, T_A = 0°C to +70°C by correlation with 100% electrical testing at T_A = 25°C. All other limits are assured by correlation with other production tests and/or product design and characterization.

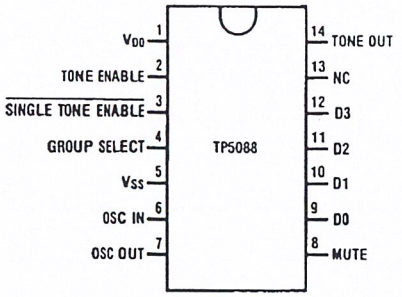
Parameter	Conditions	Min	Typ	Max	Units
Minimum Supply Voltage, V_{DD} (min)	Generating Tones	**3.5**			V
Minimum Supply Voltage for Data Input, TONE ENABLE and MUTE Logic Functions		2			V
Operating Current Idle Generating Tones	R_L = ∞, D0–D3 Open V_{DD} = 3.5V, Mute Open		55 1.5	**350** **2.5**	μA mA
Input Pull-Up Resistance D0–D3 TONE ENABLE			100 50		kΩ kΩ
Input Low Level TONE ENABLE, D0–D3				**0.2 V_{DD}**	V
Input High Level TONE ENABLE, D0–D3		**0.8 V_{DD}**			V
MUTE OUT Sink Current (TONE ENABLE LOW)	V_{DD} = 3.5V V_o = 0.5V	**0.4**			mA
MUTE OUT Leakage Current (TONE ENABLE HIGH)	V_{DD} = 3.5V V_o = V_{DD}		1		μA
Output Amplitudes Low Group High Group	R_L = 240 Ω V_{DD} = 3.5V T_A = 25°C	**130** **180**	170 230	**220** **310**	mVrms mVrms
Mean Output DC Offset	V_{DD} = 3.5V V_{DD} = 8V		1.2 3.6		V V
High Group Pre-Emphasis		**2.2**	2.7	**3.2**	dB
Dual Tone/Total Harmonic Distortion Ratio	1 MHz Bandwidth, V_{DD} = 5V R_L = 240Ω	**−20**			dB
Start-Up Time (to 90% Amplitude), t_{OSC}			4		ms
Data Set-Up Time, t_S (*Figure 2*)	V_{DD} = 5V	100			ns
Data Hold Time, t_H	V_{DD} = 5V	280			ns
Data Duration t_W	V_{DD} = 5V	600			ns

Note 1: R_L is the external load resistor connected from TONE OUT to V_{SS}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagram

Dual-In-Line Package



TL/H/5004–2

Top View

Order Number TP5088WM or TP5088N
See NS Package M14B or N14A

Functional Description

With the TONE ENABLE pin pulled low, the device is in a low power idle mode, with the oscillator inhibited and the output transistor turned off. Data on inputs D0–D3 is ignored until a rising transition on TONE ENABLE. Data meeting the timing specifications is latched in, the oscillator and output stage are enabled, and tone generation begins. The decoded data sets the high group and low group programmable counters to the appropriate divide ratios. These counters sequence two ratioed-capacitor D/A converters through a series of 28 equal duration steps per sine wave cycle. On-chip regulators ensure good stability of tone amplitudes with variations in supply voltage and temperature. The two tones are summed by a mixer amplifier, with pre-emphasis applied to the high group tone. The output is an NPN emitter-follower requiring the addition of an external load resistor to V_{SS}.

Table I shows the accuracies of the tone output frequencies and Table II is the Functional Truth Table.

TABLE I. Output Frequency Accuracy

Tone Group	Standard DTMF (Hz)	Tone Output Frequency	% Deviation from Standard
Low Group f_L	697	694.8	−0.32
	770	770.1	+0.02
	852	852.4	+0.03
	941	940.0	−0.11
High Group f_H	1209	1206.0	−0.24
	1336	1331.7	−0.32
	1477	1486.5	+0.64
	1633	1639.0	+0.37

Pin Descriptions

V_{DD} (Pin 1): This is the positive supply to the device, referenced to V_{SS}. The collector of the TONE OUT transistor is also connected to this pin.

V_{SS} (Pin 5): This is the negative voltage supply. All voltages are referenced to this pin.

OSC IN, OSC OUT (Pins 6 and 7): All tone generation timing is derived from the on-chip oscillator circuit. A low-cost 3.579545 MHz A-cut crystal (NTSC TV color-burst) is needed between pins 6 and 7. Load capacitors and a feedback resistor are included on-chip for good start-up and stability. The oscillator is stopped when the TONE ENABLE input is pulled to logic low.

TONE ENABLE Input (Pin 2): This input has an internal pull-up resistor. When TONE ENABLE is pulled to logic low, the oscillator is inhibited and the tone generators and output transistor are turned off. A low to high transition on TONE ENABLE latches in data from D0–D3. The oscillator starts, and tone generation continues until TONE ENABLE is pulled low again.

MUTE (Pin 8): This output is an open-drain N-channel device that sinks current to V_{SS} when TONE ENABLE is low and no tones are being generated. The device turns off when TONE ENABLE is high.

D0, D1, D2, D3 (Pins 9, 10, 11, 12): These are the inputs for binary-coded data, which is latched in on the rising edge of TONE ENABLE. Data must meet the timing specifications of *Figure 2*. At all other times these inputs are ignored and may be multiplexed with other system functions.

TONE OUT (Pin 14): This output is the open emitter of an NPN transistor, the collector of which is connected internally to V_{DD}. When an external load resistor is connected from TONE OUT to V_{SS}, the output voltage on this pin is the sum of the high and low group tones superimposed on a DC offset. When not generating tones, this output transistor is turned off to minimize the device idle current.

$\overline{\text{SINGLE TONE ENABLE}}$ (Pin 3): This input has an internal pull-up resistor. When pulled to V_{SS}, the device is in single tone mode and only a single tone will be generated at pin 14 (for testing purposes). For normal operation, leave this pin open-circuit or pull to V_{DD}.

GROUP SELECT (Pin 4): This pin is used to select the high group or low group frequency when the device is in single tone mode. It has an internal pull-up resistor. Leaving this pin open-circuit or pulling it to V_{DD} will generate the high group, while pulling to V_{SS} will generate the low group frequency at the TONE OUT pin.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE II. Functional Truth Table

Keyboard Equivalent	Data Inputs				TONE ENABLE	TONES OUT		MUTE
	D3	D2	D1	D0		f_L (Hz)	f_H(Hz)	
X	X	X	X	X	0	0V	0V	0V
1	0	0	0	1	⎍	697	1209	O/C
2	0	0	1	0	⎍	697	1336	O/C
3	0	0	1	1	⎍	697	1477	O/C
4	0	1	0	0	⎍	770	1209	O/C
5	0	1	0	1	⎍	770	1336	O/C
6	0	1	1	0	⎍	770	1477	O/C
7	0	1	1	1	⎍	852	1209	O/C
8	1	0	0	0	⎍	852	1336	O/C
9	1	0	0	1	⎍	852	1477	O/C
0	1	0	1	0	⎍	941	1336	O/C
*	1	0	1	1	⎍	941	1209	O/C
#	1	1	0	0	⎍	941	1477	O/C
A	1	1	0	1	⎍	697	1633	O/C
B	1	1	1	0	⎍	770	1633	O/C
C	1	1	1	1	⎍	852	1633	O/C
D	0	0	0	0	⎍	941	1633	O/C

Timing Diagram

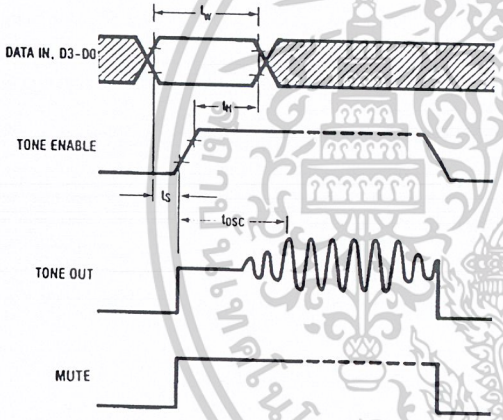


FIGURE 2

Typical Application

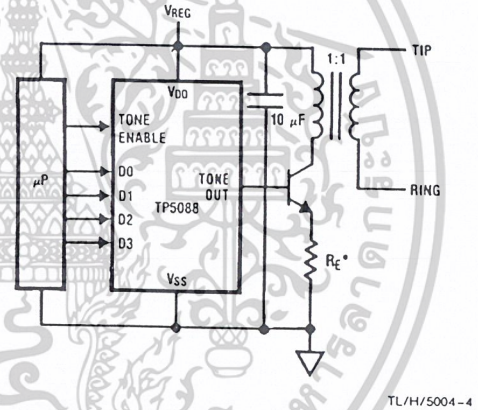
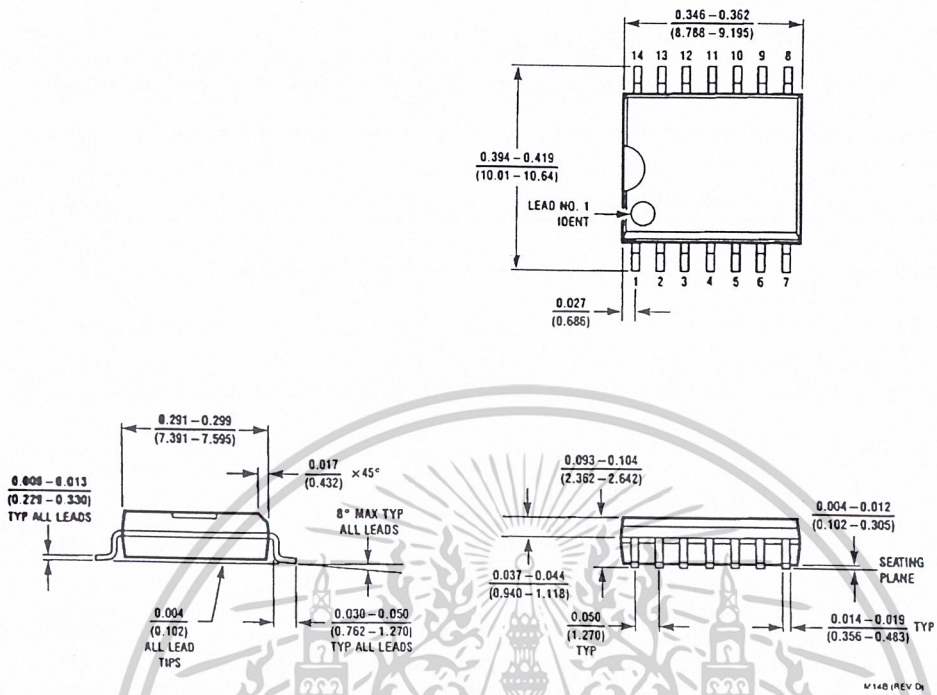


FIGURE 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

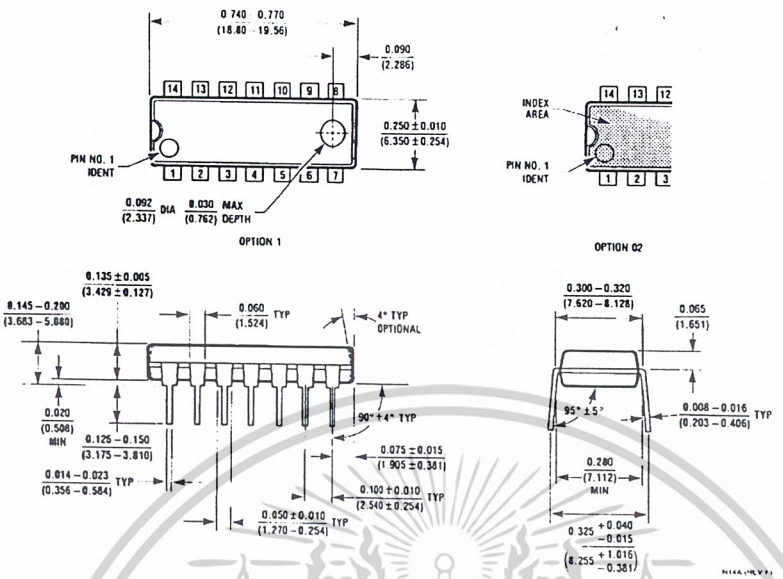
Physical Dimensions inches (millimeters)



Order Number TP5088WM
NS Package Number M14B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line (N)
Order Number TP5088N
NS Package Number N14A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

February 1995

LM567/LM567C Tone Decoder

General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

Features

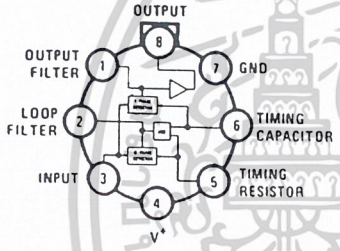
- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%
- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

Connection Diagrams

Metal Can Package

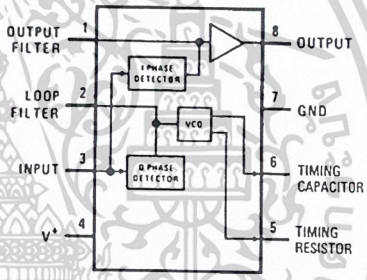


TL/H/6975–1

Top View

Order Number LM567H or LM567CH
See NS Package Number H08C

Dual-In-Line and Small Outline Packages



TL/H/6975–2

Top View

Order Number LM567CM
See NS Package Number M08A
Order Number LM567CN
See NS Package Number N08E

©1995 National Semiconductor Corporation TL/H/6975

RRD-B30M115/Printed in U. S. A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage Pin	9V
Power Dissipation (Note 1)	1100 mW
V_8	15V
V_3	−10V
V_3	V_4 + 0.5V
Storage Temperature Range	−65°C to +150°C
Operating Temperature Range	
LM567H	−55°C to +125°C
LM567CH, LM567CM, LM567CN	0°C to +70°C

Soldering Information

Dual-In-Line Package	
Soldering (10 sec.)	260°C
Small Outline Package	
Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics AC Test Circuit, T_A = 25°C, V^+ = 5V

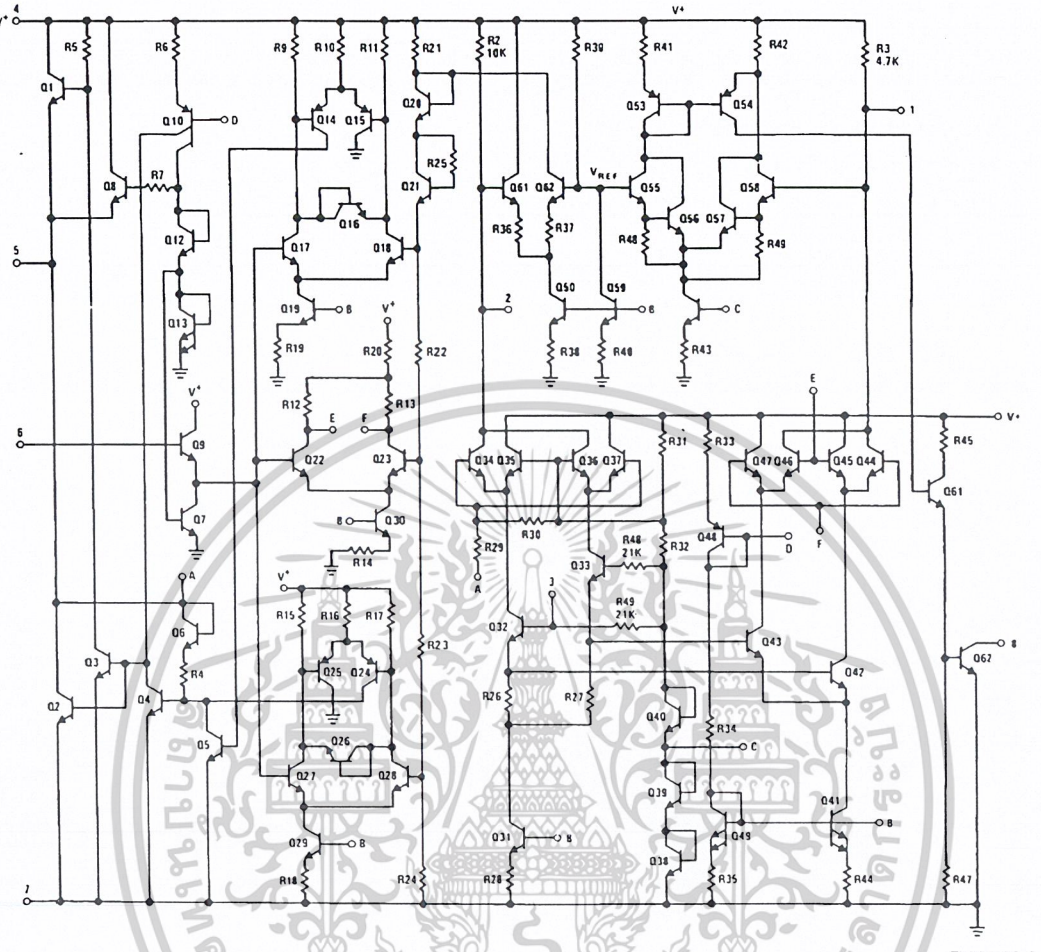
Parameters	Conditions	LM567			LM567C/LM567CM			Units
		Min	Typ	Max	Min	Typ	Max	
Power Supply Voltage Range		4.75	5.0	9.0	4.75	5.0	9.0	V
Power Supply Current Quiescent	R_L = 20k		6	8		7	10	mA
Power Supply Current Activated	R_L = 20k		11	13		12	15	mA
Input Resistance		18	20		15	20		kΩ
Smallest Detectable Input Voltage	I_L = 100 mA, $f_i = f_o$		20	25		20	25	mVrms
Largest No Output Input Voltage	I_C = 100 mA, $f_i = f_o$	10	15		10	15		mVrms
Largest Simultaneous Outband Signal to Inband Signal Ratio			6			6		dB
Minimum Input Signal to Wideband Noise Ratio	B_n = 140 kHz		−6			−6		dB
Largest Detection Bandwidth		12	14	16	10	14	18	% of f_o
Largest Detection Bandwidth Skew			1	2		2	3	% of f_o
Largest Detection Bandwidth Variation with Temperature			±0.1			±0.1		%/°C
Largest Detection Bandwidth Variation with Supply Voltage	4.75 − 6.75V		±1	±2		±1	±5	%V
Highest Center Frequency		100	500		100	500		kHz
Center Frequency Stability (4.75–5.75V)	0 < T_A < 70 −55 < T_A < +125		35 ± 60 35 ± 140			35 ±60 35 ± 140		ppm/°C ppm/°C
Center Frequency Shift with Supply Voltage	4.75V − 6.75V 4.75V − 9V		0.5	1.0 2.0		0.4	2.0 2.0	%/V %/V
Fastest ON-OFF Cycling Rate			f_o/20			f_o/20		
Output Leakage Current	V_8 = 15V		0.01	25		0.01	25	μA
Output Saturation Voltage	e_i = 25 mV, I_8 = 30 mA e_i = 25 mV, I_8 = 100 mA		0.2 0.6	0.4 1.0		0.2 0.6	0.4 1.0	V
Output Fall Time			30			30		ns
Output Rise Time			150			150		ns

Note 1: The maximum junction temperature of the LM567 and LM567C is 150°C. For operating at elevated temperatures, devices in the TO-5 package must be derated based on a thermal resistance of 150°C/W, junction to ambient or 45°C/W, junction to case. For the DIP the device must be derated based on a thermal resistance of 110°C/W, junction to ambient. For the Small Outline package, the device must be derated based on a thermal resistance of 160°C/W, junction to ambient.

Note 2: Refer to RETS567X drawing for specifications of military LM567H version.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Schematic Diagram

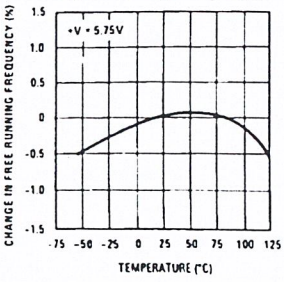


TL/H/6975-3

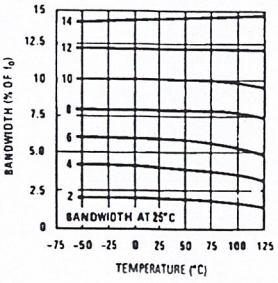
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics

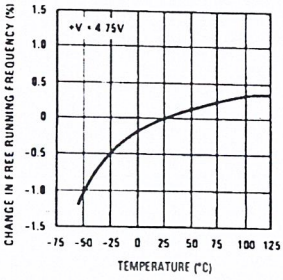
Typical Frequency Drift



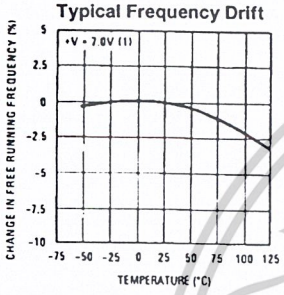
Typical Bandwidth Variation



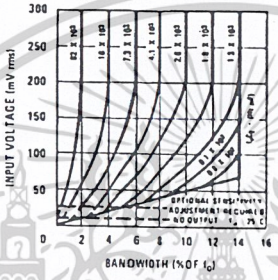
Typical Frequency Drift



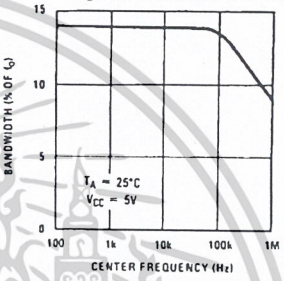
Typical Frequency Drift



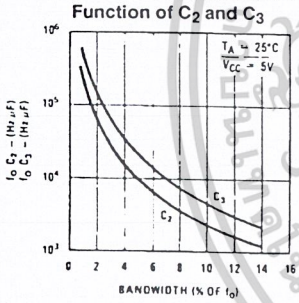
Bandwidth vs Input Signal Amplitude



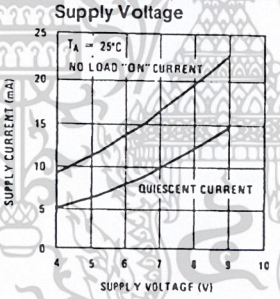
Largest Detection Bandwidth



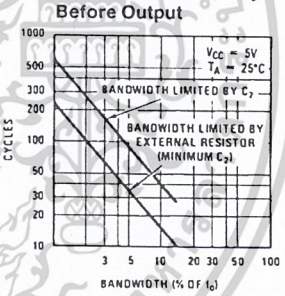
Detection Bandwidth as a Function of C_2 and C_3



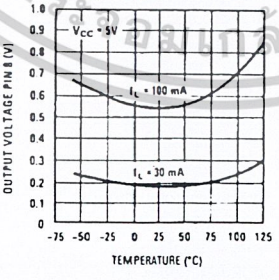
Typical Supply Current vs Supply Voltage



Greatest Number of Cycles Before Output



Typical Output Voltage vs Temperature

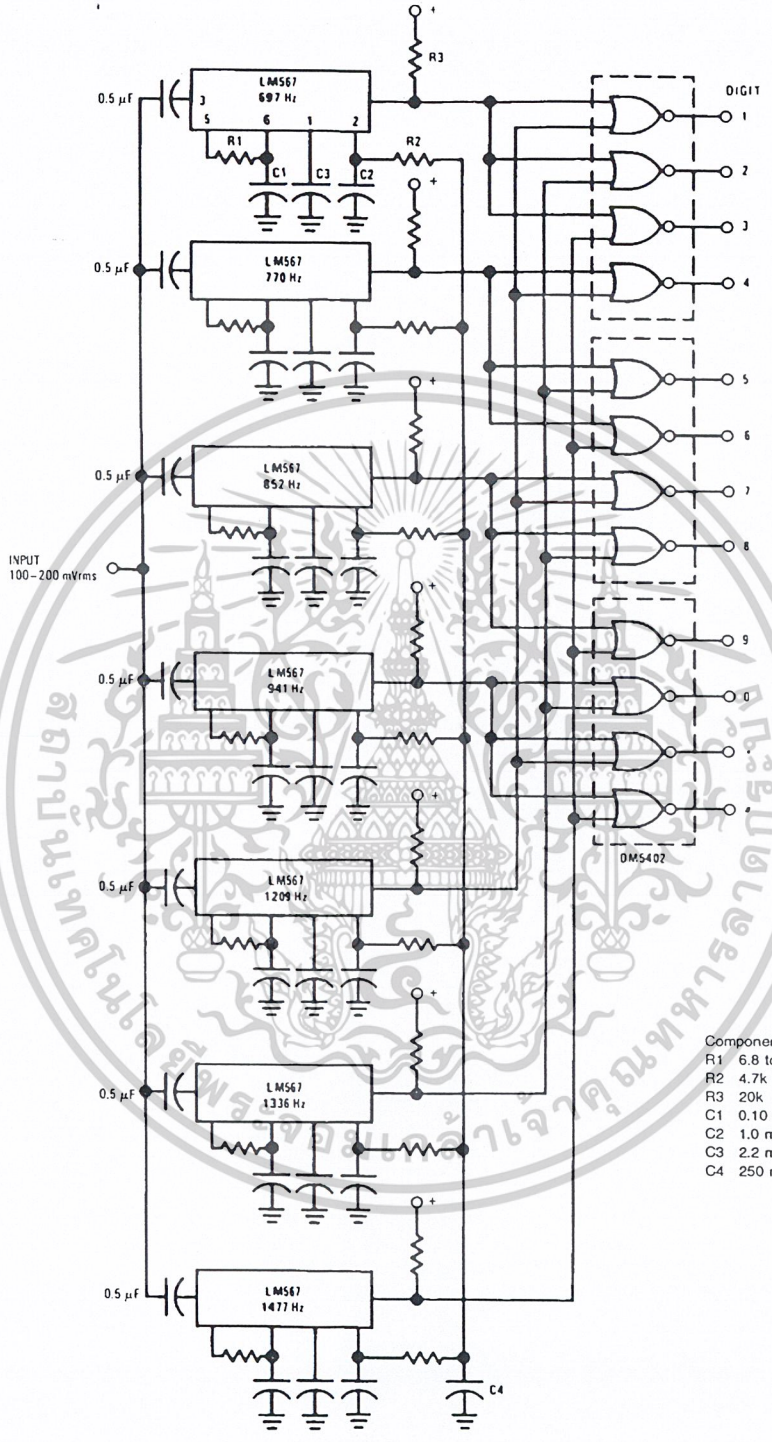


TL/H/6975–4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications

Touch-Tone Decoder



Component values (typ)

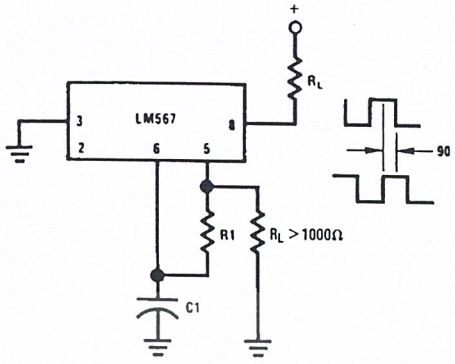
R1	6.8 to 15k
R2	4.7k
R3	20k
C1	0.10 mfd
C2	1.0 mfd 6V
C3	2.2 mfd 6V
C4	250 mfd 6V

TL/H/6975–5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

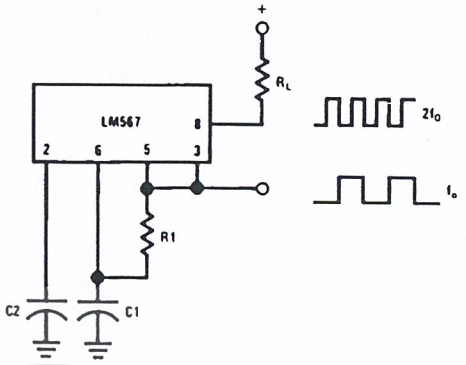
Oscillator with Quadrature Output



Connect Pin 3 to 2.8V to Invert Output

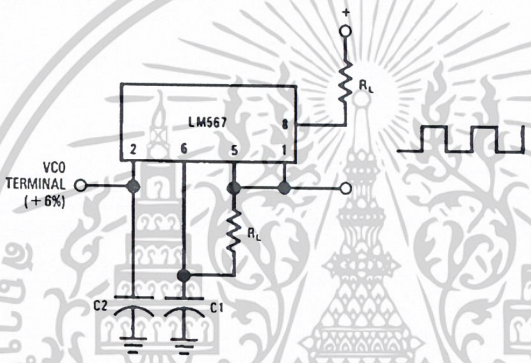
TL/H/6975–6

Oscillator with Double Frequency Output



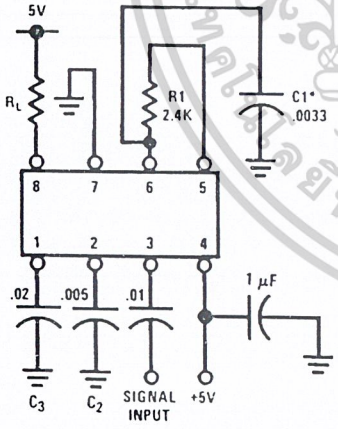
TL/H/6975–7

Precision Oscillator Drive 100 mA Loads



TL/H/6975–8

AC Test Circuit



TL/H/6975–9

f_i = 100 kHz + 5V

***Note:** Adjust for f_o = 100 kHz.

Applications Information

The center frequency of the tone decoder is equal to the free running frequency of the VCO. This is given by

$$f_o \simeq \frac{1}{1.1\, R_1 C_1}$$

The bandwidth of the filter may be found from the approximation

$$BW = 1070 \sqrt{\frac{V_i}{f_o C_2}} \text{ in \% of } f_o$$

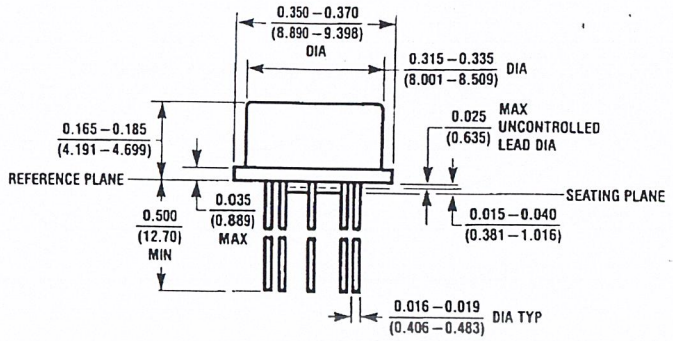
Where:

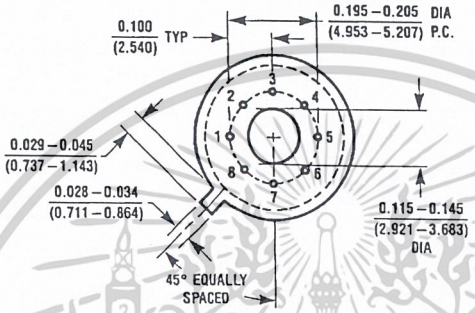
V_i = Input voltage (volts rms), $V_i \leq 200$ mV

C_2 = Capacitance at Pin 2 (μF)

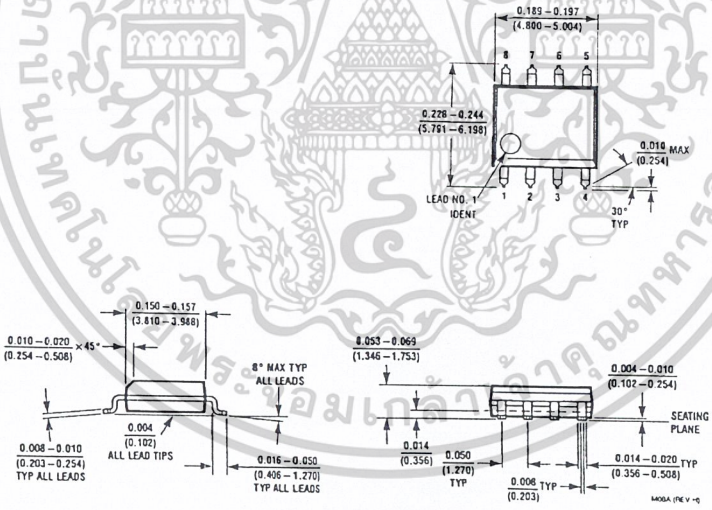
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters)





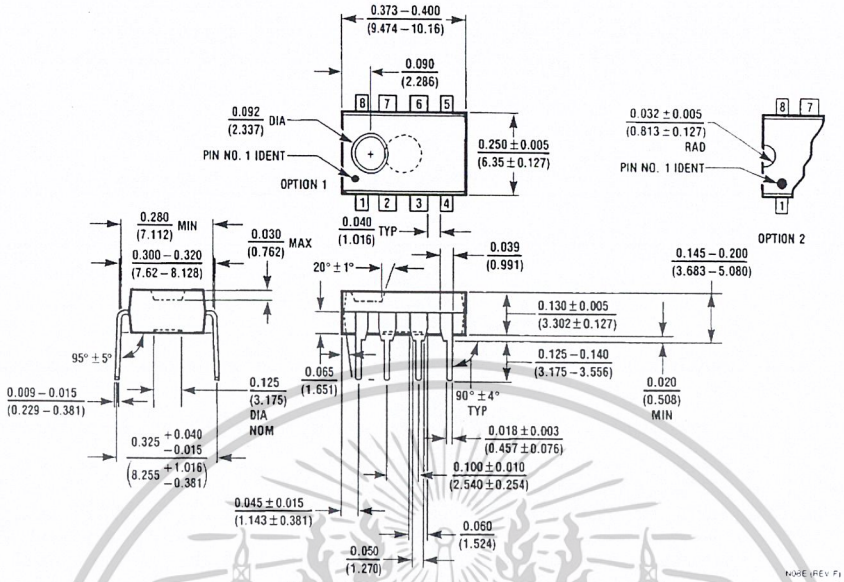
Metal Can Package (H)
Order Number LM567H or LM567CH
NS Package Number H08C



Small Outline Package (M)
Order Number LM567CM
NS Package Number M08A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line Package (N)
Order Number LM567CN
NS Package Number N08E

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมไมโครโปรเซสเซอร์

```
                ORG   0000H
                MOV   PCON, #00H     ;INITIAL SERIAL PORT
                MOV   SCON, #50H     ;BAUD RATE 9600 BAUD
                MOV   TMOD, #20H
                MOV   TH1, #0FDH
                SETB  TR1
                CLR   RI
                CLR   TI
                CLR   P1.0           ;CLEAR RELAY
                CLR   P1.1           ;LOW TRIG
                CLR   P1.2           ;LOW MUTE
                CLR   P1.7           ;CLEAR TONE ENABLE
INITIAL:        MOV   P1, #00H       ;CLEAR PORT OUT
                MOV   P2, #0FFH      ;SET PORT IN
                CALL  DELAY_100mS
MAIN:           CALL  TX_SERVICE
M1:             CALL  RX_SERVICE
                SJMP  MAIN
TX_SERVICE:     MOV R0, #00H
                CLR   P2.1           ;CLEAR RINGING
LOOP:           JNB   P2.1, M1       ;CHECK RINGING BY # LM555
                MOV   A, #'S'
                CALL  SBYTE          ;CALL SEND DATA BACK TO COMPUTER
                MOV   R1, #64H
S1:             JB    RI, S2
                CALL  SBYTE
                CALL  DELAY_100mS
                DJNZ  R1, S1
                LJMP  M1
S2:             CALL  RBYTE
                CJNE  A, #'Y', M1
                CALL  RELAYON
                SETB  P1.2
                CALL  CH_KDTMF
                CALL  CH_KBUSY
                LJMP  NO
RX_SERVICE:     JNB   RI, MAIN       ;CHECK START FROM COMPUTER
                CALL  RBYTE          ;CALL WAIT DATA FROM SERIAL
                CJNE  A, #'S', INITIAL
                MOV   A, #'Y'
                CALL  SBYTE
                CALL  STORE
                CALL  RELAYON
                CALL  DIAL_DTMF
                SETB  P1.2           ;SPEECH ON
                CALL  CH_KBUSY
                LJMP  NO
RBYTE:          JNB   RI, $          ;WAIT DATA FROM COMPUTER
                CLR   RI
                MOV   A, SBUF        ;SHIFT DATA INTO A
                RET
SBYTE:          MOV   SBUF, A        ;SHIFT DATA FROM A
                JNB   TI, $          ;SEND DATA TO COMPUTER
                CLR   TI
                CALL  DELAY
                RET
DELAY:          MOV   R5, #015H
D2:             MOV   R4, #0FFH
D1:             DJNZ  R4, D1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DJNZ R5, D2
            RET
RELAYON:    SETB P1.0          ;SET RELAY ON
            CLR  P1.1
            SETB P1.1
            RET
RELAYOFF:   CLR  P1.0          ;SET RELAY OFF
            CLR  P1.1
            SETB P1.1
            RET
NO:         MOV  A, #'N'
            CALL SBYTE
            CALL RELAYOFF
            LJMP INITIAL
;*****************************************************************
;**   CHECK DTMF BY #MT8870 & SEND TELEPHONE NUMBER TO COMPUTER  **
;*****************************************************************
CH_KDTMF:   MOV  R0, #09H      ;TELEPHONE 9 NUMBER
A1:         MOV  P2, #0FFH     ;SET PORT IN
            JB   P2.6, $
            MOV  R1, #96H
A2:         JB   P2.6, A3
            CALL DELAY_100mS
            DJNZ R1, A2
            LJMP NO
A3:         MOV  A, P2
            ANL  A, #00111100B
            CALL NUMBER
            DJNZ R0, A1
            RET
DELAY_100mS:      MOV R7, #100       ;DO 100 TIMES
DELAY_100mS_1:    MOV R6, #0E6H      ;EACH LOOP=1mS
DELAY_100mS_2:    NOP
                  NOP
                  DJNZ R6, DELAY_100mS_2
                  DJNZ R7, DELAY_100mS_1
                  RET

NUMBER:           CJNE A, #00110100B, N   ;IF NUMBER *
                  LJMP NO
N:                CJNE A, #00001100B, N0  ;IF NUMBER #
                  LJMP NO
N0:               CJNE A, #00100000B, N1  ;IF NUMBER 1
                  MOV  A, #'1'
                  CALL SBYTE
N1:               CJNE A, #00010000B, N2  ;IF NUMBER 2
                  MOV  A, #'2'
                  CALL SBYTE
N2:               CJNE A, #00110000B, N3  ;IF NUMBER 3
                  MOV  A, #'3'
                  CALL SBYTE
N3:               CJNE A, #00001000B, N4        ;IF NUMBER 4
                  MOV  A, #'4'
                  CALL SBYTE
N4:               CJNE A, #00101000B, N5        ;IF NUMBER 5
                  MOV  A, #'5'
                  CALL SBYTE
N5:               CJNE A, #00011000B, N6        ;IF NUMBER 6
                  MOV  A, #'6'
                  CALL SBYTE
N6:               CJNE A, #00111000B, N7        ;IF NUMBER 7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOV  A, #'7'
                    CALL SBYTE
N7:                 CJNE A, #00000100B, N8  ;IF NUMBER 8
                    MOV  A, #'8'
                    CALL SBYTE
N8:                 CJNE A, #00100100B, N9  ;IF NUMBER 9
                    MOV  A, #'9'
                    CALL SBYTE
N9:                 CJNE A, #00010100B, N10 ;IF NUMBER 0
                    MOV  A, #'0'
                    CALL SBYTE
N10:                RET
;************************************
;**      DIAL DTMF FROM #TP5088      **
;************************************
STORE:              CALL RBYTE          ;STORE TELEPOHE 9 NUMBER
                    MOV  31H, A         ;ADDRESS 31H-39H
                    CALL RBYTE
                    MOV  32H, A
                    CALL RBYTE
                    MOV  33H, A
                    CALL RBYTE
                    MOV  34H, A
                    CALL RBYTE
                    MOV  35H, A
                    CALL RBYTE
                    MOV  36H, A
                    CALL RBYTE
                    MOV  37H, A
                    CALL RBYTE
                    MOV  38H, A
                    CALL RBYTE
                    MOV  39H, A
                    RET

DIAL_DTMF:          MOV  A, 31H         ;DIAL TELEPHONE
                    CALL DIAL
                    MOV  A, 32H
                    CALL DIAL
                    MOV  A, 33H
                    CALL DIAL
                    MOV  A, 34H
                    CALL DIAL
                    MOV  A, 35H
                    CALL DIAL
                    MOV  A, 36H
                    CALL DIAL
                    MOV  A, 37H
                    CALL DIAL
                    MOV  A, 38H
                    CALL DIAL
                    MOV  A, 39H
                    CALL DIAL
                    RET

DIAL:               CJNE A, #'1', $+6       ;IF 1 DIAL
                    CALL DIAL1
                    CJNE A, #'2', $+6       ;IF 2 DIAL
                    CALL DIAL2
                    CJNE A, #'3', $+6       ;IF 3 DIAL
                    CALL DIAL3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CJNE A, #'4', $+6       ;IF 4 DIAL
                CALL DIAL4
                CJNE A, #'5', $+6       ;IF 5 DIAL
                CALL DIAL5
                CJNE A, #'6', $+6       ;IF 6 DIAL
                CALL DIAL6
                CJNE A, #'7', $+6       ;IF 7 DIAL
                CALL DIAL7
                CJNE A, #'8', $+6       ;IF 8 DIAL
                CALL DIAL8
                CJNE A, #'9', $+6       ;IF 9 DIAL
                CALL DIAL9
                CJNE A, #'0', $+6       ;IF 0 DIAL
                CALL DIAL0
                RET                     ;RETURN TO MAIN

TONEHOLDING:    MOV  R7, #03H           ;TONE HOLDING PERIOD LOOP
DL3:            MOV  R6, #0FFH
DL2:            MOV  R5, #0FFH
DL1:            DJNZ R5, DL1
                DJNZ R6, DL2
                DJNZ R7, DL3
                RET
DELAYNUM:       MOV  R3, #0FFH    ;DELAY NEXT NUMBER
DELAY2:         MOV  R2, #0FFH
DELAY1:         DJNZ R2, DELAY1
                DJNZ R3, DELAY2
                RET

DIAL0:          MOV  P1, #00101001B
                SETB P1.7               ;SET TONE ENABLE
                CALL TONEHOLDING
                CLR  P1.7               ;CLEAR TONE ENABLE
                CALL DELAYNUM
                RET
DIAL1:          MOV  P1, #01000001B
                SETB P1.7               ;SET TONE ENABLE
                CALL TONEHOLDING
                CLR  P1.7               ;CLEAR TONE ENABLE
                CALL DELAYNUM
                RET
DIAL2:          MOV  P1, #00100001B
                SETB P1.7               ;SET TONE ENABLE
                CALL TONEHOLDING
                CLR  P1.7               ;CLEAR TONE ENABLE
                CALL DELAYNUM
                RET
DIAL3:          MOV  P1, #01100001B
                SETB P1.7               ;SET TONE ENABLE
                CALL TONEHOLDING
                CLR  P1.7               ;CLEAR TONE ENABLE
                CALL DELAYNUM
                RET
DIAL4:          MOV  P1, #00010001B
                SETB P1.7               ;SET TONE ENABLE
                CALL TONEHOLDING
                CLR  P1.7               ;CLEAR TONE ENABLE
                CALL DELAYNUM
                RET
DIAL5:          MOV  P1, #01010001B
                SETB P1.7               ;SET TONE ENABLE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CALL TONEHOLDING
                    CLR  P1.7                ;CLEAR TONE ENABLE
                    CALL DELAYNUM
                    RET
DIAL6:              MOV  P1, #00110001B
                    SETB P1.7                ;SET TONE ENABLE
                    CALL TONEHOLDING
                    CLR  P1.7                ;CLEAR TONE ENABLE
                    CALL DELAYNUM
                    RET
DIAL7:              MOV  P1, #01110001B
                    SETB P1.7                ;SET TONE ENABLE
                    CALL TONEHOLDING
                    CLR  P1.7                ;CLEAR TONE ENABLE
                    CALL DELAYNUM
                    RET
DIAL8:              MOV  P1, #00001001B
                    SETB P1.7                ;SET TONE ENABLE
                    CALL TONEHOLDING
                    CLR  P1.7                ;CLEAR TONE ENABLE
                    CALL DELAYNUM
                    RET
DIAL9:              MOV  P1, #01001001B
                    SETB P1.7                ;SET TONE ENABLE
                    CALL TONEHOLDING
                    CLR  P1.7                ;CLEAR TONE ENABLE
                    CALL DELAYNUM
                    RET
;************************************
;**    CHECK BUSY TONE BY #LM567     **
;************************************
CH_KBUSY:    MOV  R0, #64H
             JNB  RI, CH1
             CALL RBYTE
             CJNE A, #'N', CH1
             CALL RELAYOFF
             LJMP INITIAL
CH1:         JB   P2.0, CH_KBUSY
             CALL DELAY_2_50mS
             DJNZ R0, CH1
             MOV  R1, #78H
CH2:         CALL DELAY_2_50mS
             DJNZ R1, CH2
             JNB  P2.0, CH_KBUSY
             MOV  R3, #0F0H
CH3:         CALL DELAY_2_50mS
             DJNZ R1, CH3
             JB   P2.0, CH_KBUSY
             RET
DELAY_2_50mS:     MOV  R7, #0AH      ;DELAY TIME 2.50mS
L2:               MOV  R6, #73H
L1:               DJNZ R6, L1
                  DJNZ R7, L2
                  RET
                  END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรม Visual Basic

```
Public Countnumber As Integer, Count_Send
Public Waitingtime As Variant
Public receive As Boolean
Public Effortnum  As Integer
Public User_Request As Boolean, Voice_connect
Public Sound_Receive1 As String, Sound_Receive2, Sound_Receive3,
Wave_Define, Client_Send
Option Explicit
#If Win32 Then
Private Declare Function sndPlaySoundA Lib "WinMM" (ByVal SoundName
As String, ByVal Flags As Integer) As Integer
#Else
Private Declare Function sndPlaySound Lib "WinSystem" (ByVal
SoundName As String, ByVal Flags As Integer) As Integer
#End If
Public CLOSINGAPPLICATION As Boolean          ' Application status flag
Public wStream As Object
Public Function Default_Sound()
Option Explicit
' Application User Defined Types...
' Sound Format
Public Const WAVE_FORMAT_PCM = &H1' Microsoft Windows PCM Wave Format
Public Const TIMESLICE = 0.2                  ' Time Slicing 1/5 Second
' Application Constants...
Public Const NoOfRings = 1                    ' Number Of Times In/Out
Bound Calls Ring...
Public Const phoneHungUp = 3                  ' Hangup Status Icon...
Public Const phoneRingIng = 2                 ' Ringing Status Icon...
Public Const phoneAnswered = 1                ' Answered Status Icon...
Public Const mikeNO = 6
Public Const mikeOFF = 7
Public Const mikeON = 8
Public Const speakNO = 9
Public Const speakOFF = 10
Public Const speakON = 11
Public Const RingInId = 101                   ' Ringing InBound Sound...
Public Const RingOutId = 102                  ' Ringing OutBound Sound...
' Toolbar constants...
Public Const tbCALL = 2
Public Const tbHANGUP = 3
Public Const tbAUTOANSWER = 5
'== flag values for wFlags parameter
=================================
Public Const SND_SYNC = &H0           '  play synchronously (default)
Public Const SND_ASYNC = &H1          '  play asynchronously
Public Const SND_NODEFAULT = &H2      '  don't use default sound
Public Const SND_MEMORY = &H4'  lpszSoundName points to a memory file
Public Const SND_LOOP = &H8           '  loop the sound until next
sndPlaySound
Public Const SND_NOSTOP = &H10        '  don't stop any currently
playing sound
'== MCI Wave API Declarations
================================================
Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA"
(ByVal SoundData As Any, ByVal uFlags As Long) As Long
Option Explicit
Public Sub InitServerList(ServerList As ComboBox)
    ' Populate Server List Box...
     ServerList.AddItem "127.0.0.1"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
'-------------------------------------------------------------
Public Sub DebugSocket(TCPSocket As Winsock)
' Prints Information In A TCP Socket, For Debugging TCP Events...
'-------------------------------------------------------------
    Debug.Print "TCPSocket.RemoteHost", TCPSocket.RemoteHost
    Debug.Print "TCPSocket.RemoteHostIP", TCPSocket.RemoteHostIP
    Debug.Print "TCPSocket.RemotePort", TCPSocket.RemotePort
    Debug.Print "TCPSocket.LocalHostName", TCPSocket.LocalHostName
    Debug.Print "TCPSocket.LocalIP", TCPSocket.LocalIP
    Debug.Print "TCPSocket.LocalPort", TCPSocket.LocalPort
    Debug.Print "TCPSocket.State", TCPSocket.State
    Debug.Print
"=================================================="
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------------
Public Sub ResPlaySound(ResourceId As Long)
' Uses Sound Play Sound To Play Back PreRecorded WaveFiles
'-------------------------------------------------------------------
    Dim sndBuff As String
'-------------------------------------------------------------------
    sndBuff = StrConv(LoadResData(ResourceId, "WAVE"), vbUnicode)
    Call sndPlaySound(sndBuff, SND_SYNC Or SND_MEMORY)
'-------------------------------------------------------------------
End Sub
'-------------------------------------------------------------------

'-------------------------------------------------------------
Public Sub AddConnectionToList(Socket As Winsock, ConnList As
ListBox)
' Adds A Connection Reference To A ListBox -
[(Server)(LocalPort)(RemotePort)]
'-------------------------------------------------------------
    Dim MemberID As String                  ' Connection
Reference Variable
'-------------------------------------------------------------
    ' Create MemberID From HostName and RemoteIP
    MemberID = Socket.RemoteHostIP & "  [" & _
               Format(Socket.RemotePort, "0") & "] - [" & _
               Format(Socket.LocalPort, "0") & "]"
    ConnList.AddItem MemberID               ' Add New Member To List
    ConnList.ItemData(ConnList.NewIndex) = Socket.Index
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Sub RemoveConnectionFromList(Socket As Winsock, ConnList As
ListBox)
' Removes A Connection Reference From A ListBox
'-------------------------------------------------------------
    Dim Conn As Long            ' Connection Array Element Variable
    Dim MemberID As String      ' Connection Reference Variable
'-------------------------------------------------------------
    ' Create MemberID From HostName and RemoteIP
    MemberID = Socket.RemoteHostIP & "  [" & _
               Format(Socket.RemotePort, "0") & "] - [" & _
               Format(Socket.LocalPort, "0") & "]"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    For Conn = 0 To ConnList.ListCount - 1          ' Search Each
Member In List
        If (ConnList.List(Conn) = MemberID) Then    ' Look For
MemberID In List
            ConnList.RemoveItem Conn       ' Remove MemberID From List
        End If
    Next                                           ' Next Connection
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Sub GetIdxFromMemberID(Sockets As Variant, MemberID As String,
Index As Long)
'-------------------------------------------------------------
    Dim Idx As Long                         ' Socket cntl index
    Dim LocPortID As Long                   ' Local Port ID
    Dim RemPortID As Long                   ' Remote Port ID
    Dim RemoteIP As String                  ' Remote Host IP address
    Dim sStart As Long                      ' Substring begin position
    Dim sEnd As Long                        ' Substring end position
    Dim Socket As Winsock                   ' Winsock socket
'-------------------------------------------------------------
    sStart = 1
    sEnd = InStr(1, MemberID, " ") - 1 ' Get end of remote ip address
    If (sEnd > 1) Then
        RemoteIP = Mid(MemberID, sStart, sEnd)  ' Get remote host ip
address
        sStart = InStr(sEnd, MemberID, "[") + 1     ' Get start of
remote port
        If (sStart > 1) Then                        ' If Start found
            sEnd = InStr(sStart, MemberID, "]") - 1 ' Get end of
remote port
            If (sEnd > 2) Then                      ' If end found
                RemPortID = Val(Mid(MemberID, sStart, sEnd)) ' Get
RemotePort
                sStart = InStr(sEnd, MemberID, "[") + 1      ' Get
start of local port
                If (sStart > 1) Then
                    sEnd = InStr(sStart, MemberID, "]") - 1  ' Get
end of local port
                    If (sEnd > 2) Then              ' If End Found
                        LocPortID = Val(Mid(MemberID, sStart, sEnd))
' Extract local port
                        For Each Socket In Sockets
                        If ((Socket.RemoteHostIP = RemoteIP) And _
                        (Socket.RemotePort = RemPortID) And _
                        (Socket.LocalPort = LocPortID) And _
                        (Socket.Index > 0)) Then    ' Was a match found???
                        Index = Socket.Index        ' Save and return index
                        Exit Sub                    ' Done... exit
                                End If
                            Next
                        End If
                    End If
                End If
            End If
        End If
'-------------------------------------------------------------
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'-------------------------------------------------------------
Public Invalid As Integer
Public rsToAffect As Long
Public sqlQuiz As String
Public Echo As Boolean          ' Echo On/Off flag.
Public SecRXTX As Integer
Public ModeRXTX As Integer
Public ComOnOff As Boolean
Public StatusComm As String
Public ChkStopLoopUnLess As Boolean
Public EventMsg As String, ErrorMsg As String
Public bQuit  As Boolean
Option Explicit
'== TCP Port Array Processing Const.s
==================================
Public Const MINTCP = 1' Minimum index for tcpsocket control instance
Public Const MAXTCP = 3' Maximum index for tcpsocket control instance
Public Const VOICEPORT = 1000         ' Voice chat Port To Listen On...
Public Const NULLPORTID = 0           ' Null Port ID - A Port ID That
Will Never Be Used...
Option Explicit
'-------------------------------------------------------------
Public Function InstanceTCP(TCPArray As Variant) As Long
'-------------------------------------------------------------
    Dim Ind As Long                   ' Array Index Var...
'-------------------------------------------------------------
    InstanceTCP = -1                  ' Set Default Value
    On Error GoTo InitControl     ' IF Error Then Control Is Available
    For Ind = MINTCP To MAXTCP    ' For Each Member In TCPArray() + 1
        If (TCPArray(Ind).Name = "") Then          ' If Control Is
Not Valid Then..
        End If                    ' ..A Runtime Error Will Occure
    Next                          ' Search Next Item In Array
'-------------------------------------------------------------
InitControl:                      ' Initialize New Control
'-------------------------------------------------------------
    On Error GoTo ErrorHandler    ' Enable Error Handling...
    If ((Ind >= MINTCP) And (Ind <= MAXTCP)) Then    ' Check to make
sure index value is with in range
        Load TCPArray(Ind)        ' Create New Member In TCPArray
        InstanceTCP = Ind         ' Return New TCPctl Index
    End If
    Exit Function                                    ' Exit
'-------------------------------------------------------------
ErrorHandler:                                        ' Handler
'-------------------------------------------------------------
    Debug.Print Err.Number, Err.Description          ' Debug Errors
    Resume Next                                      ' Ignore Error
And Continue
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------

'-------------------------------------------------------------------
Public Function Connect(Socket As Winsock, RemHost As String, RemPort
As Long) As Boolean
' Attempts To Open A Socket Connection
'-------------------------------------------------------------------
    Connect = False                   ' Set default return code
    Call CloseListen(Socket)          ' Stop Listening On LocalPort
    Socket.LocalPort = 0        ' Not necessary, but done just in case
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Call Socket.Connect(RemHost, RemPort)        ' Connect To Server
    Do While ((Socket.State = sckConnecting) Or _
              (Socket.State = sckConnectionPending) Or _
              (Socket.State = sckResolvingHost) Or _
              (Socket.State = sckHostResolved) Or _
              (Socket.State = sckOpen))     ' Attempting To Connect...
        DoEvents                                    ' Post Events
    Loop                                            ' Keep Waiting
    Connect = (Socket.State = sckConnected)      ' Did Socket Connect
On Port...
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Function Listen(Socket As Winsock) As Long
' Starts Listening For A Remote Connection Request On The LocalPort
ID
'-------------------------------------------------------------
    If (Socket.State <> sckListening) Then       ' Is Socket Already
Listening
        If (Socket.LocalPort = 0) Then           ' If local port is
not initialized then...
            Socket.LocalPort = VOICEPORT         ' Set standard
application port
        End If
        Call Socket.Listen                  ' Listen On Local Port...
    End If
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Function CloseListen(Socket As Winsock) As Long
' Stops Listening On A LocalPort For A Remote Connection Request...
'-------------------------------------------------------------
    If (Socket.State = sckListening) Then        ' Is Socket
Listening?
        Socket.Close                             ' Close Listen
    End If
'-------------------------------------------------------------
End Function
'-------------------------------------------------------------

'-------------------------------------------------------------
Public Sub Disconnect(Socket As Winsock)
' Disconnects A Remote WinSock TCP/IP Connection If Connected
'-------------------------------------------------------------
    If (Socket.State <> sckClosed) Then  ' Is Socket Already Closed?
        Socket.Close                             ' Close Socket
    End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------
Sound_Receive1 = "C:\My Documents\Dial.wav"
Sound_Receive2 = "C:\My Documents\Busy.wav"
Sound_Receive3 = "C:\My Documents\Redial.wav"
End Function
Public Function Decide_Receive(ByVal Desphone As String, Desip As
String)
Select Case Desphone
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Case "A"
                    User_Request = False
                    frmMain.Phonenum.Caption = ""
                    Call Reset_variable
                    frmMain.Client_Send = "Active"
                    Desip = Right$(Desip, Len(Desip) - 1)
                    If Left(Desip, 1) = 1 Then
                       frmMain.txtServer.Text = Desip
                       Exit Function
                    End If
                       frmMain.FontBold = True
                       frmMain.Status.Alignment = 2
                       frmMain.Status.Caption = Desip
            Case "0"
                    If frmMain.Client_Send <> "Active" Then Exit
Function
                    If frmMain.Phonenum.Caption = "" Then
                       frmMain.Phonenum.Caption = Desphone
                       Exit Function
                    End If
                    frmMain.Phonenum.Caption = frmMain.Phonenum.Caption
& Desphone
                    If Len(frmMain.Phonenum.Caption) = 9 Then
                       Call frmMain.Recall(frmMain.Phonenum.Caption)
                       Exit Function
                    End If
         Case "Y"
                    If frmMain.Client_Send <> "Active" Then Exit
Function
                    wStream.recording = True
                   Call frmMain.Play_sound
         Case "B"
                    If frmMain.Client_Send <> "Active" Then Exit
Function
                    If frmMain.Voice_connect = True Then
                    Call Stop_sound
                    Status = sndPlaySoundA(frmMain.Sound_Receive2,
SND_SYNC Or SND_NODEFAULT)
                    Status.Caption = "Line Busy "
                    wStream.recording = True
                    Call Play_sound
                    End If
                    Status = sndPlaySoundA(frmMain.Sound_Receive2,
SND_SYNC Or SND_NODEFAULT)
                    Call frmMain.Reset_variable
                  Status.Caption = "Line Busy "
       Case "N"
                    If frmMain.Client_Send <> "Active" Then Exit
Function
                    Call frmMain.Stop_sound
       Case 1, 2, 3, 4, 5, 6, 7, 8, 9
                    If frmMain.Client_Send <> "Active" Then Exit
Function
                       frmMain.Phonenum.Caption =
frmMain.Phonenum.Caption & Desphone
                    If Len(frmMain.Phonenum.Caption) = 9 Then
                          Call frmMain.Recall(frmMain.Phonenum.Caption)
                          Exit Function
                    End If
                  End Select
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Function Play_sound()
If Command1.Caption = "Stop" Then
Exit Function
End If
If Command1.Caption = "Play" Then
Call Command1_Click
End If
End Function
Public Function Set_Sound(Receiver1 As ComboBox, Redialer As
ComboBox, Busy As ComboBox)
                                Sound_Receive1 = Receiver1.Text
                                Sound_Receive2 = Redialer.Text
                                Sound_Receive3 = Busy.Text
                               If Receiver1.Text = "" Then
Receiver1.Text = "C:\My Documents\Dial.wav"
                               If Redialer.Text = "" Then Redialer.Text
= "C:\My Documents\Redial.wav"
                               If Busy.Text = "" Then Busy.Text =
"C:\My Documents\Busy.wav"
                           End Function
Public Function Set_Winsock(ByVal Port As String, Potocol As Integer,
IP As String)
On Error GoTo error1
                                If frmMain.Winsock1.State <> sckClosed
Then frmMain.Winsock1.Close
                                frmMain.Winsock1.LocalPort = Port
                                frmMain.Winsock1.RemotePort = Port
                                If Potocol = 0 Then
frmMain.Winsock1.Protocol = sckTCPProtocol
                                If Potocol = 1 Then
frmMain.Winsock1.Protocol = sckUDPProtocol
                                frmMain.Winsock1.RemoteHost = IP
                                frmMain.txtServer.Text =
frmMain.Winsock1.RemoteHost
                                frmMain.Text1.FontBold = True
                                frmMain.Text1.Text =
frmMain.Winsock1.LocalPort
error1:
                           If Err = 0 Then Exit Function
                           MsgBox "Error Number: " & Err & "   " & Error,
vbOKOnly
End Function
Public Function Reset_variable()
                                Count_Send = 0 'จำนวนที่ส่งไปแล้ว
                                receive = False  'รับหมายเลยแล้ว
                                Client_Send = "Non_Active"' ต้นทางยังไม่ติดต่อมา
                                User_Request = False        'ยังไม่มีผู้ใช้เรียกเข้ามา
                                Effortnum = 0         'โทรแล้ว 0 ครั้ง
                                Countnumber = 0 ' จำนวนตัวเลข
                                frmMain.Status.Alignment = 2
                                frmMain.Text2.FontBold = True
                                frmMain.Text2.Text = Winsock1.LocalIP
                                frmMain.Phonenum.FontBold = True
                                frmMain.Phonenum.Alignment = 2
                                frmMain.Phonenum.Caption = ""
End Function
Public Function Set_Comport(ByVal ComPort As String, Baud As String,
Handchecking As Integer)
On Error GoTo error1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    If frmMain.MSComm1.PortOpen = True
Then frmMain.MSComm1.PortOpen = False
                                    If ComPort = "Com1" Then
frmMain.MSComm1.CommPort = 1
                                    If ComPort = "Com2" Then
frmMain.MSComm1.CommPort = 2
                                    If Baud = "9600" Then
frmMain.MSComm1.Settings = "9600 , N, 8, 1"
                                    If Baud = "1200" Then
frmMain.MSComm1.Settings = "1200 , N, 8, 1"
                                    If Baud = "14400" Then
frmMain.MSComm1.Settings = "11400,N,8,1"
                                    If Baud = "4800" Then
frmMain.MSComm1.Settings = "4800,N,8,1"
                                    If Baud = "2400" Then
frmMain.MSComm1.Settings = "2400,N,8,1"
                                    If Handchecking = 1 Then
frmMain.MSComm1.Handshaking = comNone
                                    If Handchecking = 2 Then
frmMain.MSComm1.Handshaking = comXOnXoff
                                    If Handchecking = 3 Then
frmMain.MSComm1.Handshaking = comRTS
                                    If Handchecking = 4 Then
frmMain.MSComm1.Handshaking = comRTSXOnXOff
                                    frmMain.MSComm1.RThreshold = 1
                                    If frmMain.MSComm1.PortOpen = False
Then frmMain.MSComm1.PortOpen = True
error1:
                        If Err = 0 Then Exit Function
                        MsgBox "Error Number: " & Err & "   " & Error,
vbOKOnly
                        If Err = 8005 Then
                        MsgBox Error, vbOKOnly
                        Exit Function
                        End If
                        End Function
Public Function Stop_sound()
If Command1.Caption = "Play" Then
' If Voice_connect = False Then Call Sconnect_button_Click
   Exit Function
End If
  Call Command1_Click
End Function
Private Function Winsock_show(Socket_State As String)
Select Case Socket_State
  Case "Close"
            frmMain.Text4.FontBold = True
            frmMain.Text4.Text = "Closed"

'.....................................................................
....
   Case "Connect"
            frmMain.Text4.FontBold = True
            frmMain.Text4.Text = "Connected"

'.....................................................................
......
   Case "Request"
            frmMain.Text4.FontBold = True
            frmMain.Text4.Text = "Request Connect"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'.....
......
     Case "Send"
            frmMain.Text4.FontBold = True
            frmMain.Text4.Text = "Send Completed"

'.....
......
    Case "Error"
            frmMain.Text4.FontBold = True
            frmMain.Text4.Text = "Error State"

'.....
......
 Case "Data"
            frmMain.Text4.FontBold = True
            frmMain.Text4.Text = "Data Arrival"

'.....
......
    End Select
   End Function
Public Function Delay_Time(T As Long)
  Dim Dummy As Long
          Dummy = 0
          Do While Dummy < T * 100000
           Dummy = Dummy + 1
           DoEvents
          Loop
   End Function
Private Function Press_button(S As String)
'.....
.....
......
                If receive = False Then
                   Dim Buffer As Variant
                        Buffer = S
                        MSComm1.Output = Buffer
                        Countnumber = Countnumber + 1
                        Countnum (Countnumber)
                        Phonenum.Caption = Phonenum.Caption &
S
                End If
'.....
.....
.............
        If receive = True Then
            Count_Send = Count_Send + 1
            Winsock1.SendData S

'.....
.....................
                                If Len(Phonenum.Caption) = 9 Then
                                    Phonenum.Caption = ""
                                    Phonenum.Caption = S
                                Exit Function
                                End If

'.....
.....................
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Phonenum.Caption = Phonenum.Caption & S
        End If
'.....................................................................
.....................................................................
.............
End Function
Public Function Recall(Phone As String)
                Dim i As Integer
                Dim Buffer As Variant, timmimg
                Dim Dummy As String
                Effortnum = Effortnum + 1

'.....................................................................
.........................
                If Effortnum = 1 Then
                Status.Alignment = 2
                Status.FontBold = True
                Status.Caption = "Dialing"
                End If

'.....................................................................
.........................
                If Effortnum = 2 Then
                Status.Alignment = 2
                Status.FontBold = True
                Status.Caption = "Redialing"
                End If
                Phonenum.Caption = ""

'.....................................................................
.........................
                  Buffer = "S"
                  MSComm1.Output = Buffer
                  Call Delay_Time(1)
                                For i = 1 To 9 Step 1
                                Dummy = Phone
                                Dummy = Left$(Dummy, i)
                                Dummy = Right(Dummy, 1)
                                Buffer = Dummy
                                MSComm1.Output = Buffer
                                Phonenum.Caption = Phonenum.Caption &
Dummy
                               Call Delay_Time(0.5)
            Next i
  Waitingtime = Minute(Now)
  receive = False
End Function
Private Function Countnum(num As Integer)
If num = 1 Then
Phonenum.Caption = ""
Status.Alignment = 2
Status.FontBold = True
Status.Caption = "Connecting To"
End If
If num = 9 Then
Status.Caption = "Calling"
Countnumber = 0 'reset counter
Effortnum = Effortnum + 1 'set try number
Waitingtime = Minute(Now)
End If
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub button0_Click()
Call Press_button(0)
End Sub
Private Sub button1_Click()
Call Press_button(1)
End Sub
Private Sub button2_Click()
Call Press_button(2)
End Sub
Private Sub button3_Click()
Call Press_button(3)
End Sub
Private Sub button4_Click()
Call Press_button(4)
End Sub
Private Sub button5_Click()
Call Press_button(5)
End Sub
Private Sub button6_Click()
Call Press_button(6)
End Sub
Private Sub button7_Click()
Call Press_button(7)
End Sub
Private Sub button8_Click()
Call Press_button(8)
End Sub
Private Sub button9_Click()
Call Press_button(9)
End Sub
Private Sub buttonN_Click()
If receive = False Then
Dim Buffer As Variant
Buffer = "N"
MSComm1.Output = Buffer
End If
End Sub
Private Sub buttonS_Click()
                    If receive = False Then
                    Dim Buffer As Variant
                    Status.Caption = " Enter  Phonenumber"
                    Buffer = "S"
                    MSComm1.Output = Buffer
                    If User_Request <> False Then User_Request
= False
                    Effortnum = 0
                    Buffer = Second(Now)
                                        Do While Second(Now) -
Buffer > 0.5
                                        DoEvents
                                        Loop
                       Exit Sub
                   End If
'.........................ส่ง connection ไปที่ winsock
8000.....................................................
                  If receive = True Then
                     Call frmMain.Reset_variable
                     If Winsock1.State = sckConnected Then
Winsock1.SendData "AUser Oncom"
                  End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'-------------------------------------------------------------
Private Sub cmdTalk_Click()                                    ' Activates
Audio PlayBack
'-------------------------------------------------------------
    Dim rc As Long                                  ' Return Code Variable
    Dim iPort As Integer                            ' Local Port
    Dim itm As Integer                              ' Current listitem
'-------------------------------------------------------------
    If (Not wStream.Playing And wStream.PlayDeviceFree And _
        Not wStream.recording And wStream.RecDeviceFree) Then '
Validate Audio Device Status
        wStream.Playing = True              ' Turn Playing Status On
        cmdTalk.Caption = "&Playing"    ' Modify Button Status Caption
        Screen.MousePointer = vbHourglass  ' Set Pointer To HourGlass
        iPort = wStream.StreamInQueue
        Do While (iPort <> NULLPORTID)    ' While socket ports have
data to playback
            For itm = 0 To ConnectionList.ListCount - 1 ' Search for
listitem currently playing sound data
                If (ConnectionList.ItemData(itm) = iPort) Then ' If a
match is found...
                    ConnectionList.TopIndex = itm        ' Set that
listitem to top of listbox
                    ConnectionList.Selected(itm) = True ' Select
listitem to show who is currently talking...
                    Exit For                    ' Quit listitem search
                End If
            Next                                ' Check next listitem
            rc = wStream.PlayWave(Me.hWnd, iPort)        ' Play wave
data in iPort...
            Call wStream.RemoveStreamFromQueue(iPort)    ' Remove
PortID From PlayWave Queue
            iPort = wStream.StreamInQueue
        Loop            ' Search for next socket in playback queue
        ConnectionList.TopIndex = 0          ' Reset top image...
        If (ConnectionList.ListCount > 0) Then
            ConnectionList.Selected(0) = True            ' Deselect
previously listitem
            ConnectionList.Selected(0) = False           ' Deselect
currently selected listitem
        End If
        Screen.MousePointer = vbDefault    ' Set Pointer To Normal
        cmdTalk.Caption = "&Talk"       ' Modify Button Status Caption
        wStream.Playing = False          ' Turn Playing Status Off
    End If
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

Private Sub cmdTalk_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    wStream.recording = False            ' Stop Recording
End Sub
Public Sub Command1_Click()
  If Command1.Caption <> "Stop" And frmMain.Voice_connect = True Then
          Command1.Caption = "Stop"
          Call Option1_GotFocus
          Exit Sub
  End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'.....................................................................
.....................................................
    If Command1.Caption <> "Play" And frmMain.Voice_connect = True
Then
              Command1.Caption = "Play"
               wStream.recording = False
             Exit Sub
    End If

'.....................................................................
.....................................................
  Client_Send = False
 End Sub
Private Sub Command15_Click()
txtServer.Text = txtServer.Text + "."
End Sub
Private Sub Command2_Click()
'.....................................................................
.....................................................................
.............
              If receive = False Then
                   Command2.Caption = "Stop"
                   receive = True
                  Phonenum.Caption = ""
                  Status.Caption = "Send To Com "
               Exit Sub
            End If
'.....................................................................
.....................................................................
.....................
            If receive = True Then
                 Command2.Caption = "Send"
                 receive = False
                 Status.Caption = "Send To Port"
        End If
'.....................................................................
.....................................................................
.................
End Sub
Private Sub Command3_Click()
On Error GoTo error1
If Command3.Caption <> "Listening" Then
              Winsock1.LocalPort = Text1.Text
              Command3.Caption = "Listening"
              Winsock1.Listen
              Status.Caption = "Winsock1 Lisening"
               Exit Sub
End If
If Command3.Caption <> "Listen" Then
               Winsock1.Close
               Winsock1.LocalPort = Text1.Text
               Command3.Caption = "Listen"
               Status.Caption = "Winsock1 Stanby"
               Exit Sub
End If
error1:
     If Err = 0 Then Exit Sub
     MsgBox "Error Number: " & Err & "   " & Error, vbOKOnly
End Sub
Public Sub Command4_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'.....................................................................
.....................................................................
.....................
  On Error GoTo error1
    If Command4.Caption = "Stop" Then
                                Command4.Caption = "Connect"
                                If Winsock1.State <> sckClosed Then
                                   Winsock1.Close
                                End If
                               Status.Caption = "Winsock1.close"
                               Exit Sub
 End If
 If Command4.Caption = "Connect" Then
                               If Winsock1.State <> sckClosed Then
Winsock1.Close
                               Label1.Caption = "Client:"
                               Command4.Caption = "Stop"
                               Winsock1.RemoteHost = txtServer.Text
                               Winsock1.RemotePort = Text1.Text
                               Winsock1.Connect
                              Status.Caption = "Winsock1.Connect"
                              Phonenum.Caption = ""
                              Label1.Caption = "Client:"
                              Exit Sub
      End If
error1:
     If Err = 0 Then Exit Sub
     MsgBox "Error Number: " & Err & "  " & Error, vbOKOnly
End Sub
Private Sub Auto_Anser_Click()
Dim rc As Long
If Auto_Anser.Caption = "Auto" Then
    Auto_Anser.Caption = "Manual"
   Exit Sub
End If
    Auto_Anser.Caption = "Auto"
End Sub
Private Sub Command6_Click()
frmVBCom.Show
End Sub
Public Sub connectionlist_Click()
    Sconnect_button.Enabled = True
End Sub
'---------------------------------------------------------------
Public Sub ConnectionList_DblClick()
'---------------------------------------------------------------
    Dim MemberID As String                                  '
(Server)(TCPidx)(RemoteIP)
    Dim Idx As Long                                         ' TCP idx
'---------------------------------------------------------------
    If (ConnectionList.Text = "") Then Exit Sub
    MemberID = ConnectionList.List(ConnectionList.ListIndex) ' Get
The Conversation MemberID String From List Box
    Call GetIdxFromMemberID(TCPSocket, MemberID, Idx)  ' Get TCP idx
From Member ID
    Call RemoveConnectionFromList(TCPSocket(Idx), ConnectionList) '
Clear ListBox Entry(s)...
    Call Disconnect(TCPSocket(Idx))     ' Disconnect Socket Connection
    Unload TCPSocket(Idx)               ' Destroy socket instance
    cmdTalk.Enabled = (ConnectionList.ListCount > 0)    '
Enable/Disable Talk Button...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Sconnect_button.Enabled = (ConnectionList.Text <> "")
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Private Sub Form_Load()
On Error GoTo error1
                    If Winsock1.State <> sckClosed Then
Winsock1.Close
                '-------------------------------------------------
-----------
                    Dim rc As Long
' Return Code Variable
                    Dim Idx As Long
' Current TCP idx variable
                    Dim TCPidx As Long
' Newly created TCP idx value
              '-----------------------------------------------------
---------
                  CLOSINGAPPLICATION = False
' Set status to not closing
                  Call InitServerList(txtServer)
' Get Common Servers List
                  Call Reset_variable
                  cmdTalk.Enabled = False
' Disable Until Connect
                  Sconnect_button.Enabled = (ConnectionList.Text =
"")
                  frmMain.Sound_Receive1 = "C:\My
Documents\Dial.wav"
                  Sound_Receive2 = "C:\My Documents\Busy.wav"
                  Sound_Receive3 = "C:\My Documents\Redial.wav"
                  Text5.FontBold = True
                  Text5.Text = "PCM"
                  Call Set_Comport("Com1", 9600, 1)
                  Exit Sub
error1:
                    If Err = 0 Then Exit Sub
                    MsgBox "Error Number: " & Err & "   " & Error,
vbOKOnly
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Private Sub Form_Unload(Cancel As Integer)
If Winsock1.State <> sckClosed Then
Winsock1.Close
On Error GoTo ErrLabel:
End If
Winsock1.Close
If MSComm1.PortOpen = True Then MSComm1.PortOpen = False
'-------------------------------------------------------------
    Dim Idx As Long
' TCP socket index
    Dim Socket As Winsock                                    ' TCP
socket control
'-------------------------------------------------------------
    CLOSINGAPPLICATION = True          ' Set status flag to closing...
    For Each Socket In TCPSocket       ' For each socket instance
        Call Disconnect(Socket)        ' Close connection/listen
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next                                                        ' Next Cntl
    Set wStream = Nothing
    End                                                         ' End Program
'-------------------------------------------------------------------
ErrLabel:
Status.Caption = Status.Caption = Err.Description
MsgBox "Disconnected", vbExclamation, "error unload"
End Sub
'-------------------------------------------------------------------

Private Sub Menu_exit_Click(Index As Integer)
Call Form_Unload(Index)
End Sub
Private Sub Menu_help_Click(Index As Integer)
frmBrowser.Show
End Sub
Private Sub Menu_ipadress_Click(Index As Integer)
MsgBox "IP Address of This Computer is " & Winsock1.LocalIP,
vbOKOnly, "Voice Chat"
End Sub
Private Sub Menu_properties_Click(Index As Integer)
Properties.Show
End Sub
Private Sub MSComm1_OnComm()
    Select Case MSComm1.CommEvent
                Case comEvReceive
                Dim Check As String
                Dim Dummy As Variant
                Check = MSComm1.Input
                Dummy = Minute(Now)
    Select Case Check
        '.....................................Not success
press.....................................
Case "B"
                Call frmMain.Reset_variable
                frmMain.Status.Caption = "Not Complete"
Case "N"
                If Status.Caption = "Hard ware ready Time Call  1"
Then
                    Call Recall(Phonenum.Caption)
                    Exit Sub
                End If
                If Dummy - Waitingtime > 2 And Count_Send = 9 And
Status.Caption = "Send to Server" Then
                    Status.Caption = "Sucess Call"
                    If Winsock1.State = sckConnected Then
Winsock1.SendData "AUser HangOff"
                    Exit Sub
                End If
                If Count_Send < 9 And Status.Caption = "Receiving
Number" Then
                    Call frmMain.Reset_variable
                    Status.Caption = "Send Not Complete "
                    If Winsock1.State = sckConnected Then
Winsock1.SendData "ACancel Call"
                End If
                If Dummy - Waitingtime < 2 And Count_Send = 9 Then
                    Call frmMain.Reset_variable
                    Status.Caption = "Cancel By user "
                  If Winsock1.State = sckConnected Then
Winsock1.SendData "AUser Cancel Call"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Exit Sub
                    End If
'.....เรียกจากหน้าคอม.....
                  If Status.Caption = "Calling" Or Status.Caption =
"Dialing" Then
                      Call Recall(Phonenum.Caption)
                  Exit Sub
                  End If
                  If Status.Caption = "Hard ware ready Time Call  2"
Or Status.Caption = "Redialing" Then
                      Call Reset_variable
                    If Voice_connect = True Then
                        Call frmMain.Stop_sound
                        Status = sndPlaySoundA(Sound_Receive2,
SND_SYNC Or SND_NODEFAULT)
                        Status.Caption = "The line is Busy"
                        If Winsock1.State = sckConnected Then
Winsock1.SendData "AThe line is busy"
                        wStream.recording = True
                       Call frmMain.Play_sound
                       Exit Sub
                   End If
                     Status = sndPlaySoundA(Sound_Receive2, SND_SYNC
Or SND_NODEFAULT)
                     Status.Caption = "The line is Busy"
                     If Winsock1.State = sckConnected Then
Winsock1.SendData "AThe line is buy"
                     Exit Sub
                  End If
Case "S"
                     Dim Buffer As Variant
                     Check = "Y"
                     Buffer = Check
                     MSComm1.Output = Buffer
            If Auto_Anser.Caption = "Auto" Then
                     Call frmMain.Reset_variable
                     User_Request = True
                      If Winsock1.State <> sckConnected Then
                         Phonenum.Caption = "Reset State Now"
                         Exit Sub
                     End If
                   If Winsock1.State = sckConnected Then
                      Winsock1.SendData "AUser1 Request"
                       If Voice_connect = True Then
                         Call Stop_sound
                         Status = sndPlaySoundA(Sound_Receive1,
SND_SYNC Or SND_NODEFAULT)
                          Status.Caption = "Calling'"
                          wStream.recording = True
                          Call Play_sound
                          Exit Sub
                       End If
                       Status = sndPlaySoundA(Sound_Receive1,
SND_SYNC Or SND_NODEFAULT)
                       Status.Caption = "Calling'"
                  End If
          End If
        Case "Y"
                      If Effortnum = 0 Or Effortnum = 1 Then
                         If Voice_connect = True Then
                            Call frmMain.Stop_sound
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    Status =
sndPlaySoundA(Sound_Receive1, SND_SYNC Or SND_NODEFAULT)
                                    Status.Alignment = 0
                                    Status.FontBold = False
                                    Status.Caption = "Hard ware ready "
& "Time Call" & "  " & Effortnum
                                    wStream.recording = True
                                    Call frmMain.Play_sound
                                    Exit Sub
                                End If
                                    Status =
sndPlaySoundA(frmMain.Sound_Receive1, SND_SYNC Or SND_NODEFAULT)
                                    Status.Alignment = 0
                                    Status.FontBold = False
                                    Status.Caption = "Hard ware ready "
& "Time Call" & "  " & Effortnum
                            End If
        Case 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
                                    If User_Request = True Then
                                     receive = True
                                    Status.Caption = "Receiving
Number"
                                    If Len(Phonenum.Caption) = 9 Then
Status.Caption = "Send to Server"
                                    If Winsock1.State = sckConnected
And Auto_Anser.Caption = "Auto" Then
                                        Winsock1.SendData Check
                                        Count_Send = Count_Send + 1
                                        Waitingtime = Minute(Now)
                                        frmMain.Phonenum.Caption =
frmMain.Phonenum.Caption & Check
                                        If
Len(frmMain.Phonenum.Caption) = 9 Then Exit Sub
                                    End If
                                End If
      End Select
          Case comEventBreak            ' A Break was received.
                    If Err = 0 Then Exit Sub
                    MsgBox "Error Number: " & Err & "  " & Error,
vbOKOnly
          Case comEventFrame                        ' Framing Error
                    If Err = 0 Then Exit Sub
                    MsgBox "Error Number: " & Err & "  " & Error,
vbOKOnly
          Case comEventOverrun                      ' Data Lost.
                    If Err = 0 Then Exit Sub
                    MsgBox "Error Number: " & Err & "  " & Error,
vbOKOnly
          Case comEventRxOver           ' Receive buffer overflow.
                    If Err = 0 Then Exit Sub
                     MsgBox "Error Number: " & Err & "  " & Error,
vbOKOnly
          Case comEventRxParity                     ' Parity Error.
                   If Err = 0 Then Exit Sub
                   MsgBox "Error Number: " & Err & "  " & Error,
vbOKOnly
          Case comEventTxFull           ' Transmit buffer full.
                  If Err = 0 Then Exit Sub
                  MsgBox "Error Number: " & Err & "  " & Error,
vbOKOnly
          Case comEventDCB        ' Unexpected error retrieving DCB]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    If Err = 0 Then Exit Sub
                    MsgBox "Error Number: " & Err & "   " & Error,
vbOKOnly
      End Select
End Sub
Private Sub Option1_GotFocus()
' Activates Audio Recording...
'-------------------------------------------------------------
    Dim rc As Long                      ' Return Code Variable
'-------------------------------------------------------------
    If (Not wStream.Playing And _
        Not wStream.recording And _
            wStream.RecDeviceFree And _
            wStream.PlayDeviceFree) Then  ' Check Audio Device Status
            wStream.recording = True                              '
Set Recording Flag
        cmdTalk.Caption = "&Talking"    ' Update Button Status To
"Talking"
        rc = wStream.RecordWave(Me.hWnd, TCPSocket)      ' Record
voice and send to all connected sockets
        cmdTalk.Caption = "&Talk"       ' Reset Button Status
        If Not wStream.Playing And _
                wStream.PlayDeviceFree And _
              wStream.RecDeviceFree Then     ' Is Audio Device Free?
        End If
    End If
'-------------------------------------------------------------
End Sub
Private Sub N0_Click()
txtServer.Text = txtServer.Text + "0"
End Sub
Private Sub N1_Click()
txtServer.Text = txtServer.Text + "1"
End Sub
Private Sub N2_Click()
txtServer.Text = txtServer.Text + "2"
End Sub
Private Sub N3_Click()
txtServer.Text = txtServer.Text + "3"
End Sub
Private Sub N4_Click()
txtServer.Text = txtServer.Text + "4"
End Sub
Private Sub N5_Click()
txtServer.Text = txtServer.Text + "5"
End Sub
Private Sub N6_Click()
txtServer.Text = txtServer.Text + "6"
End Sub
Private Sub N7_Click()
txtServer.Text = txtServer.Text + "7"
End Sub
Private Sub N8_Click()
txtServer.Text = txtServer.Text + "8"
End Sub
Private Sub N9_Click()
txtServer.Text = txtServer.Text + "9"
End Sub
Private Sub Sconnect_button_Click()
'-------------------------------------------------------------
    Dim rc As Long                      ' Return Code Variable
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Dim Idx As Long                    ' TCP Socket control index
    Dim LocalPort As Long              ' LocalPort Setting
    Dim RemotePort As Long             ' RemotePort Setting
'-------------------------------------------------------------
    If Sconnect_button.Caption <> "Connect" Then
                                       Call ConnectionList_DblClick
                                       Exit Sub
    End If
   Idx = InstanceTCP(TCPSocket)
' Instance TCP Control...
   If (Idx > 0) Then
' Did control instance get created???
                                       Sconnect_button.Enabled = False
' Disable Connect Button
                                       ConnectionList.Enabled = False
' Disable connection list box
                                       On Error Resume Next
                                       If Not Connect(TCPSocket(Idx),
txtServer.Text, VOICEPORT) Then ' Attempt to connect
                                      Unload TCPSocket(Idx)
' Connect failed unload control instance
  End If
  ConnectionList.Enabled = True
' Renable connection list box
  Sconnect_button.Enabled = True
' Enable Connect Button
  End If
  End Sub
'-------------------------------------------------------------
Private Sub TCPSocket_Close(Index As Integer)
' Closing Current TCP Connection...
'-------------------------------------------------------------
    Voice_connect = False
' Status  Voice flag Off
    Call RemoveConnectionFromList(TCPSocket(Index), ConnectionList)
' Remove Connection From List
    Call Disconnect(TCPSocket(Index))
' Close Port Connection...
    cmdTalk.Enabled = (ConnectionList.ListCount > 0)
' Enable/Disable Talk Button...
    Sconnect_button.Enabled = (ConnectionList.Text = "")
    Unload TCPSocket(Index)
' Destroy socket instance
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------

'-------------------------------------------------------------
Private Sub TCPSocket_Connect(Index As Integer)
' TCP Connection Has Been Accepted And Is Open...
'-------------------------------------------------------------
    Voice_connect = True 'Flage Voice Open
    Call AddConnectionToList(TCPSocket(Index), ConnectionList)
' Add New Connection To List
    cmdTalk.Enabled = True              ' Enabled For Connection...
     Sconnect_button.Enabled = (ConnectionList.Text <> "")
'-------------------------------------------------------------
End Sub
'-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'------------------------------------------------------------
Private Sub TCPSocket_ConnectionRequest(Index As Integer, ByVal
requestID As Long)
' Accepting Inbound TCP Connection Request...
'------------------------------------------------------------
    Dim rc As Long
    Dim Idx As Long
    Dim RemHost As String
'------------------------------------------------------------
    If (TCPSocket(Index).RemoteHost <> "") Then
        RemHost = UCase(TCPSocket(Index).RemoteHost)
    Else
        RemHost = UCase(TCPSocket(Index).RemoteHostIP)
    End If
    If Auto_Anser.Caption <> "Auto" Then
        rc = MsgBox("Incomming call from [" & RemHost & "]..." &
vbCrLf & _
                    "Do you wish to answer?", vbYesNo)
' Prompt user to answer...
    Else
        rc = vbYes
    End If
    If (rc = vbYes) Then
        Idx = InstanceTCP(TCPSocket)
' Instance TCP Control...
        If (Idx > 0) Then
' Validate that control instance was created...
            TCPSocket(Idx).LocalPort = 0
' Set local port to 0, in order to get next available port.
            Call TCPSocket(Idx).Accept(requestID)
' Accept connection
            Call AddConnectionToList(TCPSocket(Idx), ConnectionList)
' Add New Connection To List
          cmdTalk.Enabled = True                              '
Enabled For Connection...
            Sconnect_button.Enabled = (ConnectionList.Text <> "")
        End If
    End If
'------------------------------------------------------------
End Sub
'------------------------------------------------------------

Private Sub TCPSocket_Error(Index As Integer, ByVal Number As
Integer, Description As String, ByVal Scode As Long, ByVal Source As
String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
TCPSocket(Index).Close                          ' Close down socket
    Debug.Print "TCPSocket_Error: Number:", Number
    Debug.Print "TCPSocket_Error: Scode:", Hex(Scode)
    Debug.Print "TCPSocket_Error: Source:", Source
    Debug.Print "TCPSocket_Error: HelpFile:", HelpFile
    Debug.Print "TCPSocket_Error: HelpContext:", HelpContext
    Debug.Print "TCPSocket_Error: Description:", Description
    Call DebugSocket(TCPSocket(Index))
End Sub
Private Sub Timer1_Timer()
                        interval.FontBold = True
                        interval.Caption = Format(Date$, "dd/mm/yyyy")
& "     " & Format(Time, "HH:MM:SS")
End Sub
'------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Tools_ButtonClick(ByVal Button As Button)
'-------------------------------------------------------------
    Dim rc As Long                          ' Return Code Variable
    Dim Idx As Long                         ' TCP Socket control index
    Dim LocalPort As Long                   ' LocalPort Setting
    Dim RemotePort As Long                  ' RemotePort Setting
'-------------------------------------------------------------
    Select Case Button.Index
    Case tbCALL
        Idx = InstanceTCP(TCPSocket)                          '
Instance TCP Control...
        If (Idx > 0) Then        'Did control instance get created???
            Button.Enabled = False          'Disable Connect Button
            ConnectionList.Enabled = False
'Disable connection list box
            On Error Resume Next
            If Not Connect(TCPSocket(Idx), txtServer.Text, VOICEPORT)
Then ' Attempt to connect
                Unload TCPSocket(Idx)
' Connect failed unload control instance
            End If
                        ConnectionList.Enabled = True           '
Renable connection list box
            Button.Enabled = True           ' Enable Connect Button
        End If
    Case 6
    Properties.Show
    Case tbHANGUP
        ConnectionList_DblClick
    Case tbAUTOANSWER
        If (Button.Value = tbrPressed) Then
            Button.Image = phoneHungUp
        Else
            Button.Image = phoneAnswered
        End If
    End Select
End Sub
'-------------------------------------------------------------
Private Sub txtServer_KeyPress(KeyAscii As Integer)
'-------------------------------------------------------------
    Dim Conn As Long                        ' Index counter
'-------------------------------------------------------------
    If (KeyAscii = vbKeyReturn) Then   ' If Return Key Was Pressed...
        For Conn = 0 To txtServer.ListCount      ' Search Each Entry
In ListBox
            If (UCase(txtServer.Text) = UCase(txtServer.List(Conn)))
Then Exit Sub
        Next                                ' If Found Then Exit
        txtServer.AddItem UCase(txtServer.Text)  ' Add Server To List
    End If
'-------------------------------------------------------------
End Sub
Private Sub Winsock1_Close()
                    Call Winsock_show("Close")
                    Winsock1.Close
End Sub
Private Sub Winsock1_Connect()
                    Call Winsock_show("Connect")
                    Winsock1.SendData "A" & Winsock1.LocalIP
 End Sub
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        frmMain.Text3.FontBold = True
                        frmMain.Text3.Text = Winsock1.RemoteHostIP
                        Call Winsock_show("Request")
                        If Winsock1.State <> sckClosed Then
Winsock1.Close

                        Winsock1.Accept requestID
End Sub
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
                        Dim strData As String, Header
                        Call Winsock_show("Data")
                        Winsock1.GetData strData
                        Header = Left(strData, 1)
                        Call Decide_Receive(Header, strData)
                        End Sub
Private Sub Winsock1_Error(ByVal Number As Integer, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile
As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
        If Err = 0 Then Exit Sub
           Call Winsock_show("Error")
           Winsock1.Close
           MsgBox "Error Number: " & Err & " " & Error, vbOKOnly
End Sub
Private Sub Winsock1_SendComplete()
                       Call Winsock_show("Send")
                        Status.Caption = "Sending.."
    '.............................Disconnect Voice
Channal.........................
                        If Count_Send = 1 Then
                        Call Stop_sound
                        Exit Sub
                        End If
    '.............................Connec Voice
Channal.........................
                        If Count_Send = 9 Then
                        Status.Caption = "Send Complete"
                         wStream.recording = False
                        Call Delay_Time(0.5)
                        Call Play_sound
                        Exit Sub
                        End If
End Sub
Private Sub cmdOnOffPort_Click()
On Error GoTo ErrH
       If Not ComOnOff Then
            MSComm1.PortOpen = True
            ChkStopLoopUnLess = False
            cmdOnOffPort.Caption = "&OffCom"
            ComOnOff = True              'on port
       Else
            MSComm1.PortOpen = False
            ChkStopLoopUnLess = True
            cmdOnOffPort.Caption = "&OnCom"
            ComOnOff = False             'off port
       End If
      SecRXTX = 0
      If MSComm1.PortOpen = True Then MSComm1.RThreshold = 1
      Call UnLessLoop
       Exit Sub
ErrH:
   MsgBox Err.Description
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub cmdSetProp_Click()
frmProperties.Show
End Sub
Private Sub UnLessLoop()
Dim Buffer
Dim Check As Boolean
MEVENT_DRIVEN:
    Check = True
    Do
        If ChkStopLoopUnLess Then Exit Sub
        DoEvents
        If SecRXTX = 1 Then Check = False
    Loop Until Check = False
      If SecRXTX = 0 Then GoTo MEVENT_DRIVEN      'mode  Event-Driven
End Sub
Private Sub cmd_Close_Click()
    If MsgBox("คุณต้องการจบการทำงาน ใช่หรือไม่ ?", vbYesNo + vbQuestion, "คำยืนยัน") =
vbYes Then
    Unload Me
    Unload frmVBCom
    End If
End Sub
Private iFlow As Integer, iTempEcho As Boolean
Sub LoadPropertySettings()
Dim i As Integer, Settings As String, Offset As Integer
' Load Port Settings
For i = 1 To 16
    cboPort.AddItem "Com" & Trim$(Str$(i))
Next i
' Load Speed Settings
cboSpeed.AddItem "110"
cboSpeed.AddItem "300"
cboSpeed.AddItem "600"
cboSpeed.AddItem "1200"
cboSpeed.AddItem "2400"
cboSpeed.AddItem "4800"
cboSpeed.AddItem "9600"
cboSpeed.AddItem "14400"
cboSpeed.AddItem "19200"
cboSpeed.AddItem "28800"
cboSpeed.AddItem "38400"
cboSpeed.AddItem "56000"
cboSpeed.AddItem "57600"
cboSpeed.AddItem "115200"
cboSpeed.AddItem "128000"
cboSpeed.AddItem "256000"
' Load Data Bit Settings
cboDataBits.AddItem "4"
cboDataBits.AddItem "5"
cboDataBits.AddItem "6"
cboDataBits.AddItem "7"
cboDataBits.AddItem "8"
' Load Parity Settings
cboParity.AddItem "Even"
cboParity.AddItem "Odd"
cboParity.AddItem "None"
cboParity.AddItem "Mark"
cboParity.AddItem "Space"
' Load Stop Bit Settings
cboStopBits.AddItem "1"
cboStopBits.AddItem "1.5"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cboStopBits.AddItem "2"
' Set Default Settings
Settings = frmVBCom.MSComm1.Settings
' In all cases the right most part of Settings will be 1 character
' except when there are 1.5 stop bits.
If InStr(Settings, ".") > 0 Then
    Offset = 2
Else
    Offset = 0
End If
cboSpeed.Text = Left$(Settings, Len(Settings) - 6 - Offset)
Select Case Mid$(Settings, Len(Settings) - 4 - Offset, 1)
Case "e"
    cboParity.ListIndex = 0
Case "m"
    cboParity.ListIndex = 1
Case "n"
    cboParity.ListIndex = 2
Case "o"
    cboParity.ListIndex = 3
Case "s"
    cboParity.ListIndex = 4
End Select
cboDataBits.Text = Mid$(Settings, Len(Settings) - 2 - Offset, 1)
cboStopBits.Text = Right$(Settings, 1 + Offset)
cboPort.ListIndex = frmVBCom.MSComm1.CommPort - 1
optFlow(frmVBCom.MSComm1.Handshaking).Value = True
If Echo Then
    optEcho(1).Value = True
Else
    optEcho(0).Value = True
End If
End Sub
Private Sub cmdCancel_Click()
    Unload Me
End Sub
Private Sub cmdOK_Click()
Dim OldPort As Integer, ReOpen As Boolean
On Error Resume Next
Echo = iTempEcho
OldPort = frmVBCom.MSComm1.CommPort
NewPort = cboPort.ListIndex + 1
If NewPort <> OldPort Then          ' If the port number changes,
close the old port.
    If frmVBCom.MSComm1.PortOpen Then
            frmVBCom.MSComm1.PortOpen = False
            ReOpen = True
    End If
    frmVBCom.MSComm1.CommPort = NewPort    ' Set the new port number.
    If Err = 0 Then
        If ReOpen Then
            frmVBCom.MSComm1.PortOpen = True
        End If
    End If
    If Err Then
        MsgBox Error$, 48
        frmVBCom.MSComm1.CommPort = OldPort
        Exit Sub
    End If
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
frmVBCom.MSComm1.Settings = Trim$(cboSpeed.Text) & "," &
Left$(cboParity.Text, 1) _
    & "," & Trim$(cboDataBits.Text) & "," & Trim$(cboStopBits.Text)
Debug.Print "Settings Com :" & frmVBCom.MSComm1.CommPort & " , " &
frmVBCom.MSComm1.Settings
frmVBCom.lblOncom.Caption = "OnCom:" & frmVBCom.MSComm1.Settings
If Err Then
    MsgBox Error$, 48
    Exit Sub
End If
frmVBCom.MSComm1.Handshaking = iFlow
If Err Then
    MsgBox Error$, 48
    Exit Sub
End If
SaveSetting App.Title, "Properties", "Settings",
frmVBCom.MSComm1.Settings
SaveSetting App.Title, "Properties", "CommPort",
frmVBCom.MSComm1.CommPort
SaveSetting App.Title, "Properties", "Handshaking",
frmVBCom.MSComm1.Handshaking
SaveSetting App.Title, "Properties", "Echo", Echo
Unload Me
End Sub
Private Sub Form_Load()
' Set the form's size
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2
' Load current property settings
LoadPropertySettings
End Sub
Private Sub optEcho_Click(Index As Integer)
If Index = 1 Then
    iTempEcho = True
Else
    iTempEcho = False
End If
End Sub
Private Sub optFlow_Click(Index As Integer)
    iFlow = Index
End Sub
''     การเขียนVisual Basic ติดต่อกับ Serial Port
'-------------------------------------------------------------
Dim SecRXTX As Integer
Dim ModeRXTX As Integer
Dim ComOnOff As Boolean
Dim StatusComm As String
Dim ChkStopLoopUnLess As Boolean
Dim EventMsg As String, ErrorMsg As String
Private Sub Form_Load()
Dim CommPort As String, Handshaking As String, Settings As String
On Error Resume Next
    App.Title = "Visual Basic : ติดต่อทาง Serial Port" 'ชื่อไตเติ้ลโปรแกรม
    'กำหนดค่าปกติของโปรแกรม
    ComOnOff = False              ' ปิดพอร์ต
    optModeRXTX.Item(0).Value = True     'การทำอินเตอร์รัพ
    SecRXTX = 0
    'ตั้งค่าสีของการเทอร์มิเนิล
    txtRXTX.SelLength = Len(txtRXTX)
    txtRXTX.SelText = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    txtRXTX.ForeColor = vbRed
    'ตั้งค่าไบรีจิตเตอร์
    Settings = GetSetting(App.Title, "Properties", "Settings", "")
    If Settings <> "" Then
        MSComm1.Settings = Settings
        If Err Then
            MsgBox Error$, 48
            Exit Sub
        End If
    End If
    CommPort = GetSetting(App.Title, "Properties", "CommPort", "")
    If CommPort <> "" Then MSComm1.CommPort = CommPort
        Handshaking = GetSetting(App.Title, "Properties",
"Handshaking", "")
   If Handshaking <> "" Then
        MSComm1.Handshaking = Handshaking
        If Err Then
            MsgBox Error$, 48
            Exit Sub
        End If
    End If
    Echo = GetSetting(App.Title, "Properties", "Echo", "")
    ChkStopLoopUnLess = False        'ตัวแปรสำหรับออกจากระบบ
On Error GoTo 0   'ถ้าเกิดผิดผลาด
End Sub
Private Sub UnLessLoop()
Dim Buffer
Dim Check As Boolean
MEVENT_DRIVEN:
    Check = True
    Do
        If ChkStopLoopUnLess Then Exit Sub
        DoEvents
        If SecRXTX = 1 Then Check = False
    Loop Until Check = False
MPOLLING:
    Check = True
    Do
        If ChkStopLoopUnLess Then Exit Sub
        DoEvents
        If MSComm1.PortOpen = True Then
            Buffer = MSComm1.Input
        End If
        ShowData txtRXTX, (StrConv((Buffer), vbUnicode))
        If SecRXTX = 0 Then Check = False
    Loop Until Check = False
    If SecRXTX = 0 Then GoTo MEVENT_DRIVEN      'mode  Event-Driven
    If SecRXTX = 1 Then GoTo MPOLLING           'mode Polling
End Sub
Private Sub cmdOnOffPort_Click()
On Error GoTo ErrH
        If Not ComOnOff Then
            MSComm1.PortOpen = True
            ChkStopLoopUnLess = False
            cmdOnOffPort.Caption = "&OffCom"
            ComOnOff = True              'on port
            lblOncom.Caption = "OnCom:" & MSComm1.Settings
            Call ShowMsgProgram(ModeRXTX)
        Else
            MSComm1.PortOpen = False
            ChkStopLoopUnLess = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cmdOnOffPort.Caption = "&OnCom"
            ComOnOff = False          'off port
            lblStatus.Caption = "Status Com : "
            lblOncom.Caption = "OffCom"
        End If
        'Set Interrupt or Polling
       Call optModeRXTX_Click(ModeRXTX)
       'SecRXTX = 0
       'If MSComm1.PortOpen = True Then MSComm1.RThreshold = 1
       Call UnLessLoop
        Exit Sub
ErrH:
    MsgBox Err.Description
End Sub
Private Sub cmdSetProp_Click()
frmProperties.Show
End Sub
Private Sub MSComm1_OnComm()
On Error GoTo ErrH
    Select Case MSComm1.CommEvent
        ' Event messages.
        Case comEvReceive
            Dim Buffer As Variant
            Buffer = MSComm1.Input
            'Buffer = Buffer & MSComm1.Input
            'If Len(Buffer) >= limit Then
            ' Call procedure to Process the received information.
            'process Left$(Buffer, limit)        ' Clean-up buffer
            'Buffer = Right$(Buffer, Len(Buffer) - limit)
           'End If
            ShowData txtRXTX, (StrConv((Buffer), vbUnicode))
            EventMsg$ = " Receive "
        Case comEvSend
            EventMsg$ = " Send "
        Case comEvCTS
            EventMsg$ = "Change in CTS Detected"
        Case comEvDSR
            EventMsg$ = "Change in DSR Detected"
        Case comEvCD
            EventMsg$ = "Change in CD Detected"
        Case comEvRing
            EventMsg$ = "The Phone is Ringing"
        Case comEvEOF
            EventMsg$ = "End of File Detected"
        ' Error messages.
        Case comBreak
            ErrorMsg$ = "Break Received"
        Case comCDTO
            ErrorMsg$ = "Carrier Detect Timeout"
        Case comCTSTO
            ErrorMsg$ = "CTS Timeout"
        Case comDCB
            ErrorMsg$ = "Error retrieving DCB"
        Case comDSRTO
            ErrorMsg$ = "DSR Timeout"
        Case comFrame
            ErrorMsg$ = "Framing Error"
        Case comOverrun
            ErrorMsg$ = "Overrun Error"
        Case comRxOver
            ErrorMsg$ = "Receive Buffer Overflow"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Case comRxParity
                ErrorMsg$ = "Parity Error"
            Case comTxFull
                ErrorMsg$ = "Transmit Buffer Full"
            Case Else
                ErrorMsg$ = "Unknown error or event"
        End Select
        If Len(EventMsg$) Then
            lblStatus.Caption = "Status Com : " & EventMsg$
            Timer1.Enabled = True
        ElseIf Len(ErrorMsg$) Then
            lblStatus.Caption = "Status Com : " & ErrorMsg$
            Beep
            Ret = MsgBox(ErrorMsg$, 1, "Click Cancel to quit, OK to
ignore.")
            If Ret = 2 Then
                MSComm1.PortOpen = False
            End If
            Timer1.Enabled = True
        End If
        Exit Sub
ErrH:
    MsgBox Err.Description
End Sub
Private Sub optModeRXTX_Click(Index As Integer)
    Select Case Index
    Case 0
        SecRXTX = 0
        If MSComm1.PortOpen = True Then MSComm1.RThreshold = 1
        ShowData txtRXTX, "Mode Event-driven" & Chr(13)    'Show msg
    Case 1
        SecRXTX = 1
        If MSComm1.PortOpen = True Then MSComm1.RThreshold = 0
        ShowData txtRXTX, "Mode Polling" & Chr(13)      'Show msg
    End Select
End Sub
Private Sub Timer1_Timer()
    lblTime.Caption = Format(Time(), "HH:MM:SS")
    If EventMsg$ = " Receive " Then
        lblStatus.Caption = "Status Com : "
   End If
End Sub
Private Sub txtRXTX_KeyPress(KeyAscii As Integer)
    If MSComm1.PortOpen Then
        MSComm1.Output = Chr$(KeyAscii)
        If Not Echo Then
            txtRXTX.SelStart = Len(txtRXTX)
            KeyAscii = 0
        End If
    End If
End Sub
Public Static Sub ShowData(Term As Control, Data As String)
    On Error GoTo Handler
    Const MAXTERMSIZE = 16000
    Dim TermSize As Long, i
    TermSize = Len(Term.Text)
    If TermSize > MAXTERMSIZE Then
       Term.Text = Mid$(Term.Text, 4097)
       TermSize = Len(Term.Text)
    End If
    Term.SelStart = TermSize
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Do
       i = InStr(Data, Chr$(8))
       If i Then
          If i = 1 Then
             Term.SelStart = TermSize - 1
             Term.SelLength = 1
             Data = Mid$(Data, i + 1)
          Else
             Data = Left$(Data, i - 2) & Mid$(Data, i + 1)
          End If
       End If
    Loop While i
    Do
       i = InStr(Data, Chr$(10))
       If i Then
          Data = Left$(Data, i - 1) & Mid$(Data, i + 1)
       End If
    Loop While i
    i = 1
    Do
       i = InStr(i, Data, Chr$(13))
       If i Then
          Data = Left$(Data, i) & Chr$(10) & Mid$(Data, i + 1)
          i = i + 1
       End If
    Loop While i
    Term.SelText = Data
    Term.SelStart = Len(Term.Text)
Exit Sub
Handler:
    MsgBox Error$
    Resume Next
End Sub
Private Sub ShowMsgProgram(ByVal MODE As Integer)
Dim strShow As String
    If MODE = 0 Then
            strShow = "Visual Basic: Communication custom control (
MSComm ) " & Chr(13) & _
      "MODE : Event-driven communications" & Chr(13) & _
      "Com Port : " & MSComm1.CommPort & Chr(13) & _
      "Setting :" & MSComm1.Settings & Chr(13)
    Else
            strShow = "Visual Basic: Communication custom control (
MSComm ) " & Chr(13) & _
      "MODE : Polling communications" & Chr(13) & _
      "Com Port : " & MSComm1.CommPort & Chr(13) & _
      "Setting : " & MSComm1.Settings & Chr(13)
    End If
    txtRXTX.Text = ""                'Clear Text
    ShowData txtRXTX, strShow        'Show msg
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วย คำปรึกษา คำแนะนำ และคำช่วยเหลือของอาจารย์ที่ปรึกษาคือ รศ.ดร. สุวิพล สิทธิชีวภาค และ ผศ. เกรียงไกร วงศ์โรจนภรณ์ รวมถึงพี่ปริญญาโทที่คอยให้คำชี้แนะในเรื่องอุปกรณ์ และเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือตลอดมา จึงขอขอบพระคุณมา ณ. โอกาสนี้

นาย สุรชัย เงินสุทธิชัยพร

นาย อนุสรณ์ อังสนานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. กิตติ ภักดีวัฒนะกุล, "Visual Basic 6 ฉบับโปรแกรมเมอร์", บริษัท ดวงกมลสมัย จำกัด, 2543
2. รศ. สมยศ จุณณะปิยะ, "การประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล MCS-51" คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2541
3. สุรพล บุญจันทร์, เอกสารประกอบการเรียนวิชา Telephone Engineering
4. สัจจะ จรัสรุ่งรวีวร, "คู่มือการเขียนโปรแกรม และใช้งาน Visual Basic6", อินโฟเพรส, 2544
5. Noel Jerke, "The Waite Group's Visual Basic 6.0 Client/Server how-to", New Delhi Techmedia, 1999
6. Walter Goralski, "TCP/IP applications and protocols" Charleston, South Carolina : Computer Technology Research, 1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้