

ระบบเสียงและแฟกซ์ผ่านอินเทอร์เน็ต

VOICE AND FAX COMMUNICATION VIA INTERNET SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สพ.

พ.ศ. ๒๕๔๕

๕๕๕

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๔๕

เลขหมู่.....

เลขทะเบียน 50114

วัน,เดือน,ปี 21 เม.ย. 2547

b.....

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีนี้นำไปใช้

ระบบเสียงและแฟกซ์ผ่านอินเทอร์เน็ต

VOICE AND FAX COMMUNICATION VIA INTERNET SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเสียงและแฟกซ์ผ่านอินเทอร์เน็ต

VOICE AND FAX COMMUNICATION VIA INTERNET SYSTEM

ผู้จัดทำ

1. นายพรพต ประวิณานุสรณ์ 43015025

.....
Bo de

(รศ.ดร.กอบชัย เดชหาญ)

อาจารย์ที่ปรึกษา

.....
M. Jit

(อาจารย์ศรวัฒน์ ชิวปรีชา)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเสียงและแฟกซ์ผ่านอินเทอร์เน็ต

VOICE AND FAX COMMUNICATION VIA INTERNET SYSTEM

โดย นายพรพต ประวีณานุสรณ์ 43015025

อาจารย์ที่ปรึกษา รศ.ดร.กอบชัย เดชหาญ

อาจารย์ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

โครงการนี้นำเสนอเกี่ยวกับระบบเสียงและแฟกซ์ผ่านอินเทอร์เน็ต ซึ่งระบบนี้จะใช้อินเทอร์เน็ต เชื่อมต่อไปยังเครือข่าย โดยการเขียนโปรแกรมขึ้นมาใช้งาน โดยที่โปรแกรมนี้จะพัฒนาด้วยภาษาวิซวล เบสิก ซึ่งจะทำงานภายใต้ระบบปฏิบัติการ ไมโครซอฟวินโดวส์ โปรแกรมที่เขียนขึ้นมานี้มีความสามารถในการติดต่อสื่อสารด้วยเสียงและแฟกซ์

ABSTRACT

This project concerns about voice and fax system via Internet to link with the other network. The program is written with Visual Basic and operates on Microsoft Windows. This written program is able to communicate with voice and fax.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ทฤษฎีระบบเครือข่าย และ วิน โควสส์ซ็อกเก็ต	3
2.1.1 ระบบเครือข่าย	3
2.1.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์และ อินเทอร์เน็ตเวิร์คกิ้ง	3
2.1.1.2 สถาปัตยกรรมของอินเทอร์เน็ต	3
2.1.1.3 ข้อกำหนดรูปแบบของเกตเวย์	4
2.1.2 โมเดล โอเอสไอ	5
2.1.3 ทีซีพี/ไอพี	7
2.1.3.1 ทีซีพี/ไอพี คืออะไร	8
2.1.3.2 ส่วนประกอบของ ทีซีพี/ไอพี	8
2.1.3.3 โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี	10
2.1.3.4 ข้อแตกต่างระหว่างชุดโพรโตคอล TCP/IP	11
2.1.3.5 ลักษณะของการติดต่อ	11
2.1.3.6 ความสัมพันธ์ของ TCP และ UDP กับอินเทอร์เน็ต	12
2.1.3.7 พอร์ตและซ็อกเก็ต	12
2.1.3.8 ข้อกำหนดรูปแบบ IP	13
2.1.3.9 ข้อกำหนดรูปแบบ ICMP	14
2.1.4 ไอพีแอดเดรส	14
2.1.5 การติดต่ออินเทอร์เน็ตของวิซวลเบสิกโดยใช้ MS Winsock Control	16
2.2 การบีบอัดเสียง	20
2.3 MCI	22
2.3.1 สถาปัตยกรรมของ MCI	22
2.3.2 การเขียนโปรแกรมอินเทอร์เฟส	22
2.3.2.1 อินเทอร์เฟส Command-String	23
2.3.2.2 อินเทอร์เฟส Command-Message	23
2.3.3 ชุดคำสั่ง MCI	23
2.3.4 เกี่ยวกับ MCI ดีไวซ์	23
2.3.5 มาตรฐานชนิด MCI ดีไวซ์	24
2.3.6 ซ็อกดีไวซ์	24

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7 การเปิด MCI ดีไวซ์	25
2.3.8 การส่งคำสั่งข้อความคำสั่ง	25
2.3.9 การระบุ MCI ดีไวซ์	26
2.3.10 ข้อความคำสั่งของ MCI ที่สำคัญ	26
2.4 พื้นฐานโทรศัพท์	28
2.4.1 ระบบสัญญาณต่างๆ ทางโทรศัพท์	28
2.4.2 ความถี่สัญญาณต่างๆ ที่เป็นมาตรฐานขององค์การโทรศัพท์	29
2.4.3 เครื่องรับโทรศัพท์	31
2.4.4 ชุมสายโทรศัพท์	37
2.5 DTMF DECODER MT8870	38
2.6 โทรสาร	44
2.6.1 หลักการทำงานของโทรสาร	45
2.6.2 การแบ่งกลุ่มของโทรสาร	46
2.6.3 คุณสมบัติทั่วไปของเครื่องโทรสารกลุ่ม 3	47
2.6.4 มาตรฐานและคุณสมบัติของเครื่องโทรสารกลุ่ม 3	48
2.6.5 รายละเอียดของเครื่องโทรสารกลุ่ม 3	48
2.7 การเข้ารหัสข้อมูล	49
2.8 ขบวนการรับส่งข้อมูลของเครื่องโทรสารกลุ่มที่ 3	50
2.8.1 ขั้นตอนการส่งโทรสาร	52
2.9 รูปแบบของ HDLC เฟรม	58
2.10 แฟกซ์โมเด็ม	59
2.11 การแบ่งระดับโมเด็มโดย EIA	60
2.12 จดหมายอิเล็กทรอนิกส์	62
2.12.1 ส่วนประกอบของระบบจดหมายอิเล็กทรอนิกส์	62
2.13 Pop3	64
บทที่ 3 การออกแบบและการสร้าง	65
3.1 ส่วนฮาร์ดแวร์	65
3.2 การออกแบบและการสร้างส่วนซอฟต์แวร์	70
3.2.1 หลักการออกแบบในส่วนของการรับส่งข้อมูลเสียง	70
3.2.2 ขั้นตอนการพัฒนา	70
3.2.2.1 การพัฒนาในส่วนของการรับส่งข้อมูลผ่านระบบเน็ตเวิร์ก	70
3.2.2.2 การพัฒนาในส่วนของการจัดการเรื่องเสียงบน Windows	70
3.2.2.3 การพัฒนาในส่วนของการจัดการเรื่อง Compression เสียงบน Windows	73
3.3 โครงสร้างโดยรวมของโปรแกรมในส่วนของการรับส่งข้อมูลเสียง	73

3.4 การออกแบบส่วนติดต่อกับผู้ใช้และการใช้งานโปรแกรม	75
3.4.1 การออกแบบส่วนติดต่อกับผู้ใช้และการใช้งานโปรแกรมส่วนของการ ส่งข้อมูลเสียง	75
3.4.2 การออกแบบส่วนติดต่อกับผู้ใช้ และการใช้งานโปรแกรมส่วน ของการส่งแฟกซ์	78
บทที่ 4 การทดลองและผลการทดลอง	79
4.1 การทดลองและผลการทดลองในส่วนของฮาร์ดแวร์	79
4.2 การทดลองและผลการทดลองในส่วนของซอฟต์แวร์	80
บทที่ 5 บทสรุปและวิจารณ์	93
ภาคผนวก	
บรรณานุกรม	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 แสดงโครงสร้างของส่วนการสื่อสารข้อมูลเสียง	1
รูปที่ 1.2 แสดงโครงสร้างของส่วนแพคเกจจิ้ง	2
รูปที่ 2.1 แสดงเครือข่ายอินเทอร์เน็ต	4
รูปที่ 2.2 แสดงคลาสของไอพีแอดเดรส	15
รูปที่ 2.3 แสดงไฟกระแสดตรงเลี้ยงคู่สาย	29
รูปที่ 2.4 แสดงสัญญาณพื้นฐานของโทรศัพท์	30
รูปที่ 2.5 บล็อกไดอะแกรมการทำงานของเครื่องโทรศัพท์	32
รูปที่ 2.6 แสดงวงจรหมุนหมายเลขแบบพัลส์อย่างง่าย	33
รูปที่ 2.7 แสดงไดอะแกรมของเวลาค่ำๆ ของสัญญาณที่เกิดจากการหมุนหมายเลข 4	33
รูปที่ 2.8 ระบบโทรศัพท์แบบ DTMF	36
รูปที่ 2.9 บล็อกไดอะแกรมการทำงานพื้นฐานของชุมสายโทรศัพท์	37
รูปที่ 2.10 แสดงโลจิกลูป	37
รูปที่ 2.11 แสดงส่วนประกอบของไฟกระแสดตรงและกระแสสลับ	38
รูปที่ 2.12 แสดงรายละเอียดขาของ MT8870	39
รูปที่ 2.13 แสดงโครงสร้างภายในของ MT8870	39
รูปที่ 2.14 แสดงความถี่ที่ได้จากภาคกรองความถี่	40
รูปที่ 2.15 แสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาคาร์ดใหม่ พร้อมวิธีคำนวณ	42
รูปที่ 2.16 แสดงการต่อวงจรภาคอินพุท	42
รูปที่ 2.17 แสดงการต่อวงจรผลิตความถี่	42
รูปที่ 2.18 แสดงวงจรการใช้งานเบื้องต้นของ MT8870	43
รูปที่ 2.19 แสดงแผนภูมิเวลา ของ MT8870	43
รูปที่ 2.20 หลักการทำงานของโทรสารด้านส่ง	46
รูปที่ 2.21 หลักการทำงานของโทรสารด้านรับ	46
รูปที่ 2.22 โครงสร้างของเครื่องโทรสารกลุ่ม 3	49
รูปที่ 2.23 แสดงตัวอย่างการเข้ารหัสข้อมูลโทรสาร	50
รูปที่ 2.24 แสดงขบวนการในการรับส่งข้อมูลของเครื่องโทรสารกลุ่มที่ 3	51
รูปที่ 2.25 รูปแบบสัญญาณซึ่งแสดงคุณสมบัติของเครื่องโทรสารที่ส่งโดยเครื่องโทรสารด้านรับไปยังเครื่องโทรสารด้านผู้เรียก	52
รูปที่ 2.26 แสดงขั้นตอนการรับส่งข้อมูลระหว่างเครื่องโทรสาร จำนวน 1 หน้า	54
รูปที่ 2.27 แสดงขั้นตอนการรับส่งข้อมูลระหว่างเครื่องโทรสารซึ่งมีจำนวนมากกว่า 1 หน้า	55

รูปที่ 2.28 แสดงรูปแบบของสัญญาณ HDLC ที่ใช้ติดต่อระหว่างเครื่องโทรสาร	58
รูปที่ 2.29 แสดงการแบ่งระดับโมเด็มโดย EIA	61
รูปที่ 3.1 แสดงบล็อกไดอะแกรมคร่าวๆ ของวงจร	65
รูปที่ 3.2 แสดงการต่อระหว่างคอมพิวเตอร์ กับอุปกรณ์	66
รูปที่ 3.3 แสดงคีย์โทรศัพท์ทั้งหมด	66
รูปที่ 3.4 การต่อวงจร MT8870	67
รูปที่ 3.5 วงจรถอดรหัสสัญญาณ BCD 1101	67
รูปที่ 3.6 วงจรควบคุมรีเลย์	68
รูปที่ 3.7 วงจรทำสัญญาณค้าง	68
รูปที่ 3.8 แสดงวงจรรวม	69
รูปที่ 3.9 แสดงผังการทำงานของโปรแกรมในการส่งเสียงผ่านระบบอินเตอร์เน็ต	71
รูปที่ 3.10 แสดงผังการทำงานในส่วนการรับเสียงผ่านระบบอินเตอร์เน็ต	72
รูปที่ 3.11 แสดงผังการทำงานของโปรแกรมในส่วนของแฟกซ์	74
รูปที่ 3.12 แสดงโปรแกรม Client Send	75
รูปที่ 3.13 แสดงโปรแกรม Client Dial	76
รูปที่ 3.14 แสดงโปรแกรม Client Receive	76
รูปที่ 3.15 แสดงโปรแกรม Server Receive	77
รูปที่ 3.16 แสดงโปรแกรม Server Send	77
รูปที่ 3.17 แสดงโปรแกรม Server Dial	78
รูปที่ 3.18 แสดงโปรแกรมในการรับข้อมูลจากจดหมายอิเล็กทรอนิกส์เพื่อทำการส่งแฟกซ์	78
รูปที่ 4.1 แสดงวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ	79
รูปที่ 4.2 แสดงวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ A	80
รูปที่ 4.3 แสดงการวางจรั้งสัญญาณ	80
รูปที่ 4.4 แสดงผลที่ได้จากการพิมพ์ข้อความโต้ตอบกันที่ฝั่งไคลเอ็นท์	81
รูปที่ 4.5 แสดงผลที่ได้จากการพิมพ์ข้อความโต้ตอบกันที่ฝั่งเซิร์ฟเวอร์	81
รูปที่ 4.6 แสดงสัญญาณเสียงที่วัดได้จากการส่งสัญญาณเสียงจากฝั่งไคลเอ็นท์มาฝั่งเซิร์ฟเวอร์	82
รูปที่ 4.7 แสดงสัญญาณเสียงที่วัดได้จากการส่งสัญญาณเสียงจากฝั่งเซิร์ฟเวอร์มาฝั่งไคลเอ็นท์	82
รูปที่ 4.8 แสดงคุณภาพของสัญญาณที่ Speaker Output เมื่อป้อนสัญญาณความถี่ 2 kHz ที่ Microphone Input	82
รูปที่ 4.9 แสดงหน้าจอ โปรแกรม Client Dial ขณะทำการกดหมายเลข	83
รูปที่ 4.10 แสดงหน้าจอโปรแกรม Server Dial ขณะรับหมายเลข	83
รูปที่ 4.11 แสดงสัญญาณการกดคีย์หมายเลข 1	84

รูปที่ 4.12 แสดงสัญญาณการกดคีย์หมายเลข 2	84
รูปที่ 4.13 แสดงสัญญาณการกดคีย์หมายเลข 3	84
รูปที่ 4.14 แสดงสัญญาณการกดคีย์หมายเลข 4	85
รูปที่ 4.15 แสดงสัญญาณการกดคีย์หมายเลข 5	85
รูปที่ 4.16 แสดงสัญญาณการกดคีย์หมายเลข 6	85
รูปที่ 4.17 แสดงสัญญาณการกดคีย์หมายเลข 7	86
รูปที่ 4.18 แสดงสัญญาณการกดคีย์หมายเลข 8	86
รูปที่ 4.19 แสดงสัญญาณการกดคีย์หมายเลข 9	86
รูปที่ 4.20 แสดงสัญญาณการกดคีย์หมายเลข 0	87
รูปที่ 4.21 แสดงสัญญาณการกดคีย์ #	87
รูปที่ 4.22 แสดงสัญญาณการกดคีย์ *	87
รูปที่ 4.23 แสดงสัญญาณการกดคีย์ Dail	88
รูปที่ 4.24 แสดงสัญญาณเสียงที่ได้จากสปีคเกอร์ของเครื่องคอมพิวเตอร์ ที่เป็นเครื่องโคลนอื่นที่	88
รูปที่ 4.25 แสดงสัญญาณเสียงที่ได้จากหูฟังโทรศัพท์	88
รูปที่ 4.26 แสดงสัญญาณการกดคีย์ Hang Up	89
รูปที่ 4.27 แสดงหน้าจอของโปรแกรมที่ใช้ในการส่งแฟกซ์	90
รูปที่ 4.28 แสดงการทำงานของโปรแกรม เมื่อทำการติดต่อกับเมลล์เซิร์ฟเวอร์และ เริ่มรับข้อมูลจากจดหมายอิเล็กทรอนิกส์	90
รูปที่ 4.29 แสดงการทำงานของโปรแกรมเมื่อทำงานเสร็จ	91
รูปที่ 4.30 แสดงหน้าจอแสดงเอกสารที่รับได้จากเมลล์เซิร์ฟเวอร์	91
รูปที่ 4.31 แสดงตัวอย่างเอกสารที่ได้จากเครื่องโทรสารปลายทาง	92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 โอ เอส ไอโมเดล	6
ตารางที่ 2.2 โครงสร้างของชุด โพรโทคอล ทีซีพี/ไอพี เปรียบเทียบกับ แบบจำลอง โอ เอส ไอ	10
ตารางที่ 2.3 หมายเลขพอร์ตที่ให้บริการต่างๆ	13
ตารางที่ 2.4 คุณสมบัติวินซ็อกคอนโทรล	18
ตารางที่ 2.5 คุณลักษณะสถานะ	18
ตารางที่ 2.6 Method ของวินซ็อกคอนโทรล	19
ตารางที่ 2.7 Event ของวินซ็อก	20
ตารางที่ 2.8 แสดงค่าความถี่ต่างๆ ที่ใช้ในระบบ DTMF	35
ตารางที่ 2.9 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ	41
ตารางที่ 2.10 แสดงคุณสมบัติของเครื่องโทรสารกลุ่ม 3 ตามมาตรฐาน ITU-T	48
ตารางที่ 2.11 แสดงขั้นตอนการรับส่งข้อมูลเมื่อเครื่องโทรสารต้นทางเป็นผู้ส่งเอกสารให้ปลายทาง	56
ตารางที่ 2.12 แสดงขั้นตอนการรับส่งข้อมูลเมื่อเครื่องโทรสารปลายทางเป็นผู้ส่งเอกสาร	57
ตารางที่ 2.13 มาตรฐานของโมเด็มที่ใช้ในเครื่องโทรสารกลุ่มที่ 3 ตามข้อกำหนดของ ITU-T	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

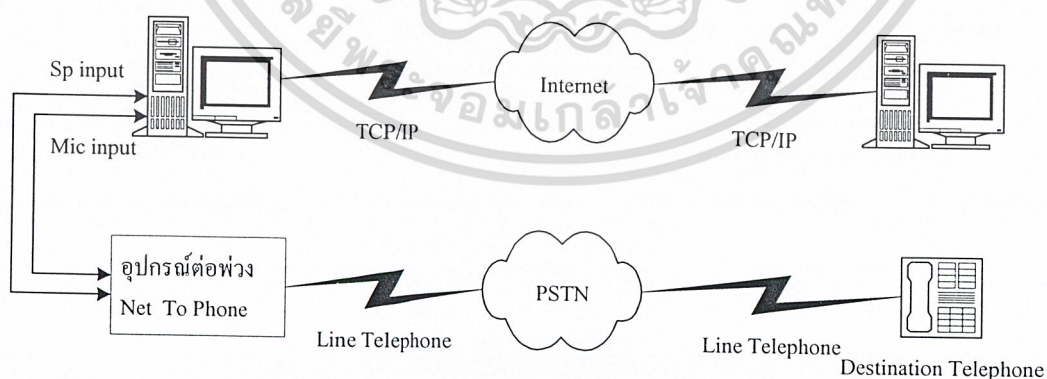
ในปัจจุบันเทคโนโลยีทางการสื่อสารมีการพัฒนาไปอย่างรวดเร็วมากเครือข่ายอินเทอร์เน็ต (Internet) เป็นเทคโนโลยีหนึ่งที่น่าจับตามอง เนื่องจากมีการเจริญเติบโตอย่างรวดเร็ว โดยที่มีความสามารถในการสื่อสาร แลกเปลี่ยนข้อมูล ข่าวสารที่รวดเร็วและฉับไว ซึ่งรูปแบบที่ใช้ในการสื่อสารสามารถใช้ได้หลายรูปแบบ ไม่ว่าจะเป็นข้อมูลตัวอักษร ภาพ เสียง ฯลฯ และยังมีเครือข่ายครอบคลุมทั่วโลก และจะเห็นว่าในปัจจุบันค่าใช้จ่ายในการเชื่อมต่ออินเทอร์เน็ตมีราคาที่ถูกลง

จากข้อดีต่าง ๆ ของอินเทอร์เน็ตจะเห็นว่าถ้าหากเราต้องการพูดคุยกับบุคคลที่อยู่ห่างไกลจากเรา เราสามารถใช้เครือข่ายอินเทอร์เน็ตเป็นตัวกลางในการสื่อสารได้ เพราะถ้าหากเราใช้เครือข่ายโทรศัพท์ในการสื่อสารนั้นจะเห็นว่าค่าใช้จ่ายจะสูงมาก และถ้าหากเราสามารถติดต่อไปยังเครื่องโทรศัพท์ที่อยู่นอกพื้นที่ให้บริการได้ก็คงดี

เครื่องโทรสารเป็นอุปกรณ์สื่อสารที่ใช้เชื่อมต่อกับเครือข่ายโทรศัพท์ ซึ่งมีใช้งานกันอย่างแพร่หลาย ซึ่งข้อดีของโทรสารก็คือ ข่าวสารที่มาถึงสามารถพิมพ์ออกมาเป็นเอกสารได้ทันที

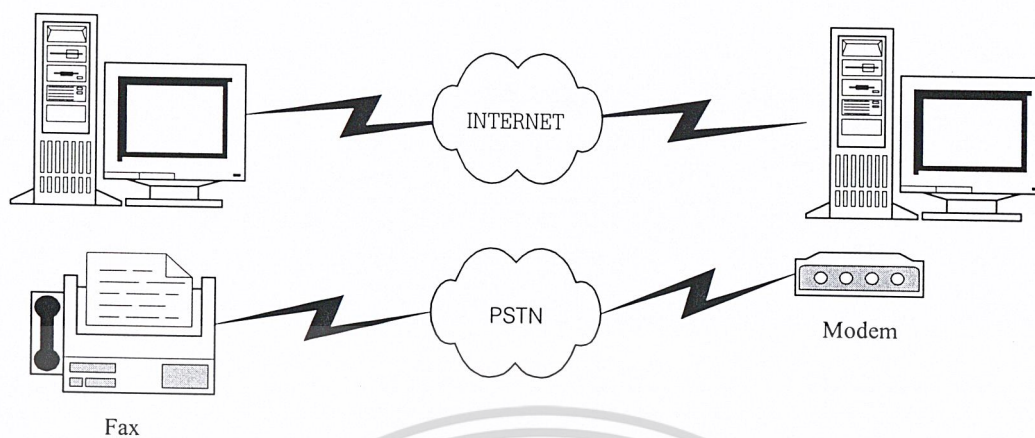
จะเห็นได้ว่าถ้าหากเราทำการเชื่อมต่อระหว่างเครือข่ายอินเทอร์เน็ตกับเครือข่ายโทรศัพท์ได้จะเกิดประโยชน์อย่างมากในการประหยัดค่าใช้จ่ายที่จะต้องเสียไป

โครงการนี้จะต่ออยู่ระหว่างเครือข่ายอินเทอร์เน็ตกับเครือข่ายโทรศัพท์ ทำให้สามารถติดต่อสื่อสารผ่านอินเทอร์เน็ตและเชื่อมต่อเข้ากับเครือข่ายโทรศัพท์ โดยในโครงการนี้ทำการติดต่อสื่อสารกันระหว่างเครือข่ายโดยข้อมูลเสียง และทำการรับจดหมายอิเล็กทรอนิกส์จากเมลเซิร์ฟเวอร์มาทำการส่งไปยังเครื่องโทรสารปลายทาง



รูปที่ 1.1 แสดงโครงสร้างของส่วนการสื่อสารข้อมูลเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 แสดงโครงสร้างของส่วนแฟกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

ส่วนซอฟต์แวร์ (Software part)

2.1 ทฤษฎีระบบเครือข่าย และ วินโดวส์ซ็อกเก็ต

2.1.1 ระบบเครือข่าย (Network System)

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควร ในหัวข้อนี้จะอธิบาย หลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนา โปรแกรมประยุกต์บนระบบเครือข่ายซึ่งจะกล่าวดังต่อไปนี้

2.1.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์และอินเทอร์เน็ตเวิร์กคิง (Internet working)

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End System) ซึ่งระบบปลายทางเป็นระบบที่อิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นไปได้ตั้งแต่ ไมโครคอมพิวเตอร์ (Micro computer) ไปจนถึงซูเปอร์คอมพิวเตอร์ (Super computer) ขนาดใหญ่ เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งข้อมูลในการแบ่งปันทรัพยากรของระบบ เช่น ไฟล์ข้อมูล, เครื่องพิมพ์ (Printer) โมเด็ม (Modem) ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing Database)

อินเทอร์เน็ตเวิร์คคิง หรือ อินเทอร์เน็ต (Internet) คือ การเชื่อมต่อระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้น คอมพิวเตอร์บนระบบ เครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่ายอื่นๆ ได้

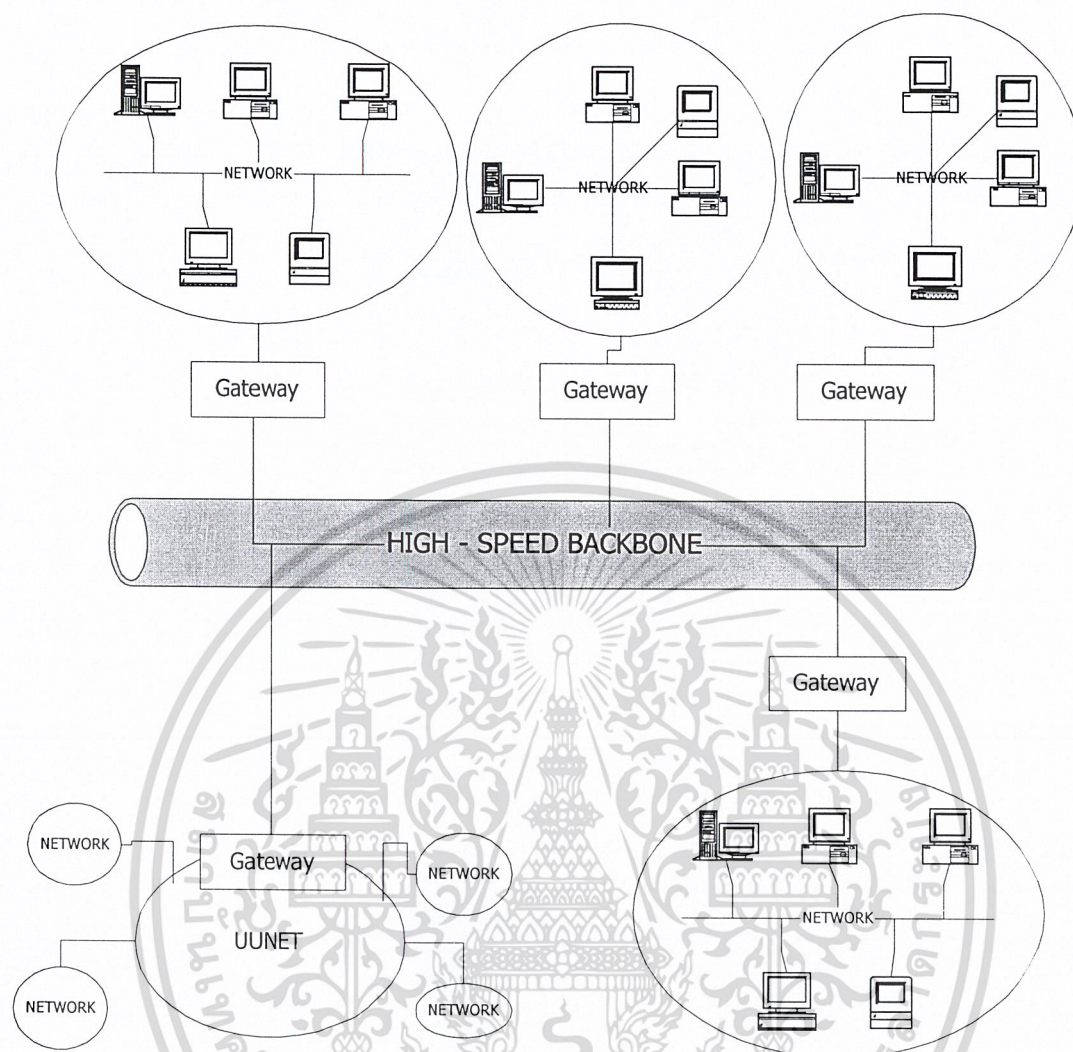
2.1.1.2 สถาปัตยกรรมของอินเทอร์เน็ต

คำกล่าวถึงอินเทอร์เน็ตว่าเป็นเครือข่ายที่ใหญ่ที่สุดในโลก เครือข่ายที่มีการต่อเสียบนกับใยแมงมุมครอบคลุมโลก อินเทอร์เน็ตเชื่อมโยงโลกเข้าด้วยกันอย่างไร้มิติ จากคำกล่าวเหล่านี้ทำให้เราควรรู้ว่า อินเทอร์เน็ตจัดการส่วนประกอบต่าง ๆ อย่างไร จึงทำให้แต่ละส่วนสามารถรับส่งข้อมูลและทำงานสัมพันธ์กันได้อย่างดี เราจึงจำเป็นต้องศึกษาสถาปัตยกรรมของอินเทอร์เน็ตดังต่อไปนี้

อินเทอร์เน็ตประกอบด้วยสายสื่อสารความเร็วสูงเป็นแบคโบน (Back bone) เป็นสายโทรศัพท์ตาม มาตรฐาน TS สามารถวิ่งด้วยความเร็ว 44.736 เมกะบิตต่อวินาที

เครือข่ายที่ต้องการเชื่อมโยงโดยตรงกับอินเทอร์เน็ตจะต่อกับแบคโบน ด้วยอุปกรณ์ที่เรียกว่า "เกตเวย์" (Gateway) ซึ่งเป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่ควบคุมการ ผ่านเข้า-ออกของข่าวสารระหว่างเครือข่ายกับแบคโบน เกตเวย์ทุกตัวสามารถกำหนดการติดต่อกับเกตเวย์ตัวอื่น หรือเครือข่ายอื่น ได้โดยใช้ ไอพีแอดเดรส (IP Address) ของเครือข่าย อ้างอิงถึงกัน เช่น อินเทอร์เน็ต ประเทศไทยเป็นเกตเวย์ของเครือข่ายในกลุ่มติดต่อกับ เกตเวย์ของ UUNET ที่รัฐเวอร์จิเนีย สหรัฐอเมริกา

ข่าวสารจากเครือข่ายถูกส่งออกไปผ่านเกตเวย์เข้าสู่อินเทอร์เน็ต โดยที่เกตเวย์เป็นตัวเลือกที่ทางการเดินทางเพื่อไปยังปลายทางที่ต้องการ แต่ตามเส้นทางอาจจะต้องผ่านเกตเวย์อีกหลายตัว เพื่อรับช่วงส่งข่าวสารจนถึงที่หมาย ถึงแม้ว่าจะต้องเดินทางระยะไกลก็ตาม แต่ด้วยสายสื่อสารความเร็วสูงทำให้การส่งข่าวสารทำได้อย่างรวดเร็ว



รูปที่ 2.1 แสดงเครือข่ายอินเทอร์เน็ต

2.1.1.3 ข้อกำหนดรูปแบบของเกตเวย์ (Gateway Protocols)

เกตเวย์ต้องมีข้อมูลของเกตเวย์ตัวอื่นและรู้จักเครือข่ายปลายทาง เพื่อนำมาใช้ในการกำหนดเส้นทางที่ข่าวสารสามารถเดินทางไปถึงได้เร็วที่สุด เกตเวย์จะมีการแลกเปลี่ยนข้อมูลระหว่างกันเกี่ยวกับเส้นทาง รายละเอียด สถานะเครือข่ายและคุณสมบัติเครือข่ายย่อยที่ติดต่อเข้าสู่เครือข่ายใหญ่ตามลำดับชั้นจึงต้องมีการ กำหนด รูปแบบพิเศษสำหรับเกตเวย์ขึ้นมา

ข้อกำหนดรูปแบบเกตเวย์แบ่งออกตามการใช้งานได้ดังนี้

1. IGP (Interior Gateway Protocol)_ ถูกนำใช้กับเกตเวย์ที่อยู่ในเครือข่ายลูกติดต่อกับเครือข่ายลูกที่อยู่ภายในเครือข่ายแม่เดียวกัน หรือติดต่อกันระหว่างเครือข่ายแม่กับเครือข่ายลูก การเชื่อมโยงเกตเวย์ประเภทนี้มักจะไม่ค่อยมีการเปลี่ยนแปลง จึงเรียกว่าระบบอิสระจากกัน (Autonomous หรือ Selfcomplete)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. EGP (External Gateway Protocol) ในเครือข่ายใหญ่ๆ การติดต่อกับเครือข่ายอื่นจะมีการเปลี่ยนแปลงอยู่เสมอ ตามการเปลี่ยนแปลง ของเครือข่ายย่อยที่เชื่อมโยงอยู่เป็นจำนวนมากจึงมีข้อกำหนดรูปแบบที่ใช้กับการ สื่อสารระหว่างเกตเวย์ของเครือข่ายด้วย EGP

3. GGP (Gateway-to-Gateway Protocol) การเดินทางของข่าวสารระยะไกลบนแบคโบน อาจจะต้องผ่าน เกตเวย์หลายตัวกว่าจะถึงปลายทาง GGP เป็นข้อกำหนดรูปแบบของการสื่อสารระหว่างเกตเวย์บนแบคโบน เพื่อให้การจราจรบนแบคโบน ไม่ติดขัด ข่าวสารเคลื่อนที่ไปได้รวดเร็ว

2.1.2 โมเดลโอเอสไอ (OSI Model)

เพื่อลดปัญหาในความยุ่งยากสับสนในการติดต่อสื่อสารข้อมูล โครงสร้างของการสื่อสารข้อมูล ภายใน อุปกรณ์คอมพิวเตอร์ส่วนใหญ่จะถูกแบ่งเป็นชั้นๆ โดยแต่ละชั้นมีอิสระไม่ขึ้นต่อกันทำให้การเปลี่ยนแปลงบริการชั้นใดๆ ไม่ก่อปัญหาต่อบริการ ชั้นอื่นการเพิ่มเติมการบริการใหม่ทำได้โดยไม่ต้องเปลี่ยนแปลงโปรแกรมระบบเดิมและสิ่งที่สำคัญ ก็คือ การทำระดับชั้นนั้นทำให้ตัวโปรแกรมมีขนาดเล็กสามารถระบุ ส่วนที่จะต้องปรับปรุงได้แน่นอน ไม่ต้องวิตกกังวลถึงโปรแกรมส่วนอื่น ทำให้การพัฒนาประสิทธิภาพของระบบทำได้ง่าย และดียิ่งขึ้น

ISO (International Standardization for Organization) ซึ่งเป็นองค์การที่จัดขึ้นมาเพื่อดูแลและส่งเสริมตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ ได้พัฒนาสถาปัตยกรรม ข้อกำหนดรูปแบบ สำหรับ เครือข่าย ซึ่งเป็นลักษณะระบบเปิดที่เรียกว่า "Open System Interconnection Model " (OSI) โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรมแบบระบบเปิด (Open System) เพราะมุ่งหมายที่จะให้ระบบคอมพิวเตอร์ในหลายๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโปรโตคอล(Protocol) ในการสื่อสารออกเป็น 7 เลเยอร์ (Layer) ซึ่งโปรโตคอล คือ ชุดหรือข้อตกลงในการติดต่อ ข้อสังเกต โมเดล OSI เป็นเพียงข้อเสนอแนะ มิใช่ข้อกำหนด และควรรู้อย่างไรที่ไม่มีระบบการเชื่อมต่อที่สร้างเหมือนกับโมเดล OSI จริง

ข้อกำหนดรูปแบบแต่ละระดับชั้นของ โอ เอส ไอ มีการชุดหน้าที่ต้องปฏิบัติแน่นอน โดยจะต้องเป็นตาม กลุ่มการบริการดังนี้

1. ระดับชั้น 7 แอปพลิเคชัน (Application Layer) เป็นเลเยอร์ชั้นบนสุดของรูปแบบ OSI ซึ่งเป็นชั้นที่ติดต่อระหว่างผู้ใช้โดยตรงระหว่างผู้ใช้ โดยตรงซึ่งได้แก่โฮสต์คอมพิวเตอร์เทอร์มินัลหรือคอมพิวเตอร์ส่วนบุคคล (PC) เป็นต้น แอปพลิเคชันใน เลเยอร์นี้ สามารถนำเข้าหรือออกจากระบบเครือข่ายได้โดยไม่ต้องสนใจจะมีขั้นตอนการทำงานอย่างไร เพราะจะมีชั้นพีรีเซนเตชันเป็นผู้รับผิดชอบอยู่แล้ว ในรูปแบบ OSI นี้ แอปพลิเคชันจะการติดต่อกับชั้นพีรีเซนเตชันโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับชั้น	ชื่อระดับชั้น	ระดับหน้าที่
7	Application	Upper Layers
6	Presentation	
5	Session	
4	Transport	Lower Layers
3	Network	
2	Data Link	
1	Physical	

ตารางที่ 2.1 โอ เอส ไอ โมเดล

2. ระดับชั้น 6 ชั้นพรีเซนเตชันเลเยอร์ (Presentation Layer) คอยรวบรวมข้อความและแปลงรหัสหรือแปลงในแบบของข้อมูลที่ได้รับมาจากระดับชั้น 7 เพื่อให้เป็นไปตามแบบการสื่อสารเดียวกัน เพื่อช่วยลดปัญหาต่างๆที่อาจจะเกิดขึ้นกับผู้ใช้งานในระบบ

3. ระดับชั้น 5 เซสชันเลเยอร์ (Session Layer) ทำหน้าที่เชื่อมโยงระหว่างผู้ใช้คอมพิวเตอร์เครื่องอื่นๆ โดยผู้ใช้จะใช้คำสั่ง หรือข้อความที่กำหนดไว้ป้อนเข้าไปในระบบ ในการสร้างการเชื่อมโยงนี้ผู้ใช้จะต้องกำหนดรหัสตำแหน่งของ จุดหมายปลายทางที่ต้องการติดต่อด้วยในบางครั้งเครือข่ายอาจมีชั้นเซสชันและชั้นทรานสปอร์ตรวม เป็นชั้นเลเยอร์เดียวกัน

4. ระดับชั้น 4 ทรานสปอร์ตเลเยอร์ (Transport Layer) บางครั้งเรียกว่าชั้น เครื่องต่อเครื่อง (Host to Host) และจากเลเยอร์ชั้นที่ 4 ถึง ชั้นที่ 7 นี้เรียกรวมกันว่าเลเยอร์เอนทูเอน (End-to-End) ในเลเยอร์ชั้นทรานสปอร์ต (Transport) นี้จะเป็นการสื่อสารระหว่างต้นทางและปลายทางกันจริงๆ สามารถแบ่งการบริการเป็น 2 กรณีคือ

กรณีที่หนึ่ง ส่งข้อมูลของระดับชั้น 5 โดยนำมาใส่ข้อมูลของปลายทาง เช่นที่อยู่แล้วส่งออกไป โดยแบ่งช่วงข่าวสารยาวๆให้เป็นหน่วยที่เล็กลง เรียกว่าแพ็คเกจ (Packet) หรือเฟรม (Frame)

กรณีที่สอง เป็นบริการที่ทำการถอดรหัสข้อมูล ที่รับเข้ามาพิจารณาตัดสินว่ามี ความสัมพันธ์เกี่ยวข้องกับข้อมูล ที่ได้ส่งออกไปหรือไม่

เลเยอร์ชั้นทรานสปอร์ต จะทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งมาจากชั้นเลเยอร์ที่ 5 นั้นไปถึงปลายทางจริงหรือไม่ ดังนั้นการกำหนดตำแหน่งของข้อมูลจึงสำคัญในชั้นนี้

5. ระดับชั้น 3 เน็ตเวิร์คเลเยอร์ (Network Layer) เป็นชั้นที่ออกแบบหรือกำหนดเส้นทางการเดินทางของข้อมูลที่ส่ง-รับในการ ส่งผ่านข้อมูลระหว่างต้นทางและปลายทางซึ่งเป็นที่แน่นอนว่าในการสื่อสารข้อมูลผ่านเครือข่ายการสื่อสารจะต้องมีเส้นทางการรับ ส่ง ข้อมูลมากกว่า 1 เส้นทางดังนั้นในชั้นนี้จะมีหน้าที่เลือกเส้นทางที่ใช้เวลาในการสื่อสารน้อยที่สุดและระยะทางสั้นที่สุดด้วย

6. ระดับชั้น 2 ดาต้าลิงก์เลเยอร์ (Data Link Layer) เป็นเสมือนผู้ควบคุมการร่วมบิตเข้าเป็นตัวอักษรและจัด ข้อมูลให้เป็น แพ็คเกตรวมทั้งตรวจสอบความผิดพลาดในการส่งข้อมูล ถ้าผู้รับได้รับข้อมูลที่

แล้วถูกต้องก็จะส่งสัญญาณยืนยันกลับมายาว่าได้รับข้อมูลแล้วเรียกว่าสัญญาณ ACK (Acknowledge) ให้กับผู้ส่ง แต่ถ้าผู้ส่งไม่ได้รับสัญญาณ ACK หรือรับสัญญาณ NAK (Negative Acknowledge) กลับมาผู้ส่งอาจ จะทำการส่งข้อมูลไปใหม่อีกหน้าที่ของชั้นนี้ คือ ป้องกันไม่ให้เครื่องส่งทำการส่งข้อมูลเร็วจนเกินขีด ความสามารถของเครื่องผู้รับข้อมูลจะรับได้

7. ระดับชั้น 1 ฟิสิกส์คอลเลเยอร์ (Physical Layer) เป็นระดับชั้นการติดต่อระหว่างอุปกรณ์กับสื่อ กลางของเครือข่ายจริงๆ การส่ง ข้อมูลระดับชั้นนี้ข้อมูลนี้มีลักษณะเป็นบิต สื่อกลางที่ใช้ อาจจะเป็นสายไฟเบอร์ออปติก ไมโครเวฟ หรือ ดาวเทียม ตามระบบที่ใช้ โดยมาตรฐานสำหรับเลเยอร์ชั้นนี้จะกำหนดว่าแต่ละคอนเนคเตอร์ เช่น RS-232-C มีกี่พิน แต่ละพินทำหน้าที่อะไรบ้าง ใช้สัญญาณไฟฟ้ากี่โวลต์ เทคนิคการมัลติเพลกซ์แบบต่างๆ ก็จะถูกกำหนดอยู่ในชั้นเลเยอร์นี้

ในการสื่อสารกันระหว่างระบบหรือระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ โพรโตคอลแบบหนึ่ง ก็จะถูกใช้สำหรับการสื่อสารระหว่างชั้น N ของคอมพิวเตอร์เครื่องหนึ่งกับชั้น N เดียวกันของคอมพิวเตอร์ อีกเครื่องหนึ่งซึ่งในระหว่างการสื่อสารข้อมูลกันจริงๆ นั้น โพรโตคอลชั้น N ของอุปกรณ์ทั้งสองเครื่องจะสื่อสารกันผ่านชั้นล่างซึ่งเป็นกายภาพหรือวัตถุ (สื่อกลางการสื่อสาร) แต่โดยแนวความคิดของการสื่อสาร ข้อมูลแล้วถือว่าชั้น ที่ N ของเครื่องหนึ่งกำลังติดต่อโดยตรงกับชั้น N ของอีกเครื่องหนึ่งเรียกการสื่อสารแบบนี้ว่าการสื่อสาร แบบเสมือนจริง (Virtual Communication)

โพรโตคอลของในเลเยอร์แต่ละชั้นจะแตกต่างกันออกไปแต่อย่างไรก็ตามการที่เครื่องคอมพิวเตอร์หลายๆ เครื่องจะติดต่อสื่อสารกันได้ในแต่ละเลเยอร์ของแต่ละเครื่องจะต้องใช้โพรโตคอลแบบเดียวกันหรือถ้าใช้โพรโตคอลต่างชนิดกันต้องมีอุปกรณ์ หรือซอฟต์แวร์ที่สามารถแปลงโพรโตคอลที่ต่างกันนั้นให้มีรูปแบบ อย่างเดียวกันเพื่อเชื่อมโยงให้คอมพิวเตอร์ทั้ง 2 เครื่องสามารถติดต่อกันได้ในหนึ่งชั้นของเลเยอร์ไม่ได้ มีการกำหนดว่าจะต้องมีเพียงหนึ่งโพรโตคอลเท่านั้นที่อยู่ระดับเลเยอร์เดียวกันและ ในทางตรงข้าม ชุดของโพรโตคอลใดๆ อาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกันเป็นข้อกำหนดของระบบเครือข่าย เรียกว่า ชุดโพรโตคอล (Protocol Suite) เป็นชุดโพรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเลเยอร์คือ กำหนดการติดต่อระหว่างเลเยอร์ทำ โดยไม่ต้องคำนึงถึงการเปลี่ยนแปลงในเลเยอร์ใดๆ ที่ติดกัน

2.1.3 ทีซีพี/ไอพี [TCP/IP (Transmission Control Protocol/Internet Protocol)]

อินเทอร์เน็ตนับได้ว่าเป็นเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นเชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่นๆ แต่หากที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันก็คือ แต่ละเครือข่ายใช้ คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อ และระบบปฏิบัติการที่ต่างกัน มาตรฐานของ ทีซีพี (TCP) จึงถูกใช้เป็นกฎเกณฑ์สำคัญในการแก้ปัญหาเหล่านี้ โดยกลายเป็นระบบเปิดที่สมบูรณ์ แบบที่มีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่เครื่องพีซี จนถึงเครื่องเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ ทีซีพี จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน ทีซีพี/ไอพี มากมาย ดังนั้นจึงนับได้ว่า ทีซีพี/ไอพี เป็นหัวใจของอินเทอร์เน็ตเลยทีเดียว

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์หรือการสงวนสิทธิ์ในชื่อการค้าของ บริษัท หรือ องค์กรใด ๆ ในประเทศไทย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.1 ทีซีพี/ไอพี คืออะไร

ทีซีพี/ไอพี เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Network Protocol) จัดทำเพื่อใช้ เป็นกฎเกณฑ์ให้เครื่องคอมพิวเตอร์ใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็น คอมพิวเตอร์ชนิดใดหรือระบบใดก็ตาม จะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันใช้ได้

ทีซีพี/ไอพี เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่างๆ ที่กำหนดให้ทำเมื่อเกิดความผิดพลาด หรือต้องทำเพื่อป้องกันเพื่อไม่ให้เกิดความผิดพลาด

ทีซีพี/ไอพี เกิดจากการนำข้อกำหนดของรูปแบบต่างๆ กันมาใช้ร่วมกัน ทีซีพี (TCP) และ ไอพี (IP) ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ (แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า ทีซีพี/ไอพี) ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกัน หรือต่างเครือข่ายกันก็ได้และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบ เครือข่ายจะใช้ ทีซีพี/ไอพี เป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ (Local Area Network) และเครือข่ายบริเวณกว้าง (Wide Area Network) ไม่ได้ใช้งานเฉพาะ กับ อินเทอร์เน็ตเท่านั้น

2.1.3.2 ส่วนประกอบของ ทีซีพี/ไอพี

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่างๆซึ่งแบ่งเป็นกลุ่มได้ ดังนี้

2.1.3.2.1 กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocols) ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูล ระหว่างคอมพิวเตอร์สองเครื่อง แบ่งย่อยออกได้เป็นสองชนิด คือ

1.TCP (Transmission Control Protocol) เป็นการบริการแบบ คอนเน็คชันเบสเซอร์วิส (Connection Based Service) ซึ่งคอมพิวเตอร์ด้านผู้รับและส่งต้องส่งต่อถึงกันอยู่ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบก็คล้ายกับระบบโทรศัพท์ที่ต้องติดต่อกันให้ได้ ก่อนจะพูดคุยกันได้

2.UDP (User Datagram Protocol) เป็นการให้บริการแบบ คอนเน็คชันเซอร์วิส (Connection Service) คอมพิวเตอร์ด้านผู้ส่งไม่จำเป็นต้องติดต่อกับด้านผู้รับก่อน เพียงรู้ที่อยู่ของด้านผู้รับแล้วใส่ที่อยู่ นั้นไปกับข้อมูลที่ส่งออก ข้อมูลจะเดินทางตามเส้นทางต่างๆ เพื่อไปถึงปลายทางตามที่อยู่ คล้ายกับการส่งจดหมายที่ไปรษณีย์จะส่งให้ตามที่อยู่ที่กำหนดโดยผู้ส่งและผู้รับไม่ต้องติดต่อกัน

2.1.3.2.2 กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol) ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่ กลุ่มข้อมูลรูปแบบนี้จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสม แล้วส่งออกไป เมื่อถึงผู้รับปลายทาง กลุ่มข้อมูลนี้จะทำหน้าที่ตรงข้าม คือรวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผล กลุ่มข้อกำหนดรูปแบบกลุ่มนี้ประกอบด้วย

1. IP (Internet Protocol) เป็นการกำหนดรูปแบบการส่งข้อมูล

2. ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับ

สถานะของ IP เช่นข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ในเครือข่าย

ไม่มีการแจ้งเตือน ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. RIP (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับทำการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุดเหมาะสมกับข้อมูลมากที่สุด

4. OSPF (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินใจเลือกเส้นทางโดยพิจารณาจากทางที่สั้นที่สุดก่อน

2.1.3.2.3 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address) ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยที่ไม่ว่าเครือข่ายจะใหญ่โตสักเพียงใดหรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตาม ที่อยู่ต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบนี้มีดังนี้

1. ARP (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน

2. DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะการใช้งานจริงนั้นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้

3. RARP (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เดียวกับ ARP แต่จะทำตรงข้ามกับ ARP

2.1.3.2.4 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocols) และ สนับสนุนข้อมูลสถานะเพื่อนำไปใช้เลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1. EGP (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะหาการถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์ กับเครือข่ายภายนอกเพื่อทำการสื่อสาร

2. GGP (Gateway to Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์ กับ เกตเวย์

3. IGP (Interior Gateway Protocol) ข้อกำหนดในแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

2.1.3.2.5 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services) ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดรูปแบบนี้ประกอบด้วย

1. BOOTP (BOOT Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายให้เริ่มทำงาน ข้อกำหนดรูปแบบนี้จะอ่านโปรแกรมควบคุมการทำงานจากคอมพิวเตอร์ให้บริการ (Server Computer) มาให้

2. FTP (File Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนไฟล์ข้อมูลระหว่างคอมพิวเตอร์ ซึ่งอาจจะอยู่บนเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้

3. TELNET เป็นข้อกำหนดรูปแบบที่ทำให้บริการเกี่ยวกับการควบคุมการติดต่อระยะทางไกล กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญๆ จัดทำไว้ให้บนเครือข่ายที่สนใจมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. NFS (Network File System) เป็นข้อกำหนดรูปแบบที่ทำให้ผู้ใช้คอมพิวเตอร์เครื่องหนึ่งสามารถเข้าใช้งานไฟล์ข้อมูลและดูไฟล์ข้อมูลซึ่งอยู่ในคอมพิวเตอร์เครื่องอื่นได้

2. NIS (Network Information System) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Logins และ Password

3. RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล

4. SMTP (Simple Procedure Call) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนจดหมายอิเล็กทรอนิกส์ (Electronic Mail) ระหว่างคอมพิวเตอร์

5. SNMP (Simple Network Management Protocol) ข้อกำหนดรูปแบบที่ให้บริการข่าวสารต่างๆ ที่แสดงสถานะของเครือข่ายและอุปกรณ์ที่ต่ออยู่บนเครือข่าย

2.1.3.3 โครงสร้างของชุดโพรโทคอล ทีซีพี/ไอพี

OSI Model	TCP/IP(Internet)
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	Physical

ตารางที่ 2.2 โครงสร้างของชุดโพรโทคอล ทีซีพี/ไอพี เปรียบเทียบกับ แบบจำลอง โอ เอส ไอ

1. **Application Layer** ในชั้นนี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้ในเครือข่ายเช่นโปรแกรมส่งถ่ายข้อมูล (File-Transfer Program) และอาจกล่าวได้ว่าโพรโทคอล TCP/IP ก็คือโพรโทคอลในชั้นแอปพลิเคชันร่วมกับชั้นพีรีเซนเตชันของ OSI โมเดลนั่นเอง

2. **Transport Layer** ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายเปรียบเทียบกับชั้นทรานสปอร์ตเลเยอร์นั่นเอง โดยโพรโทคอล TCP/IP มีซ็อกเก็ต (Socket) เป็นจุดปลาย (End-Point) ในการสื่อสารซึ่งซ็อกเก็ตนี้ประกอบไปด้วย หมายเลขของคอมพิวเตอร์และหมายเลขพอร์ต (Port) ของเครื่องที่ต้องการส่งข้อมูลไปถึงในชั้นนี้มีการรับรองให้ถึงที่หมายและลำดับข้อมูลที่ส่งโดยปราศจากข้อมูลซ้ำซ้อน โดยในชั้นนี้มีโพรโทคอลหลัก 2 ตัว คือ TCP และ UDP

3. **Internet Layer** ในชั้นนี้มีการกำหนดดาต้าแกรม (Datagram) และทำการหาเส้นทางการส่งเอกสารทำงานในชั้นนี้จะเป็แบบ Connectionless เนื่องจากไม่มีการเชื่อมต่อระหว่างต้นทางกับปลายทาง ไม่ว่าจะกี่ครั้ง ทุกสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนดำเนินการแต่ละตัวสามารถเลือกเส้นทางไปโดยอิสระและไม่มีการรับประกันความถูกต้องข้อมูลหรือลำดับการส่ง

4. Network Interface Physical Layer ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็น ส่วน ๆ เรียกว่า เฟรม หรือแพคเกจและส่งข้อมูลที่ได้ไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

2.1.3.4 ข้อแตกต่างระหว่างชุดโพรโทคอล TCP/IP

1. ลำดับการติดต่อสื่อสารของชั้นเลเยอร์ ในรูปแบบ OSI นั้นจะกำหนดลำดับชั้นการสื่อสารที่เป็นลำดับขั้นตอนการติดต่อที่แน่นอน โดยเฉพาะการอินเตอร์เฟสระหว่างชั้นเลเยอร์ซึ่งทำให้รูปแบบ OSI สามารถเป็นระบบเปิดสำหรับระบบเครือข่ายคอมพิวเตอร์ทั่วไป เพราะไม่ว่าจะมีการเปลี่ยนแปลงโพรโทคอลในเลเยอร์ชั้นใดก็ตามจะไม่มีผลกระทบต่อ การสื่อสารเลเยอร์ชั้นถัดไป ในขณะที่ชุดโพรโทคอล TCP/IP จะไม่มีการกำหนดรูปแบบการติดต่อที่ตายตัวเพื่อให้ผู้ออกแบบเครือข่ายมีอิสระสามารถเปลี่ยนแปลงโครงสร้างของเครือข่ายได้ง่ายขึ้น

2. การติดต่อสื่อสารระหว่างเครือข่ายหรือการติดต่อผ่านอินเทอร์เน็ต คือ การติดต่อสื่อสารข้อมูลระหว่างระบบคอมพิวเตอร์ 2 ระบบที่ไม่สามารถติดต่อสื่อสารกันได้โดยผ่านทางเครือข่ายการสื่อสารข้อมูลเพียงเครือข่ายเดียวได้ต้องอาศัยเครือข่ายตั้งแต่ 2 เครือข่ายขึ้นไปในการติดต่อสื่อสารกันและเครือข่ายเหล่านี้จะมีลักษณะของเครือข่ายที่ต่างกันก็ได้

ความแตกต่างในเรื่องของอินเทอร์เน็ตระหว่างชุดโพรโทคอล TCP/IP กับรูปแบบ OSI ก็คือในชุดโพรโทคอล TCP/IP จะใช้โพรโทคอลสำหรับอินเทอร์เน็ตที่เรียกว่า โพรโทคอล IP (Internet Protocol) ซึ่งในรูปแบบ OSI จะเรียกว่าโพรโทคอลสำหรับการอินเทอร์เน็ตว่า โพรโทคอลเน็ตเวิร์ค

3. การบริการการเชื่อมต่อสื่อสาร (Connection Service) ในชุดโพรโทคอล TCP/IP นั้นจะมีการบริการการเชื่อมต่อสื่อสารระหว่างต้นทางและปลายทาง 2 แบบ คือการบริการแบบ Connectionless และแบบ Connection-Oriented ส่วนในรูปแบบ OSI จะให้ความสำคัญเฉพาะกับการบริการแบบ Connection-Oriented เท่านั้น

4. โพรโทคอลควบคุมการจัดการสื่อสาร ในชุดโพรโทคอล TCP/IP จะใช้โพรโทคอล TCP เป็นโพรโทคอลสำหรับควบคุมการสื่อสาร กำหนดตำแหน่งต้นทางและปลายทาง และอื่น ๆ กับข้อมูลซึ่งในรูปแบบ OSI นั้นจะแบ่งแยกการควบคุมการสื่อสารออกจากกันโดยใช้โพรโทคอลเซสชันและโพรโทคอลทรานสปอร์ตตามลำดับ

2.1.3.5 ลักษณะของการติดต่อ

แบ่งออกเป็น 2 ชนิดคือ

1. **Connection-Oriented** คือการติดต่อที่ต้องมีการเชื่อมโพรเซส (Process) ที่จะมีการส่งหรือรับข้อมูลซึ่งใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโพรเซส ถึงแม้ว่าข้อมูลนี้อาจจะจ่าย Packet-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสาร

ต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งขั้นตอนการทำงานออกเป็น ๓ ขั้นตอนไปใช้ประโยชน์ด้านการค้า
- ขั้นการสร้างการติดต่อ (Connection Establishment) อาจอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขั้นตอนการส่งผ่านข้อมูล (Data Transfer)
- ขั้นตอนยกเลิกการติดต่อ (Connection Termination)

2. **Connectionless** หรือดาต้าแกรม ก็จะไม่มีการสร้างการติดต่อ และขั้นตอนการยกเลิกการติดต่อ แต่จะมีขั้นตอนการส่งผ่านข้อมูลเพียงอย่างเดียว โดยข้อมูลซึ่งเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

2.1.3.6 ความสัมพันธ์ของ TCP และ UDP กับอินเทอร์เน็ต

ตามสถาปัตยกรรมของชุดข้อกำหนดรูปแบบ TCP/IP ระดับชั้น ทรานสปอร์ต จะมีบริการส่งข้อมูลอยู่ 2 ประเภท คือ TCP และ UDP

ความแตกต่างของ TCP กับ UDP อยู่ที่วิธีการสื่อสารระหว่างเครื่องคอมพิวเตอร์ สองเครื่อง TCP ต้องสร้างการติดต่อให้ได้ก่อนแล้วจึงส่งข้อมูลและยกเลิกเส้นทางที่ต่อกันเมื่อจบการส่งข้อมูล ส่วน UDP จะไม่มีการสร้างการติดต่อ แต่จะใช้ ไอพีแอดเดรสของคอมพิวเตอร์ปลายทางใส่รวมกับข้อมูลแล้วส่งออกไป ข้อมูลจะเดินตามไอพีแอดเดรสด้วยเส้นทางที่ไม่แน่นอน ขึ้นอยู่กับความเหมาะสมจนกว่าจะถึงคอมพิวเตอร์ปลายทาง

TCP เป็นวิธีการสื่อสารที่มีความเชื่อมั่นได้มากกว่า UDP ทั้งการส่งและการตอบรับข่าวสาร วิธีของ UDP จะไม่มีการรับประกันว่าข่าวสารจะไปถึงผู้รับแต่ UDP มีวิธีการที่จะตรวจสอบโดยให้เครื่องคอมพิวเตอร์ปลายทางส่งข่าวสารการตอบรับกลับมาเมื่อได้รับข่าวสารที่ส่งไปให้ ทางด้านผู้ส่งจะตั้งเวลารอานเท่าใดควรมีการตอบกลับรับกลับมา ถ้าเกินเวลา (time-out) แล้วยังไม่มีการตอบรับกลับมาก็สันนิษฐานได้ว่าข่าวสารสูญหาย UDP ก็จะทำการส่งข่าวสารเดิมให้ใหม่เครือข่ายขนาดใหญ่เช่น อินเทอร์เน็ต ซึ่งมีเครื่องคอมพิวเตอร์บนเครือข่ายเป็นจำนวนมาก การใช้ TCP ที่ต้องสร้างการติดต่อและยกเลิกอยู่บ่อย ๆ จึงเป็นไปได้ในทางปฏิบัติดังนั้นอินเทอร์เน็ต จึงใช้การส่งข่าวสารด้วย UDP แทน

การบริการที่ดำเนินการด้วยชุดข้อกำหนดรูปแบบ TCP/IP ต้องออกแบบการให้บริการเป็น TCP หรือ UDP อย่างใดอย่างหนึ่งเท่านั้น การบริการอย่างเดียวกันจะให้บริการทั้งสองประเภทไม่ได้เช่นการ Login จากระยะไกล (Telnet) และการ โอนย้ายไฟล์ข้อมูล FTP ใช้การให้บริการแบบ TCP ส่วน TFTP (Trivial File Transfer Protocol) ใช้การให้บริการแบบ UDP

2.1.3.7 พอร์ตและซ็อกเก็ต (TCP Port and Socket)

การใช้บริการของ TCP ทำโดยผ่านทางพอร์ต โดยมีการกำหนดหมายเลขให้กับการบริการต่าง ๆ การบริการที่ต่างกันจะมีหมายเลขพอร์ตไม่เหมือนกัน

พอร์ตถูกกำหนดตามชนิดของการบริการ เช่น ต้องการกำหนดหมายเลขให้กับการบริการต่าง การบริการที่ต่างกันจะมีหมายเลขพอร์ตไม่เหมือนกัน

พอร์ตถูกกำหนดตามชนิดของการบริการ เช่น ต้องการสื่อสารกันด้วย Telnet เครื่องคอมพิวเตอร์ที่ต้องการจะส่งความต้องการออกทางพอร์ตหมายเลข 23 ที่จัดไว้สำหรับบริการ Telnet ไปยังเครื่องคอมพิวเตอร์ที่ต้องการติดต่อด้วย ถ้าเครื่องคอมพิวเตอร์ปลายทางยอมรับการติดต่อ การสื่อสาร Telnet ก็ จะเริ่มขึ้น ในช่วงเวลานี้ถ้ามีความต้องการการสื่อสารด้วย Telnet จากเครื่องคอมพิวเตอร์อื่นเข้ามา จะไม่สามารถทำการติดต่อได้เนื่องจากพอร์ตหมายเลข 23 Telnet จากเครื่องคอมพิวเตอร์อื่นเข้ามา จะไม่

สามารถทำการติดต่อได้เนื่องจากพอร์ตหมายเลข 23 สำหรับบริการ Telnet ถูกใช้งานอยู่ TCP พอร์ตเป็นพอร์ตทาง Logical ไม่ใช่พอร์ตที่อยู่ด้านหลังของเครื่องคอมพิวเตอร์ที่ใช้คอนเน็กเตอร์ (Connector) บนแบคโบนเพื่อให้สามารถสื่อสารบริการหลายประเภทได้พร้อม ๆ กัน

ซ็อกเก็ตเป็นหมายเลขกำหนดวงจรซึ่ง ทางเข้า-ออก ของระดับชั้น TCP ของเครื่องคอมพิวเตอร์บนเครือข่าย หมายเลขซ็อกเก็ตประกอบด้วยเลข ไอพีแอดเดรส กับหมายเลขพอร์ตรวมกัน หมายเลขซ็อกเก็ตที่ใช้กับการสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่อยู่ต่างเครือข่ายได้เนื่องจาก ไอพีแอดเดรสของเครือข่ายจะไม่ซ้ำกัน ถึงแม้ว่าหมายเลขพอร์ตจะซ้ำกันได้

เครื่องคอมพิวเตอร์บนเครือข่ายจะสร้างตารางพอร์ตขึ้นมาเก็บรายการของพอร์ตที่มีอยู่เพื่อใช้ตรวจสอบว่าจะใช้บริการพอร์ตที่ต้องการได้หรือไม่และพอร์ตมีสถานะเป็นอย่างไรการที่พอร์ตที่อยู่ในรายการตารางพอร์ตของแต่ละเครื่อง ร้องขอการสื่อสารซึ่งกันและกัน เรียกว่า 'Binding' ของพอร์ตและวิธีที่ทำให้พอร์ตที่ถูกใช้สามารถสื่อสาร ได้อีกจากการขอใช้บริการเดียวกันเรียกว่า 'Multiplexing'

TCP พอร์ต	บริการ
20	FTP Data
21	FTP Control
23	Telnet
25	Simple Mail Transfer Protocol
53	Domain Name Server
69	Trivial File Transfer Protocol
79	Finger

ตารางที่ 2.3 หมายเลขพอร์ตที่ให้บริการต่างๆ

2.1.3.8 ข้อกำหนดรูปแบบ IP (Internet Protocol)

IP เป็นข้อกำหนดรูปแบบหนึ่งในชุดข้อกำหนดรูปแบบ TCP/IP ลักษณะการให้บริการเป็นแบบ Connectionless เช่นเดียวกับ UDP มีหน้าที่หลักเกี่ยวกับการจัดการ คาด้าแกรม ระหว่างเครื่องคอมพิวเตอร์ เช่น ที่อยู่ปลายทาง วิธีการหาเส้นทางที่ดีที่สุดที่จะไปถึงปลายทาง และจัดเตรียมวิธีการแก้ปัญหาที่อาจจะเกิดขึ้นในกรณีต่าง ๆ เป็นต้น นอกจากนี้ยังมีหน้าที่แบ่งย่อยคาด้าแกรมที่มีขนาดใหญ่เกินกว่าที่ IP Message กำหนดไว้ (ประมาณ 64 กิโลไบต์) และรวบรวมคาด้าแกรม ย่อย ๆ กลับมาเป็น Message ให้เหมือนเดิมที่เครื่องปลายทาง การแบ่งย่อยและรวบรวมคาด้าแกรม มีอยู่ 4 วิธีคือ

1. Segmentation วิธีแบ่งคาด้าแกรมขนาดใหญ่ให้เป็นคาด้าแกรมย่อย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตีตแบลงเนอติ และต้องอย่างองตงใจ ของเอกสารทุกครั้งที่มีการนำไปใช้

2. Reassemble วิธีรวบรวมคำด้าแกรมย่อย ๆ ให้กลับอยู่ในสภาพเดิม
3. Concatenation วิธีรวมคำด้าแกรมให้เป็นบล็อก
4. Separation วิธีแบ่งบล็อกให้เป็นคำด้าแกรม

2.1.3.9 ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol)

ปัญหาหนึ่งที่เกิดขึ้นได้ง่ายกับคำด้าแกรม ก็คือ คำด้าแกรมเดินทางผิดพลาด ทำให้สูญหายหรือ บางส่วนข่าวสารเสียหาย เครื่องคอมพิวเตอร์ด้านส่งต้องรู้เงื่อนไขที่จะทำให้เกิดความผิดพลาดต่าง ๆ บน เครือข่ายเพื่อให้คำด้าแกรมถูกส่งออกไปอย่างถูกต้อง

ICMP จะรายงานความผิดพลาดในระบบให้กับไอพีพิเศษ ICMP มี Header เหมือนกับข่าวสารไอพีอื่น ๆ โดยจะถูกส่งรวมไปกับข่าวสารอื่นสู่เครื่องปลายทาง ซึ่งเครื่องปลายทางจะส่งรายงานของความผิดพลาดกลับมาให้เครื่องด้านส่งได้รู้ว่าเกิดอะไรขึ้นบ้างในการเดินทาง ซึ่งข้อมูลในรายงานจะถูกนำมาใช้แก้ปัญหาต่อไป

2.1.4 ไอพีแอดเดรส (IP Address)

เครื่องคอมพิวเตอร์ที่ต่อเชื่อมกับอินเทอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP มีอยู่เป็นจำนวนมาก เครื่องคอมพิวเตอร์แต่ละเครื่องต้องสามารถระบุหรือ อ้างอิงได้โดยไม่เกิดความซ้ำซ้อนกับเครื่องคอมพิวเตอร์อื่น มิฉะนั้นแล้วข่าวสารที่เครือข่ายรับมาจะไม่สามารถส่งไปให้กับเครื่องคอมพิวเตอร์ที่ต้องการข่าวสารนั้นได้ จึงต้องมีการจัดระบบที่ดี เครือข่ายอินเทอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP ได้ออกแบบการจัดการระบบตรงส่วนนี้ไว้แล้ว เครือข่ายอินเทอร์เน็ตใช้รหัสหมายเลขมากำหนดให้แต่ละเครือข่ายและเครื่องคอมพิวเตอร์ภายในเครือข่ายที่เชื่อมโยงเรียกรหัสหมายเลขเหล่านี้ว่า อินเทอร์เน็ตแอดเดรส (Internet Address)

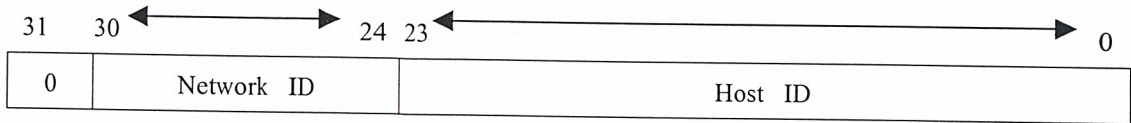
ไอพีแอดเดรส ประกอบด้วยเลขฐานสองจำนวน 32 บิต แบ่งออกเป็น 4 ส่วน แต่ละส่วนมี 8 บิต เมื่อดูเฉพาะแต่ละส่วนเป็นเลขฐานสิบจะได้เลขจำนวน 256 ค่าไม่ซ้ำกัน (0-255) ไอพีแอดเดรสจะนำเอาหมายเลขทั้ง 4 ส่วนมารวมกัน โดยแยกแต่ละส่วนด้วยจุดตั้งนั้นหมายเลขทั้งหมดที่เป็นไปได้ โดยไม่ซ้ำกันคือ 256^4 หรือ 4,294,967,296 จำนวน มีค่าหมายเลขจาก 000.000.000.000 จนถึง 255.255.255.255 หมายเลขเหล่านี้เองที่อินเทอร์เน็ตใช้กำหนดให้กับเครือข่ายและเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิงถึง

ไอพีแอดเดรสบางหมายเลขสงวนไว้ใช้ ด้วยจุดหมายกรณีพิเศษ ทำให้ ไอพีแอดเดรสที่ใช้งานทั่วไปลดลงจากจำนวนที่เป็นไปได้ ความหมายของไอพีแอดเดรสจะแบ่งได้เป็น 2 กลุ่มดังนี้

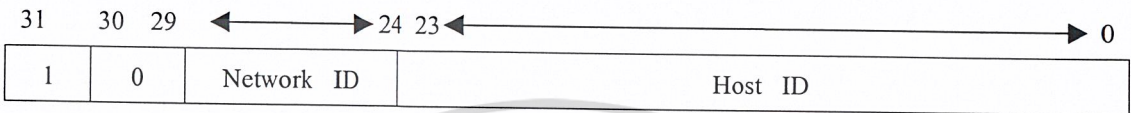
1. กลุ่มที่ใช้เป็นรหัสประจำเครือข่าย
2. กลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย (Host Computers) ไอพีแอดเดรสในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์สามารถซ้ำกันได้ แต่กลุ่มรหัสประจำเครือข่ายจะซ้ำกันไม่ได้ ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิงถึง

เอกสารนี้เป็นเอกสารนอกจากนี้เพื่อความเหมาะสมในการกำหนดไอพีแอดเดรสให้กับผู้ขอ ทางผู้บริหารอินเทอร์เน็ตจะไม่แบ่งคลาสของผู้ขอไอพีแอดเดรสตามขนาดของเครือข่ายเพื่อให้ทรัพยากรส่วนนี้ถูกใช้อย่างคุ้มค่าที่สุด

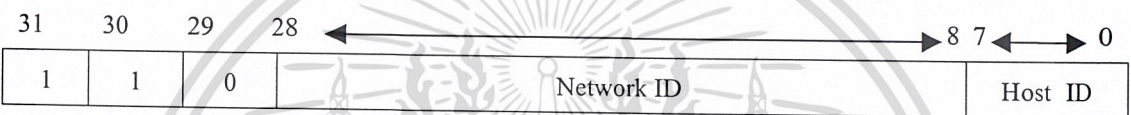
องค์กรขนาดใหญ่ก็จะจัดให้อยู่ในคลาสที่สามารถกำหนดไอพีแอดเดรสให้กับเครื่องคอมพิวเตอร์ในเครือข่ายได้มาก การแบ่งคลาสจะแบ่งได้ดังนี้



Class A IP address format



Class B IP address format



Class C IP address format

รูปที่ 2.2 แสดงคลาสของไอพีแอดเดรส

การกำหนดหมายเลขของเครือข่ายและเครื่องคอมพิวเตอร์นอกจากจะแบ่งไอพีแอดเดรสเป็นคลาสทั้ง 5 ประเภทแล้ว ยังมีข้อกำหนดปลีกย่อยอีกหลายประการที่ผู้วางระบบต้องรู้ไว้ เพื่อที่จะกำหนดแอดเดรสใช้งานได้อย่างถูกต้องไม่ผิดพลาด ได้แก่ ไอพีแอดเดรสที่เป็น '0' ทุกบิต จะไม่ใช้งานทั่วไป แต่นำไปใช้กับอุปกรณ์หาเส้นทาง (Router) เพื่อกำหนด 'Default Route'

1. ไอพีแอดเดรสในแต่ละส่วนคือ ส่วนหมายเลขเครือข่ายและส่วนหมายเลขประจำเครื่องคอมพิวเตอร์จะเป็น '0' หรือ '1' ทุกบิตไม่ได้

- ถ้าส่วนของหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '0' ทุกบิต หมายถึง หมายเลขเครือข่ายนั้นใช้งานร่วมกับอุปกรณ์หาเส้นทางหรือเพื่อบอกให้เครื่องคอมพิวเตอร์รู้ว่าตัวเองอยู่ในเครือข่ายใด

- ส่วนของหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '1' บิต หมายถึง เลขเครือข่ายนั้นใช้สำหรับการกระจายข่าวภายในเครือข่าย (Broadcast Address)

- ถ้าไอพีแอดเดรสทั้งสองส่วนเป็น '1' ทุกบิต หมายถึง แอดเดรสที่ใช้กระจายข่าวหรืออีกนัยหนึ่งเครื่องคอมพิวเตอร์ที่ส่งแอดเดรสนี้เข้าสู่ระบบไม่ทราบว่าตนเองอยู่ในเครือข่ายใด

2. ไอพีแอดเดรสของคลาส A หมายเลข 127.0.0.0 จะสงวนไว้ใช้งานเฉพาะอย่าง เช่น IPC (Inter - Process Communication)

จากรูปที่ 2.2 จะแสดงประเภทของไอพีแอดเดรสและข้อกำหนดปลีกย่อยต่าง ๆ เราสามารถทราบได้ว่าเครือข่ายขององค์กรถูกจัดอยู่ใน Class ไດ โดยดูจากค่าหมายเลขของ 8 บิตแรกซ้ายมือสุดดังนี้

เอกสารนี้เป็นเอ Class A: ไอพีแอดเดรสอยู่ในช่วง 1 ถึง 126 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด Class B: ไอพีแอดเดรสอยู่ในช่วง 128 ถึง 191 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class C: ไอพีแอดเดรสอยู่ในช่วง 192 ถึง 233

2.1.5 การติดต่ออินเทอร์เน็ตของวิซวลเบสิกโดยใช้ MS Winsock Control

ในการที่เราจะทำการเขียน Internet Application โดยใช้ Visual Basic จะมี Control (ในที่นี้คือ ActiveX Control) อยู่หลายตัวที่ทำหน้าที่ช่วยเรา ซึ่ง Control เหล่านี้บางตัวก็จะถูกออกแบบมาสำหรับบริการบางประเภทเท่านั้นซึ่งต้องเลือกเอาเอง ตามความเหมาะสม ซึ่ง Control เหล่านี้บางตัวก็จะถูกออกแบบมาสำหรับบริการบางประเภทเท่านั้นซึ่งต้องเลือกเอาเอง ตามความเหมาะสม ซึ่งบริการเหล่านั้นจะเป็นบริการพื้นฐานเท่านั้น ถ้าเราต้องการจะสร้าง Application เพื่อติดต่อกับบริการบางประเภทที่ไม่เป็นที่นิยม หรือเป็นรูปแบบบริการที่เราสร้างขึ้นเอง ก็คงจำเป็นที่จะต้องใช้ Control ที่เป็น Control ด้าน TCP หรือ UDP ซึ่งเป็น Control ที่ค่อนข้างจะเป็น Control ที่ทำหน้าที่จัดการเกี่ยวกับการติดต่อสื่อสารโดยใช้ Protocol ที่อยู่บน IP (Internet Protocol) โดยสามารถ สร้างการติดต่อสื่อสารที่เป็น TCP (Transfer Control Protocol) ซึ่งเป็น Protocol ที่เป็น Connection-Oriented กับอีก Protocol หนึ่งก็คือ UDP (User Datagram Protocol) ซึ่งเป็น Protocol แบบ Connection-Less ซึ่งก็เพียงพอที่เราจะสร้าง Application ใดๆก็ได้สำหรับ Internet

MS Winsock Control เป็น Control ที่จะทำให้การใช้งาน TCP และ UDP ได้ง่ายขึ้น ซึ่ง Control ตัวนี้สามารถที่จะใช้ได้ทั้ง MS Visual Basic, MS Visual C++, MS Visual FoxPro หรือแม้แต่ MS Access ซึ่งทำให้เราสามารถที่จะสร้าง Client หรือ Server Application ได้โดยที่ไม่จำเป็นต้องทราบรายละเอียดเกี่ยวกับ TCP หรือวิธีที่จะทำการเรียกใช้ Windows API ระดับล่างเลยเราเพียงแค่ว่า กำหนด Properties ต่างๆ แล้วเรียกใช้ Method เหล่านั้นเราก็จะสามารถที่จะสร้างการติดต่อสื่อสารกับเครื่องอื่นๆแล้วแลกเปลี่ยนข้อมูลข่าวสารซึ่งกันและกัน

โพรโทคอล TCP เบื้องต้น

TCP เป็นการติดต่อสื่อสารที่เป็น Connection Oriented ที่จะมีการสร้าง Connection กันไว้ ซึ่งก็จะทำให้ทั้งสองฝั่งสามารถที่จะส่งข้อมูลที่เป็นลักษณะ stream ไปมาระหว่างสองเครื่องนั้นได้ การสร้าง Application ในลักษณะนี้จะมีเครื่องอยู่สองฝั่งโดยฝั่งหนึ่งจะทำงานเป็น Server ที่จะรอให้เครื่อง Client ทำการขอการติดต่อสื่อสารเข้ามา

ในกรณีที่เราจะสร้าง Application ให้เป็น Client เราจำเป็นที่จะต้องทราบที่อยู่ของเครื่องปลายทางซึ่งอาจจะเป็น Hostname หรือหมายเลข IP เพื่อที่จะใช้ในการระบุใน RemoteHost property และต้องทราบหมายเลข Port เพื่อจะใช้ระบุใน RemotePort property ที่ทำการ Listen (รอการ connect) อยู่จากนั้นก็ทำการเรียก Connect method เพื่อสร้าง Connection

สำหรับถ้าต้องการสร้าง Application ที่เป็น Server ในส่วน Server นี้เราต้องกำหนด LocalPort property แล้วทำการเรียก Listen method เพื่อรอการร้องขอเข้ามาจาก Client เมื่อมีการร้องขอเข้ามา ก็จะเกิดเหตุการณ์ ConnectionRequest event เข้ามาเพื่อที่จะ สร้าง Coection เราจำเป็นต้องเรียก Accept method ในขณะที่เกิดเหตุการณ์ ConnectionRequest event

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสร้าง Connection แล้วเครื่อง Computer ทั้งสองฝั่งก็จะสามารถที่จะรับและส่ง Data ไปมา ระหว่างกันได้ ถ้าต้องการจะส่ง Data ก็ทำการเรียก SendData method และเมื่อมีข้อมูลเข้ามาก็จะเกิดเหตุการณ์ DataArrival event ขึ้น ถ้าจะอ่านข้อมูลก็ทำการเรียก GetData method ในขณะที่เกิดเหตุการณ์ DataArrival event

โพรโทคอล UDP เบื้องต้น

UDP (User Datagram Protocol) เป็น Protocol ที่ทำงานในลักษณะ Connection-less ซึ่งจะต่างจาก TCP ตรงที่ UDP ไม่มีการการสร้าง Connection และเครื่องที่มีการส่งการติดต่อสื่อสารกันนั้นเครื่องจะเป็น Client หรือ Server ก็ได้

ในการที่จะส่งข้อมูลหากันนั้น เครื่องที่เป็น Client จะเพียงแค่กำหนดหมายเลข Port ใน LocalPort property และกำหนดหมายเลข Port ใน RemotePort property ซึ่งจะต้องเป็นหมายเลขเดียวกันที่กำหนดไว้ที่เครื่อง Server จากนั้นก็ทำการเรียก SendData method เพื่อส่งข้อมูล และเมื่อเครื่อง Client ต้องการจะอ่านข้อมูลก็เพียงเรียก GetData method ในขณะที่มีเหตุการณ์ DataArrival event

คุณสมบัติของ Winsock Control (Properties)

Property นี้จะใช้สำหรับการกำหนดคุณสมบัติต่างให้กับวินซ็อก เช่น เลขหมายเครื่องปลายทาง เลขหมายพอร์ต ที่เราต้องการจะติดต่อ หรือจะเป็นการเรียกดูคุณสมบัติของ วินซ็อกในขณะนั้น เช่น ว่าเป็นวินซ็อก มีสถานะเป็นอย่างไร อ่านค่า หมายเลขที่อยู่ IP หรือ ค่าพอร์ต ของเราเอง เป็นต้น

ชื่อ Property	Return Value	อ่านได้อย่างเดียว?	คำอธิบาย
BytesReceived	Long	ใช่	เป็นการคืนค่ากลับจำนวนของ ไบต์ ที่รอในการรับจาก buffer ซึ่งใช้ GetData method ในการรับข้อมูล
LocalHostName	String	ใช่	เป็นคืนค่า ชื่อที่อยู่ของเรา เพื่อให้เรารู้ว่าเป็นอะไร
LocalIP	String	ใช่	คืนค่าของ Local host ที่อยู่ IP ในรูปแบบของ String เช่น 198.164.0.94 เป็นต้น
LocalPort	Long	ไม่ใช่	กำหนด port ที่รับข้อมูลของเครื่องเรา เช่น 1000 ถ้ามีข้อมูลส่งมาที่ IP เราและตรงกับ LocalPort ในที่นี้ คือ 1000 เราก็จะได้รับ ข้อมูลนี้จากวินซ็อกตัวที่มี LocalPort อยู่ที่ 1000
Protocol	Long	ไม่ใช่	เป็นการ ตั้งค่าให้กับ Winsock ว่า จะใช้ Protocol อะไรซึ่งมีให้เลือก คือ skTCPProtocol และ skUDPProtocol (TCP และUDP) ซึ่งการคืนค่าคือ 0 และ 1 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ต่อ)

ชื่อ Property	Return Value	อ่านได้เพียงอย่างเดียว?	คำอธิบาย
RemoteHost	String	ไม่ใช่	คืนค่าหรือตั้ง เครื่องปลายทางที่ติดต่อ ซึ่งสามารถที่จะตั้งให้เป็นแบบ String หรือ เลขหมายที่อยู่ IP ก็ได้แบบ String ก็เช่น www.kmitl.ac.th และ แบบเลขหมายที่อยู่ ก็เช่น 161.246.10.21
RemoteHostIP	String	ใช่	คืนค่า ที่อยู่ IP ของเครื่องปลายทาง สำหรับ TCP นี้จะได้ค่าเมื่อการติดต่อเสร็จแล้วเท่านั้น ส่วน UDP จะคืนค่านี้ได้เมื่อเกิดเหตุการณ์ DataArrival , ซึ่งตอนนั้นก็จะได้รับค่า ที่อยู่ IP ของเครื่องที่ส่งข้อมูลมาด้วย
RemotePort	Long	ไม่ใช่	จะคืนค่าหรือ ตั้งค่าพอร์ต เพื่อที่เราจะส่งข้อมูลไป
SocketHandle	Long	ใช่	คืนค่าที่ตรงกับที่ใช้ SocketHandle
State	Integer	ใช่	คืนค่าสถานะของตัวควบคุม, ซึ่งจะมีเป็นแต่ละชนิดไป

ตารางที่ 2.4 คุณสมบัตินวินซ็อกคอนโทรล

คุณลักษณะสถานะ (State Property)

คุณลักษณะสถานะนี้ จะส่งค่าสถานะของวินซ็อกเป็นค่า ๆ ออกมาว่าขณะนั้นวินซ็อกตัวนั้นอยู่ในสถานะใด อาจจะเอาสถานะตัวนี้ไปเป็นเงื่อนไขในการเขียน โปรแกรมเพื่อป้องกันการเกิดการผิดพลาดที่เกี่ยวกับข้อบังคับของวินซ็อกได้ หรือจะใช้เพื่อเช็คว่าขณะนี้กำลังทำอะไรอยู่เพื่อที่เราจะได้เขียนโปรแกรมได้ว่าถ้าเป็นอย่างนี้จะทำอะไรได้บ้าง ซึ่งโดยที่รายละเอียดทั้งหมดจะระบุอยู่ที่ตารางนี้

ค่าConstant	ค่า	ความหมาย
SckClosed	0	ปิด เป็นค่าปกติ
SckOpen	1	เปิด ซ็อกเกต
SckListening	2	การรอรับรู้อำนาจการเชื่อมต่อ
SckConnectionPending	3	การร้องขอการเชื่อมต่อมาถึงแล้วแต่ยังไม่เสร็จ
SckResolvingHost	4	Host ทำตามความต้องการ
SckHostResolved	5	ความต้องการของ Host เป็นจริง
SckConnecting	6	การร้องขอการติดต่อเริ่มทำ แต่ ยังไม่เสร็จ
SckConnected	7	การเชื่อมต่อสำเร็จ
SckClosing	8	เริ่มต้นการปิด
SckError	9	มีการผิดพลาดเกิดขึ้น

ตารางที่ 2.5 คุณลักษณะสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Methods ของ Winsock Control

Methods นี้จะเป็นวิธีการที่จะให้ วินซ็อกทำตามความต้องการของเราในการเขียนโปรแกรม การกำหนดก็จะมีลักษณะที่จะขึ้นต้นด้วยชื่อของ วินซ็อกแล้วตามด้วยวิธีที่จะให้วินซ็อกทำ เช่น ต้องการให้วินซ็อกที่ชื่อ winsock1 สร้างการเชื่อมต่อ ก็จะใช้คำสั่งดังนี้ Winsock1.connect

การที่จะใช้ Methods นี้ส่วนใหญ่จะต้องมีการกำหนด คุณสมบัติ ให้กับวินซ็อกก่อนจึงต้องระวังด้วยว่าเราได้กำหนดค่าให้หรือไม่ ตัวอย่างเช่น การที่เราจะใช้ Winsock1.connect ได้นั้นต้องมีการกำหนด หมายเลขIP และ หมายเลข Port ปลายทางให้กับ วินซ็อกก่อน เป็นต้น ซึ่งรายละเอียดต่างๆของ Method สามารถแสดงรายละเอียดได้ดังตารางต่อไปนี้

Method	พารามิเตอร์	Return Value	คำอธิบาย
Accept	RequestID	Void	สำหรับการติดต่อแบบ TCP เท่านั้น ใช้ Method นี้ ในการยอมรับการติดต่อเมื่อเกิดเหตุการณ์ ConnectionRequest
Bind	LocalPort LocalIP	Void	ทำการเชื่อมต่อ ซ็อกเก็ตให้กับ LocalPort และ LocalIP ใช้ Bind ถ้ามี Network หลายตัว Bind จะต้องถูกเรียกก่อน Listen
Close	None	Void	ปิดการติดต่อ หรือ ปิดการรอรับการติดต่อ
Connect	RemoteHost RemotePort	Void	สร้างการเชื่อมต่อของ TCP ไปยังหมายเลข RemotePort ที่อยู่บน RemoteHost
GetData	Data Type MaxLen	Void	รับเอาข้อมูลที่ค้างอยู่เข้ามา มี Type และ MaxLen เป็นพารามิเตอร์ควบคุม พารามิเตอร์ Type จะใช้กำหนดชนิดของข้อมูลสำหรับอ่าน พารามิเตอร์ MaxLen จะใช้จำนวนไบต์ หรือ ตัวอักษรที่รับเข้ามา GetData จะข้ามพารามิเตอร์ MaxLen สำหรับชนิดอื่น ๆ ที่มากกว่า ไบต์อะเรย์ และ string.
Listen	None	Void	สร้าง ซ็อกเก็ตและวางมันไว้ในโหมดการรอรับการติดต่อ Listen จะใช้ สำหรับการติดต่อแบบ TCP เท่านั้น
PeekData	Data Type MaxLen	Void	คุณสมบัติเหมือนกับ GetData ยกเว้นแต่ว่าข้อมูลไม่ได้ถูกย้ายจาก Buffer ของระบบ
SendData	Data	Void	ส่งข้อมูลไปยังเครื่องปลายทาง ถ้า UNICODE string ผ่าน มันก็จะถูกแปลงเป็น ANSI string ก่อน จำไว้ว่าต้องใช้ ไบต์อะเรย์สำหรับข้อมูลไบนารีเสมอ

ตารางที่ 2.6 Method ของวินซ็อกคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Events ของวินซ็อก

Event คือเหตุการณ์ที่เกิดขึ้นเมื่อมีลักษณะของ ซ็อกเก็ตจากภายในและภายนอกเปลี่ยนไปจากเดิม เช่นการเข้ามาของข้อมูล การทำการติดต่อสำเร็จ

เราสามารถใช้ประโยชน์ของ Events ในการที่เรากำหนดให้กับโปรแกรมว่าเราต้องการให้โปรแกรมทำอะไรเมื่อเกิดเหตุการณ์นี้ขึ้น ในกรณีเช่น เมื่อมีข้อมูลเข้ามาถึงซ็อกเก็ต (ในที่นี้คือ DataArrival)ของเราให้เรารับข้อมูลนี้ไว้ (โดยใช้ GetData Method) ซึ่งแต่ละ Event ก็มีดังตารางข้างล่างนี้

Event	Arguments	คำอธิบาย
Close	ไม่มี	เกิดขึ้นเมื่อขาดการติดต่อกันกับเครื่องปลายทางที่เราติดต่อด้วย
Connect	ไม่มี	เกิดขึ้นหลังจาก Connect method ติดต่อได้สำเร็จแล้ว
ConnectionRequest	RequestID	เกิดขึ้นเมื่อเครื่องปลายทางต้องการจะติดต่อด้วย
DataArrival	BytesTotal	เกิดขึ้นเมื่อมีข้อมูลใหม่เข้ามา
Error	Number Description Scode Source HelpFile HelpContext CancelDisplay	เมื่อ Winsock มีข้อผิดพลาด Event นี้ก็จะถูกสร้างขึ้น
SendComplete	ไม่มี	เกิดขึ้นเมื่อคำสั่งการส่งทำได้สำเร็จเรียบร้อยแล้ว
SendProgress	BytesSent bytesRemaining	เกิดขึ้นขณะที่ข้อมูลทำการส่ง

ตารางที่ 2.7 Event ของวินซ็อก

2.2 การบีบอัดเสียง (Voice Compression)

ในการที่จะส่งเสียงผ่านระบบ อินเทอร์เน็ตนั้นในเบื้องต้นจะมีการเปลี่ยนสัญญาณเสียงพูดที่ได้รับจากไมค์โครโฟนซึ่งเป็น Analog เป็น Digital ก่อนซึ่งส่วนนี้ระบบ ปฏิบัติการและของ การ์ดเสียงจะเป็นตัวทำให้โดยอัตโนมัติอยู่แล้วจึงจะไม่กล่าวถึง แต่ว่าการที่เราจะส่งสัญญาณออกไปนั้นจะเอาสัญญาณที่ได้จากการแปลงข้างต้นนั้นไม่ไหวแน่เพราะว่าขนาดของข้อมูลจะใหญ่มากซึ่งจะสังเกตได้จาก เมื่อเราทำการบันทึกเสียงเข้าทางไมค์โครโฟนไฟล์ที่ได้มาจะมีขนาดใหญ่มาก เพราะฉะนั้น จึงต้องมีการบีบอัดข้อมูลของเสียงก่อนที่จะส่งออกไปซึ่งการ Compression ที่ใช้มีดังนี้

เอ็กสแอมเปิลของเสียงก่อนที่จะส่งออกไปซึ่งการ Compression ที่ใช้มีดังนี้ เอ็งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GSM610 Compression

วิธีการบีบอัดข้อมูลแบบ GSM นี้ได้ถูกพัฒนาขึ้นที่วิทยาลัยเทคนิคของเบอร์ลินในปี ค.ศ.1992 วิธีการบีบอัดแบบ GSM นี้เป็นวิธีที่มีความซับซ้อนในการบีบอัดข้อมูลเป็นอย่างมาก ซึ่งผลที่ได้รับคืออัตราการบีบอัดข้อมูลมีมากถึง 10 ต่อ 1 การทำงานของ GSM610 Compression มีพื้นฐานแนวความคิดมาจาก Global System for Mobile Telecommunication Protocol ซึ่ง Protocol นี้เป็นที่นิยมอย่างมากในการสื่อสารของระบบ โทรศัพท์เซลลูล่าร์ดิจิทัล

Input ของ GSM ประกอบด้วย Frame ข้อมูลขนาด 160 ตัวแบบมีเครื่องหมายซึ่งได้มาจากวิธีการ PCM แบบที่ได้ข้อมูล 13 บิต ต่อการ แซมปลิง (Sampling) 1 ครั้งซึ่งความถี่ในการ แซมปลิง คือ 8000 ครั้ง ต่อ 1 วินาที GSM จะมีวิธีการแปลง เฟรมข้อมูลของ PCM ไปเป็น GSM และสามารถทำการแปลงกลับได้ ตัวเข้ารหัสจะทำการบีบอัดข้อมูลขนาด 160 ตัว แต่ละตัวมีขนาด 16 บิต ให้เป็นของข้อมูลลงเหลือเพียง 1625 ไบต์ เท่านั้น หากว่าถ้าเราทำการบีบอัดข้อมูลแบบ GSM ลงในหน่วยความจำขนาด 1 MB. เราจะสามารถเก็บข้อมูลได้นานถึง 10 นาที

เป็นต้นว่าหากใช้ข้อมูล 16 Bit/การ แซมปลิง 1 ครั้ง จะได้

$$(264 \text{ Bit} \times 8000 \text{ Samples/Sec.})/160 \text{ Sample} = 13.2 \text{ Kbits/second}$$

True Speech

True Speech 6.3/5.3 หรืออีกชื่อหนึ่งคือ True Speech G.723 เป็นสมาชิกใหม่ของกลุ่ม True Speech ซึ่งเป็นวิธีการบีบอัดข้อมูลเสียง ถูกพัฒนาโดย บริษัท DSP GROUP INC. มีอัตราบีบอัดข้อมูลอยู่ในช่วง 20 ต่อ 1 และ 24 ต่อ 1 (6.3 KBPS และ 5.3 KBPS ตามลำดับ) True Speech 6.3/5.3 หรือ G.723 และได้เป็นมาตรฐานของ International Telecommunication Union (ITU) ที่ได้กำหนดไว้ G.723 ได้ถูกนำมาใช้เป็นมาตรฐานของการสื่อสารด้วยภาพและเสียงผ่านเครือข่ายโทรศัพท์สาธารณะ และเป็นส่วนหนึ่งของ ITU H.324 ซึ่งเป็นมาตรฐานในการสื่อสารสัญญาณเสียงและสัญญาณภาพ โดยที่ True Speech 6.3 มีค่าอัตรา 6.3 kbps ใช้การบีบอัดที่ 20 ต่อ 1 และ True Speech 5.3 มีค่าอัตรา 5.3 kbps ใช้การบีบอัดที่ 24 ต่อ 1

คุณภาพของสัญญาณเสียงถูกทดสอบโดยการทดสอบที่เรียกว่า MEAN OPINION SCORE (MOS) ซึ่งใช้ค่าตัวเลขบอกถึงคุณภาพในระดับ สเกล (Scale) ที่สูงที่สุด แสดงถึงคุณภาพของเสียงที่ใช้ในสายโทรศัพท์ธรรมดา จากการทดสอบ True Speech G.723 ได้รับค่าถึง 3.98 ดังนั้น True Speech G.723 จึงสามารถสื่อสารด้วยเสียงผ่านระบบ Internet ได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 MCI (Multimedia Control Interface)

2.3.1 สถาปัตยกรรมของ MCI (The Architecture of MCI)

MCI (Media Control Interface) คือฟังก์ชันจำนวนหนึ่งที่ วินโดวส์ ได้จัดเตรียมไว้ เพื่อให้โปรแกรมเมอร์ สามารถเขียนโปรแกรม ควบคุมอุปกรณ์และไฟล์มัลติมีเดีย (เช่น .wav, .mid, .avi, หรือ CD Audio) ได้ โดยไม่ต้องเข้าใจการทำงานระดับล่างของฮาร์ดแวร์ และไม่ต้องเข้าใจฟอร์แมตของไฟล์มัลติมีเดีย เพียงแต่เขียนโปรแกรมในลักษณะที่เข้าใจได้ง่าย เช่น สั่งให้ MCI เล่นเพลง, หยุด เป็นต้น

2.3.2 การเขียนโปรแกรมอินเทอร์เฟซ(The MCI Programming Interfaces)

MCI ได้เตรียมโปรแกรมสำหรับอินเทอร์เฟซไว้อยู่ 2 ตัวคือ command string และ command message

2.3.2.1 อินเทอร์เฟซ Command-String (The Command-String Interface)

อินเทอร์เฟซ Command-String จะทำตามคำสั่งคุณ โดยการใช้ ภาษาอังกฤษ เป็นคำสั่งในการติดต่อ กับ ส่วนของ MCI สำหรับตัวอย่าง หากต้องการจะใช้ ไฟล์เวฟ (WAVE file) ที่ชื่อว่า TIMPANI.WAV ก็ใช้

```
play timpani.wav
```

อินเทอร์เฟซ Command-String นี้เป็นการออกแบบสำหรับใช้กับ การโปรแกรมระดับสูง (high-level programming) และผู้ที่เขียนโปรแกรมที่เป็นสภาพแวดล้อมอย่างเช่น วิชาลเบสิก (Microsoft Visual Basic) อซิเมตริกทูลบุ๊ก (Asymetrix ToolBook) เดลฟี (Delphi) และอื่นๆ จะใช้อินเทอร์เฟซ command-string สำหรับการประยุกต์ นั่นคือจะต้องมีการเตรียมอินเทอร์เฟซ text-based ที่ให้กับผู้ใช้ การควบคุมอุปกรณ์ MCI

2.3.2.2 อินเทอร์เฟซ Command-Message (The Command-Message Interface)

อินเทอร์เฟซ command-message นี้ใช้ แบบ message-passing ในการติดต่อกับอุปกรณ์ MCI ตามตัวอย่างก็มีแบบเหมือนกับการดำเนินการเหมือนกับตัวอย่าง command-string คือ

```
WORD wDeviceID;
```

```
MCI_PLAY_PARMS mciPlayParms;
```

```
mciSendCommand(wDviceID, /* device identifier */
```

```
MCI_PLAY, /* command message */
```

```
0, /* flags */
```

```
(DWORD) (LPVOID) &mciPlayParms); /* structure */
```

อินเทอร์เฟซ command-message เป็นการออกแบบโดยใช้การต้องการของแอปพลิเคชัน (application) เช่น อินเทอร์เฟซของภาษาซี ในการควบคุมอุปกรณ์มัลติมีเดีย ใช้อินเทอร์เฟซ command-message สำหรับแอปพลิเคชันนั้นเป็นการควบคุมอุปกรณ์มัลติมีเดียโดยตรง

2.3.3 ชุดคำสั่ง MCI (The MCI Command Set)

การเซ็ตคำสั่ง MCI เป็นการเตรียมชื่อในจำพวกของคำสั่งที่จะควบคุมความแตกต่างแต่ละชนิดของ อุปกรณ์มีเดีย สำหรับตัวอย่าง MCI ใช้คำสั่งเหมือนกับการเริ่มต้นเล่นกลับของ เสียงรูปแบบเวฟ วิดีโอดิสก์เทร็ค (videodisc track) และ อนิเมชันซีเควนซ์ (animation sequence) บางอุปกรณ์มีความสามารถอย่างเดียว เช่น ความสามารถในการเล่น อนิเมชัน ที่ใช้รูปแบบเวลา (time format) แบบ frame-based สำหรับอุปกรณ์เหล่านั้น MCI จะจัดเตรียมคำสั่งส่วนขยายนั้นคืออุปกรณ์จะมีส่วนของชนิดเพียงส่วนเดียว คำสั่งในอินเทอร์เฟซ command-string ได้มีการจัดเตรียมโอเวอร์วิว (overview) ของชุดคำสั่ง MCI แต่ละคำสั่งเป็นการแสดงโดยเหมือนกับข้อความใน command-message สำหรับตัวอย่างคำสั่ง close ของ string ก็จะมีเหมือนกับข้อความคำสั่ง คือ MCI_CLOSE ตามหลักจะใช้บางคำสั่งดังนี้

คำสั่ง(Command)	คำอธิบาย(Description)
capability	ข้อมูลที่ต้องการเกี่ยวกับความสามารถของดีไวซ์(device)
close	ปิดดีไวซ์หลังจากที่ใช้เสร็จแล้ว
info	ข้อมูลที่ต้องการเกี่ยวกับดีไวซ์(ตัวอย่าง , การอธิบายของความร่วมมือของส่วนเครื่องกับดีไวซ์)
open	เปิดและเริ่มต้นกับดีไวซ์
pause	หยุดชั่วคราว ในการเล่น หรือ การบันทึก บนดีไวซ์
play	เริ่มต้นเล่นบนดีไวซ์
record	เริ่มต้นบันทึกบนดีไวซ์
resume	กลับมา เล่นหรือบันทึก เมื่อทำการหยุดชั่วคราว
seek	เปลี่ยนจุดที่อยู่ในมีเดีย
set	เปลี่ยนการจัดการควบคุม, เช่น รูปแบบของเวลา ดีไวซ์ที่ใช้
status	ต้องการข้อมูลที่เกี่ยวข้องกับสถานะของดีไวซ์(ยกตัวอย่างได้คือ, ในขณะที่ดีไวซ์ทำการเล่นหรือหยุดอยู่)
stop	หยุดการเล่นหรือการบันทึกบนดีไวซ์

2.3.4 เกี่ยวกับ MCI ดีไวซ์ (About MCI Devices)

ตัวจัดการ MCI สามารถที่จะจัดหมวดหมู่เป็นแต่ละชนิดตาม ซิมเปิ้ลดีไวซ์(simple device) หรือ คอมปาวด์ดีไวซ์ (compound device) ซิมเปิ้ลดีไวซ์ นั้นจะไม่ต้องการไฟล์ข้อมูลสำหรับเล่นกลับ เช่น เครื่องเล่นวีดีโอดิสก์ (videodisc players) และ คอมแพคดิสก์ (compact disc (CD)) ส่วนคอมปาวด์ดีไวซ์ นั้น ต้องการเพิ่มข้อมูลสำหรับเล่นกลับ เช่น MIDI ซีเคว้นซ์ และ เสียงรูปแบบเวฟ (waveform audio)

เพิ่มข้อมูลนี้จะร่วมกับ คอมปาวด์ดีไวซ์เป็นการรู้กันในชื่อ ดีไวซ์อิลิเมนต์ (device element) ตัวอย่างของ ดีไวซ์อิลิเมนต์ก็เช่น เพิ่ม MIDI เพิ่ม WAVE

2.3.5 มาตรฐานชนิด MCI ดีไวซ์(Standard MCI Device Types)

ดีไวซ์ใหญ่ (device type) ถูกกำหนดโดยชั้นของ MCI ดีไวซ์ นั่นคือเป็นการตอบสนองชุดหลักของคำสั่งตามที่ทำให้ความหมายของดีไวซ์ใหญ่ได้ดังนี้

ดีไวซ์ใหญ่	คำอธิบาย
animation	แอนิเมชันดีไวซ์ (Animation device)
cdaudio	เครื่องเล่นเสียง ซีดี (CD audio player)
dat	วิดีโอดิจิทัลในวินโดวส์
other	อื่นๆที่ไม่กำหนดใน MCI ดีไวซ์
overlay	โอเวอร์เลย์ดีไวซ์ (overlay device)(อะนาล็อกวิดีโอในวินโดวส์)
scanner	อิมเมจสแกนเนอร์ (Image scanner)
sequencer	มีดีซีควนเซอร์ (MIDI sequencer)
videodisc	เครื่องเล่นวิดีโอดีสก์ (Videodisc player)
waveaudio	ออดิโอดีไวซ์ (audio device) ใช้เล่นเพิ่มรูปแบบเวฟ ดิจิตอล

2.3.6 ชื่อดีไวซ์ (Device Name)

สำหรับทุก ๆ ดีไวซ์ใหญ่ที่ให้มา พบว่าเป็นไดรฟ์เวอร์ MCI นั่นเป็นการแบ่งชุดคำสั่งแต่การดำเนินการเป็นการทำบนความแตกต่างบนรูปแบบข้อมูล การกำหนดไดรฟ์เวอร์ MCI จะใช้ชื่อของดีไวซ์ ชื่อดีไวซ์ จะเป็นตัวกำหนดในส่วน[mci]ของการขึ้นทะเบียน นี่เป็นส่วนทั้งหมดที่กำหนด MCI ดีไวซ์ไดรฟ์เวอร์ (device drivers) ที่วินโดวส์ตามส่วนของชนิด [mci]

[mci]

waveaudio=mciwave.driv

sequencer=mciSEQ.driv

MMMovie=mcimmp.driv

Cdaudio=mcicda.driv

คีย์เนม ((key name) จากด้านบนทางซ้ายของเครื่องหมายเท่ากับ) เป็นชื่อดีไวซ์ ค่าที่สอดคล้องกับชื่อดีไวซ์จะกำหนดชื่อไฟล์ ของไดรฟ์เวอร์ MCI ชื่ออุปกรณ์ มีบอຍที่ไปเหมือนกับชนิดอุปกรณ์ (device type) สำหรับไดรฟ์เวอร์อย่างเช่น ในเหตุการณ์ของ waveaudio, sequencer และ cdaudio

ถ้าไดรฟ์เวอร์ MCI ถูกติดตั้งโดยใช้ชื่ออุปกรณ์นั้นก็จะเริ่มอยู่ในส่วน [mci] แล้ว วินโดวส์ก็จะเพิ่มจำนวนเต็มชื่อดีไวซ์ของไดรฟ์เวอร์ใหม่ สร้างชื่อดีไวซ์ที่มีอยู่ชื่อเดียว ในตัวอย่างที่ผ่านมา ไดรฟ์เวอร์จะถูกติดตั้ง ชื่อดีไวซ์ของ cdaudio จะกำหนดชื่อดีไวซ์เป็น cdaudio1 ดีไวซ์ของ cdaudio ตัวต่อมาก็จะ

ไม่ใช้การกำหนดชื่อ ดีไวซ์เป็น cdaudio2 แปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7 การเปิด MCI ดีไวซ์

ก่อนที่จะใช้ MCI ดีไวซ์ เราต้องเริ่มต้นโดยการเปิดมันขึ้นมา การเปิดนี้เป็นการที่ดีไวซ์ นำเอา ไดรฟ์เวอร์ของมันเข้าไปไว้ในหน่วยความจำและตั้งค่าระบุให้ ดีไวซ์ในคำสั่งที่ตามมา (การระบุค่านี้อาจจะไม่ใช้ในอินเทอร์เน็ตเฟส command-string) ซึ่งวิธีที่เราจะกำหนด อุปกรณ์ที่ต้องการเปิด

- กับซิมเบิลดีไวซ์ เราสามารถเปิดอุปกรณ์แบบนี้โดยการกำหนดชื่ออุปกรณ์
- กับคอมปาวด์ดีไวซ์ เราสามารถเปิดอุปกรณ์โดยกำหนดชื่ออุปกรณ์ ส่วนอุปกรณ์หรือทั้งสองส่วน

สำหรับตัวอย่าง ตามคำสั่ง command string ก็คือที่เปิด CD ออดิโอดีไวซ์ (นี่เป็นตัวอย่างอุปกรณ์) โดยกำหนดชื่ออุปกรณ์ดังนี้

```
open cdaudio
```

ถัดไปก็เป็น command string ที่เปิด เวฟฟอร์มออดิโอดีไวซ์ (เป็นคอมปาวด์ดีไวซ์) โดยกำหนดชื่อ ดีไวซ์และอิลิเมนต์ดีไวซ์

```
open bell.wav type waveaudio
```

เรายังสามารถเปิด คอมปาวด์ดีไวซ์ ได้โดยกำหนดแค่ อิลิเมนต์ดีไวซ์เท่านั้นซึ่งจะแสดงให้ดูในตัวอย่างข้างล่างนี้

```
open bells.wav
```

ถ้าเรากำหนด อิลิเมนต์ดีไวซ์เพียงอย่างเดียวเมื่อทำการเปิด คอมปาวด์อิลิเมนต์ MCI ก็จะใช้ชื่อไฟล์ตามชื่อ อิลิเมนต์ดีไวซ์ ซึ่งดีไวซ์จะถูกเปิดการขึ้นทะเบียนก็จะใส่ส่วน [mci extensions] นั้นจะร่วมมือกับ ชื่อไฟล์ขยาย (filename extensions) และสอดคล้องกับชนิดของ ดีไวซ์ MCI ตามส่วนของ [mci extensions]

```
[mci extensions]
```

```
wav = waveaudio
```

```
mid = sequencer
```

```
rmi = sequencer
```

2.3.8 การส่งคำสั่งข้อความคำสั่ง (Sending Command Messages)

วินโดวส์ได้มีการจัดเตรียมฟังก์ชันที่จะใช้ส่งข้อความคำสั่งไปยัง ดีไวซ์ และ คิวรีดีไวซ์ (query devices) สำหรับข้อมูลที่เกิดผิดพลาด

ฟังก์ชัน

คำอธิบาย

mciSendCommand

ส่งข้อความคำสั่ง ไปยัง MCI ดีไวซ์

mciGetErrorString

นำค่าสตริง (string) ที่ผิดพลาดกลับมาโดยที่สอดคล้องกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.9 การระบุ MCI ดีไวซ์ (The MCI Device Identifier)

MCI จะส่งค่าของการระบุดีไวซ์กลับก็เมื่อ เราทำการเปิดดีไวซ์โดยใช้ข้อความคำสั่ง MCI_OPEN ใช้การระบุที่เหมือนกันนี้ในการเปิดดีไวซ์เมื่อส่งคำสั่งต่อมา

MMSYSTEM.H ก็คือเป็นค่าคงที่พิเศษ MCI_ALL_DEVICE_ID จะเป็นตัวแสดงว่านั่นคือคำสั่งข้อความที่เป็นคำสั่ง ส่ง ทุก ๆ MCI ดีไวซ์ นั่นคือทุกแอปพลิเคชันจะถูกเปิด เราสามารถใช้ MCI_ALL_DEVICE_ID กับข้อความคำสั่งทุกๆตัว นั่นคือก็จะไม่มีการส่งข้อมูลกลับมา ยกตัวอย่างเช่น การปิดทุกๆ MCI ดีไวซ์ นั้นมีการเปิดโดยแอปพลิเคชัน

```
DWORD dwReturn;
/*
*ปิดทุกๆ MCI ดีไวซ์ เปิด โดยแอปพลิเคชันนี้.
*คอยจนกว่าทุกๆดีไวซ์จะปิดก่อนที่จะส่งค่ากลับ.
*/
if (dwReturn = mciSendCommand(MCI_ALL_DEVICE_ID, MCI_CLOSE,
MCI_WAIT, NULL))
/*เกิดการผิดพลาดขึ้น, ยกเลิกการปิดทุกๆ ดีไวซ์*/
return FALSE;
```

2.3.10 ข้อความคำสั่งของ MCI ที่สำคัญ (Summary of MCI Command Messages)

MCI จะแบ่งแยกออกเป็นประเภทประเภทได้อยู่ 4 ประเภทของข้อความคำสั่ง ข้อความคำสั่งและทางเลือกใน 2 ประเภทเป็นการอธิบายถึง ชุดคำสั่งอย่างเล็ก ๆ สำหรับ ทุก ๆ MCI ไดรฟ์เวอร์ดังนี้

- ระบบข้อความคำสั่งเป็นเครื่องมือโดยตรงโดย MCI มากกว่าที่จะโดยไดรฟ์เวอร์
- ความต้องการข้อความคำสั่งเป็นเครื่องมือโดยไดรฟ์เวอร์ ทุก ๆ ไดรฟ์เวอร์จะเลือกเอาข้อความคำสั่งและทางเลือกที่เข้ากันได้

ข้อความคำสั่งในอีก 2 ประเภทที่ไม่ซัพพอร์ตกับทุก

- พื้นฐาน (หรือ ทางเลือก) ข้อความคำสั่ง เป็นการใช้โดยบางดีไวซ์ ถ้าดีไวซ์ซัพพอร์ตกับ ข้อความคำสั่งพื้นฐานมันต้องซัพพอร์ตกับชุดของทางเลือกสำหรับข้อความคำสั่ง
- ข้อความคำสั่งที่ขยายออกเป็นการกำหนดโดยแน่นอน ชนิดดีไวซ์ หรือ ไดรฟ์เวอร์ จะรวมเอาคำสั่งใหม่ๆเข้ามาด้วย

MCI โดยตรงการทำงานจะตามระบบข้อความคำสั่ง จะเร็วกว่าที่จะผ่าน MCI ดีไวซ์ไดรฟ์เวอร์

คำสั่ง

คำอธิบาย

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ บริษัท ไมโครซอฟท์ จำกัด คำอธิบายนี้เป็นเพียงเอกสารตัวอย่างเท่านั้น ไม่ควรนำมาใช้เพื่อการค้า

MCI_BREAK สำหรับการแจ้ง ตั้งค่าหยุดของคีย์สำหรับ MCI ดีไวซ์ ให้นำไปใช้ประโยชน์ด้านการค้า

MCI_SYSINFO สำหรับให้ข้อมูลเกี่ยวกับ MCI ดีไวซ์กลับ สารทุกครั้งที่มีการนำไปใช้

ความต้องการ ข้อความคำสั่งต้องเข้ากันได้กับ ทั้งหมดของ MCI ดีไวซ์ไดรวเวอร์ ตาม ตารางแสดงรายละเอียดความต้องการของ ข้อความคำสั่ง

คำสั่ง	คำอธิบาย
MCI_CLOSE	ปิดดีไวซ์
MCI_GETDEVCAPS	บรรจุความสามารถของดีไวซ์
MCI_INFO	บรรจุข้อมูลต้นฉบับจากดีไวซ์
MCI_OPEN	ขั้นแรกของดีไวซ์
MCI_STATUS	บรรจุข้อมูลสถานะภาพจากดีไวซ์

ตามข้อความคำสั่งพื้นฐาน จะใช้ข้อความคำสั่งโดยดีไวซ์เป็นทางเลือก

คำสั่ง	คำอธิบาย
MCI_LOAD	โหลดข้อมูลจากไฟล์
MCI_PAUSE	หยุดการเล่น
MCI_PLAY	เริ่มส่งข้อมูลไปยังเอาต์พุต(output)
MCI_RECORD	เริ่มบันทึกข้อมูล
MCI_RESUME	กลับมาเล่น หรือ บันทึกบนดีไวซ์ของการหยุดชั่วคราว
MCI_SAVE	บันทึกข้อมูลลงไฟล์
MCI_SEEK	เลื่อนไปหน้า หรือ กลับหลัง
MCI_SET	ตั้งค่าปฏิบัติการของดีไวซ์
MCI_STATUS	บรรจุข้อมูลสถานะที่เกี่ยวกับ ดีไวซ์
MCI_STOP	หยุดการเล่น หรือ บันทึก

ตามส่วนขยายข้อความคำสั่ง ที่สำคัญของชนิดดีไวซ์

คำสั่ง	ชนิดดีไวซ์	คำอธิบาย
MCI_CUE	waveaudio	เตรียมพร้อมสำหรับเล่นหรือบันทึก
MCI_DELETE	waveaudio	ลบข้อมูลเซกเมนต์จากมีเดียอิลิเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 พื้นฐานโทรศัพท์

ระบบโทรศัพท์ถือกำเนิดขึ้นโดยการคิดค้นของอเล็กซานเดอร์ เกรแฮมเบลล์ โทรศัพท์ระบบแรกที่เกิดขึ้นคือ ระบบโทรศัพท์แบบพัลส์ (pulse) การเรียกหมายเลขต่างๆ จะใช้สัญญาณพัลส์เป็นตัวกำเนิดทั้งสิ้นเช่น เลข 0 ก็คือสัญญาณพัลส์ 10 ลูก เลข 9 คือ สัญญาณพัลส์ 9 ลูก เป็นต้น

โทรศัพท์ คือ เครื่องมือสื่อสาร เชื่อมโยงที่นำเสียงพูดระหว่างผู้ใช้ที่อยู่ ณ สถานที่แห่งหนึ่งกับบุคคลที่ต้องการติดต่อกับ ณ สถานที่อีกแห่งหนึ่ง ให้สามารถพูดจาติดต่อกันได้เหมือนบุคคลทั้งสองนั่งสนทนาอยู่ด้วยกัน

ระบบโทรศัพท์มีส่วนประกอบที่สำคัญ 3 ส่วน คือ

1. เครื่องรับโทรศัพท์ เครื่องรับโทรศัพท์เป็นอุปกรณ์ที่ผู้ใช้จะใช้ในการติดต่อระหว่างกัน ประกอบด้วย เครื่องส่ง (Transmitter) เครื่องรับ (Receiver) กระดิ่ง (Ringer) ฮุกสวิทช์ (Hook switch) หน้ำปัทม์ สำหรับหมุนหรือกดหมายเลข

2. สายโทรศัพท์ เครื่องรับโทรศัพท์แต่ละเครื่อง จะมีสายโทรศัพท์ 1 คู่ เพื่อเชื่อมโยงและเป็นสื่อนำสัญญาณต่างๆ จากชุมสายมายังตัวเครื่องรับโทรศัพท์ในขนาดเดียวกัน ซึ่งทำหน้าที่เป็นสื่อนำในการส่งสัญญาณไฟฟ้าแปลงมาจากสัญญาณเสียงระหว่างเครื่องรับโทรศัพท์

สายโทรศัพท์ ที่ต่อเชื่อมโยงระหว่างชุมสายเพื่อให้บริการระหว่างชุมสาย จะเรียกว่า ทริงก์ (Trunk)

3. ชุมสายโทรศัพท์ ชุมสายโทรศัพท์ เป็นสถานที่ที่รวมคู่สายของเครื่องรับโทรศัพท์แต่ละเครื่องในพื้นที่ใกล้เคียงกันและทำหน้าที่เชื่อมคู่สายให้กับผู้ใช้โทรศัพท์ พร้อมกับส่งสัญญาณแจ้งภาวะการใช้ต่างๆ ให้ผู้ใช้ทราบ ชุมสายโทรศัพท์ระบบเก่าจะเป็นระบบใช้พนักงานต่อ (Manual Telephone) ซึ่งในปัจจุบันได้รับการพัฒนามาเป็นระบบอัตโนมัติ (Automatic Telephone) คือ จะมีอุปกรณ์ที่ทำหน้าที่ต่อสลับคู่สายระหว่างสมาชิกผู้เรียก กับผู้รับโดยอัตโนมัติ

2.4.1 ระบบสัญญาณต่างๆ ทางโทรศัพท์

ขั้นตอนการทำงานของโทรศัพท์แบ่งออกเป็น สองกรณี คือ ผู้เรียก (CALLING) กับผู้ถูกเรียก (CALLED)

กรณีผู้เรียก (CALLING SUBSCRIBER)

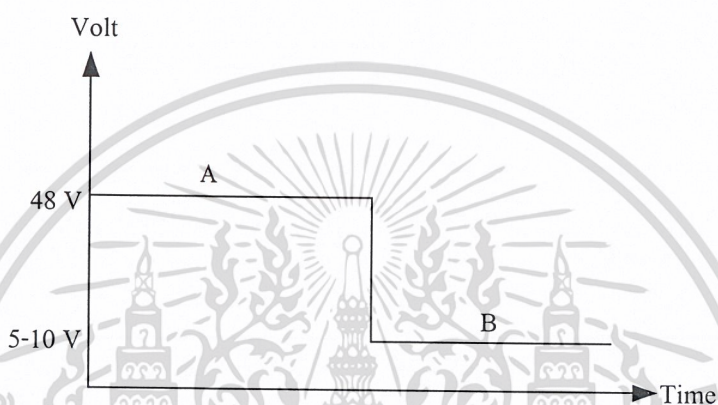
ขณะที่หูโทรศัพท์วางไว้นั้นจะมี ไฟ กระแสตรงตกคร่อม คู่สายโทรศัพท์ อยู่ +48 โวลต์และเมื่อหูโทรศัพท์ถูกยกขึ้นไฟกระแสตรงที่ตกคร่อมคู่สายโทรศัพท์ +48 โวลต์ จะตกลงมาเหลือ 5-10 โวลต์ ทั้งนี้ขึ้นอยู่กับระบบชุมสายย่อย ขณะเดียวกันนั้น ก็จะมีสัญญาณส่งมาจากชุมสายเสียงที่เราได้ยินคือสัญญาณ DIAL TONE แสดงว่าพร้อมที่จะหมุนเลขหมายได้หรือพร้อมที่จะกดเลขหมายได้ ถ้าเลขหมายที่ถูกเรียกไม่ว่าง ผู้เรียกจะได้ยินเสียง BUSY TONE ในกรณีที่คู่สายเลขหมายจะถูกเรียกว่างคือ ชุมสายจะต่อหมายเลขที่เรียกให้ได้ยินเสียงสัญญาณ RING BACK TONE หรือ RINGING TONE แสดงว่าเลขหมายที่เรียก

ไปพร้อมจะพูดได้ให้คอยจนกว่าผู้ถูกเรียกจะยกหูรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีผู้ถูกเรียก (CALLED SUBSCRIBER)

ขณะที่คู่สายว่างจะมีไฟกระแสตรงคร่อมคู่สาย +48 โวลต์ และเมื่อมีการเรียกเลขหมายทางชุมสายต่อไปก็จะส่งสัญญาณเรียก RINGING เป็นแรงดันไฟสลับประมาณ 110-150 โวลต์ และเมื่อมีการยกหูโทรศัพท์ทำให้วงจรภายในของเครื่องรับโทรศัพท์ที่มีอินพีเดนซ์ ประมาณ 600 โอห์ม ต่อเข้ากับชุมสายในขณะเดียวกันชุมสายจะหยุดส่งสัญญาณ RINGING และทำการต่อคู่สายโทรศัพท์ให้



รูปที่ 2.3 แสดงไฟกระแสตรงเลี้ยงคู่สาย

- ช่วง A ขณะที่วางคู่สายโทรศัพท์ที่อยู่มีไฟ DC ตกรวม +48 โวลต์
- ช่วง B ขณะที่ผู้โทรศัพท์ถูกยกขึ้นมีไฟ DC ตกรวม 5 - 10 โวลต์

2.4.2 ความถี่สัญญาณต่าง ๆ ที่เป็นมาตรฐานขององค์การโทรศัพท์

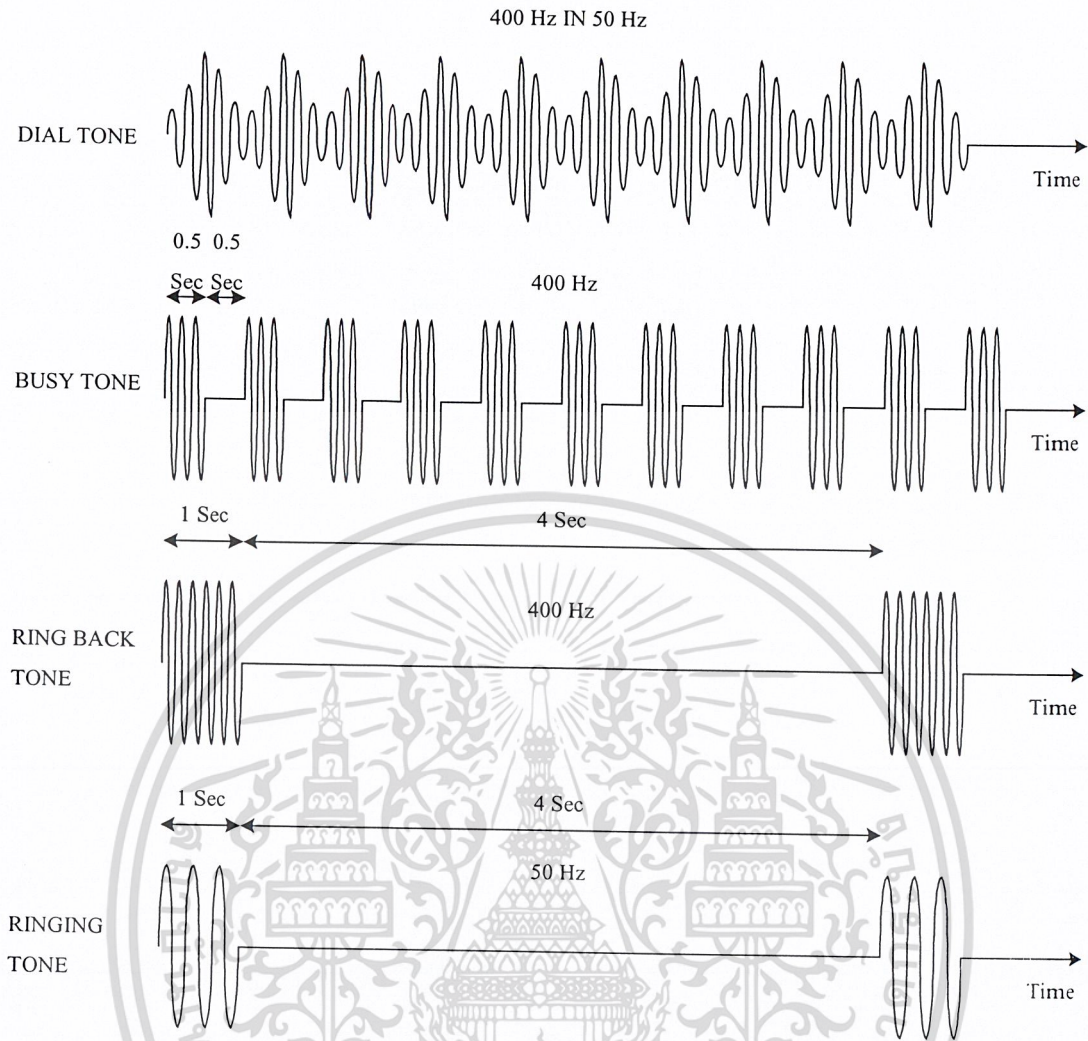
- DIAL TONE ใช้เพื่อแสดงให้ผู้เช่าทราบว่าจะขณะนี้ผู้เช่าสามารถที่จะเรียกไปยังหมายเลขอื่นได้ ลักษณะสัญญาณเป็นดังรูปที่ 2.4

- BUSY TONE ใช้เพื่อให้ผู้เรียกทราบว่าโทรศัพท์หมายเลขที่ต้องการติดต่อกับขณะนี้ยังไม่ว่าง ควรจะวางหูสักระยะหนึ่งก่อนจึงเรียกหมายเลขใหม่อีกทีลักษณะสัญญาณเป็นดังรูปที่ 2.4

- RINGING BACK TONE ใช้แสดงให้ผู้เรียกทราบว่าสามารถที่จะติดต่อกับผู้ที่ติดต่อก่อนหน้านี้ได้แล้ว แต่อยู่ระหว่างรอการยกหู โดยลักษณะสัญญาณเป็นดังรูปที่ 2.4

-RINGING TONE ใช้พร้อมกับ RINGING BACK TONE เมื่อสัญญาณเรียกดังก็จะมีสัญญาณเรียกดังพร้อมๆ กัน แต่สัญญาณนี้ดังแรงมากเพื่อไปทำให้กระดิ่งในเครื่องโทรศัพท์ดัง โดยลักษณะสัญญาณเป็นดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงสัญญาณพื้นฐานของโทรศัพท์

หน้าที่ทั่วไปของสัญญาณที่ใช้กับสายโทรศัพท์ในปัจจุบันมีอยู่ 4 หน้าที่ คือ

1. การเตรียมพร้อม (Alerting)
2. การส่งที่อยู่ของข่าวดสาร (Transmitting address information)
3. การตรวจตรา (Supervising)
4. การส่งสัญญาณข่าวดสาร (Transmitting information Signaling)

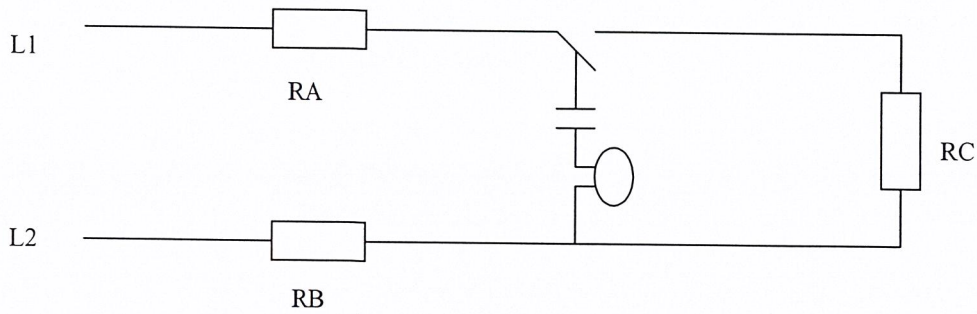
สัญญาณระหว่างผู้เช่ากับชุมสาย (Subscriber Signaling)

1. สัญญาณที่ส่งจากผู้เช่าไปยังชุมสาย

1.1 OFF HOOK คือ สภาวะผู้เช่ายกหู โทรศัพท์สายจะมีสภาวะ Closed Loop (Low Impedance)

1.2 ON HOOK คือ สภาวะผู้เช่าวางหู หรือสภาวะวาง สายจะมีสภาวะ Open Loop (High Impedance)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้เอาไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



RA, RB = ค่าความต้านทานของสายโทรศัพท์

RC = ค่าความต้านทานของเครื่องโทรศัพท์

1.3 Dialing คือ สภาพที่ผู้เช่าหมุนเลขหมายเข้าเครื่องเป็น Rotary Dial สัญญาณจะเป็น Pulsing ค่า Impedance จะสูงต่ำ สลับกันไปตามที่หมุนเลขหมาย ถ้าเครื่องเป็นแบบกดปุ่ม Touch Tone สัญญาณออกจะเป็นความถี่ DTMF ส่งออกไปชุมสาย

สัญญาณติดต่อระหว่างชุมสายกับชุมสาย (INTER EXCHANGE SIGNALING)

สัญญาณพื้นฐานมีอยู่ 5 ประเภท คือ

1. Seizure (สัญญาณจับวงจร) เป็นสัญญาณให้ชุมสายปลายทางทราบว่า คู่สายขณะนี้ถูกใช้งานอยู่ ชุมสายปลายทางจะทำการจัดเตรียมอุปกรณ์ที่รับเลขหมายของผู้ถูกเรียกที่จะส่งมา
2. Address information เป็นสัญญาณบอกเลขหมายหรือประเภทของผู้เช่า
3. Answer signal (สัญญาณตอบรับ) สัญญาณนี้ถูกส่งเมื่อผู้เช่าฝ่าย B ยกหูรับ หน้าทีหลักของสัญญาณก็คือ
 - 3.1 เริ่มต้นคิดเงิน
 - 3.2 ส่งสัญญาณคิดเงิน
 - 3.3 ตัดวงจรจับเวลาการใช้อุปกรณ์
4. สัญญาณยกเลิกการต่อตรง (Clear Forward) จะถูกส่งเมื่อฝ่าย A วางหู ผลของสัญญาณนี้จะทำให้วางจรทางด้านปลายทางทำการยกเลิกการต่อวงจรต่าง ๆ
5. สัญญาณยกเลิกการตอบกลับ (Clear Back) จะถูกส่งเมื่อผู้เช่าฝ่าย B วางหู ผลของสัญญาณนี้จะทำให้ชุมสายต้นทางเริ่มการจับเวลา เมื่อเวลาผ่านไป 90-120 วินาที ชุมสายต้นทางจะยกเลิกการติดต่อมาพร้อมกับส่งสัญญาณ Clear Forward ออกไปเพื่อให้ชุมสายปลายทางยกเลิกเช่นกัน

2.4.3 เครื่องรับโทรศัพท์

เครื่องรับโทรศัพท์ (Telephone Set) จัดเป็นอุปกรณ์ปลายทางอย่างหนึ่ง ทำหน้าที่รับส่งสัญญาณเสียงพูดระหว่างเช่า (Subscriber) โดยทำหน้าที่แปลงสัญญาณเสียงเป็นสัญญาณไฟฟ้า ส่งไปในสายและในทางกลับกันก็เปลี่ยนพลังงานไฟฟ้า กลับมาเป็นพลังงานเสียง นอกจากนี้เครื่องโทรศัพท์ยังทำหน้าที่ต่อไปนี้

1. ทำหน้าที่ส่งสัญญาณเรียก ไปยังชุมสายท้องถิ่น (Local-Exchange), (Hook off)

2. ทำหน้าที่ส่งสัญญาณ Code ที่ใช้แทนหมายเลขผู้ถูกเรียก (B Subscriber)

3. ทำหน้าที่รับเสียงโทน (Tone) ที่ตอบรับจากชุมสาย ตลอดจนสัญญาณเรียก (Ringing Tone)

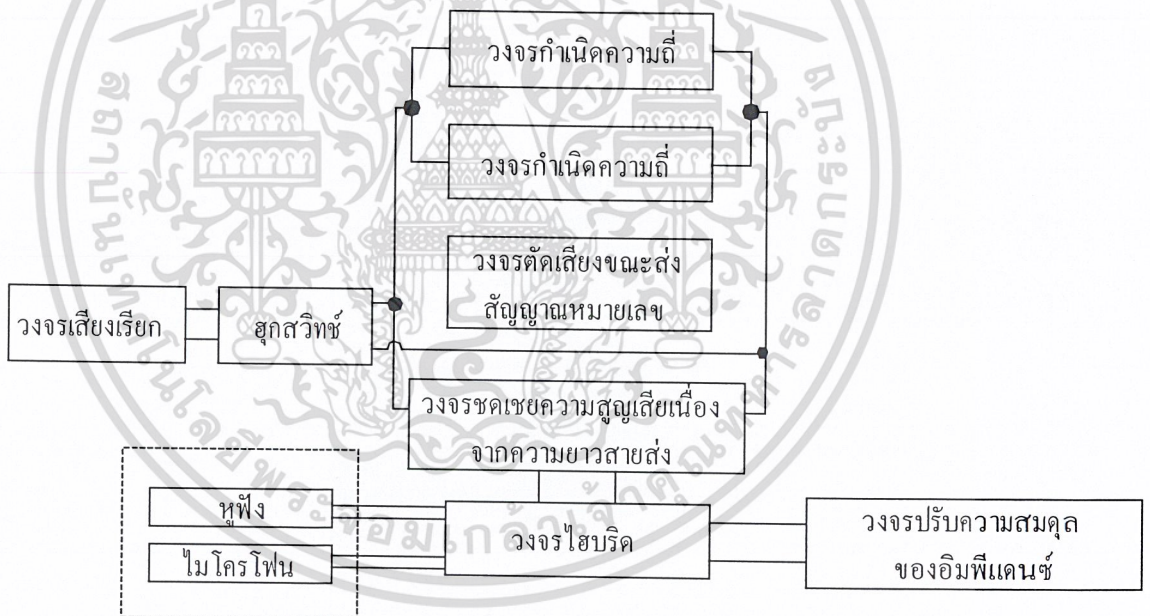
เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ พงษ์สัน อภิพงษ์ห้ามมิให้ตีแบบลงเนื้อหา และต้องอิงอิงลงใจ ใช้เอกสารทุกชิ้นนี้มาทำไปใช้

4. ทำหน้าที่ส่งสัญญาณยกเลิกการติดต่อเรียกไปยังชุมสาย (Hook on)

บล็อกไดอะแกรมการทำงานของโทรศัพท์

ในรูปที่ 2.5 บล็อกไดอะแกรมการทำงานของโทรศัพท์ ไม่ว่าจะป็นระบบพัลส์หรือโทนม็มีลักษณะเหมือนกัน แตกต่างกันตรงที่การกำเนิดสัญญาณเลขหมายว่าจะป็นระบบพัลส์หรือ DTMF วงจรเสียงเรียกทำหน้าที่แจ้งให้ผู้ใช้โทรศัพท์ทราบว่ามีการเรียกเข้ามา สุกสวิชเป็นตัวบอกให้ชุมสายโทรศัพท์รับรู้ว่ามีกรยกหูใช้งานโทรศัพท์แล้ว ก็จะทำการตัดต่อคู่สายให้ติดต่อกันได้ ในส่วนของวงจรตัดเสียงขณะทำการส่งเลขหมาย มีความชัดเจนถูกต้อง ไม่ถูกรบกวนด้วยสัญญาณเสียงพูด

ในขณะที่วงจรชดเชยความสูญเสีย เนื่องจากความยาวของสายส่งจะทำให้สัญญาณที่ติดต่อระหว่างต้นทางและปลายทางมีความแรงและชัดเจนมากที่สุด แม้ว่าต้นทางและปลายทางจะมีความห่างกันแค่ไหนก็ตาม วงจรไฮบริดจ์ทำหน้าที่เสมือนวงจรขยาย 2 ทิศทาง หรือสามารถให้สัญญาณผ่านเข้าออกได้ตลอดเวลา จึงมีเสียงจากปลายทางมาปรากฏที่หูฟัง ในขณะที่สัญญาณจากปากพูดก็จะผ่านออกไปทางคู่สายได้ สุดท้ายวงจรปรับความสมดุลย์ของอิมพีแดนซ์ มีไว้เพื่อทำให้อิมพีแดนซ์ของส่วนต่าง ๆ ในโทรศัพท์เหมาะสม เพื่อให้การถ่ายทอดสัญญาณเป็นไปได้อย่างมีประสิทธิภาพมากที่สุด



รูปที่ 2.5 บล็อกไดอะแกรมการทำงานของเครื่องโทรศัพท์

ส่วนประกอบหลักของเครื่องโทรศัพท์ แบ่งออกได้ 3 ส่วนดังนี้

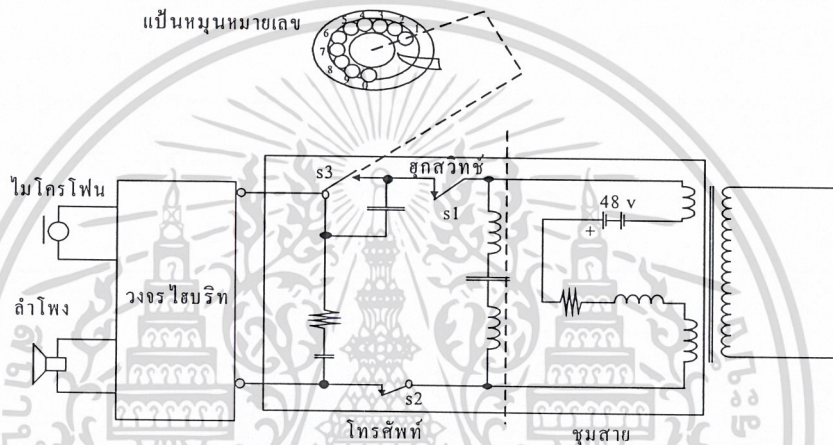
1. ส่วนรับส่งสัญญาณเสียงพูด (Speech transmission)
2. ส่วนกำเนิดสัญญาณเลขหมายของผู้เรียก (Generator Tone Code)
3. ส่วนที่รับสัญญาณเรียกจากชุมสาย (Ringing Tone)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้เครื่องโทรศัพท์ที่ขังแบ่งออกเป็นสองชนิด คือ

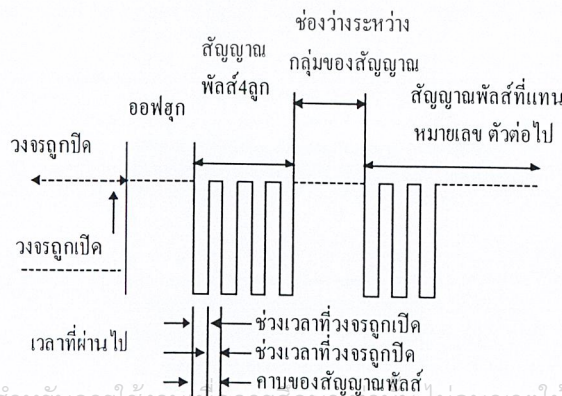
1. ระบบโทรศัพท์แบบพัลส์

จากรูปที่ 2.6 เป็นวงจรที่ใช้ส่งหมายเลขโทรศัพท์ในแบบพัลส์ เห็นได้ว่าสวิทช์ S3 จะถูกเปิดวงจรออกเมื่อมีการหมุนหมายเลขโทรศัพท์ เมื่อสวิทช์ S3 ถูกเปิดวงจรออกก็จะไม่มีกระแสไหลผ่านเข้าไปในวงจรส่วนที่อยู่ถัดไปได้ จึงเสมือนว่าเป็น การขัดจังหวะ (interruption) การไหลของกระแส สำหรับจำนวนครั้งที่สวิทช์ S3 ถูกเปิดกระแสออกจะขึ้นอยู่กับระยะห่างของ แป้นหมุน (dialer) ที่ถูกหมุนไป กับตำแหน่งปกติในขณะที่ไม่มีการหมุนหมายเลขใดๆ เป็นต้นว่า ถ้าหมุนหมายเลข 4 สวิทช์ S3 ถูกทำให้เปิดออก 4 ครั้ง ซึ่งสวิทช์ S3 จะถูกเปิดวงจรในช่วงที่ปล่อยให้แป้นหมุนกลับสู่ตำแหน่งเดิมเท่านั้น ไม่ได้เกิดขึ้นในระหว่างที่ทำการหมุนหมายเลขอยู่



รูปที่ 2.6 แสดงวงจรหมุนหมายเลขแบบพัลส์อย่างง่าย

จากรูปที่ 2.7 จะแสดงถึงลักษณะของรูปสัญญาณเมื่อมีการหมุนหมายเลขโทรศัพท์ จากรูปนี้จะเห็นว่าในตอนแรก โทรศัพท์อยู่ในสภาวะ On hook คือ หูโทรศัพท์วางอยู่บนที่วางหูโทรศัพท์ตามปกติ ไม่มีกระแสจากชุมสายเข้าสู่โทรศัพท์ แต่เมื่อมีการยกหูโทรศัพท์ขึ้น โทรศัพท์จะอยู่ในสภาวะ off hook สวิทช์จะถูกเปิดวงจร ทำให้มีกระแสไหลครบวงจรได้ และเมื่อมีการหมุนหมายเลข 4 จะทำให้มีการเปิดวงจรออกด้วยสวิทช์ S3 จำนวน 4 ครั้ง ได้รูปสัญญาณตามที่จะเห็น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใด รูปที่ 2.7 แสดงไดอะแกรมของเวลาที่ก้าว ๆ ของสัญญาณที่เกิดจากการหมุนหมายเลข 4 การนำไปใช้

ในระบบโทรศัพท์แบบที่ส่งสัญญาณแบบพัลส์นี้ จะถูกกำหนดให้สามารถส่งสัญญาณในอัตรา 10 พัลส์ต่อวินาที (10 PPS, 10 pulse per sec) หรือ 20 พัลส์ต่อวินาที (20 PPS) และ เพื่อความเข้าใจที่ตรงกันในการพิจารณาสัญญาณที่เกิดขึ้น จึงควรทราบความหมายต่อไปนี้

- อัตราการส่งสัญญาณพัลส์ (pulse rate) = จำนวนพัลส์ที่ถูกส่งออกไปใน 1 วินาที = 1000/คาบเวลาของสัญญาณพัลส์ (เป็นมิลลิวินาที)

- เปอร์เซ็นต์ของการเปิดวงจร (percent break) = $100 * \text{อัตราส่วนการเปิดวงจร (break ratio)} = 100 * \text{ช่วงเวลาที่ยังถูกเปิดออก/คาบของสัญญาณพัลส์}$

- ช่วงเวลาระหว่างกลุ่มของสัญญาณ (interdict interval) ถูกกำหนดให้มีค่าอย่างต่ำ 700 มิลลิวินาที

2. ระบบโทรศัพท์แบบ DTMF

ในยุคแรกเริ่มของการใช้โทรศัพท์ เครื่องรับโทรศัพท์ที่ใช้จะเป็นแบบหน้าปัทม์หมุน ซึ่งการหมุนหมายเลขจะทำให้เกิดพัลส์ (Pulse) ของกระแสในจำนวนเท่ากับหมายเลขที่หมุน พัลส์ที่เกิดขึ้นจะถูกส่งไปยังชุมสายด้วยความเร็ว 10 พัลส์ ต่อ วินาที (Pulse per second, pps) หรือ 20 พัลส์ ต่อ วินาที

เนื่องจากโทรศัพท์ที่ใช้ระบบหน้าปัทม์หมุนสำหรับการติดต่อผ่านชุมสายไม่ค่อยจะอำนวยความสะดวกให้กับผู้ใช้โทรศัพท์เท่าใดนัก เพราะเป็นระบบเชิงกล ทำงานค่อนข้างช้า ดังนั้นจึงได้มีการคิดสร้างโทรศัพท์กดปุ่มขึ้น ระบบโทรศัพท์ชนิดกดปุ่มนี้เรียกว่า ระบบ DTMF (Dual Tone multi frequency) เนื่องจากการกดปุ่มหมายเลขแต่ละปุ่มบนหน้าปัทม์ เครื่องรับโทรศัพท์นั้นทำให้เกิดสัญญาณที่ประกอบขึ้นจาก 2 ความถี่ ส่งออกไปตามสายโทรศัพท์ไปยังชุมสาย เพื่อเรียกใช้ชุมสายรู้ว่าผู้ใช้โทรศัพท์ต้องการติดต่อกับโทรศัพท์เครื่องใดแทนการส่งพัลส์ของกระแสการหมุนหน้าปัทม์ของเครื่องรับโทรศัพท์หน้าปัทม์หมุน

ข้อดีของการใช้โทรศัพท์แบบกดปุ่ม

1. ลดเวลาในการเรียกหมายเลขลง
2. การเรียกเลขหมายทำได้ง่ายขึ้น
3. สามารถใช้วงจรทางโซลิตูดเทออิเล็กทรอนิกส์แทนอุปกรณ์เชิงกล
4. มีความผิดพลาดในการส่งหมายเลขน้อย
5. ใช้สัญญาณระบบความถี่ซึ่งสามารถส่งระหว่างสถานีได้และสามารถนำไปใช้งานได้หลายด้าน

สามารถทำการเพิ่มปุ่มได้อีก 4 ปุ่ม ในคอลัมน์ที่ 4 เพื่อการใช้งานอย่างอื่น

สัญญาณของระบบ DTMF

ระบบ DTMF นี้จะมีความถี่มาตรฐานในย่านความถี่เสียงที่แตกต่างกัน 8 ความถี่ โดยจะแบ่งความถี่เหล่านี้ออกเป็น 2 กลุ่มคือ กลุ่มความถี่ต่ำ 4 ความถี่ และกลุ่มความถี่สูง 4 ความถี่ และสัญญาณ DTMF จะมาจากรวมสัญญาณความถี่จากกลุ่มความถี่ต่ำ 1 ความถี่และสัญญาณความถี่จากกลุ่มความถี่สูงอีก 1 ความถี่ ดังนั้นสัญญาณ DTMF จึงมีได้ทั้งหมด 16 สัญญาณ (4 ความถี่ต่ำ x 4 ความถี่สูง)

การเลือกความถี่มาตรฐานของระบบ DTMF นี้ ผู้ออกแบบระบบได้ใช้ความพยายามอย่างมากในการเลือกความถี่ที่จะใช้เพราะเนื่องจากต้องระวังไม่ให้สัญญาณเสียงต่างๆ ที่เกิดขึ้นในสายโทรศัพท์ เช่น สัญญาณแฉ่งภาวะการใช้งานต่าง ๆ สัญญาณรบกวนภายในสายโทรศัพท์ที่มีความถี่อยู่ในช่วงความถี่

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น มิใช่ผู้จัดทำหรือเจ้าของลิขสิทธิ์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DTMF และยังคงต้องระวังความถี่ที่อาจเกิดขึ้นจากการรวมตัวกันแบบฮาร์โมนิกของความถี่ใดความถี่หนึ่งของความถี่ DTMF และในที่สุดก็ได้ความถี่มาตรฐานทั้ง 8 ดังนี้

1. ความถี่มาตรฐานในกลุ่มความถี่ต่ำ 4 ความถี่ คือ 697, 770, 852 และ 941 เฮิรตซ์
2. ความถี่มาตรฐานในกลุ่มความถี่สูง 4 ความถี่ คือ 1209, 1336, 1477 และ 1633 เฮิรตซ์

ระบบ DTMF นี้เรียกอีกอย่างหนึ่งว่า ระบบ 4x4 เนื่องจากการใช้เป็นกชขนาด 4 x 4 ในการสร้างสัญญาณ DTMF และได้กำหนดปุ่มแต่ละปุ่มเหล่านั้นได้ด้วยตัวเลข 0-9, * (star หรือ saterisk), # (pound หรือ octophorpe), A, B, C และ D ซึ่งในการกดปุ่มๆ หนึ่งจะให้สัญญาณความถี่คู่หนึ่งออกมาดังแสดงไว้ดังตาราง แสดงค่าความถี่

ในการใช้งานทั่วไป จะใช้เฉพาะปุ่มตัวเลข 0-9 เท่านั้น ส่วนปุ่ม *, #, A, B, C และ D สามารถนำมาใช้งานอื่น ๆ ได้

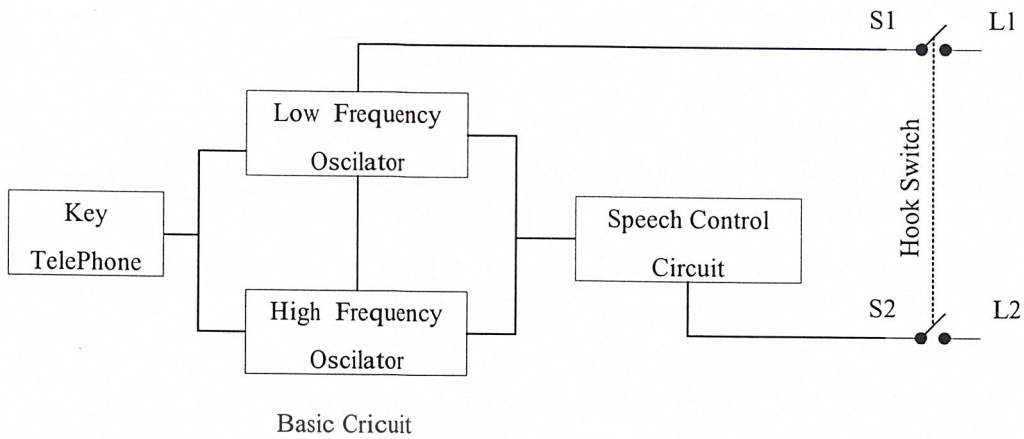
โทรศัพท์ที่ใช้ระบบ DTMF ผู้ใช้สามารถฟังเสียงสัญญาณ DTMF ได้โดยการยกหูโทรศัพท์ขึ้นฟังแล้วกดปุ่มใดปุ่มหนึ่งบนหน้าปัทม์ ตัวอย่างเช่น การกดปุ่มหมายเลข 8 จะเกิดสัญญาณความถี่ 852 เฮิรตซ์ และ 1336 เฮิรตซ์ ขึ้นพร้อมกัน สัญญาณจะถูกส่งผ่านคู่สายไปยังชุมสายและถูกถอดรหัสโดยตัวรับ DTMF ที่ชุมสายโทรศัพท์จะประกอบด้วยอุปกรณ์ควบคุมที่ทำหน้าที่ในการจัดการติดต่อโทรศัพท์ภายในท้องถิ่น ในบริเวณหนึ่งภายในพื้นที่หนึ่ง ๆ ซึ่งถูกกำหนดด้วยเลข 3 ตัวแรกของเลขหมายหลังจากที่ชุมสายทำการเชื่อมต่อคู่สายระหว่างผู้เรียกและผู้ถูกเรียกเรียบร้อยแล้ว ตัวรับสัญญาณ DTMF ของชุมสายจะหยุดทำงาน เพราะในการกดปุ่มบนหน้าปัทม์ โทรศัพท์ ครั้งต่อไปจะเป็นการติดต่อกันโดยตรงระหว่างผู้เรียกและผู้ถูกเรียก

ความถี่ (Hz)	รหัสหรือหมายเลข			
679	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D
ความถี่ (Hz)	1290	1336	1447	1633

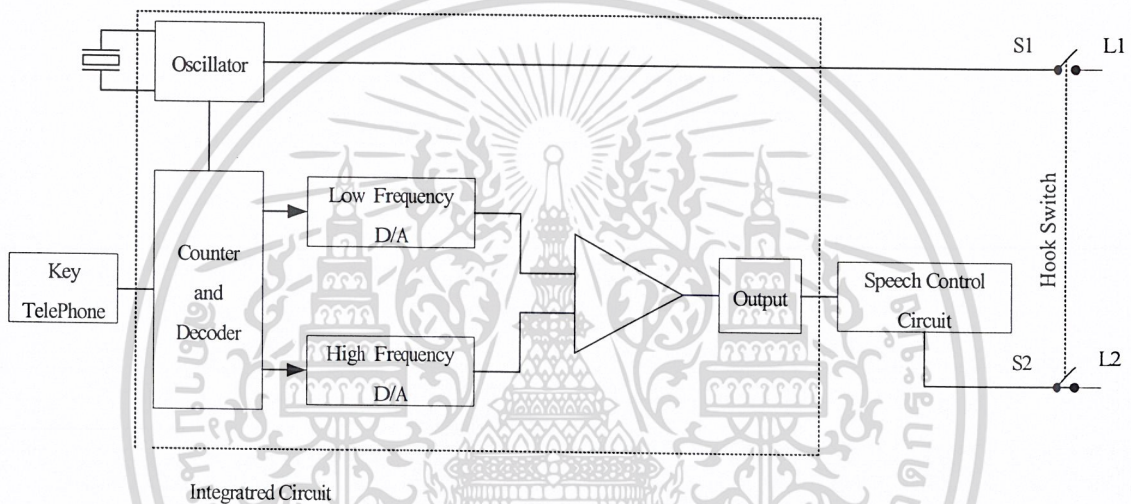
ตารางที่ 2.8 แสดงค่าความถี่ต่างๆ ที่ใช้ในระบบ DTMF

การส่งสัญญาณแบบ DTMF ด้วยการใช้อิซีสสำเร็จรูปในปัจจุบันจะเหมาะสมกว่าการนำอุปกรณ์มาต่อกันเพื่อทำการผลิตความถี่ต่างๆ ของเลขหมายที่ต้องการจะติดต่อ หลักการของอิซีสจำพวกนี้ก็ใช้หลักการเดิม คือ การนำความถี่ที่แตกต่างกัน 2 ความถี่เกิดจากการกดปุ่มหมายเลขโทรศัพท์มาอดดูเลทกัน แล้วส่งไปยังชุมสายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 (ก) ระบบโทรศัพท์แบบ DTMF

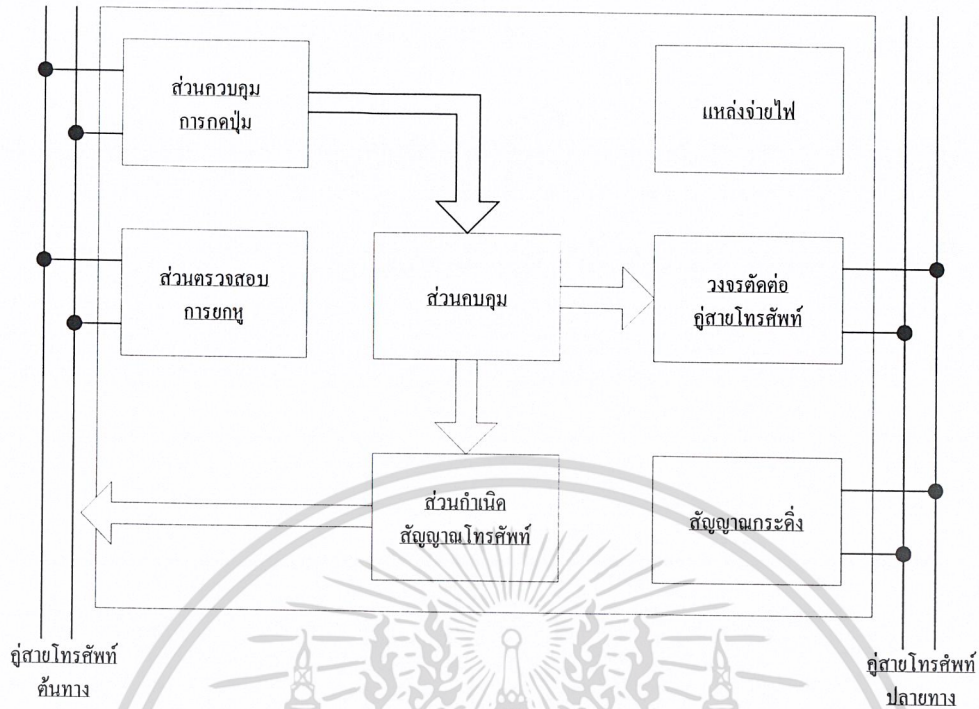


รูปที่ 2.8 (ข) ระบบโทรศัพท์แบบ DTMF

ในรูปที่ 2.8 ก เป็นบล็อกไดอะแกรมของการส่งสัญญาณแบบ DTMF ซึ่งในระบบนี้ยังคงต้องใช้ อุปกรณ์พาสซีฟ (passive elements) ในการวงจรออสซิลเลเตอร์ ซึ่งแน่นอนปัญหาที่จะพบในวงจรที่ใช้ อุปกรณ์เหล่านี้ คือ ความคลาดเคลื่อนเนื่องจากสถานะแวดล้อมที่เปลี่ยนแปลงไปตลอดจนอายุการใช้งาน ซึ่งผลที่ตามมาคือความถี่ที่ผลิตออกมาย่อมเปลี่ยนแปลงความถี่ไป ด้วยผลสุดท้ายก็จะทำให้ชุมสายเกิดการ ทำงานผิดพลาดในการติดต่อกันผู้ที่ถูกเรียกใช้ ดังนั้น การสร้างไอซีสำหรับรูปมาใช้งานแทนอุปกรณ์ พาสซีฟ ย่อมที่จะแก้ไขปัญหเหล่านี้ได้ในระดับหนึ่ง ใน รูปที่ 2.8 ข เป็นบล็อกไดอะแกรมของไอซีที่นำมา ใช้สร้างสัญญาณในระบบ DTMF ซึ่งวงจรภายในจะประกอบด้วย วงจรนับและถอดรหัส (counter and Decoder) ซึ่งก็จะแยกแยะว่าการกดหมายเลขแต่ละครั้งจะตรงกับตำแหน่งใดบ้างในแต่ละ แถวหรือแนว เมื่อทำการถอดรหัสจากการกดได้แล้ว จะนำค่าในแถวและแนวไปหารจากค่าความถี่หลัก สัญญาณที่ออก จากวงจรนับและถอดรหัสก็จะออกมาเป็นสัญญาณดิจิทัล 2 สัญญาณที่มีความถี่แตกต่างกัน หลังจากนั้นก็นำสัญญาณทั้ง 2 ไปผ่านวงจรแปลงสัญญาณดิจิทัลไปเป็นอะนาล็อก (D/A converter และนำมารวมกัน โดยการนำไปผ่านวงจรรวมและขยายสัญญาณแล้วจึงถูกส่งผ่านต่อไปยังวงจรควบคุมเสียงพูด (speech network) และผ่านต่อไปยังชุมขนสายโทรศัพท์ในที่สุด

เอกสารนี้เขียนขึ้นเพื่อใช้สำหรับเรียนการสอนในชั้นเรียนที่มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 ชุมสายโทรศัพท์

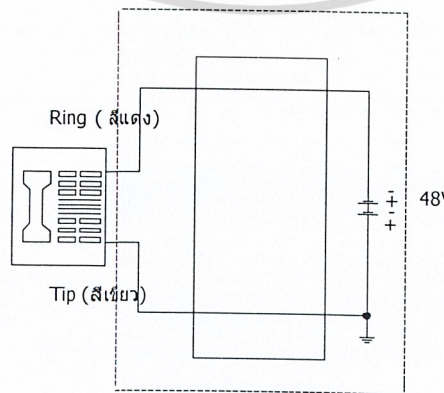


รูปที่ 2.9 บล็อกไดอะแกรมการทำงานพื้นฐานของชุมสายโทรศัพท์

ชุมสายโทรศัพท์ให้ต้นทางและปลายทางสามารถติดต่อพูดคุยกันได้จากระบบชุมสายใหญ่ๆ ก็มีการนำมาประยุกต์ใช้ในสำนักงาน เพื่อให้การติดต่อระหว่างแผนกต่างๆ ในสำนักงานคล่องตัว จากเดิมที่ใช้รีเลย์เป็นตัวติดต่อคู่สาย ทำให้ชุมสายมีขนาดใหญ่โตมาก ก็มีการพัฒนามาเป็นอิเล็กทรอนิกส์สวิตช์ ในรูปที่ 2.9 เป็นบล็อกไดอะแกรมของชุมสายโทรศัพท์อัตโนมัติ

Local Loop

ความหมายของ Local Loop คือ สายส่งสองสายจากเครื่องโทรศัพท์ไปชุมสาย ปลายทางและมีค่าอิมพีแดนซ์ของสายเองประมาณ 500-1000 โอห์ม แต่ค่าที่ใช้ทั่วไปคือ 600 โอห์ม ถ้าในชุมสายปลายทางมีการติดตั้งแหล่งจ่ายไฟร่วม DC ขนาด 48 โวลต์ ให้แต่ละรูปของผู้ใช้โทรศัพท์ ลวดตัวนำ 2 เส้นในรูป มีชื่อว่าทิป (Tip) และ ริง (Ring) โดยริงจะต่อกับสัญญาณไฟ 48 โวลต์ ดีซี ทิปจะต่อกับกราวด์ ดังรูปที่ 2.10

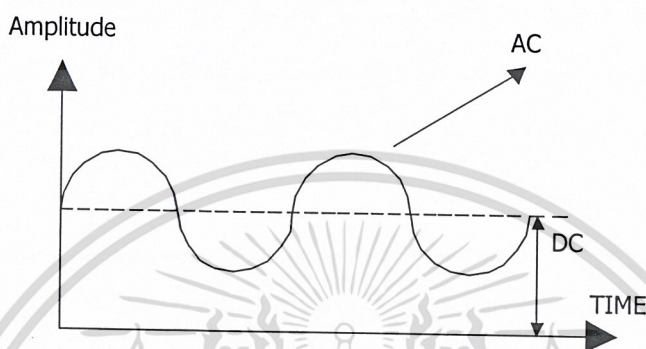


ชุมสายปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.10 แสดง โวลเคิลรูป
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้โทรศัพท์ยกหูโทรศัพท์ มีผลทำให้สวิตช์ปิดลง (Hook off) จากนั้นกระแสไฟตรงขนาด 20 มิลลิแอมป์ไหลวนอยู่ในรูปซึ่งสภาวะยกหูโทรศัพท์นี้ ระดับแรงดันไฟฟ้าระหว่างทีปกับริงมีค่าลดลงเหลือประมาณ 4 โวลต์

สัญญาณเสียงพูดจากเครื่องโทรศัพท์ถูกส่งไปในทิศทางหนึ่งในรูปแบบ โดยทำให้เกิดการเปลี่ยนแปลงเล็กน้อย ภายในกระแสรูป (20 mA) ซึ่งเกิดจากสัญญาณ AC ทับบนกระแสรูป DC ดังรูปแสดงส่วนประกอบของไฟกระแสตรงและกระแสลับ



รูปที่ 2.11 แสดงส่วนประกอบของไฟกระแสตรงและกระแสลับ

2.5 DTMF DECODER MT8870

MT8870 เป็นไอซีถอดรหัสความถี่ (INTEGRATED DTMF RECEIVER) หรือเป็นตัวแปลงสัญญาณ DTMF เป็นสัญญาณดิจิทัล ซึ่งมีคุณสมบัติดังนี้

- เป็นตัวรับและถอดรหัสความถี่ (DTMF receiver)
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทม์ (Guard time) ได้
- เป็นไอซีคุณภาพสูง

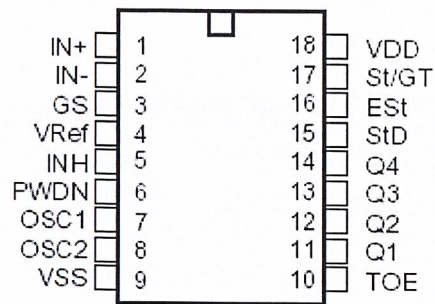
1. การนำ MT8870 ไปใช้งาน

- นำไปใช้งานด้านรีโมทคอนโทรล
- เครื่องป้องกันโทรศัพท์ทางไกล
- ใช้ในงานเกี่ยวกับเครดิตการ์ด
- ใช้งานร่วมกับคอมพิวเตอร์
- ใช้ในเครื่องชุมสายขนาดย่อมหรือ PABX
- ใช้กับงานทางด้านโทรศัพท์ทั่วไป
- เครื่องกันขโมย
- การควบคุมอุปกรณ์ทางโทรศัพท์
- ใช้ทำเครื่องสอบถามทางโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

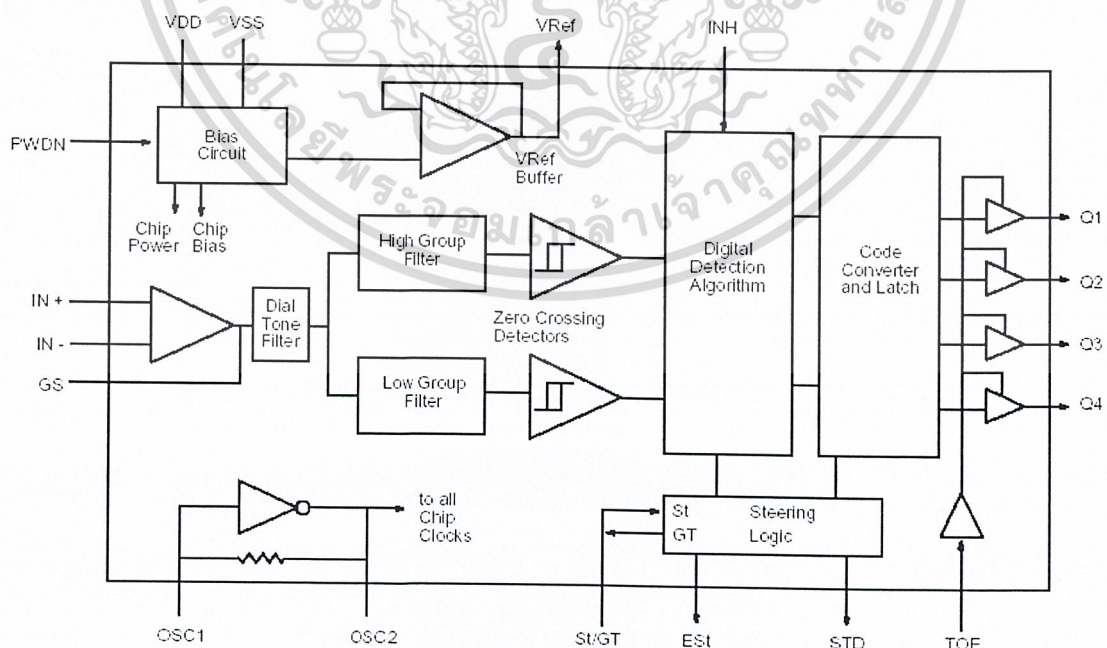
2. โครงสร้างของ MT8870



18 PIN CERDIP/PLASTIC DIP/SOIC

รูปที่ 2.12 แสดงรายละเอียดขาของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่ และวงจรถอดรหัสฟังก์ชันทางดิจิทัล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO²-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิตช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิทัลเพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิต และเช็คช่วงเวลาที่สำคัญเข้ามา ส่วนภาคอินพุทเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอกเอาท์พุทเป็นวงจรแลตช์ 3 สถานะ รูปที่ 2.13 แสดงขาของ MT8870 และรูปที่ แสดงโครงสร้างภายในของ MT8870



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.13 แสดงโครงสร้างภายในของ MT8870
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

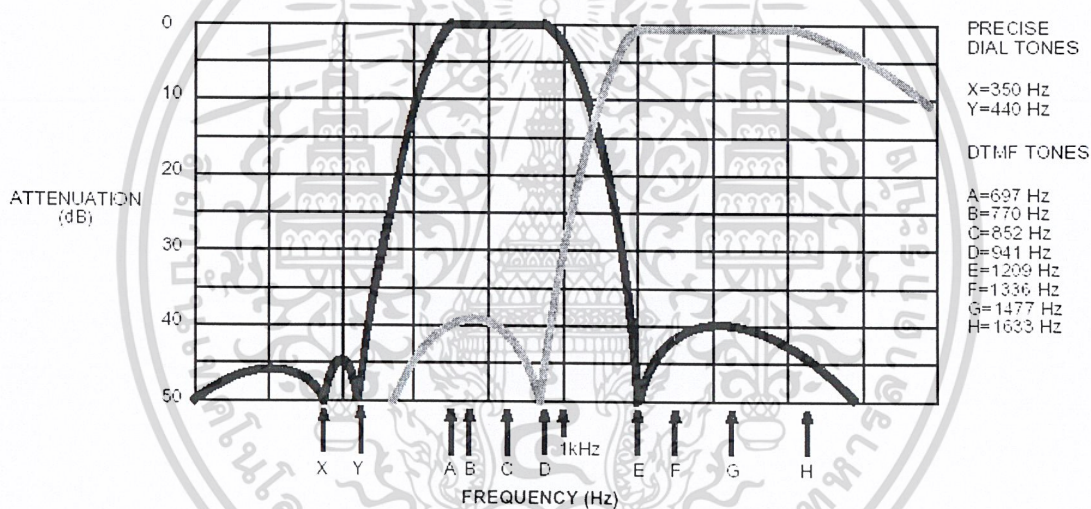
3. ฟังก์ชันการทำงานภายใน MT8870

โครงสร้างภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วนคือ

1. ภาคกรองความถี่ (Filter section)
2. ภาคถอดรหัส (Decoder section)
3. ภาคตรวจสอบสัญญาณ (Steering circuit)
4. ภาคขยายสัญญาณความแตกต่าง (Differential input)
5. ภาคกำเนิดความถี่ (Oscillator)

4. ภาคกรองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็นกลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิตซ์คาปาซิเตอร์ (six order switched capacitor band pass filter) ซึ่งความถี่ที่แยกได้นี้มี 2 ช่วง คือช่วงความถี่ต่ำ



รูปที่ 2.14 แสดงความถี่ที่ได้จากภาคกรองความถี่

5. ภาคถอดรหัส

ความถี่ DTMF ที่ถูกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการนับแบบดิจิทัล และมีการตรวจสอบความถี่เข้ามาเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสมเมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา Est (Early steering) ก็จะแอกทีฟสำหรับค่าถอดรหัสจากความถี่ต่าง ๆ นั้นแสดงในตารางที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digit	TOE	INH	EST	Q ₄	Q ₃	Q ₂	Q ₁
ANY	L	X	H	Z	Z	Z	Z
1	H	X	H	0	0	0	1
2	H	X	H	0	0	1	0
3	H	X	H	0	0	1	1
4	H	X	H	0	1	0	0
5	H	X	H	0	1	0	1
6	H	X	H	0	1	1	0
7	H	X	H	0	1	1	1
8	H	X	H	1	0	0	0
9	H	X	H	1	0	0	1
0	H	X	H	1	0	1	0
*	H	X	H	1	0	1	1
#	H	X	H	1	1	0	0
A	H	L	H	1	1	0	1
B	H	L	H	1	1	1	0
C	H	L	H	1	1	1	1
D	H	L	H	0	0	0	0

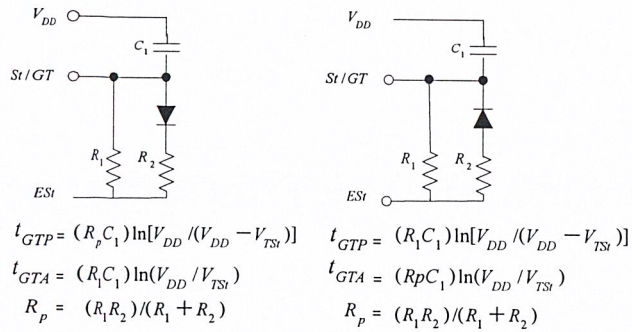
ตารางที่ 2.9 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ

6. ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควรมิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดนั้นสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา EST จะเป็น "HIGH" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 11 เมื่อขา EST เป็น "HIGH" ทำให้ V_e สูงขึ้น ตัวเก็บประจุ C จะคายประจุ ทำให้แรงดัน V_e สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัส จึงจะถอดรหัสออกมาเป็นตัวเลขขนาด 4 บิต รายละเอียดการทำงานขอให้ดูจากแผนภูมิเวลาหรือไทมิงไดอะแกรม (timing diagram) ในรูปที่ 2.19 จะเข้าใจได้ง่ายกว่า

สำหรับคำว่าการ์ดไทม์ (guard time) นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่เรารับไว้ จึงจะได้รับการยอมรับว่าสัญญาณความถี่นั้นถูกต้องหรือพูดได้ว่าเวลาที่เรารับไว้โดย RC ก็คือการ์ดไทม์นั่นเอง เมื่อสัญญาณความถี่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลาและคำนวณเวลาดูได้จากรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



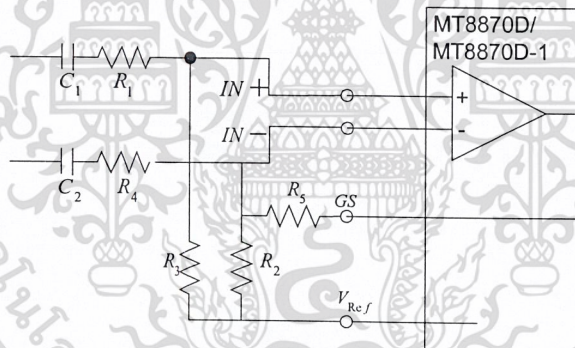
รูปที่ 2.15 แสดงวงจรตรวจสอบสัญญาณอย่างง่ายและแสดงการกำหนดเวลาคาร์ดใหม่ (guard time) พร้อมวิธีคำนวณ

7. ภาควิทยาสัญญาณความแตกต่าง

วงจรส่วนอินพุทของ MT8870 เป็นภาควิทยาสัญญาณความแตกต่างที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้าไปรูปที่ 2.16 แสดงการต่อวงจรภายนอกเข้ากับอินพุทซึ่งสามารถคำนวณอัตราขยายความแตกต่างของสัญญาณอินพุทและอิมพีแดนซ์ได้ดังนี้

อัตราขยาย ($A_v \text{ diff}$) = R_5 / R_1

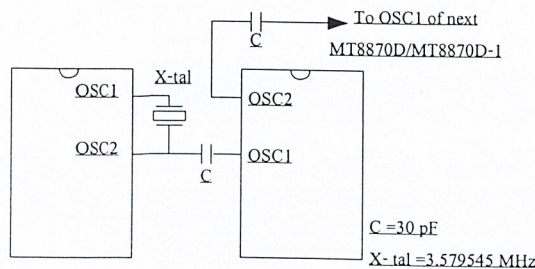
อินพุทอิมพีแดนซ์ ($Z_{in \text{ diff}}$) = $\sqrt{R_1^2 + (1/\omega C)^2}$



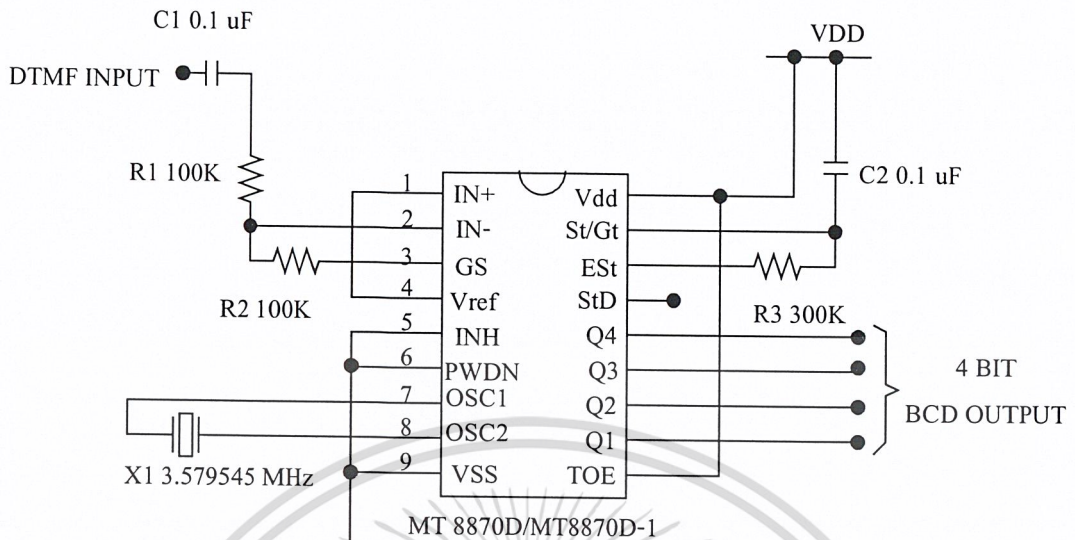
รูปที่ 2.16 แสดงการต่อวงจรภาคอินพุท

8. ภาควิทยากำเนิดความถี่

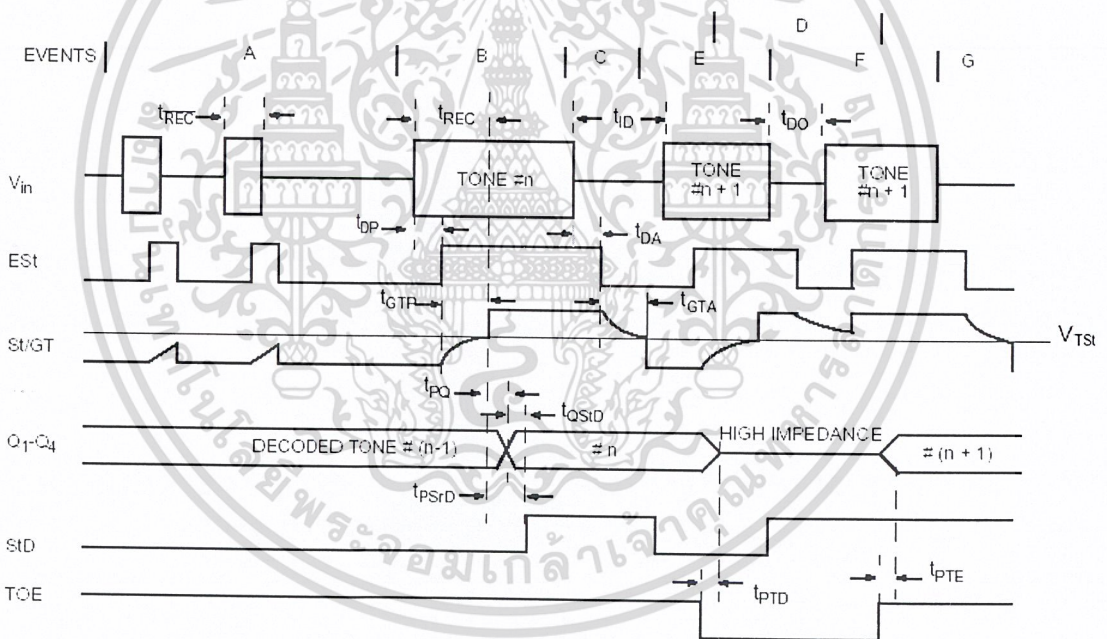
ในภาคนี้ภายในไอซีจะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.58 MHz ก็สามารถใช้งานได้ทันทีการต่อวงจรกำเนิดความถี่แสดงในรูปที่ 2.17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีรูปที่ 2.17 แสดงการต่อวงจรผลิตความถี่ของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงวงจรการใช้งานเบื้องต้นของ MT8870



รูปที่ 2.19 แสดงแผนภูมิเวลา (timing diagram) ของ MT8870

อธิบายขั้นตอนการทำงาน

A - ตรวจพบความถี่เข้ามา แต่คาบเวลา ไม่ถูกต้อง เอาท์พุทไม่เปลี่ยน

B - ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลตช์ไว้ที่

เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใด C - จับความถี่ # n ช่วงห่างถูกต้อง เอาท์พุทยังคงแลตช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่ไปใช้

D - เอาท์พุทเปลี่ยนเป็นไฮอิมพีแดนซ์

E - ความถี่ # n + 1 ถูกตรวจพบคาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแสดงไว้

F - ความถี่ # n + 1 หายไป ช่วงห่างไม่ถูกต้อง เอาท์พุทยังคงแสดงอยู่

G - จบความถี่ # n + 1 ช่วงห่างถูกต้อง เอาท์พุทยังคงแสดงอยู่จนถึงความถี่ใหม่ที่ถูกต้อง

อธิบายคำศัพท์

V_{in} - สัญญาณความถี่ DTMF ที่เข้ามา

Est - Early Steering Output ใช้แสดงความถี่ที่ถูกต้อง

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

Q₁-Q₄ - เอาท์พุท BCD ขนาด 4 บิต

StD - Delayed Steering output ใช้แสดงความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE - Tone Output Enable (input) ใช้ควบคุม Q₁-Q₄ ให้เป็นไฮอิมพีแดนซ์

t_{REC} - คาบเวลานานสุดที่ตรวจพบความถี่ DTMF แล้วยังไม่ถูกต้อง

t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้อง 2 สัญญาณ

t_{DD} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DTMF

t_{GTA} - การ์ดไทม์ของการหายไปของความถี่ DTMF

2.6 โทรสาร (Facsimile)

โทรสารเป็นเครื่องมือสื่อสารที่มีความสามารถในการส่งภาพจากต้นทางไปยังปลายทาง ก่อนที่จะมีการคิดค้นโทรพิมพ์ (Typewriter) โทรสารมีการพัฒนาอย่างต่อเนื่องแต่ก็มักจะมีประสบปัญหาด้านโครงสร้างภายในของระบบการสื่อสารสมัยก่อน ในช่วงปี ค.ศ. 1920 โทรสารมีความสำคัญอย่างยิ่งในด้านการข่าว โดยสำนักข่าวส่วนใหญ่มาใช้ในการส่งภาพข่าว

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโทรสารใช้ในการส่งเอกสารและรับเอกสาร ผ่านโครงข่ายการสื่อสาร โดยใช้เทคนิคของการส่งข้อมูลตัวอักษรในแบบเดียวกัน และมีความเหมาะสมกับการส่งเอกสารที่ประกอบด้วยตัวหนังสืออย่างเดียว ส่วนเทคนิคอื่นๆ ก็คือ การส่งภาพนิ่งของต้นฉบับในแบบอิเล็กทรอนิกส์ โดยจะมีแผ่นฟังบิตเป็นส่วนในการสร้างภาพนิ่งเพื่อใช้ในการสื่อสารโครงข่าย การบริการของโทรสารนี้จะรวมไปถึงตัวอักษร ภาพกราฟิก (Graphic) หรือ รูปภาพ

ระบบโทรสารจะมีข้อดีเหนือกว่าการส่งอื่นในรูปแบบของเอกสารหรือการส่งข้อความ การใช้งานของโทรสารจะช่วยประหยัดเวลาในการเตรียมเอกสารอีกด้วย เนื่องจากการส่งข่าวสารไม่จำเป็นต้องมาผ่านการส่งโดยการกดจากเป็นพิมพ์ แต่จะใช้ตัวต้นฉบับของเอกสารนั้นส่งไปยังปลายทางได้เลย โดยต้นฉบับที่เป็นเอกสารนี้จะถูกสแกน (scan) หาความสว่างและความมืดของเอกสาร แล้วจะแปลงเป็นสัญญาณอิเล็กทรอนิกส์เพื่อส่งไปยังปลายทางระบบโทรสารได้ถูกพัฒนาในปี ค.ศ 1843 แต่เพิ่งถูกนำมาใช้งานในตอนกลางปี ค.ศ 1970 และได้มีการเจริญอย่างรวดเร็ว

ได้มีการพัฒนาและการยอมรับมาตรฐาน ITU - T เกี่ยวกับโทรสาร ทำให้โทรสารสามารถทำงานร่วมกันระหว่างประเทศได้

เทคโนโลยีของโทรสารที่มีความก้าวหน้าขึ้นมี ความเร็วสูงขึ้น คุณภาพดีกว่าเดิม ขนาดของเครื่องเล็กลง ค่าใช้จ่ายของเครื่องโทรสารลดลง และมีการใช้เครื่องที่ง่ายขึ้น

การคิดค้นระบบโทรสารแบบดิจิทัลทำให้แต่ละหน้าของข่าวสาร สามารถดำเนินการได้อย่างง่ายโดยใช้คอมพิวเตอร์ ทำให้สามารถเก็บข้อมูลข่าวสารลงในดิสก์ (Disk) หรือเทปแม่เหล็ก โทรสารยังคงมีการพัฒนาเรื่อยมาจนปัจจุบันการพัฒนาของระบบโทรสารแบบดิจิทัลในแง่ของข่าวสารที่สามารถส่งระหว่างเครื่องโทรสารด้วยกันได้โดยตรงโดยไม่ต้องผ่านขั้นตอนบนกระดาษก่อน การส่งสามารถถูกเตรียมโดยการใช้โปรแกรมเขียนแบบเวิร์ดโปรเซสเซอร์ (Word Processor) ความสามารถในการส่งตัวหนังสือและกราฟิกพร้อมกันได้ ดังนี้ผู้ใช้สามารถแก้ไขตัวหนังสือได้ รวมทั้งรูปภาพกราฟิกต่าง ๆ ก่อนที่จะมีการส่งโทรสารออกไปยังปลายทาง และเนื่องจากโทรสารสามารถนำมาต่อเข้ากับเครือข่ายโทรศัพท์ได้ทันที มีการใช้งานได้สะดวกรวดเร็วทำให้ โทรสารเป็นอุปกรณ์สื่อสารที่มีการใช้กันอย่างแพร่หลายในสำนักงานและราคาที่ถูกลงอย่างมาก

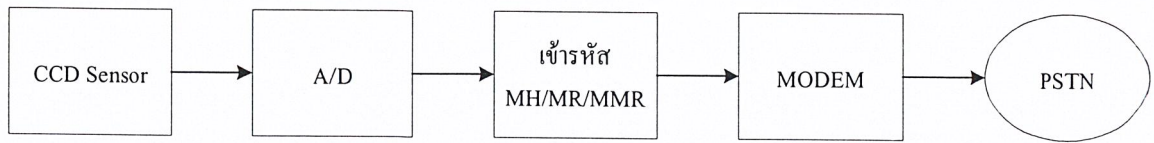
2.6.1 หลักการทำงานของโทรสาร

โดยทั่วไปโทรสารจะมีการทำงานโดยใช้ CCD (Charge Coupled Device) ในการสแกนภาพโดยจะทำการเปลี่ยนภาพเป็นภาพไฟฟ้าจากการตรวจความเข้มของพิกเซล (Pixel) แต่ละพิกเซลที่ละพิกเซล โดยในการสแกนใน 1 เส้น จะมีจำนวน 1728 พิกเซล (ตามมาตรฐาน ITU - T T.30)

หลังจากได้ข้อมูลจากการสแกนโดย CCDแล้วจะนำไปผ่านอุปกรณ์แปลงสัญญาณ แอนะล็อกเป็นดิจิทัลเพื่อนำเอาสัญญาณดิจิทัลที่ได้มาเข้ารหัสซึ่งสามารถทำได้ 3 รูปแบบ คือ HM (Modified Huffman), MR(Modified Reading), MMR(Modified Modified Reading) ซึ่งจะช่วยในการลดขนาดของข้อมูลและช่วยให้สามารถส่งข้อมูลได้เร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นจะส่งผ่าน โมเด็มซึ่งจะแปลงสัญญาณกลับไปเป็นแอนะล็อกอีกครั้งเพื่อส่งไปปลายทางโดยผ่านเครือข่ายโทรศัพท์



รูปที่ 2.20 หลักการทำงานของโทรสารด้านส่ง

ทางด้านรับจะนำข้อมูลที่ได้มาผ่านโมเด็มเพื่อทำการแปลงสัญญาณกลับมาเป็นดิจิทัลเพื่อนำไปถอดรหัสให้อยู่ในรูปพิกเซลขาวและ ดำ เพื่อให้อุปกรณ์การพิมพ์แสดงผลออกมา



รูปที่ 2.21 หลักการทำงานของโทรสารด้านรับ

2.6.2 การแบ่งกลุ่มของโทรสาร

ITU - T ได้แบ่งอุปกรณ์โทรสารสำหรับใช้ในโครงข่ายสาธารณะออกเป็น 4 แบบ

กลุ่มที่ 1 (Group 1 Facsimile) จะเป็นโทรสารในแบบแอนะล็อก ที่มีความเร็วในการส่งต่ำ การส่งเป็นแบบ FM (Frequency Modulation) ในการแบ่งแยกกระดับของสีเทาหลายๆ ระดับ ที่อยู่ระหว่างกลางของสีขาวกับสีดำ ซึ่งสามารถส่งเอกสารขนาด A4 ISO (210x297 มิลลิเมตร หรือ 8.25x11.7 นิ้ว) ที่ประมาณ 4 เส้น ต่อ มิลลิเมตร (100 เส้นต่อนิ้ว) ในเวลาประมาณ 6 นาที และมีข้อกำหนดต่างๆ ตามข้อกำหนด T.2 ของ ITU - T

กลุ่มที่ 2 (Group 2 Facsimile) เป็นโทรสารแบบแอนะล็อกที่มีการปรับปรุงขึ้นมาโดยใช้เทคนิคในการลดแบนด์วิดท์ลงเพื่อให้ได้ความเร็วเพิ่มขึ้นเป็น 2 เท่า หรือมากกว่าที่รายละเอียดของภาพเท่ากับในกลุ่มที่ 1 การส่งจะเป็นแบบ duo binary PM (Phase Modulation) และยังคงมีระดับของสีเทาต่างๆ ในระหว่างสีขาวกับสีดำ ต้องสามารถทำการส่งเอกสาร A4 ได้ภายในเวลา 3 นาที และมีข้อกำหนดต่างๆ ตามข้อกำหนดของ T.3 ของ ITU - T

กลุ่มที่ 3 (Group 3 Facsimile) เป็นโทรสารระบบแรกที่ใช้ระบบดิจิทัลระบบนี้จะจัดให้มีค่าเพียงขาวกับ ดำเท่านั้น โดยจะมีการสุ่มสัญญาณที่ความหนาแน่น 200 พิกเซลต่อนิ้ว ทางแนวนอนขวางของกระดาษ โดยจะมีสแกน 100 หรือ 200 เส้น ต่อหนึ่งนิ้วทางแนวตั้งของหน้ากระดาษ โทรสารกลุ่มที่ 3 จะใช้การเข้ารหัสดิจิทัลและมีการลดค่าเฉลี่ยของข่าวสารที่ซ้ำซ้อนกันในสัญญาณที่ได้จากการเข้ารหัสของข่าวสารก่อนที่จะทำการมอดูเลชัน และถ้ามีการใช้โมเด็มในการส่งสัญญาณผ่านโครงข่ายโทรศัพท์แอนะล็อก เวลาในการส่งจะเร็วขึ้นเป็น 3 เท่าหรือมากกว่าเมื่อเปรียบเทียบกับโทรสารในกลุ่มที่ 2 ต้องสามารถทำการส่งเอกสาร A4 ได้ภายในเวลา 1 นาที และมีข้อกำหนดต่างๆตามข้อกำหนด T.4 ของ ITU - T

กลุ่มที่ 4 (Group 4 Facsimile) เป็นโทรสารระบบดิจิทัล คำ/ขาว เช่นเดียวกับกลุ่มที่ 3 แต่จะถูกเน้นสำหรับใช้บนโครงข่ายดิจิทัลที่มีความเร็วสูงขึ้น ถึง 64 kbps และมีการจัดการเกี่ยวกับการตรวจสอบความผิดพลาดของข้อมูลที่ได้รับ ความละเอียดสามารถใช้ได้จาก 200 ถึง 400 พิกเซลต่อนิว การส่งในโทรสารกลุ่มที่ 4 จะใช้เวลาในการส่งลดลงเหลือเพียงไม่กี่วินาที มาตรฐานโทรสารกลุ่มที่ 4 เป็นการทำงานแบบดิจิทัลซึ่งใช้งานร่วมกับระบบ ISDN

โทรสารในกลุ่มที่ 1 และ 2 ได้มีการกำหนดมาตรฐานในปี 1976 ส่วนกลุ่มที่ 3 ทำขึ้นในปี 1980 ส่วนในกลุ่มที่ 4 ได้ตั้งขึ้นเมื่อปี 1984

ปัจจุบันมีการใช้งานโทรสารทั่วไปจะใช้ตามมาตรฐาน กลุ่มที่ 3 โดยสามารถให้การเข้ารหัสข้อมูลแบบ ดิจิตอล (มาตรฐานกลุ่ม 1 และ 2 มีระบบการทำงานแบบแอนะล็อก ส่วนมาตรฐานกลุ่ม 3 และ 4 มีระบบการทำงานแบบดิจิทัล) โดยตามมาตรฐานโทรสารกลุ่มที่ 3 จะมีความละเอียดในแนวตั้ง 3.85 เส้นต่อมิลลิเมตร และในแนวนอนจะมีความละเอียดในแนวนอน 1728 พิกเซลต่อความยาว 215 มิลลิเมตร เท่ากับ 8 พิกเซลต่อมิลลิเมตร อย่างไรก็ตามในมาตรฐาน T.4 ให้เพิ่มความละเอียดได้สูงถึง 15.4 เส้นต่อมิลลิเมตร ในแนวตั้งและ 16 พิกเซลต่อมิลลิเมตรในแนวนอน

2.6.3 คุณสมบัติทั่วไปของเครื่องโทรสารกลุ่ม 3

- ความละเอียด

โดยจะมีความละเอียดของภาพที่ได้จากการรับส่ง เท่ากับ 203 พิกเซล (pixel) ต่อนิว ในแนวนอน และ 98 พิกเซล ต่อนิว ในแนวตั้ง

- เวลาที่ใช้ในการส่ง

เครื่องโทรสารกลุ่ม 3 จะใช้เวลาในการส่งแต่ละหน้าไม่แน่นอน ขึ้นอยู่กับจำนวนของจุดคำและรายละเอียดของข้อมูลที่ส่ง แต่โดยทั่วไปจะส่งข้อมูลได้โดยเฉลี่ย 1 หน้า ในเวลา 10-30 วินาที การเพิ่มขนาดความจุของหน่วยความจำในเครื่องโทรสารกลุ่ม 3 ทำให้สามารถส่งได้เร็วขึ้น

- การใช้งานง่าย

ผู้ใช้งานสามารถป้อนต้นฉบับได้อย่างง่ายดาย และเพียงกดปุ่มเพื่อทำการส่งไปยังที่หมายตามต้องการ โดยที่เครื่องโทรสารกลุ่ม 3 สามารถเรียกหมายเลขปลายทางอัตโนมัติได้ และสามารถตอบรับสัญญาณเรียกเข้าได้โดยอัตโนมัติเช่นกัน หลังจากรับข้อมูลเรียบร้อยแล้ว เครื่องจะวางสายและยกเลิกการติดต่อจนกว่าจะมีการเรียกครั้งใหม่เข้ามา

- สามารถใช้กับคู่สายโทรศัพท์ทั่วไปได้

เครื่องโทรสารกลุ่ม 3 สามารถใช้กับสายโทรศัพท์ตามปกติได้ โดยเพียงแต่ต่อเครื่องโทรสารเข้ากับโทรศัพท์เท่านั้น ก็สามารถส่งข้อมูลไปยังปลายทางที่มีคู่สายโทรศัพท์ได้ตามต้องการ

- โมเด็ม(modem)

โมเด็ม ที่ใช้ในเครื่องโทรสารกลุ่ม 3 สามารถตรวจสอบสัญญาณการติดต่อระหว่างเครื่องโทรสารทั้ง 2 ข้างได้ และสามารถเลือกความเร็วที่เหมาะสมในการส่งข้อมูลนั้นๆ ได้โดยอัตโนมัติ โดยจะเลือกความเร็วสูงสุดที่เป็นไปได้ในการส่งข้อมูลระหว่างเครื่องโทรสารทั้ง 2 ด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น มิใช่เอกสารที่เผยแพร่อย่างเป็นทางการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงชื่อเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 มาตรฐานและคุณสมบัติของเครื่องโทรสารกลุ่ม 3

คุณสมบัติของเครื่องโทรสารกลุ่ม 3 ตามมาตรฐาน ITU – T แสดงดังต่อไปนี้

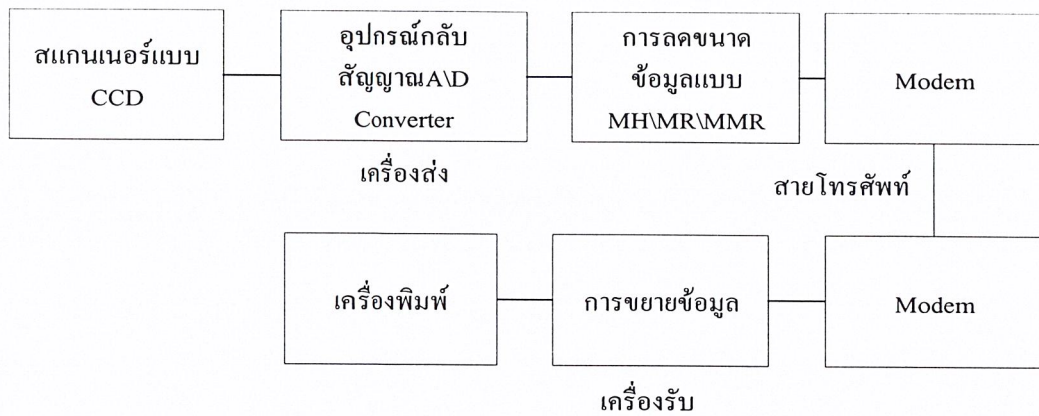
Item	Standard	Option	Small Copy(A5&A6)			
Scan Width			4.2	5.9	5.9	4.2
In	8.46	10 11.9	107	151	151	107
Mm	215	255 303	854	1216	1728	1728
Pels/line	1728	2048 2432	203	203	290	406
H\in	203		8	8	11.4	16
\mm	8		196/392	138/176	138/176	196/392
V\in	97.8	196	7.7/15.4	5.44/10.9	5.44/10.9	7.7/15.4
\mm	3.85	7.7				
Ms\line						
Coding	20	0,5,10,40				
	Modified	Modified Read, Modified-Modified Read				
Modem	Huffman					
Fax signal	V.27ter	V.29	V.27			
Bits\s	2400/4800	9600/7200	14400,1200,9600,7200			
Handshake						
Bits\s	V.21(Ch21)	V.27ter				
Error	300	2400				
Correction	None	Error Correction Mode				

ตารางที่ 2.10 แสดงคุณสมบัติของเครื่องโทรสารกลุ่ม 3 ตามมาตรฐาน ITU - T

2.6.5 รายละเอียดของเครื่องโทรสารกลุ่ม 3

โครงสร้างของเครื่องโทรสารกลุ่ม 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 โครงสร้างของเครื่องโทรสารกลุ่ม 3

โครงสร้างของเครื่องโทรสารกลุ่ม 3 ซึ่งแสดงดังบล็อกไดอะแกรมประกอบด้วย

สแกนเนอร์แบบ ซีซีดี (CCD: Charged Coupled Device) ทำหน้าที่ในการเปลี่ยนภาพให้เป็นไฟฟ้า โดยอาศัยโฟโตไดโอดเซนเซอร์ (photosensors) เป็นตัวตรวจจับความเข้มของจุดแต่ละจุดแล้วสร้างพัลส์ ขึ้นมาแทนจุดแต่ละจุดนั้น ซึ่งใน 1 เส้นจะทำการตรวจจับ 1728 จุด ทำให้ได้พัลส์จำนวน 1728 ลูกเช่นกัน

ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล จะทำการเปลี่ยนสัญญาณอนาล็อกให้เป็นดิจิทัล

ลดขนาดข้อมูล อาจจะเป็นแบบ โมดิไฟด์ ฮัฟฟ์แมน โมดิไฟด์ ริดจ์ หรือ โมดิไฟด์ ริดจ์ ซึ่ง จะทำการลดจำนวนบิตของข้อมูลให้น้อยลง โดยการเข้ารหัสตัวใหม่แทนข้อมูลเดิมทำให้มีขนาดเล็กลงกว่าเดิมช่วยให้สามารถส่งข้อมูลได้เร็วขึ้นด้วย

โมเด็ม จะทำการเปลี่ยนสัญญาณที่ถูกเข้ารหัสเป็นดิจิทัลเรียบร้อยแล้ว ให้เป็นสัญญาณอนาล็อกอีกครั้งหนึ่ง เพื่อให้สามารถส่งไปบนสายโทรศัพท์

ทางด้านรับจะมีโมเด็มทำการเปลี่ยนสัญญาณอนาล็อกที่รับมาจากทางด้านส่ง ให้เป็นสัญญาณดิจิทัล แล้วส่งเข้าไปยังส่วนขยายข้อมูล ซึ่ง จะทำการเปลี่ยนรหัสต่างๆ ให้อยู่ในรูปของจุดขาวดำ เพื่อให้เครื่องพิมพ์สามารถแสดงผลออกมาได้เหมือนต้นฉบับเดิม

2.7 การเข้ารหัสข้อมูล

ในเครื่องโทรสารกลุ่ม 3 กำหนดให้มีการเข้ารหัสเพื่อให้ข้อมูลที่ส่งผ่านระบบสื่อสารสัญญาณมีจำนวนน้อยลง ช่วยให้การส่งเร็วขึ้น การเข้ารหัสมีให้เลือก 2 แบบ คือ 1 มิตี และ 2 มิตี แต่จะแสดงรายละเอียด การเข้ารหัสแบบ 1 มิตี ซึ่งใช้หลักการเข้ารหัสแบบโมดิไฟด์ฮัฟฟ์แมน (Modified Huffman) ซึ่งข้อมูลที่มีค่าความน่าจะเป็นสูง (เกิดขึ้นบ่อย) จะถูกแทนด้วยรหัสที่มีความยาวน้อย ส่วนข้อมูลที่มีความน่าจะเป็นต่ำ (เกิดขึ้นน้อย) จะถูกแทนด้วยรหัสที่มีความยาวมาก ซึ่งในมาตรฐาน ITU - T T.4 ได้กำหนดรหัสที่ใช้ในการแทนมาอย่างแน่นอนแล้ว

ข้อกำหนดในการเข้ารหัสมีดังต่อไปนี้

1. การเข้ารหัสจะเริ่มต้นด้วยพิกเซลขาวก่อนเสมอ กรณีที่พิกเซลแรกเป็นดำ รหัสของพิกเซลขาวที่มีความยาวเท่ากับ 0 จะถูกส่งออกไปก่อน แล้วจึงตามด้วยพิกเซลดำ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อมูลพิกเซลขาวหรือพิกเซลดำสามารถมีความยาวได้ถึง 1728 ซึ่งเป็นความยาวสูงสุดสำหรับเส้นสแกนมาตรฐาน 1 เส้น

3. รหัสที่ใช้แทนข้อมูลมี 2 ชนิด คือ

-Terminating Code ใช้แทนพิกเซลที่มีความยาวตั้งแต่ 0 ถึง 63

-Make-up Code ใช้แทนค่าพิกเซลที่มีความยาวเป็นจำนวนเท่าของ 64 จนถึง 1728

4. กรณีของพิกเซลที่มีความยาวตั้งแต่ 64-1728 จะแทนด้วย Make-up Code ซึ่งมีค่าเท่ากับหรือน้อยกว่าความยาวขอพิกเซลก่อน โดยความยาวของพิกเซลที่เหลือจากการแทนด้วย Make-up Code จะมีค่าไม่เกิน 63 พิกเซล ซึ่งสามารถแทนค่าที่เหลือด้วย Terminating Code ได้

5. จุดสิ้นสุดของเส้นสแกน (EOF: End-Of-Line) เมื่อสิ้นสุดการสแกนข้อมูลครบ 1 เส้น ข้อมูลจะต้องตามด้วยรหัส EOF เพื่อแสดงจุดสิ้นสุดของเส้นสแกน นอกจากนี้รหัส EOF ยังใช้แนะนำเส้นสแกนเส้นแรกของแต่ละหน้าด้วย

6. บิตเติม (Fill) เป็นบิตที่เติมขึ้นระหว่างบิตของข้อมูลกับบิตสิ้นสุด เพื่อให้เวลาในการส่งข้อมูลในเส้นสแกนไม่ต่ำกว่าเวลาที่เครื่องพิมพ์ด้านรับสามารถทำงานได้ทัน

7. เมื่อจบการส่งข้อมูลแต่ละแผ่นจะต้องตามด้วยรหัส RTC (Return to Control) เพื่อเป็นการกลับเข้าสู่โหมดการควบคุมเครื่องโทรสาร

จำนวน Pixel 31 33 103 100 127 169 228 833

ภาพ 1 เส้น



สี/พิกเซล W31 + B33 + W64 + W39 B64 + B36 + W64 + W63 + B128 + B11 + W320 + W8 + B832 + B1

MH Code 000110010 000001101011 11011 00101000 0000001111 000011010100
11011 00110100 000011001000 0000101 00110110 10011 000000110100 010

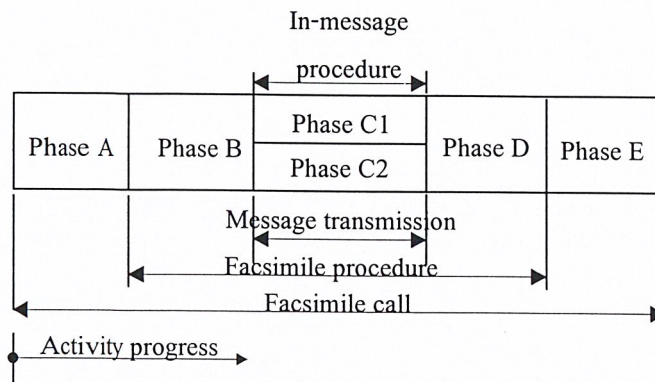
รูปที่ 2.23 แสดงตัวอย่างการเข้ารหัสข้อมูลโทรสาร

2.8 ขบวนการรับส่งข้อมูลของเครื่องโทรสารกลุ่มที่ 3

ตามมาตรฐานของ ITU - T T.30 ได้แบ่งขั้นตอนในการรับส่งข้อมูลของเครื่องโทรสารกลุ่มที่ 3 ออกเป็น 5 เฟส

- เฟส A ขั้นตอนการติดต่อ (Call establishment)
- เฟส B ขั้นตอนก่อนการส่งข่าวสาร (Pre- message)
- เฟส C ขั้นตอนการส่งข้อมูล (Message transmission)
- เฟส D ขั้นตอนหลังการส่งข่าวสาร (Post-message procedure)
- เฟส E ปลดสาย (Call release)

เอกสารนี้เป็นสำหรับขั้นตอนในการส่งโทรสารทั้ง 5 ขั้นตอนได้แสดงดังในรูปที่ 2.24
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 แสดงขบวนการในการรับส่งข้อมูลของเครื่องโทรสารกลุ่มที่ 3

- ช่วงการทำงานตั้งแต่เฟส A-E เรียกว่า แฟกซ์ไมล์ เซชชั่น (Facsimile session)
- ช่วงการทำงานตั้งแต่เฟส B-D เรียกว่า แฟกซ์ไมล์ โพรซีเจอร์ (Facsimile procedure)

เฟส A ขั้นตอนการติดต่อ (Call establishment)

หรือเรียกว่า การเริ่มต้นการเรียก เป็นขั้นตอนที่เชื่อมโยงเครื่องโทรสารต้นทางเข้ากับเครื่องโทรสารปลายทาง โดยผ่านทางสายโทรศัพท์ อาจเป็นแบบควบคุมโดยโอเปอเรเตอร์ ให้โอเปอเรเตอร์ติดต่อกันได้แล้วจึงทำการส่งข้อมูล หรืออาจเป็นแบบอัตโนมัติ คือรับหรือ ส่งเอกสารโดยอัตโนมัติที่ด้านใดด้านหนึ่งหรือสามารถรับอัตโนมัติและส่งอัตโนมัติได้ทั้งสองด้าน

ในขั้นแรกเครื่องโทรสารทางด้านผู้เรียกไปยังเครื่องโทรสารปลายทางที่ต้องการติดต่อกับสัญญาณ ไดออลโทน (Dial tone) และส่งสัญญาณ CNG (Calling tone) ไป เพื่อเริ่มการติดต่อส่วนทางด้านถูกเรียกจะตอบสนองต่อการเรียกนี้โดยส่งสัญญาณ CNG (Call station identification) ซึ่งเป็นสัญญาณความถี่ 2100 เฮิรตซ์ กลับไปยังเครื่องโทรสารด้านผู้เรียกทุกๆ 3 วินาที

เฟส B ขั้นตอนการส่งข่าวสาร (Pre-message procedure)

เป็นขั้นตอนที่เครื่องโทรสารต้นทางเชื่อมโยงกับเครื่องโทรสารปลายทางแล้วต้องทำการตรวจสอบคุณสมบัติและความสามารถในการรับและส่งให้ตรงกัน ก่อนที่จะเริ่มส่งข่าวสาร มี 2 ขั้นตอนคือ

1. ส่วนการตรวจสอบสัญญาณ ทำการส่งสัญญาณเพื่อบอกให้ทราบว่าเป็นเครื่องโทรสารอยู่ในกลุ่มใด (Group identification) พร้อมทั้งจะรับข้อมูลหรือไม่ (Confirmation for reception) หมายเลขโทรศัพท์ของเครื่องโทรสารต้นทาง (Subscriber identification) รวมถึงความสามารถอื่น ๆ ที่ไม่ใช่มาตรฐานของ ITU - T

2. ส่วนคำสั่ง ส่งสัญญาณเพื่อตรวจสอบระบบสื่อสารสัญญาณ DIS (Digital identification signal) ไปยังด้านผู้เรียกเพื่อแสดงถึงคุณสมบัติและความสามารถต่างๆ ของเครื่อง ซึ่งเมื่อทางด้านผู้เรียกได้รับสัญญาณนี้แล้วก็จะส่งสัญญาณ DSC (Digital command signal) มายังด้านผู้ถูกเรียกเพื่อแจ้งให้ทราบว่าได้เลือกใช้คุณสมบัติและความสามารถของเครื่องที่ถูกเรียกเป็นข้อมูลครั้งนี้ ซึ่งเครื่องโทรสารด้านถูกเรียกก็ต้องเลือกโหมดของตนเองให้เป็นไปตามข้อกำหนดนั้น จากนั้นเครื่องโทรสารด้านผู้เรียกจะส่งสัญญาณความเร็วสูงที่เรียกว่า เทรนนิ่ง (training) เพื่อตรวจสอบระบบสื่อสาร ซึ่งหลังจากเครื่องโทรสาร

ด้านถูกเรียกได้รับสัญญาณเทรนนิงและเมื่อตรวจสอบเรียบร้อยแล้ว ก็จะส่งสัญญาณ CFR (Confirmation to receive) กลับไปยังเครื่องโทรสารต้นทางเพื่อยืนยันความพร้อมในการรับข้อมูล

เฟส C ขั้นตอนการส่งข้อมูล (Message transmission)

เฟส C ประกอบด้วย

เฟส C1 ส่งสัญญาณควบคุมข่าวสาร

เป็นขั้นตอนส่งสัญญาณควบคุมเพื่อการซิงโครไนซ์ การตรวจสอบข้อผิดพลาด การแก้ไขข้อผิดพลาด และตรวจสอบสถานะของระบบสื่อสารสัญญาณในขณะที่กำลังส่งข่าวสาร

เฟส C2 ส่งข่าวสาร

เป็นขั้นตอนการส่งข่าวสาร โดยมีรูปแบบตามแต่วิธีการของเครื่องโทรสารแต่ละกลุ่ม

เฟส D ขั้นตอนหลังการส่งข่าวสาร (Post-message procedure)

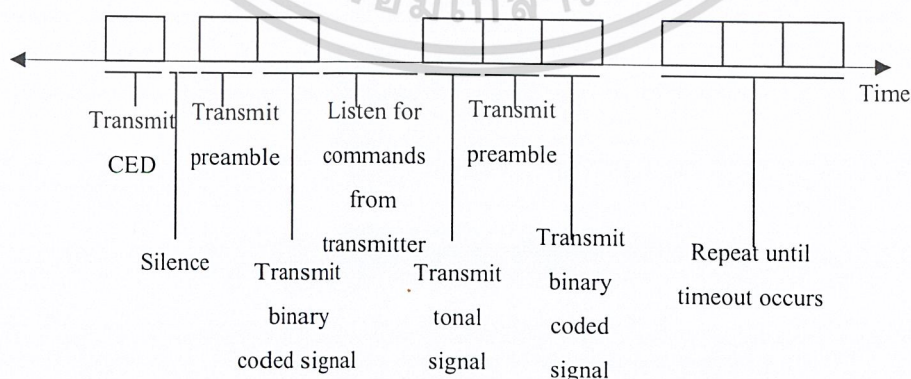
เครื่องโทรสารด้านผู้เรียกระยะส่งสัญญาณ RTC (Return to control) ไปยังด้านถูกเรียกเพื่อทำการปรับโมเด็มให้กลับไปอยู่ที่ความเร็ว 300 บิต/วินาที จากนั้นจึงจะส่งสัญญาณ EOP (End of page) ซึ่งทางด้านถูกเรียกก็จะส่งสัญญาณ MFC (Message confirmation) กลับมาให้ทางด้านต้นทางเพื่อยืนยันการรับข้อมูลที่ถูกต้องเรียบร้อยแล้ว ในกรณีที่มีการส่งข้อมูลหลายๆ หน้า เมื่อจบหน้าแรกจะมีการส่งสัญญาณอื่นๆ ที่แสดงว่ามีข้อมูลที่จะส่งต่อไปอีก แทนการส่งสัญญาณ EOP เช่น EOM (End of message) หรือ MPS (Multi page signal)

เฟส E ปลดสาย (Call release)

เป็นขั้นตอนเลิกการติดต่อระหว่างเครื่องโทรสารต้นทาง และเครื่องโทรสารปลายทาง อาจเป็นแบบใช้มือ (manual) หรือ แบบอัตโนมัติ (automatic) ก็ได้ โดยเครื่องโทรสารด้านผู้เรียกระยะส่งสัญญาณ DCN (disconnect) ไปยังด้านปลายทางเพื่อยกเลิกการติดต่อ

2.8.1 ขั้นตอนการส่งโทรสาร (The Facsimile Procedure)

เครื่องโทรสารทางด้านรับทุกเครื่องจะบอกคุณสมบัติของเครื่องให้ทางด้านผู้เรียกรทราบได้ โดยการส่งขบวนสัญญาณและรหัสฐาน 2 ดังแสดงในรูปที่ 2.25



รูปที่ 2.25 รูปแบบสัญญาณซึ่งแสดงคุณสมบัติของเครื่องโทรสารที่ส่งโดยเครื่องโทรสารด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในชื่อของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไปยังเครื่องโทรสารด้านผู้เรียก
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณแรกที่ส่งออกไปคือ CED ประมาณ 1.8-2.5 วินาที หลังจากเครื่องรับถูกต่อเข้ากับสายเรียบร้อย จะส่งสัญญาณความถี่ประมาณ 2100 เฮิรตซ์ โดยสัญญาณนี้จะถูกส่งอย่างน้อย 2.6 วินาที แต่ไม่เกิน 4 วินาที หลังจากนั้นเครื่องโทรสารด้านถูกเรียกจะหยุดส่งสัญญาณ 75 มิลลิวินาที ก่อนจะส่งสัญญาณตัวต่อไป

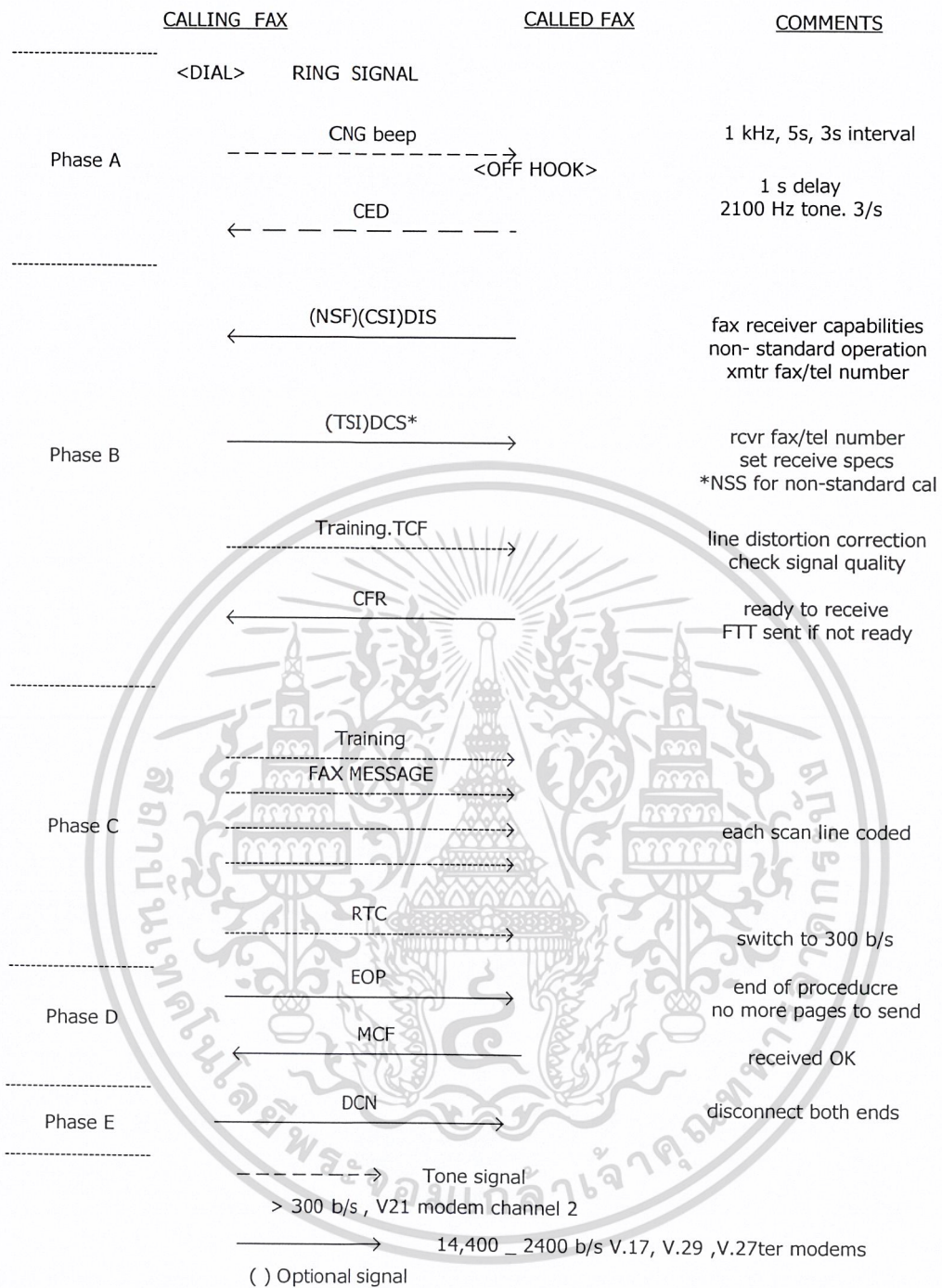
สัญญาณพรีแอมเบิล (preamble) เป็นสัญญาณที่จะส่งนำหน้าสัญญาณรหัสฐาน 2 ตัวอื่น ๆ เสมอ ทุกครั้งที่เริ่มต้นการส่งข้อมูลครั้งใหม่ สัญญาณพรีแอมเบิลเป็นสิ่งที่รับประกันว่าองค์ประกอบต่าง ๆ ในช่องสื่อสารเป็นไปตามกำหนด สำหรับเครื่องโทรสารกลุ่ม 3 ตามมาตรฐานได้กำหนดอัตราส่งสัญญาณพรีแอมเบิลไว้ 300 บิตต่อวินาที หรือ 400 บิตต่อวินาที เป็นทางเลือก สัญญาณรหัสฐาน 2 ที่ถูกส่งตามหลังพรีแอมเบิลนี้จะส่งโดยใช้โครงสร้าง HDLC เฟรม (high level data link control) ซึ่งโครงสร้างพื้นฐานของ HDLC เฟรมจะได้กล่าวถึงในตอนหลัง ในระหว่างเฟส B สัญญาณดังกล่าวนี้จะนำข่าวสารพื้นฐานและข่าวสารที่แสดงความสามารถของเครื่องโทรสารด้านรับ ซึ่งเครื่องโทรสารด้านส่งจะไม่สนใจข่าวสารที่แสดงถึงความสามารถที่มันไม่รู้จัก

หลังจากส่งสัญญาณรหัสฐาน 2 เครื่องโทรสารด้านรับจะรอรับสัญญาณตอบรับจากเครื่องโทรสารด้านส่งประมาณ 3 วินาที ถ้าไม่มีสัญญาณตอบรับกลับมาเครื่องโทรสารด้านรับจะส่งสัญญาณที่สอดคล้องกับแอฟพาราตัส (apparatus) ของเครื่องโทรสารกลุ่ม 1 และ 2 ดังแสดงใน มาตรฐาน T.2 และ T.3 ทั้งนี้ขึ้นอยู่กับความสามารถของเครื่องโทรสารด้านรับ

ประมาณ 1.5 วินาที เครื่องโทรสารด้านรับจะส่งสัญญาณความถี่ 1650 เฮิรตซ์ สำหรับเครื่องโทรสารกลุ่ม 1 หรือความถี่ 1850 เฮิรตซ์ สำหรับกลุ่ม 2 ถ้าเครื่องสามารถรองรับได้ทั้งกลุ่ม 1 และกลุ่ม 2 จะส่งสัญญาณความถี่ 1650 เฮิรตซ์ เป็นเวลา 1.5 วินาที ตามด้วยความถี่ 1850 เฮิรตซ์ ทันทีเป็นเวลา 0.75 วินาที ดังแสดงในรูปที่ 2.25 ซึ่งสัญญาณพรีแอมเบิล รหัสฐาน 2 การหยุดชั่วคราวและสัญญาณความถี่ต่างๆ จะถูกส่งซ้ำ ๆ จนกว่าเครื่องโทรสารด้านผู้เรียกจะส่งสัญญาณตอบรับหรือจนหมดเวลาของไทม์เอาท์

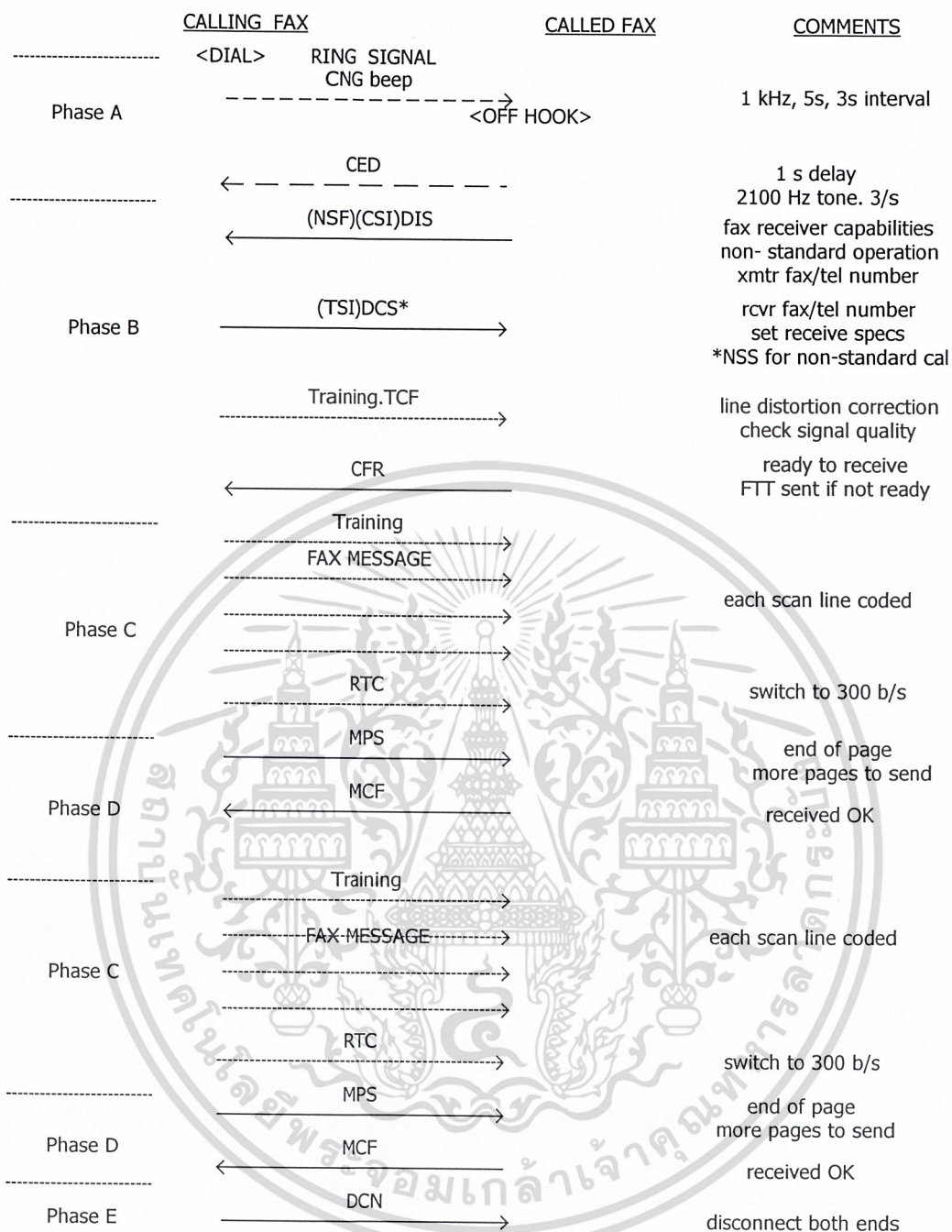
สำหรับรูปแบบของสัญญาณในเฟส C แสดงไว้ในมาตรฐาน T. 2 ,T.3 และ T.4 และสัญญาณในเฟส D คำสั่งหลังการส่งข่าวสารจะถูกส่งจากด้านผู้ส่งไปยังด้านรับ และขึ้นอยู่กับความสามารถของเครื่องรับรวมทั้งคุณสมบัติของการสื่อสาร ด้านส่งอาจจะส่งข้อความจำนวนหนึ่งต่อไปอีกก็ได้ เช่น คำสั่งแสดงการสิ้นสุดข้อความอาจจะถูกส่งออกไปแล้วเครื่องส่งกลับเข้าเฟส B อีกครั้งก็ได้ คำสั่งมัลติเพจ (multipage) แสดงถึงว่าเครื่องส่งจะกลับเข้าสู่เฟส C เป็นต้น ดังรูปที่ 2.26 รูปที่ 2.27 และ แสดงการรับส่งข้อมูลระหว่างเครื่องโทรสาร 1 หน้า และมากกว่า 1 หน้า ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 แสดงขั้นตอนการรับส่งข้อมูลระหว่างเครื่องโทรสาร จำนวน 1 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. Some Group 3 fax machines use EOM incorrectly. If EOM is sent instead of MPS, the next page starts from Phase B, not C. More time is then needed to retrain. If EOM is sent instead of EOP, the fax machine will remain on line for 35 seconds and indicate an error.

รูปที่ 2.27 แสดงขั้นตอนการรับส่งข้อมูลระหว่างเครื่องโทรสารซึ่งมีจำนวนมากกว่า 1 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องโทรสารต้นทาง	เครื่องโทรสารปลายทาง
1. ส่งสัญญาณ CNG เพื่อเริ่มการติดต่อ 4. รับสัญญาณ DIS และคุณสมบัติของเครื่องปลายทาง 5. ส่งสัญญาณ DCS 8. ส่งสัญญาณ training เพื่อทดสอบคุณภาพของสาย 11. รับสัญญาณ CRF เข้าโหมดการส่งข้อมูล 12. ส่งข้อมูลภาพ 14. เมื่อส่งข้อมูลภาพหมดแล้วส่งสัญญาณเพื่อแสดงการสิ้นสุดของข้อมูลเช่นEOM,EOP,MPS เป็นต้น 16. รับสัญญาณ MCF แล้วส่ง DCN เพื่อยกเลิกการติดต่อ	2. ส่งสัญญาณ CED ตอบรับการติดต่อ 3. ส่งสัญญาณ DIS 6. รับสัญญาณ DCS และข้อกำหนดการส่งข้อมูล 7. เลือกโหมดตามข้อกำหนดรับข้อมูล 9. รับสัญญาณ training และตรวจสอบว่ามีความผิดพลาดหรือไม่ 10. ถ้าไม่มีความผิดพลาดส่งสัญญาณ CFR 13. รับข้อมูลภาพ 15. รับสัญญาณแสดงการสิ้นสุดของข้อมูล เช่น EOMแล้วส่งสัญญาณ MCF ไปเพื่อยืนยันว่ารับข้อมูลเรียบร้อยแล้ว

ตารางที่ 2.11 แสดงขั้นตอนการรับส่งข้อมูลเมื่อเครื่องโทรสารต้นทางเป็นผู้ส่งเอกสารให้ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องโทรสารต้นทาง	เครื่องโทรสารปลายทาง
1. ส่งสัญญาณ CNG เริ่มการติดต่อ	2. ส่งสัญญาณ CED ตอบรับการเรียก
4. รับสัญญาณ DIS และคุณสมบัติของเครื่องปลายทาง	3. ส่งสัญญาณ DIS
5. ส่งสัญญาณ DTC	6. รับสัญญาณ DTC และคุณสมบัติของเครื่องต้นทาง
8. รับสัญญาณ DCS และเลือกโหมดตามข้อกำหนดรับข้อมูล	7. กำหนดคุณสมบัติที่จะใช้รับส่งข้อมูลที่ส่งไปรับสัญญาณ DCS
10. รับสัญญาณ training ตรวจสอบว่ามีความผิดพลาดหรือไม่	9. ส่งสัญญาณ training เพื่อตรวจสอบคุณภาพสาย
11. ถ้าไม่มีความผิดพลาดส่งสัญญาณ CFR	12. รับสัญญาณ CFR เข้าสู่โหมดการส่งข้อมูล
14. รับข้อมูล	13. ส่งข้อมูล
16. รับสัญญาณการสิ้นสุดของข้อมูลเช่น EOM แล้วส่งสัญญาณ MCF เพื่อยืนยันการรับข้อมูล	15. เมื่อส่งข้อมูลหมดแล้ว ส่งสัญญาณเพื่อแสดงว่าสิ้นสุดข้อมูลแล้ว ได้แก่ EOM, EOP, MPS เป็นต้น
17. ส่งสัญญาณ DCN ยกเลิกการติดต่อ	

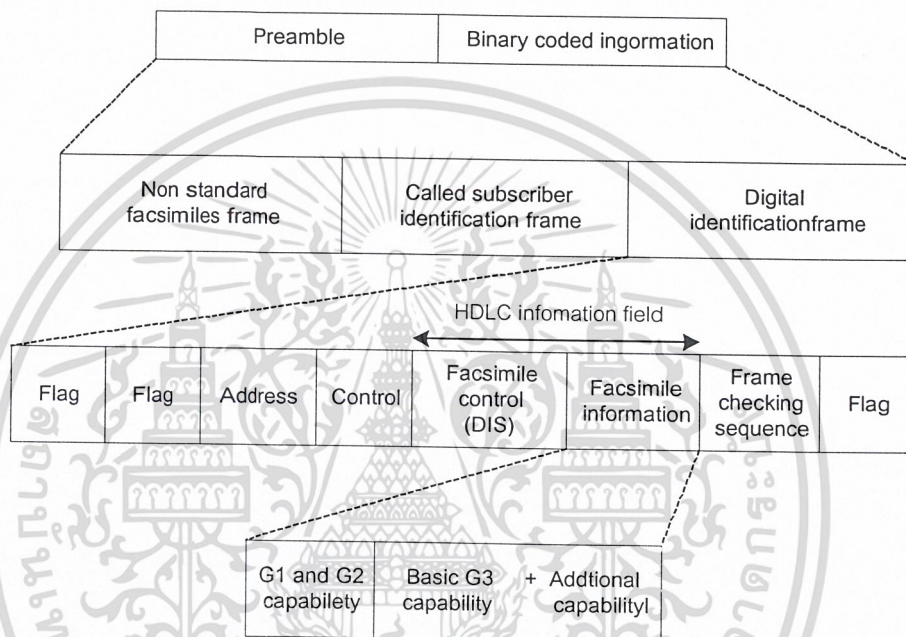
ตารางที่ 2.12 แสดงขั้นตอนการรับส่งข้อมูลเมื่อเครื่องโทรสารปลายทางเป็นผู้ส่งเอกสาร

หมายเหตุ ในกรณีที่เครื่องโทรสารกลุ่ม 3 บางเครื่องมีความสามารถและคุณสมบัติพิเศษที่นอกเหนือไปจากมาตรฐาน จะมีการส่งสัญญาณ NSF (Nonstandard facilities) และ CSI (Called subscriber identification) นำหน้าสัญญาณ DIS ซึ่ง NSF เป็นสัญญาณที่บอกให้ทราบว่าเครื่องโทรสารต้นทางมีความสามารถพิเศษ นอกเหนือไปจากมาตรฐานตามข้อกำหนด มาตรฐาน T.Scies หลังจากด้านส่งได้รับสัญญาณ NSF แล้วจะส่งสัญญาณ NSS (Nonstandard setup) กลับไปยังด้านรับเพื่อกำหนดให้เครื่องโทรสาร

ด้านรับอยู่ในระบบคำสั่งพิเศษ อย่างไรก็ตาม แม้ว่าเครื่องโทรสารส่วนหนึ่งจะมีความสามารถพิเศษเฉพาะกลุ่มที่ผลิตจากบริษัทเดียวกัน แต่จะมีความสามารถสนองตอบต่อคำสั่งมาตรฐานได้เช่นกัน

2.9 รูปแบบของ HDLC เฟรม (high level data link control)

ในการติดต่อรับส่งข้อมูลระหว่างเครื่องโทรสาร จะมีการใช้ HDLC เฟรม (high level data link control) ช่วยให้การรับส่งข้อมูลเป็นไปอย่างถูกต้อง ซึ่งสัญญาณรูปแบบ HDLC เฟรมประกอบไปด้วยส่วนต่างๆ ดังรูปที่ 2.28 นี้



รูปที่ 2.28 แสดงรูปแบบของสัญญาณ HDLC ที่ใช้ติดต่อระหว่างเครื่องโทรสาร

1. 프리แอมเบิล (Preamble)

เป็นส่วนแรกของขบวนสัญญาณเสมอเมื่อเป็นการเริ่มต้นการสื่อสาร ซึ่งฟรีแอมเบิลจะช่วยให้แน่ใจได้ว่าสัญญาณที่อยู่ถัดมา มีความถูกต้องตามเงื่อนไขที่เราต้องการ รูปแบบของฟรีแอมเบิล คือ 0111 1110

2. แฟล็ก (flag)

มีรูปแบบไบนารี 8 บิต (0111 1110) ทำหน้าที่เป็นตัวชี้จุดเริ่มต้นและจุดสิ้นสุดของเฟรม และเป็นเฟรมซิงค์โครไนเซชัน

3. ส่วนของที่อยู่ (address field)

มีรูปแบบไบนารี 8 บิต (1111 1111) ใช้ในการกำหนดการติดต่อในระบบการติดต่อแบบหลายจุด

จุด

4. ส่วนควบคุม (control field)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรูปแบบไบนารี 8 บิต(1100 X000) ซึ่งสำหรับการส่งข้อมูลของเครื่องโทรสาร X จะมีค่าเป็น 1 เมื่อเป็นเฟรมสุดท้ายในขบวนการจัดการ

5. ส่วนของข้อมูล (information field)

ความยาวของฟิลด์นี้จะขึ้นอยู่กับข้อมูลที่บรรจุอยู่ภายใน ซึ่งได้แก่ คำสั่งควบคุมและข่าวสารที่แลกเปลี่ยนกันระหว่างเครื่องโทรสาร ตามข้อกำหนดของ มาตรฐาน T.30 ได้แบ่งฟิลด์นี้ออกเป็น 2 ส่วนได้แก่

5.1 ส่วนควบคุม FCF (Facsimile Control Field)

เป็นสัญญาณ 8 บิต หรือ 1 บิตแรกของส่วนข้อมูลซึ่ง FCF จะประกอบไปด้วยข้อมูลที่เกี่ยวข้องชนิดของข่าวสารที่แลกเปลี่ยนกันและหน้าที่เฟรม โดยมีรูปแบบดังต่อไปนี้

- บิตแรกจะถูกกำหนดให้เป็น 1 โดยสถานีด้านผู้เรียก
- บิตแรกจะถูกกำหนดให้เป็น 0 โดยสถานีด้านถูกเรียก
- บิตแรกจะมีค่าคงที่ไม่เปลี่ยนแปลงจนกว่าจะเข้าสู่ขั้นตอนก่อนการส่งข่าวสาร(เฟส B)

5.2 ส่วนของข้อมูล FIF (Facsimile Information Field)

ส่วนของข้อมูล FIF จะตามหลังส่วนควบคุม FCF โดยมีขนาด 8 บิต แสดงถึงขบวนการในการรับส่งข้อมูลของเครื่องโทรสารซึ่งได้แก่สัญญาณDIS,DCS,DTC,CSE,CIG,TST,NSC,SCF,VSS,CTC,PPS และPPR

6.ส่วนของการตรวจสอบ

FCS มักจะถูกเรียกอีกชื่อหนึ่งว่า CRC ซึ่งใช้สำหรับการตรวจสอบข้อมูลในเฟรม มีขนาด 16 บิต ซึ่งจะคำนวณได้จากจำนวนบิต ทั้งหมดในเฟรมนั้นก่อนที่จะถึง FCS โดยที่โมเด็มจะสร้าง FCS เพื่อใช้ในการส่งข้อมูลใน 1เฟรมและจะตรวจสอบ FCS ที่รับมาจากเครื่องโทรสารปลายทาง

2.10 แฟกซ์โมเด็ม

หากมองด้านรูปลักษณะภายนอกที่มองเห็น เราไม่สามารถแยกความแตกต่างระหว่างแฟกซ์ โมเด็มและดาต้าโมเด็มออกจากกันได้ และวิธีการผสมสัญญาณหรือวิธีการตรวจสอบความผิดพลาดที่ใช้ในดาต้าแฟกซ์โมเด็มก็คล้ายคลึงกับที่ใช้ในแฟกซ์โมเด็มเช่นกัน ข้อแตกต่างกันที่เด่นชัดคือ โปรโตคอล ซึ่งควบคุมการติดต่อสื่อสาร แฟกซ์โมเด็มจะต้องสนับสนุนโปรโตคอลที่กำหนดได้ในมาตรฐาน T.4,T.6 และT.30 ของ ITU - T กล่าวคือแฟกซ์โมเด็มจะต้องจัดหาคำสั่งใหม่ๆ และตอบสนองต่อการเริ่มต้นที่จะนำไปสู่การส่งโทรสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit per Second	Baud Rate	Bits per Sample	Type	Carrier Frequency	Bandwidth in Hertz
14400	2400	6	V.17	1800	550-3050
12000	2400	5	V.17	1800	550-3050
9600	2400	4	V.29	1700	450-2950
7200	2400	3	V.29	1700	450-2950
4300	1600	3	V.27 ter	1800	950-2650
2400	1200	2	V.27 ter	1800	1150-2450

ตารางที่ 2.13 มาตรฐานของโมเด็มที่ใช้ในเครื่องโทรสารกลุ่มที่ 3 ตามข้อกำหนดของ ITU – T

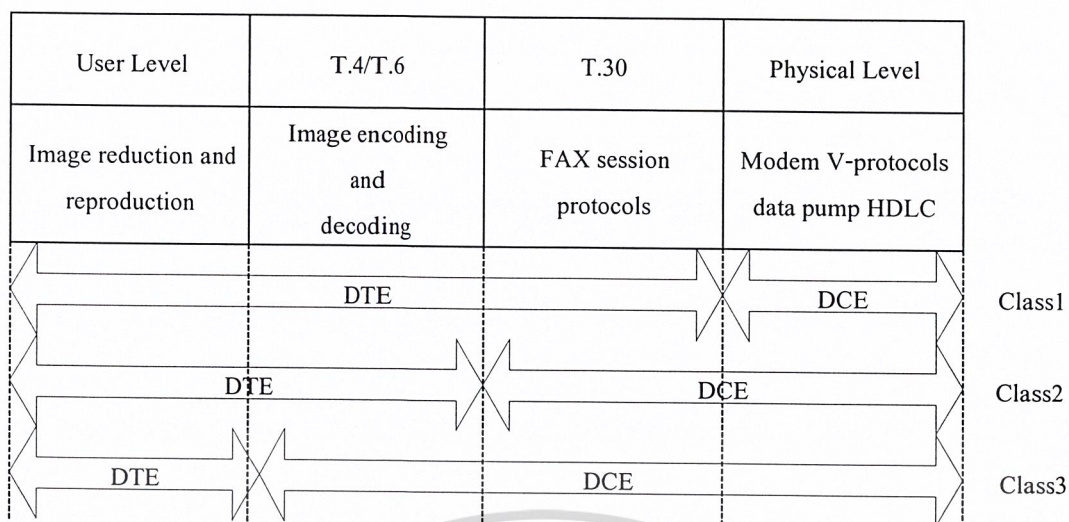
โมเด็มที่ใช้ในเครื่องโทรสารกลุ่ม 3 จะตรวจสอบการติดต่อระหว่างเครื่องโทรสารทั้ง 2 ด้านก่อนการส่งข้อมูล และโมเด็มจะพยายามส่งข้อมูลด้วยความเร็วสูงสุดเท่าที่จะเห็นไปได้ ส่วนใหญ่โมเด็มในเครื่องโทรสารได้ถูกกำหนดได้ตามมาตรฐาน V.27ter และ V.29 แต่จะมีเครื่องโทรสารรุ่นใหม่ๆบางเครื่องที่จะใช้มาตรฐาน V.30 โดย V.27ter เป็นมาตรฐานเดียวกับโมเด็มที่ส่งข้อมูลด้วยความเร็ว 4800/400 บิตต่อวินาที ผ่านข่ายสายโทรศัพท์ โดยการผสมสัญญาณแบบเลื่อนเฟส โดยมีความถี่พาห้ที่ 1800 ± 1 เฮิรตซ์ ส่วน V.29เป็นข้อแนะนำในการส่งแบบ 9600 บิต/วินาที โดยความถี่พาห้เป็น 1700 ± 1 เฮิรตซ์ และ V.33 จะส่งข้อมูลด้วยความเร็วสูงสุด 14400 บิต/วินาที ผสมสัญญาณแบบ QAM

โดยส่วนใหญ่ในการส่งข้อมูลของเครื่องโทรสารกลุ่ม 3 นั้น จะเริ่มต้นที่ความเร็ว 9600 บิตต่อวินาที แต่ถ้าหากไม่สามารถส่งข้อมูลด้วยความเร็วนี้ได้ โมเด็มจะลดความเร็วในการส่งลงเป็น 7200, 4800 หรือ 2400 บิตต่อวินาที แทนตามความเหมาะสม

2.11 การแบ่งระดับโมเด็มโดย EIA

ความสามารถของเครื่องโทรสารที่เพิ่มขึ้นจากโมเด็มปกติ ทำให้จำเป็นต้องมีคำสั่งใหม่ๆ ในการติดต่อแต่ไม่มีบริษัทใดเป็นผู้กำหนดมาตรฐานนี้ ดังนั้นแฟกซ์โมเด็มจึงถูกพัฒนามาตรฐานซึ่งกำหนดขั้นตอนและคำสั่งที่ใช้ระหว่างคอมพิวเตอร์ และ โมเด็ม ขึ้น โดยแบ่งโมเด็ม ออกเป็น 3 ระดับตามความสามารถของโมเด็มในการนำไปสู่การโทรสารดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 แสดงการแบ่งระดับ โมเด็มโดย EIA

แฟกซ์โมเด็มระดับ 1

EIA ได้กำหนดแฟกซ์โมเด็มระดับ 1 ไว้ใน EIA/TIA -578 ซึ่งโมเด็มระดับนี้จะหมายถึงแฟกซ์โมเด็ม ที่มีบริการเท่าที่จำเป็นต่อการรองรับขั้นตอนของโทรสารกลุ่ม 3 เท่านั้น คอมพิวเตอร์จะตอบสนองต่อการเข้ารหัสของภาพโดย มาตรฐาน T.4 และการจัดการในการส่งข้อมูลโดยมาตรฐาน T.30 แฟกซ์โมเด็มระดับ 1 จะมีบริการต่างๆต่อไปนี้

- GSTN interface
- หมายเลขเลขอัตโนมัติ
- V series signal conversion(modulation)
- การรับส่งข้อมูล
- การใช้รูปแบบ HDLC เฟรม ความชัดเจนของข้อมูล และการตรวจสอบความผิดพลาด
- คำสั่งควบคุมและการตอบสนอง

การส่งโทรสารที่แฟกซ์โมเด็มระดับ 1 จะต้องใช้ภายใต้การควบคุมของซอฟต์แวร์ ต่างจากการส่งแบบใช้ค้ำโมเด็ม ซึ่งการกำหนดเวลา การเข้ารหัสและการสำหรับการส่งต้องอาศัย มาตรฐาน T.30 ทำให้ไม่สามารถควบคุมการส่ง โทรสาร โดยใช้คำสั่งจากผู้ควบคุมเครื่องได้

แฟกซ์โมเด็มระดับ 2

แฟกซ์โมเด็มระดับนี้มีความสามารถสูงกว่าแฟกซ์โมเด็มระดับแรก การแบ่งส่วนต่างๆ ก็จะเปลี่ยนไป ซึ่งขั้นตอนที่ยุงยากในการส่งโทรสารโดยใช้ มาตรฐาน T.30 จะถูกตัดออกจากคอมพิวเตอร์ไป เห็นหน้าที่ของโมเด็มแทน ซึ่งโมเด็มระดับ นี้มีบริการต่างๆต่อไปนี้

- GSTN interface
- การหมุนหมายเลขอัตโนมัติ
- V series signal conversion

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การส่งข้อมูลเฟส C
- padding สำหรับเวลาในการสแกนที่น้อยที่สุด
- การตรวจสอบข้อมูลที่รับได้
- โปรโตคอลแบบแพ็คเกจ (packet) สำหรับ DTE/DCE interface

แพ็คเกจโมเด็มระดับ 2 จะเริ่มต้นการเรียก จัดการขบวนการติดต่อสื่อสาร ส่งข้อมูลภาพและหน้าที่อื่นๆ ระหว่างรูปแบบของ มาตรฐาน T.4 (เครื่องโทรสารกลุ่ม3) และมาตรฐาน T.6 (เครื่องโทรสารกลุ่ม 4) เครื่องคอมพิวเตอร์จะทำหน้าที่ในการจัดเตรียมและลดขนาดข้อมูลภาพสำหรับการสื่อสารและตีความข้อมูลที่ถูกลดขนาดมาแล้วในทางด้านรับ

แพ็คเกจโมเด็มระดับ 3

การกำหนดมาตรฐานของแพ็คเกจโมเด็มระดับ 3 ยังอยู่ในระหว่างการศึกษา แต่มีแนวโน้มในการโอนหน้าที่ต่างๆ ในขบวนการส่งโทรสารจากเครื่องคอมพิวเตอร์ไปให้กับโมเด็ม จากมาตรฐาน T.30 และหน้าที่ทั่วไปของโมเด็มในการส่งข้อมูล แพ็คเกจโมเด็มระดับ 3 จะเปลี่ยนข้อมูลภาพให้เป็นภาพที่ถูกลดขนาดตามมาตรฐาน T.4 และ T.6 ซึ่งโมเด็มอาจจะเป็นตัวขยายภาพทางด้านรับด้วยก็ได้

2.12 จดหมายอิเล็กทรอนิกส์ (E-Mail , Electronic Mail)

จดหมายอิเล็กทรอนิกส์ เป็นบริการหนึ่งบนระบบเครือข่ายสื่อสารคอมพิวเตอร์ โดยมีการใช้งานมาตั้งแต่ยุคของคอมพิวเตอร์แบบเมนเฟรม ซึ่งต่างก็ใช้เทคโนโลยีของตนเองและเป็นระบบปิด อาทิ เช่น บริษัท IBM พัฒนาระบบที่เรียกว่า PROFS (Professional Office System) และยังมีระบบจดหมายอิเล็กทรอนิกส์ ที่ใช้อยู่บน UNIX และยังมี การพัฒนาระบบแอปพลิเคชัน (Application) ที่ทำงานบนระบบเครือข่ายต่างๆกัน เช่น Microsoft Mail ของบริษัทไมโครซอฟต์ (Microsoft) cc: Mail ของบริษัทโลตัส (Lotus) ดังนั้นการส่งจดหมายอิเล็กทรอนิกส์ไปยังผู้ใช้ระบบจึงเป็นเรื่องที่อยู่ยาก

ในยุคต่อมาระบบเครือข่ายต่างๆ มีมาตรฐานและการพัฒนาเป็นระบบเปิด (Open System) มากขึ้น ก็มีการปรับเปลี่ยนการทำงานระบบจดหมายอิเล็กทรอนิกส์เป็นระบบไคลเอนต์ – เซิร์ฟเวอร์ (Client – Server) ที่เป็นพื้นฐานที่ใช้กับในระบบ Unix และการพัฒนาอีเมลล์เซิร์ฟเวอร์ (E – mail server) ขึ้นมาโดยเฉพาะ ซึ่งทำให้การติดต่อเข้าสู่อีเมลล์เซิร์ฟเวอร์ สามารถทำได้โดยผ่านระบบ LAN (Local Area Network) หรือใช้โมเด็มหมุนเข้ามา ทำให้การใช้งานจดหมายอิเล็กทรอนิกส์ทำได้ง่ายขึ้น

2.12.1 ส่วนประกอบของระบบจดหมายอิเล็กทรอนิกส์

จดหมายอิเล็กทรอนิกส์เป็นบริการที่ทำหน้าที่รับส่งข่าวสารบนระบบเครือข่ายคอมพิวเตอร์ ซึ่งมีลักษณะการใช้งานคล้ายกับการรับส่งจดหมายในชีวิตประจำวัน ระบบจดหมายอิเล็กทรอนิกส์จะมีศูนย์ไปรษณีย์กลาง (Post office) เมลล์บ็อกซ์ (Mailbox) แต่ต้องใช้งานบนระบบเครือข่ายคอมพิวเตอร์จดหมายอิเล็กทรอนิกส์ทำให้ผู้ใช้สามารถรับส่งข่าวสารบนระบบเครือข่ายทั่วโลก อย่างรวดเร็วและประหยัดค่าใช้จ่าย ระบบจดหมายอิเล็กทรอนิกส์สามารถจัดเก็บข่าวสารที่ได้รับให้เป็นหัวข้อ รวมทั้งมีการส่งข่าวสารต่อไปยังผู้รับรายอื่นได้ และสามารถ ส่งไฟล์ (File) ข้อมูลอื่นๆ นอกเหนือจากข่าวสารที่ต้องการส่งไปได้ เช่น

เอกสารเป็นเอกสารที่ส่งฟรีสำหรับบริการเชิงพาณิชย์ที่ให้บริการแก่ผู้ใช้และผู้ให้บริการที่มีค่าใช้จ่ายในการดำเนินการ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ข้อมูลภาพและเสียง ข้อดีต่างๆ ของจดหมายอิเล็กทรอนิกส์ ทำให้ปัจจุบันจดหมายอิเล็กทรอนิกส์จัดเป็นโปรแกรมที่มีการใช้งานมากที่สุดในระบบเครือข่ายคอมพิวเตอร์

การทำงานของระบบอีเมลไคลเอนต์ – เซิร์ฟเวอร์ มีส่วนประกอบดังนี้

- Mail User Agent (MUA) เป็นโปรแกรมคอมพิวเตอร์ทางด้านผู้ใช้งานแบ่งเป็นส่วนคือ
- ส่วนของผู้ส่งและ ส่วนของผู้รับ โดย User Agent จะต่อเข้าสู่เซิร์ฟเวอร์ ของตนโดยผ่านระบบ LAN หรือใช้โมเด็มหมุนเข้ามา ซึ่งในส่วนนี้ผู้ใช้จะติดตั้งโปรแกรมไคลเอนต์ของจดหมายอิเล็กทรอนิกส์เพื่อเรียกใช้ บริการจดหมายอิเล็กทรอนิกส์ เช่น Outlook Express หรือ Eudora
- Mail Transfer Agent (MTA) เป็นโปรแกรมคอมพิวเตอร์ที่จะทำหน้าที่ส่งจดหมายไปยังผู้รับปลายทาง ซึ่งโปรแกรมเหล่านี้จะทำหน้าที่ช่วยกันส่งจดหมายอิเล็กทรอนิกส์ ต่อกันไปเป็นทอดๆ จนถึงเครือข่ายที่มี Account หรือ เมลล์บ็อกซ์ของผู้รับ และถ้าหากไม่สามารถส่งจดหมายอิเล็กทรอนิกส์ ถึงผู้รับได้ จะทำหน้าที่ส่ง error mail กลับมายังผู้ส่งได้อีกด้วย โดยทั่วไปเครื่องคอมพิวเตอร์ที่มี MTA ทำงานอยู่มักจะมีเมลล์บ็อกซ์ ของผู้ใช้อยู่ด้วย ซึ่งเป็น primary mailbox และเรียกเครื่องคอมพิวเตอร์นั้น ว่า Mail Server

การทำงานทั่วไปของจดหมายอิเล็กทรอนิกส์โดยทั่วไปมี 2 ประเภท คือ การส่งจดหมายอิเล็กทรอนิกส์ และการรับจดหมายอิเล็กทรอนิกส์ โดยการทำงานทั้ง 2 ประเภทจะมีโปรโตคอล (Protocol) SMTP (Simple Mail Transfer Protocol) จะใช้ในขณะส่งจดหมายอิเล็กทรอนิกส์ จาก User agent มายัง MTA และโปรโตคอล POP 3 (Post Office Protocol Version 3) ใช้เมื่อทำการส่งจดหมายอิเล็กทรอนิกส์จากเครื่องผู้ใช้ไปยังเครื่องของผู้ใช้เครื่องที่มี MTA อยู่

รูปแบบของข้อมูลที่ใช้ในโปรโตคอลต่างๆ ของจดหมายอิเล็กทรอนิกส์ นี้ถูกกำหนดไว้ใน RFC822 ซึ่งแบ่งส่วนประกอบภายในจดหมายอิเล็กทรอนิกส์ เป็น 2 ส่วน คือ ส่วนที่จําหน่ายจดหมายอิเล็กทรอนิกส์ และ ข้อมูลของจดหมายอิเล็กทรอนิกส์ ในส่วนของการจําหน่ายจดหมายอิเล็กทรอนิกส์มีไว้เพื่อให้ส่งไปถึงผู้รับ รูปแบบข้อมูลจะเป็นข้อมูลจะเป็นข้อความ โดยข้อมูลจะนำหน้าด้วยคำสำคัญ (Keyword) เช่น From หมายถึงชื่อผู้ส่ง และ To หมายถึงผู้รับ โดยในข้อมูลแต่ละบรรทัดจะปิดท้ายด้วย Carriage Return (CR) และ Line Feed (LF) สำหรับส่วนที่เป็นข้อมูลของจดหมายอิเล็กทรอนิกส์ จะแบ่งออกย่อยเป็น 2 ส่วน ส่วนหัว (Header) และส่วนเนื้อหาของจดหมายอิเล็กทรอนิกส์ โดยส่วนหัวจะถูกสร้างขึ้นโดยอัตโนมัติ โดย User Agent ของผู้ส่ง เพื่อให้ MTA ต่าง ๆ ระหว่างทางที่ส่งผ่านจดหมายอิเล็กทรอนิกส์ ฉบับนั้นได้อ่านไปใช้งาน ซึ่งจะมีข้อมูลต่างๆหลายประเภท เช่น เลขทะเบียนของจดหมายอิเล็กทรอนิกส์ (Message Header) วันที่ส่ง เวลาที่ส่ง เป็นต้น เนื้อหาของจดหมายอิเล็กทรอนิกส์ นั้นจะอยู่แยกจากส่วนหัว โดย ถูกขึ้นด้วยบรรทัดว่างๆ (Null Line) และในแต่ละบรรทัดของข้อความจะสิ้นสุดบรรทัดด้วย Carriage Return (CR) และ Line Feed (LF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

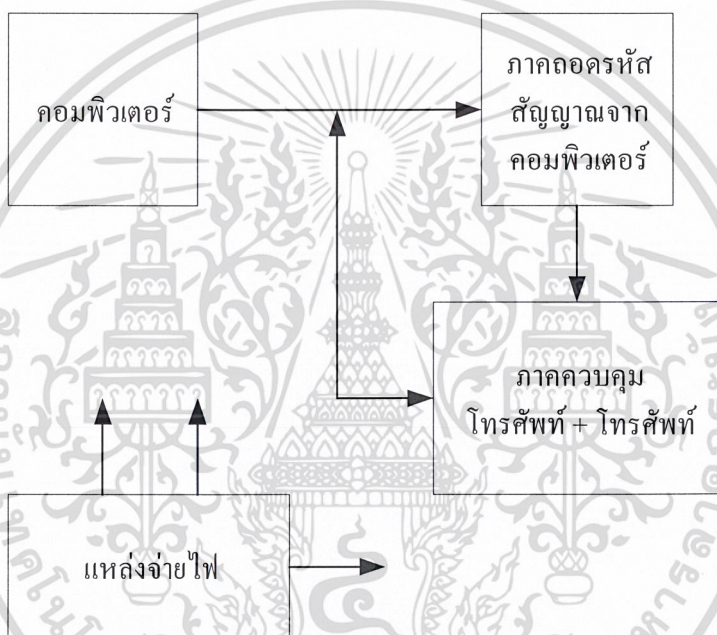
บทที่ 3

การออกแบบและการสร้าง

3.1 ส่วนฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์ เราจะต้องออกแบบให้วงจรสามารถทำการยกหูเมื่อมีการส่งสัญญาณจากคอมพิวเตอร์มา และเมื่อยกหูแล้วก็จะให้โทรศัพท์นั้นทำการหมุนหมายเลข โดยเราก็จะใช้สัญญาณจากคอมพิวเตอร์อีกครั้งในการส่งสัญญาณที่เป็นหมายเลขปลายทางเพื่อทำการโทรไปยังบุคคลที่เราต้องการจะคุยด้วย และ ก็ทำการติดต่อกันแบบปกติต่อไป

จากแนวคิดสามารถเขียนเป็นบล็อกไดอะแกรมได้ดังนี้

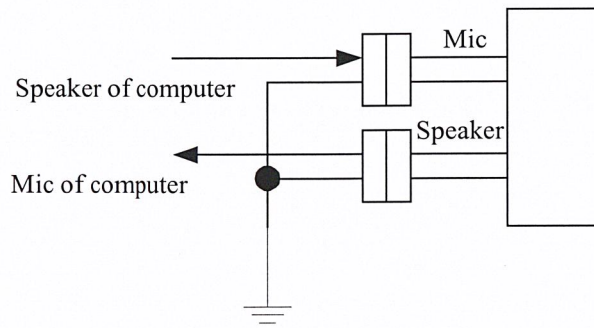


รูปที่ 3.1 แสดงบล็อกไดอะแกรมคร่าวๆ ของวงจร

เราจะควบคุมวงจรโดยใช้สัญญาณโทรศัพท์ที่ส่งมาจากคอมพิวเตอร์ คือเราจะต่อสัญญาณเอาท์พุทจากคอมพิวเตอร์โดยต่อจากขาตัวคาร์ดของคอมพิวเตอร์ นั่นคือสัญญาณที่ได้มาเป็นสัญญาณเสียงนั่นเอง ซึ่งเราก็ต้องใช้อยู่แล้วในการพูดคุยผ่านเครือข่าย นั่นคือ เราจะต่อสายออกมาคอมพิวเตอร์มาเข้าวงจร 2 สาย คือ สายสัญญาณที่ต่อออกมาจากช่องเสียบสปีกเกอร์ของคอมและช่องเสียบไมค์ของคอมพิวเตอร์

โดยสปีกเกอร์ของคอมพิวเตอร์จะเป็นส่วนส่งสัญญาณเสียงออกมา ทั้งสัญญาณ DTMF และสัญญาณเสียงจากคอมพิวเตอร์อีกเครื่องที่ถูกส่งเข้ามา และ ไมค์ของคอมพิวเตอร์จะรับสัญญาณเสียงจากโทรศัพท์แล้วก็ส่งกลับไปยังเครื่องคอมพิวเตอร์อีกเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการต่อระหว่างคอมพิวเตอร์ กับอุปกรณ์

จากการต่อดังที่กล่าวแล้วนั้นคือเราสามารถรับส่งเสียงระหว่างคอมพิวเตอร์กับโทรศัพท์ได้แล้ว แต่จะเราสามารถรับส่งได้ก็ต่อเมื่อโทรศัพท์มีการยกหูแล้วเท่านั้น

ในโครงงานนี้เราต้องออกแบบให้โทรศัพท์ยกหูโดยใช้สัญญาณจากคอมพิวเตอร์เข้ามาทำการยกหูให้ โดยเราออกแบบให้ส่งสัญญาณ DTMF จากคอมพิวเตอร์ออกไป ซึ่งต้องใช้ส่งอยู่แล้วในการหมุนโทรศัพท์ แต่ในการหมุนจะเห็นได้ว่าเราจะกดตัวเลขเพียง 0-9 เท่านั้น แล้ว * กับ # ก็ไม่ค่อยได้ใช้ แต่บางกรณี * กับ # ต้องได้ใช้ แต่ใน คีย์โทรศัพท์จริงแล้วยังมีแนวตั้งที่ 4 อยู่ซึ่งโทรศัพท์ทั่วไปไม่มีดังรูปที่ 3.3

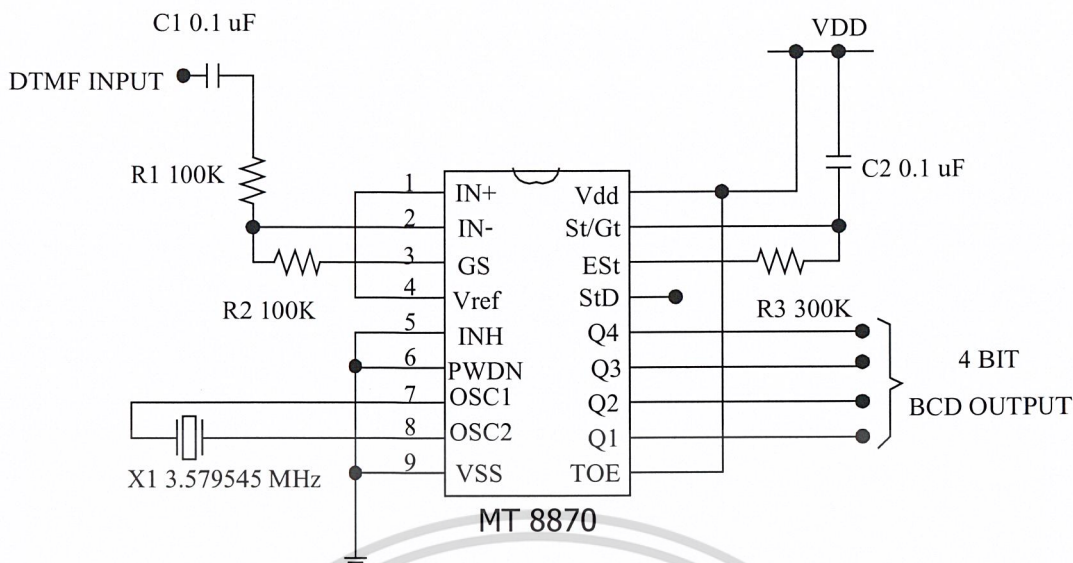
1	2	3	A
4	5	6	B
7	8	9	C
*	0	#	D

รูปที่ 3.3 แสดงคีย์โทรศัพท์ทั้งหมด

จะเห็นได้ในโทรศัพท์ปกติจะไม่มี คีย์ A B C และ D ดังนั้นเราจะออกให้สัญญาณ A เป็นตัวควบคุมการยกหูและวางหูโทรศัพท์

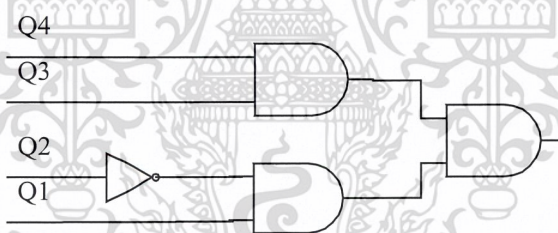
โดยเราจะทำการต่อสัญญาณจาก Speaker of Computer เข้ามาวงจรถอดรหัส DTMF ซึ่งเราได้กล่าวไว้แล้วในส่วนของทฤษฎี นั่นคือจะใช้ IC เบอร์ MT8870 มาทำการถอดรหัส DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การต่อวงจร MT8870

จากรูปที่ 3.4 เป็นวงจรที่อยู่ใน Data Sheet ของ MT8870 ซึ่งบริษัทได้ออกแบบมาให้แล้ว โดยอินพุตหรือสัญญาณเสียงที่ต่อจาก Speaker of Computer ก็จะเข้าที่ C1 ผ่าน R1 เข้าสู่ไอซี และเอาท์พุทที่ได้คือ Q4-Q1 จะเป็นเอาท์พุทที่อยู่ในรูปของ BCD



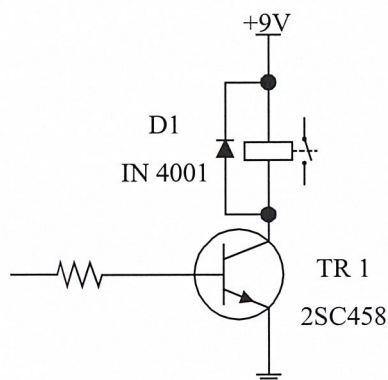
รูป 3.5 วงจรถอดรหัสสัญญาณ BCD 1101

จากทฤษฎี เมื่อเราดูในส่วนเอาท์พุทที่ได้จากการถอดรหัส A เอาท์พุทที่ได้คือ 1101 และเราจะนำ 1101 ไปถอดรหัสให้มีเอาท์พุทเป็น 1 เพื่อไปใช้ควบคุมให้ยกหู

จากรูปที่ 3.5 วงจรจะเห็นได้ว่าเมื่อ มีสัญญาณเข้ามาเป็น 1101 เอาท์พุทจะเป็น 1 ถ้าเป็นสัญญาณอื่นๆ เอาท์พุทก็จะเป็น 0

เราก็เอาเอาท์พุทของวงจรนี้ไปควบคุมให้โทรศัพท์ยกหูได้โดยต่อวงจรรวมดังรูปที่ 3.8

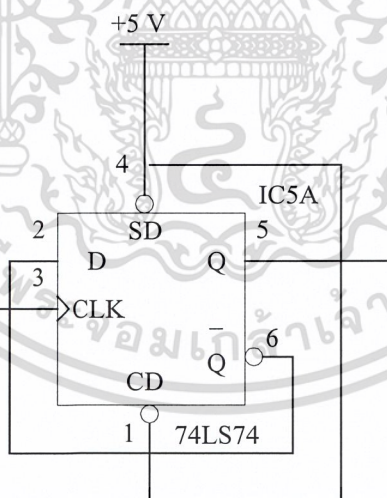
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรควบคุมรีเลย์

จากเอาต์พุตของวงจรถอดรหัสสัญญาณ BCD 1101 เรานำมาต่อเข้าวงจรควบคุมรีเลย์ เมื่อ อินพุตเป็น 1 ทรานซิสเตอร์จะทำงาน รีเลย์ก็จะทำงาน สวิตช์ของรีเลย์ก็จะต่อวงจร ซึ่งเราจะต่อสวิตช์ของรีเลย์กับสวิตช์ของโทรศัพท์ในลักษณะขนานกัน

จากรูปที่ 3.6 วงจรจะเห็นได้ว่าเมื่อรีเลย์ทำงานสวิตช์ก็จะปิดวงจร นั่นก็คือ โทรศัพท์ทำการยกหูแล้วนั่นเอง แต่รีเลย์จะทำงานก็ต่อเมื่อ สัญญาณ A มาเท่านั้น แต่เมื่อในการใช้งานจริงเราส่ง A มาเพียงครั้งเดียวเพื่อเป็นการยกหูเท่านั้น จากนั้นก็ส่งหมายเลขต่างๆ ของหมายเลขโทรศัพท์ที่เราจะติดต่อด้วยตามมา ฉะนั้นเราจะต้องออกแบบให้ เมื่อส่ง A มาแล้ว รีเลย์จะต้องทำงานค้าง โดยเราสามารถออกแบบได้ดังรูปที่ 3.7

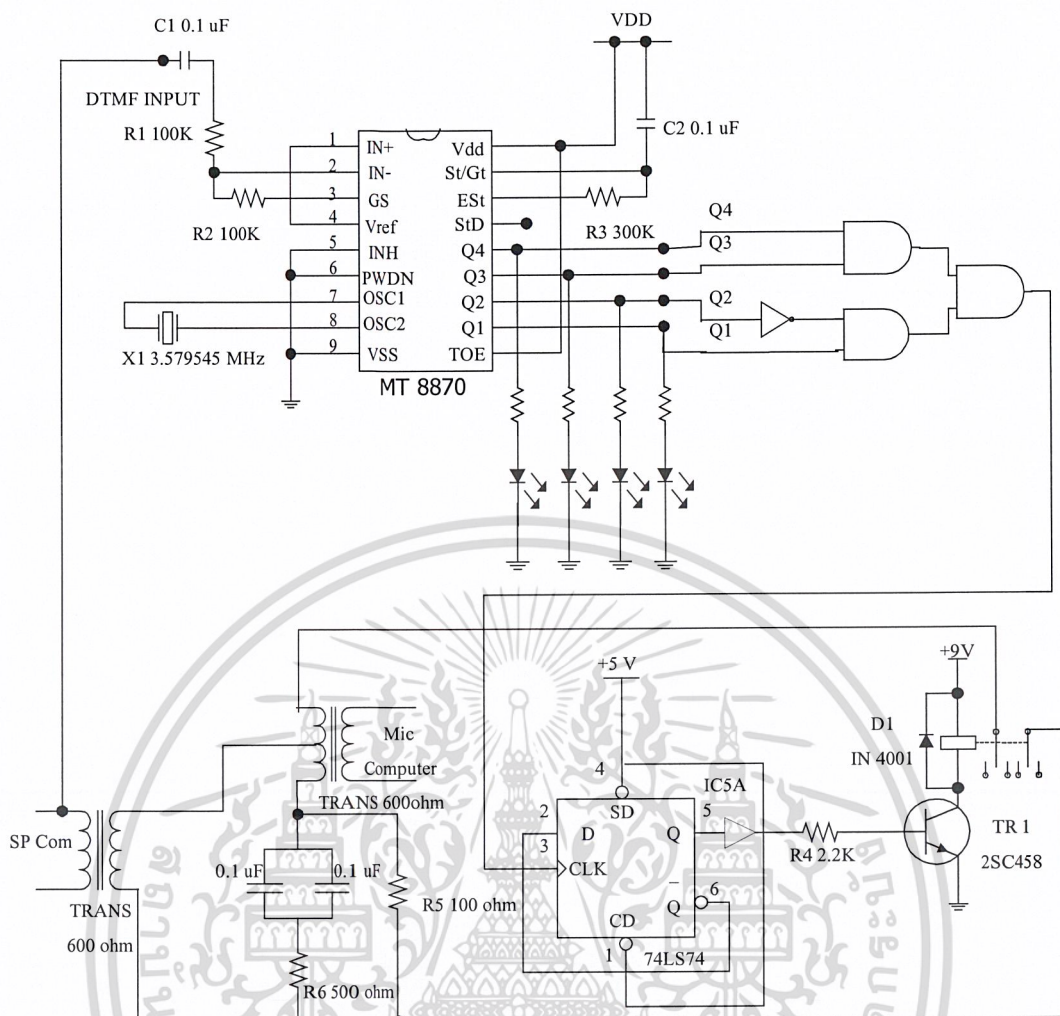


รูปที่ 3.7 วงจรทำสัญญาณค้าง

จากรูปที่ 3.7 วงจรเป็นรูปวงจรทำสัญญาณค้างหรือวงจรที่ทำหน้าที่เหมือน T Flip-flop นั่นเอง ดังแสดงในรูปที่ 3.7

เราก็จะนำวงจรนี้มาต่อเข้า โดยอินพุตจะรับมาจากเอาต์พุตของวงจรถอดรหัสสัญญาณ BCD เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่ในเชิงพาณิชย์ การค้า 1101 และเอาต์พุตก็จะต่อไปยังวงจรควบคุมรีเลย์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบและการสร้างส่วนซอฟต์แวร์

3.2.1 หลักการออกแบบในส่วนของข้อมูลเสียง

การออกแบบตัวโปรแกรมจะแบ่งออกเป็น 3 ส่วน คือ

1. การรับและส่งข้อมูลผ่านระบบเน็ตเวิร์ก
2. การจัดการเกี่ยวกับบันทึกและเล่นเสียง
3. การจัดการในส่วนของสัญญาณ DTMF

การทำงานของโครงการในส่วนของการรับส่งเสียงดังแสดงได้ดังรูปที่ 3.9

3.2.2 ขั้นตอนการพัฒนา

ในการพัฒนาเราได้ใช้ภาษา Visual Basic เป็นเครื่องมือในการพัฒนา ซึ่งมีความสามารถในการสร้าง แอปพลิเคชันเพื่อใช้งานได้อย่างรวดเร็ว โดยเน้นส่วนติดต่อผู้ใช้ที่เป็นแบบกราฟิก และการสร้างแอปพลิเคชันแบบไคลเอ็นท์/เซิร์ฟเวอร์ นอกจากนี้ยังสนับสนุนการสร้างแอปพลิเคชันสำหรับอินเทอร์เน็ตด้วย จากการศึกษาพบว่า Visual Basic มีการเขียนโปรแกรมในลักษณะ Event Driven คือ การเขียนโปรแกรมในลักษณะที่ว่า ถ้าเหตุการณ์นี้เกิดขึ้น เราจะดำเนินการอย่างไร หรือ จะพูดอีกอย่างว่า เมื่อเหตุการณ์ต่างๆ ดำเนินไป เราต้องคิดว่าจะจัดการกับแต่ละเหตุการณ์ที่เกิดขึ้นอย่างไร เมื่อคิดออกก็มาเขียนโปรแกรมรองรับกับเหตุการณ์ต่างๆ เหล่านั้น

3.2.2.1 การพัฒนาในส่วนของการรับส่งข้อมูลผ่านระบบเน็ตเวิร์ก (Network Connection)

เราได้เลือกใช้โพรโตคอล TCP/IP เป็นโพรโตคอลที่ใช้ในการรับ และ ส่งข้อมูลเสียงในระบบเน็ตเวิร์ก เนื่องจาก TCP/IP นั้นจะเป็นการติดต่อ แบบ คอนเน็คชันเบสเซอร์ส ซึ่ง คอมพิวเตอร์ด้านรับและด้านส่งจะติดต่อกันอยู่ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบก็คล้ายกับระบบโทรศัพท์ที่ต้องติดต่อกันให้ได้ก่อน และใช้ช่องสัญญาณนั้นในการติดต่อสื่อสารกัน ดังนั้นเราจึงเขียนโปรแกรมขึ้นมาจัดการกับการรับและส่งข้อมูลเสียงบน โพรโตคอล TCP/IP ซึ่งสามารถทำได้โดยใช้

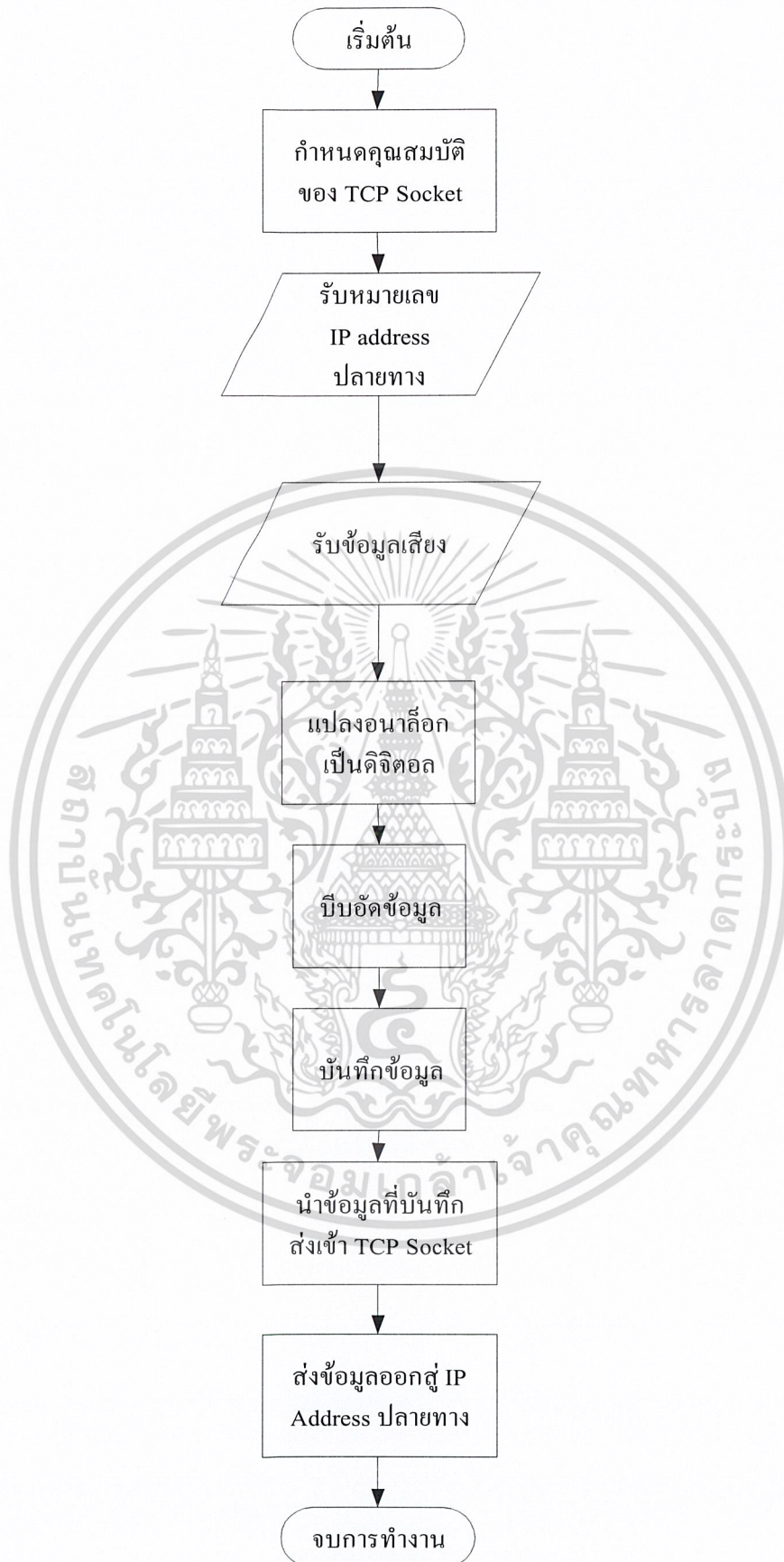
1. การใช้ไฟล์ Winsock.bas ซึ่งข้างในจะประกาศคำสั่งในการเรียกใช้ฟังก์ชันใน Winsock.dll โดยการประกาศนี้จะประกาศออกมาให้อยู่ในรูปของตัวแปรที่ Visual Basic สนับสนุน และในการเรียกใช้ก็จะทำเหมือนกับการเรียก Function DLL จาก API ของ Windows โดยทั่วไป

2. การใช้ Control Winsock ซึ่งเป็น Control ชุด ของ Active X Internet Control Pack ซึ่งทาง Microsoft ได้เตรียมมาผู้พัฒนาโปรแกรมที่เกี่ยวกับ Internet ได้ใช้กัน โดยวิธีนี้จะช่วยให้ผู้พัฒนาแอปพลิเคชันที่เกี่ยวกับการเชื่อมต่อเข้าสู่เครือข่าย Internet ไม่ต้องเรียนรู้ถึงการ Control การใช้ Windows Socket อย่างลึกซึ้ง ซึ่งจะช่วยประหยัดเวลาเป็นอย่างมาก

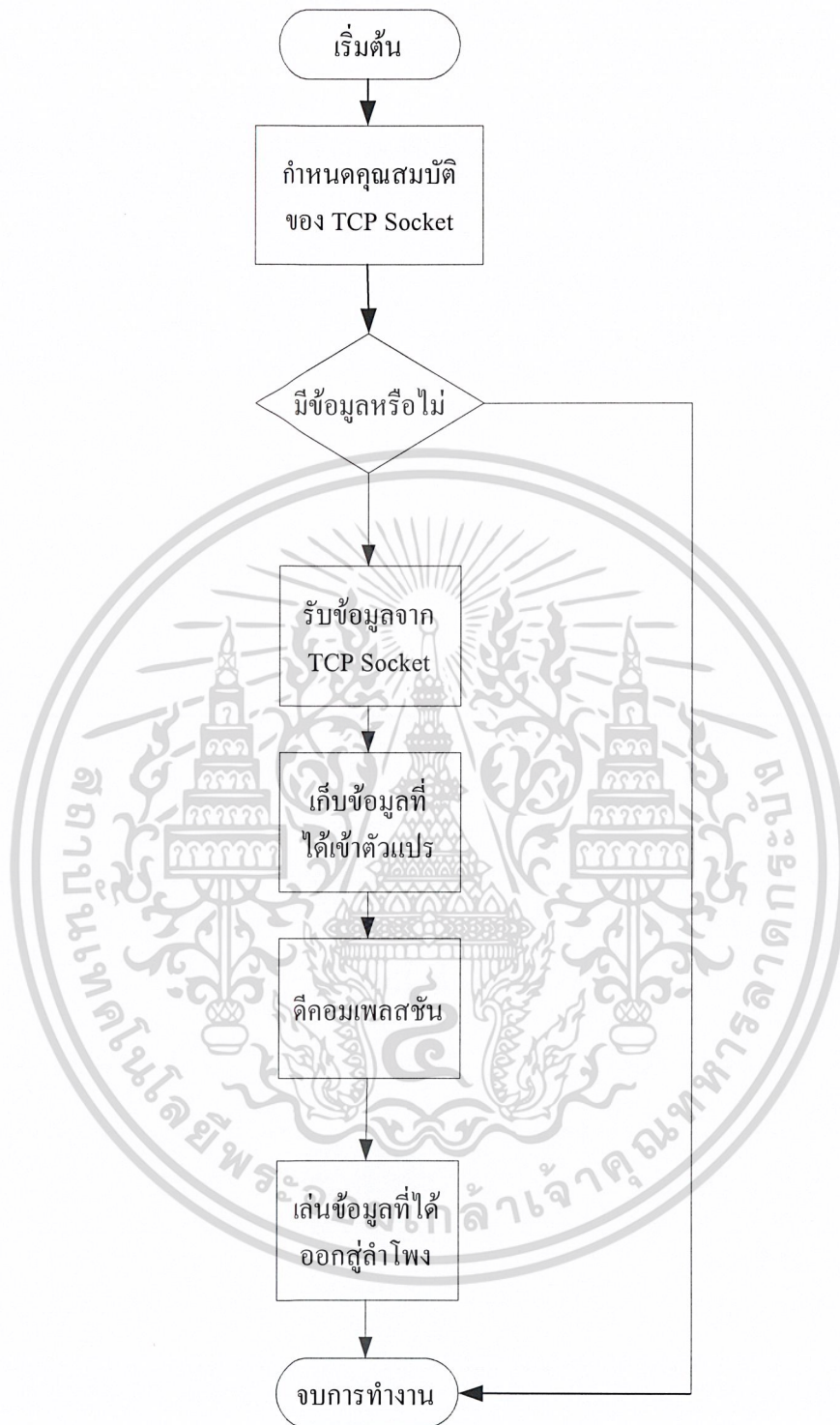
3.2.2.2 การพัฒนาในส่วนของการจัดการเรื่องเสียงบน Windows

ในส่วนการจัดการเรื่องเสียงบน Windows โดยการติดต่อกับการ์ดเสียงโดยตรงนี้ทำได้ยากมาก ดังนั้นในการจัดการเรื่องเกี่ยวกับเสียง เราจะต้องกระทำผ่านไดร์เวอร์ของการ์ดเสียงนั้นๆ ซึ่งตัวโปรแกรม Windows เองก็ออกแบบมาให้ผู้พัฒนากระทำผ่านตัวไดร์เวอร์ของ Windows แต่ในความจริงแล้วเราสามารถที่จะจัดการระบบเสียงได้หลายวิธี ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.9 แสดงผังการทำงานของ โปรแกรมในการส่งเสียงผ่านระบบอินเทอร์เน็ต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงผังการทำงานในส่วนการรับเสียงผ่านระบบอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การเรียกใช้โดยผ่าน MCI (Multimedia Command String) ซึ่งวิธีนี้เป็นวิธีที่ง่ายที่สุด แต่่ววิธีนี้ จะไม่สามารถเข้าจัดการระบบเสียงของระบบได้อย่างเต็มที่และมีประสิทธิภาพเท่าที่ควร

2. การเรียก ฟังก์ชันระดับต่ำจากตัวระบบของ Windows เอง วิธีการนี้จะสามารถเข้าถึงฟังก์ชัน พื้นฐานในระดับลึกที่จะเข้าไปจัดการได้อย่างมีประสิทธิภาพตามที่ไดร์เวอร์ที่ติดตั้งอยู่จะสามารถทำได้

3.2.2.3 การพัฒนาในส่วนของการจัดการเรื่อง Compression เสียงบน Windows

ในส่วนนี้เราได้ใช้ประโยชน์ที่ได้จากไดร์เวอร์ของการ์ดเสียง และการเรียกใช้ API ซึ่งเป็น ฟังก์ชันระดับต่ำจากตัวระบบของ Windows โดยการเรียกใช้ตัวจัดการเกี่ยวกับการ Compression/Decompression ที่ชื่อว่า Audio Compression Manager (ACM) ซึ่ง ACM นี้กระทำโดยสัมพันธ์กับไดร์เวอร์ของการ์ดเสียงที่ติดตั้งอยู่ในระบบ ตัว Audio Compression Manager จะกระทำการ Compression/Decompression จากรูปแบบหนึ่งเป็นรูปแบบหนึ่ง เช่น จาก GSM 610 เป็น PCM หรือกลับกัน

จากรูปที่ 3.9 และรูปที่ 3.10 โปรแกรมทั้งทางด้านฝั่งไคลเอ็นท์ และ ฝั่งเซิร์ฟเวอร์จำเป็นต้องมีโปรแกรมในทั้ง 2 ส่วนนี้ ซึ่งก็คือ ส่วนการรับ และ ส่วนการส่งเสียง

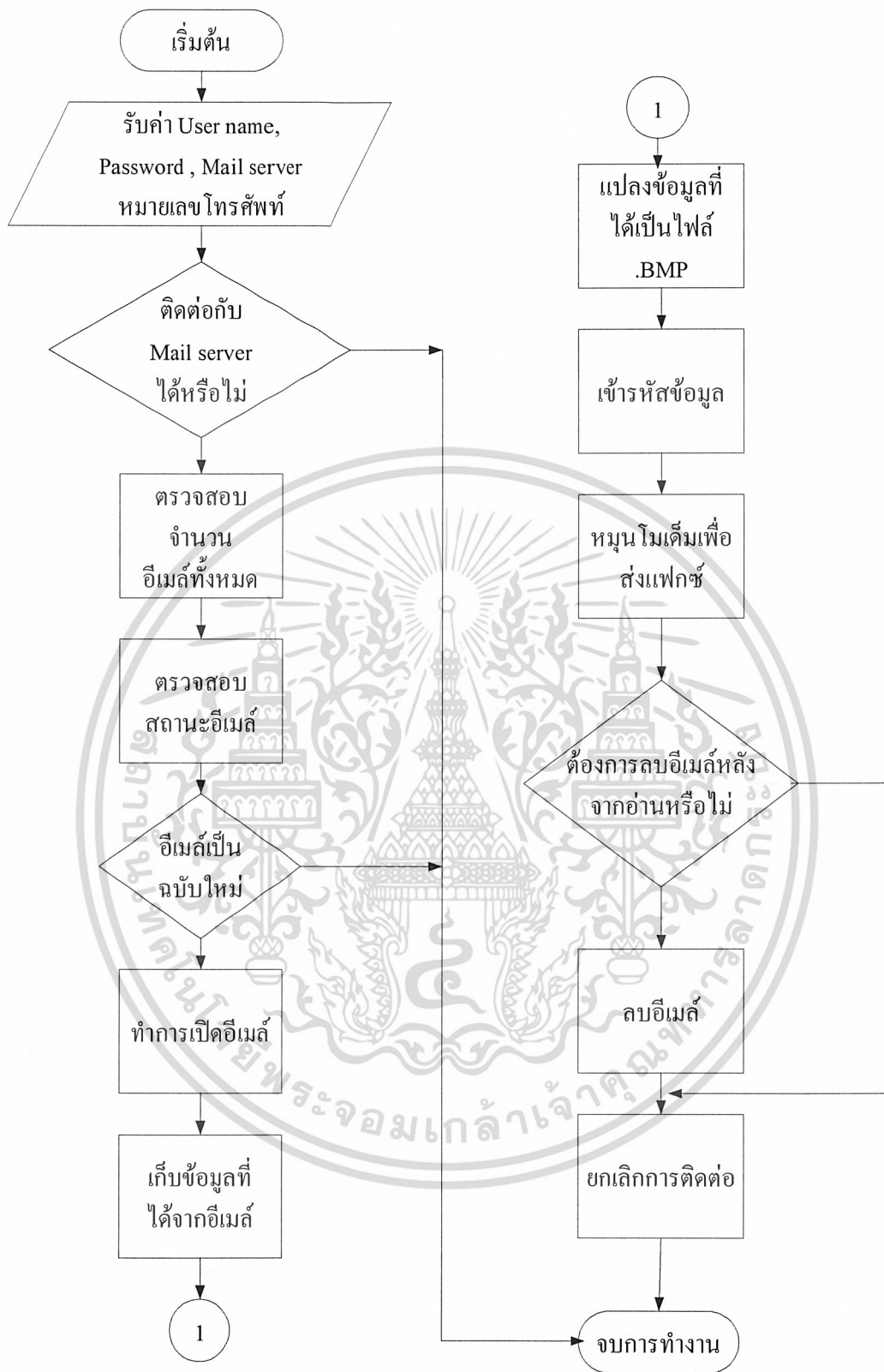
3.3 โครงสร้างโดยรวมของโปรแกรมในส่วนของคุณข้อมูลเสียง

การทำงานของโปรแกรมจะมีลักษณะการทำงานเป็นการสื่อสารแบบ Full - Duplex ตัวโปรแกรมจะใช้โพรโตคอล TCP/IP ในการนำข้อมูลเสียงไปถึงผู้รับปลายทาง ซึ่งการใช้โพรโตคอล TCP/IP นั้นกำหนดไว้ว่าเครื่องคอมพิวเตอร์ที่ใช้โพรโตคอลนี้จะมี IP Address ที่แน่นอนเป็นของตนเองและไม่ซ้ำกับใคร ในเครือข่ายเน็ตเวิร์ค ซึ่งเปรียบเสมือนกับเบอร์โทรศัพท์ นั่นเอง โปรแกรมจะทำการบันทึกเสียงแล้วทำการ Compression ตามที่กำหนดจากนั้นก็ส่งเข้าไปแพ็คเกจข้อมูลให้อยู่ในรูปแบบของ TCP/IP แล้วจึงทำการส่งออกไป และในทางกลับกันก็จะนำแพ็คเกจของ TCP/IP ที่ส่งมาถึงตนเองทำการ Decompression ข้อมูลที่เป็นเสียงออกมาแล้วทำการส่งออกไปเล่นเสียงออกสู่ลำโพง

การออกแบบและการสร้างส่วนของการส่งแพ็คเกจ

โครงงานในส่วนนี้จะเชื่อมต่อระหว่าง 2 เครื่องข่าย คือ เครื่องข่ายอินเทอร์เน็ต และเครื่องข่ายโทรศัพท์ ซึ่งผ่านคอมพิวเตอร์ที่มีโปรแกรมทำหน้าที่ในการรับจดหมายอิเล็กทรอนิกส์ ซึ่งใช้โพรโตคอล POP3 ในการติดต่อกับเมลล์เซิร์ฟเวอร์ ซึ่งเราสามารถเลือกได้ว่าจะทำการลบจดหมายอิเล็กทรอนิกส์หลังจากการอ่านหรือไม่ และในโปรแกรมจะทำการอ่านเฉพาะจดหมายอิเล็กทรอนิกส์ที่ยังไม่ได้เคยเปิดอ่านมาก่อน กล่าวคือจะอ่านเฉพาะจดหมายที่ยังไม่เคยเปิดอ่านนั่นเอง ซึ่งสามารถทำการตรวจสอบได้จากส่วนของสถานะ (Status): ซึ่งข้อมูลหลังสถานะจะต้องไม่ใช่ RO (Status: RO) ซึ่งแสดงว่าจดหมายอิเล็กทรอนิกส์ฉบับนี้ได้อ่านไปแล้ว โดยเมื่อได้ข้อมูลจากจดหมายอิเล็กทรอนิกส์มาแล้วจะทำการเข้ารหัสข้อมูลที่ได้แล้วส่งต่อไปยังแพ็คเกจโมเด็มเพื่อทำการส่งแพ็คเกจไปยังปลายทาง หลังจากการทำงานเสร็จก็จะทำการยกเลิกการติดต่อกับเมลล์เซิร์ฟเวอร์ การทำงานโดยรวมดังแสดง ได้ดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงผังการทำงานของโปรแกรมในส่วนของแฟกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบส่วนติดต่อกับผู้ใช้ (User – Interface) และการใช้งานโปรแกรม

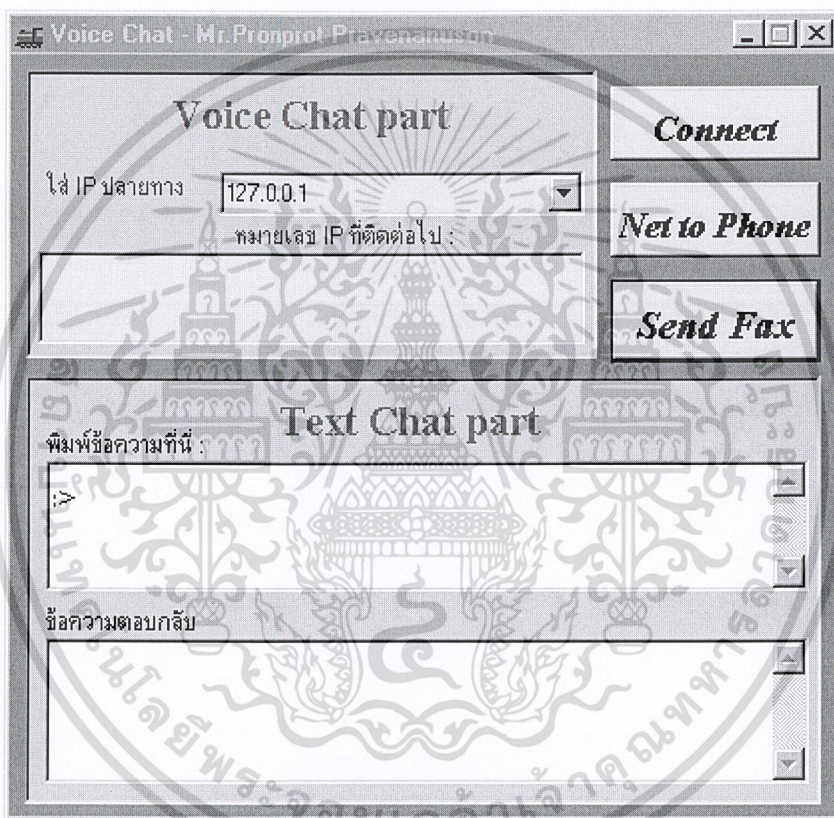
3.4.1 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface) และการใช้งานโปรแกรมส่วนของการส่งข้อมูลเสียง

ส่วนของการรับส่งข้อมูลเสียงผ่านอินเทอร์เน็ต แบ่งออกเป็น 2 ฝั่ง

1. ฝั่งไคลเอ็นท์ (Client Path)
2. ฝั่งเซิร์ฟเวอร์ (Server Path)

โปรแกรมฝั่งไคลเอ็นท์ ประกอบด้วยโปรแกรมต่างๆ ดังนี้

โปรแกรม Client Send ซึ่งมีลักษณะแสดงดังรูปที่ 3.12



รูปที่ 3.12 แสดงโปรแกรม Client Send

การใช้งานโปรแกรมผู้ใช้จะต้องใส่หมายเลข IP Address ปลายทางที่ต้องการติดต่อก่อนแล้วจึงทำการกดปุ่ม Connect เพื่อทำการเริ่มการติดต่อกับหมายเลข IP Address ปลายทางโดยการใส่หมายเลข IP Address ปลายทางนั้นมีรูปแบบมาตรฐานดังนี้ xxx.xxx.xxx.xxx กล่าวคือเลขฐานสิบ 0 – 255 ซึ่งก็คือ 8 บิต ในเลขฐานสอง ซึ่งแต่ละ 8 บิต จะขึ้นด้วย จุด (.) ซึ่งรวมกันได้ทั้งหมด 32 บิต ตามมาตรฐานที่กำหนด

ปุ่ม Net To Phone ใช้สำหรับกรณีที่ต้องการติดต่อกับโทรศัพท์ภายในท้องถิ่นของหมายเลข IP Address ปลายทางที่เราติดต่อดังอยู่ แต่การจะใช้นั้นต้องให้ทางฝั่ง เซิร์ฟเวอร์ทำการเปิดโปรแกรม

Server Dial ขึ้นมาก่อน และทางฝั่งเซิร์ฟเวอร์ต้องทำการต่ออุปกรณ์เชื่อมต่อกับเครือข่ายโทรศัพท์ไว้ให้เรียบร้อยก่อน จึงสามารถทำการต่อออกไปยังหมายเลขโทรศัพท์ปลายทางในเขตชุมสายที่หมายเลข IP Address ปลายทางนั้นอยู่ได้

ปุ่ม Send Fax ใช้สำหรับเมื่อต้องการส่งแฟกซ์โดยเมื่อกดปุ่มจะทำการเปิดโปรแกรมที่ใช้สำหรับการติดต่อกับเมลล์เซิร์ฟเวอร์และนำเมลล์มาทำการส่งแฟกซ์

โปรแกรม Client Send สามารถที่จะทำการพิมพ์ข้อความสื่อสารกันได้ด้วย ซึ่งเหมือนกับโปรแกรมประเภท Chat ทั่ว ๆ ไป

โปรแกรม Client Dial ซึ่งมีลักษณะดังรูปที่ 3.13



รูปที่ 3.13 แสดงโปรแกรม Client Dial

โปรแกรม Client Receive จะทำการรับสัญญาณข้อมูลเสียงที่ได้รับจาก TCP Socket มาทำการ Decompression และ แปลงจากสัญญาณดิจิทัลคอลเป็น สัญญาณ อนาล็อก ซึ่งกระทำได้โดยความสามารถของการ์ดเสียง แล้วทำการเล่นเสียงออกสู่ลำโพง

โปรแกรม Client Receive มีลักษณะดังรูปที่ 3.14

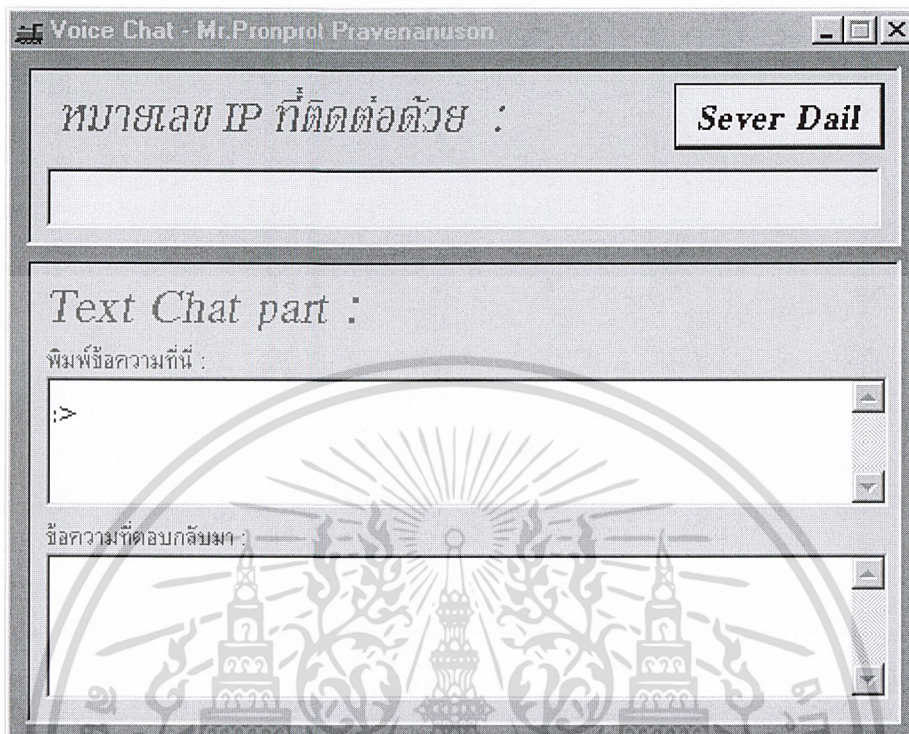


รูปที่ 3.14 แสดงโปรแกรม Client Receive

ถ้าหากต้องการที่จะยกเลิกการติดต่อก็เพียงแคเปิดโปรแกรมเท่านั้น โปรแกรมจะทำการยกเลิกการติดต่อทั้งหมด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทางฝั่งเซิร์ฟเวอร์ ประกอบด้วยโปรแกรมต่างๆ ดังนี้

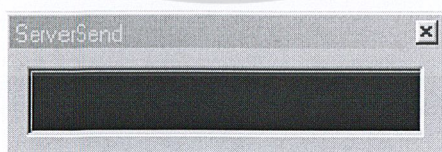
โปรแกรม Server Receive ซึ่งมีลักษณะดังรูปที่ 3.15



รูปที่ 3.15 แสดงโปรแกรม Server Receive

โปรแกรม Server Receive นั้นจะทำการรอการติดต่อจากโปรแกรม Client Send ที่จะทำการร้องขอการติดต่อเข้ามา ซึ่งโปรแกรม Server Receive นั้นทำหน้าที่ในการรับข้อมูลจาก TCP Socket แล้วทำการ Decompression และแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อกแล้ว ทำการเล่นเสียงออกสู่ลำโพง ซึ่งจะเหมือนกับโปรแกรม Client Receive

โปรแกรม Server Send ซึ่งมีลักษณะดังรูปที่ 3.16



รูปที่ 3.16 แสดงโปรแกรม Server Send

โปรแกรมนี้จะทำการส่งข้อมูลเสียงที่ได้ผ่านการ Compression แล้วเข้าสู่ TCP Socket เพื่อส่งออกไปยังหมายเลข IP Address ปลายทางที่ถูกระบุไว้
 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การนำเอกสารนี้ไปใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 โปรแกรม Server Dial มีลักษณะดังแสดงในรูปที่ 3.17
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นที่ ผมได้เห็นแต่เพียงเนื้อหาแต่เพียงอย่างเดียวจึงอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

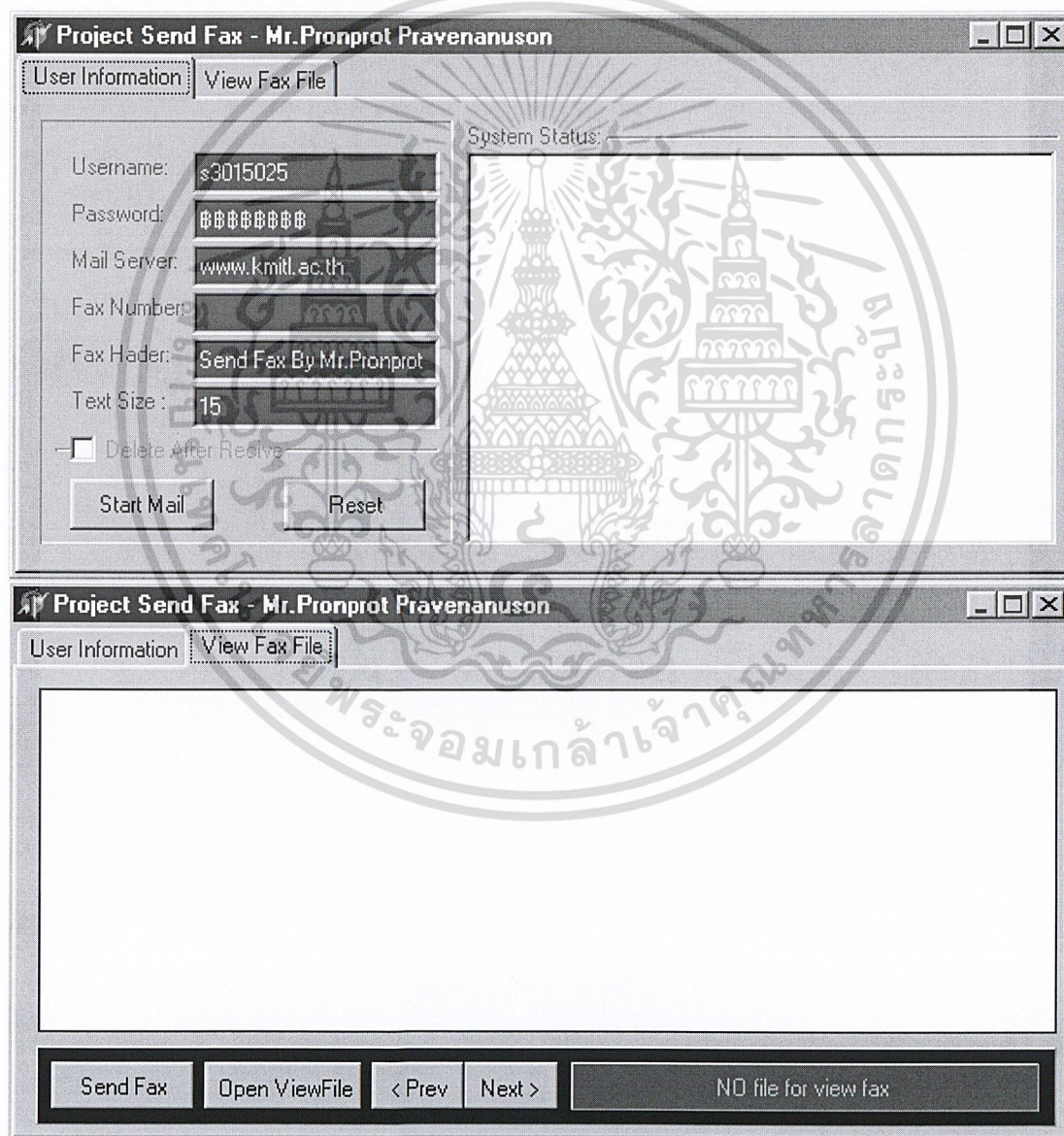
ServerDial

รูปที่ 3.17 แสดงโปรแกรม Server Dial

โปรแกรม Server Dial นี้จะทำหน้าที่ในการเล่นเสียง DTMF ตามข้อมูลที่ส่งมาจากโปรแกรม Client Dial เพื่อทำการต่อไปยังโทรศัพท์ภายในท้องถิ่นที่อยู่ของเครื่องที่เซิร์ฟเวอร์อยู่

3.4.2 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface) และการใช้งานโปรแกรมส่วนของการส่งแฟกซ์

หน้าตาของโปรแกรมในการติดต่อกับเมลเซิร์ฟเวอร์เพื่อนำข้อความในจดหมายอิเล็กทรอนิกส์มาทำการส่งต่อไปยังหมายเลขแฟกซ์ปลายทาง ดังแสดงได้ดังรูปที่ 3.18



รูปที่ 3.18 แสดงโปรแกรมในการรับข้อมูลจากจดหมายอิเล็กทรอนิกส์เพื่อมาทำการส่งแฟกซ์

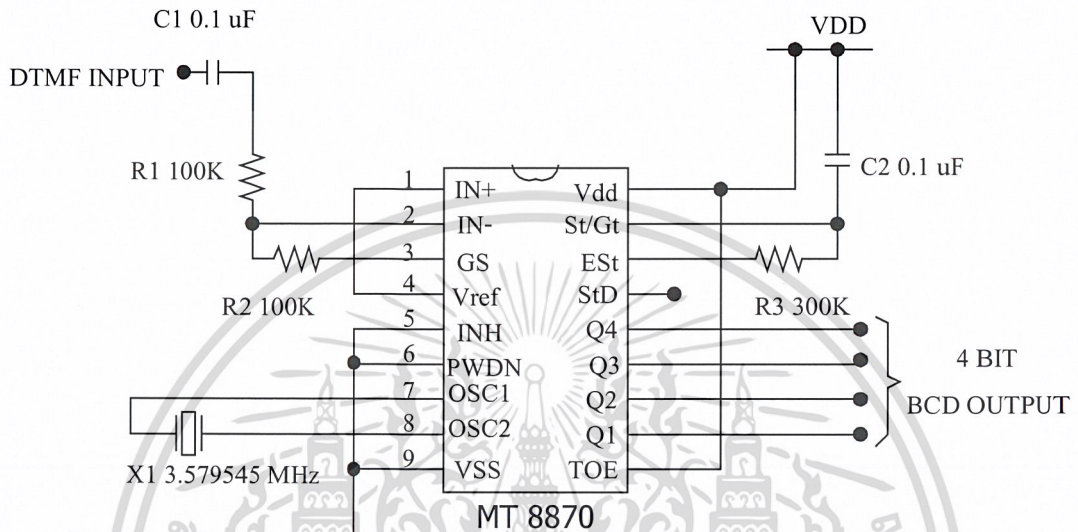
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองและผลการทดลองในส่วนของฮาร์ดแวร์

การทดลองวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ (DTMF Decoder)



รูปที่ 4.1 แสดงวงจรถอดรหัสสัญญาณดีทีเอ็มเอฟ

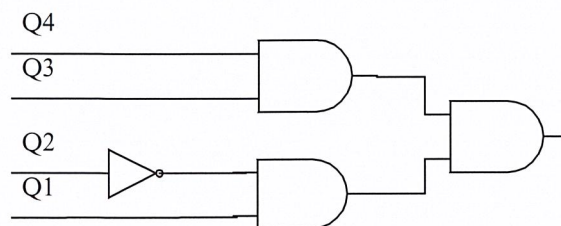
ผลการทดลองที่ได้จากการทดลอง

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
A	1101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

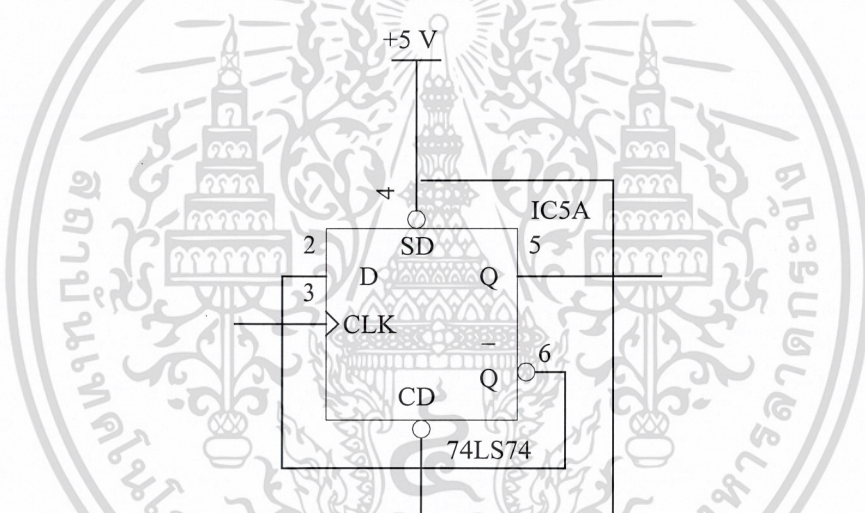
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองวงจรถอดรหัสสัญญาณ A เพื่อนำไปควบคุมการทำงานของรีเลย์เพื่อทำการยกหูหรือวางหูโทรศัพท์ ดังแสดงดังรูปที่ 4.2



รูปที่ 4.2 แสดงวงจรถอดรหัสสัญญาณดิจิตอลที่เอ็มเอฟ A

การทดลองก็นำสัญญาณเอาต์พุตที่ได้จากการทดลองในขั้นแรกมาป้อนเข้าวงจรแล้ววัดสัญญาณที่เอาต์พุตของวงจร จะได้ผลว่าจะไม่มีเอาต์พุตออกนอกจากกรณีที่สัญญาณอินพุตเป็น สัญญาณดิจิตอลที่เอ็มเอฟ A เท่านั้น และนำสัญญาณเอาต์พุตที่ได้ไปใช้ในการทริกเป็นคล็อกให้กับ D Flip - Flop ซึ่งใช้เป็นวงจรค้างสัญญาณซึ่งในที่นี้เราใช้ D Flip - Flop มาทำการต่อเป็น T Flip - Flop ซึ่งแสดงดังรูปที่ 4.3



รูปที่ 4.3 แสดงการวางวงจรค้างสัญญาณ

4.2 การทดลองและผลการทดลองในส่วนของซอฟต์แวร์

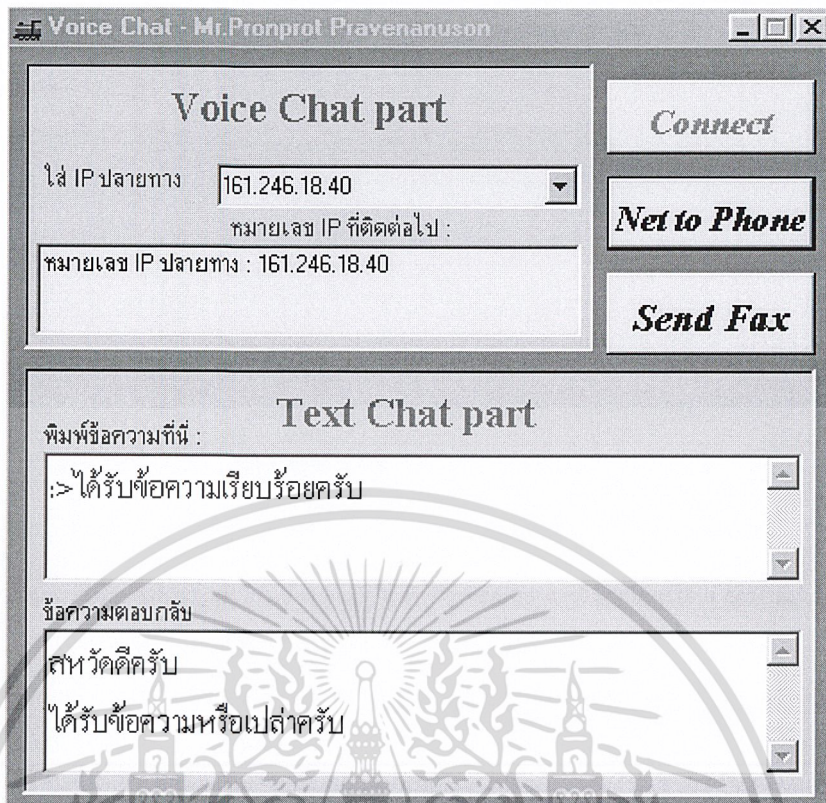
1. ทำการเปิดโปรแกรม Client Send และ โปรแกรม Client Recive ขึ้นมาในคอมพิวเตอร์เครื่องที่ต้องการเป็นผู้ร้องขอการติดต่อ (เครื่องนี้ทำหน้าที่เป็นไคลเอ็นท์)

2. ทำการเปิดโปรแกรม Server Send และ โปรแกรม Server Recive ขึ้นมาที่เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์

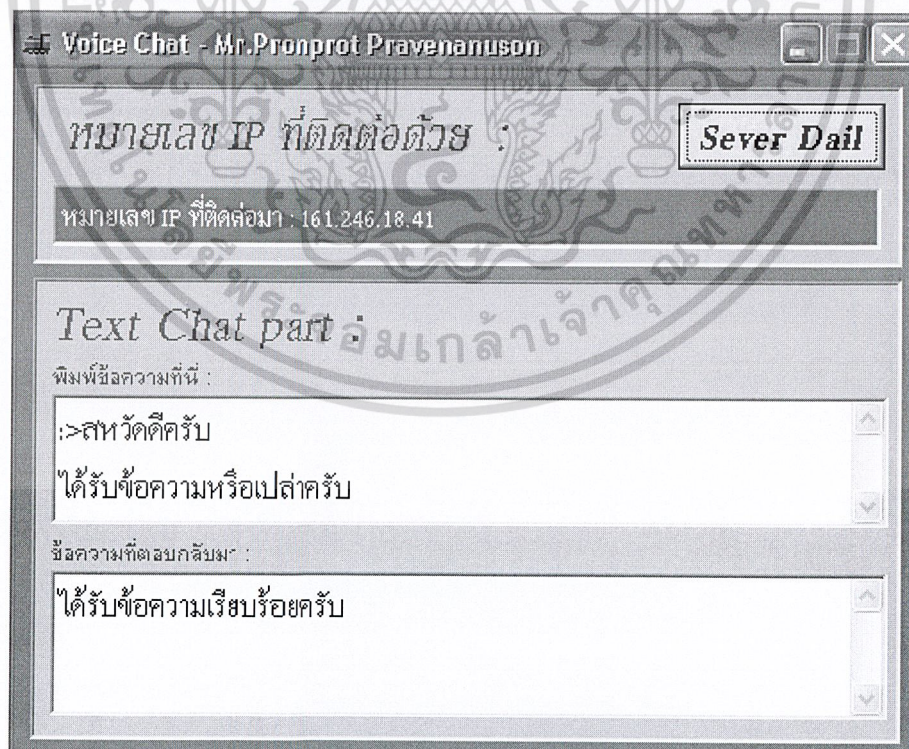
3. ที่เครื่องไคลเอ็นท์ทำการป้อนหมายเลข ไอพี แอดเดรสปลายทางที่ต้องการติดต่อ แล้วทำการกดปุ่ม Connect

4. ทำการพิมพ์ข้อความส่งถึงกัน ผลที่ได้ดังแสดงดังรูปที่ 4.4 และรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

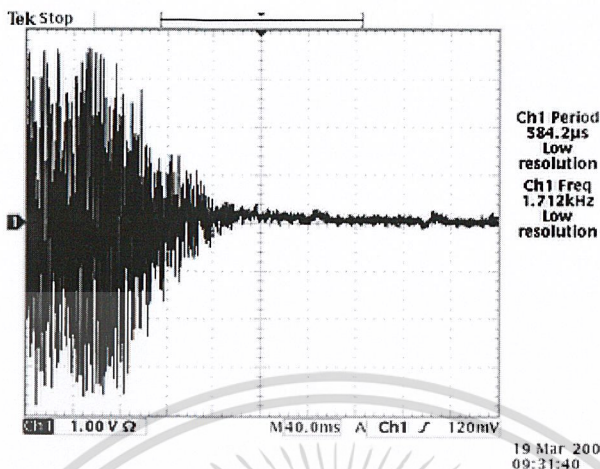


รูปที่ 4.4 แสดงผลที่ได้จากการพิมพ์ข้อความ ได้ตอบกันที่ฝั่งไคลเอ็นท์

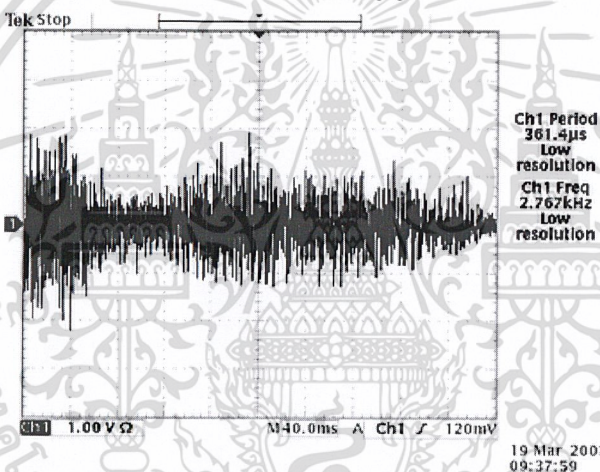


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.5 แสดงผลที่ได้จากการพิมพ์ข้อความ ได้ตอบกันที่ฝั่งเซิร์ฟเวอร์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

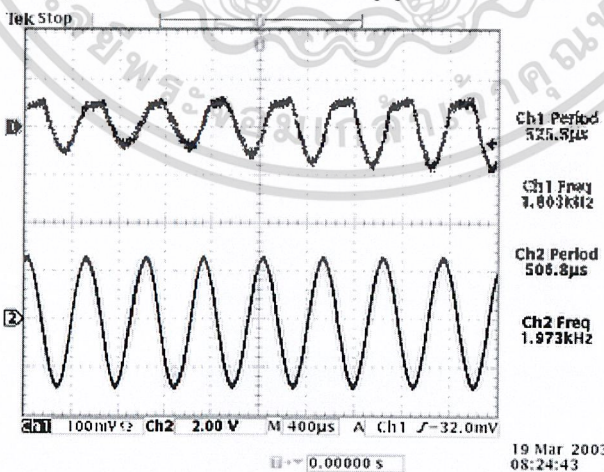
5. ทำการทดลองสนทนาโต้ตอบกัน ผลที่ได้คือ เราสามารถที่จะสนทนาพูดคุยกันได้ ผลการที่ตั้งแสดงดังรูปที่ 4.6



รูปที่ 4.6 แสดงสัญญาณเสียงที่วัดได้จากการส่งสัญญาณเสียงจากฝั่งไคลเอ็นท์มาฝั่งเซิร์ฟเวอร์



รูปที่ 4.7 แสดงสัญญาณเสียงที่วัดได้จากการส่งสัญญาณเสียงจากฝั่งเซิร์ฟเวอร์มาฝั่งไคลเอ็นท์



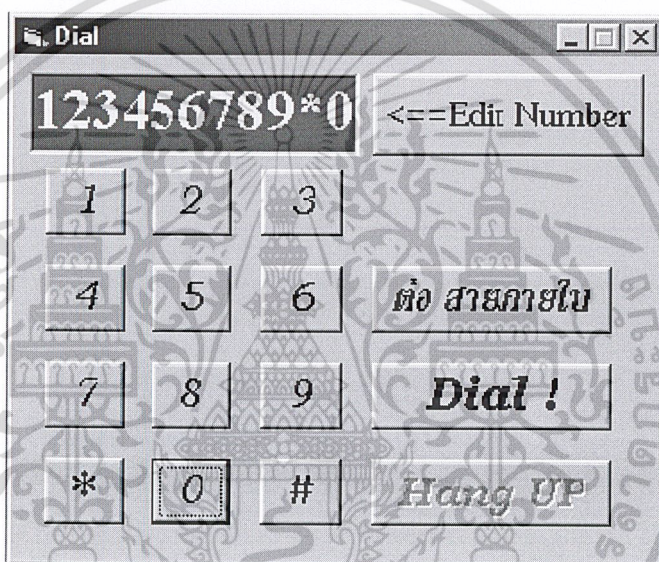
รูปที่ 4.8 แสดงคุณภาพของสัญญาณที่ Speaker Output เมื่อป้อนสัญญาณความถี่ 2 kHz ที่

เอกสารนี้เป็น Microphone Input หนึ่งการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทำการเปิดโปรแกรม Server Dial ขึ้นมาที่ฝั่งเซิร์ฟเวอร์ และเปิดโปรแกรม Client Dial ที่ฝั่งไคลเอ็นท์

7. ทำการต่ออุปกรณ์ต่อพ่วงเพื่อทำหน้าที่ในการเชื่อมต่อกับเครือข่ายโทรศัพท์เข้ากับคอมพิวเตอร์เครื่องที่เป็นเซิร์ฟเวอร์

8. ทำการต่อโทรศัพท์จากที่ฝั่งไคลเอ็นท์โดยการกด หมายเลขปลายทางแล้วกด Dial เพื่อที่จะส่งตัวเลขที่เราต้องการจะติดต่อออกไป ที่ฝั่งเซิร์ฟเวอร์ซึ่งตอนนี้การกดเสียงจะส่งสัญญาณไปให้ที่ตัวเครื่องต่อโทรศัพท์ทำการยกหู และหมุนหมายเลขด้วยเทคนิคของ DTMF เมื่อทำการติดต่อกับเครื่องโทรศัพท์ปลายทางได้ก็สนทนากันได้ตามปกติ เพียงแต่ว่าสัญญาณเสียงที่ได้ยินนั้นจะ เกิดการหน่วงเวลาซึ่งก็เป็นปกติของข้อมูลที่ส่งผ่านอินเทอร์เน็ตก็ต้องมีเวลาในการเดินทาง



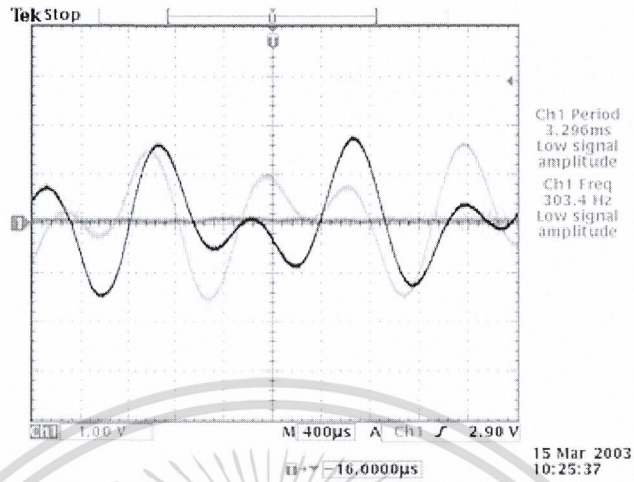
รูปที่ 4.9 แสดงหน้าจอโปรแกรม Client Dial ขณะทำการกดหมายเลข



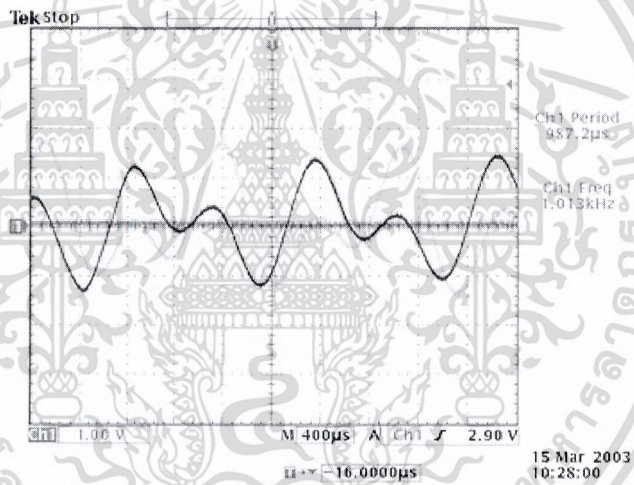
รูปที่ 4.10 แสดงหน้าจอโปรแกรม Server Dial ขณะรับหมายเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

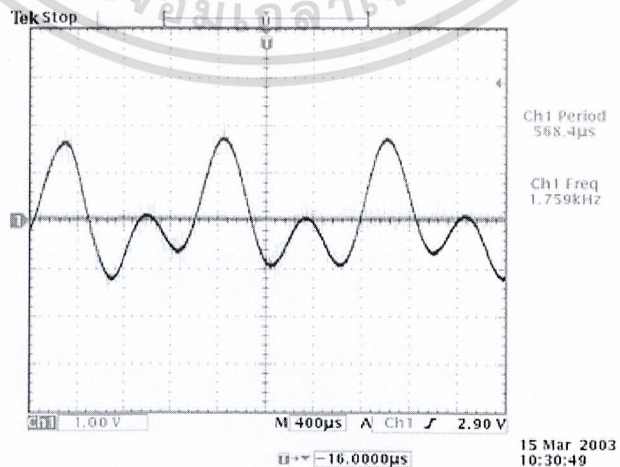
ผลจากการทดลองกหนดหมายเลขต่างๆ ดังแสดงได้ดังนี้



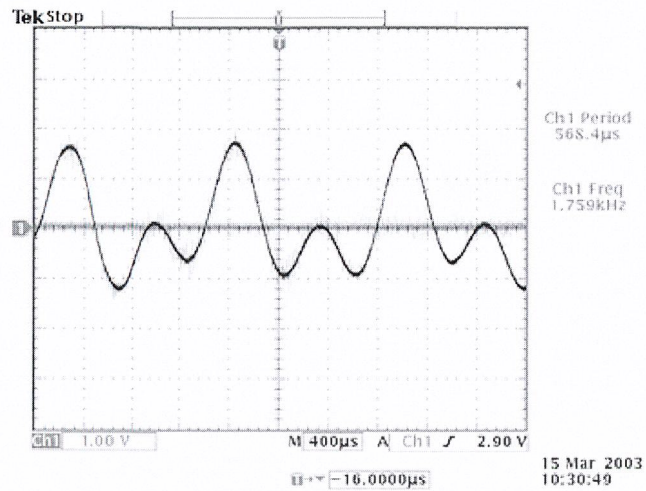
รูปที่ 4.11 แสดงสัญญาณการกคคี่หมายเลข 1



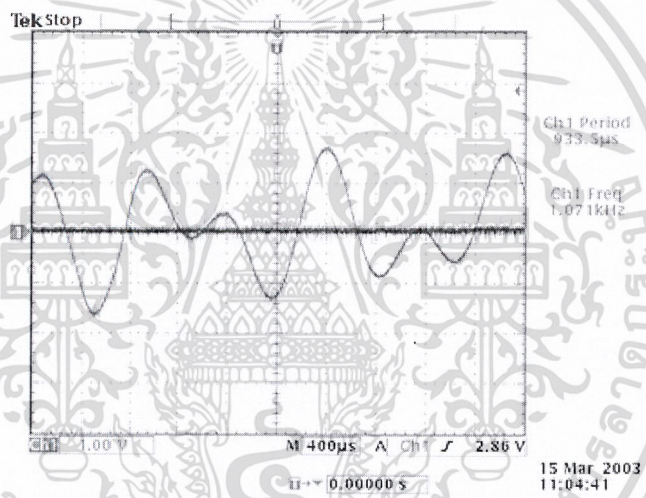
รูปที่ 4.12 แสดงสัญญาณการกคคี่หมายเลข 2



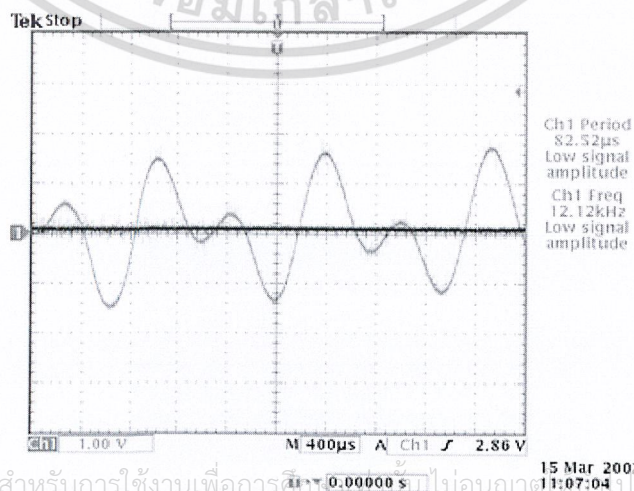
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.13 แสดงสัญญาณการกคคี่หมายเลข 3 เติหน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



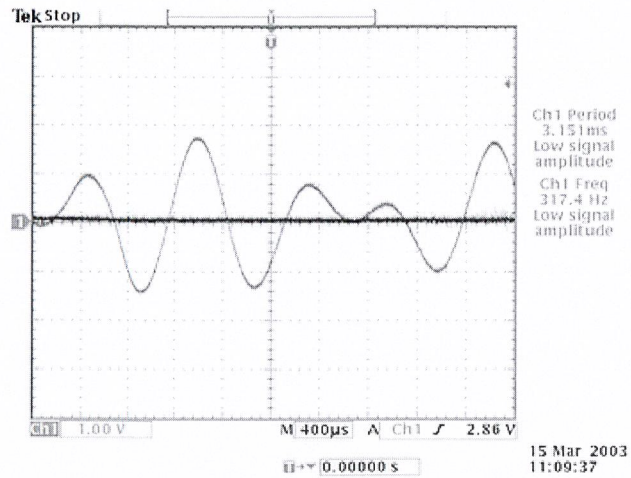
รูปที่ 4.14 แสดงสัญญาณการกดคีย์หมายเลข 4



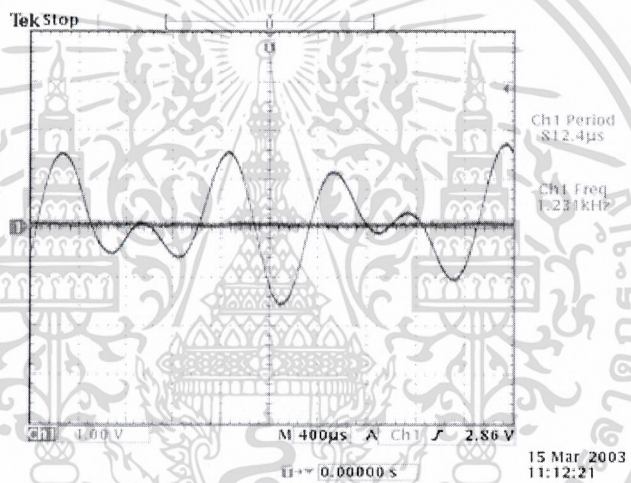
รูปที่ 4.15 แสดงสัญญาณการกดคีย์หมายเลข 5



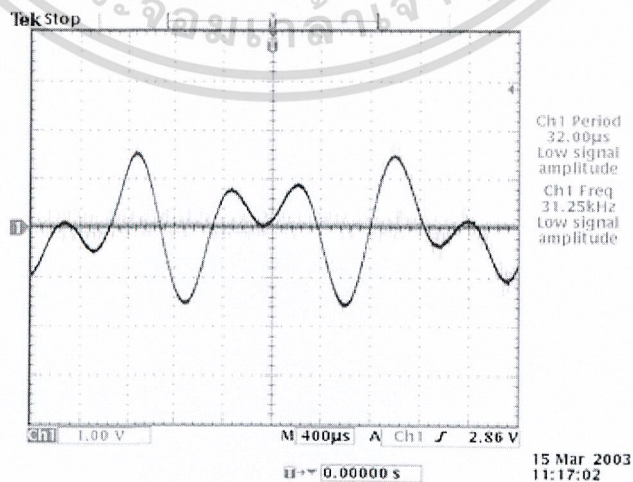
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาระดับปริญญาตรีเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.16 แสดงสัญญาณการกดคีย์หมายเลข 6
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงบนสื่อออนไลน์ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



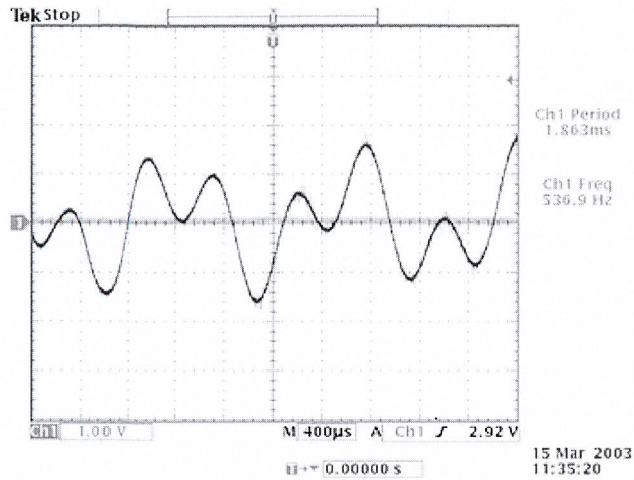
รูปที่ 4.17 แสดงสัญญาณการกคคีย์หมายเลข 7



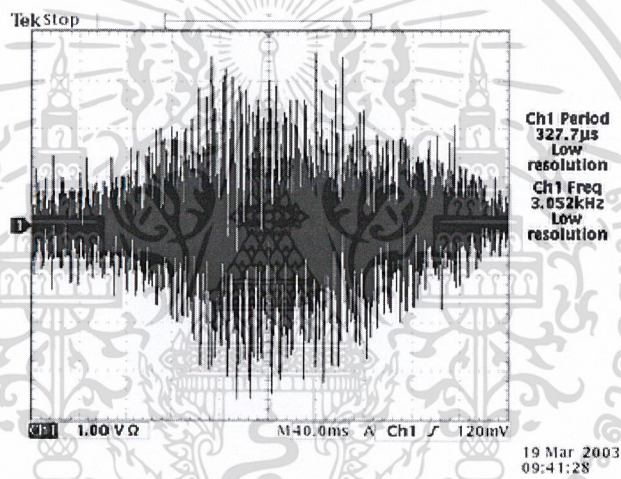
รูปที่ 4.18 แสดงสัญญาณการกคคีย์หมายเลข 8



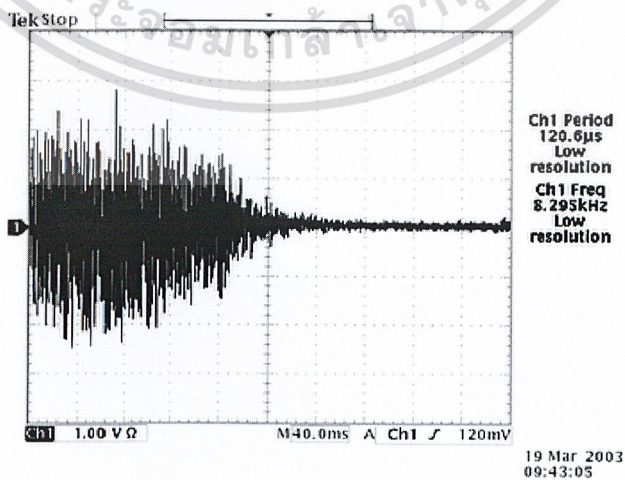
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.19 แสดงสัญญาณการกคคีย์หมายเลข 9 เตให้หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 แสดงสัญญาณการกดคีย์ Dail

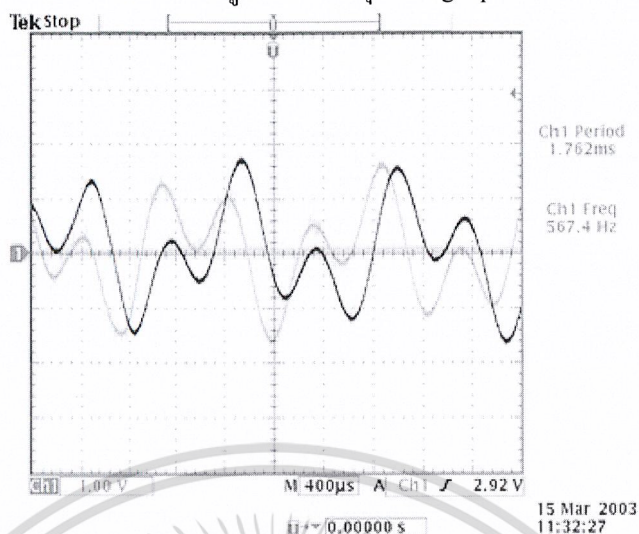


รูปที่ 4.24 แสดงสัญญาณเสียงที่ได้จากสปีกเกอร์ของเครื่องคอมพิวเตอร์ที่เป็นเครื่องโคลเอินท์



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.25 แสดงสัญญาณเสียงที่ได้จากหูฟังโทรศัพท์ที่นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เมื่อชนทนกันเสร็จแล้วก็ทำการวางหูโดยการกดปุ่ม Hang Up



รูปที่ 4.26 แสดงสัญญาณการกดคีย์ Hang Up

สรุปผลการทดลองในส่วนของการสื่อสารเสียง

เราสามารถที่จะทำการส่งข้อความและ ทำการสนทนากันผ่านเครือข่ายอินเทอร์เน็ตได้ โดยที่จะเกิดการหน่วงของสัญญาณเสียงเกิดขึ้นเนื่องจากข้อมูลเสียงต้องใช้เวลาในการเดินทาง และเราสามารถที่จะใช้อุปกรณ์ต่อพ่วงทำการติดต่อกับเครือข่ายโทรศัพท์ภายในท้องถิ่นที่เครื่องเซิร์ฟเวอร์อยู่ได้

การทดลองในส่วนแพคเกจ

1.เปิดโปรแกรมสำหรับการส่งแพคเกจ โดยกดปุ่ม Send Fax ในโปรแกรม Client Send จะปรากฏหน้าจอตั้งแสดง ได้ดังรูปที่ 4.27

Project Send Fax - Mr. Pronprot Pravenanuson

User Information View Fax File

Username: 3015025

Password: #####

Mail Server: www.kmitl.ac.th

Fax Number:

Fax Header: Send Fax By Mr. Pronprot

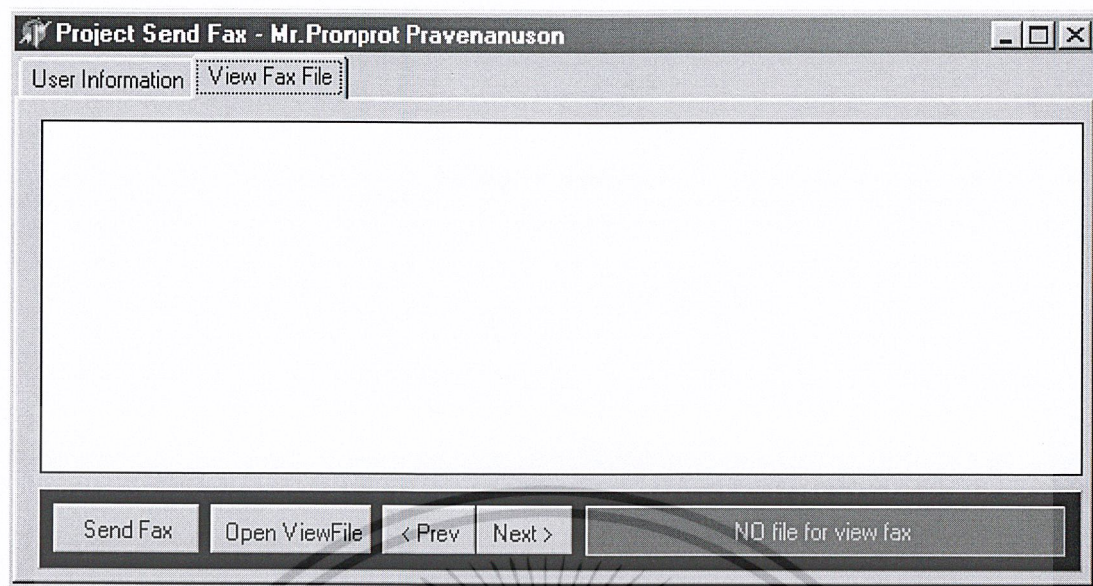
Text Size: 15

Delete After Recive

Start Mail Reset

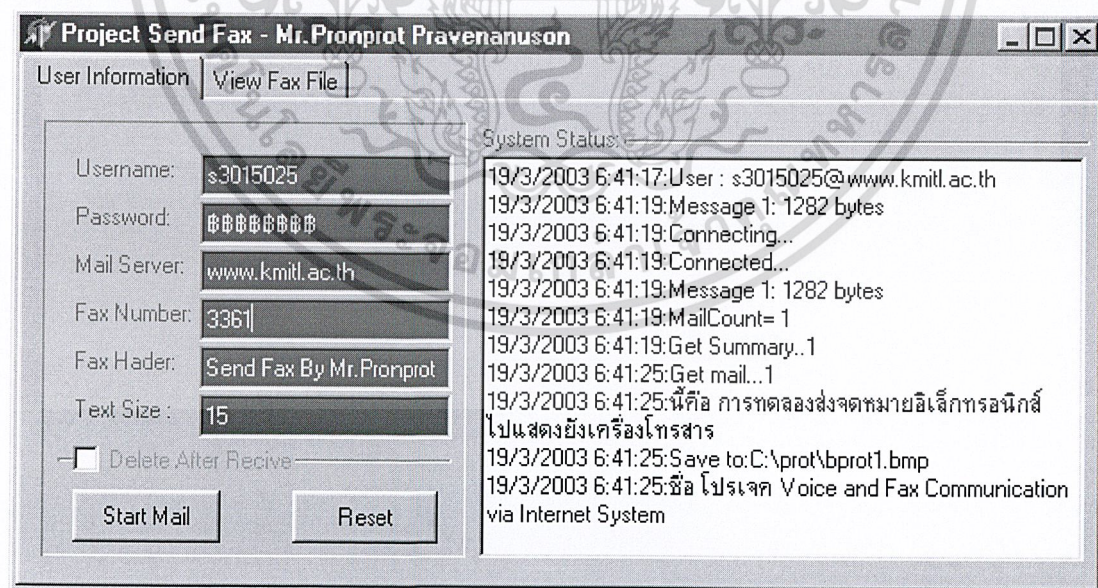
System Status:

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำเอกสารนี้ไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



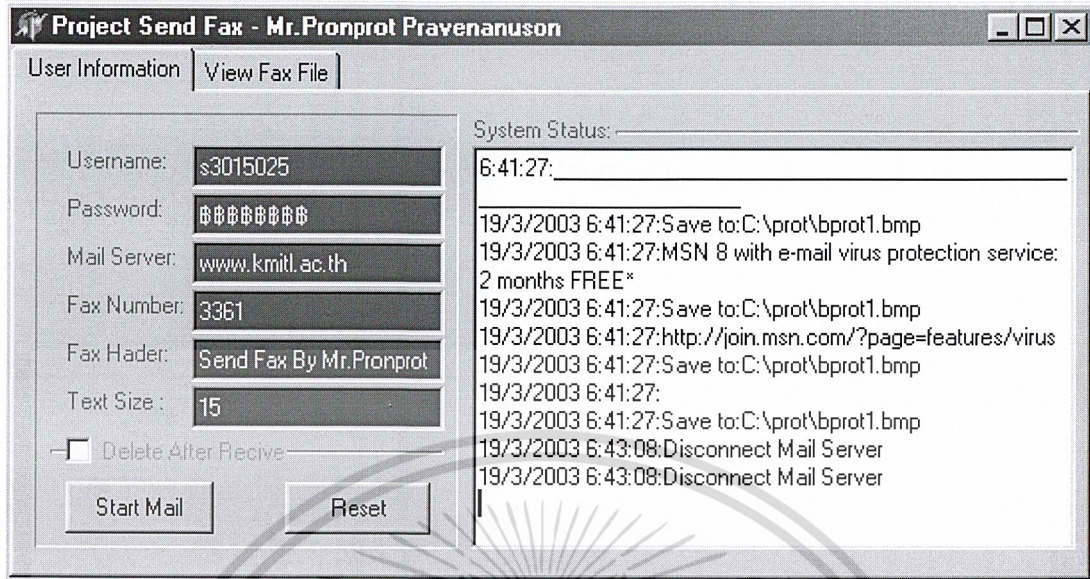
รูปที่ 4.27 แสดงหน้าจอของโปรแกรมที่ใช้ในการส่งแฟกซ์

1. ทำการกำหนดรายละเอียดต่างๆ ได้แก่ ชื่อผู้ใช้งาน (Username) รหัสผ่าน (Password) เมล์เซิร์ฟเวอร์ (Mail Server) และเบอร์โทรสารที่ต้องการส่งไป (Fax Number) จำหน่ายส่วนหัวของแฟกซ์ (Fax Header) ขนาดของตัวอักษร (Text Size)
2. เมื่อกำหนดค่าต่างๆ เรียบร้อยแล้ว ก็ทำการกดปุ่ม Start Mail เพื่อเริ่มการทำงานของโปรแกรม

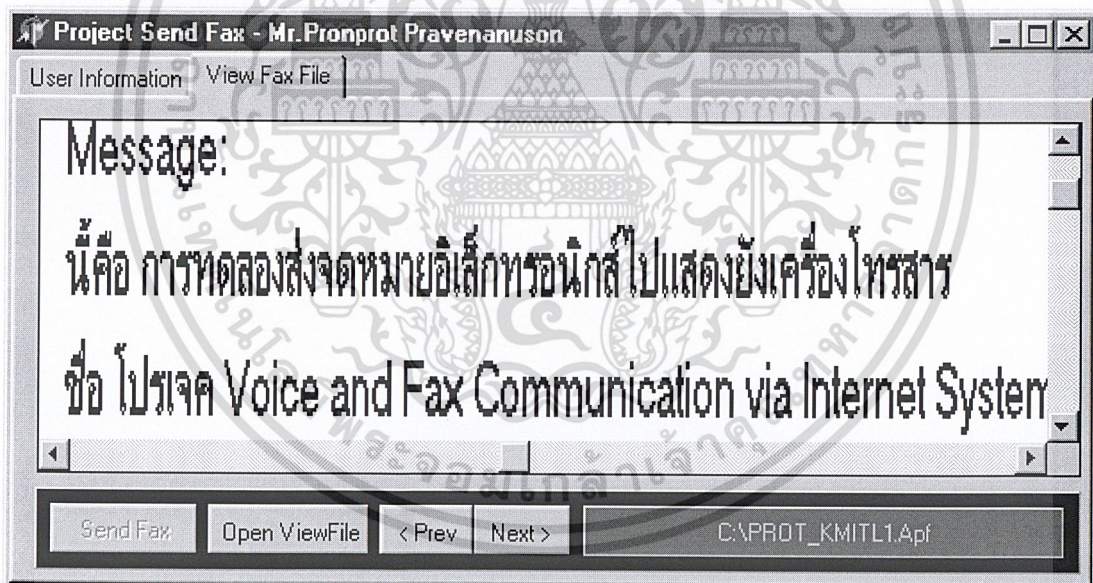


รูปที่ 4.28 แสดงการทำงานของโปรแกรม เมื่อทำการติดต่อกับเมล์เซิร์ฟเวอร์และเริ่มรับข้อมูลจากจดหมาย

อิเล็กทรอนิกส์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 แสดงการทำงานของโปรแกรมเมื่อทำงานเสร็จ



รูปที่ 4.30 แสดงหน้าจอแสดงเอกสารที่รับได้จากเมลล์เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Send Fax By Mr.Pronprot Pravenanrson

From : "pronprot pravenanrson" <pew_p@hotmail.com>

Subject: Test E-mail To Fax

Message:

นี่คือ การทดลองส่งจดหมายอิเล็กทรอนิกส์ไปแสดงยังเครื่องโทรสาร
ชื่อ โปรแกรม Voice and Fax Communication via Internet System
ระบบเสียงและแฟกซ์ผ่านอินเทอร์เน็ต
1234567890/*-+
abcdefghijklmnopqrstuvwxyz
ขอบคุณครับ - Thanks you.

MSN 8 with e-mail virus protection service: 2 months FREE*

<http://join.msn.com/?page=features/virus>

รูปที่ 4.31 แสดงตัวอย่างเอกสารที่ได้จากเครื่องโทรสารปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

ในโครงการนี้จะแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ ส่วนของการรับส่งข้อมูลเสียง และส่วนของการส่งแฟกซ์

ส่วนของการรับส่งข้อมูลเสียง

ในส่วนนี้จะประกอบด้วย 2 ส่วนด้วยกัน

1. ส่วนของฮาร์ดแวร์ ส่วนนี้จะทำหน้าที่ในการเชื่อมต่อระหว่างเครือข่ายอินเทอร์เน็ตเข้ากับข่ายโทรศัพท์

2. ส่วนของซอฟต์แวร์ ส่วนนี้จะทำหน้าที่ในการรับและส่งข้อมูลเสียงผ่านเครือข่ายอินเทอร์เน็ตซึ่งการทำงานของโปรแกรมจะเป็นลักษณะไคลเอ็นท์ เซิร์ฟเวอร์

การทดสอบการทำงานของส่วนนี้ผลที่ได้คือเราสามารถส่งเสียงพูดคุยกันผ่านเครือข่ายอินเทอร์เน็ตได้ การสนทนาโดยใช้ไมโครโฟนกับลำโพงที่ตัวคอมพิวเตอร์ เราสามารถสนทนาได้ชัดเจนดี แต่มีการหน่วงของสัญญาณเสียงเกิดขึ้น ก็คือเสียงจะมาช้ากว่าปกติเพราะต้องผ่านเครือข่ายมานั่นเอง การหน่วงของเสียง ซึ่งก็เป็นปกติของข้อมูลที่ส่งผ่านอินเทอร์เน็ตก็ต้องมีเวลาการเดินทางไป

เมื่อนำอุปกรณ์ฮาร์ดแวร์ต่อเข้ากับคอมพิวเตอร์ จากการทดลองก็สามารถทำการติดต่อไปยังหมายเลขโทรศัพท์ปลายทางที่ต้องการได้ และสามารถรับและส่งสัญญาณเสียงต่อไปยังเครือข่ายโทรศัพท์ได้

ส่วนของการส่งแฟกซ์

โครงการในส่วนนี้เป็นการการพัฒนาโปรแกรมเพื่อใช้งานเป็นตัวกลางในการเชื่อมต่อระหว่างเครือข่ายอินเทอร์เน็ตกับเครือข่ายโทรศัพท์ โดยจะทำการตรวจสอบจดหมายอิเล็กทรอนิกส์แล้วนำข้อมูลที่ได้ไปทำการเข้ารหัสเพื่อทำการส่งแฟกซ์ไปยังหมายเลขแฟกซ์ปลายทาง

จากการทดลองการรับข้อมูลจากจดหมายอิเล็กทรอนิกส์เพื่อนำมาส่งแฟกซ์ เราสามารถรับจดหมายอิเล็กทรอนิกส์ได้ถูกต้อง และนำข้อมูลมาทำการส่งแฟกซ์ไปยังปลายทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวกโปรแกรม

ภาคผนวกโปรแกรมในส่วนของการรับส่งเสียง

Client Send , Server Send

Option Explicit

Public CLOSINGAPPLICATION As Boolean

Public wStream As Object

Dim cnt As Integer

Dim Iport1 As Integer

Private Sub cmdTalk_Click()

' Activates Audio PlayBack

Dim rc As Long

Dim iPort As Integer

Dim itm As Integer

iPort = wStream.StreamInQueue

Iport1 = iPort

Do While (iPort <> NULLPORTID)

For itm = 0 To ConnectionList.ListCount - 1

If (ConnectionList.ItemData(itm) = iPort) Then

ConnectionList.TopIndex = itm

ConnectionList.Selected(itm) = True

Exit For

End If

Next

Loop

ConnectionList.TopIndex = 0

If (ConnectionList.ListCount > 0) Then

ConnectionList.Selected(0) = True

ConnectionList.Selected(0) = False

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

Private Sub Command1_Click()

Dim Idx As Long

Command1.Enabled = False

Idx = InstanceTCP(TCPSocket)

If (Idx > 0) Then

ConnectionList.Enabled = False

On Error Resume Next

If Not Connect(TCPSocket(Idx), txtServer.Text, VOICEPORT)

Then

Unload TCPSocket(Idx)

End If

ConnectionList.Enabled = True

Winsock2.RemoteHost = txtServer.Text

Winsock2.SendData ":>"

'If (Winsock1.State = sckClosed) Then

'Winsock1.Connect

'Winsock1.SendData txtServer.Text

' End If

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ' Command1.Enabled = False
        End If
        Option1.SetFocus
End Sub
'-----
Private Sub Command2_Click()           'Delphi Interface Visual Basic
OLE1.DoVerb
End Sub
'-----
Private Sub Command3_Click()
Dial.Show
End Sub
'-----
Private 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)
    Call GetIdxFromMemberID(TCPSocket, MemberID, Idx)
    Call RemoveConnectionFromList(TCPSocket (Idx), ConnectionList)
    Call Disconnect(TCPSocket (Idx))
    Unload TCPSocket (Idx)
'-----
End Sub
'-----
Private Sub Form_Load()
'-----
    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
    Call InitServerList(txtServer)
    txtServer.Text = txtServer.List(0)
    Set wStream = CreateObject("WaveStreaming.WaveStream")
    Call wStream.InitACMCodec(WAVE_FORMAT_GSM610, TIMESLICE)
    Text1.Text = " :>"
End Sub
'-----
Private Sub Form_Unload(Cancel As Integer)
'-----
    Dim Idx As Long                    ' TCP socket
index
    Dim Socket As Winsock ' TCP socket control
'-----
    CLOSINGAPPLICATION = True
    For Each Socket In TCPSocket
        Call Disconnect(Socket)
    Next
    Set wStream = Nothing
End
'-----
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่มีการตีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและข้อมูลเชิงลึกของเอกสารชุดนี้ที่มีการนำไปใช้


```

        TCPSocket(Index).RemoteHost = RemHost
        TCPSocket(Index).Connect
    Else
        RemHost = UCase(TCPSocket(Index).RemoteHostIP)
    End If
    rc = vbYes
    If (rc = vbYes) Then
        Idx = InstanceTCP(TCPSocket)
        If (Idx > 0) Then
            TCPSocket(Idx).LocalPort = 0
            Call TCPSocket(Idx).Accept(requestID) ' Accept
connection
            Call AddConnectionToList(TCPSocket(Idx), ConnectionList)
        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 Text1_KeyPress(KeyAscii As Integer)
Text1 = Text1
Winsock2.SendData(KeyAscii)
End Sub
-----
Private Sub Text2_KeyPress(KeyAscii As Integer)
Text2 = Text2
Winsock2.GetData(KeyAscii)
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
    End If ' If Found
Then Exit
        txtServer.AddItem UCase(txtServer.Text) ' Add Server
To List
End If
End Sub

```

เอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

-----
Function InstrBt(dt, chrb) As Long
Dim I As Long
Dim x As Long
I = LenB(dt) - 1
For x = 0 To I
    If dt(x) = chrb Then
        InstrBt = x
        Exit Function
    End If
Next x
End Function
-----

```

```

-----
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)
'End Sub
-----

```

```

-----
Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)
Dim strData As String
' Then tell winsock to get that data and import it.
Winsock2.GetData strData, vbString
' Changes the remote IP to where the incoming data
' came from.
Text3.Text = Winsock2.RemoteHostIP
Winsock2.RemoteHost = Winsock2.RemoteHostIP
' If the users press the backspace key, then it
' will register on both ends and send the data.
If Asc(strData) = 8 And Len(Text2) > 0 Then
    Text2.Text = Mid(Text2, 1, (Len(Text2) - 1))
Else
    Text2 = Text2 & strData
End If
' If the user presses the enter key, then it
' will register on both ends and send the data.
If Asc(strData) = 13 Then
    Text2 = Text2 & vbNewLine
End If
' This will make the textbox the other person is
' talking through automatically move down when it
' gets to the bottom of the textbox.
Text2.SelStart = Len(Text2)
End Sub
-----

```

Client Dail

```

-----
Private Sub Command1_Click()
    Command1.Enabled = False
    Command2.Enabled = True
    Command3.Enabled = True
    Command4.Enabled = True
    Command5.Enabled = True
    Command6.Enabled = True
    Command7.Enabled = True
    Command8.Enabled = True
    Command9.Enabled = True
    Command15.Enabled = True
    Command13.Enabled = True
    Command10.Enabled = True
    Command11.Enabled = True
    Command12.Enabled = True

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรนำเอกสารนี้ไปใช้ในการเรียนการสอนโดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย
 ไม่ควรนำเอกสารนี้ไปใช้ในการค้าขายโดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

```
Command16.Enabled = True
Label2.Caption = ""
Winsock1.SendData "a1"
```

```
End Sub
```

```
-----
Private Sub Command10_Click()
If Command15.Enabled = False Then
Winsock1.SendData "*"
End If
If Label2.Caption = "Enter num." Then
Label2.Caption = "*"
Else: Label2.Caption = Label2.Caption & "*"
End If
```

```
End Sub
```

```
-----
Private Sub Command11_Click()
If Command15.Enabled = False Then
Winsock1.SendData "0"
End If
If Label2.Caption = "Enter num." Then
Label2.Caption = 0
Else: Label2.Caption = Label2.Caption & 0
End If
End Sub
```

```
-----
Private Sub Command12_Click()
If Command15.Enabled = False Then
Winsock1.SendData "#"
End If
If Label2.Caption = "Enter num." Then
Label2.Caption = "#"
Else: Label2.Caption = Label2.Caption & "#"
End If
```

```
End Sub
```

```
-----
Private Sub Command13_Click()
If Command15.Enabled = False Then
Winsock1.SendData "1"
End If
If Label2.Caption = "Enter num." Then
Label2.Caption = 1
Else: Label2.Caption = Label2.Caption & 1
End If
```

```
End Sub
```

```
-----
Private Sub Command14_Click()
'Command1.Enabled = False
```

```
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
Command6.Enabled = True
Command7.Enabled = True
Command8.Enabled = True
Command9.Enabled = True
```

```
Command15.Enabled = True
Command13.Enabled = True
```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่อาจรณินใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Command10.Enabled = True
Command11.Enabled = True
Command12.Enabled = True
'Command16.Enabled = True
End Sub

```

```

-----
Private Sub Command15_Click()
Dim Dial As String
Text2.Text = "a" & Label2.Caption
Dial = Text2.Text

```

```

If (Winsock1.State = sckClosed) Then
Winsock1.RemoteHost = Text1.Text
Winsock1.Connect
End If

```

```

Winsock1.SendData Dial
Command15.Enabled = False
Command13.Enabled = False
Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = False
Command6.Enabled = False
Command7.Enabled = False
Command8.Enabled = False
Command9.Enabled = False
Command10.Enabled = False
Command11.Enabled = False
Command12.Enabled = False
Command16.Enabled = False

```

```

Command1.Enabled = True
End Sub

```

```

-----
Private Sub Command16_Click()
Dim datasuit As String
datasuit = Label2.Caption
If (Label2.Caption <> "") Then
Label2.Caption = Mid(datasuit, 1, (Len(datasuit) - 1))
End If
End Sub

```

```

-----
Private Sub Command2_Click()
If Command15.Enabled = False Then
Winsock1.SendData "2"
End If
If Label2.Caption = "Enter num." Then
Label2.Caption = 2
Else: Label2.Caption = Label2.Caption & 2
End If

```

```

End Sub

```

```

-----
Private Sub Command3_Click()
If Command15.Enabled = False Then
Winsock1.SendData "3"
End If
If Label2.Caption = "Enter num." Then
Label2.Caption = 3
Else: Label2.Caption = Label2.Caption & 3
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```
-----  
Private Sub Command4_Click()  
If Command15.Enabled = False Then  
Winsock1.SendData "4"  
End If  
If Label2.Caption = "Enter num." Then  
Label2.Caption = 4  
Else: Label2.Caption = Label2.Caption & 4  
End If
```

End Sub

```
-----  
Private Sub Command5_Click()  
If Command15.Enabled = False Then  
Winsock1.SendData "5"  
End If  
If Label2.Caption = "Enter num." Then  
Label2.Caption = 5  
Else: Label2.Caption = Label2.Caption & 5  
End If
```

End Sub

```
-----  
Private Sub Command6_Click()  
If Command15.Enabled = False Then  
Winsock1.SendData "6"  
End If  
If Label2.Caption = "Enter num." Then  
Label2.Caption = 6  
Else: Label2.Caption = Label2.Caption & 6  
End If
```

End Sub

```
-----  
Private Sub Command7_Click()  
If Command15.Enabled = False Then  
Winsock1.SendData "7"  
End If  
If Label2.Caption = "Enter num." Then  
Label2.Caption = 7  
Else: Label2.Caption = Label2.Caption & 7  
End If
```

End Sub

```
-----  
Private Sub Command8_Click()  
If Command15.Enabled = False Then  
Winsock1.SendData "8"  
End If  
If Label2.Caption = "Enter num." Then  
Label2.Caption = 8  
Else: Label2.Caption = Label2.Caption & 8  
End If
```

End Sub

```
-----  
Private Sub Command9_Click()  
If Command15.Enabled = False Then  
Winsock1.SendData "9"
```

ไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
If Label2.Caption = "Enter num." Then
Label2.Caption = 9
Else: Label2.Caption = Label2.Caption & 9
End If

End Sub
'-----
Private Sub Form_Load()
Command1.Enabled = False
'Winsock2.Bind
Text1.Text = frmChat.txtServer.Text
End Sub
'-----
'Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)
'Dim RMhost As String
'Winsock2.GetData RMhost
'Text1.Text = RMhost
'End Sub
=====
CLIENT RECIVE , SERVER RECIVE
Option Explicit
Public CLOSINGAPPLICATION As Boolean ' Application
status flag
Public wStream As Object
Dim cnt As Integer
Dim Iport1 As Integer
'-----
Private Sub cmdTalk_Click() ' Activates
Audio PlayBack
'-----
Dim rc As Long
Dim iPort As Integer
Dim itm As Integer
'-----
If (Not wStream.Playing And wStream.PlayDeviceFree And
Not wStream.Recording And wStream.RecDeviceFree) Then
wStream.Playing = True
cmdTalk.Caption = "&Playing"
Screen.MousePointer = vbHourglass

iPort = wStream.StreamInQueue
Iport1 = iPort
Do While (iPort <> NULLPORTID)

For itm = 0 To ConnectionList.ListCount - 1
If (ConnectionList.ItemData(itm) = iPort) Then
ConnectionList.TopIndex = itm
ConnectionList.Selected(itm) = True
Exit For
End If

Next
If cnt = 1 Then cnt = 3
If cnt = 0 Then cnt = 1
rc = wStream.PlayWave(Me.hWnd, iPort) ' Play wave data
in iPort...
Call wStream.RemoveStreamFromQueue(iPort) ' Remove PortID
From PlayWave Queue
iPort = wStream.StreamInQueue
Loop
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

        ConnectionList.TopIndex = 0
        If (ConnectionList.ListCount > 0) Then
            ConnectionList.Selected(0) = True
            ConnectionList.Selected(0) = False
        End If
        Screen.MousePointer = vbDefault
        wStream.Playing = False                                ' Turn Playing
Status Off
    End If
End Sub
-----
Private Sub cmdTalk_MouseDown(Button As Integer, Shift As Integer, x As
Single, Y As Single)
' Activates Audio Recording...
-----
    Dim rc As Long
-----
    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"
        Screen.MousePointer = vbHourglass
        rc = wStream.RecordWave(Me.hWnd, TCPSocket)            ' Record voice
and send to all connected sockets

        Screen.MousePointer = vbDefault
        cmdTalk.Caption = "&Talk"

        If Not wStream.Playing And _
            wStream.PlayDeviceFree And _
            wStream.RecDeviceFree Then                            ' Is Audio
Device Free?
            Call cmdTalk_Click
        End If
    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
-----
Private 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)

    Call GetIdxFromMemberID(TCPSocket, MemberID, Idx)
    Call RemoveConnectionFromList(TCPSocket(Idx), ConnectionList)

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรคัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

    Call Disconnect(TCPSocket(Idx))           ' Disconnect
Socket Connection
    Unload TCPSocket(Idx)                   ' Destroy socket
instance

    cmdTalk.Enabled = (ConnectionList.ListCount > 0) ' Enable/Disable
Talk Button...

'-----
End Sub
'-----
Private Sub Form_Load()
'-----
    Dim rc As Long
    Dim Idx As Long
    Dim TCPidx As Long

'-----
    CLOSINGAPPLICATION = False
    Call InitServerList(txtServer)
    txtServer.Text = txtServer.List(Idx)

    Set wStream = CreateObject("WaveStreaming.WaveStream")
    Call wStream.InitACMCodec(WAVE_FORMAT_GSM610, TIMESLICE)
    cmdTalk.Enabled = False

    Call Listen(TCPSocket(Idx))
'-----
End Sub
'-----
Private Sub Form_Unload(Cancel As Integer)
'-----
    Dim Idx As Long
    Dim Socket As Winsock

'-----
    CLOSINGAPPLICATION = True
    For Each Socket In TCPSocket
        Call Disconnect(Socket)
    Next
    Set wStream = Nothing
    End

'-----
End Sub
'-----
Private Sub Option1_GotFocus()
' Activates Audio Recording...
'-----
    Dim rc As Long

'-----
    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

        rc = wStream.RecordWave(Me.hWnd, TCPSocket) ' Record voice
and send to all connected sockets

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        If Not wStream.Playing And _
            wStream.PlayDeviceFree And _
            wStream.RecDeviceFree Then ' Is Audio
Device Free?
    Call cmdTalk_Click
    End If
    End If
'-----
End Sub
'-----
Private Sub TCPSocket_Close(Index As Integer)
' Closing Current TCP Connection...
'-----
    Call RemoveConnectionFromList(TCPSocket(Index), ConnectionList)
    Call Disconnect(TCPSocket(Index))
    cmdTalk.Enabled = (ConnectionList.ListCount > 0)
    If cmdTalk.Enabled Then
    End If
    Unload TCPSocket(Index)
'-----
End Sub
'-----
Private Sub TCPSocket_Connect(Index As Integer)
' TCP Connection Has Been Accepted And Is Open...
'-----
    Call AddConnectionToList(TCPSocket(Index), ConnectionList)

    cmdTalk.Enabled = True ' Enabled
For Connection...
'-----
End Sub
'-----
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
    rc = vbYes
    If (rc = vbYes) Then
        Idx = InstanceTCP(TCPSocket)
        If (Idx > 0) Then
            TCPSocket(Idx).LocalPort = 0
            Call TCPSocket(Idx).Accept(requestID) ' Accept
connection
            Call AddConnectionToList(TCPSocket(Idx), ConnectionList)
            ' Call ResPlaySound(RingInId)
            cmdTalk.Enabled = True
Enabled For Connection...
        End If
    End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub
'-----
Private Sub TCPSocket_DataArrival(Index As Integer, ByVal BytesTotal As
Long)
' Incomming Buffer On...
'-----
    Dim rc As Long                                     ' Return Code
Variable
    Dim WaveData() As Byte ' Byte array of wave data
    Dim WaveData1() As Byte
    Static ExBytes(MAXTCP) As Long
    Static ExData(MAXTCP) As Variant
    Dim ExData1(MAXTCP) As String
    Dim i As Long
    Dim x As Long
'-----
With wStream
    If (TCPSocket(Index).BytesReceived > 0) Then
        Do While (TCPSocket(Index).BytesReceived > 0)
            If (ExBytes(Index) = 0) Then
                If (.waveChunkSize <= TCPSocket(Index).BytesReceived)
Then
                    Call TCPSocket(Index).GetData(WaveData, vbByte +
vbArray, .waveChunkSize)
                    i = UBound(WaveData)
                    x = InstrBt(WaveData, 255)
                    Call .SaveStreamBuffer(Index, WaveData)
                    Call .AddStreamToQueue(Index)
                Else
                    ExBytes(Index) = TCPSocket(Index).BytesReceived '
Save Extra bytes
                    Call TCPSocket(Index).GetData(ExData(Index), vbByte
+ vbArray, ExBytes(Index))
                    End If
                Else
                    Call TCPSocket(Index).GetData(WaveData, vbByte +
vbArray, .waveChunkSize - ExBytes(Index))
                    ExData(Index) = MidB(ExData(Index), 1) & MidB(WaveData, 1)
                    WaveData1 = ExData(Index)
                    i = UBound(WaveData1)
                    x = InstrBt(WaveData1, 255)
                    Call .SaveStreamBuffer(Index, WaveData1)
                    Call .AddStreamToQueue(Index)
                End If
            End If
        Loop
    End With
    Data Chunk                                     ' Look for next

    If (Not .Playing And .PlayDeviceFree And _
Not .Recording And .RecDeviceFree) Then
        Device Status                               ' Check Audio
        Call cmdTalk_Click                           ' Start
    End If
End With
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'-----
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 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
            txtServer.AddItem UCase(txtServer.Text)
        End If
    End Sub
'-----
Function InstrBt(dt, chrb) As Long
'-----
'End Function
Dim i As Long
Dim x As Long
i = LenB(dt) - 1
For x = 0 To i
    If dt(x) = chrb Then
        InstrBt = x
        Exit Function
    End If
Next x
End Function
=====
TCP_FUNCTION
Option Explicit
'-----
Public Function InstanceTCP(TCPArray As Variant) As Long
'-----
    Dim Ind As Long
'-----
    InstanceTCP = -1
    On Error GoTo InitControl

    For Ind = MINTCP To MAXTCP
        If (TCPArray(Ind).Name <> "") Then
            เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
            ไม่ควรแก้ไข ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
        End If
    Next Ind
End Function

```

```

Next
-----
InitControl:
Control                                     ' Initialize New
-----
On Error GoTo ErrorHandler

If ((Ind >= MINTCP) And (Ind <= MAXTCP)) Then
    Load TCPArray(Ind)
    InstanceTCP = Ind
End If

Exit Function                                     ' Exit
-----
ErrorHandler:
                                           ' Handler
-----
Debug.Print Err.Number, Err.Description
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)
Socket.LocalPort = 0
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 = VOICEPORT2         ' Set standard
application port
    End If
    Call Socket.Listen                           ' Listen On Local
Port
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'-----
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
'-----
=====
CHAT-DEFS
Option Explicit
' Application User Defined Types...
' Sound Format
Public Const WAVE_FORMAT_GSM610 = &H31      ' GSM610 Wave Format
Public Const TIMESLICE = 0.2               ' Time Slicing 1/5 Second
' Application Constants...
'== 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
'-----
TCP-DEF
Option Explicit
'== TCP Port Array Processing Const.s
=====
Public Const MINTCP = 1                    ' Minimum index for
tcpsocket control instance
Public Const MAXTCP = 32                   ' Maximum index for
tcpsocket control instance
Public Const VOICEPORT = 701               ' Voice chat Port To Listen
On...
Public Const VOICEPORT2 = 702
Public Const NULLPORTID = 0                ' Null Port ID - A Port ID
That Will Never Be Used...
=====

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAT-FUNCTION

Option Explicit

```
Public Sub InitServerList(ServerList As ComboBox)
```

```
    ' Populate Server List Box...  
    ServerList.AddItem "127.0.0.1"
```

```
End Sub
```

```
Public Sub DebugSocket(TCPSocket1 As Winsock)
```

```
    ' Prints Information In A TCP Socket, For Debugging TCP Events...
```

```
    Debug.Print "TCPSocket1.RemoteHost", TCPSocket1.RemoteHost  
    Debug.Print "TCPSocket1.RemoteHostIP", TCPSocket1.RemoteHostIP  
    Debug.Print "TCPSocket1.RemotePort", TCPSocket1.RemotePort  
    Debug.Print "TCPSocket1.LocalHostName", TCPSocket1.LocalHostName  
    Debug.Print "TCPSocket1.LocalIP", TCPSocket1.LocalIP  
    Debug.Print "TCPSocket1.LocalPort", TCPSocket1.LocalPort  
    Debug.Print "TCPSocket1.State", TCPSocket1.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 = "หมายเลข IP ปลายทาง : " & 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
```

เอกสารนี้เป็นเอกสารของบริษัทฯ ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

-----
' Create MemberID From HostName and RemoteIP
MemberID = Socket.RemoteHostIP & " [" & _
          Format(Socket.RemotePort, "0") & "]" - [" & _
          Format(Socket.LocalPort, "0") & "]"

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
sEnd = InStr(sStart, MemberID, "]") - 1 ' Get end of remote
port
If (sEnd > 2) Then                       ' If end found
RemotePort = 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???
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return index                               Index = Socket.Index           ' Save and
exit                                         Exit Sub                          ' Done...

                                           End If
                                           Next
                                           End If
                                           End If
                                           End If
                                           End If
End Sub
-----

```

ภาคผนวกโปรแกรมส่วนของโปรแกรมส่งแฟกซ์ (Delphi)

```

unit Unit1;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  StdCtrls, ComCtrls, Ucon, ExtCtrls, UFax0, Buttons, OoMisc, AdFView;

type
  TForm1 = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    system_memo_log: TMemo;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Button1: TButton;
    Button2: TButton;
    Bevel1: TBevel;
    DelCheckbox1: TCheckBox;
    Bevel2: TBevel;
    Bevel3: TBevel;
    TabSheet2: TTabSheet;
    OpenFileDialog1: TOpenDialog;
    ApdFaxViewer1: TApdFaxViewer;
    Panel2: TPanel;
    Panel1: TPanel;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    Button3: TButton;
    BitBtn3: TBitBtn;
    Edit5: TEdit;
    Label6: TLabel;
    Edit6: TEdit;
    Label7: TLabel;
    procedure Button1Click(Sender: TObject);
  end;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

procedure Button2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure ViewFile();
procedure ViewMail();
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure mailtofax();
private
  { Private declarations }
public
  { Public declarations }
  procedure system_log(ss:string);
end;

var
  Form1: TForm1;

implementation

uses UMail0;

{$R *.DFM}
procedure TForm1.system_log(ss:string);
var
  log:string;
begin
  log:=DateTimeToStr(Now)+'.'+ ss ;
  with system_memo_log do
    begin
      if Lines.Count>200 then lines.Delete(0);
      lines.add(DateTimeToStr(now)+'.'+ ss );
      system_memo_log.Lines.EndUpdate;
    end;
end;
//-----
procedure TForm1.Button1Click(Sender: TObject);
var
  inuu:user_rec;
begin
  inuu.username := Edit1.text;
  inuu.password := Edit2.text;
  inuu.server := Edit3.text;
  inuu.faxnumber := Edit4.text;
  DataModule2.mail_start(inuu);
end;
//-----
procedure TForm1.Button2Click(Sender: TObject);
begin
  try
    DataModule2.NMPOP31.Reset;
    DataModule2.NMPOP31.Disconnect ;
  except
    system_log('Reset Error');
  end;
end;
//-----
procedure TForm1.FormCreate(Sender: TObject);
begin
  รนเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
  ไม่ว่าจะณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

    panell.Caption := 'NO file for view fax ';
end;
//-----
procedure TForm1.mailtofax();
begin
    DataModule3.SendToFax();
end;
//-----
procedure TForm1.Button3Click(Sender: TObject);
begin
    //showmessage('Plase Wait minite ');
    DataModule3.OpenSendToFax;

end;
//-----

//-----
procedure TForm1.ViewMail();
var
    FName : string;
begin
    if FileExists('C:\PROT_KMITL.APF') then
        begin
            FName := ('C:\PROT_KMITL.APF');
            ApdFaxViewer1.FileName := FName;
        end;
end;
//-----
//-----
//-----
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    if ApdFaxViewer1.ActivePage > 1 then
        ApdFaxViewer1.ActivePage := ApdFaxViewer1.ActivePage - 1;
end;
//-----
//-----
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    if ApdFaxViewer1.ActivePage < ApdFaxViewer1.NumPages then
        ApdFaxViewer1.ActivePage := ApdFaxViewer1.ActivePage + 1;
end;
//-----
//-----
procedure TForm1.BitBtn3Click(Sender: TObject);
begin
    if opendialog1.Execute then
        Form1.ApdFaxViewer1.FileName := Form1.OpenDialog1.FileName;
        Panell.Caption := OpenDialog1.FileName;
end;
//-----
//-----
procedure TForm1.ViewFile();
var
    FNameF : string;
begin
    if FileExists('C:\APF_FILE.APF') then
        FNameF := ('C:\APF_FILE.APF');
        ApdFaxViewer1.FileName := FNameF;
end;
end.
=====
unit Umail0;
เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  Psock, NMpop3, Ucon ;

type
  TDataModule2 = class(TDataModule)
    NMPOP31: TNMPOP3;
    procedure mail_start( uu:user_rec);
    procedure NMPOP31Connect(Sender: TObject);
    procedure NMPOP31RetrieveEnd(Sender: TObject);
    procedure NMPOP31InvalidHost(var Handled: Boolean);
    procedure NMPOP31Disconnect(Sender: TObject);
    procedure NMPOP31ConnectionFailed(Sender: TObject);
    procedure NMPOP31Failure(Sender: TObject);
    procedure NMPOP31ConnectionRequired(var Handled: Boolean);
    procedure NMPOP31AuthenticationFailed(var Handled: Boolean);
    procedure NMPOP31DecodeStart(var FileName: String);
    procedure NMPOP31List(Msg, Size: Integer);
    procedure NMPOP31Reset(Sender: TObject);
    //-----

  private
    { Private declarations }

  public
    { Public declarations }
    mail_error:boolean;
    mail_RetrieveFinish, mail_isconnected:boolean;

  end;

var
  DataModule2: TDataModule2;
  mail_buffer:Tstringlist;
  Fsize:longint;
  procedure convertsTobmp(var k:integer);
  procedure FreeMemory1(var p:integer);
  //----- function FindSize(p:integer):longint;
  implementation

  {$R *.DFM}
  uses Unit1;
  //-----
  procedure TDataModule2.mail_start( uu:user_rec);
  var
    user_mail_count,i,j,send :integer;
    mailheader:string;
    //----- FiSize:longint;
    userN : user_rec;

begin
  mail_buffer:=Tstringlist.Create;
  with NMPOP31 do
    begin
      ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    end
  end

```

```

if Connected then Disconnect ;
mail_error := false;
Host := uu.server;
UserID:= uu.username;
Password:= uu.password;
AttachFilePath := 'C:\PROT_ATTACHFILE' ;
mail_isconnected:= false;
Form1.system_log('User : '+ uu.username + '@' + uu.server);
try
  {$i-}

Connect;
  {$i+}
except
  on EsockError do
  begin
    Form1.system_log('Check Internet Connection');
    exit;
  end;
end;
end;
Form1.system_log('Connecting... ');
while (not mail_isconnected) and (not mail_error ) do
  application.ProcessMessages;
  if (mail_error) then
  begin
    Form1.system_log('Mail error ');
    exit ;
  end;
  Form1.system_log('Connected... ');
  NMPOP31.List;
  if NMPOP31.MailCount = 0 then
  begin
    Form1.system_log(uu.username+'No E-mail in Mailbox');
    NMPOP31.Disconnect;
    mail_error := false;
    exit;
  end
  else
  begin
    user_mail_count :=NMPOP31.MailCount;
    Form1.system_log('MailCount= '+inttostr(user_mail_count));
    for i := 1 to user_mail_count do
    begin
      mail_RetrieveFinish := false;
      NMPOP31.GetSummary(i);
      Form1.system_log('Get Summary..' + inttostr(i));
      while (not mail_RetrieveFinish) and (not mail_error)
do
      Application.ProcessMessages;
      if (mail_error ) then
      begin
        Form1.system_log('Error When GetSummary');
        exit;
      end;
      mailheader:= NMpop31.Summary.header.Text;
      if (pos('Status: RO',mailheader) = 0 ) or (pos('Status:
R',mailheader) = 0 ) and ((pos('Content-Type: text',mailheader)<> 0 )
or (pos('Content-Type: TEXT',mailheader)<>0 )) then
      begin
        {$i-}
        //showmessage('you ok!');
        ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
      end
    end
  end
end

```

```

mail_RetrieveFinish:=false;
NMPOP31.GetMailMessage(i);
Form1.system_log('Get mail...'+inttostr(i));
while not mail_RetrieveFinish do
Application.ProcessMessages;
mail_buffer.Clear;
mail_buffer.Add('');
mail_buffer.Add('From : ' +
NMPOP31.MailMessage.From);
mail_buffer.Add('');
mail_buffer.Add('Subject: '+NMPOP31.MailMessage.Subject);
mail_buffer.Add('');
mail_buffer.Add('Message:');
for j:= 0 to NMPOP31.MailMessage.Body.Count -1 do
begin
mail_buffer.Add(NMPOP31.MailMessage.Body.strings[j]);
Form1.system_log(NMPOP31.MailMessage.Body.strings[j]);
send := i ;
try
convertsTobmp(send);
except
Form1.system_log('Error convertsTobmp');
Form1.system_log('Maybe you must restart
your computer for restore Memory');
send := i;
FreeMemory1(send);
//----- FiSize :=Fsize;
//----- FreeMem(
end;
end;
//-----
if FileExists('C:\prot\bprot'+ inttostr
(i)+''.bmp') then
begin
Form1.Button3.Enabled := False;
showmessage('you have file to send fax ');
Form1.mailtofax();
end
else
Form1.Button3.Enabled := True;
//-----
if Form1.DelCheckbox1.Checked then
NMPOP31.DeleteMailMessage(i);
end; //END IF
end; //END FOR
NMPOP31.Disconnect;
end; //END IF
end; //END PROCEDURE
//-----
procedure TDataModule2.NMPOP31Connect(Sender: TObject);
begin
mail_isconnected :=true;
NMPOP31.List;
end;

```

สารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

```

procedure TDataModule2.NMPOP31RetrieveEnd(Sender: TObject);
begin
    mail_RetrieveFinish:=true;
end;
//-----
procedure TDataModule2.NMPOP31InvalidHost(var Handled: Boolean);
begin
    Form1.system_log('Invalid Host');
    mail_error:= true;
    handled:=false;
end;
//-----
procedure TDataModule2.NMPOP31Disconnect(Sender: TObject);
begin
    Form1.system_log('Disconnect Mail Server');
    mail_isconnected:= false;
end;
//-----
procedure TDataModule2.NMPOP31ConnectionFailed(Sender: TObject);
begin
    Form1.system_log('Connection Failed');
    mail_error := true;
end;
//-----
procedure TDataModule2.NMPOP31Failure(Sender: TObject);
begin
    Form1.system_log('Failed');
    mail_error := true;
end;
//-----
procedure TDataModule2.NMPOP31ConnectionRequired(var Handled: Boolean);
begin
    mail_error := true;
    handled:= false;
    Form1.system_log('Connection failed');
end;
//-----
procedure TDataModule2.NMPOP31AuthenticationFailed(var Handled:
Boolean);
begin
    mail_error := true;
    Form1.system_log('Password failed');
end;
//-----
procedure TDataModule2.NMPOP31DecodeStart(var FileName: String);
var
    S: String;
begin
    S := FileName;
    if InputQuery('Save File Attachment', 'Filename?', S) then
        FileName := S;
end;
//-----
procedure TDataModule2.NMPOP31List(Msg, Size: Integer);
begin
    Form1.system_log('Message '+IntToStr(Msg)+' : '+IntToStr(Size)+'
bytes');
end;
//เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
//ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นหากไม่มีเหตุใดแบบสงวนเนื้อหา และต้องอยู่ภายใต้
procedure TDataModule2.NMPOP31Reset(Sender: TObject);

```

```

begin
    Form1.system_log('Delete flags reset');
end;
//-----
{*-----converts to bmp-----
*}
procedure convertsTobmp(var k:integer);
var
    //re,k:Integer;
    h,j:Integer;
    pic:TBitmap;
    //x,y,codelength,thenext,count:integer;
    //filetime:string;
    bmpname:string;
    maxline:integer;
    user:user_rec;

begin
    //mail_buffer:= Tstringlist.Create;
    pic := TBitmap.Create;
    pic.Width := 1728;
    pic.Height := 1148*2 ;
    pic.PixelFormat := pf1bit;
    pic.Canvas.Font.Size:= strToInt(Form1.Edit5.Text);
    h:= -(pic.Canvas.Font.Height - 10);
    maxline := mail_buffer.Count;
    if maxline > 50 then maxline := 50 ;
    for j:= 0 to maxline -1 do
        begin
            pic.canvas.TextOut(100, 60+(j*h),mail_buffer.Strings[j]);
        end;

        bmpname := 'C:\prot\bprot'+inttostr(k)+'.bmp';
        pic.SaveToFile(bmpname);

        //----- Fsize :=FindSize(k);
        Form1.system_log('Save to:'+ bmpname);
    end;
//-----
{*function FindSize(p:integer):longint;
var
    f:file of byte;
    size:longint;
begin
    AssignFile(f, 'C:\prot\bprot'+inttostr(p)+'.bmp');
    Reset(f);
    size := filesize(f);
    result := size;
end;*}
//-----
procedure FreeMemory1(var p:integer);
var
    F: file;
    Size: Integer;
    Buffer: PChar;
    userP: user_rec;
begin
    AssignFile(F, 'C:\prot\bprot'+ inttostr(p)+'.bmp');
    Reset(F, 1);
    try
        ไม่ทราบกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
    end;
end;

```

```

Size := FileSize(F);
GetMem(Buffer, Size);
try
  BlockRead(F, Buffer^, Size);
  //ProcessFile(Buffer, Size);
finally
  FreeMem(Buffer);
end;
finally
  CloseFile(F);
end;
end;
//-----
end.

```

```

=====
unit UFax0;

```

```

interface

```

```

uses

```

```

  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  AdFax, AdFStat, AdFaxCvt, OoMisc, AdPort ;

```

```

type

```

```

  TDataModule3 = class(TDataModule)
  ApdComPort1: TApdComPort;
  ApdSendFax1: TApdSendFax;
  ApdFaxStatus1: TApdFaxStatus;
  ApdFaxConverter1: TApdFaxConverter;
  procedure SendToFax();
  procedure OpenSendToFax();

```

```

  private

```

```

    { Private declarations }

```

```

  public

```

```

    { Public declarations }

```

```

end;

```

```

var

```

```

  DataModule3: TDataModule3;

```

```

implementation

```

```

  uses Unit1, UMail0, Ucon;

```

```

  {$R *.DFM}

```

```

  //-----

```

```

  procedure TDataModule3.SendToFax();

```

```

  var

```

```

    userF:user_rec;

```

```

    sFile : string;

```

```

    i: integer;

```

```

    F: file;

```

```

    extStr : string;

```

```

    Faxlist:Tstringlist;

```

```

  begin

```

```

    Faxlist := Tstringlist.Create;

```

```

    for i := 1 to DataModule2.NMPOP31.MailCount do

```

```

      begin

```

เอกสารเป็นเอกสารที่ส่งไปรษณีย์ไปเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if FileExists('C:\prot\bprot' + intToStr(i)+'.bmp') then
begin
    //showmessage('mail to fax complete');
    DataModule3.ApdFaxConverter1.DocumentFile :=
('C:\prot\bprot'+ inttostr(i)+'.bmp');
    DataModule3.ApdFaxConverter1.InputDocumentType := idBmp;
    DataModule3.ApdFaxConverter1.OutFileName :=
('C:\PROT_KMITL'+intToStr(i)+'.Apf');
    DataModule3.ApdFaxConverter1.ConvertToFile;
    //Faxlist.Add('C:\PROT_KMITL'+intToStr(i)+'.Apf');
    //DataModule3.ApdSendFax1.FaxFileList :=Faxlist ;

//DataModule3.ApdSendFax1.ConcatFaxes('C:\PROT_KMITL.APF');
    DataModule3.ApdSendFax1.PhoneNumber := Form1.Edit4.Text ;
    DataModule3.ApdSendFax1.FaxFile :=
('C:\PROT_KMITL'+intToStr(i)+'.Apf');
    Form1.ViewMail();
    //showmessage('mail to fax complete1');
    DataModule3.ApdSendFax1.HeaderLine := Form1.Edit6.Text ;
    DataModule3.ApdSendFax1.StartTransmit;
end

end;
end;
//-----
procedure TDataModule3.OpenSendToFax();
var
    sFile : string;
    extStr : string;
    userF:user_rec;
begin
    //showmessage('OpenSend To Fax is Comple');
    if ( Form1.OpenDialog1.Execute ) then
begin
    sFile := Form1.OpenDialog1.FileName;
    extStr :=uppercase (ExtractFileExt(sFile));
    delete(extStr,1,1);
    DataModule3.ApdFaxConverter1.DocumentFile := sFile;
    if (extStr = 'TXT') Then
DataModule3.ApdFaxConverter1.InputDocumentType:= idText
    else
    if (extStr = 'TIF') then
DataModule3.ApdFaxConverter1.InputDocumentType:= idTiff
    else
    if (extStr = 'PCX') then
DataModule3.ApdFaxConverter1.InputDocumentType:= idPcx
    else
    if (extStr = 'DCX') then
DataModule3.ApdFaxConverter1.InputDocumentType:= idDcx
    else
    if (extStr = 'BMP') then
DataModule3.ApdFaxConverter1.InputDocumenttype:= idBMP;
    DataModule3.ApdFaxConverter1.OutFileName :=
('C:\APF_FILE.APF');
    DataModule3.ApdFaxConverter1.ConvertToFile;
    //-----
    DataModule3.ApdSendFax1.PhoneNumber :=
Form1.Edit4.text;
    // showmessage('is'+ Form1.Edit4.text);
    DataModule3.ApdSendFax1.FaxFile :=('C:\APF_FILE.APF');
    Form1.ViewFile();

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น (สงวนลิขสิทธิ์) ถ้าไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                DataModule3.ApdSendFax1.HeaderLine :=
Form1.Edit6.Text ;
                DataModule3.ApdSendFax1.StartTransmit;
            end;
        end;
//-----
end.
=====
unit Ucon;

interface
type user_rec=record
    username :string;
    password :string;
    server: string;
    faxnumber :string;
    dirname: string;
    AttachFilePathP:string;
end;
implementation
end.

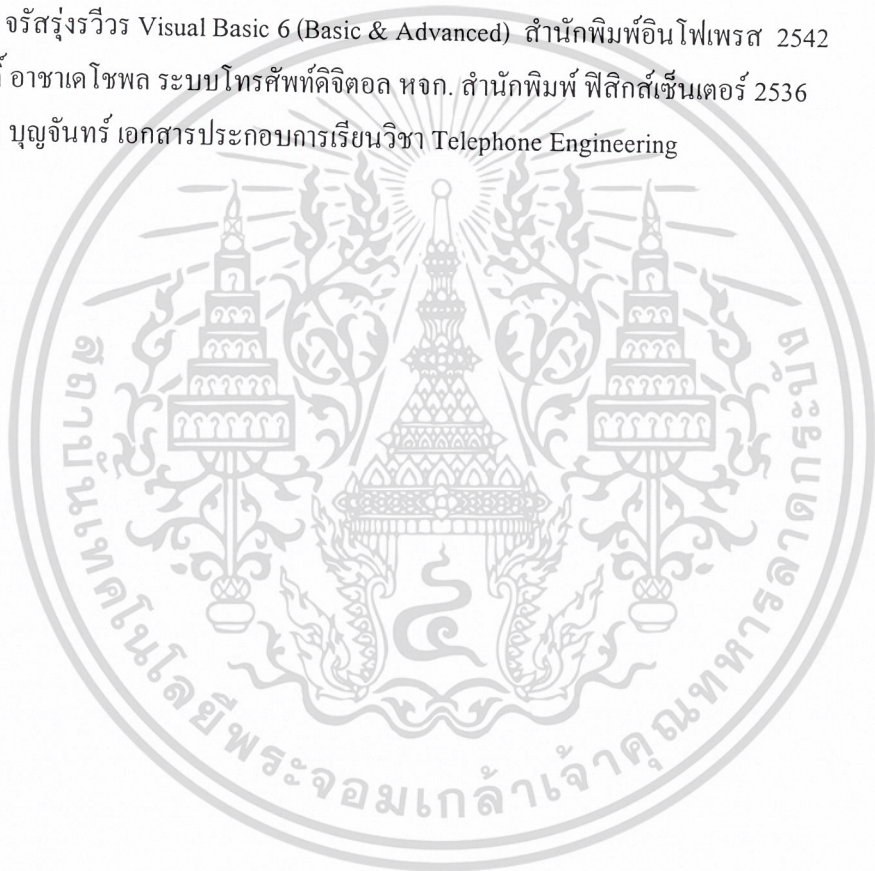
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] สัจจะ จรัสรุ่งรวีร์ และ จักรพงษ์ สุขประเสริฐ เริ่มต้นอย่างมืออาชีพด้วย Delphi 5 ฉบับสมบูรณ์ กรุงเทพฯ: สำนักพิมพ์อินโฟเพรส 2543
- [2] กนก กุสุมาลย์นุกุล ไกรวุฒิ มั่นเสถียรสิน คู่มือการเขียนโปรแกรม Delphi 4 กรุงเทพฯ: ชัคเชกมีเดีย 2541
- [3] ประพนธ์ อัสภาณวัฒน์ Delphi Episode II เทคนิคและการพัฒนาโปรแกรมด้วยเคลไฟ กรุงเทพฯ: ซีเอ็ดยูเคชั่น 2543
- [4] ชาริน สิทธิธรรมชารี Visual Basic Version 6.0 สำนักพิมพ์ ชัคเชกมีเดีย
- [5] สัจจะ จรัสรุ่งรวีร์ Visual Basic 6 (Basic & Advanced) สำนักพิมพ์อินโฟเพรส 2542
- [6] วิสันต์ อาษาเดโชพล ระบบโทรศัพท์ดิจิทัล หจก. สำนักพิมพ์ ฟิสิกส์เซ็นเตอร์ 2536
- [7] สุรพล บุญจันทร์ เอกสารประกอบการเรียนวิชา Telephone Engineering



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้