สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบโทรศัพท์ทางไกลผ่านเครือข่ายอินเตอร์เน็ต

Long-Distance Telephone Via Internet Network



โดย

นายจตุรงค์ โพธิกุล

นายเทพพิชัย ศรีพาระกุล

นายปฐมพงศ์ สว่างวงศ์ธรรม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหมู่.....
เลขทะเบียน 46569
วัน, เดือน, ปี 4 เม.ย. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์ทางไกลผ่านเครือข่ายอินเตอร์เน็ต

Long-Distance Telephone Via Internet Network

โดย

นายจตุรงค์ โพธิกุล 41014061

นายเทพพิชัย ศรีพาระกุล 41014164

นายปฐมพงศ์ สว่างวงศ์ธรรม 41014240

อาจารย์ที่ปรึกษา

รศ.ดร.สุวิพล สิทธิชีวภาค

ผศ.เกรียงไกร วงศ์โรจนาภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบโทรศัพท์ทางไกลผ่านเครือข่ายอินเตอร์เน็ต**

Long-Distance Telephone Via Internet Network

ผู้จัดทำ

1. นายจตุรงค์ โพธิกุล	**41014061**
2. นายเทพพิชัย ศรีพาระกุล	**41014164**
3. นายปฐมพงศ์ สว่างวงศ์ธรรม	**41014240**

อาจารย์ที่ปรึกษา

(ผศ.เกรียงไกร วงศ์โรจนาภรณ์)

อาจารย์ที่ปรึกษา

(รศ.ดร.สุวิพล สิทธิชีวภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์ทางไกลผ่านเครือข่ายอินเตอร์เน็ต

Long-Distance Telephone Via Internet Network

โดย นายจตุรงค์ โพธิกุล 41014061
นายเทพพิชัย ศรีพาระกุล 41014164
นายปฐมพงศ์ สว่างวงศ์ธรรม 41014240

อาจารย์ที่ปรึกษา ผศ.เกรียงไกร วงศ์โรจนภรณ์
รศ.ดร.สุวิพล สิทธิชีวภาค

บทคัดย่อ

เนื่องจากในปัจจุบันการติดต่อสื่อสารมีความจำเป็นอย่างมาก โดยเฉพาะการใช้ระบบโทรศัพท์เป็นวิธีการที่ได้รับความนิยมเป็นอย่างสูง แต่มีข้อเสียในด้านค่าใช้จ่ายโดยเฉพาะการใช้โทรศัพท์ทางไกล ซึ่งยังมีค่าใช้จ่ายค่อนข้างสูง โครงงานนี้ได้มีความพยายามที่จะช่วยลดภาระค่าใช้จ่ายในส่วนนี้ลงโดยการเชื่อมต่อเครื่องโทรศัพท์เข้ากับเครื่องคอมพิวเตอร์ และติดต่อไปยังเครื่องคอมพิวเตอร์ปลายทางผ่านเครือข่ายอินเตอร์เน็ต ซึ่งมีค่าใช้จ่ายที่ถูกกว่าการใช้โทรศัพท์ทางไกล แล้วติดต่อไปยังเครื่องโทรศัพท์ของผู้รับ

ABSTRACT

Nowadays communication is very necessary especially using telephone system which is very popular, but there is a disadvantage about the cost of long-distance call.

This project is presented to reduce the cost in this case by connecting the telephone to the computer and connect to the remote computer via internet network, which is lower cost than long-distance call and connect to the reciever's telephone.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่1 บทนำ 1

บทที่2 ทฤษฎีและหลักการ 3

2.1 ส่วนที่ติดต่อกับเครือข่ายอินเตอร์เน็ท 3

2.1.1 ระบบเครือข่ายคอมพิวเตอร์ (Computer Network System) 3

2.1.1.1 OSI Model 3

2.1.1.2 การติดต่อสื่อสารระหว่างเครื่องเวิร์คสเตชัน 6

2.1.1.3 รูปแบบการเชื่อมต่อของระบบเน็ตเวิร์ค (Topology Of Network) 6

2.1.1.4 โปรโตคอลมาตรฐานของระบบเครือข่าย 8

2.1.2 ระบบเครือข่ายอินเตอร์เน็ต 11

2.1.2.1 ประวัติอินเตอร์เน็ต 11

2.1.2.2 สถาปัตยกรรมของอินเตอร์เน็ต 12

2.1.2.3 บริการในอินเตอร์เน็ต 12

2.1.3 โปรโตคอลในการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ผ่านอินเตอร์เนต 13

2.1.3.1 ความหมายของ TCP/IP 13

2.1.3.2 ชุดโปรโตคอลทีซีพี/ไอพี (TCP/IP protocol suite) 13

2.1.3.3 แบบอ้างอิงTCP/IP 16

2.1.3.4 TCP/IP Client-Server 16

2.1.3.5 ข้อเปรียบเทียบและความสัมพันธ์ระหว่างโปรโตคอล TCP และ UDP 17

2.1.3.6 พอร์ตของ TCP/IP 18

2.1.3.7 ซ็อกเก็ตของ TCP/IP 18

2.1.3.8 หมายเลข IP Address 18

2.1.3.8.1 การจัดคลาสเครือข่าย 19

2.1.3.8.2 ลักษณะสำคัญของแต่ละคลาส 20

2.2 ส่วนติดต่อกับตัวโมเด็ม 20

2.2.1 ชุดคำสั่ง AT ของโมเด็ม 20

2.2.1.1 การใช้งานชุดคำสั่ง AT 21

2.2.1.2 ประเภทของชุดคำสั่งAT 21

2.2.1.3 ชุดคำสั่งเพิ่มเติม 21

2.2.1.4 ชุดคำสั่งATที่สำคัญและใช้ในการทดลอง 21

2.3 เครือข่ายระบบสื่อสารแพ็กเก็ตเสียง 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 เหตุผลในการใช้แพ็กเก็ตเสียง 23
2.3.2 การเข้ารหัสเสียง 23
2.3.3 เทคนิคในการบีบอัดข้อมูล 24
2.4 ข้อมูลการ์ดเสียง 25

บทที่3 การคำนวณ และการสร้าง 28
3.1 ส่วนของโปรแกรมทางด้านผู้เรียก (ทางฝั่งไคลเอ็นต์) 29
3.2 ส่วนของโปรแกรมทางด้านผู้ถูกเรียก (ทางฝั่งเซิร์ฟเวอร์) 34

บทที่4 การทดลองและผลการทดลอง 37
4.1 โปรแกรมทางฝั่งผู้เรียกหรือฝั่งไคลเอนต์ 37
4.2 โปรแกรมทางฝั่งผู้ถูกเรียกเรียกหรือฝั่งเซิร์ฟเวอร์ 43

บทที่5 บทวิจารณ์และบทสรุป 49
5.1 บทวิจารณ์ 49
5.2 สรุปผลการทดลอง 49
5.3 ปัญหาที่พบจากการทดลอง 49
5.4 แนวทางในการพัฒนา 49

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 แสดงการจำลองระบบโทรศัพท์ทางไกลผ่านอินเตอร์เน็ต 2

รูปที่ 2.1 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI Model 4

รูปที่ 2.2 แสดงการเชื่อมต่อแบบ Bus 7

รูปที่ 2.3 แสดงการเชื่อมต่อแบบ Star 7

รูปที่ 2.4 แสดงการเชื่อมต่อแบบ Ring 8

รูปที่ 2.5 แสดงประเภทของอีเทอร์เน็ตแบบ 10Base5 8

รูปที่ 2.6 แสดงประเภทของอีเทอร์เน็ตแบบ 10Base2 9

รูปที่ 2.7 แสดงประเภทของอีเทอร์เน็ตแบบ 10BaseT 9

รูปที่ 2.8 แสดงหลักการทำงานของโทเคนริง 10

รูปที่ 2.9 แสดงโทโปโลยีเอฟดีดีไอ 11

รูปที่ 2.10 แสดงเอฟทีพีไคลเอ็นต์ – เซิร์ฟเวอร์ 17

รูปที่ 2.11 แสดงรูปแบบของไอพีแอดเดรส 19

รูปที่ 2.12 แสดงการแบ่งคลาสเครือข่าย 19

รูปที่ 2.13 แสดงพารามิเตอร์ที่ใช้คู่กับคำสั่งในการหมุนหมายเลขโทรศัพท์ 22

รูปที่ 2.14 แสดงรูปของการ์ดเสียง 25

รูปที่ 2.15 แสดงกระบวนการแปลงสัญญาณของการ์ดเสียง 26

รูปที่ 2.16 แสดงค่าที่ได้จากการแซมปลิ้งสัญญาณ 26

รูปที่ 3.1 แสดงโฟลว์ชาร์ทการทำงานทั่วไปของโปรแกรม 28

รูปที่ 3.2 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Main Window 30

รูปที่ 3.3 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect By Manual 31

รูปที่ 3.4 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect From Phone Bookmark เมื่อทำการโทรศัพท์ออกตามชื่อที่เราต้องการ 32

รูปที่ 3.5 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect From Phone Bookmark เมื่อทำการบันทึกชื่อและเบอร์โทรศัพท์ลงไปในตัวโปรแกรม 33

รูปที่ 3.6 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect From Phone Bookmark เมื่อทำการแก้ไขชื่อและเบอร์โทรศัพท์ที่ได้บันทึกไว้ 34

รูปที่ 3.7 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรมทางฝั่งเซิร์ฟเวอร์ 35

รูปที่ 3.8 แสดงวงจรลดสัญญาณรบกวนหรือวงจรแมตช์ชิ่ง 36

รูปที่ 4.1 แสดงโปรแกรมหน้าต่างหลักหรือMain Window 37

รูปที่ 4.2 แสดงโปรแกรม Client Window เมื่อเริ่มต้นโปรแกรม 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 แสดงโปรแกรม Client Window เมื่อผู้ใช้ใส่ค่า IP Address , Port และเบอร์โทรศัพท์ พร้อมที่จะทำการติดต่อแล้ว 38
รูปที่ 4.4 แสดงโปรแกรม Client Window เมื่อผู้ใช้ทำการกดปุ่มรูปโทรศัพท์สีเขียว เพื่อทำการConnect 39
รูปที่ 4.5 แสดงโปรแกรม Phone Book Window เมื่อเริ่มต้นโปรแกรม 39
รูปที่ 4.6 แสดงการใส่ชื่อและเบอร์โทรลงในโปรแกรมPhone Book Window 40
รูปที่ 4.7 แสดงโปรแกรม Phone Book เมื่อทำการกดปุ่ม Save เพื่อบันทึกค่าไปแล้ว 41
รูปที่ 4.8 แสดงโปรแกรม Modify Window 41
รูปที่ 4.9 แสดงโปรแกรม Phone Book Window เมื่อแก้ไขค่าเรียบร้อยแล้ว 42
รูปที่ 4.10 แสดงโปรแกรมClient Window เมื่อClick ที่ชื่อในPhone Book Window 42
รูปที่ 4.11 แสดงโปรแกรม Server Window เมื่อทำการเริ่มต้นโปรแกรม 43
รูปที่ 4.12 แสดงโปรแกรม Server Window เมื่อทำการรับการติดต่อจากไคลเอ็นต์ และทำการสั่งโมเด็มให้หมุนไปยังหมายเลขโทรศัพท์ปลายทาง 43
รูปที่ 4.13 แสดงเวลาที่ดีเลย์ในระหว่างสนทนา 44
รูปที่ 4.14 แสดงสัญญาณอินพุทที่ป้อนให้กับระบบโดยเป็นสัญญาณซายน์ที่ความถี่ 500Hz แอมปลิจูด 10Vp-p 45
รูปที่ 4.15 แสดงสัญญาณเอาท์พุทของระบบเมื่อป้อนอินพุทสัญญาณซายน์ที่ความถี่ 500Hz แอมปลิจูด 10Vp-p 45
รูปที่ 4.16 แสดงสัญญาณอินพุทที่ป้อนให้กับระบบโดยเป็นสัญญาณซายน์ที่ความถี่ 2.5KHz แอมปลิจูด 10Vp-p 46
รูปที่ 4.17 แสดงสัญญาณเอาท์พุทของระบบเมื่อป้อนอินพุทสัญญาณซายน์ที่ความถี่ 2.5KHz แอมปลิจูด 10Vp-p 46
รูปที่ 4.18 แสดงสัญญาณอินพุทที่ป้อนให้กับระบบโดยเป็นสัญญาณซายน์ที่ความถี่ 3KHz แอมปลิจูด 10Vp-p 47
รูปที่ 4.19 แสดงสัญญาณเอาท์พุทของระบบเมื่อป้อนอินพุทสัญญาณซายน์ที่ความถี่ 3KHz แอมปลิจูด 10Vp-p 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

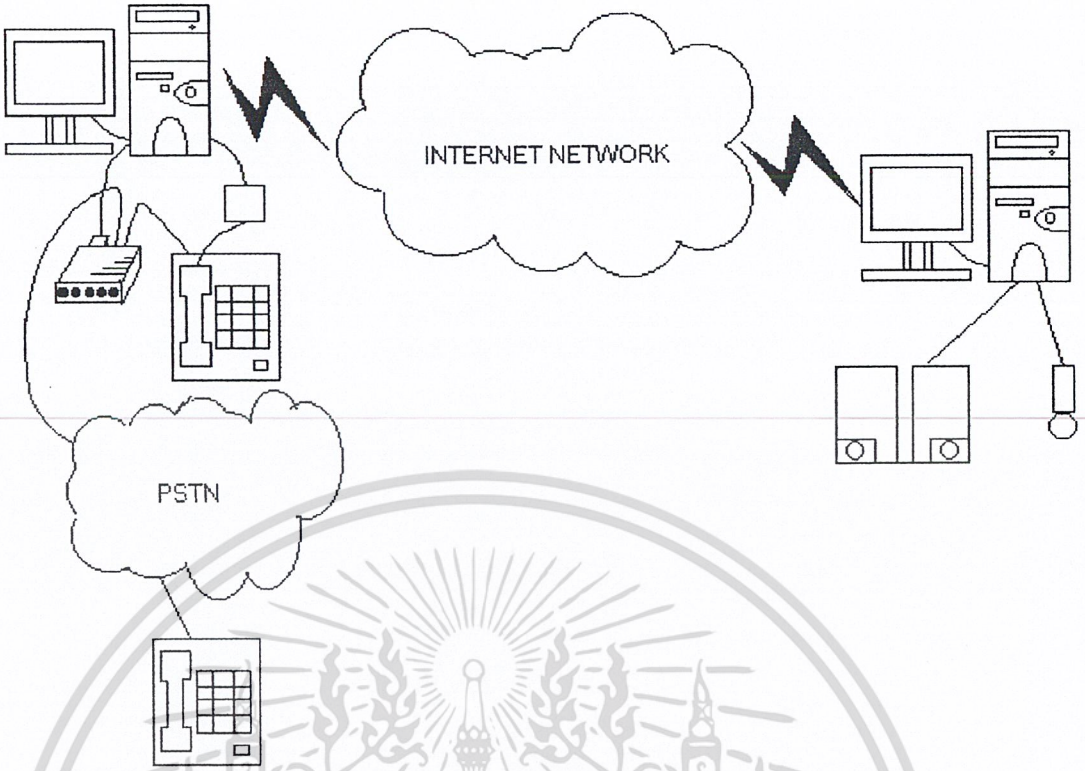
ในปัจจุบันคอมพิวเตอร์มีการพัฒนาไปอย่างรวดเร็ว มีประสิทธิภาพสูงในการประมวลผล อีกทั้งระบบการสื่อสารถูกพัฒนาไปอย่างมาก ทำให้สามารถส่งข้อมูลภาพ และเสียงได้ ทำให้มีการพัฒนาโปรแกรมประยุกต์(Software Application) ขึ้นมารองรับการใช้งานอย่างแพร่หลาย ทั้งโปรแกรมแบบสำหรับผู้ใช้คนเดียว และโปรแกรมแบบสำหรับผู้ใช้แบบกลุ่มงาน ทำให้เกิดการแลกเปลี่ยนข้อมูลภายในระบบคอมพิวเตอร์เครื่องเดียวกัน และการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ต่างเครื่องกัน

ซึ่งผลจากการที่คอมพิวเตอร์มีราคาถูกลงในปัจจุบัน แต่ประสิทธิภาพของเครื่องคอมพิวเตอร์และระบบเครือข่ายอินเตอร์เน็ตมีการพัฒนาก้าวไกลไปมาก ทำให้เกิดแนวความคิดในการใช้งานคอมพิวเตอร์เป็นระบบสื่อสารให้มีประสิทธิภาพสูงสุด โดยมีการพัฒนาเทคนิควิธีการ มีการพัฒนาการบีบอัดข้อมูล การส่งกระจายข้อมูลไปในเครือข่าย รวมทั้งการส่งข้อมูลแบบแพ็กเกต ทำให้เครือข่ายมีการขยายตัวจนเป็นที่แพร่หลายไปทั่วโลกมีการสร้าง Web Server และสถานีบริการต่างๆมากมาย เทคโนโลยีในการส่งสัญญาณข้อมูลในเครือข่ายก็มีมากขึ้น มีการส่งแบบ Real Audio การส่งแบบกระจายข่าว การส่งสัญญาณเสียงแบบแพ็กเกต รวมทั้งการใช้งานวิดีโอคอนเฟอเรนท์

จากเหตุผลดังกล่าว จึงนำมาสู่การศึกษาค้นคว้าโครงงานนี้ขึ้น โดยโครงงานนี้จะเป็นการเชื่อมต่อเส้นทางรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์สองเครื่อง โดยจะเป็นในรูปแบบของไคลเอ็นต์และเซิร์ฟเวอร์เพื่อรับส่งข้อมูลเสียง โดยจะเป็นการผนวกเครือข่ายคอมพิวเตอร์เข้ากับเครือข่ายโทรศัพท์พื้นฐาน (PSTN :Public Switched Telephone Network) ทำให้สามารถติดต่อรับส่งข้อมูลกันในลักษณะของสัญญาณเสียงระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ และระหว่างเครื่องคอมพิวเตอร์กับเครือข่ายโทรศัพท์พื้นฐาน โดยนำสัญญาณเสียงที่เป็นอนาล็อกมาแปลงเป็นดิจิตอลให้อยู่ในรูปแบบแพ็กเก็ตเสียงและส่งผ่านเครือข่ายโดยใช้โปรโตคอลTCP/IP (Transmission Control Protocol / Internet Protocol) และจะใช้ตัวโมเด็มทำการติดต่อระหว่างเครื่องคอมพิวเตอร์และเครือข่ายโทรศัพท์พื้นฐาน

ซึ่งสามารถแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงการจำลองระบบโทรศัพท์ทางไกลผ่านอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ส่วนที่ติดต่อกับเครือข่ายอินเตอร์เน็ท

2.1.1 ระบบเครือข่ายคอมพิวเตอร์ (Computer Network System)

การใช้งานเครื่องคอมพิวเตอร์มีอยู่ 2 แบบด้วยกันคือ แบบ Stand alone และแบบ LAN(Local Area Network) ซึ่งการใช้งานแบบ Stand alone เพียงเครื่องเดียวเหมาะสำหรับนักเรียนหรือธุรกิจเล็กๆไม่ใหญ่มาก แต่ถ้าเป็นธุรกิจขนาดกลางไปจนถึงขนาดใหญ่ดูจะไม่เหมาะสมเพราะต้องใช้ทรัพยากรร่วมกันเพื่อแก้ไขข้อมูลให้ทันสมัยและถูกต้องอยู่เสมอ ตลอดจนระบบรักษาความปลอดภัยของข้อมูล จากทั้งหมดที่กล่าวมา ระบบเครือข่ายคอมพิวเตอร์จึงเป็นทางเลือกที่ดีที่สุด

ระบบเครือข่ายคอมพิวเตอร์หมายถึง การนำเครื่องคอมพิวเตอร์ตั้งแต่2เครื่องขึ้นไปมาเชื่อมต่อกันเพื่อทำการแบ่งปันข้อมูล และใช้ทรัพยากรร่วมกันเช่น ไฟล์ข้อมูล ,เครื่องพิมพ์และโมเด็ม ตลอดจนการให้บริการฐานข้อมูล ซึ่งระบบเครือข่ายคอมพิวเตอร์สามารถแบ่งออกเป็น3ประเภทด้วยกันคือ

1. LAN (Local Area Network) หรือระบบเครือข่ายระดับท้องถิ่น

เป็นระบบเน็ตเวิร์คในระยะไม่เกิน 10 กิโลเมตร ไม่ต้องใช้โครงข่ายการสื่อสารขององค์การโทรศัพท์ โดยจะเป็นระบบเน็ตเวิร์คที่อยู่ภายในอาคารเดียวกัน หรือต่างอาคารกันในระยะใกล้ๆ

2. MAN (Metropolitan Area Network) หรือระบบเครือข่ายระดับเมือง

เป็นระบบเน็ตเวิร์คที่จะต้องใช้โครงข่ายการสื่อสารขององค์การ โทรศัพท์ หรือการสื่อสารแห่งประเทศไทย เพราะมีการติดต่อกันภายในเมือง เช่น มีเครื่องเวิร์กสเตชันอยู่ที่ลาดกระบัง มีการติดต่อสื่อสารกับเครื่องเวิร์กสเตชันที่อยู่ที่คอนเมือง

3. WAN (Wide Area Network) หรือระบบเครือข่ายระดับกว้างไกล

เป็นระบบเน็ตเวิร์คที่มีการติดต่อสื่อสารกันในระดับประเทศ ข้ามทวีป จะต้องใช้มีเดีย (Media) ในการสื่อสารขององค์การ โทรศัพท์ หรือการสื่อสารแห่งประเทศไทย เช่น Dial-up line , Leased line , ISDN(Integrated Service Digital Network) เป็นต้น ซึ่งสามารถส่งข้อมูลได้ทั้ง ข้อมูล ภาพ และเสียง ในเวลาเดียวกันได้

โดยระบบเน็ตเวิร์คเป็นระบบที่มีการใช้ทรัพยากรและใช้ข้อมูลร่วมกัน สามารถแบ่งประโยชน์ออกเป็น

1. การใช้ข้อมูลหรือโปรแกรมประยุกต์ร่วมกัน (Shared Application)
2. การใช้อุปกรณ์ฮาร์ดแวร์ร่วมกัน (Shared Hardware)
3. การกระจายการประมวลผล (Distributed Processing)
4. การติดต่อสื่อสารแบบรวดเร็ว (Rapid Communication)
5. การติดต่อสื่อสารระหว่างเวิร์กสเตชัน (Interconnected Station)

2.1.1.1 OSI Model

องค์กรมาตรฐานสากล ISO (International Standardization Organization) ได้กำหนดมาตรฐานของเครือข่ายโดยจัดแบ่งกิจกรรมของเครือข่ายออกเป็นงานย่อยๆ และกำหนดโมเดลแบ่งออกเป็นขั้นตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับเรียกว่า มาตรฐานOSI (Open System Interconnection) โดยที่จะแบ่งกิจกรรมที่ซับซ้อนในเครือข่ายออกเป็นงานย่อยๆก็จะช่วยในการออกแบบระบบ และการใช้งานเครือข่ายรวมถึงการเชื่อมโยงกันเป็นไปด้วยความสะดวก ตลอดจนมีวิธีการทำงานอยู่ในกรอบเดียวกัน แสดงได้ดังรูป

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

รูปที่ 2.1 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI Model

ในแต่ละขั้นของ OSI Model จะมีการติดต่อสื่อสารกันเป็นชั้นๆตามลำดับลงมาเช่น Application Layer ก็จะติดต่อสื่อสารกับ Presentation Layer ตามลำดับไปจนถึงชั้นแรกสุดคือ Physical Layer โดย OSI Model มีลักษณะสถาปัตยกรรมเป็นระบบเปิด เพราะมุ่งที่จะทำให้ระบบคอมพิวเตอร์ในหลาย ๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ โดยได้แบ่งขั้นตอนในการสื่อสารออกเป็น 7 ชั้น ซึ่งแต่ละชั้นจะเสมือนเชื่อมต่ออยู่กับชั้นที่เทียบเท่ากันของคอมพิวเตอร์อีกด้านหนึ่ง ส่วนการเชื่อมต่อกันจริง ๆ จะมีเพียงชั้นแรก (Physical Layer) ซึ่งเป็นชั้นล่างสุดเท่านั้นที่มีการรับส่งข้อมูลเกิดขึ้นผ่านสายส่งข้อมูลระหว่างคอมพิวเตอร์ทั้งสองระบบ ส่วนชั้นอื่น ๆ จะไม่ได้เชื่อมต่อกันจริง เพียงแต่ทำงานเสมือนกับว่ามีการติดต่อรับส่งข้อมูลกับกลไกในชั้นเดียวกันของคอมพิวเตอร์อีกด้านหนึ่ง

ข้อกำหนดรูปแบบแต่ละระดับชั้นของ OSI Model มีการระบุหน้าที่ที่ต้องปฏิบัติแน่นอน โดยจะต้องแบ่งเป็นไปตามกลุ่มบริการดังนี้

1. ระดับชั้นที่7 (Application Layer) เป็นระดับชั้นบนสุดของโมเดล เป็นส่วนที่จะทำให้การติดต่อระหว่างเครือข่ายกับผู้ใช้เป็นไปได้ตามต้องการ ตัวอย่างการใช้งานเช่น ระบบรับส่งจดหมายอิเล็กทรอนิกส์ (E-mail) , การถ่ายโอนแฟ้มข้อมูล (File Transfer) , การเชื่อมต่อเครือข่ายจากคอมพิวเตอร์ระยะไกล (Telnet)เป็นต้น

2. ระดับชั้นที่6 (Presentation Layer) เป็นระดับชั้นที่คอยรวบรวมข้อความและแปลรหัสหรือรูปแบบข้อมูลที่ได้รับมาจากระดับชั้นที่ 7เพื่อทำให้เป็นรูปแบบการสื่อสารแบบเดียวกันเพื่อช่วยลดปัญหาต่าง ๆ ที่เกิดขึ้นกับผู้ใช้งานในระบบ

3. ระดับชั้นที่5 (Session Layer) เป็นระดับชั้น ที่จัดการในเรื่องการติดต่อแต่ละครั้ง (session) ให้กับระบบคอมพิวเตอร์ทั้งสองฝ่าย โดยทำหน้าที่ตั้งแต่เริ่มการติดต่อ ดูแลในการส่งผ่านข้อมูลในการติดต่อครั้งนั้นๆให้เป็นไปโดยไม่มีปัญหาจนถึงการยกเลิกการติดต่อเมื่อเลิกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ระดับชั้นที่4 (Transport Layer) เป็นระดับชั้นที่ทำหน้าที่ควบคุมปริมาณและรายละเอียดวิธีการรับส่งข้อมูลให้เป็นไปตามข้อกำหนดที่ได้ตั้งไว้ และจัดการให้การเชื่อมโยงเครือข่ายเป็นไปด้วยความเรียบร้อย รวมทั้งควบคุมคุณภาพของการรับส่งข้อมูลให้มีมาตรฐานในระดับที่ตกลงกันของทั้งสองฝ่าย

5. ระดับชั้นที่3 (Network Layer) เป็นระดับชั้นที่ทำหน้าที่ควบคุมวิธีการส่งผ่านข้อมูลระหว่างเครือข่ายทั้งสองให้ถูกต้องและเป็นไปตามเส้นทางที่กำหนด โดยจะจัดการส่งผ่านแพ็กเก็ตข้อมูลผ่านอุปกรณ์ต่างๆไปยังเครือข่ายย่อยได้อย่างถูกต้องตามที่ต้องการ นอกจากนี้ยังทำหน้าที่จัดการดูแลเส้นทางในการส่งข้อมูล (Routing Table) เพื่อทำการเลือกเส้นทางที่ดีที่สุด สั้นที่สุด และใช้เวลาส่งน้อยที่สุด และกั้นหรือกรองแพ็กเก็ตข้อมูลที่ส่งไปยังเครือข่ายเดียวกันไม่ให้ข้ามไปยังเครือข่ายอื่นซึ่งจะช่วยลดปริมาณข้อมูลที่จะวิ่งไปบนเครือข่ายได้ส่วนหนึ่ง

6. ระดับชั้นที่2 (Data Link Layer) เป็นระดับชั้นที่ทำหน้าที่เรียกใช้หรือกำหนดช่องทางในการส่งข้อมูลที่ถูกต้อง รวมถึงการลำดับและอัตราการรับส่งข้อมูล (Flow Control) และสถานที่ที่จะส่งข้อมูลไป ทั้งนี้ในชั้นนี้จะเป็นชั้นแรกที่ทำการจัดการแปลงข้อมูลจากระดับบิต (Bit) ให้เป็นแพ็กเก็ตของข้อมูล โดยจะมีการเพิ่มข้อมูลเพื่อตรวจสอบความถูกต้อง ในกรณีที่ส่งข้อมูลออกไปหรือในกรณีที่อ่านข้อมูลเข้ามาก็จะตรวจสอบผ่านผลรวมของข้อมูล (Checksum) เพื่อดูว่าข้อมูลที่ได้รับมาถูกต้องครบถ้วนหรือไม่ และถ้าได้รับแพ็กเก็ตข้อมูลมาไม่ครบถ้วนก็จะไม่นำข้อมูลนั้นไปใช้งาน และจะทำการบอกให้ต้นทางส่งข้อมูลเดิมนั้นมาให้ใหม่

7. ระดับชั้นที่1 (Physical Layer) เป็นระดับชั้นที่ทำหน้าที่รับผิดชอบดูแลในรายละเอียดในการส่งข้อมูลทางด้านฮาร์ดแวร์ เช่น การควบคุมNetwork Interface Card , การส่งสัญญาณผ่านสายสัญญาณแบบต่างๆ เป็นต้น การเชื่อมต่อเข้ากับเครือข่ายแบบต่างๆภายในชั้นนี้จะจัดการสร้างสัญญาณทางไฟฟ้า, สัญญาณเสียง หรือสัญญาณที่จำเป็นในการสื่อสารโดยตรง

การใช้ข้อมูลหรือโปรแกรมประยุกต์ร่วมกัน สามารถจะทำให้ผู้ใช้หลายๆคนใช้โปรแกรมหรือข้อมูลร่วมกันได้ และเป็นการประหยัดเวลาในการติดตั้งโปรแกรม โดยจะทำการติดตั้งโปรแกรมใช้งานและเก็บข้อมูลไว้ที่เครื่องเซิร์ฟเวอร์(Server) เพียงเครื่องเดียว ทำให้เครื่องเวิร์คสเตชันไม่จำเป็นต้องมีดิสค์ไดร์ฟ(Disk Drive) หรือฮาร์ดดิสค์(Harddisk)ไว้เก็บข้อมูลก็สามารถใช้งานโปรแกรมและข้อมูลจากเครื่องเซิร์ฟเวอร์

การใช้อุปกรณ์ฮาร์ดแวร์ร่วมกัน ในระบบเน็ตเวิร์คเป็นระบบที่สามารถที่จะใช้อุปกรณ์ฮาร์ดแวร์ร่วมกันได้ดีเช่น ในระบบมีเครื่องเวิร์คสเตชันอยู่ n เครื่อง แต่ละเครื่องมีงานที่ต้องพิมพ์ผ่านเครื่องพิมพ์ ซึ่งถ้าไม่มีระบบเน็ตเวิร์คก็ต้องใช้งบประมาณซื้อเครื่องพิมพ์จำนวน n เครื่อง แต่สำหรับระบบเน็ตเวิร์คแล้วสามารถใช้เครื่องพิมพ์จำนวน1-2เครื่องก็เพียงพอ โดยการทำเป็นเครื่อง Printer Server คอยรับคำสั่งงานพิมพ์จากเครื่องเวิร์คสเตชันทั้ง n เครื่อง นอกจากนี้ยังสามารถใช้งานอุปกรณ์โมเด็ม และฮาร์ดดิสค์ร่วมกันได้อีกด้วย

การกระจายการประมวลผล เนื่องจากโปรแกรมประยุกต์ที่ใช้งานต่างๆเช่น Word,Excel,Office เป็นต้น ได้ถูกติดตั้งบนฮาร์ดดิสค์ของเครื่องเซิร์ฟเวอร์ ส่วนเครื่องเวิร์คสเตชันจะมีเพียงจอภาพ ซีพียู และคีย์บอร์ดสำหรับการใช้งาน เมื่อผู้ใช้ของเครื่องเวิร์คสเตชันเครื่องใดเครื่องหนึ่งต้องการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Word เครื่องเซิร์ฟเวอร์จะทำการคัดลอก(Copy)โปรแกรมWordจากฮาร์ดดิสค์ของเครื่องเซิร์ฟเวอร์มาเก็บไว้ในหน่วยความจำ(Ram) ของเครื่องเวิร์คสเตชันเครื่องนั้น ก่อนที่ซีพียูของเครื่องเวิร์คสเตชันเครื่องนั้นจะทำการประมวลผลข้อมูลที่อยู่ภายในหน่วยความจำต่อไป ซึ่งเป็นการกระจายการประมวลผลได้อย่างมีประสิทธิภาพ

การติดต่อสื่อสารแบบรวดเร็ว ด้วยมาตรฐานIEEE 802.3 สามารถที่จะส่งข้อมูลด้วยอัตราเร็ว 10 เมกะบิตต่อวินาที ด้วยความเร็วระดับนี้ทำให้การปรับปรุง (Update) ข้อมูลเป็นไปได้อย่างรวดเร็ว นอกจากนี้ระบบ Ethernet ยังถูกพัฒนาให้สามารถส่งข้อมูลด้วยอัตราเร็วถึง 100 เมกะบิตต่อวินาที

2.1.1.2 การติดต่อสื่อสารระหว่างเครื่องเวิร์คสเตชัน

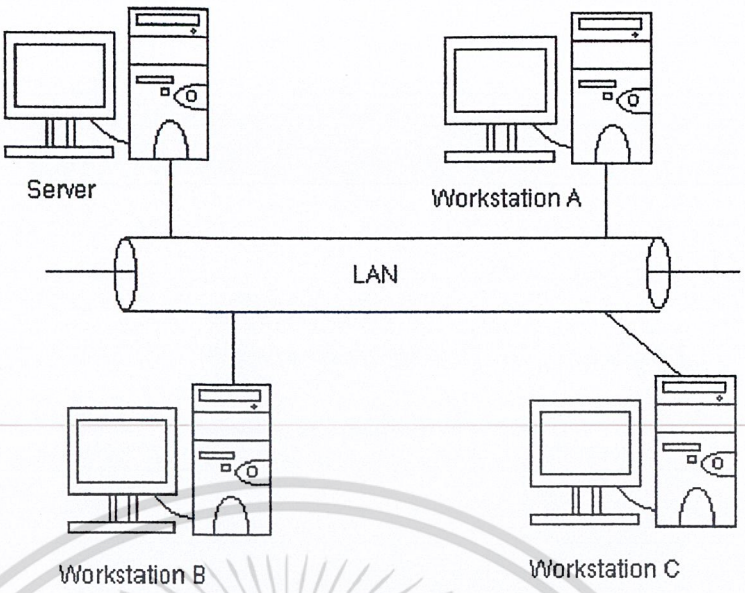
- **Peer to Peer** เป็นระบบที่เครื่องเวิร์คสเตชันทุกเครื่องบนระบบเน็ตเวิร์คมีฐานะเท่าเทียมกัน กล่าวคือทุกเครื่องสามารถที่จะใช้ไฟล์ข้อมูลในเครื่องอื่นๆได้ และสามารถให้เครื่องอื่นๆมาใช้ไฟล์ข้อมูลของเครื่องตนเองได้เช่นกัน โดยระบบPeer to Peer มีลักษณะการทำงานแบบ Distributed System โดยกระจายทรัพยากรต่างๆไปสู่เครื่องเวิร์คสเตชันอื่นๆ แต่จะมีปัญหาคือ เรื่องของการรักษาความปลอดภัย (Security) เพราะข้อมูลที่เป็นความลับก็จะถูกส่งออกไปสู่เวิร์คสเตชันเครื่องอื่นเช่นกัน ตัวอย่างของโปรแกรมที่มีความสามารถของ Peer to Peer คือ Windows for Workgroup , Personal Netware เป็นต้น

- **Client/Server** มีลักษณะการทำงานแบบ Distributed Processing หรือการประมวลผลแบบกระจาย โดยจะแบ่งการประมวลผลกันระหว่างเครื่องเซิร์ฟเวอร์กับเครื่องเวิร์คสเตชัน และเมื่อใดที่เครื่องเวิร์คสเตชันต้องการผลลัพธ์ของข้อมูลบางส่วนก็จะทำการเรียกไปยังเครื่องเซิร์ฟเวอร์เพื่อนำผลลัพธ์ของข้อมูลส่วนนั้นส่งกลับมายังเครื่องเวิร์คสเตชันเพื่อทำการประมวลผลข้อมูลนั้นต่อไป

2.1.1.3 รูปแบบการเชื่อมต่อของระบบเน็ตเวิร์ค (Topology Of Network)

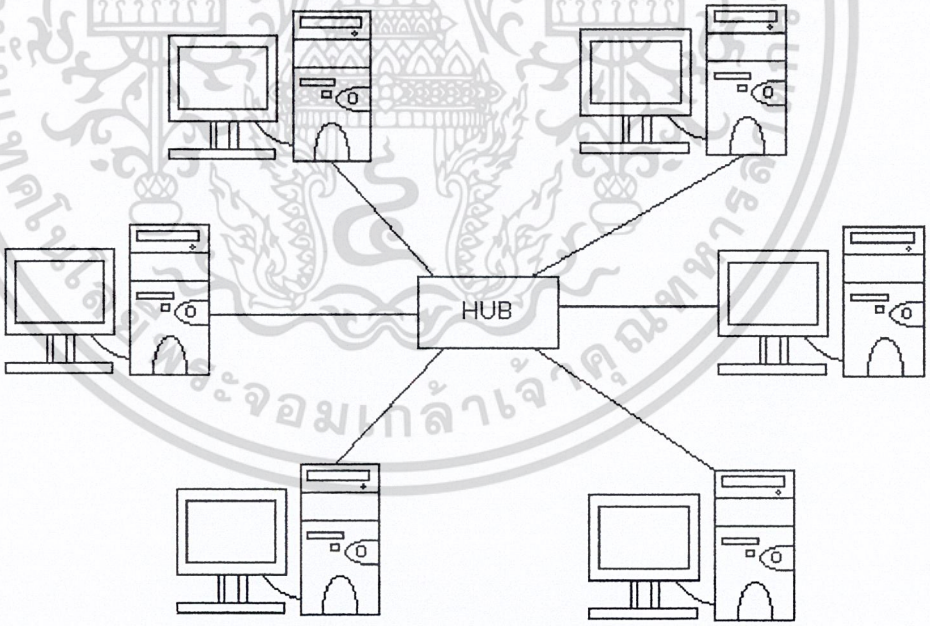
1. Bus การเชื่อมต่อแบบบัสจะมีสายในการรับส่งข้อมูลอยู่ 1สาย โดยเครื่องคอมพิวเตอร์ทั้งเซิร์ฟเวอร์และเวิร์คสเตชันทุกเครื่องจะต้องต่อกับสายหลักเส้นนี้ ซึ่งเครื่องคอมพิวเตอร์จะถูกมองเป็นNodeหนึ่ง เมื่อเครื่องคอมพิวเตอร์A (Node A) ต้องการจะส่งข้อมูลให้เครื่องคอมพิวเตอร์C (Node C) จะต้องส่งข้อมูลและแอดเดรสของNode C ลงไปบนบัสสายหลัก เมื่อคอมพิวเตอร์C (Node C) ได้รับข้อมูลก็จะนำข้อมูลนั้นไปใช้งานต่อ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการเชื่อมต่อแบบ Bus

2. Star การเชื่อมต่อแบบสตาร์นี้จะใช้ฮับ (Hub) เป็นศูนย์กลางในการเชื่อมต่อ กล่าวคือเครื่องคอมพิวเตอร์ทุกเครื่องจะต่อผ่านฮับด้วยสายเคเบิลที่ส่วนมากจะเป็นสาย UTP(Unshield Twisted Pair) และFiber Optic ในการส่งผ่านข้อมูล โดยฮับจะเปรียบเสมือนตัวทวนสัญญาณ(Repeater) นอกจากนี้ฮับบางรุ่นยังสามารถตรวจจับข้อมูล(Data Detection)ได้ ดังรูป

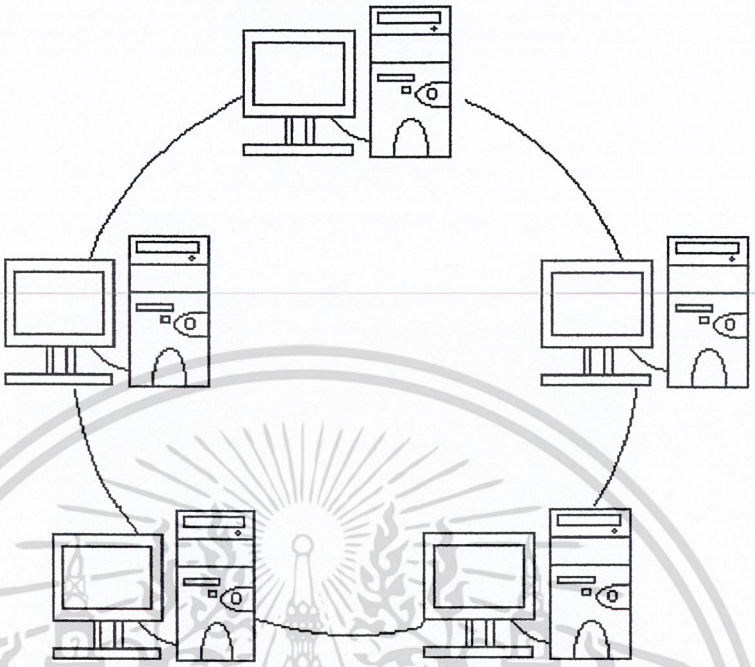


รูปที่ 2.3 แสดงการเชื่อมต่อแบบ Star

3. Ring การเชื่อมต่อแบบวงแหวนนี้จะเป็นการเชื่อมต่อจากเครื่องคอมพิวเตอร์หนึ่งไปยังอีกเครื่องหนึ่งจนครบวงจร โดยการส่งข้อมูลจะส่งออกมาที่สายสัญญาณวงแหวนโดยจะเป็นการส่งผ่านจากเครื่องคอมพิวเตอร์หนึ่งไปยังอีกเครื่องหนึ่งจนกว่าจะถึงปลายทาง เครื่องคอมพิวเตอร์ปลายทางก็จะดึงข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลนั้นออกไป จากนั้นเมื่อข้อมูลนี้กลับไปยังต้นทางเครื่องคอมพิวเตอร์ต้นทางก็จะทำการลบข้อมูลทิ้งไป เมื่อทำการตรวจแล้วว่าข้อมูลนี้สามารถส่งถึงปลายทางเรียบร้อยแล้ว แสดงได้ดังรูป



รูปที่ 2.4 แสดงการเชื่อมต่อแบบ Ring

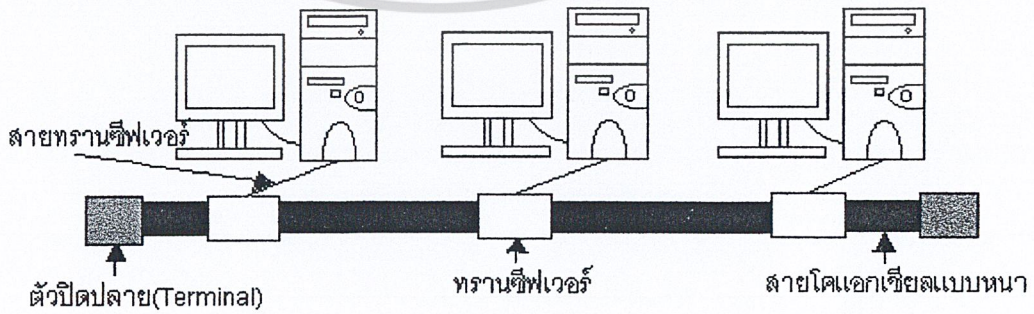
2.1.1.4 โปรโตคอลมาตรฐานของระบบเครือข่าย

หน่วยงานสากลที่มีหน้าที่ในการกำหนดมาตรฐานของการออกแบบผลิตภัณฑ์อิเล็กทรอนิกส์ , กำหนดรูปแบบการส่งสัญญาณ เป็นต้น คือ IEEE (Institute of Electrical and Electronic Engineers)

โดยIEEE ได้กำหนดเครือข่ายเฉพาะที่ หรือ แลน โดยใช้ตัวเลข802ตามด้วยตัวเลขย่อยเป็นรหัสประจำแต่ละมาตรฐานซึ่งแบ่งเป็น

1. อีเทอร์เน็ต (Ethernet) หรือ IEEE 802.3 เป็นเครือข่ายที่มีความเร็วในการรับส่งข้อมูล10 เมกะบิตต่อวินาที สถานีในเครือข่ายอาจมีโทโปโลยีแบบบัสหรือแบบดาว โดยIEEE ได้กำหนดมาตรฐานของอีเทอร์เน็ตซึ่งทำงานที่ความเร็ว10เมกะบิตต่อวินาทีไว้หลายประเภทตามชนิดของสายสัญญาณเช่น

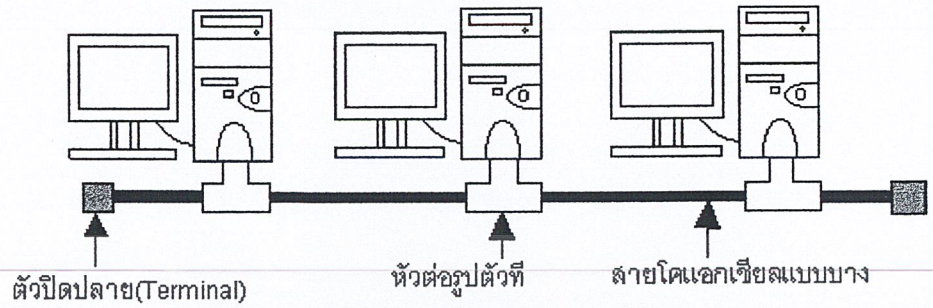
- **10Base5** อีเทอร์เน็ตโทโปโลยีแบบบัสซึ่งใช้สายโคแอกเชียลแบบหนา (Thick Ethernet) ซึ่งความยาวของสายในเซกเมนต์หนึ่งๆไม่เกิน 500 เมตร แสดงดังรูป



รูปที่ 2.5 แสดงประเภทของอีเทอร์เน็ตแบบ 10Base5

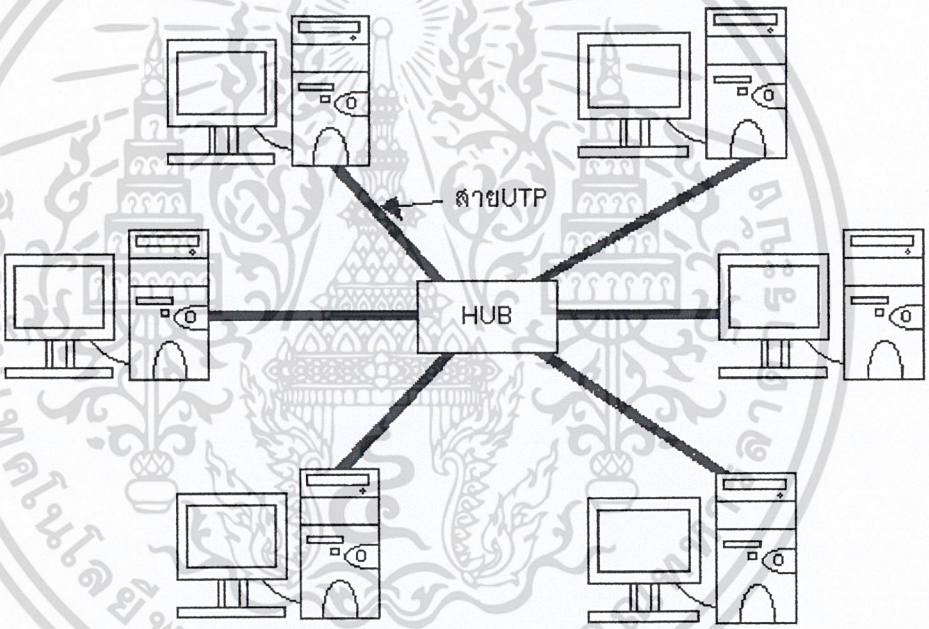
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 10Base2 อีเทอร์เน็ตโทโปโลยีแบบบัสซึ่งใช้สายโคแอกเชียลแบบบาง (Thin Ethernet) ซึ่งความยาวของสายในเซกเมนต์หนึ่งๆไม่เกิน 185 เมตร แสดงดังรูป



รูปที่ 2.6 แสดงประเภทของอีเทอร์เน็ตแบบ 10Base2

10BaseT อีเทอร์เน็ตโทโปโลยีแบบดาวซึ่งใช้ฮับเป็นศูนย์กลาง สถานีและฮับเชื่อมด้วยสายUTP (Unshield Twisted Pair) ด้วยความยาวไม่เกิน 100เมตร แสดงดังรูป



รูปที่ 2.7 แสดงประเภทของอีเทอร์เน็ตแบบ 10BaseT

โดยรหัสขึ้นต้นด้วย10 หมายถึงความเร็วสายสัญญาณ10เมกะบิตต่อวินาที คำว่าBase หมายถึงสายสัญญาณชนิดBaseband รหัสถัดมาหากเป็นตัวเลขหมายถึงความยาวสายต่อเซกเมนต์ในหน่วย100 เมตร (5=500,2แทนค่า185) หากตัวเป็นอักษรจะหมายถึงชนิดของสายเช่น T หมายถึงTwisted Pair หรือ F คือ Fiber Optic เป็นต้น

ส่วนมาตรฐานอีเทอร์เน็ตความเร็ว100เมกะบิตต่อวินาทีที่นิยมใช้ในปัจจุบันได้แก่ 100BaseTX และ100BaseFX สำหรับอีเทอร์เน็ตความเร็วสูงแบบกิกะบิตอีเทอร์เน็ตเริ่มแพรหลายมากขึ้น ตัวอย่างในปัจจุบันได้แก่ 1000BaseT,1000BaseLX และ1000BaseFX เป็นต้น

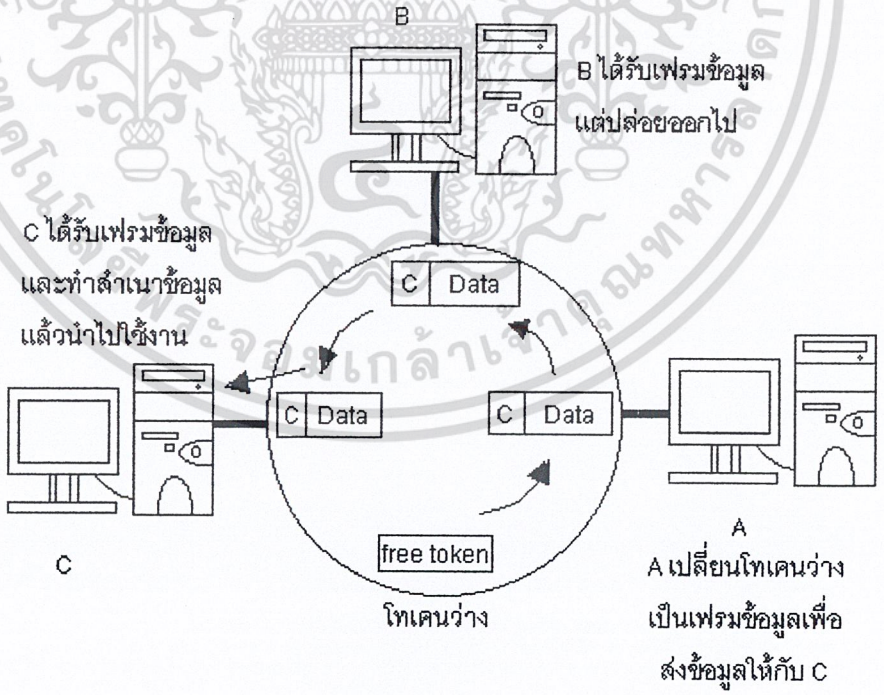
อีเทอร์เน็ตใช้โปรโตคอล ซีเอ็สเอ็มเอ/ซีดี (CSMA/CD:Carrier Sense Multiple Access with Collision Detection) เป็นตัวกำหนดขั้นตอนให้สถานีเข้าครอบครองสายสัญญาณในขณะเวลาหนึ่งจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียงสถานีเดียวที่เข้าครอบครองสายสัญญาณเพื่อรับส่งข้อมูล โดยสถานีที่ต้องการส่งข้อมูลจะต้องตรวจสอบสายสัญญาณว่ามีสถานีอื่นใช้อยู่หรือไม่ถ้าสายสัญญาณว่าก็จะรับส่งข้อมูลได้ทันที หากไม่ว่างก็ต้องคอยจนกว่าสายสัญญาณว่างจึงจะรับส่งข้อมูลได้ซึ่งขณะที่สถานีหนึ่งๆกำลังส่งข้อมูลก็ต้องตรวจสอบสายสัญญาณไปพร้อมกันด้วยเพื่อตรวจว่าในจังหวะเวลาที่ใกล้เคียงนั้นมีสถานีอื่นซึ่งตรวจพบว่าสายสัญญาณว่างและส่งข้อมูลมาหรือไม่ หากเกิดกรณีเช่นนี้ขึ้นแล้วข้อมูลจากทั้งสองสถานีจะเกิดการชนกัน(Collision) และนำไปใช้งานไม่ได้ สถานีจะต้องหยุดส่งข้อมูลและสุ่มหาเวลาเพื่อเข้าใช้สายสัญญาณใหม่

ในเครือข่ายอีเทอร์เน็ตที่มีสถานีที่มาต่อเป็นจำนวนมากจะพบว่าการทำงานจะล่าช้าเพราะแต่ละสถานพยายามยึดช่องสัญญาณเพื่อส่งข้อมูลและเกิดการชนกันตลอดเวลาโดยไม่สามารถกำหนดได้ว่าสถานีใดจะได้ใช้สายสัญญาณเมื่อเวลาใด ดังนั้นอีเทอร์เน็ตจึงไม่เหมาะกับการใช้งานในระบบเวลาจริง

2. โทเคนริง (Token Ring) หรือ IEEE 802.5 เป็นเครือข่ายที่ใช้โทโปโลยีแบบวงแหวนด้วยสายคู่ตีเกลียวหรือเส้นใยนำแสงโดยอัตราการรับส่งข้อมูลของโทเคนริงที่ใช้โดยทั่วไปคือ 4 และ 16 เมกะบิตต่อวินาที ซึ่งหลักการทำงานของโทเคนริงคือ จะมีเฟรมพิเศษที่เรียกว่า โทเคนว่าง (free token) วิ่งวนอยู่สถานีที่ต้องการรับส่งข้อมูลจะรอให้โทเคนว่างเดินทางมาถึงแล้วรับโทเคนว่างมาเปลี่ยนเป็น เฟรมข้อมูล (data frame) โดยใส่ flag แสดงเฟรมข้อมูลและบรรจุแอดเดรสของสถานีต้นทางและปลายทางตลอดจนข้อมูลอื่นๆจากนั้นสถานีจึงปล่อยเฟรมนี้ออกไป เมื่อทางสถานีปลายทางได้รับเฟรมข้อมูลก็จะทำสำเนาข้อมูลไว้และปล่อยเฟรมข้อมูลกลับมายังสถานีต้นทาง สถานีต้นทางจะตรวจสอบเฟรมข้อมูลหากพบว่าสถานีปลายทางได้เก็บข้อมูลไปแล้วก็จะทำการลบข้อมูลนั้นออกไปและปล่อยโทเคนว่างคืนสู่เครือข่ายให้สถานีอื่นมีโอกาสใช้ต่อไป แสดงได้ดังรูป



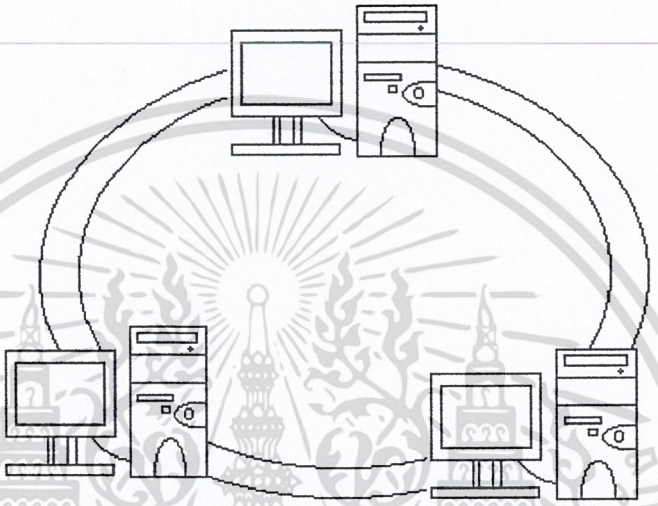
รูปที่ 2.8 แสดงหลักการทำงานของโทเคนริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกลไกแบบส่งผ่านโทเคนจัดอยู่ในประเภทประเมินเวลาได้ กล่าวคือสามารถคำนวณเวลาสูงสุดที่แต่ละสถานีมีสิทธิ์จับโทเคนริงเพื่อส่งข้อมูลได้ โทเคนริงจึงเหมาะกับระบบที่ต้องการความแน่นอนทางเวลาหรือทำงานแบบเวลาจริง

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

โทโปโลยีของเอฟดีดีไอเป็นแบบวงแหวนคู่ดังรูป



รูปที่ 2.9 แสดงโทโปโลยีเอฟดีดีไอ

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

2.1.2 ระบบเครือข่ายอินเตอร์เน็ต

2.1.2.1 ประวัติอินเตอร์เน็ต

จุดกำเนิดของอินเตอร์เน็ตมาจากความคิดเชิงยุทธศาสตร์ทางทหาร ในช่วงปี พ.ศ.2512เป็นช่วงสงครามเย็นระหว่างสหรัฐอเมริกากับรัสเซีย ทำให้เกิดความวิตกว่าถ้าศูนย์คอมพิวเตอร์ที่ใช้ควบคุมสั่งการถูกทำลายแล้วศูนย์คอมพิวเตอร์อื่นจะไม่สามารถทำงานทดแทนได้ เนื่องจากระบบคอมพิวเตอร์ที่ใช้กันมีหลายรูปแบบหลายยี่ห้อเช่น IBM,Apple,VAX เป็นต้น อีกทั้งระบบปฏิบัติการที่ใช้ก็ต่างกันเช่น UNIX,VM,MVS เป็นต้น ดังนั้นจึงเกิดความคิดที่จะเชื่อมโยงคอมพิวเตอร์ต่างๆเหล่านี้เข้าด้วยกัน ให้สามารถทำงานได้ตลอดแม้ว่าระบบคอมพิวเตอร์บางส่วนจะถูกทำลายไป แต่ส่วนที่เหลือต้องทำงานต่อได้

จากแนวคิดดังกล่าวทำให้มีการจัดตั้งหน่วยงานร่วมระหว่างสถาบันการศึกษากับฝ่ายวิจัยทางการทหารขึ้นมาเพื่อพัฒนาระบบสื่อสารและเครือข่ายคอมพิวเตอร์ภายใต้โครงการ US Advance Research Projects Agency โดยเรียกระบบเครือข่ายนี้ว่า APANET (Advanced Research Projects Agency NETwork) โดยใช้โปรโตคอล NCP (Network Control Protocol) แต่ต่อมาพบว่าโปรโตคอล NCP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีข้อจำกัดอยู่มากจึงมีการพัฒนาโปรโตคอลตัวใหม่ขึ้นมาเรียกว่า TCP/IP (Transmission Control Protocol/Internet Protocol) จนกระทั่งเป็นรูปแบบมาตรฐานแทน NCP

การพัฒนาเครือข่ายได้ดำเนินต่อมาเรื่อยๆจนกระทั่งกระทรวงกลาโหมของสหรัฐอเมริกาซึ่งเป็นผู้ริเริ่มได้ยกเลิกการสนับสนุนโดยให้ทางกลุ่มสถาบันการศึกษาดำเนินการต่อเองทำให้อินเตอร์เน็ตถูกพัฒนาให้มีความหลากหลายและขยายตัวจนเป็นเครือข่ายที่เสมือนครอบคลุมโลกไว้

2.1.2.2 สถาปัตยกรรมของอินเตอร์เน็ต

โครงสร้างของอินเตอร์เน็ตประกอบด้วยเครือข่ายย่อยจำนวนมากต่อเชื่อมกันผ่านเราเตอร์ (Router) ซึ่งเป็นอุปกรณ์เครือข่ายที่มีหน้าที่เลือกเส้นทางที่ดีที่สุดเพื่อนำส่งข้อมูลในรูปแพ็กเก็ต หากเปรียบเทียบกับการส่งจดหมายทางไปรษณีย์แล้ว เราเตอร์ทำหน้าที่เสมือนที่ทำการไปรษณีย์ พนักงานไปรษณีย์จะพิจารณาจุดหมายปลายทางของจดหมายและเลือกเส้นทางส่งจดหมายไปยังที่ทำการไปรษรีย์ถัดไปจนกว่าจดหมายจะถึงมือผู้รับ

ในยุคเริ่มต้นของอินเตอร์เน็ตจะใช้คำว่า เกตเวย์ (Gateway) แทนเราเตอร์ เพื่อสื่อความหมายถึงอุปกรณ์เชื่อมโยงเครือข่าย แต่ในปัจจุบันเราเตอร์และเกตเวย์เป็นอุปกรณ์ซึ่งมีหน้าที่แตกต่างกันคำว่าเกตเวย์ที่ใช้แต่เดิมนั้นในปัจจุบันใช้คำว่าเราเตอร์แทน โดยหน้าที่หลักของเราเตอร์คือ เชื่อมเครือข่ายที่ใช้โปรโตคอลเดียวกันและเลือกเส้นทางส่งข้อมูล ขณะที่เกตเวย์หมายถึงฮาร์ดแวร์หรือซอฟต์แวร์ซึ่งเป็นตัวแปลงระหว่างสองระบบที่มีโปรโตคอล โครงสร้างการจัดข้อมูล หรือสถาปัตยกรรมที่ต่างกัน

2.1.2.3 บริการในอินเตอร์เน็ต

ผู้ใช้อินเตอร์เน็ตสามารถเลือกโปรแกรมที่เหมาะสมเพื่อขอใช้บริการได้ตามต้องการ หากจะแยกประเภทของการบริการในอินเตอร์เน็ตแล้วสามารถแบ่งออกได้ดังนี้

- **ไปรษณีย์อิเล็กทรอนิกส์** (Electronic Mail) หรือ E-mail เป็นการรับส่งข้อความผ่านเครือข่ายคอมพิวเตอร์ ผู้ใช้สามารถรับส่งจดหมายด้วยโปรแกรมที่มีอย่างแพร่หลายเช่น pine , Eudora หรือ Microsoft Outlook เป็นต้น

- **ใช้โปรแกรมบนเครื่องคอมพิวเตอร์อื่น** บริการสำคัญในอินเตอร์เน็ตคือขอใช้คอมพิวเตอร์ระบบอื่นในที่ห่างไกล (Remote Login) ผู้ใช้ซึ่งมีบัญชีอยู่ในคอมพิวเตอร์เครื่องปลายทางสามารถขอเข้าใช้คอมพิวเตอร์เครื่องนั้นจากคอมพิวเตอร์ที่เชื่อมอยู่กับอินเตอร์เน็ตด้วยโปรแกรม เช่น telnet , pcAnywhere เป็นต้น

- **ถ่ายโอนแฟ้มข้อมูล** การถ่ายโอนแฟ้มข้อมูลด้วยโปรแกรม ftp เป็นบริการสำคัญอีกประเภทหนึ่งที่มีในอินเตอร์เน็ต เครือข่ายหลายแห่งเปิดบริการให้ผู้ใช้ภายนอกถ่ายโอนข้อมูลโดยไม่คิดมูลค่า แฟ้มที่ให้ถ่านโอนได้มีหลากหลายเช่น ข้อมูลทั่วไป,ข่าวประจำวัน,บทความ รวมทั้งโปรแกรมประยุกต์ต่างๆ

- **สนทนาทางเครือข่าย** การสนทนาทางเครือข่ายเป็นบริการในอินเตอร์เน็ตที่นิยมใช้อย่างแพร่หลาย ผู้ใช้สามารถสื่อสารกันแบบออนไลน์โดยพิมพ์ข้อความส่งถึงกันทางจอภาพหรือสนทนาพร้อมกันเป็นกลุ่มได้ โปรแกรมที่ใช้งานแพร่หลายคือ mIRC , icq เป็นต้น

- **จดหมายข่าวจดหมายเวียน** ระบบจดหมายข่าวให้บริการกระจายข่าวในกลุ่มสมาชิก ศูนย์บริการจะเก็บรายชื่อ (mailing list) ของสมาชิกไว้ เมื่อสมาชิกต้องการส่งข่าวไปยังสมาชิกรายอื่นก็เพียงแต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งไปรษณีย์อิเล็กทรอนิกส์ไปยังแอดเดรสเฉพาะเพื่อการกระจายข่าว ข่าวหรือจดหมายที่ส่งออกไปอาจเป็นการสนทนาทั่วไป การซักถาม หรือขอความช่วยเหลือ

- **บริการสืบค้นข้อมูล** อินเตอร์เน็ตในปัจจุบันยังคงขยายตัวออกไปอย่างต่อเนื่อง และมีเครือข่ายใหม่ๆเกิดขึ้นอยู่เสมอ ดังนั้นจึงต้องมีเครื่องมือที่ใช้ในการช่วยสืบค้นหาข่าวสารที่มากมายเกี่ยวกับเครือข่าย ซึ่งได้แก่ whois เป็นต้น

- **กระดานข่าว** ผู้ใช้เครือข่ายอินเตอร์เน็ตทั่วโลกสามารถแลกเปลี่ยนความคิดเห็นในหัวข้อต่างๆ ผ่านทางผ่านทางยูสเน็ต (Usenet) ซึ่งระบบข่าวแลกเปลี่ยนข่าวในอินเตอร์เน็ต ข่าวจะกระจายจากเครือข่ายหนึ่งไปสู่เครือข่ายอื่น ผู้อ่านยูสเน็ตนอกจากจะเป็นผู้อ่านรายงานข่าวแล้วยังสามารถทำหน้าที่เป็นคอลัมนิสต์เพื่อส่งข่าวได้ตลอดเวลา ยูสเน็ตจัดกลุ่มข่าวแยกตามหัวข้อที่เรียกว่า กลุ่มข่าว (news groups) โดยแบ่งเป็นหัวข้อย่อยเช่น sci (วิทยาศาสตร์) , comp (คอมพิวเตอร์) , soc (สังคม-วัฒนธรรม) , rec (นันทนาการ) เป็นต้น กลุ่มข่าวเหล่านี้เป็นกลุ่มข่าวใหญ่โดยยังมีกลุ่มข่าวย่อยประจำแต่ละกลุ่มอีกจำนวนมาก

- **เว็บ** World Wide Web หรือ WWW เป็นบริการข้อมูลข่าวสารในอินเตอร์เน็ตที่ได้รับความนิยมสูงสุดในปัจจุบัน เนื่องจากมีรูปแบบง่ายต่อการใช้งานและได้ผนวกบริการอื่นไว้เช่น เอฟทีพี , ยูสเน็ต หรืออีเมล เป็นต้น ลักษณะพิเศษของเว็บคือให้บริการทั้งภาพ เสียง หรือภาพเคลื่อนไหว ศูนย์บริการแทบทุกแห่งจะจัดสร้างโฮมเพจเพื่อให้บริการแก่ผู้ใช้ทั่วไป

2.1.3 โปรโตคอลในการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ผ่านอินเตอร์เน็ต

อินเตอร์เน็ตเป็นเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นๆ เชื่อมโยงเข้ามาใช้งานได้ แต่ปัญหาที่เกิดในการเชื่อมโยงเครือข่ายเข้าด้วยกันก็คือ เครือข่ายมีความต่างชนิด ต่างยี่ห้อ และระบบปฏิบัติการที่แตกต่างกัน มาตรฐาน TCP/IP เป็นสิ่งที่ถูกนำมาแก้ปัญหานี้ให้เป็นระบบเปิดที่สมบูรณ์แบบที่สามารถเชื่อมต่อเครือข่ายที่ทั่วโลกยอมรับ

2.1.3.1 ความหมายของ TCP/IP

TCP/IP เป็นโปรโตคอลที่ใช้งานอย่างแพร่หลายในทุกเครือข่ายไม่ว่าจะเป็นเครือข่ายเฉพาะที่ (LAN) หรือเครือข่ายในบริเวณกว้าง (WAN) TCP/IP เชื่อมกลุ่มเครือข่ายย่อยเข้าด้วยกันเป็นเครือข่ายขนาดใหญ่ที่เรียกว่า อินเตอร์เน็ต

TCP/IP ผ่านการออกแบบให้เป็นอิสระจากชนิดของคอมพิวเตอร์ ฮาร์ดแวร์ และระบบปฏิบัติการ กลไกของโปรโตคอลมีความเชื่อถือได้สูงและทำงานได้แม้ในบางภาวะที่การสื่อสารมีความผิดปกติ รวมทั้งสามารถเลือกเส้นทางส่งข้อมูลตามสภาพเครือข่ายได้ในกรณีที่บางเส้นทางเกิดความชำรุดเสียหาย

ชื่อ TCP/IP มาจากโปรโตคอลสองโปรโตคอลคือ ทีซีพี (TCP: Transmission Control Protocol) และ ไอพี (Internet Protocol) โดยไอพีทำหน้าที่กำหนดแอดเดรส จัดแบ่งขนาดข้อมูลให้พอเหมาะ และเลือกเส้นทางส่งข้อมูล ส่วนทีซีพีมีหน้าที่รับประกันความถูกต้องในการรับส่งข้อมูล

TCP และ IP ไม่ได้เป็นเพียงสองโปรโตคอลที่มีอยู่เท่านั้น หากแต่ยังมีโปรโตคอลสนับสนุนอีกเป็นจำนวนมากและจัดรวมกันเป็น ชุดโปรโตคอลทีซีพี/ไอพี (TCP/IP protocol suite)

2.1.3.2 ชุดโปรโตคอลทีซีพี/ไอพี (TCP/IP protocol suite)

จากที่กล่าวมาแล้วว่า TCP/IP ประกอบด้วยชุดข้อกำหนดต่างๆ ซึ่งสามารถรวมเป็นกลุ่มได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocol) ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลระหว่าง เครื่องคอมพิวเตอร์ 2 เครื่อง แบ่งย่อยได้เป็น 2 ชนิดคือ

1. TCP (Transmission Control Protocol) ทำหน้าที่นำส่งข้อมูลโดยรับประกันความเชื่อถือ ทีซีพีด้านส่งต้องส่งแพ็กเก็ตซ้ำใหม่หากแพ็กเก็ตสูญหาย ทีซีพีด้านรับมีหน้าที่จัดแพ็กเก็ตให้ถูกต้องตามลำดับและกำจัดแพ็กเก็ตที่ซ้ำซ้อน ทีซีพีเป็นโปรโตคอลแบบ connection oriented คือต้องสถาปนาการเชื่อมต่อระหว่างสถานีต้นทางและสถานีปลายทางก่อนการรับส่งข้อมูล

ทีซีพีต้นทางจัดแบบข้อมูลเพื่อส่งให้ไอพีดำเนินการ ทีซีพีปลายทางเมื่อรับแพ็กเก็ตจากไอพีก็จะส่งต่อให้โปรโตคอลประยุกต์ไปดำเนินการต่อไป

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

ลักษณะเด่นของยูดีพีคือใช้การประมวลผลต่ำกว่าทีซีพี เนื่องจากเฮดเดอร์มีขนาดเล็กและไม่ต้องสถาปนาการเชื่อมต่อ

- กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol) ทำหน้าที่ในการพิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือขนาดใหญ่ กลุ่มข้อมูลจะทำการแบ่งเป็นส่วนย่อยๆ ก่อนส่งออกไป เมื่อถึงปลายทางก็จะทำการรวบรวมข้อมูลที่แบ่งกลับมาเป็นข้อมูลที่สมบูรณ์อีกครั้ง โดยประกอบด้วย

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

2. ICMP (Internet Control Message Protocol) เป็นโปรโตคอลที่ใช้รายงานสถานะความผิดพลาดที่เกิดขึ้น

3. RIP (Routing Information Protocol) เป็นข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับทำการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุดและเหมาะกับข้อมูลมากที่สุด

4. OSPF (Open Shortest Path First) เป็นข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากเส้นทางที่สั้นที่สุดก่อน

- กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเครือข่าย (Network Address) ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทางโดยไม่ว่าโครงข่ายจะใหญ่โตหรือมีจำนวนมากเพียงใดก็ตามที่อยู่จะต้องไม่ซ้ำกัน กลุ่มข้อกำหนดมีดังนี้

1. ARP (Address Resolution Protocol) เป็นโปรโตคอลที่เป็นข้อกำหนดรูปแบบที่ใช้พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดกรณีที่อยู่ซ้ำกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. RARP (Reverse Address Resolution Protocol) เป็นโปรโตคอลที่เป็นข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เช่นเดียวกับ ARP แต่จะทำงานตรงข้ามกับ ARP

3. DNS (Domain Name System) เป็นโปรโตคอลที่ให้บริการสอบถามไอพีแอดเดรสหรือชื่อโดเมนก่อนการติดต่อไปยังโฮสต์ใดๆ ชื่อโฮสต์จะถูกส่งไปสอบถามผ่านเซิร์ฟเวอร์ที่ให้บริการดีเอ็นเอสเพื่อขอไอพีแอดเดรสกลับมา นอกจากนี้ยังให้บริการเกี่ยวกับฐานข้อมูลประจำเครื่องด้วย

- กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocol) และสนับสนุนข้อมูลสถานะเพื่อนำไปใช้ในการเลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1. EGP (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะทำการถ่ายโอนข้อมูลเส้นทางกันระหว่าง gateway กับเครือข่ายภายนอกเพื่อทำการสื่อสาร

2. GGP (Gateway to Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำการถ่ายโอนข้อมูลเส้นทางกันระหว่าง gateway กับ gateway

3. IGP (Interior Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

- กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Service) ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดนี้ประกอบด้วย

1. BOOTP (Bootstrap Protocol) ให้บริการบูตระบบสำหรับสถานีที่ไม่มีดิสค์ สถานีที่เป็นบูตพีไคลเอ็นต์จะติดต่อกับบูตพีเซิร์ฟเวอร์เพื่อขอถ่ายโอนระบบปฏิบัติการ

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

3. TFTP (Trivial File Transfer Protocol) ทำหน้าที่เป็นโปรโตคอลถ่ายโอนแฟ้มข้อมูลเช่นเดียวกับเอฟทีพี แต่ทีเอฟทีพีใช้บริการผ่านยูดีพี ซึ่งประโยชน์ของการใช้งานทีเอฟทีพีได้แก่การใช้ในสถานีไร้ดิสค์ (Diskless Workstation) ซึ่งไม่มีดิสค์ประจำตัว สถานีไร้ดิสค์จะบูตระบบด้วยโปรโตคอลบูตพี และใช้ทีเอฟทีพีเพื่อถ่ายโอนระบบปฏิบัติการ

4. TELNET (Telecommunication Network) เป็นโปรโตคอลสำหรับขอเข้าใช้โฮสต์ระยะไกลหรือ รีโมตล็อกอิน (Remote Login) เทลเน็ตให้บริการเข้าใช้คอมพิวเตอร์ในเครือข่ายโดยเสมือนกับว่ากำลังทำงานอยู่ที่เทอร์มินัลของคอมพิวเตอร์เครื่องนั้น เทลเน็ตเซิร์ฟเวอร์ที่คอมพิวเตอร์ปลายทางจะรอรับการขอใช้บริการจากเทลเน็ตไคลเอ็นต์ ผู้ขอใช้เทลเน็ตจะต้องมีบัญชีประจำเครื่องที่ให้บริการเทลเน็ต

- กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญจัดทำไว้บนเครือข่ายที่น่าสนใจมีดังนี้

1. NFS (Network File System) เป็นโปรโตคอลที่พัฒนาขึ้นเพื่อใช้งานในระบบรีโมตดิสค์เพื่อให้คอมพิวเตอร์เครื่องหนึ่งสามารถใช้แฟ้มข้อมูลที่เก็บอยู่ในดิสค์ของอีกเครื่องหนึ่งเสมือนว่าเป็นแฟ้มที่อยู่ในดิสค์ของเครื่องนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. NIS (Network Information System) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Account ข้ามเครือข่าย

3. RPC (Remote Procedure Call) เป็นข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะไกล

4. SNMP (Simple Network Management Protocol) ทำหน้าที่เป็นโปรโตคอลช่วยบริหารระบบ เช่น เก็บรวบรวมข้อมูลการทำงานของอุปกรณ์และคอมพิวเตอร์ในเครือข่าย ตรวจสอบปริมาณข้อมูลที่ไหลเวียนหรือช่วยวิเคราะห์หาข้อผิดปกติในระบบ เป็นต้น

2.1.3.3 แบบอ้างอิงTCP/IP

เครือข่ายคอมพิวเตอร์มีแบบอ้างอิงที่ใช้เป็นมาตรฐานคือ OSI Model ในขณะที่ TCP/IP มีแบบอ้างอิงเมื่อเทียบกับ OSI Model ดังนี้

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

- **Transport Layer** ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายเปรียบเทียบได้กับ Session Layer รวมกับ Transport Layer โดยโปรโตคอลจะมีซ็อคเก็ต (Socket) เป็นจุดปลายในการสื่อสาร โดยซ็อคเก็ตประกอบด้วยหมายเลขของเครื่องคอมพิวเตอร์และพอร์ทของเครื่องที่ต้องการข้อมูลส่งเข้าไปถึง ในชั้นนี้มีการรองรับให้ถึงที่หมายและลำดับข้อมูลไม่มีการซ้ำซ้อน โดยในชั้นนี้จะมีโปรโตคอลหลัก 2 ตัว คือ TCP และ UDP

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

- **Network Interface Physical Layer** ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็นส่วน ๆ เรียกว่าเฟรม หรือ แพ็คเก็ต และส่งข้อมูลออกไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

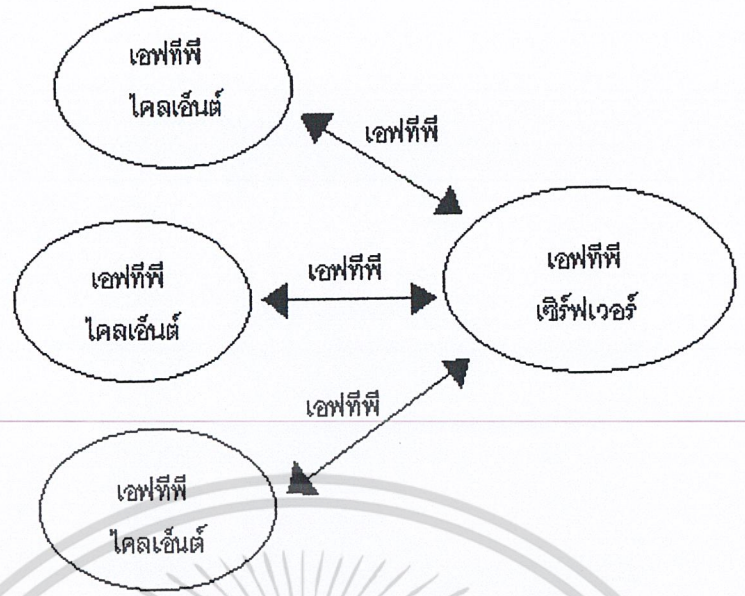
2.1.3.4 TCP/IP Client-Server

โปรโตคอลใน TCP/IP อาศัยหลักการทำงานตามแบบ Client-Server ซึ่งมีองค์ประกอบ 3 ส่วนคือ

1. โปรเซสไคลเอ็นต์ซึ่งทำหน้าที่ขอบริการ
2. โปรเซสเซิร์ฟเวอร์ซึ่งทำหน้าที่ให้บริการ
3. โปรโตคอลที่ใช้สื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.10 แสดงเอฟทีพีไคลเอ็นต์ – เซิร์ฟเวอร์

จากรูปแสดงการเชื่อมโยงระหว่างไคลเอ็นต์ 3 โปรเซส ซึ่งขอใช้บริการถ่ายโอนแฟ้มข้อมูลจากเซิร์ฟเวอร์ผ่านเอฟทีพี โปรเซสไคลเอ็นต์และเซิร์ฟเวอร์ไม่จำเป็นต้องอยู่ต่างเครื่องกัน เนื่องจากระบบปฏิบัติการส่วนใหญ่ที่ใช้ทีซีพี/ไอพีมักทำงานแบบผู้ใช้หลายคนและพร้อมกันหลายงาน แต่ละไคลเอ็นต์จึงอาจเป็นโปรเซสของผู้ใช้ต่างบุคคลที่ทำงานภายในเครื่องเดียวกันและขอบริการ ไปยังเซิร์ฟเวอร์ต่างเครื่องหรือเครื่องเดียวกันกับไคลเอ็นต์ก็ได้

2.1.3.5 ข้อเปรียบเทียบและความสัมพันธ์ระหว่างโปรโตคอล TCP และ UDP

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

การติดต่อสื่อสารระหว่างกันจะต้องเป็นแบบ connection-oriented คือต้องมีการสร้างการติดต่อกันเป็นเซสชั่น (session) ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมๆ กัน (Full Duplex) เมื่อแอพพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้นโพรเซสเลเยอร์ (Process Layer) ติดต่อไปและมีการสร้างช่องสัญญาณข้อมูลผ่านพอร์ตที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโปรโตคอล TCP ในระหว่างการรับส่งข้อมูลนี้ โปรโตคอล TCP จะเพิ่มขบวนการสอบทานข้อมูลเพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น ซึ่งทำให้ความน่าเชื่อถือของการส่งผ่านข้อมูลโดยโปรโตคอล TCP จะมีมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบมากกว่าในการทำงานเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
46569

ส่วนโปรโตคอล UDP โนการทำงานในชั้นการทำงานระหว่างโฮสต์ นอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ก็ยังมีโปรโตคอล UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบที่ทั้ง 2 ด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกันระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลแบบนั้น ๆ ด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอพพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการการตรวจสอบขึ้นมาเอง

2.1.3.6 พอร์ตของ TCP/IP

เมื่อพิจารณาข้อมูลที่ไอพีลำเลียงถึงสถานีปลายทางและส่งผ่านตามระดับชั้นขึ้นมาถึง Transport Layer ในไอพีเฮดเดอร์จะระบุชนิดของโปรโตคอลว่าต้องการเรียกใช้บริการของทีซีพีหรือยูดีพี ต่อจากนั้นทีซีพีหรือยูดีพีจะส่งข้อมูลไปยังโปรโตคอลระดับบนตามหมายเลขประจำโปโตคอล หมายเลขที่ใช้ระบุโปโตคอลประยุกต์เป็นตัวเลขขนาด 16 บิต ทำให้สามารถมีหมายเลขได้ตั้งแต่1 ถึง 65536 และเรียกว่า พอร์ต (Port) หรือกล่าวได้ว่าพอร์ตเสมือนเป็นแอดเดรสประจำโปรโตคอลในชั้นประยุกต์

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

โดยโปรโตคอลประยุกต์หนึ่งๆอาจเรียกใช้บริการได้ทั้งทีซีพีและยูดีพีพร้อมกัน โดยใช้หมายเลขพอร์ตเดียวกันแต่เป็นอิสระต่อกัน เช่น echo มีหมายเลขพอร์ตเท่ากับ 7 ทั้งทีซีพีและยูดีพี หากมีการขอบริการทีซีพีและยูดีพีพร้อมกันที่พอร์ตนี้ โปรเซสที่เกิดขึ้นจะเป็นอิสระต่อกัน

2.1.3.7 ซ็อกเก็ตของ TCP/IP

ซ็อกเก็ต (socket) หรือ ซ็อกเก็ตแอดเดรส (socket address) หมายถึงคู่ของไอพีแอดเดรสและหมายเลขพอร์ต ทีซีพีและยูดีพีอาศัยซ็อกเก็ตเป็นตัวแยกแยะโปรเซสต้นทางและปลายทางที่ติดต่อระหว่างกัน หากพิจารณาถึงโปรโตคอลประยุกต์ซึ่งอนุญาตให้บริการได้หลายเซสชันพร้อมๆกัน และจำเป็นต้องแยกแยะเซสชันที่เกิดขึ้นให้ได้เพื่อให้ข้อมูลรับส่งได้ถูกต้องตามคู่ต้นทางและปลายทาง

โดยเซิร์ฟเวอร์ปลายทางจะใช้หมายเลขพอร์ตตามพอร์ตมาตรฐานประจำแอพลิเคชันสำหรับสร้างซ็อกเก็ต แต่ไคลเอ็นต์ต้นทางจะเลือกหมายเลขพอร์ตนอกเหนือจากพอร์ตมาตรฐานมาใช้เป็นซ็อกเก็ตต้นทาง

2.1.3.8 หมายเลข IP Address

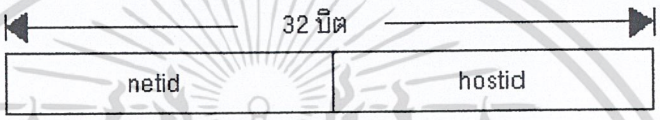
อุปกรณ์ที่เชื่อมเข้าเครือข่ายและสามารถทำงานได้ตามข้อกำหนดของ TCP/IP จะต้องมีแอดเดรสประจำอุปกรณ์นั้น โดยไอพีรุ่นสี่ได้กำหนดให้ใช้ไอพีแอดเดรสขนาด 32 บิต อุปกรณ์ที่เชื่อมกับอินเตอร์เน็ตจะมีไอพีแอดเดรสขนาด32บิตประจำอินเตอร์เฟสที่ไม่ซ้ำกัน อุปกรณ์ที่มีหลายอินเตอร์เฟสจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีไอพีแอดเดรสหลายค่าตามจำนวนอินเตอร์เฟสที่มีอยู่โดยไม่ซ้ำค่ากัน แต่คอมพิวเตอร์หรือโฮสต์ที่พบโดยทั่วไปมักมีเพียงอินเตอร์เฟสเดียวจึงมักเรียกไอพีแอดเดรสว่าเป็นแอดเดรสประจำโฮสต์

แอดเดรสขนาด32บิตมีจำนวนแอดเดรสรวมเท่ากับ 4291967296 แต่เมื่อนำมาจัดสรรแล้วจะไม่สามารถใช้งานได้ครบทั้งหมด ไอพีแอดเดรสนิยมเขียนในรูปเลขฐานสิบตามแบบ dotted decimal โดยแบ่งตัวเลข32บิตออกเป็น4ไบต์ แต่ละไบต์แทนด้วยเลขฐานสิบหนึ่งตัวและคั่นแต่ละไบต์ด้วยเครื่องหมายจุด เช่น แอดเดรส 10011110 01101100 00000010 00000001 จะได้เป็น 158.108.2.1

แอดเดรสขนาด32บิตประกอบขึ้นจากหมายเลข 2 ส่วนคือ หมายเลขเครือข่าย (Network Identifier) และ หมายเลขโฮสต์ (Host Identifier) หมายเลขเครือข่ายใช้สำหรับจัดคลาสเครือข่าย ส่วนหมายเลขโฮสต์ใช้ระบุหมาเลขโฮสต์ในเครือข่าย ดังนั้นไอพีแอดเดรสจึงแบ่งได้เป็น2ส่วนดังรูป จำนวนบิตที่ใช้สำหรับหมายเลขเครือข่ายและหมายเลขโฮสต์จะขึ้นกับคลาสที่สังกัด

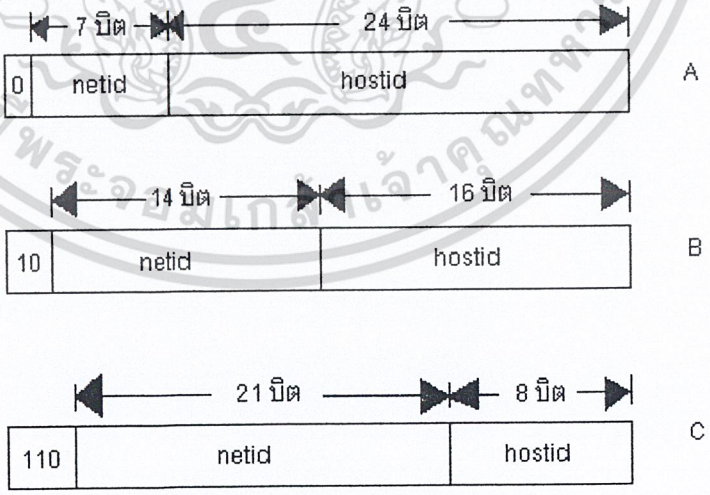


รูปที่ 2.11 แสดงรูปแบบของไอพีแอดเดรส

ในปัจจุบันฟิลด์กำหนดหมายเลขเครือข่ายนิยมเรียกว่า พรีฟิกซ์เครือข่าย (network-prefix) เพราะทุกโฮสต์ในเครือข่ายจะต้องมีพรีฟิกซ์หรือบิตนำหน้าเหมือนกัน

2.1.3.8.1 การจัดคลาสเครือข่าย

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



รูปที่ 2.12 แสดงการแบ่งคลาสเครือข่าย

การจัดแบ่งคลาสโดยใช้พรีฟิกซ์เป็นการผนวกข้อมูลเพื่อใช้ในการเลือกเส้นทาง เช่น หากตรวจสอบพบว่าพรีฟิกซ์ 2 บิตแรกมีค่าเป็น 10 แสดงว่าเป็นแอดเดรสในคลาส B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.8.2 ลักษณะสำคัญของแต่ละคลาส

จำนวนเครือข่ายในแต่ละคลาสและจำนวนโฮสต์สูงสุดที่มีได้สามารถคำนวณได้จากจำนวนบิตที่ใช้งานตามสูตร 2^n เมื่อ n คือจำนวนบิต เช่นในคลาส B มีเลขโฮสต์จำนวน16บิต จึงมีโฮสต์ได้ไม่เกิน 2^{16} บิตซึ่งเท่ากับ 65536 แต่เลขโฮสต์ที่ทุกบิตเป็น 0 และ 1 จะสงวนไว้ใช้งานเฉพาะกรณี จำนวนโฮสต์จึงลดไป2 โฮสต์ในทุกๆเครือข่ายหรือมีโฮสต์ได้ไม่เกิน 2^{16}-2 = 65534 เป็นต้น

คลาส A

เครือข่ายในคลาส A มีบิตซ้ายสุดเป็น 0 และใช้7บิตถัดมากำหนดเลขเครือข่าย ส่วนอีก24บิตเป็นเลขโฮสต์ คลาส A จึงมีจำนวนเครือข่ายได้ 2^7 หรือ128ค่า แต่เครือข่าย 0.0.0.0 และ 127.0.0.0 ถูกสงวนไว้เป็นแอดเดรสเฉพาะคือ แอดเดรส 0.0.0.0 ใช้เป็นแอดเดรสกำหนดเส้นทางโดยปริยาย (default route) ส่วนแอดเดรส 127.0.0.0 เป็นแอดเดรสลูปแบ็ค ดังนั้นจำนวนเครือข่ายในคลาส A จึงมีได้ 126 เครือข่าย

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

คลาส B

เครือข่ายในคลาส B มีบิตเริ่มเป็น 10 และใช้14บิตถัดมาเป็นตัวกำหนดเลขเครือข่าย จำนวนบิตที่กำหนดเลขโฮสต์มีขนาด16บิต คลาส B จึงมีสามาชิกเครือข่ายได้ 2^{14}-2 หรือ 16382ค่า แต่ละเครือข่ายมีเลขโฮสต์ได้ 2^{16}-2 หรือ 65534 แอดเดรส

เครือข่ายในคลาส B มักจัดสรรให้กับหน่วยงานขนาดกลาง ในปัจจุบันมีเครือข่ายในคลาส B เหลือไม่มากนัก และมักไม่จัดสรรเครือข่ายในคลาสนี้ให้กับผู้ขอจดทะเบียนรายใหม่หากไม่มีความจำเป็นอย่างแท้จริง

คลาส C

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

จำนวนแอดเดรสที่มีจำกัดเพียง 254 แอดเดรส ทำให้เครือข่ายในคลาสนี้เหมาะกับหน่วยงานขนาดเล็ก

2.2 ส่วนติดต่อกับตัวโมเด็ม

2.2.1 ชุดคำสั่ง AT ของโมเด็ม

โมเด็มส่วนใหญ่ จะถูกควบคุมการทำงานโดยชุดคำสั่งที่ส่งออกมาทางพอร์ตสื่อสารอนุกรม ซึ่งชุดคำสั่งเหล่านี้มีต้นกำเนิดมาจากบริษัท Hayes และบริษัทผู้ผลิตโมเด็มรายอื่นๆก็ได้พัฒนาเพิ่มเติมคำสั่งพิเศษเข้ามาใช้กันอีกหลายคำสั่ง เพื่อให้เหมาะสมกับการทำงานของโมเด็มของตน แม้กระทั่งยี่ห้อเดียวกันแต่รุ่นต่างกัน ชุดคำสั่งที่พิเศษเหล่านี้ก็อาจต่างกันไปด้วย ซึ่งคำสั่งเหล่านี้สามารถสังเกตได้จากขั้นตอนการเซตอัพประเภทโมเด็มในโปรแกรมสื่อสาร ผู้ใช้จะต้องเลือกยี่ห้อและรุ่นของโมเด็มให้ถูกต้อง เพื่อที่ว่าโปรแกรมสื่อสารจะได้ส่งคำสั่งเริ่มต้นที่ถูกต้องให้กับโมเด็มได้ แต่อย่างไรก็ตามชุดคำสั่งของโมเด็มยี่ห้อและรุ่นต่างๆ ก็จะมีอยู่บางคำสั่งที่เป็นมาตรฐานเดียวกันโดยอ้างอิงมาจากชุดคำสั่งมาตรฐานที่มีอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเด็มของบริษัท Hayes และบริษัท Hayes เองก็กำลังพยายามผลักดันให้ชุดคำสั่ง AT ของตนเป็นที่ยอมรับให้เป็นมาตรฐานจาก ITU

2.2.1.1 การใช้งานชุดคำสั่ง AT

ขั้นตอนการเชื่อมต่อระหว่างโมเด็มต้นทาง และโมเด็มปลายทางจำเป็นจะต้องมีพื้นฐานที่เหมือนกันบางประการ เช่น ความเร็ว รูปแบบของข้อมูล และโปรโตคอลในการรับส่งข้อมูล โดยการเชื่อมต่อโมเด็มในแต่ละครั้ง อย่างน้อยที่สุดผู้ที่ใช้โมเด็มต้นทางจำเป็นต้องทราบว่าจะให้โมเด็มโทรศัพท์ออกด้วยหมายเลขใด ควรจะตั้งเวลาในการรอเสียงตอบจากโมเด็มปลายทางนานเท่าใด ระบบโทรศัพท์ของตนนั้นใช้ระบบ Tone หรือ Pulse และในขณะเดียวกันผู้ที่ใช้โมเด็มปลายทางก็จะต้องกำหนดให้โมเด็มรับทราบด้วยว่าจะให้โมเด็มรับสายโทรเข้าหลังจากที่มีเสียง Ringing กี่ครั้ง และจะต้องรอสัญญาณโต้ตอบนานเท่าไรเป็นต้น ซึ่งการดำเนินงานทั้งหมดนี้จะสามารถควบคุมได้โดยใช้ชุดคำสั่ง AT

2.2.1.2 ประเภทของชุดคำสั่งAT

เราสามารถจะแบ่งชุดคำสั่ง AT ออกเป็น 2 ประเภทใหญ่ๆ คือ ชุดคำสั่งที่ใช้ในการปฏิบัติงาน เช่น ATD(คำสั่งให้หมุนหมายเลขโทรศัพท์) และ ชุดคำสั่งที่ใช้กำหนดค่าหรือเปลี่ยนแปลงค่าต่างๆ เช่น ATS7=90 เป็นการกำหนดค่าให้รีจิสเตอร์ S7 มีค่าเป็น 90 โดยตามปกติแล้วโปรแกรมสื่อสารมักจะตั้งค่าของรีจิสเตอร์ S ต่างๆที่จำเป็นเอาไว้ตั้งแต่ตอนที่ผู้ใช้เรียกโปรแกรมนั้นๆขึ้นมา ซึ่งผู้ใช้จะสามารถควบคุมและเปลี่ยนแปลงค่าของรีจิสเตอร์ S ได้ในภายหลัง

2.2.1.3 ชุดคำสั่งเพิ่มเติม

ชุดคำสั่งรุ่นแรกๆของ Hayes นั้นจะใช้ตัวอักษรตั้งแต่AจนถึงZ เช่น ATAจนถึงATZ ซึ่งก็หมายความว่าคำสั่งจะมีให้ใช้ได้ไม่เกิน26คำสั่งเท่านั้น แต่โมเด็มในปัจจุบันก็ได้มีความสามารถต่างๆที่พิเศษเพิ่มเข้ามา ดังนั้นHayesจึงได้กำหนดชุดคำสั่งเพิ่มเติมขึ้นมาโดยใช้ตัวอักษร&เป็นตัวเข้ามาประกอบ ดังตัวอย่าง AT&F หมายถึงให้โมเด็มเรียกค่า Default Setting ขึ้นมาใช้ ซึ่งโมเด็มของผู้ผลิตรายอื่นๆก็อาจจะเพิ่มเติมชุดคำสั่งที่ใช้ตัวอักษรอื่นๆประกอบเช่น *หรือ% เป็นต้น

2.2.1.4 ชุดคำสั่งATที่สำคัญและใช้ในการทดลอง

- ATD

คำสั่ง ATD จะทำให้โมเด็มยกหูโทรศัพท์ขึ้น (Off-Hook) และหมุนหมายเลขโทรศัพท์ออกด้วย ซึ่งจะมีรูปแบบของคำสั่งดังตัวอย่างเช่น ATD023197777 เป็นการสั่งให้โมเด็มยกหูโทรศัพท์ขึ้นและหมุนไปที่เบอร์โทรศัพท์ 0-23197777 ซึ่งคำสั่ง ATD นี้จะต้องการพารามิเตอร์บางอย่างทุกครั้งที่ใช้คำสั่งนี้เสมอ เช่น “T”,”P”,”W” เป็นต้น ดังแสดงไว้ในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์	ความหมาย
!	วางหูโทรศัพท์ชั่วคราว
,	หยุดรอเป็นเวลา2วินาทีขณะหมุนหมายเลข
;	หลังจากหมุนหมายเลขแล้วจะส่งไปที่สภาวะคำสั่ง
P	หมุนหมายเลขด้วยระบบ Pulse
T	หมุนหมายเลขด้วยระบบ Tone
R	เรียกโหมด Reverse
W	รอฟังเสียงสัญญาณ"สายว่าง"
S=x	เก็บหมายเลขโทรศัพท์ไว้ในพื้นที่ x

รูปที่ 2.13 แสดงพารามิเตอร์ที่ใช้คู่กับคำสั่งในการหมุนหมายเลขโทรศัพท์

พารามิเตอร์"P" หมายถึงให้โมเด็มหมุนหมายเลขโทรศัพท์ด้วยระบบ Pulse ซึ่งเป็นระบบที่ใช้กันในโทรศัพท์แบบหมุน ส่วนพารามิเตอร์"T" หมายถึงให้โมเด็มหมุนหมายเลขโทรศัพท์ออกด้วยระบบแบบ Tone ซึ่งเป็นระบบที่ใช้กันในโทรศัพท์แบบกดปุ่มทั่วไป (บางครั้งอาจเรียก Tone ว่า DTMF:Dual Tone Multi-Frequency) ตัวอย่างเช่น ATDP023197777,ATDT023197777 ซึ่งถ้าไม่ได้กำหนดว่าเป็นแบบใด โมเด็มจะทำการหมุนหมายเลขโทรศัพท์ด้วยวิธีการล่าสุดที่เคยหมุนหมายเลขโทรศัพท์มา เช่นถ้าโมเด็มได้รับคำสั่ง ATDT023197777 มาก่อน หลังจากนั้นโมเด็มได้รับคำสั่งATD023197777 ก็หมายความว่าโมเด็มจะทำการหมุนหมายเลขด้วยระบบ Tone

พารามิเตอร์"W"และ"," มักใช้ในกรณีโมเด็มต่ออยู่กับชุมสายโทรศัพท์ภายใน (PBX:Private Branch Exchange) โดยที่Wมีความหมายว่าให้โมเด็มรอฟังเสียงDial Tone(เสียงที่บอกว่าสายว่าง) ครั้งที่สองก่อนที่จะหมุนหมายเลขโทรศัพท์ออก ดังตัวอย่าง ATDT9W023197777 หมายถึงให้โมเด็มหมุนหมายเลขโทรศัพท์ออกด้วยระบบToneโดยเริ่มหมุนหมายเลข 9(เพื่อที่จะตัดสายออก) จากนั้นก็ให้รอฟังเสียง Dial Tone อีกครั้งหนึ่งแล้วจึงหมุนหมายเลข 023197777 ซึ่งDial Tone ที่โมเด็มได้ยินครั้งแรกก่อนที่จะหมุนหมายเลข 9 จะเป็นเสียงที่ส่งมาจากชุมสายโทรศัพท์ภายใน และเสียงDial Tone ครั้งที่สองหลังจากหมุนหมายเลข 9 นั้นจะเป็นเสียงที่ส่งออกมาจากชุมสายโทรศัพท์โดยตรง

- ATE

เป็นคำสั่งที่ใช้ในการเปิดและปิดการแสดงตัวอักษร(Echo) ที่ผู้ใช้กำลังพิมพ์ออกมาทางจอภาพ ถ้าEcho Offก็หมายความว่าเมื่อกดปุ่มใดๆแล้วผู้ใช้จะไม่สามารถมองเห็นตัวอักษรบนหน้าจอขณะที่กำลังพิมพ์เข้าไป ถ้าOnก็จะเห็นตัวอักษรที่เราพิมพ์ออกทางจอภาพ

ATE0 หรือ ATE : Echo Off

ATE1 : Echo On

- ATH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นคำสั่งที่ทำให้โมเด็มวางหู หรือยกหูโทรศัพท์ โดยมีรูปแบบคือ

ATH0 หรือ ATH : ให้โมเด็มวางหูโทรศัพท์

ATH1 : ให้โมเด็มยกหูโทรศัพท์

ซึ่งคำสั่งATH0หรือATH มักจะใช้หลังจากที่ส่ง Escape Sequence (+++) ให้แก่โมเด็มเพื่อให้โมเด็มกลับเข้าสู่สภาวะคำสั่ง ซึ่งโปรแกรมสื่อสารทั่วๆไปมักจะใช้วิธีนี้เพื่อทำการหยุดการสื่อสารหรือตัดสาย

2.3 เครือข่ายระบบสื่อสารแพ็กเก็ตเสียง

ในปัจจุบันเครือข่ายโทรศัพท์เป็นเครือข่ายได้รับการพัฒนาและใช้งานอย่างต่อเนื่องทำให้เครือข่ายโทรศัพท์ขยายตัวจนในปัจจุบันเครือข่ายโทรศัพท์ขยายตัวจนกลายเป็นเครือข่ายที่เชื่อมโยงกันทั่วโลก ทำให้สามารถติดต่อกันได้อย่างกว้างขวาง แต่เนื่องจากเครือข่ายโทรศัพท์ยังมีต้นทุนสูงในเชิงค่าใช้จ่าย ทำให้ต้องหาวิธีในการลดต้นทุน เพิ่มประสิทธิภาพ โดยเฉพาะวิธีการทางเทคโนโลยีสมัยใหม่หลายอย่างได้เข้ามามีบทบาทที่ทำให้การใช้งานกว้างขวางและยังลดต้นทุนค่าใช้จ่ายให้ถูกลงอีกด้วย

แนวคิดที่จะทำให้ระบบสื่อสารถูกลงคือ การจัดระบบสื่อสารเป็นรูปแพ็กเก็ต เพื่อว่าข้อมูลที่รับส่งกันเป็นกลุ่มก้อนทำให้เกิดการจัดการที่จะบริหารแถบกว้างให้แบ่งกันใช้เพื่อได้ประสิทธิภาพสูงสุด เช่น เครือข่ายอินเตอร์เน็ต

2.3.1 เหตุผลในการใช้แพ็กเก็ตเสียง

แรงขับดันที่ทำให้เทคโนโลยีแพ็กเก็ตเสียงเป็นจริงขึ้นมา ทั้งนี้เพราะเทคโนโลยีอื่นเข้ามามีบทบาทและเกี่ยวข้องอย่างมาก ได้แก่

- ฮาร์ดดิสค์ มีความจุมากขึ้น ขณะที่ราคาลดลง
- ความเร็วของซีพียูที่มีความเร็วเพิ่มขึ้น
- การใช้หน่วยความจำในเครื่องคอมพิวเตอร์มากขึ้น
- ความเร็วของตัวซีดีรอมเพิ่มมากขึ้น

สำหรับในระบบสื่อสารแล้ว การเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล เพื่อให้ประมวลผลข้อมูลและสื่อสารง่าย การประมวลผลสัญญาณใช้ซีพียูพิเศษที่ใช้ในการคำนวณหรือประมวลผลเรียกว่า DSP (Digital Signal Processing) ในปัจจุบันมีการสร้าง DSP ที่ประมวลผลสัญญาณได้เร็วมากสามารถนำมาใช้ในระบบสื่อสารได้ดี เช่น ในโมเด็มสมัยใหม่ที่ทำงานได้ดีมากขึ้น เพราะDSP การทำการสื่อสารในดิจิตอลจึงต้องประมวลผลเพื่อสื่อสารให้ได้ตามต้องการ

นอกจากนี้ DSP ยังทำให้การบีบอัดเสียงได้ดีขึ้นสามารถสร้างการ์ดเสียง (Soundcard) ที่ใช้ในคอมพิวเตอร์ให้สามารถมีการบีบอัดเสียงที่ดีขึ้น

2.3.2 การเข้ารหัสเสียง

โดยธรรมชาติหูของมนุษย์ได้ยินเสียงในรูปแบบสัญญาณที่เป็นแบบอนาล็อก ดังนั้นระบบโทรศัพท์ในยุคแรกจึงใช้วิธีการส่งสัญญาณเสียงแบบอนาล็อกโดยมีอุปกรณ์สวิตชิงเชื่อมโยงวงจรให้สัญญาณเสียงเชื่อมต่อระหว่างต้นทางกับปลายทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งสัญญาณแบบอนาล็อกมีข้อเสียหลายอย่างโดยเฉพาะยากต่อการประมวลผลและจัดการอีกทั้งยังมีสัญญาณอื่นมารบกวนได้ง่ายและขจัดได้ยาก การขยายสัญญาณอนาล็อกทำให้เกิดการขยายสัญญาณรบกวนตามไปด้วยการรับส่งสัญญาณแบบอนาล็อกจึงได้คุณภาพที่ไม่ดีนัก

เมื่อมีการพัฒนาเทคโนโลยีการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลที่ประกอบด้วยค่า 0 กับ 1 มีการเข้ารหัสเพื่อแทนสิ่งต่างๆโดยเฉพาะสัญญาณที่ใช้มีการเข้ารหัสแบบพัลซ์โค้ดมอดูเลชัน หรือ PCM โดยลักษณะของการแปลงสัญญาณPCMคือ มีการค่าจำนวน8000ครั้งต่อวินาทีหรือการสุ่มแต่ละครั้งห่างกัน 125ไมโครวินาที เมื่อสุ่มค่าได้แล้วก็แปลงเป็นรหัสดิจิตอลขนาด8บิต โดยการสุ่มขนาด 125 ไมโครวินาทีนี้เพียงพอที่จะสุ่มสัญญาณอนาล็อกที่มีความถี่ได้ไม่เกิน 4000 เฮิรตซ์ ซึ่งพอเพียงสำหรับความถี่เสียงที่เราใช้งาน

การสุ่มค่า 8000 ครั้งต่อวินาที และแต่ละครั้งแปลงสัญญาณเป็นดิจิตอลขนาด 8 บิตทำให้ PCM ที่ใช้กับสัญญาณเสียงหนึ่งๆมีขนาดความเร็ว 64000 บิตต่อวินาที โดยการแปลงสัญญาณอนาล็อกเป็นดิจิตอลด้วยอัตราสุ่ม 8000 ครั้งต่อวินาทีเป็นไปตามมาตรฐานข้อกำหนดที่ชื่อว่า G.711 ทำให้ขนาดของความกว้างของสัญญาณเสียงที่ใช้จึงมีค่าเท่ากับ 64 กิโลบิตต่อวินาที แต่เพื่อลดขนาดความกว้างในการรับส่งให้น้อยลงเพื่อจะได้ส่งได้ด้วยจำนวนช่องสัญญาณเสียงได้มากขึ้น จึงได้กำหนดมาตรฐานขึ้นมาใหม่ชื่อ ADPCM หรือ Adaptive Differential PCM โดยมีการเข้ารหัสจาก 8 บิตเหลือเพียง 4 บิตทำให้แต่ละการสุ่มเปรียบเทียบความแตกต่างจากครั้งก่อนหรือค่าผลต่างจึงทำให้การสุ่มเท่าเดิมแต่การเข้ารหัสเพียง 4 บิตจึงลดแถบกว้างให้เหลือเพียง 32 กิโลบิตต่อวินาที ทำให้สัญญาณเสียงมีจำนวนช่องในการสื่อสารระบบสายสัญญาณดิจิตอลได้มากขึ้น

ซึ่งการลดขนาดแถบกว้างในการส่งยังทำได้อีกเช่น ถ้าใช้หลักการ ADPCM แต่เก็บค่าเพียง2บิตที่เป็นผลต่างของข้อมูลการสุ่มครั้งก่อนก็จะทำให้การรับส่งใช้แถบกว้างเพียง 16 กิโลบิตต่อวินาที แต่คุณภาพเสียงจะลดลงตามเช่นกัน

2.3.3 เทคนิคในการบีบอัดข้อมูล

ด้วยเทคนิคของการประมวลผลสัญญาณเชิงดิจิตอล ทำให้มีการพัฒนาเทคนิคของการบีบอัดข้อมูลที่มีประสิทธิภาพดียิ่งขึ้นซึ่งอัลกอริธึมในการบีบอัดข้อมูลเสียงที่ได้รับการพัฒนาคือ LPC (Linear Prediction Code) วิธีการบีบอัดนี้มีรูปแบบเพื่อความเหมาะสมกับงานที่ใช้แทนเสียงพูด

การใช้งานทางโทรศัพท์ส่วนใหญ่ใช้แบบ PCM ที่เป็นแบบ 64 กิโลบิตต่อวินาทีทั้งนี้เพราะการประมวลผลสัญญาณแบบนี้ใช้กันมากในงานโทรศัพท์ที่แพร่หลายและมีอุปกรณ์รองรับอยู่มาก แต่เมื่อใช้แบบ LPC ซึ่งสามารถลดให้เหลือเพียงประมาณ 8 กิโลบิตต่อวินาทีได้

มาตรฐานการสร้างรหัสของข้อมูลเสียงจึงมีหลายมาตรฐานโดย ITU (International Telephony Union) เป็นผู้กำหนดโดยจัดอยู่ในข้อแนะนำที่เรียกว่า G – series ซึ่งประกอบด้วยมาตรฐานสำคัญคือ

- **G.711** เป็นมาตรฐานที่ใช้สัญญาณเข้ารหัสแบบ PCM มีขนาด 64000 บิตต่อวินาที ใช้ในกิจการโทรศัพท์ รวมถึงการสวิตช์ใน PBX ทั่วไป

- **G.726** เป็นมาตรฐานการบีบอัดเสียงด้วยวิธีการ ADPCM ทำให้ได้ขนาดสัญญาณลดลงเป็น 40,32,24 และ 16 กิโลบิตต่อวินาที เป็นที่นิยมใช้ในการสร้างแพ็กเก็ตเสียงที่ใช้ในสวิตชิงแบบ PBX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

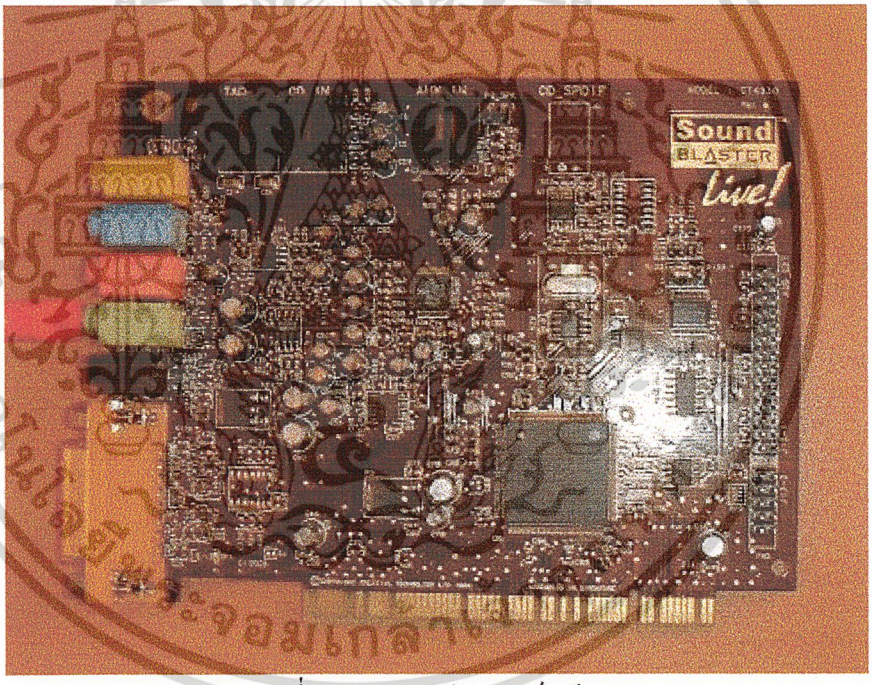
- G.728 เป็นมาตรฐานที่ใช้หลักการของ LPC ที่ชื่อว่า CELP (Code Excited Linear Prediction) สามารถบีบอัดสัญญาณให้เหลือเพียง 16 กิโลบิตต่อวินาที G.728 เป็นมาตรฐานที่ใช้กันในการแปลงสัญญาณเสียงเพื่อส่งกันระหว่าง PBX

- G.729 ใช้ CELP เช่นกัน แต่บีบอัดข้อมูลเสียงให้เหลือเพียงแถบกว้าง 8 กิโลบิตต่อวินาที มีรูปแบบมาตรฐานอยู่สองรูปแบบคุณภาพของเสียงยังคงได้เท่ากับแบบ ADPCM

- G.723.1 เป็นมาตรฐานที่สร้างสายสัญญาณเสียงที่ใช้ในเรื่องมัลติมีเดีย ซึ่งใช้กันในระบบการรับส่งข้อมูลที่ความเร็วไม่สูงมาก สามารถบีบอัดข้อมูลให้เหลือเพียง 5.3 และ 6.3 กิโลบิตต่อวินาที โดยคุณภาพเสียงยังดีอยู่

2.4 ข้อมูลการ์ดเสียง

ต้นตำรับของเสียงสังเคราะห์เกิดจากนายโรเบิร์ต มูก เป็นผู้ริเริ่มนำเอาเสียงสังเคราะห์มาใช้โดยการรวมคีย์บอร์ดเข้ากับวงจรอิเล็กทรอนิกส์ เพื่อให้ได้เสียงที่แตกต่างกัน นายโรเบิร์ต ใช้วงจรออสซิลเลเตอร์เป็นวงจรกรองสัญญาณ และวงจรขยายสำหรับสังเคราะห์เสียง และสามารถใช้ได้จริง ยังสามารถพัฒนาให้สังเคราะห์เสียงดนตรีต่างๆเพิ่มเติม

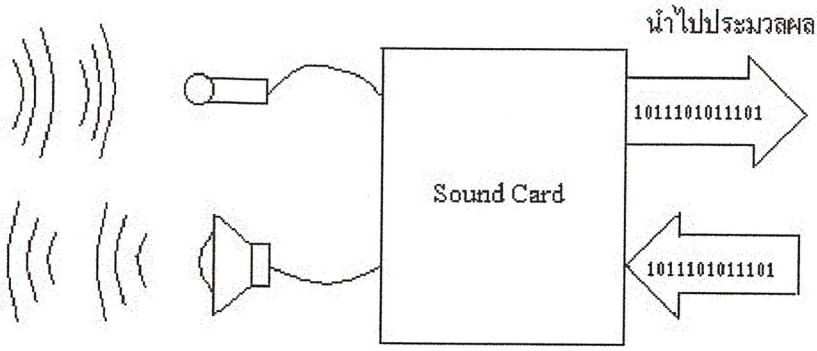


รูปที่ 2.14 แสดงรูปของการ์ดเสียง

คุณสมบัติของเสียงโดยทั่วไปมักเป็นสัญญาณAnalog ที่มีค่าขนาดของสัญญาณต่อเนื่องกันไป โดยสามารถแทนค่าขนาดของสัญญาณด้วยแรงดันไฟฟ้า และความถี่เสียงด้วยความดังมากกว่าสัญญาณที่มีความถี่สูงกว่า แต่ถ้าเป็นสัญญาณดิจิตอล จะเก็บและประมวลผลไว้ในลักษณะของค่า 0 และ 1

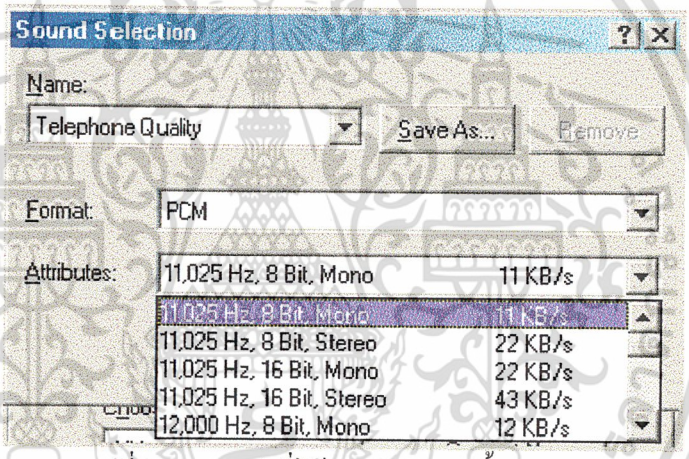
ดังนั้นการเก็บเสียงในคอมพิวเตอร์จึงต้องแปลงสัญญาณเสียงให้เป็นสัญญาณดิจิตอลก่อน เพื่อนำไปเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ และเมื่อต้องการรับฟังเสียงที่บันทึกไว้ก็จะต้องนำข้อมูลดิจิตอลมาแปลงกลับเป็นสัญญาณAnalogเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงกระบวนการแปลงสัญญาณของการ์ดเสียง

กระบวนการแปลงสัญญาณAnalogเป็นสัญญาณดิจิตอล ในการ์ดเสียงจะเป็นจะเป็นหน้าที่ของชิพ ADC โดยจะทำหน้าที่วัดขนาดสัญญาณที่คลื่นเสียงตามระยะเวลาที่คงที่ เรียกว่า อัตราแซมปลิ้ง (Sampling Rate) และเรียกเวลาในการแปลงขนาดของสัญญาณเป็นข้อมูลตัวเลข เพื่อนำไปประมวลผลของชิพ ADC ว่าคาบเวลาของการแซมปลิ้ง (Sampling Time) ค่าตัวเลขที่ได้เป็นเลขฐานสิบ ส่วนเอาท์พุทจริงจากชิพ ADCเป็นเลขฐานสอง



รูปที่ 2.16 แสดงค่าที่ได้จากการแซมปลิ้งสัญญาณ

ดังนั้นจึงต้องมีกระบวนการแปลงเลขฐานสิบเป็นเลขฐานสอง ค่าที่เก็บจะเริ่มจากเลขศูนย์ตามด้วยเลขฐานสองต่างๆ กรรมวิธีนี้เป็นกระบวนการมอดูเลชั่นแบบหนึ่งเรียกว่า พัลซ์โค้ดมอดูเลชั่น(Pulse Code Modulation:PCM)

โดยการสังเคราะห์เสียงดิจิตอลชนิดเอฟเอ็ม สามารถเลียนเสียงต่างๆได้มากมายแต่ไม่ทั้งหมด เสียงที่มีฮาร์โมนิคเปลี่ยนแปลงอย่างรวดเร็วจะไม่สามารถทำการสังเคราะห์เสียงได้ แต่จะบันทึกเสียงเหล่านั้นไว้แล้วแปลงเป็นข้อมูลดิจิตอลจากนั้นมาเล่นกลับ ภายหลังจากการสังเคราะห์เสียงต่างๆด้วยการบันทึกลงซีดี ทำได้โดยนำเสียงดนตรีแต่ละชิ้นทำการแซมปลิ้งเสียงลงแผ่นซีดี 16 บิต และเก็บไว้ด้วยความเร็ว 44.1 KHz โดยการปรับเปลี่ยนอัตราการแซมปลิ้งทำให้พิตซ์เปลี่ยนและสร้างโน้ตเสียงใหม่ๆได้อีกมากมาย

โดยคุณสมบัติของการ์ดเสียงมี 5 ข้อคือ

1. สามารถเล่นไป – กลับโดยใช้ข้อมูลที่เก็บไว้ในรูป pulse code modulation

2. สามารถสังเคราะห์เสียงได้ 128 เสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนมิกเซอร์ (Mixer) เสียง จะทำหน้าที่ควบคุมแหล่งจ่ายเสียง และระดับสัญญาณเสียงที่ลำโพง ในบางการ์ดอาจมีวงจรควบคุมเสียงทุ้มและเสียงแหลมเพิ่มเติมด้วย

4. การบันทึกเสียงจะบันทึกในรูปแบบของ pulse code modulation อาจบันทึกเป็นแบบ stereo หรือ mono และสามารถควบคุมอัตราการแซมปลิ้งที่เหมาะสมกับความคมชัดของเสียงได้

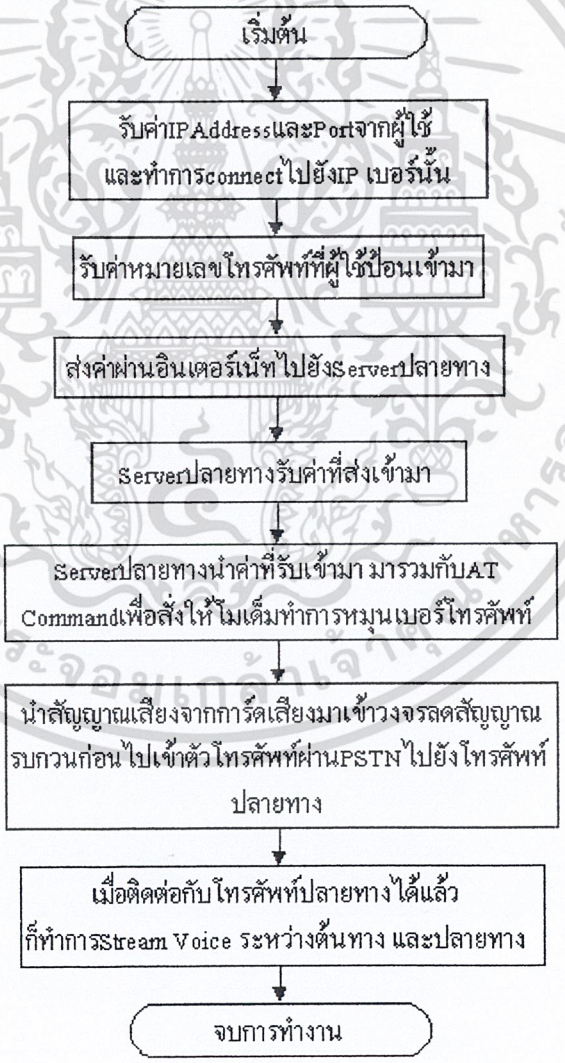
5. ต้องมี input และ output เพื่อให้สามารถใช้การ์ดเสียงกับเครื่องดนตรีมีดี้ภายนอกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณ และการสร้าง

สำหรับโครงงานระบบโทรศัพท์ทางไกลผ่านเครือข่ายอินเตอร์เน็ตนี้ เป็นการเขียนโปรแกรมในส่วนของซอฟต์แวร์เพื่อทำการติดต่อคอมพิวเตอร์สองเครื่องโดยเครื่องหนึ่งเป็นฝั่งไคลเอ็นต์ ส่วนอีกเครื่องเป็นฝั่งเซิร์ฟเวอร์ โดยทางฝั่งไคลเอ็นต์จะติดต่อไปยังฝั่งเซิร์ฟเวอร์โดยใช้เบอร์ไอพี แอดเดรสและเบอร์พอร์ตและส่งค่าเบอร์โทรศัพท์ที่ต้องการจะติดต่อไปยังทางฝั่งเซิร์ฟเวอร์เพื่อไปสั่งให้ตัวโมเด็มทางฝั่งเซิร์ฟเวอร์หมุนเบอร์ตามที่เราต้องการ โดยทำงานร่วมกับคำสั่ง ATคอมมานด์ด้วย

นอกจากนี้ทางกลุ่มได้ทำการเขียนโปรแกรมจัดการเบอร์โทรศัพท์ที่ต้องการจะติดต่อให้เป็นระเบียบสามารถแก้ไข เปลี่ยนแปลง เพิ่ม ลบเบอร์โทรศัพท์ได้ง่ายโดยรองรับเบอร์โทรศัพท์ที่ต้องการจะบันทึกได้จำนวน10เบอร์และยังสามารถสั่งให้ทางฝั่งเซิร์ฟเวอร์หมุนหมายเลขที่ต้องการ โดยการclick ที่ชื่อคนๆนั้นในโปรแกรมจัดการได้ทันที



รูปที่ 3.1 แสดงโฟลว์ชาร์ทการทำงานทั่วไปของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งหลักการทำงานโดยทั่วไปของโปรแกรมนี้คือ จะต้องทำการตั้งค่าของไอพีแอดเดรส และ ค่าพอร์ตของเครื่องเซิร์ฟเวอร์ที่จะติดต่อด้วยในตัวโปรแกรม เมื่อติดต่อได้แล้วโปรแกรมทางด้านผู้เรียก (หรือทางฝั่งไคลเอ็นต์) ก็จะทำการส่งเบอร์โทรศัพท์ของผู้ถูกเรียกไปยังโปรแกรมด้านผู้ถูกเรียก (หรือทางฝั่งเซิร์ฟเวอร์) ผ่านทางอินเตอร์เน็ตโดย ส่งเป็นสตริง (String) ของข้อมูล เมื่อโปรแกรมทางด้านผู้ถูกเรียก (หรือทางฝั่งเซิร์ฟเวอร์) รับสตริงข้อมูลเข้ามาแล้วก็จะทำการรวมเบอร์โทรศัพท์ที่รับเข้ามาเข้ากับคำสั่ง เอทีคอมมานด์ (AT Command) เพื่อไปสั่งให้โมเด็มทำการหมุนหมายเลขโทรศัพท์ตามที่ต้องการได้ เมื่อโมเด็มทำการติดต่อกับปลายทางได้แล้วก็จะเริ่มต้นส่งสตรีมเสียงทั้งด้านต้นทางหรือไคลเอ็นต์และปลายทางหรือเซิร์ฟเวอร์ ดังจะมีลักษณะขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.1

3.1 ส่วนของโปรแกรมทางด้านผู้เรียก (ทางฝั่งไคลเอ็นต์)

สำหรับโปรแกรมในส่วนนี้จะประกอบด้วย ไฟล์โปรเจก 1 โปรเจกคือ Client.vbp โดยภายในจะประกอบด้วยฟอร์ม (Form) คือ Form1.frm,Form2.frm,Form3.frm,frmClient.frm และ frmIPPORT.frm โดยการติดต่อกันเพื่อรับส่งแพ็กเก็ตเสียงทางด้านไคลเอ็นต์จะใช้ตัว Active X Control ที่ชื่อว่า AudioShareClient.ocx และ โปรแกรมย่อยที่มีลักษณะเป็นโมดูล คือ Module1.bas

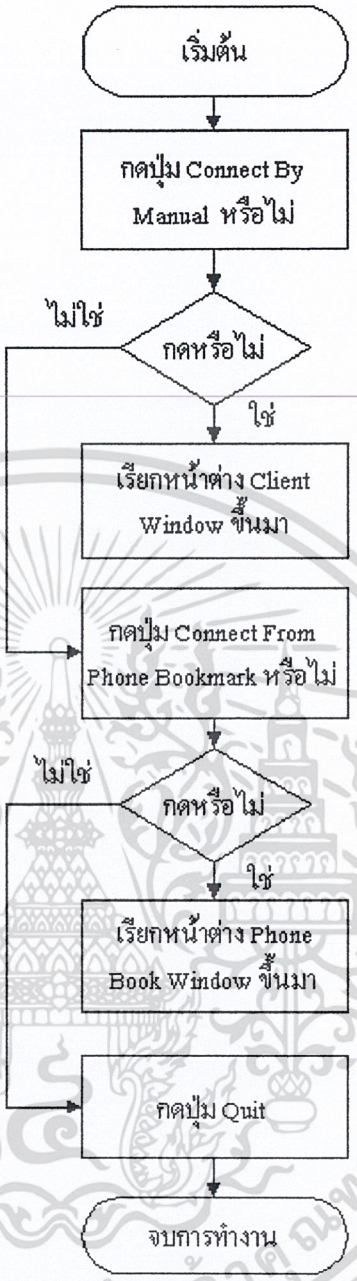
โดยโปรแกรมทางด้านนี้จะประกอบด้วยโปรแกรมย่อยๆดังนี้

- โปรแกรมหลัก (Main Window)

ซึ่งลักษณะการทำงานคือ จะเป็นการให้ผู้ใช้งานเลือกว่าจะเรียกใช้โปรแกรม Connect By Manual หรือ Connect Form Phone Bookmark หรือ ออกจากโปรแกรม

ซึ่งจะมีขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



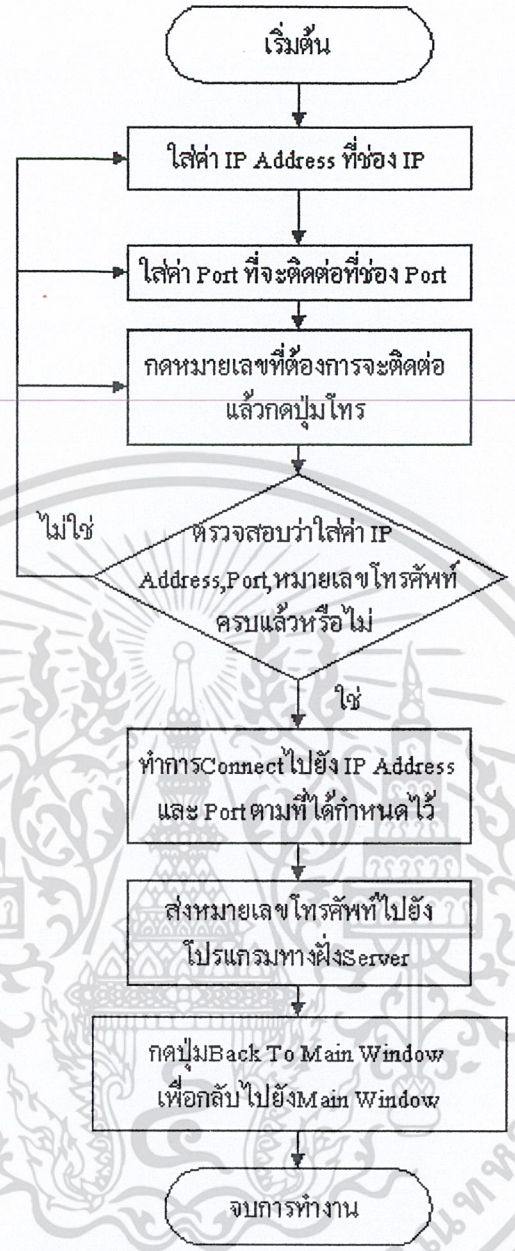
รูปที่ 3.2 แสดง โฟลว์ชาร์ทการทำงานของ โปรแกรม Main Window

- โปรแกรม **Connect By Manual**

ซึ่งลักษณะการทำงานคือ จะเป็นการให้ผู้ใช้งานใส่ค่า IP Address , Port และหมายเลขที่ต้องการจะติดต่อ แล้วติดต่อไปยัง IP Address , Port ปลายทาง และส่งเบอร์โทรศัพท์ไปหมุนโมเด็มทางฝั่งปลายทาง

ซึ่งจะมีขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



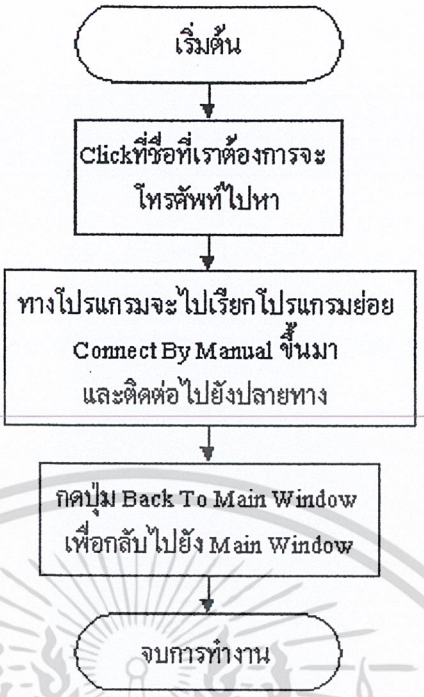
รูปที่ 3.3 แสดง โฟลว์ชาร์ทการทำงานของโปรแกรม Connect By Manual

- โปรแกรม **Connect From Phone Bookmark**

ซึ่งลักษณะการทำงานคือ จะเป็นการจัดการกับเบอร์โทรศัพท์ที่บันทึกเอาให้ให้สามารถใช้งานโทรออกได้ง่ายเพียงแค่ เปิดโปรแกรมแล้ว click ที่ชื่อที่เราต้องการจะติดต่อ

ซึ่งจะมีขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

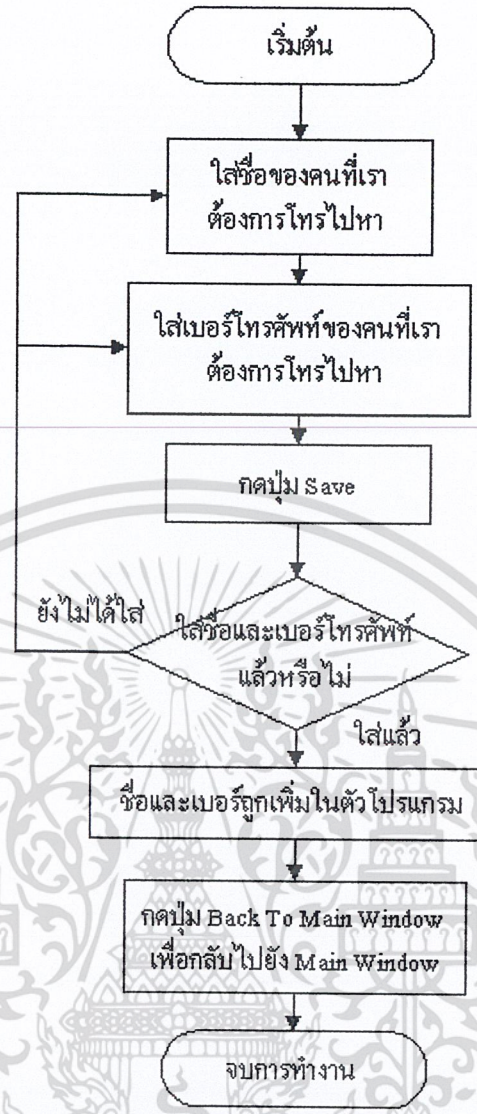


รูปที่ 3.4 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect From Phone Bookmark เมื่อทำการ โทรศัพท์ออกตามชื่อที่เราต้องการ

นอกจากนี้เรายังสามารถเพิ่มชื่อและหมายเลข โทรศัพท์ลงไปใน โปรแกรม Phone Bookmark ได้จำนวน 10 ชื่อ ซึ่งทำให้ง่ายต่อการที่จะ โทรศัพท์ออกไปหาคนๆนั้น

ซึ่งจะมีขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.5

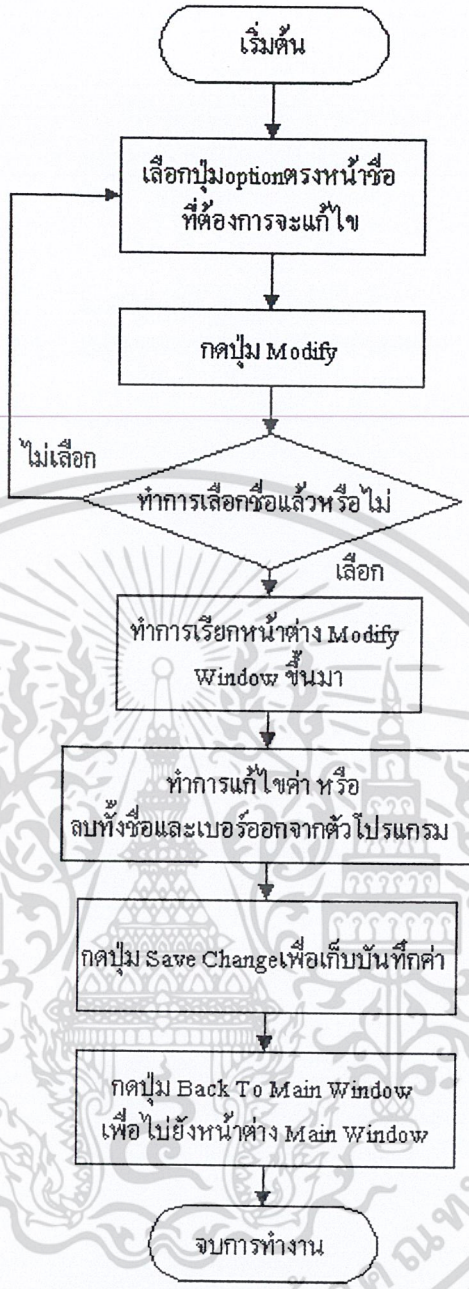
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect From Phone Bookmark เมื่อทำการบันทึกชื่อและเบอร์โทรศัพท์ลงไปในตัวโปรแกรม

นอกจากนี้แล้วยังสามารถที่จะแก้ไข หรือ ลบ ชื่อและเบอร์โทรศัพท์ออกจากตัวโปรแกรมได้ด้วย ซึ่งจะมีขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรม Connect From Phone Bookmark เมื่อทำการแก้ไขชื่อและเบอร์โทรศัพท์ที่ได้บันทึกไว้

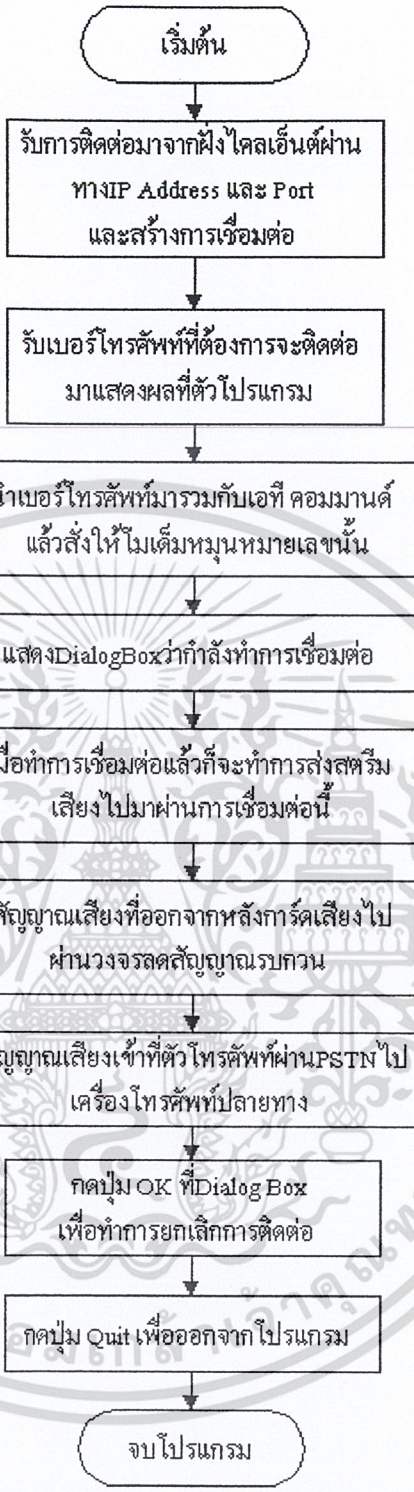
3.2 ส่วนของโปรแกรมทางด้านผู้ถูกเรียก (ทางฝั่งเซิร์ฟเวอร์)

สำหรับโปรแกรมในส่วนนี้จะประกอบด้วย ไฟล์โปรเจก 1 โปรเจกคือ Server.vbp โดยภายในจะประกอบด้วยฟอร์ม (Form) 1 อัน คือ frmServer.frm โดยการติดต่อกันเพื่อรับส่งแพ็กเก็ตเสียงทางด้านเซิร์ฟเวอร์จะใช้ตัว Active X Control ที่ชื่อว่า AudioShareServer.ocx

ซึ่งลักษณะการทำงานคือ รับการติดต่อมาจากฝั่งไคลเอ็นต์เพื่อสร้างเส้นทางเชื่อมการติดต่อ รับเบอร์โทรศัพท์แล้วมารวมกับเอทีคอมมานด์ แล้วไปสั่งให้โมเด็มหมุนหมายเลขนี้แล้วพูดคุยหลังจากการเชื่อมต่อเรียบร้อยแล้ว

ซึ่งจะมีขั้นตอนการทำงาน ตาม โฟลว์ชาร์ท ดังรูปที่ 3.7

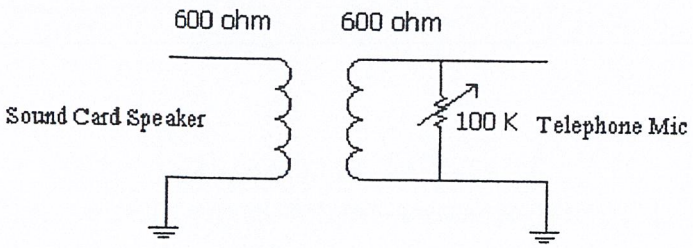
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงโฟลว์ชาร์ทการทำงานของโปรแกรมทางฝั่งเซิร์ฟเวอร์

แต่เนื่องจากเมื่อนำสัญญาณเสียงที่ได้จากช่องหลังการ์ดเสียงไปเข้าเป็นสัญญาณเสียงในสายโทรศัพท์พบว่าได้เกิดสัญญาณรบกวนขึ้นมาก ทางกลุ่มจึงได้ทำการแก้ไขโดยการต่อวงจรเป็นวงจรไอโซเลเตอร์ หรือ แมตช์ชิ่ง ทำหน้าที่ในการแยกกราวด์ของการ์ดเสียงและของระบบโทรศัพท์ให้แยกออกจากกัน ดังรูปที่ 3.8 เพื่อลดสัญญาณรบกวนที่เกิดในสายสัญญาณดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงวงจรลดสัญญาณรบกวนหรือวงจรแมตช์ชิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

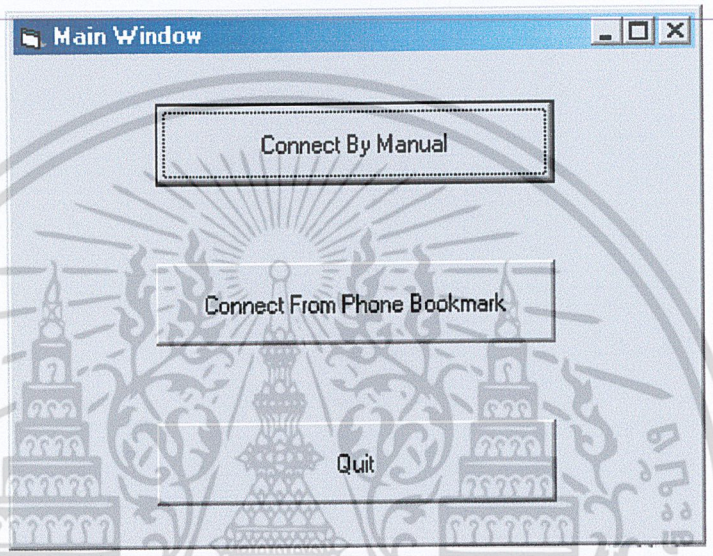
บทที่ 4

การทดลองและผลการทดลอง

ทางกลุ่มได้ทำการพัฒนาโปรแกรมขึ้นมาเพื่อใช้กับโครงงานนี้ โดยใช้ภาษาเบสิก ใช้โปรแกรม Microsoft Visual Basic 6.0 เป็นตัวเขียนโปรแกรม ซึ่งสามารถแบ่งโปรแกรมออกเป็น 2 ส่วน คือ ส่วนทางฝั่งผู้เรียกหรือฝั่งไคลเอนต์ และ ส่วนทางฝั่งผู้ถูกเรียกหรือฝั่งเซิร์ฟเวอร์

4.1 โปรแกรมทางฝั่งผู้เรียกหรือฝั่งไคลเอนต์

เมื่อเราทำการเรียกโปรแกรมทางฝั่งไคลเอ็นต์ขึ้นมา จะปรากฏหน้าต่างหลักขึ้นมาดังรูป 4.1



รูปที่ 4.1 แสดงโปรแกรมหน้าต่างหลักหรือMain Window

โดยจะให้ผู้ใช้เลือกว่าจะติดต่อไปยังปลายทางโดยให้ผู้ใช้กรอกเบอร์เอง หรือ จะติดต่อไปยังปลายทางโดยเลือกจากชื่อ และเบอร์โทรศัพท์ที่เก็บอยู่ในPhone Bookmark ซึ่งสามารถจัดการแก้ไขหรือเพิ่มชื่อและเบอร์โทรศัพท์ในตัวโปรแกรมนี้ด้วย และกดปุ่ม Quit เมื่อต้องการออกจากโปรแกรม Main Window

เมื่อผู้ใช้ทำการ Click เลือกที่ปุ่ม Connect By Manual ก็จะทำการไปเรียกโปรแกรม Client Window ขึ้นมาดังรูป 4.2

โดยให้ผู้ใช้ใส่เบอร์ IP Address , Port ,ชื่อ และกดเบอร์โทรศัพท์ที่ต้องการจะติดต่อในช่องต่างๆ ตามรูปโปรแกรม และทำการกดที่ปุ่มรูปโทรศัพท์สีเขียวเพื่อทำการติดต่อไปยังปลายทางตาม IP Address และ Port ตามที่กรอกไว้เพื่อส่งค่าเบอร์โทรศัพท์ไปยังปลายทางนั้นตามรูป 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงโปรแกรม Client Window เมื่อเริ่มต้นโปรแกรม



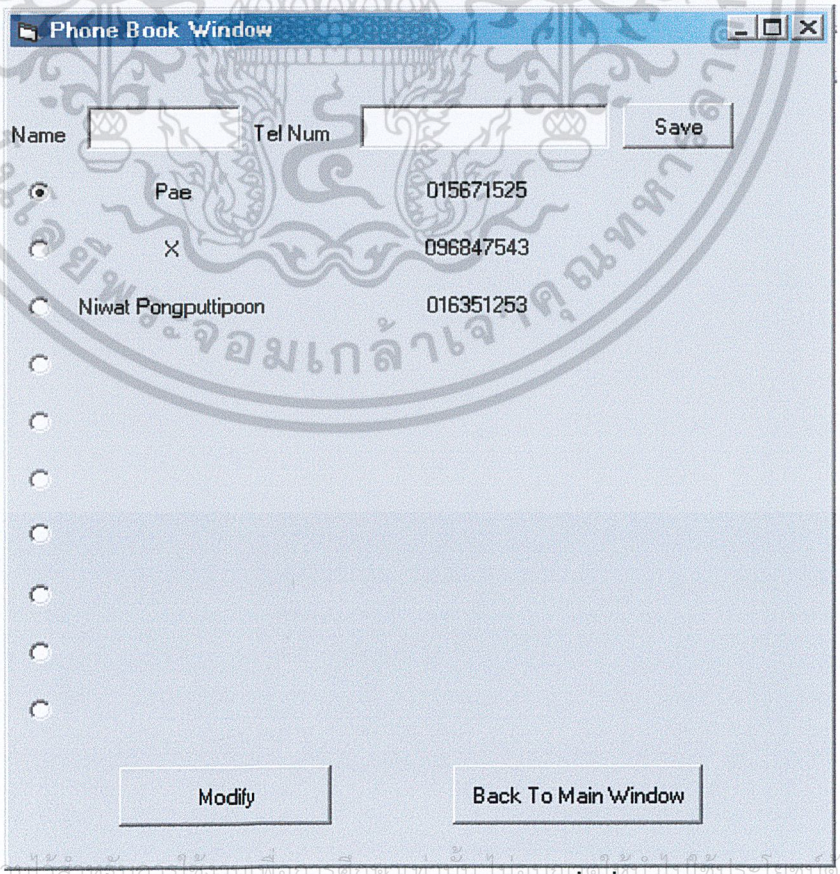
รูปที่ 4.3 แสดงโปรแกรม Client Window เมื่อผู้ใช้ใส่ค่า IP Address , Port และเบอร์โทรศัพท์พร้อมที่จะทำการติดต่อแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงโปรแกรม Client Window เมื่อผู้ใช้ทำการกดปุ่มรูปโทรศัพท์สีเขียว เพื่อทำการConnect

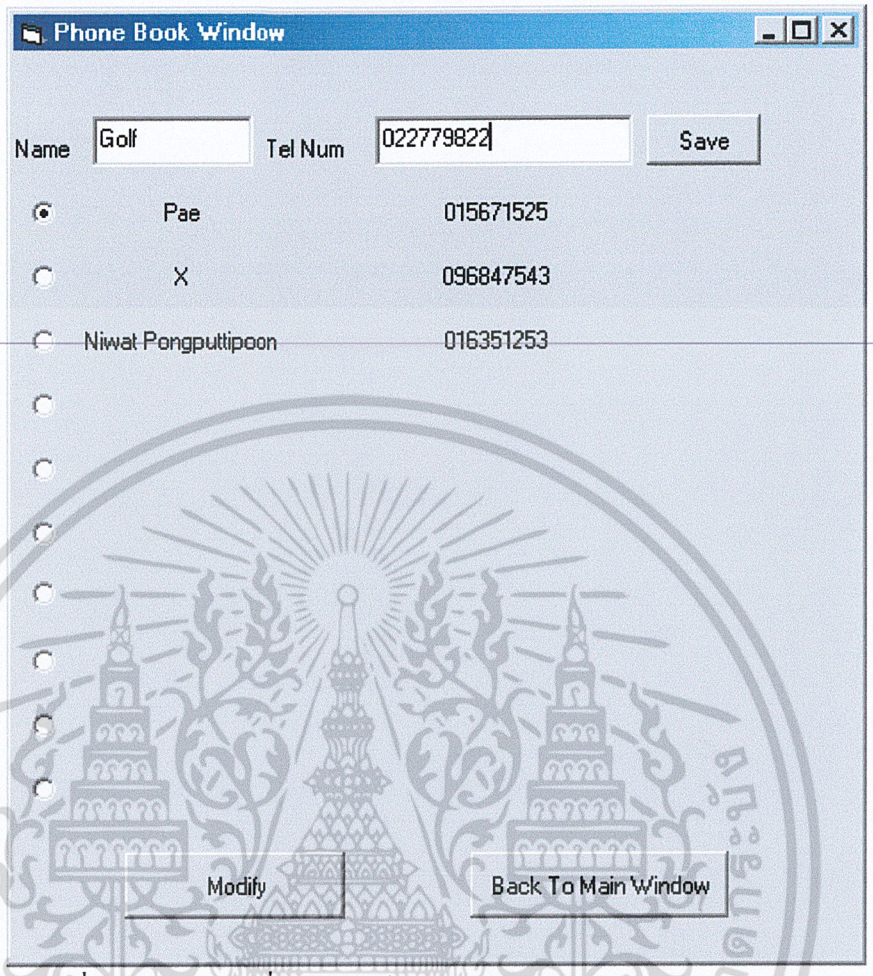
เมื่อผู้ใช้ทำการ Click เลือกที่ปุ่ม Connect From Phone Bookmark ก็จะทำการไปเรียกโปรแกรม Client Window ขึ้นมาดังรูป 4.5



รูปที่ 4.5 แสดงโปรแกรม Phone Book Window เมื่อเริ่มต้นโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

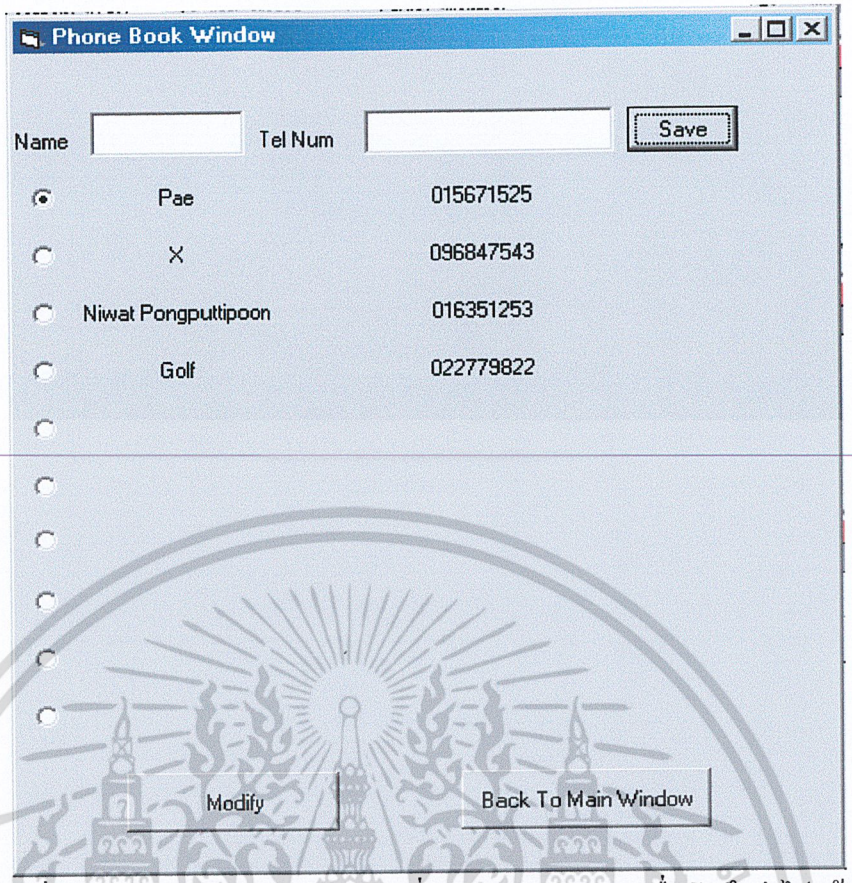
โดยให้ผู้ใช้ใส่ชื่อและเบอร์โทรที่ต้องการจะเก็บลงไปในตัวโปรแกรมแล้วกดปุ่ม Saveแสดงดังรูป 4.6



รูปที่ 4.6 แสดงการใส่ชื่อและเบอร์โทรลงในโปรแกรมPhone Book Window

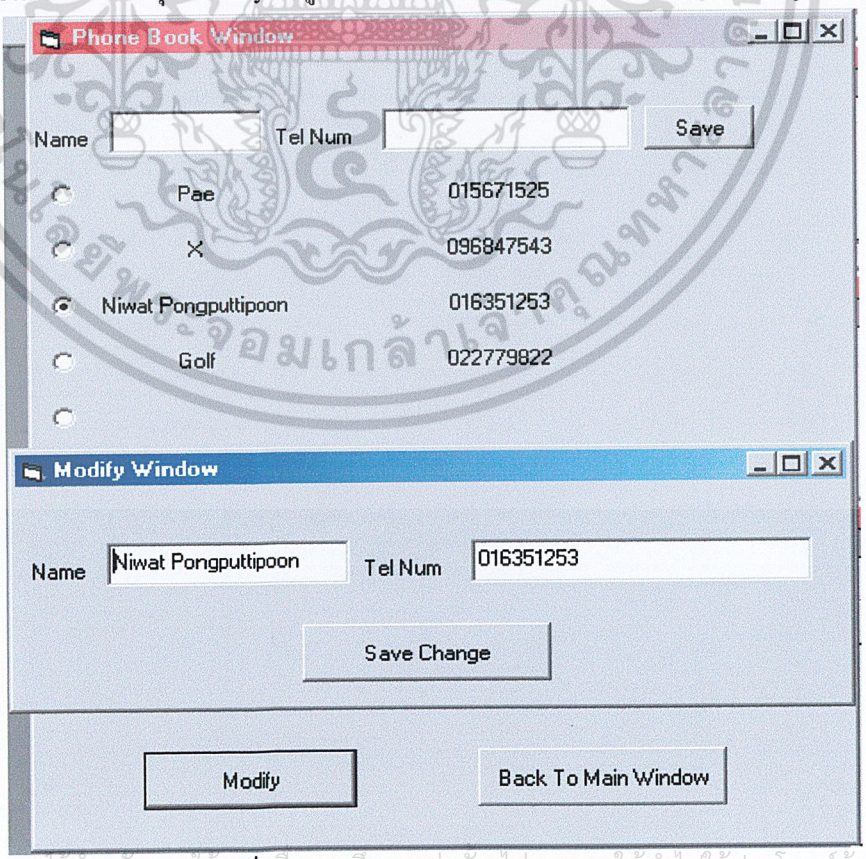
เมื่อทำการกดปุ่ม Save แล้วค่าที่ใส่จะถูกเก็บอยู่ในตัวโปรแกรมและแสดงผลให้เห็นด้านล่างเรียงตามลำดับลงมาโดยสามารถบันทึกชื่อและหมายเลขโทรศัพท์ได้จำนวน 10ชื่อ โดยข้อมูลทั้งหมดจะถูกเก็บไว้ตลอดไปไม่ว่าจะปิดตัวโปรแกรมไปแล้วก็ตาม แสดงดังรูป 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงโปรแกรม Phone Book เมื่อทำการกดปุ่ม Save เพื่อบันทึกค่าไปแล้ว

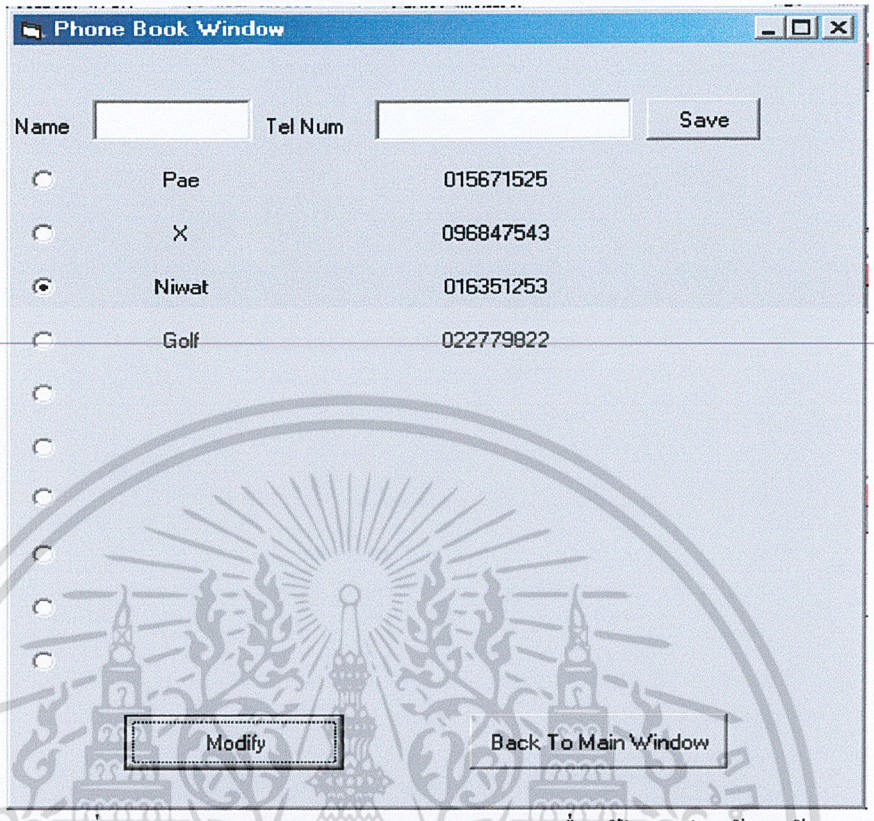
นอกจากนี้เรายังสามารถแก้ไขค่า และลบค่าที่ได้บันทึกไปแล้วได้โดยการเลือกที่ช่องข้างหน้าชื่อที่ต้องการจะแก้ไขแล้วกดปุ่ม Modify ที่อยู่ด้านล่างโปรแกรมย่อย Modify Window ก็จะถูกเรียกขึ้นมา



รูปที่ 4.8 แสดงโปรแกรม Modify Window

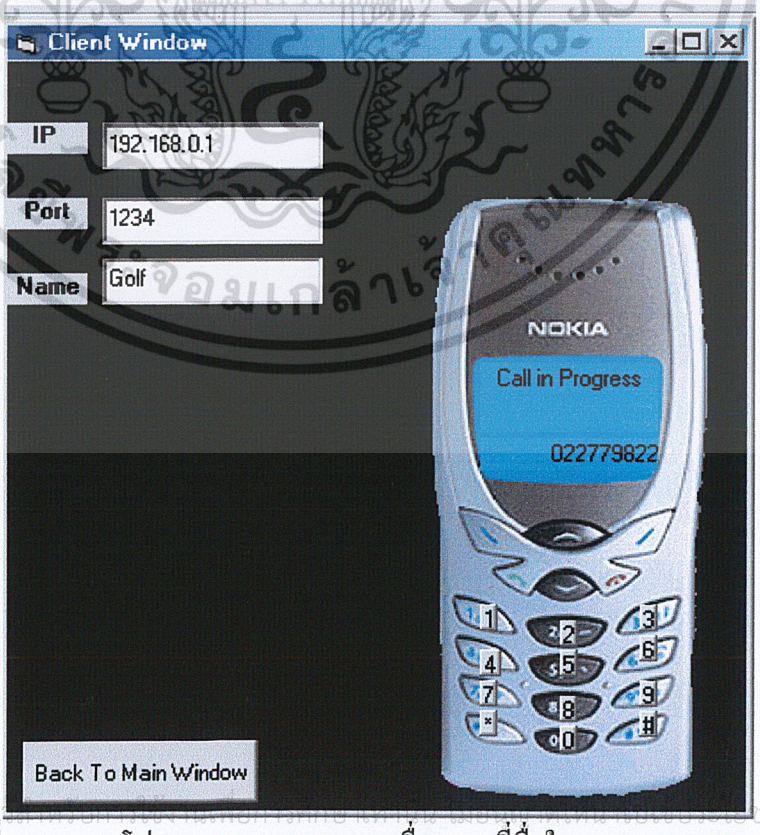
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการแก้ไขค่า หรือลบค่าเรียบร้อยแล้วให้ทำการกดปุ่ม Save Change เพื่อทำการเก็บบันทึกค่า ก็จะกลับมายังโปรแกรม Phone Book Window ดังรูป



รูปที่ 4.9 แสดงโปรแกรม Phone Book Window เมื่อแก้ไขค่าเรียบร้อยแล้ว

โดยถ้าเราต้องการโทรศัพท์หาใครในโปรแกรม Phone Book Window นี้ก็ทำการ click ไปยังชื่อที่เราต้องการ ตัวโปรแกรมก็จะไปเรียกโปรแกรม Client Window ขึ้นมาและทำการติดต่อทันทีดังรูป

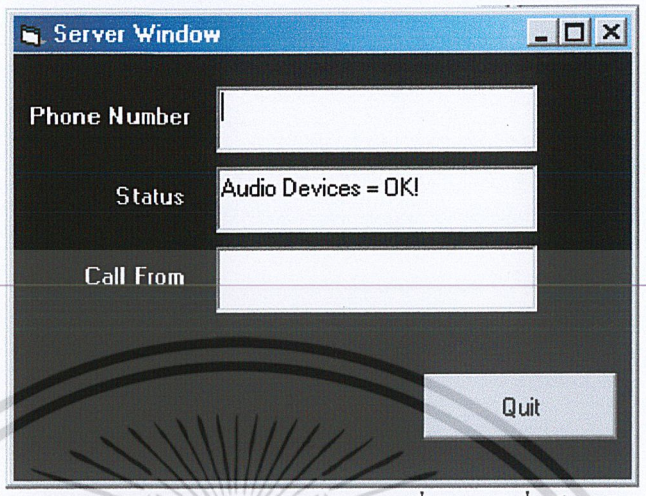


รูปที่ 4.10 แสดงโปรแกรมClient Window เมื่อClick ที่ชื่อในPhone Book Window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โปรแกรมทางฝั่งผู้ถูกเรียกเรียกหรือฝั่งเซิร์ฟเวอร์

เมื่อเราทำการเรียกโปรแกรมทางฝั่งเซิร์ฟเวอร์ขึ้นมา จะปรากฏหน้าต่างขึ้นมาดังรูป 4.11



รูปที่ 4.11 แสดงโปรแกรม Server Window เมื่อทำการเริ่มต้นโปรแกรม

เมื่อโปรแกรมทางฝั่งไคลเอ็นต์ทำการติดต่อมา โปรแกรมทางฝั่งเซิร์ฟเวอร์จะทำการตอบรับการติดต่อดังกล่าวดังรูป



รูปที่ 4.12 แสดงโปรแกรม Server Window เมื่อทำการรับการติดต่อจากไคลเอ็นต์ และทำการสั่งโมเด็มให้หมุนไปยังหมายเลขโทรศัพท์ปลายทาง

ซึ่งเมื่อทางกลุ่มได้ทำการทดลองปฏิบัติจริงๆ พบว่าสามารถทำการติดต่อไปยังฝั่งเซิร์ฟเวอร์ได้ และทำการหมุนโมเด็มไปยังหมายเลขโทรศัพท์ปลายทางได้ แต่ไม่สามารถทำการพูดคุยกับทางปลายทางได้ เนื่องจากจะเกิดสัญญาณรบกวนขึ้นมากขณะทำการสนทนา ทางกลุ่มจึงได้แก้ไขโดยการต่อวงจรเพื่อลดสัญญาณรบกวนหรือวงจรไอโซเลเตอร์หรือวงจรแมตช์ชิ่ง ทำหน้าที่ในการแยกกราวด์ของการ์ดเสียงและของระบบโทรศัพท์ให้แยกออกจากกัน เนื่องจากกราวด์ของการ์ดเสียงและของโทรศัพท์มีค่าไม่เท่ากัน ซึ่งผลที่ได้พบว่าสัญญาณรบกวนได้หายไปเกือบหมดทำให้สามารถทำการพูดคุยกันรู้เรื่องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้จากการทดลองยังพบว่า เมื่อทำการติดต่อได้สำเร็จและทำการสนทนากัน จะสามารถทำการสนทนาได้ประมาณ 4 นาที 10วินาที ถึง 4นาที 15วินาที จากนั้นตัวโมเด็มจะทำการตัดสายลงทำให้ไม่สามารถพูดคุยสนทนากันต่อได้ และระหว่างการสนทนาไปมาจะเกิดการดีเลย์ของสัญญาณเสียงก่อนที่แพ็กเก็ตเสียงจะไปถึงปลายทาง จึงทำการวัดเวลาที่ดีเลย์ได้ค่าดังรูป

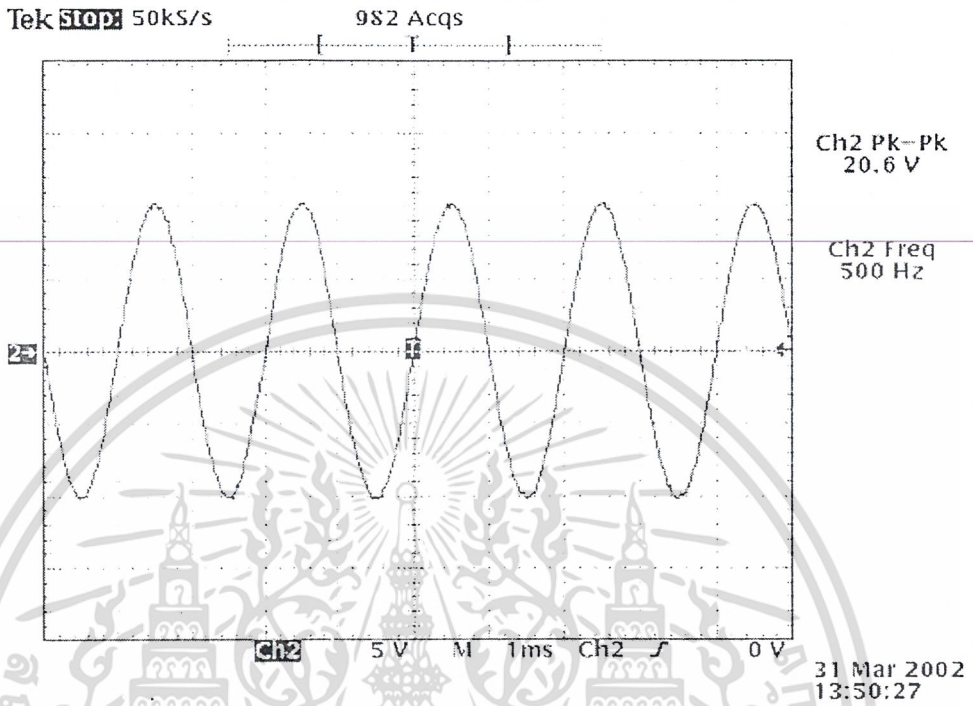
การวัดครั้งที่	เวลาที่ดีเลย์(วินาที)
1	2.2
2	2.1
3	2.2
4	2.2
5	2.3
6	2
7	2.1
8	2.1
9	2.2
10	2.1

รูปที่ 4.13 แสดงเวลาที่ดีเลย์ในระหว่างสนทนา

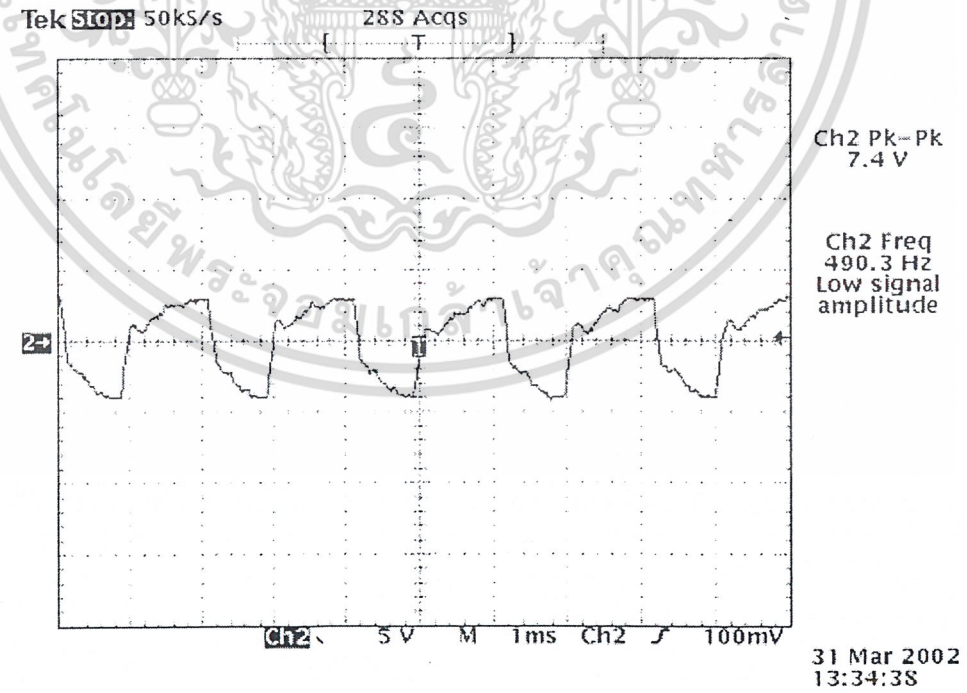
เมื่อทำการหาค่าเฉลี่ยเวลาดีเลย์จะได้ 2.15 วินาที ซึ่งสามารถทำให้สนทนาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองการวัดสมรรถนะของระบบ โดยทางกลุ่มจะทำการป้อนสัญญาณซายน์ผ่านตัวระบบโดยความถี่ที่ป้อนนั้นจะอยู่ในช่วงความถี่ที่ใช้พูดในระบบโทรศัพท์ ซึ่งความถี่ที่ทำการทดลองป้อนคือ 500 Hz , 2.5 KHz และ 3 KHz ที่แอมปลิจูดขนาด 10Vp-p เท่ากันหมด ซึ่งได้ผลการทดลองดังนี้

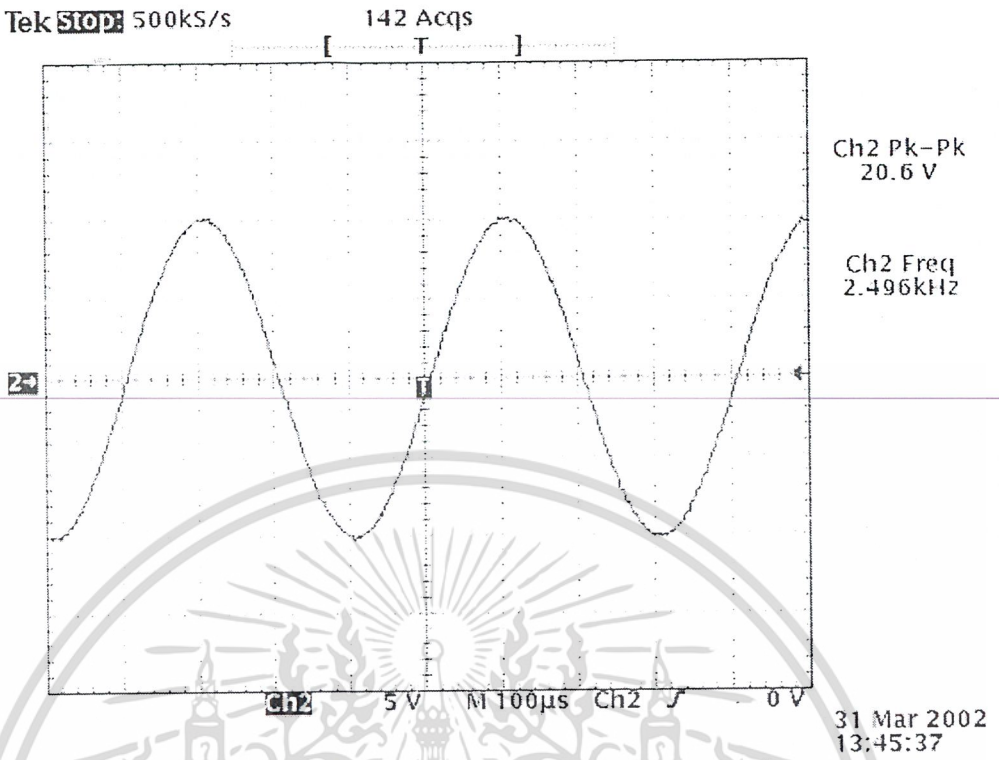


รูปที่ 4.14 แสดงสัญญาณอินพุทที่ป้อนให้กับระบบโดยเป็นสัญญาณซายน์ที่ความถี่ 500Hz แอมปลิจูด 10Vp-p

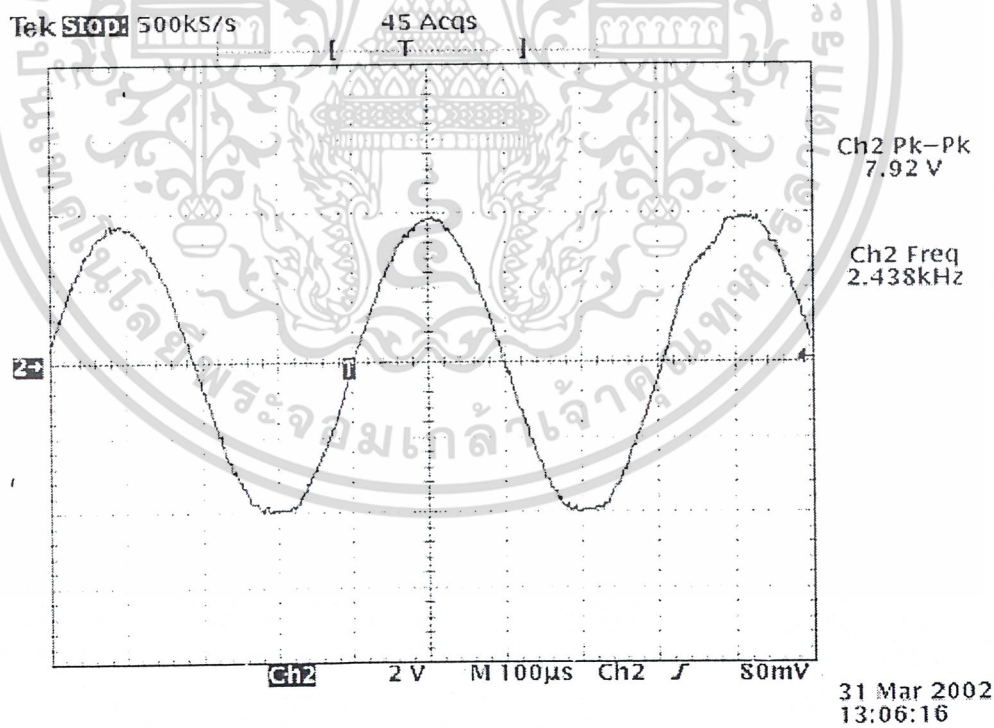


รูปที่ 4.15 แสดงสัญญาณเอาท์พุทของระบบเมื่อป้อนอินพุทสัญญาณซายน์ที่ความถี่ 500Hz แอมปลิจูด 10 Vp-p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

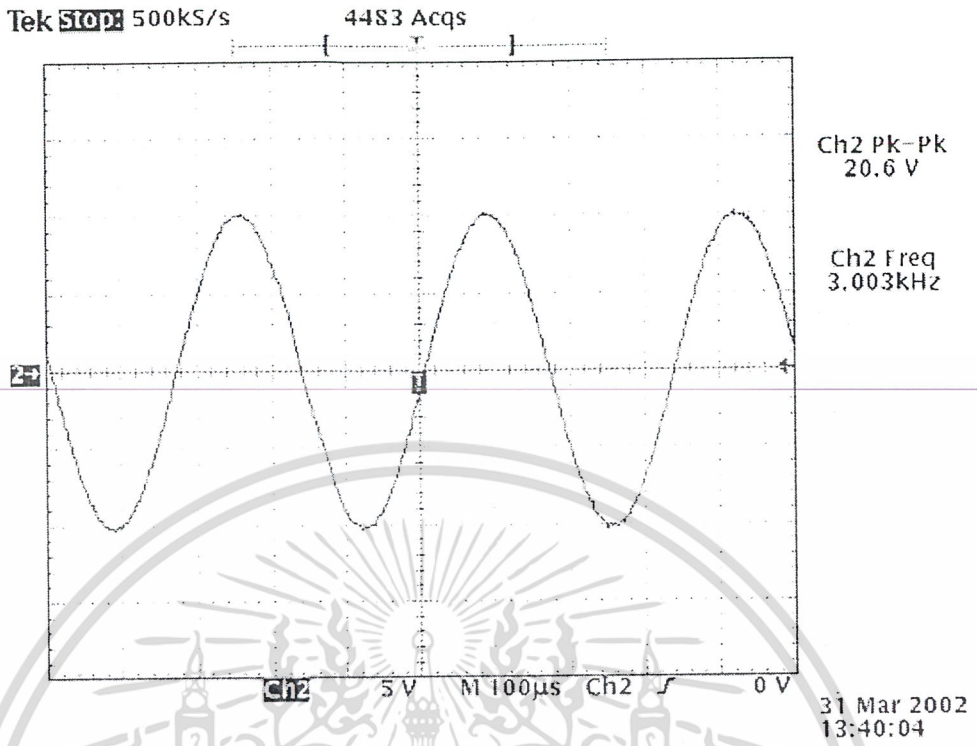


รูปที่ 4.16 แสดงสัญญาณอินพุทที่ป้อนให้กับระบบโดยเป็นสัญญาณซายน์ที่ความถี่ 2.5KHz แอมปลิจูด 10 Vp-p

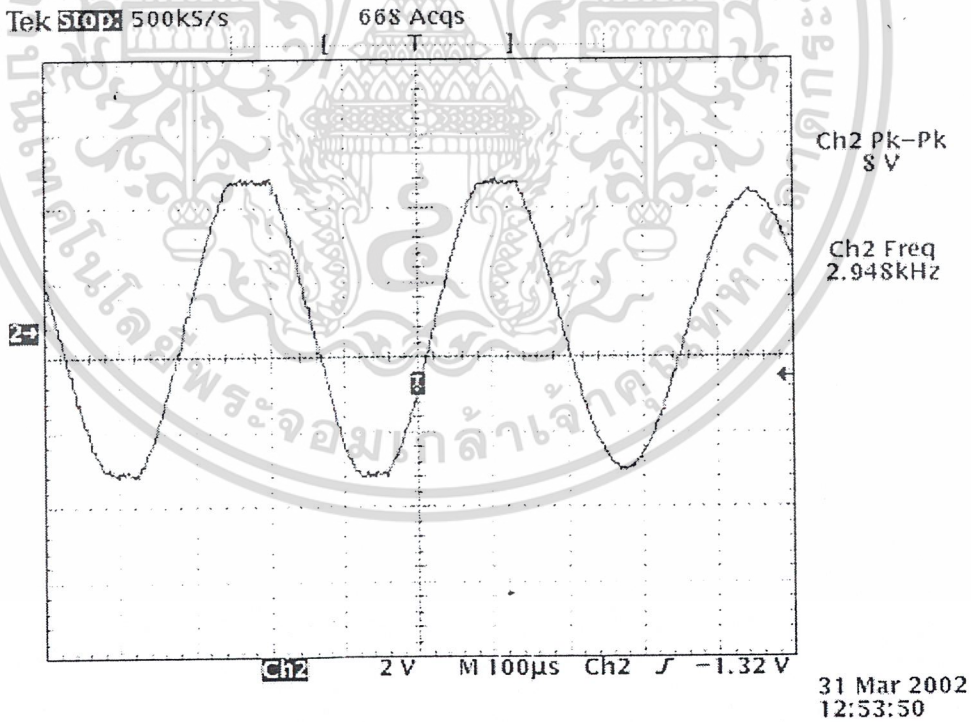


รูปที่ 4.17 แสดงสัญญาณเอาท์พุทของระบบเมื่อป้อนอินพุทสัญญาณซายน์ที่ความถี่ 2.5KHz แอมปลิจูด 10 Vp-p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงสัญญาณอินพุทที่ป้อนให้กับระบบโดยเป็นสัญญาณซายน์ที่ความถี่ 3KHz แอมปลิจูด 10 Vp-p



รูปที่ 4.19 แสดงสัญญาณเอาท์พุทของระบบเมื่อป้อนอินพุทสัญญาณซายน์ที่ความถี่ 3KHz แอมปลิจูด 10 Vp-p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองจะพบว่าสัญญาณเอาท์พุทที่ได้จะมีค่าแอมปลิจูดลดลงมาก ซึ่งเกิดจากการลดทอนของสัญญาณเมื่อสัญญาณผ่านเข้าไปในตัวระบบ ส่วนค่าความถี่ของเอาท์พุทจะมีค่าคงที่เท่าเดิมกับความถี่ของอินพุท

โดยเมื่อทางกลุ่มป้อนสัญญาณอินพุทที่ความถี่ 500Hz พบว่าสัญญาณเอาท์พุทที่ได้จะมีรูปสัญญาณที่ไม่ค่อยเป็นสัญญาณซายน์ และมีขนาดแอมปลิจูดลดลงจาก 20.6V เหลือ 7.4V

เมื่อทางกลุ่มป้อนสัญญาณอินพุทที่ความถี่ 2.5KHz พบว่าสัญญาณเอาท์พุทที่ได้จะมีรูปสัญญาณเป็นสัญญาณซายน์ที่ช่วงบนของรูปสัญญาณจะมีการแหว่งไปเล็กน้อยและมีสัญญาณรบกวนเล็กน้อยซ้อนทับอยู่บนตัวสัญญาณซายน์ และมีขนาดแอมปลิจูดลดลงจาก 20.6V เหลือ 7.92V

เมื่อทางกลุ่มป้อนสัญญาณอินพุทที่ความถี่ 3KHz พบว่าสัญญาณเอาท์พุทที่ได้จะมีรูปสัญญาณเป็นสัญญาณซายน์ที่ช่วงจุดยอดของรูปสัญญาณจะถูกขลิบออกไปไม่โค้งเหมือนเดิมโดยมีสัญญาณรบกวนเล็กน้อยไปซ้อนทับอยู่บนช่วงจุดยอดของตัวสัญญาณซายน์ และมีขนาดแอมปลิจูดลดลงจาก 20.6V เหลือ 8V

ซึ่งจะพบว่าเมื่อทำการป้อนสัญญาณซายน์ที่มีความถี่มากขึ้นจะทำให้เอาท์พุทของระบบมีความเป็นสัญญาณซายน์มากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5
บทวิจารณ์และบทสรุป

5.1 บทวิจารณ์

เนื่องจากโครงงานนี้ ทางกลุ่มได้เน้นการพยายามศึกษาหาข้อมูล และเก็บข้อมูลเกี่ยวกับระบบโทรศัพท์ทางไกลผ่านอินเตอร์เน็ท ซึ่งเป็นเทคโนโลยีที่ค่อนข้างใหม่ในประเทศไทย แม้จะมีการใช้งานแต่ก็เป็นส่วนน้อย จึงเป็นอุปสรรคในการค้นหาข้อมูลเพื่อนำมาประยุกต์ใช้ในโครงงานนี้

โดยผลการทดลองหลักๆจะเป็นการเชื่อมเส้นทางในการรับส่งข้อมูลเสียงระหว่างเครื่องคอมพิวเตอร์แล้วเชื่อมไปยังโครงข่ายโทรศัพท์พื้นฐาน ซึ่งปัญหาที่พบส่วนใหญ่เกี่ยวกับการเกิดสัญญาณรบกวนระหว่างสัญญาณเสียงที่ออกจากหลังการ์ดเสียงไปเข้าตัวโทรศัพท์เป็นสัญญาณเสีย่งของสายโทรศัพท์ จึงได้มีการแก้ปัญหาโดยการต่อวงจรเพื่อแก้ปัญหานี้ทำให้สามารถลดสัญญาณรบกวนไปได้ส่วนหนึ่ง

5.2 สรุปผลการทดลอง

จากผลการทดลองที่ได้ ทางกลุ่มสามารถเชื่อมโยงเส้นทางการรับส่งแพ็กเก็ตเสียงระหว่างเครื่องไคลเอ็นต์และเซิร์ฟเวอร์ได้และสามารถทำการสั่งการโมเด็มได้ผลเป็นที่น่าพอใจ โดยสัญญาณเสียงที่รับส่งระหว่างเครื่องไคลเอ็นต์และเซิร์ฟเวอร์มีคุณภาพเสียงที่ดี มีเสียงรบกวนน้อยมากซึ่งบางครั้งก็ไม่มีเลย แต่เมื่อนำสัญญาณเสียงไปเข้าตัวโทรศัพท์เป็นสัญญาณเสียงของสายโทรศัพท์ พบว่าเกิดสัญญาณรบกวนขึ้นมาก ทางกลุ่มจึงได้ต่อวงจรขึ้นมาแก้ไขทำให้สัญญาณรบกวนลดลง ทำให้สามารถสื่อสารพูดคุยกันรู้เรื่องได้ ทำให้คุณภาพเสียงดีขึ้น

5.3 ปัญหาที่พบจากการทดลอง

1. พบว่าสัญญาณเสียงที่รับส่งไปมาจะเกิดการดีเลย์ไปเล็กน้อย ซึ่งเมื่อใช้งานไปนานๆแล้วก็จะรู้จังหวะในการพูดไปมาระหว่างกันทำให้เกิดความเคยชินไปเอง

2. พบว่าเกิดสัญญาณรบกวนขึ้นมากเมื่อนำสัญญาณเสียงไปเข้าตัวโทรศัพท์เป็นสัญญาณเสียงของสายโทรศัพท์ ทางกลุ่มจึงแก้ไขโดยต่อวงจรขึ้นมาแก้ไขทำให้สัญญาณรบกวนนั้นลดลงไป คุณภาพเสียงดีขึ้น และทำการเปลี่ยนสายสัญญาณและหัวต่อต่างๆให้มีคุณภาพดีขึ้น ทำการเปลี่ยนการ์ดเสียงให้มีคุณภาพของเสียงดีขึ้นซึ่งทำให้เสียงที่ได้โดยรวมมีคุณภาพดีขึ้น

3. พบว่าในบางครั้งขณะทำการพดคุยจะเกิดเสียงสัญญาณรบกวนจากตัวโมเด็ม แต่เมื่อเราทำการปิดและเปิดโมเด็มใหม่อีกครั้งพบว่าสัญญาณรบกวนดังกล่าวหายไป

4. เมื่อทำการเปลี่ยนการ์ดเสียงมาเป็นชนิดที่มีประสิทธิภาพดีขึ้น พบว่าคุณภาพเสียงที่ได้ดีขึ้น

5.4 แนวทางในการพัฒนา

เนื่องจากที่ฝั่งเซิร์ฟเวอร์จะต้องมีคนคอยจัดการในการกดสวิตช์ของโทรศัพท์เพื่อเปลี่ยนสัญญาณ BUSY TONE เป็น DIAL TONE ของตัวโทรศัพท์จึงจะสามารถหมุนโมเด็มออกได้ และต้องคอยควบคุมโปรแกรมเมื่อยุติการสนทนาแล้ว อาจมีการพัฒนาโดยใช้สวิตช์เป็นตัวเปลี่ยนสถานะของตัวโทรศัพท์ทำให้ไม่ต้องมีคนมาคอยคุมที่เครื่องฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมทางฝั่งไคลเอ็นต์ (Client.vbp)

Form1.frm

```
Option Base 1
Dim a(10) As String
Public x As Integer
Dim up As Integer
Public c As Integer
Dim Fso As New Scripting.FileSystemObject
Dim Txt As TextStream
Dim z As Integer
Dim f As Integer
Dim e() As String
Public b As String
Private Sub Command1_Click()
If Text1.Text = "" And Text2.Text <> "" Then
MsgBox ("Please Enter Name")
ElseIf Text2.Text = "" And Text1.Text <> "" Then
MsgBox ("Please Enter Telephone Number")
ElseIf (Text1.Text = "") And (Text2.Text = "") Then
MsgBox ("Please Enter Name and Telephone Number")
ElseIf c = 0 Then
b = ""
b = Text1.Text + "&" + Text2.Text
Set Txt = Fso.CreateTextFile("c:\user.txt")
Txt.Write (b)
c = c + 1
a(c) = b
Call Showresult
Text1.Text = ""
Text2.Text = ""
ElseIf c >= 10 Then
MsgBox ("Please Delete Some Your Friend's Name Before Add New Your
Friend's Instead")
Else
b = b & vbCrLf + Text1.Text + "&" + Text2.Text
Set Txt = Fso.CreateTextFile("c:\user.txt")
Txt.Write (b)
Set Txt = Fso.OpenTextFile("c:\user.txt")
c = 0
Do
c = c + 1
a(c) = Txt.ReadLine
Loop While Txt.AtEndOfLine = False
Call Showresult
Text1.Text = ""
Text2.Text = ""
End If
End Sub

Private Sub Command2_Click()
If b = "" Or b = "Test" Then
Txt.Close
Fso.DeleteFile ("c:\user.txt")
End If
End
End Sub

Private Sub Command3_Click()

If Option1.Value = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Form2.Text1.Text = Name1.Caption
    Form2.Text2.Text = Tel1.Caption
    x = 1
    Form2.Show
ElseIf Option2.Value = True Then
    Form2.Text1.Text = Name2.Caption
    Form2.Text2.Text = Tel2.Caption
    x = 2
    Form2.Show
ElseIf Option3.Value = True Then
    Form2.Text1.Text = Name3.Caption
    Form2.Text2.Text = Tel3.Caption
    x = 3
    Form2.Show
ElseIf Option4.Value = True Then
    Form2.Text1.Text = Name4.Caption
    Form2.Text2.Text = Tel4.Caption
    x = 4
    Form2.Show
ElseIf Option5.Value = True Then
    Form2.Text1.Text = Name5.Caption
    Form2.Text2.Text = Tel5.Caption
    x = 5
    Form2.Show
ElseIf Option6.Value = True Then
    Form2.Text1.Text = Name6.Caption
    Form2.Text2.Text = Tel6.Caption
    x = 6
    Form2.Show
ElseIf Option7.Value = True Then
    Form2.Text1.Text = Name7.Caption
    Form2.Text2.Text = Tel7.Caption
    x = 7
    Form2.Show
ElseIf Option8.Value = True Then
    Form2.Text1.Text = Name8.Caption
    Form2.Text2.Text = Tel8.Caption
    x = 8
    Form2.Show
ElseIf Option9.Value = True Then
    Form2.Text1.Text = Name9.Caption
    Form2.Text2.Text = Tel9.Caption
    x = 9
    Form2.Show
ElseIf Option10.Value = True Then
    Form2.Text1.Text = Name10.Caption
    Form2.Text2.Text = Tel10.Caption
    x = 10
    Form2.Show
Else
MsgBox ("Please Select On Some Name(Not Blank)")
End If
End Sub

Private Sub Command4_Click()
If b = "" Or b = "Test" Then
Txt.Close
Fso.DeleteFile ("c:\user.txt")
End If
Form3.Show
Unload Me
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub Form_Load()
If Dir("c:\user.txt") = "user.txt" Then
Set Txt = Fso.OpenTextFile("c:\user.txt")
b = Txt.ReadAll
c = 0
If b <> "Test" Then
Set Txt = Fso.OpenTextFile("c:\user.txt")
Do
c = c + 1
a(c) = Txt.ReadLine
Loop While Txt.AtEndOfLine = False
Call Showresult
End If
Else
Set Txt = Fso.CreateTextFile("c:\user.txt")
Txt.Write ("Test")
Set Txt = Fso.OpenTextFile("c:\user.txt")
b = Txt.ReadAll
End If
End Sub

Public Sub Showresult()
Name1.Caption = ""
Tel1.Caption = ""
Name2.Caption = ""
Tel2.Caption = ""
Name3.Caption = ""
Tel3.Caption = ""
Name4.Caption = ""
Tel4.Caption = ""
Name5.Caption = ""
Tel5.Caption = ""
Name6.Caption = ""
Tel6.Caption = ""
Name7.Caption = ""
Tel7.Caption = ""
Name8.Caption = ""
Tel8.Caption = ""
Name9.Caption = ""
Tel9.Caption = ""
Name10.Caption = ""
Tel10.Caption = ""
If c <> 0 Then
c = 0
Set Txt = Fso.OpenTextFile("c:\user.txt")
Do
c = c + 1
a(c) = Txt.ReadLine
Loop While Txt.AtEndOfLine = False
Select Case c
Case 1
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
Case 2
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Name2.Caption = e(0)
Tel2.Caption = e(1)
Case 3
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
Case 4
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
Case 5
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
e = Split(a(5), "&", -1, vbTextCompare)
Name5.Caption = e(0)
Tel5.Caption = e(1)
Case 6
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
e = Split(a(5), "&", -1, vbTextCompare)
Name5.Caption = e(0)
Tel5.Caption = e(1)
e = Split(a(6), "&", -1, vbTextCompare)
Name6.Caption = e(0)
Tel6.Caption = e(1)
Case 7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
e = Split(a(5), "&", -1, vbTextCompare)
Name5.Caption = e(0)
Tel5.Caption = e(1)
e = Split(a(6), "&", -1, vbTextCompare)
Name6.Caption = e(0)
Tel6.Caption = e(1)
e = Split(a(7), "&", -1, vbTextCompare)
Name7.Caption = e(0)
Tel7.Caption = e(1)
Case 8
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
e = Split(a(5), "&", -1, vbTextCompare)
Name5.Caption = e(0)
Tel5.Caption = e(1)
e = Split(a(6), "&", -1, vbTextCompare)
Name6.Caption = e(0)
Tel6.Caption = e(1)
e = Split(a(7), "&", -1, vbTextCompare)
Name7.Caption = e(0)
Tel7.Caption = e(1)
e = Split(a(8), "&", -1, vbTextCompare)
Name8.Caption = e(0)
Tel8.Caption = e(1)
Case 9
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
e = Split(a(5), "&", -1, vbTextCompare)
Name5.Caption = e(0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Tel5.Caption = e(1)
e = Split(a(6), "&", -1, vbTextCompare)
Name6.Caption = e(0)
Tel6.Caption = e(1)
e = Split(a(7), "&", -1, vbTextCompare)
Name7.Caption = e(0)
Tel7.Caption = e(1)
e = Split(a(8), "&", -1, vbTextCompare)
Name8.Caption = e(0)
Tel8.Caption = e(1)
e = Split(a(9), "&", -1, vbTextCompare)
Name9.Caption = e(0)
Tel9.Caption = e(1)
Case 10
e = Split(a(1), "&", -1, vbTextCompare)
Name1.Caption = e(0)
Tel1.Caption = e(1)
e = Split(a(2), "&", -1, vbTextCompare)
Name2.Caption = e(0)
Tel2.Caption = e(1)
e = Split(a(3), "&", -1, vbTextCompare)
Name3.Caption = e(0)
Tel3.Caption = e(1)
e = Split(a(4), "&", -1, vbTextCompare)
Name4.Caption = e(0)
Tel4.Caption = e(1)
e = Split(a(5), "&", -1, vbTextCompare)
Name5.Caption = e(0)
Tel5.Caption = e(1)
e = Split(a(6), "&", -1, vbTextCompare)
Name6.Caption = e(0)
Tel6.Caption = e(1)
e = Split(a(7), "&", -1, vbTextCompare)
Name7.Caption = e(0)
Tel7.Caption = e(1)
e = Split(a(8), "&", -1, vbTextCompare)
Name8.Caption = e(0)
Tel8.Caption = e(1)
e = Split(a(9), "&", -1, vbTextCompare)
Name9.Caption = e(0)
Tel9.Caption = e(1)
e = Split(a(10), "&", -1, vbTextCompare)
Name10.Caption = e(0)
Tel10.Caption = e(1)
End Select
End If
End Sub

Private Sub Name1_Click()
frmClient.number = Tel1.Caption
frmClient.myid.Text = Name1.Caption
frmClient.lblNumber.Caption = Tel1.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Name10_Click()
frmClient.number = Tel10.Caption
frmClient.myid.Text = Name10.Caption
frmClient.lblNumber.Caption = Tel10.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name2_Click()
frmClient.number = Tel2.Caption
frmClient.myid.Text = Name2.Caption
frmClient.lblNumber.Caption = Tel2.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name3_Click()
frmClient.number = Tel3.Caption
frmClient.myid.Text = Name3.Caption
frmClient.lblNumber.Caption = Tel3.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name4_Click()
frmClient.number = Tel4.Caption
frmClient.myid.Text = Name4.Caption
frmClient.lblNumber.Caption = Tel4.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name5_Click()
frmClient.number = Tel5.Caption
frmClient.myid.Text = Name5.Caption
frmClient.lblNumber.Caption = Tel5.Caption
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name6_Click()
frmClient.number = Tel6.Caption
frmClient.myid.Text = Name6.Caption
frmClient.lblNumber.Caption = Tel6.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name7_Click()
frmClient.number = Tel7.Caption
frmClient.myid.Text = Name7.Caption
frmClient.lblNumber.Caption = Tel7.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name8_Click()
frmClient.number = Tel8.Caption
frmClient.myid.Text = Name8.Caption
frmClient.lblNumber.Caption = Tel8.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name9_Click()
frmClient.number = Tel9.Caption
frmClient.myid.Text = Name9.Caption
frmClient.lblNumber.Caption = Tel9.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name6_Click()
frmClient.number = Tel6.Caption
frmClient.myid.Text = Name6.Caption
frmClient.lblNumber.Caption = Tel6.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name7_Click()
frmClient.number = Tel7.Caption
frmClient.myid.Text = Name7.Caption
frmClient.lblNumber.Caption = Tel7.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name8_Click()
frmClient.number = Tel8.Caption
frmClient.myid.Text = Name8.Caption
frmClient.lblNumber.Caption = Tel8.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub

Private Sub Name9_Click()
frmClient.number = Tel9.Caption
frmClient.myid.Text = Name9.Caption
frmClient.lblNumber.Caption = Tel9.Caption
frmClient.Show
Unload Me
If frmClient.txtIP.Text = "" Or frmClient.txtPort.Text = "" Then
     frmIPPORT.Show vbModal
Else: frmClient.IP = frmClient.txtIP.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      frmClient.Port = Val(frmClient.txtPort.Text)
Call frmClient.ImgCall_Click
End If
End Sub
```

Form2.frm

```
Option Explicit
Public n As String
Dim p As String
Dim q As String
Public o As Integer
Dim m As Integer
Private Sub Command1_Click()
Dim Fso As New Scripting.FileSystemObject
Dim Txt As TextStream
Set Txt = Fso.OpenTextFile("c:\user.txt")
n = Txt.ReadAll
If Text1.Text = "" And Text2.Text <> "" Then
MsgBox ("Please Enter Name")
ElseIf Text2.Text = "" And Text1.Text <> "" Then
MsgBox ("Please Enter Telephone Number")
ElseIf (Text1.Text = "") And (Text2.Text = "") Then
m = Form1.x
Select Case m
Case 1
    p = Form1.Name1.Caption
    q = Form1.Tel1.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
ElseIf Form1.c <> 1 Then
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 2
    p = Form1.Name2.Caption
    q = Form1.Tel2.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 3
    p = Form1.Name3.Caption
    q = Form1.Tel3.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 4
    p = Form1.Name4.Caption
    q = Form1.Tel4.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.c = Form1.c - 1
Case 5
    p = Form1.Name5.Caption
    q = Form1.Tel5.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 6
    p = Form1.Name6.Caption
    q = Form1.Tel6.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 7
    p = Form1.Name7.Caption
    q = Form1.Tel7.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 8
    p = Form1.Name8.Caption
    q = Form1.Tel8.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 9
    p = Form1.Name9.Caption
    q = Form1.Tel9.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
Case 10
    p = Form1.Name10.Caption
    q = Form1.Tel10.Caption
If Form1.c <> m Then
    n = Replace(n, p + "&" + q & vbCrLf, "", 1, 1, vbTextCompare)
Else
    n = Replace(n, vbCrLf + p + "&" + q, "", 1, 1, vbTextCompare)
End If
Form1.c = Form1.c - 1
End Select
Set Txt = Fso.CreateTextFile("c:\user.txt")
Txt.Write (n)
Unload Me
o = 1
Form1.b = n
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Else
m = Form1.x
n = Form1.b
Select Case m
Case 1
    p = Form1.Name1.Caption
    q = Form1.Tel1.Caption
    Form1.Name1.Caption = Text1.Text
    Form1.Tel1.Caption = Text2.Text
    n = Replace(n, p, Form1.Name1.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel1.Caption, 1, 1, vbTextCompare)
Case 2
    p = Form1.Name2.Caption
    q = Form1.Tel2.Caption
    Form1.Name2.Caption = Text1.Text
    Form1.Tel2.Caption = Text2.Text
    n = Replace(n, p, Form1.Name2.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel2.Caption, 1, 1, vbTextCompare)
Case 3
    p = Form1.Name3.Caption
    q = Form1.Tel3.Caption
    Form1.Name3.Caption = Text1.Text
    Form1.Tel3.Caption = Text2.Text
    n = Replace(n, p, Form1.Name3.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel3.Caption, 1, 1, vbTextCompare)
Case 4
    p = Form1.Name4.Caption
    q = Form1.Tel4.Caption
    Form1.Name4.Caption = Text1.Text
    Form1.Tel4.Caption = Text2.Text
    n = Replace(n, p, Form1.Name4.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel4.Caption, 1, 1, vbTextCompare)
Case 5
    p = Form1.Name5.Caption
    q = Form1.Tel5.Caption
    Form1.Name5.Caption = Text1.Text
    Form1.Tel5.Caption = Text2.Text
    n = Replace(n, p, Form1.Name5.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel5.Caption, 1, 1, vbTextCompare)
Case 6
    p = Form1.Name6.Caption
    q = Form1.Tel6.Caption
    Form1.Name6.Caption = Text1.Text
    Form1.Tel6.Caption = Text2.Text
    n = Replace(n, p, Form1.Name6.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel6.Caption, 1, 1, vbTextCompare)
Case 7
    p = Form1.Name7.Caption
    q = Form1.Tel7.Caption
    Form1.Name7.Caption = Text1.Text
    Form1.Tel7.Caption = Text2.Text
    n = Replace(n, p, Form1.Name7.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel7.Caption, 1, 1, vbTextCompare)
Case 8
    p = Form1.Name8.Caption
    q = Form1.Tel8.Caption
    Form1.Name8.Caption = Text1.Text
    Form1.Tel8.Caption = Text2.Text
    n = Replace(n, p, Form1.Name8.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel8.Caption, 1, 1, vbTextCompare)
Case 9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    p = Form1.Name9.Caption
    q = Form1.Tel9.Caption
    Form1.Name9.Caption = Text1.Text
    Form1.Tel9.Caption = Text2.Text
    n = Replace(n, p, Form1.Name9.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel9.Caption, 1, 1, vbTextCompare)
Case 10
    p = Form1.Name10.Caption
    q = Form1.Tel10.Caption
    Form1.Name10.Caption = Text1.Text
    Form1.Tel10.Caption = Text2.Text
    n = Replace(n, p, Form1.Name10.Caption, 1, 1, vbTextCompare)
    n = Replace(n, q, Form1.Tel10.Caption, 1, 1, vbTextCompare)
End Select
Set Txt = Fso.CreateTextFile("c:\user.txt")
Txt.Write (n)
Unload Me
o = 1
Form1.b = n
End If
Call Form1.Showresult
End Sub
```

Form3.frm

```
Private Sub Command1_Click()
frmClient.Show
Unload Me
End Sub

Private Sub Command2_Click()
Form1.Show
Unload Me
End Sub

Private Sub Command3_Click()
End
End Sub
```

---

frmClient.frm

```
Option Explicit
Public number As String
Public IP As String
Public Port As Integer
Public Index As Integer

Private Sub Command1_Click()
Form3.Show
Unload Me
End Sub

Private Sub Form_Load()
IP = txtIP.Text
Port = Val(txtPort.Text)
audioserver.AudInitServer True
audioserver.AudStream True
audioclient.AudReceiveSound True
Status.Text = "Checking Audio devices.."
Timer1.Enabled = True
End Sub

Private Sub audioclient_OnConnectionAccepted()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
audioclient.AudConnect txtIP.Text
wsckClient.SendData number
End Sub
Private Sub audioclient_OnConnectionLost()
ImgCall.Enabled = True
ImgHangup.Enabled = False
End Sub
Private Sub audioclient_OnConnectionRefused()
Status.Text = "Connection Refused"
End Sub
Private Sub audioclient_OnErrorConnecting()
Status.Text = "Error Connecting ..."
End Sub
Private Sub audioserver_OnClientConnected(ByVal IP As String)
Status.Text = "Call in Progress"
audioclient.AudConnect IP
End Sub
Private Sub audioserver_OnClientDisconnected(ByVal IP As String)
audioclient.AudHangUp
ImgCall.Enabled = True
ImgHangup.Enabled = False

Status.Text = "idle ..."
End Sub

Private Sub audioserver_OnConnectionRequested(ByVal IP As String,
ByVal user As String)
Status.Text = "Call from : " + user
 If (MsgBox("Would you like to accept the Connection ?", vbYesNo) =
vbYes) Then
 audioserver.AudAcceptConnection True
 Status.Text = "Connection Accepted"
 ImgCall.Enabled = False
 ImgHangup.Enabled = True

 Else
 audioserver.AudAcceptConnection False
 Status.Text = "Connection Refused"
 ImgCall.Enabled = True
ImgHangup.Enabled = False

 End If

End Sub

Private Sub cmdNumber_Click(Index As Integer)
lblNumber.Caption = lblNumber.Caption + cmdNumber(Index).Caption
number = lblNumber.Caption
End Sub
Public Sub ImgCall_Click()
If txtIP.Text = "" Then
    MsgBox "Please Insert IP Address"
ElseIf txtPort.Text = "" Then
    MsgBox "Please Insert Port"
ElseIf lblNumber.Caption = "" Then
    MsgBox "Please Insert Telephone Number"
Else
IP = txtIP.Text
Port = Val(txtPort.Text)
audioclient.AudRequestConnection txtIP.Text, myid.Text
wsckClient.RemoteHost = IP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
wsckClient.RemotePort = Port
wsckClient.Connect
ImgCall.Enabled = False
ImgHangup.Enabled = True
End If
End Sub
Private Sub ImgHangup_Click()
audioclient.AudHangUp
ImgCall.Enabled = True
ImgHangup.Enabled = False
wsckClient.Close
lblNumber.Caption = ""
End Sub
```

```
Private Sub Timer1_Timer()
If audioclient.AudAudioAvailable = True And
audioserver.AudAudioAvailable = True Then
Status.Text = "Audio Devices = OK!"
Else
Status.Text = "Error detecting Audio Devices!"
End If
Timer1.Enabled = False
End Sub
```

---

frmIPPORT.frm

```
Option Explicit

Private Sub Command1_Click()
If Text1.Text <> "" And Text2.Text <> "" Then
frmClient.txtIP.Text = Text1.Text
frmClient.txtPort.Text = Text2.Text
Call frmClient.ImgCall_Click
Unload Me
End If
End Sub
```

ส่วนของโปรแกรมทางฝั่งเซิร์ฟเวอร์ (Server.vbp)

frmServer.frm

```
Option Explicit
Public Number As String

Private Sub Command1_Click()
End
End Sub

Private Sub Form_Load()
'Set Port number
wsckServer.LocalPort = "1234"
'Waiting for incoming connect
wsckServer.Listen
audioserver.AudInitServer True
audioserver.AudStream True
audioclient.AudReceiveSound True
Status.Text = "Checking Audio devices.."
Timer1.Enabled = True
End Sub
Private Sub audioclient_OnConnectionAccepted()
audioclient.AudConnect txtIP.Text
End Sub
Private Sub audioclient_OnConnectionRefused()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Status.Text = "Connection Refused"
End Sub
Private Sub audioclient_OnErrorConnecting()
Status.Text = "Error Connecting ..."
End Sub
Private Sub audioserver_OnClientConnected(ByVal Ip As String)
Status.Text = "Call in Progress"
audioclient.AudConnect Ip
End Sub
Private Sub audioserver_OnClientDisconnected(ByVal Ip As String)
audioclient.AudHangUp
Status.Text = "idle ..."
End Sub
Private Sub audioserver_OnConnectionRequested(ByVal Ip As String,
ByVal user As String)
 myid.Text = user
 audioserver.AudAcceptConnection True
 Status.Text = "Connection Accepted"
 End Sub
Private Sub wsckServer_Close()
wsckServer.Close
wsckServer.Listen
End Sub

Private Sub wsckServer_ConnectionRequest(ByVal requestID As Long)
wsckServer.Close
wsckServer.Accept requestID
End Sub

Private Sub wsckServer_DataArrival(ByVal bytesTotal As Long)
Dim strA As String
wsckServer.GetData Number
txtInput.Text = Number
strA = Number
comOne.CommPort = 1
comOne.Settings = "9600,N,8,1"
comOne.PortOpen = True
comOne.Output = "ATDT" & strA & vbCr
MsgBox "Dialing " & strA & vbCrLf & "Just A Moment..." & vbCrLf &
"Press OK To Disconnect", _
        vbOKOnly, "Dial-A-Phone"
comOne.PortOpen = False
End Sub

Private Sub Timer1_Timer()
If audioclient.AudAudioAvailable = True And
audioserver.AudAudioAvailable = True Then
Status.Text = "Audio Devices = OK!"
Else
Status.Text = "Error detecting Audio Devices!"
End If
Timer1.Enabled = False
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ Active X Control

```
Option Explicit
'== ACM API Constants
Public Const ACMERR_BASE = 512
Public Const ACMERR_NOTPOSSIBLE = (ACMERR_BASE + 0)
Public Const ACMERR_BUSY = (ACMERR_BASE + 1)
Public Const ACMERR_UNPREPARED = (ACMERR_BASE + 2)
Public Const ACMERR_CANCELED = (ACMERR_BASE + 3)

' AcmStreamSize Flags...
Public Const ACM_STREAMSIZEF_SOURCE = &H0&
Public Const ACM_STREAMSIZEF_DESTINATION = &H1&
Public Const ACM_STREAMSIZEF_QUERYMASK = &HF&

' acmStreamConvert Flags...
Public Const ACM_STREAMCONVERTF_BLOCKALIGN = &H4&
Public Const ACM_STREAMCONVERTF_START = &H10&
Public Const ACM_STREAMCONVERTF_END = &H20&

' Done Bits For ACMSTREAMHEADER.fdwStatus
Public Const ACMSTREAMHEADER_STATUSF_DONE = &H10000
Public Const ACMSTREAMHEADER_STATUSF_PREPARED = &H20000
Public Const ACMSTREAMHEADER_STATUSF_INQUEUE = &H100000

' Done Bits For acmStreamOpen Formats
Public Const ACM_STREAMOPENF_QUERY = &H1&
Public Const ACM_STREAMOPENF_ASYNC = &H2&
Public Const ACM_STREAMOPENF_NONREALTIME = &H4&

'== ACM API Declarations
Public Declare Function acmStreamOpen Lib "MSACM32" (hAS As Long,
ByVal hADrv As Long, wfxSrc As WAVEFORMATEX, wfxDst As WAVEFORMATEX,
ByVal wFltr As Long, ByVal dwCallback As Long, ByVal dwInstance As
Long, ByVal fdwOpen As Long) As Long
Public Declare Function acmStreamClose Lib "MSACM32" (ByVal hAS As
Long, ByVal dwClose As Long) As Long
Public Declare Function acmStreamPrepareHeader Lib "MSACM32" (ByVal
hAS As Long, hASHdr As ACMSTREAMHEADER, ByVal dwPrepare As Long) As
Long
Public Declare Function acmStreamUnprepareHeader Lib "MSACM32" (ByVal
hAS As Long, hASHdr As ACMSTREAMHEADER, ByVal dwUnPrepare As Long) As
Long
Public Declare Function acmStreamConvert Lib "MSACM32" (ByVal hAS As
Long, hASHdr As ACMSTREAMHEADER, ByVal dwConvert As Long) As Long
Public Declare Function acmStreamReset Lib "MSACM32" (ByVal hAS As
Long, ByVal dwReset As Long) As Long
Public Declare Function acmStreamSize Lib "MSACM32" (ByVal hAS As
Long, ByVal cbInput As Long, dwOutBytes As Long, ByVal dwSize As
Long) As Long

'== ACM User Defined Datatypes
Type WAVEFILTER
    cbStruct      As Long
    dwFilterTag   As Long
    fdwFilter     As Long
    dwReserved(5) As Long
End Type

Type ACMSTREAMHEADER            ' [ACM STREAM HEADER TYPE]
    cbStruct As Long            ' Size of header in bytes
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    dwStatus As Long            ' Conversion status buffer
    dwUser As Long              ' 32 bits of user data specified by
application
    pbSrc As Long               ' Source data buffer pointer
    cbSrcLength As Long         ' Source data buffer size in bytes
    cbSrcLengthUsed As Long     ' Source data buffer size used in
bytes
    dwSrcUser As Long           ' 32 bits of user data specified by
application
    cbDst As Long               ' Dest data buffer pointer
    cbDstLength As Long         ' Dest data buffer size in bytes
    cbDstLengthUsed As Long     ' Dest data buffer size used in bytes
    dwDstUser As Long           ' 32 bits of user data specified by
application
    dwReservedDriver(9) As Long ' Reserved and should not be used
End Type
-------------------------------------------------------------------
Option Explicit

' Application User Defined Types...
' Sound Format
'Public Const WAVE_FORMAT_PCM = &H1                  ' Microsoft
Windows PCM Wave Format
'Public Const WAVE_FORMAT_ADPCM = &H11               ' ADPCM Wave
Format
'Public Const WAVE_FORMAT_IMA_ADPCM = &H11           ' IMA ADPCM Wave
Format
'Public Const WAVE_FORMAT_DVI_ADPCM = &H11           ' DVI ADPCM Wave
Format
'Public Const WAVE_FORMAT_DSPGROUP_TRUESPEECH = &H22 ' DSP Group Wave
Format
'Public Const WAVE_FORMAT_GSM610 = &H31              ' GSM610 Wave
Format
'Public Const WAVE_FORMAT_MSN_AUDIO = &H32           ' MSN Audio Wave
Format

Public Const TIMESLICE = 0.2            ' Time Slicing 1/5 Second

' Application Constants...
Public Const NoOfRings = 1                  ' Number Of Times In/Out
Bound Calls Ring...

Public Const phoneHungUp = 3                ' Hangup Status Icon...
Public Const phoneRinging = 2               ' Ringing Status Icon...
Public Const phoneAnswered = 1              ' Answered Status Icon...
Public Const mikeNO = 6
Public Const mikeOFF = 7
Public Const mikeON = 8
Public Const speakNO = 9
Public Const speakOFF = 10
Public Const speakON = 11

Public Const RingInId = 101                 ' Ringing InBound
Sound...
Public Const RingOutId = 102                ' Ringing OutBound
Sound...

' Toolbar constants...
Public Const tbCALL = 2
Public Const tbHANGUP = 3
Public Const tbAUTOANSWER = 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'== flag values for wFlags parameter
'Public Const SND_SYNC = &H0                    '  play synchronously
(default)
'Public Const SND_ASYNC = &H1                   '  play asynchronously
'Public Const SND_NODEFAULT = &H2               '  don't use default
sound
'Public Const SND_MEMORY = &H4                  '  lpszSoundName points
to a memory file
'Public Const SND_LOOP = &H8                    '  loop the sound until
next sndPlaySound
'Public Const SND_NOSTOP = &H10                 '  don't stop any
currently playing sound
```

```
'== MCI Wave API Declarations
Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA"
(ByVal SoundData As Any, ByVal uFlags As Long) As Long
-----------------------------------------------------------------------
Option Explicit
Public Sub InitServerList(ServerList As ComboBox)
    ' Populate Server List Box...
    ServerList.AddItem "localhost"
    ServerList.AddItem "127.0.0.1"
End Sub

Public Sub DebugSocket(TCPSocket As Winsock)
' Prints Information In A TCP Socket, For Debugging TCP Events...

    Debug.Print "TCPSocket.RemoteHost", TCPSocket.RemoteHost
    Debug.Print "TCPSocket.RemoteHostIP", TCPSocket.RemoteHostIP
    Debug.Print "TCPSocket.RemotePort", TCPSocket.RemotePort
    Debug.Print "TCPSocket.LocalHostName", TCPSocket.LocalHostName
    Debug.Print "TCPSocket.LocalIP", TCPSocket.LocalIP
    Debug.Print "TCPSocket.LocalPort", TCPSocket.LocalPort
    Debug.Print "TCPSocket.State", TCPSocket.State
    Debug.Print
End Sub

Public Sub ResPlaySound(ResourceId As Long)
' Uses Sound Play Sound To Play Back PreRecorded WaveFiles

    Dim sndBuff As String
    sndBuff = StrConv(LoadResData(ResourceId, "WAVE"), vbUnicode)
    Call sndPlaySound(sndBuff, SND_SYNC Or SND_MEMORY)

End Sub

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

    ConnList.AddItem MemberID                   ' Add New Member To
List
    ConnList.ItemData(ConnList.NewIndex) = Socket.Index
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Public Sub RemoveConnectionFromList(Socket As Winsock, ConnList As
ListBox)
' Removes A Connection Reference From A ListBox

    Dim Conn As Long                                    ' Connection
Array Element Variable
    Dim MemberID As String                              ' Connection
Reference Variable

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

    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
postition
    Dim Socket As Winsock                             ' Winsock
socket

    sStart = 1
    sEnd = InStr(1, MemberID, " ") - 1              ' Get end of
remote ip address
    If (sEnd > 1) Then
        RemoteIP = Mid(MemberID, sStart, sEnd)      ' Get remote host
ip address
        sStart = InStr(sEnd, MemberID, "[") + 1     ' Get start of
remote port
        If (sStart > 1) Then                        ' If Start found
            sEnd = InStr(sStart, MemberID, "]") - 1 ' Get end of
remote port
            If (sEnd > 2) Then                      ' If end found
                RemPortID = Val(Mid(MemberID, sStart, sEnd)) ' Get
RemotePort
                sStart = InStr(sEnd, MemberID, "[") + 1      ' Get
start of local port
                If (sStart > 1) Then
                    sEnd = InStr(sStart, MemberID, "]") - 1  ' Get
end of local port
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    If (sEnd > 2) Then                          ' If End
Found
                        LocPortID = Val(Mid(MemberID, sStart, sEnd))
' Extract local port
                        For Each Socket In Sockets
                            If ((Socket.RemoteHostIP = RemoteIP) And
_
                                (Socket.RemotePort = RemPortID) And _
                                (Socket.LocalPort = LocPortID) And _
                                (Socket.Index > 0)) Then    ' Was a
match found???
                                Index = Socket.Index        ' Save
and return index
                                Exit Sub                    ' Done...
exit
                            End If
                        Next
                    End If
                End If
            End If
        End If
    End If
End Sub
-----------------------------------------------------------------------
' Public Constants
Public Const MAXEXTRABYTES = 3          ' Maximum (Extra Bytes + 1)
In Non PCM Wave Formats...
Public Const MAXBUFFERS = 500           ' Maximum Wave Buffer Array
Member
Public Const MINBUFFERS = 0             ' Minimum Wave Buffer Array
Member

'== MCI Wave Constants
' Sound Quality
Public Const c8_0kHz = 8000             ' 8.0 khz
Public Const c11_025kHz = 11025         ' 11.025 khz
Public Const c22_05kHz = 22050          ' 22.05 khz
Public Const c44_1kHz = 44010           ' 44.1 khz

' Sound Format
Public Const WAVE_FORMAT_PCM = &H1                  ' Microsoft
Windows PCM Wave Format
Public Const WAVE_FORMAT_ADPCM = &H11               ' ADPCM Wave
Format
Public Const WAVE_FORMAT_IMA_ADPCM = &H11           ' IMA ADPCM Wave
Format
Public Const WAVE_FORMAT_DVI_ADPCM = &H11           ' DVI ADPCM Wave
Format
Public Const WAVE_FORMAT_DSPGROUP_TRUESPEECH = &H22 ' DSP Group Wave
Format
Public Const WAVE_FORMAT_GSM610 = &H31              ' GSM610 Wave
Format
Public Const WAVE_FORMAT_MSN_AUDIO = &H32           ' MSN Audio Wave
Format

' PCM Wave Format Types
Public Const WAVE_FORMAT_1M08 = &H1     '  11.025 kHz, Mono,   8-bit
Public Const WAVE_FORMAT_1M16 = &H4     '  11.025 kHz, Mono,   16-bit
Public Const WAVE_FORMAT_1S08 = &H2     '  11.025 kHz, Stereo, 8-bit
Public Const WAVE_FORMAT_1S16 = &H8     '  11.025 kHz, Stereo, 16-bit
Public Const WAVE_FORMAT_2M08 = &H10    '  22.05  kHz, Mono,   8-bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Const WAVE_FORMAT_2M16 = &H40     '  22.05  kHz, Mono,   16-bit
Public Const WAVE_FORMAT_2S08 = &H20     '  22.05  kHz, Stereo, 8-bit
Public Const WAVE_FORMAT_2S16 = &H80     '  22.05  kHz, Stereo, 16-bit
Public Const WAVE_FORMAT_4M08 = &H100    '  44.1   kHz, Mono,   8-bit
Public Const WAVE_FORMAT_4M16 = &H400    '  44.1   kHz, Mono,   16-bit
Public Const WAVE_FORMAT_4S08 = &H200    '  44.1   kHz, Stereo, 8-bit
Public Const WAVE_FORMAT_4S16 = &H800    '  44.1   kHz, Stereo, 16-bit

'== Wave...Open() Constants Public Const WAVE_FORMAT_QUERY = &H1&    '
Query wave format flag
Public Const WAVE_MAPPER = (-1)          ' Maps To First Available
Sound Device
Public Const WAVE_ALLOWSYNC = &H2&       ' Asynchronous playback flag
Public Const CALLBACK_WINDOW = &H10000   ' dwCallback is a HWND
Public Const CALLBACK_NULL = &H0&        ' no callback

'== MCI WaveHeader Bit Values In dwflags
Public Const WHDR_DONE = &H1&            '[00001][01] done bit
Public Const WHDR_PREPARED = &H2&        '[00010][02] set if this
header has been prepared
Public Const WHDR_BEGINLOOP = &H4&       '[00100][04] loop start block
Public Const WHDR_ENDLOOP = &H8&         '[01000][08] loop end block
Public Const WHDR_INQUEUE = &H10&        '[10000][16] reserved for
driver

'== MCI MM Return Codes
Public Const ERROR_SHARING_VIOLATION = 32
Public Const MMSYSERR_NOERROR = 0                    '  no
error
Public Const MMSYSERR_BASE = 0
Public Const MMSYSERR_ERROR = (MMSYSERR_BASE + 1)          '
unspecified error
Public Const MMSYSERR_BADDEVICEID = (MMSYSERR_BASE + 2)    '  device
ID out of range
Public Const MMSYSERR_NOTENABLED = (MMSYSERR_BASE + 3)     '  driver
failed enable
Public Const MMSYSERR_ALLOCATED = (MMSYSERR_BASE + 4)      '  device
already allocated
Public Const MMSYSERR_INVALHANDLE = (MMSYSERR_BASE + 5)    '  device
handle is invalid
Public Const MMSYSERR_NODRIVER = (MMSYSERR_BASE + 6)       '  no
device driver present
Public Const MMSYSERR_NOMEM = (MMSYSERR_BASE + 7)          '  memory
allocation error
Public Const MMSYSERR_NOTSUPPORTED = (MMSYSERR_BASE + 8)   '
function isn't supported
Public Const MMSYSERR_BADERRNUM = (MMSYSERR_BASE + 9)      '  error
value out of range
Public Const MMSYSERR_INVALFLAG = (MMSYSERR_BASE + 10)     '  invalid
flag passed
Public Const MMSYSERR_INVALPARAM = (MMSYSERR_BASE + 11)    '  invalid
parameter passed
Public Const MMSYSERR_LASTERROR = (MMSYSERR_BASE + 11)     '  last
error in range

Public Const WAVERR_BASE = 32
Public Const WAVERR_BADFORMAT = (WAVERR_BASE + 0)          '
unsupported wave format
Public Const WAVERR_STILLPLAYING = (WAVERR_BASE + 1)        '  still
something playing
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Const WAVERR_UNPREPARED = (WAVERR_BASE + 2)          '  header
not prepared
Public Const WAVERR_LASTERROR = (WAVERR_BASE + 3)           '  last
error in range
Public Const WAVERR_SYNC = (WAVERR_BASE + 3)                '  device
is synchronous

'== 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 timeGetTime Lib "winmm.dll" () As Long
'Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA"
(ByVal SoundData As Any, ByVal uFlags As Long) As Long

Declare Function waveInOpen Lib "winmm.dll" (lphWaveIn As Long, ByVal
uDeviceID As Long, lpFormat As WAVEFORMATEX, ByVal dwCallback As
Long, ByVal dwInstance As Long, ByVal dwFlags As Long) As Long
Declare Function waveInPrepareHeader Lib "winmm.dll" (ByVal hWaveIn
As Long, wH As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveInUnprepareHeader Lib "winmm.dll" (ByVal hWaveIn
As Long, wH As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveInAddBuffer Lib "winmm.dll" (ByVal hWaveIn As
Long, wH As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveInStart Lib "winmm.dll" (ByVal hWaveIn As Long)
As Long
Declare Function waveInStop Lib "winmm.dll" (ByVal hWaveIn As Long)
As Long
Declare Function waveInReset Lib "winmm.dll" (ByVal hWaveIn As Long)
As Long
Declare Function waveInClose Lib "winmm.dll" (ByVal hWaveIn As Long)
As Long

Declare Function waveOutOpen Lib "winmm.dll" (lphWaveOut As Long,
ByVal uDeviceID As Long, lpFormat As WAVEFORMATEX, ByVal dwCallback
As Long, ByVal dwInstance As Long, ByVal dwFlags As Long) As Long
Declare Function waveOutPrepareHeader Lib "winmm.dll" (ByVal hWaveOut
As Long, wH As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveOutUnprepareHeader Lib "winmm.dll" (ByVal
hWaveOut As Long, wH As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveOutWrite Lib "winmm.dll" (ByVal hWaveOut As
Long, wH As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveOutClose Lib "winmm.dll" (ByVal hWaveOut As
Long) As Long
Declare Function waveOutReset Lib "winmm.dll" (ByVal hWaveOut As
Long) As Long
Declare Function waveOutPause Lib "winmm.dll" (ByVal hWaveOut As
Long) As Long
Declare Function waveOutRestart Lib "winmm.dll" (ByVal hWaveOut As
Long) As Long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'== Global Memory Functions
Declare Function GlobalAlloc Lib "kernel32" (ByVal wFlags As Long,
ByVal dwBytes As Long) As Long
Declare Function GlobalFree Lib "kernel32" (ByVal hMem As Long) As
Long
Declare Function GlobalLock Lib "kernel32" (ByVal hMem As Long) As
Long
Declare Function GlobalUnlock Lib "kernel32" (ByVal hMem As Long) As
Long

Declare Sub CopyPTRtoBYTES Lib "Kernel32.dll" Alias "RtlMoveMemory"
(ByRef ByteDest As Byte, ByVal PtrSrc As Long, ByVal length As Long)
Declare Sub CopyBYTEStoPTR Lib "Kernel32.dll" Alias "RtlMoveMemory"
(ByVal PtrDest As Long, ByRef ByteSrc As Byte, ByVal length As Long)

Public Const GMEM_MOVEABLE = &H2&
Public Const GMEM_SHARE = &H2000&
Public Const GMEM_ZEROINIT = &H40&

'== MCI User Defined Data
Types...=====================================
Type WAVEHDR
    lpData As Long                ' pointer to locked data buffer
    dwBufferLength As Long        ' length of data buffer
    dwBytesRecorded As Long       ' used for input only
    dwUser As Long                ' for client's use
    dwFlags As Long               ' assorted flags (see defines)
    dwLoops As Long               ' loop control counter
    wavehdr_tag As Long           ' reserved for driver
    Reserved As Long              ' reserved for driver
    hData As Long                 ' handle to locked data buffer
End Type

Type WAVEFORMATEX
    wFormatTag As Integer         ' format type
    nChannels As Integer          ' number of channels (i.e. mono,
stereo, etc.)
    nSamplesPerSec As Long        ' sample rate
    nAvgBytesPerSec As Long       ' for buffer estimation
    nBlockAlign As Integer        ' block size of data
    wBitsPerSample As Integer     ' Bits Per Sample
    cbSize As Integer             ' Size Of (FACT CHUNCK)
    xBytes(MAXEXTRABYTES) As Byte ' (FACT CHUNCK)
End Type
-----------------------------------------------------------------------
Option Explicit

'== SUB MAIN
Sub main()
End Sub
'== SUB MAIN
-----------------------------------------------------------------------
Option Explicit

Global Number As String


Global Ip As String
Global Port As Integer
-----------------------------------------------------------------------
Option Explicit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'== 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 NULLPORTID = 0                      ' Null Port ID - A Port
ID That Will Never Be Used...
-------------------------------------------------------------------
Option Explicit
Public Function InstanceTCP(TCPArray As Variant) As Long
    Dim Ind As Long                                      ' Array Index
Var...
    InstanceTCP = -1                                     ' Set Default
Value
    On Error GoTo InitControl                            ' IF Error Then
Control Is Available

    For Ind = MINTCP To MAXTCP                           ' For Each Member
In TCPArray() + 1
        If (TCPArray(Ind).Name = "") Then                ' If Control Is
Not Valid Then..
        End If                                           ' ..A Runtime
Error Will Occure
    Next                                                 ' Search Next
Item In Array

InitControl:                                             ' Initialize New
Control
    On Error GoTo ErrorHandler                           ' Enable Error
Handling...

    If ((Ind >= MINTCP) And (Ind <= MAXTCP)) Then        ' Check to make
sure index value is with in range
        Load TCPArray(Ind)                               ' Create New
Member In TCPArray
        InstanceTCP = Ind                                ' Return New
TCPctl Index
    End If

    Exit Function                                        ' Exit
ErrorHandler:                                            ' Handler

    Debug.Print Err.Number, Err.Description              ' Debug Errors
    Resume Next                                          ' Ignore Error
And Continue
End Function

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Do While ((Socket.State = sckConnecting) Or _
              (Socket.State = sckConnectionPending) Or _
              (Socket.State = sckResolvingHost) Or _
              (Socket.State = sckHostResolved) Or _
              (Socket.State = sckOpen))            ' Attempting To
Connect...
        DoEvents                                   ' Post Events
    Loop                                           ' Keep Waiting

    Connect = (Socket.State = sckConnected)        ' Did Socket Connect
On Port...
End Function

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

Public Function CloseListen(Socket As Winsock) As Long
' Stops Listening On A LocalPort For A Remote Connection Request...
    If (Socket.State = sckListening) Then          ' Is Socket
Listening?
        Socket.Close                               ' Close Listen
    End If
End Function
Public Sub Disconnect(Socket As Winsock)
' Disconnects A Remote WinSock TCP/IP Connection If Connected
    If (Socket.State <> sckClosed) Then            ' Is Socket Already
Closed?
        Socket.Close                               ' Close Socket
    End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ฉัททวุฒิ พืชผล และ พิชิต สันติกุลานนท์,"คู่มือเรียนVisual Basic 6",บริษัท โปรวิชัน จำกัด.

[2] สุรชัย เพิ่มสินทวี และ ธเนศ สุวรรณผ่อง,"การสื่อสารข้อมูลด้วยModem",บริษัท ซีเอ็ดยูเคชั่น จำกัด.

[3] สุรศักดิ์ สงวนพงศ์,"สถาปัตยกรรมและโปรโตคอลTCP/IP",โครงการตำราวิชาการ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

[4] John Clark Craig และ Jeff Webb,"การเขียนโปรแกรมบนวินโดว์ด้วย ไมโครซอฟท์วิชวลเบสิก 6.0 ภาคปฏิบัติ",บริษัท ไมโครซอฟท์เพรส จำกัด.

[5] Guy Eddon และ Henry Eddon,"การพัฒนาคอมโปเนนท์ด้วย วิชวลเบสิก 6.0",บริษัท ไมโครซอฟท์เพรส จำกัด.

[6] พนัส ปัทมาสราวุธ,"Tip and Trick สร้างลูกเล่นด้วยVisual Basic",บริษัท ซีเอ็ดยูเคชั่น จำกัด.

[7] เรืองไกร รังสิพล,"เจาะระบบ TCP/IP จุดอ่อนของโปรโตคอลและวิธีป้องกัน",บริษัท โปรวิชัน จำกัด.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้