สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

กล้องรักษาความปลอดภัยผ่านคู่สายโทรศัพท์

SECURITY CAMERA VIA TELEPHONE LINE

โดย

นายณรงค์ศักดิ์ ต้นสี

นายอนุชา เดชพรหม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่ 5 49 ต้น ส

เลขทะเบียน 054964

วัน,เดือน,ปี - 4 ต.ค. 2548

กล้องรักษาความปลอดภัยผ่านคู่สายโทรศัพท์

SECURITY CAMERA VIA TELEPHONE LINE

โดย

นายณรงค์ศักดิ์ ต้นสี 44015057

นายอนุชา เดชพรหม 44015091

อาจารย์ที่ปรึกษา

อ.สุรพล บุญจันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องรักษาความปลอดภัยผ่านคู่สายโทรศัพท์

SECURITY CAMERA VIA TELEPHONE LINE

โดย นายณรงค์ศักดิ์ ต้นสี 44015057
นายอนุชา เดชพรหม 44015091

บทคัดย่อ

โครงงานนี้เป็นการเขียนโปรแกรมประยุกต์เพื่อทำให้เครื่องคอมพิวเตอร์ที่ติดตั้งกล้องดิจิตอลไว้ให้ทำหน้าที่รักษาความปลอดภัยในสถานที่หรือจุดที่ไม่มีคนดูแล ในโปรแกรมนี้จะทำให้คอมพิวเตอร์ที่อยู่นอกสถานที่สามารถติดต่อกับเครื่องที่ติดตั้งกล้องซึ่งอยู่ในสถานที่เพื่อที่จะทำการรับภาพที่กล้องได้ถ่ายไว้และสามารถบันทึกภาพจากเครื่องที่ติดต่อเข้ามาได้ โดยในโปรแกรมต่างๆ เหล่านี้ได้ใช้โปรแกรมเดลไฟล์ 5 เป็นเครื่องมือสำหรับพัฒนาโปรแกรม

Abstract

This project present a software for use computer and camera to secure the area. This program can use by another computer to connect the server for receive and record the picture files. This software was wrote by Delphi 5 for program development.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	3
2.1 การสื่อสารอนุกรม	3
2.1.1 การส่งข้อมูลแบบอนุกรม	3
2.1.2 การสื่อสารแบบ Synchronous และ Asynchronous	3
2.1.3 มาตรฐานการเชื่อมต่อแบบ RS - 232 - C	4
2.1.4 พอร์ตสื่อสารอนุกรมของเครื่อง PC	7
2.1.5 รหัส ASCII	8
2.2 แบบจำลองของ OSI (The OSI Reference model)	9
2.2.1 ฟิสิกส์คัลเลเยอร์ (Physical layer)	10
2.2.2 ดาต้าลิงค์เลเยอร์ (Data link layer)	11
2.2.3 เน็ตเวิร์คเลเยอร์ (Network layer)	11
2.2.4 ทรานส์พอร์ตเลเยอร์ (Transport layer)	12
2.2.5 เซสชั่นเลเยอร์ (Session layer)	13
2.2.6 พรีเซนเตชั่นเลเยอร์ (Presentation layer)	13
2.2.7 แอพพลิเคชั่นเลเยอร์ (Application layer)	14
2.3 TCP/IP	14
2.3.1 ส่วนประกอบของ TCP/IP	15
2.3.2 โครงสร้างของชุดโปรโตคอล TCP/IP	17
2.3.3 ข้อแตกต่างระหว่างชุดโปรโตคอล TCP/IP กับ OSI Model	18
2.3.4 ลักษณะของการติดต่อ	19
2.4 พอร์ตและซ็อกเก็ต (TCP Port and Socket)	19
2.4.1 ข้อกำหนดรูปแบบ IP (Internet Protocol)	20
2.4.2 ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol)	21
2.4.3 ซ็อกเก็ต (Socket)	21
2.4.3.1 เอพีไอ (API) ข้อกำหนดในการติดต่อสื่อสาร	21
2.4.3.2 ชนิดของซ็อกเก็ต	22
2.5 การสื่อสารข้อมูลด้วยระบบโมเด็ม	24
2.5.1 ประเภทชุดคำสั่ง AT	26
2.5.2 การออนไลน์และออฟไลน์	26
2.5.3 รูปแบบของชุดคำสั่ง AT	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

2.5.4 การสนองคำสั่งของโมเด็ม 27

2.5.5 ชุดคำสั่งเพิ่มเติม 28

2.5.6 การตั้งค่าในรีจิสเตอร์ S 28

2.5.7 การส่งหลายคำสั่งในครั้งเดียว 28

2.5.8 การเก็บค่าต่าง ๆ ที่เซตเอาไว้ใน Profile 29

2.5.8.1 การสร้างและการเรียกใช้ User Profile 29

2.5.8.2 การเรียกดูข้อมูลที่ตั้งไว้ทั้งหมด 30

2.5.8.3 การเก็บหมายเลขโทรศัพท์ไว้ในโมเด็ม 31

2.5.9 การเชื่อมต่อกับ RS-232-C 31

2.5.9.1 Data Terminal Ready 31

2.5.9.2 Carrier Detect 32

2.5.10 การควบคุมการไหล 32

2.5.10.1 การควบคุมการไหลแบบฮาร์ดแวร์ 32

2.5.10.2 การควบคุมการไหลแบบซอฟต์แวร์ 33

2.6 สัญญาณภาพ 33

2.6.1 มาตรฐานวิดีโอ 34

2.6.2 Windows Media Player MPEG-4 36

บทที่ 3 การคำนวณและการสร้าง 37

3.1 โปรแกรม Project3TServer 37

3.1.1 Component ที่สำคัญในโปรแกรม Project3Tserver 37

3.1.2 การทำงานของโปรแกรม Project3TServer แต่ละส่วน 39

3.2 โปรแกรม Project3TClient 40

3.2.1 Component ที่สำคัญในโปรแกรม Project3TClient 40

3.2.2 การทำงานของโปรแกรม Project3TClient แต่ละส่วน 43

บทที่ 4 การทดลองและผลการทดลอง 44

4.1การเตรียมการใช้งานของเครื่องคอมพิวเตอร์ก่อนการใช้งานโปรแกรมที่สร้างขึ้นมา 45

4.1.1 การเตรียมการใช้งานของเครื่องแม่ 45

4.1.2 การติดตั้งและวิธีการใช้งานโปรแกรม Project3TSever 47

4.2 การติดตั้งและการใช้งานของเครื่องลูก (Client) 52

บทที่ 5 บทวิจารณ์และบทสรุป 58

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

บทที่ 2 รูป 2.1 รูปแสดงการต่ออุปกรณ์สื่อสารผ่านทางพอร์ตอนุกรม RS – 232 – C 5

รูป 2.2 สายนำสัญญาณที่ใช้ในมาตรฐาน RS – 232 – C 6

รูป 2.3 แผนภาพการเชื่อมต่อของชิพควบคุมการสื่อสารเบอร์ 8250 บนเครื่อง PC ทั่วไป 9

รูป 2.4 โครงสร้างสถาปัตยกรรมของเน็ตเวิร์ค ที่ใช้ในหนังสือเล่มนี้วางอยู่บนแบบจำลอง OSI 11

รูป 2.5 แสดงการเปรียบเทียบระหว่างซ็อกเก็ตและ SUN RPC application programing 23

รูป 2.6 แสดงลักษณะทั้งสามแบบของซ็อกเก็ต 24

รูป 2.7 แสดงโครงสร้างของข้อมูลที่ใช้ในการกำหนดซ็อกเก็ตแอดเดรส 25

รูป 2.8 Profile ซึ่งแสดงข้อมูลที่เก็บไว้เรียกใช้โดยคำสั่ง AT & V 31

บทที่ 3 รูป 3.1 บล็อกไดอะแกรมของกล้องรักษาความปลอดภัยผ่านคู่สายโทรศัพท์ 37

รูป 3.2 แสดงแผนผังการทำงานของโปรแกรม Project 3Tserver 38

รูป 3.3 แสดงแผนผังการทำงานของโปรแกรม Project 3TClient 41

บทที่ 4 รูปที่ 4.1 โปรโตคอล TCP/IP 44

รูปที่ 4.2 Incoming Connections 45

รูปที่ 4.3 การตอบรับ Incoming Connection 46

รูปที่ 4.4 Status ของ Incoming Connection 46

รูปที่ 4.5 การติดตั้งโปรแกรม Project3TSever 47

รูปที่ 4.6 หน้าจอหลักของโปรแกรมเมื่อยังไม่มีการเชื่อมต่อ 48

รูปที่ 4.7 หน้าจอหลักของโปรแกรมเมื่อมีการเชื่อมต่อ 48

รูปที่ 4.8 การสร้างรหัสเพื่อควบคุมการใช้งาน 49

รูปที่ 4.9 หน้าจอควบคุมการใช้งานหลังจากสร้างรหัสแล้ว 49

รูปที่ 4.10 หน้าจอควบคุมการใช้งานเมื่อมีการล็อก 50

รูปที่ 4.11 หน้าจอควบคุมการส่งภาพ 51

รูปที่ 4.12 หน้าจอการเซ็ตค่าของภาพ 51

รูปที่ 4.13 การติดตั้งโปรแกรม Project3TClient 52

รูปที่ 4.14 หน้าจอหลักของเครื่องลูก 52

รูปที่ 4.15 หน้าจอโมเด็ม 53

รูปที่ 4.16 หน้าจอพอร์ตและการควบคุม 54

รูปที่ 4.17 หน้าจอแสดงภาพก่อนที่จะรับภาพ 55

รูปที่ 4.18 หน้าจอแสดงภาพเมื่อมีการรับภาพ 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูปที่ 4.19 หน้าจอแสดงภาพเมื่อเกิดการเปลี่ยนแปลงของภาพ 56
รูปที่ 4.20 หน้าจอแสดงภาพหลังจากเก็บภาพความเคลื่อนไหวไว้ได้ 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
บทที่ 2	ตารางที่ 2.1 โครงสร้างชุดโปรโตคอล TCP/IP เปรียบเทียบกับ OSI Model	17
	ตารางที่ 2.2 หมายเลขพอร์ตที่ให้บริการต่าง ๆ	20
	ตารางที่ 2.3 กระบวนการทำงานของโมเด็มต้นทางและปลายทางตั้งแต่ขั้นตอนที่ 1-12	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องด้วยการให้ความช่วยเหลือเอื้อเฟื้อในทุกๆ ด้านจากอาจารย์ที่ปรึกษา จึงขอขอบคุณท่านอาจารย์สุรพล บุญจันทร์ ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่คอยให้คำปรึกษาและให้ความสนับสนุน รวมทั้งอาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม และเจ้าหน้าที่ที่เกี่ยวข้องทุกท่าน ซึ่งได้ให้ความสะดวกในการใช้สถานที่ตลอดจนวัสดุอุปกรณ์ เครื่องมือ และอื่นๆ ในการทำปริญญานิพนธ์ รวมทั้งเพื่อนๆพี่ๆ ทุกคนที่คอยให้คำแนะนำ และให้กำลังใจตลอดมา

สุดท้ายนี้กราบขอบพระคุณ บิดา มารดา ที่ให้ความรัก ความห่วงใย ความปรารถนาดี ให้การสนับสนุนด้านการศึกษามาตั้งแต่เด็กตลอดมาจนถึงทุกวันนี้

นาย ณรงค์ศักดิ์ ต้นสี

นาย อนุชา เดชพรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

หลักการและเหตุผล

ในยุคปัจจุบันคอมพิวเตอร์ PC ได้มีการใช้งานอย่างแพร่หลาย เนื่องจากราคาที่ไม่แพงและ Softwear ที่ใช้งานได้หลายอย่างเช่น เล่นเกมส์ ดูหนัง ฟังเพลง และอินเตอร์เน็ต การติดต่อกับโลกภายนอกโดยมีคอมพิวเตอร์และโมเด็ม (Modem) เป็นตัวจัดการ

โทรศัพท์การสื่อสารที่เร็วที่สุด การส่งข่าวสารได้ในขณะนั้นโดยเสียงตามสายการพูดคุยกันในที่ห่างไกล ประโยชน์ของโทรศัพท์ที่มีโครงข่ายทั่วโลกไม่ได้มีแค่การพูดคุยกันเท่านั้นแต่ยังมีการส่งข้อมูลที่เป็นดิจิตอลได้โดยใช้โมเด็มเป็นตัวกลางในการเชื่อมต่อระหว่างคอมพิวเตอร์กับโครงข่ายโทรศัพท์ ตัวอย่างเช่น การใช้อินเตอร์เน็ต

การสื่อสารข้อมูลของคอมพิวเตอร์ปกติการสื่อสารข้อมูลระหว่างคอมพิวเตอร์วิธีที่ง่ายที่สุดคือการ รับ – ส่ง ผ่านสายเคเบิลโดยตรงถ้าระยะทางไม่ห่างมากนักหรืออาจ รับ – ส่ง ผ่านเครือข่ายของคอมพิวเตอร์เองเช่น LAN (Local Area Network) , WAN (Wide Area Network) หรือ PDN (Public Data Network)ซึ่งเป็นการ รับ – ส่ง ข้อมูลระหว่างกันโดยผ่านโมเด็ม ไปตามสายโทรศัพท์ทำให้สามารถส่งข้อมูลไปได้ไกลเท่าที่ระบบโทรศัพท์จะเข้าไปถึง

ปัจจุบันโครงข่ายโทรศัพท์ก็ครอบคลุมพื้นที่และมีใช้ได้ทั่วไปอีกทั้งคอมพิวเตอร์ก็สามารถที่จะทำการส่งข้อมูลผ่านคู่สายโทรศัพท์ได้โดยผ่านโมเด็ม เราก็สามารถที่จะดูภาพของกล้องวีดีโอที่ติดตั้งไว้ในห้อง หรือ ในอาคารต่างๆ จากปลายทางที่ไกลๆ ที่ซึ่งมีโทรศัพท์เข้าไปถึงได้โดยมีคอมพิวเตอร์และโมเด็มเป็นอุปกรณ์

วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการเชื่อมต่อกันระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องผ่านโมเด็ม
2. เพื่อศึกษาโปรแกรมการจัดการกับเครื่องคอมพิวเตอร์ในการส่งข้อมูล
3. เพื่อที่จะสามารถรับภาพผ่านโครงข่ายโทรศัพท์จากปลายทางที่ซึ่งติดตั้งกล้องไว้เพื่อดูแลรักษาความปลอดภัย
4. เพื่อเป็นแนวทางในการพัฒนาโปรแกรมให้รักษาความปลอดภัยโดยการออกคำสั่งผ่านเครือข่ายโทรศัพท์

ขอบเขตของโครงงาน

การสร้างและการศึกษาโครงงานนี้เพื่อที่จะทำการส่งข้อมูลภาพไปตามสายโทรศัพท์โดยผ่านโมเด็มและมีคอมพิวเตอร์เป็นอุปกรณ์การส่งข้อมูล ในโครงงานนี้ได้ใช้โปรแกรมเดลไฟล์ 5 (Delphi 5) เป็นตัวพัฒนาโปรแกรม ข้อมูลที่ทำการส่งไปตามสายโทรศัพท์คือภาพจากกล้องดิจิตอล โดยทำการติดตั้งโปรแกรมไว้ในคอมพิวเตอร์ต้นทางและปลายทาง โดยทางต้นทางต่อไปนี้จะเรียกว่าเครื่องแม่ (Sever) และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปลายทางเรียกว่าเครื่องลูก (Client) โดยที่เครื่องแม่จะติดตั้งกล้องดิจิตอลไว้แล้วเปิดโปรแกรมไว้โดยไม่มีใครดูแลและเครื่องลูกมีหน้าที่ในการติดต่อจากภายนอกเข้ามาทางโมเด็มผ่านคู่สายโทรศัพท์แล้วเปิดโปรแกรมเพื่อรับภาพจากเครื่องแม่

ประโยชน์ที่ได้รับจากโครงการ

1. เพื่อที่จะรักษาความปลอดภัยในสถานที่ซึ่งไม่มีใครดูแล
2. เพื่อบันทึกภาพจากสถานที่โดยที่อยู่นอกสถานที่นั้นได้
3. เพื่อพัฒนาโปรแกรมสำหรับการใช้งานที่มากกว่านี้ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

2.1 การสื่อสารแบบอนุกรม

2.1.1 การส่งข้อมูลแบบอนุกรม

ข้อมูลที่ถูกส่งออกไปแบบอนุกรมนั้น โดยปกติแล้วจะเป็นข้อมูล "0" หรือ "1" เรียงต่อกันเป็นชุด ๆ ซึ่งปกติแล้วข้อมูลที่ใช้สื่อสารกันทั่วไป จะประกอบไปด้วยข้อมูลจำนวนแปดบิตต่อหนึ่งตัวอักษร โดยข้อมูลแต่ละชุดนี้จะสามารถแทนตัวอักษรที่แตกต่างกันได้ถึง 256 แบบ ในเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer : PC) ข้อมูลจะถูกเก็บอยู่ในรูปของข้อมูลขนาดแปดบิตหรือหนึ่งไบต์อยู่แล้ว ไม่ว่าจะใช้ระบบปฏิบัติการ DOS, OS/2 หรือใช้เครื่อง PC ชนิด 32 บิต ก็ตาม

ข้อมูลบางแบบนั้นอาจจะมีขนาดเพียงเจ็ดบิต อย่างเช่นรหัส ASCII ที่ใช้กันในคอมพิวเตอร์ทั่ว ๆ ไปนั้น ก็เป็นรหัสขนาดเจ็ดบิต ดังนั้น มันจึงใช้แทนตัวอักษรได้เพียง 128 ตัวเท่านั้น ยกตัวอย่างเช่น ตัวอักษร A นั้น จะมีรหัส ASCII ที่ใช้แทนมัน คือ รหัสหมายเลข 65

ในการส่งข้อมูลผ่านพอร์ตแบบอนุกรมนั้น ข้อมูลแต่ละไบต์ จะประกอบด้วย

1. บิตเริ่มต้น (start bit) 1 บิต
2. บิตข้อมูล (data bit) 7 หรือ 8 บิต
3. พาริตี้บิต (parity bit) จะมีหรือไม่ก็ได้
4. บิตสิ้นสุด (stop bit) 1 หรือ 2 บิต

ข้อมูลแบบอนุกรมจะถูกส่งไปตามสายนำสัญญาณโดยทยอยส่งออกไปทีละบิต ซึ่งบิตต่ำสุดจะถูกส่งออกไปก่อน แล้วติดตามด้วยบิตที่สูงขึ้นมาเรื่อย ๆ จนครบชุดข้อมูลหนึ่ง ๆ สมมติว่าถ้าต้องการที่จะส่งตัวอักษร A ออกไปแบบอนุกรมซึ่งตัวอักษร A นั้นมีรหัส ASCII ที่เป็นเลขฐานสอง คือ 01000001 ดังนั้น บิตที่ถูกส่งออกไป ก็คือ บิตที่อยู่ทางขวาสุดซึ่ง ก็คือ 1 นั่งเอง และหลังจากนั้นก็เป็น 0 ตัวถัดมาทางซ้ายเรียงต่อ ๆ กัน ตามลำดับไปจนครบข้อมูลหนึ่งชุด

2.1.2 การสื่อสารแบบ Synchronous และ Asynchronous

การสื่อสารของเครื่อง PC ส่วนใหญ่นั้นเป็นแบบ Asynchronous ซึ่งก็หมายความว่าไม่มีการกำหนดช่วงเวลาตายตัวที่ใช้ส่งข้อมูลหนึ่งตัวอักษรออกไปที่พอร์ตอนุกรม นั่นก็หมายความว่า การส่งข้อมูลแต่ละครั้งนั้นจะต้องมีบิตเริ่มและบิตหยุดเป็นตัวกำหนดขอบเขตของข้อมูล บิตที่แสดงขอบเขตนี้จะถูกสร้างขึ้นโดยอัตโนมัติจากวงจรสื่อสาร แต่สำหรับการสื่อสารแบบ Synchronous นั้น บิตเริ่มและบิตหยุดไม่มีความจำเป็น เนื่องจากว่า ได้กำหนดช่วงเวลาการส่งข้อมูลต่อหนึ่งตัวอักษรเอาไว้แล้ว

เมื่อ PC ได้ทำการเชื่อมต่อเครื่องคอมพิวเตอร์โฮสต์อื่น ๆ โมเด็มจะใช้การติดต่อแบบ Asynchronous เสมอ ส่วนการสื่อสารแบบ Synchronous นั้นจะใช้เฉพาะในงานพิเศษ ๆ ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

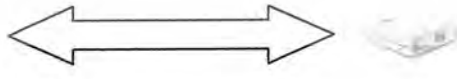
ความเร็วในการการส่งข้อมูลสูงมาก เช่น ในการส่งข้อมูลระหว่าง LAN หรือระบบเครือข่าย Packet swithching ซึ่งวงจรสื่อสารที่พบในเครื่อง PC ทั่วไปนั้นไม่สามารถทำงานในแบบ Synchronous ได้

การสื่อสารแบบ Synchronous นั้นมีประสิทธิภาพการทำงานที่สูงกว่าแบบ Asynchronous มาก แต่จะต้องการวงจรพิเศษที่ต้องใช้สำหรับการสื่อสารแบบ Synchronous โดยเฉพาะข้อมูลที่ส่งแบบ Synchronous นั้น จะไม่มีการขาดตอน โดยข้อมูลจะถูกแยกออกเป็นชุด ๆ จากสัญญาณนาฬิกาที่ส่งไปพร้อม ๆ กันในสายนำสัญญาณอีกเส้นหนึ่ง ดังนั้น จึงมีความเร็วในการส่งข้อมูลที่สูงมาก

2.1.3 มาตรฐานการเชื่อมต่อแบบ RS – 232 – C

การเชื่อมต่อแบบอนุกรมนั้นเป็นวิธีการเชื่อมต่อสื่อสารทั่วไปของอุปกรณ์ทางอิเล็กทรอนิกส์ต่าง ๆ องค์กรอุตสาหกรรมอิเล็กทรอนิกส์ของสหรัฐ (EIA) ซึ่งเป็นองค์กรที่เกี่ยวข้องโดยตรงกับอุตสาหกรรมอิเล็กทรอนิกส์ทั่วโลก ได้กำหนดมาตรฐานของการเชื่อมต่อและคุณสมบัติของพอร์ตอนุกรมสื่อสารขึ้นมา และมาตรฐานอันหนึ่งที่ได้รับการยอมรับโดย EIA คือ มาตรฐานที่ 232 แบบ C หรือเป็นที่รู้จักกันในชื่อของ RS – 232 – C (Regulation Standard No. 232 Type C) มาตรฐานการเชื่อมต่อแบบนี้มีความคล้ายคลึงกับมาตรฐาน CCITT V.24 และมีคุณสมบัติทางไฟฟ้าใกล้เคียงกับมาตรฐาน CCITT V.28 และนอกจากนั้นก็มีความใกล้เคียงกับมาตรฐาน Mil – Std – 188 C ของกระทรวงกลาโหมสหรัฐฯ ด้วย

มาตรฐาน RS – 232 – C นั้น แบ่งการเชื่อมต่อออกเป็น 2 ลักษณะ คือ การต่อกับเทอร์มินอล (DTE : Data Terminal Equipment) และการต่อกับอุปกรณ์สื่อสารข้อมูล (DCE : Data Communication Equipment) ซึ่งในกรณีปกตินั้น DCE จะต้องต่อเข้ากับ DTE เสมอ ยกตัวอย่างเช่น การต่อโมเด็มเข้ากับเครื่อง PC โดยเครื่อง PC จะเป็นอุปกรณ์ DTE และโมเด็มจะเป็นอุปกรณ์ DCE



คอมพิวเตอร์ (DTE)　　โมเด็ม (DCE)

รูป 2.1 รูปแสดงการต่ออุปกรณ์สื่อสารผ่านทางพอร์ตอนุกรม RS – 232 – C

จากรูปที่ 2.1 นั้น เป็นรูปที่แสดงการเชื่อมต่อระหว่างเครื่อง PC กับโมเด็มโดยผ่านทางพอร์ตอนุกรม RS – 232 – C โดยที่เครื่อง PC จะส่งและรับข้อมูลจากโมเด็ม การเชื่อมต่อตามมาตรฐาน RS –

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

232 – C นั้น โดยปกติจะใช้คอนเน็คเตอร์รูปตัว D ชนิด 25 ขา กำหนดให้ปลายสายสัญญาณด้านหนึ่งที่เป็นคอนเน็กเตอร์ตัวผู้ใช้ต่ออยู่กับอุปกรณ์ DCE และปลายของสายสัญญาณอีกด้านหนึ่ง จะต้องเป็นคอนเน็คเตอร์ตัวเมียใช้ต่ออยู่กับอุปกรณ์ DTE แต่ต่อมา บริษัท IBM นั้นก็ได้ออกแบบคอนเน็คเตอร์แบบ 9 ขา ขึ้นมาใช้แทนคอนเน็คเตอร์แบบ 25 ขา

รูป 2.2 สายนำสัญญาณที่ใช้ในมาตรฐาน RS – 232 – C

จากรูปที่ 2.2 แสดงสายนำสัญญาณที่ใช้ในมาตรฐาน RS – 232 – C นั้นจะมีทั้งหมด 9 เส้น ประกอบด้วย วงจรข้อมูล 2 วงจร คือ

วงจรชุดที่ 1 ประกอบด้วย

- ขานำสัญญาณหมายเลข 2 และ 3 ซึ่งจะใช้ในการส่งผ่านข้อมูลระหว่างเครื่อง PC และโมเด็ม
- ขานำสัญญาณหมายเลข 1 และ 7 เป็นกราวด์

วงจรชุดที่ 2 ประกอบไปด้วย

- ขานำสัญญาณหมายเลข 4 และ 5 เป็นวงจรควบคุมสัญญาณที่ เรียกว่า RTS และ CTS ซึ่งจะใช้ในการควบคุมการไหลของข้อมูลระหว่าง PC และโมเด็ม PC และ โมเด็ม (Hardware Flow Control)
- ขานำสัญญาณหมายเลข 8 เป็นสัญญาณ Carrier Detect (CD) ใช้บอก PC ว่าในขณะนั้นโมเด็มได้รับสัญญาณพาหะอยู่หรือไม่
- ขานำสัญญาณหมายเลข 20 เป็นสัญญาณ Data Terminal Ready (DTR) ใช้บอกโมเด็มว่าในขณะนั้นเทอร์มินอลพร้อมที่จะติดต่อกับโมเด็มหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขานำสัญญาณหมายเลข 22 จะใช้เป็นสัญญาณแสดงว่าในขณะนั้นโมเด็มได้รับสัญญาณกระดิ่งหรือไม่ ซึ่งมักจะไม่ได้ถูกใช้งานมากนัก

ในการสื่อสารแบบอนุกรมนั้น การที่จะทำให้อุปกรณ์อิเล็กทรอนิกส์ สื่อสารกันได้ จำเป็นจะต้องใช้สายไฟฟ้าเชื่อมต่อระหว่างอุปกรณ์ 2 ตัวไม่ต่ำกว่า 2 เส้น แต่จะเห็นว่ามาตรฐาน RS – 232 – C นั้นมีขานำสัญญาณถึง 25 ขา เหตุผลก็เพราะว่าอุปกรณ์สื่อสารแบบอนุกรมตามมาตรฐานนั้น จะต้องมีวงจรไฟฟ้าที่เกี่ยวข้องกับการรับ – ส่งข้อมูลอยู่ 3 วงจรด้วยกัน ก็คือ วงจรส่งข้อมูล วงจรรับข้อมูล และวงจรควบคุม ดังที่จะกล่าวถึงรายละเอียดแยกเป็นประเภทดังต่อไปนี้

1. วงจรส่งข้อมูล ทำหน้าที่ส่งข้อมูลจาก DTE ไป DCE
2. วงจรรับข้อมูล ทำหน้าที่รับข้อมูลจาก DCE เพื่อส่งไปให้ DTE

จากรูปที่ 2.2 จะเห็นว่าเครื่อง PC นั้นจะส่งข้อมูลไปให้โมเด็มทางขานำสัญญาณหมายเลข 2 และรับข้อมูลกลับจากโมเด็มทางขานำสัญญาณหมายเลข 3 โดยใช้ขานำสัญญาณหมายเลข 7 เป็นกราวด์ร่วมของทั้งสองวงจร

สัญญาณทางไฟฟ้าที่ขานำสัญญาณหมายเลข 2 , 3 และ 7

ตามมาตรฐาน RS – 232 – C รูปแบบของสัญญาณทางไฟฟ้าที่ใช้แทนข้อมูลนั้นเป็นรูปแบบสัญญาณง่าย ๆ คือ สัญญาณสี่เหลี่ยมที่ถูกสร้างจากไฟฟ้ากระแสตรงระดับแรงดันประมาณ 3 ถึง 25 โวลท์ แทนข้อมูล "0" และ -3 ถึง -25 โวลท์จะแทนข้อมูล "1" สำหรับช่วงแรงดัน -3 ถึง 3 โวลท์นั้นจะเป็นช่วงระดับแรงดันที่ใช้ในการแบ่งแยกระดับสถานะของสัญญาณระหว่างสถานะ "0" และสถานะ "1"

คุณสมบัติของสัญญาณแบบสี่เหลี่ยมนี้ เมื่อถูกส่งออกมาในสายไฟฟ้าแบบธรรมดา มักจะเกิดความผิดเพี้ยนได้ง่าย เนื่องจากข้อจำกัดของสายนำสัญญาณ หรืออาจจะมีสาเหตุมาจากสัญญาณรบกวนจากภายนอกอื่น ๆ ทำให้การรับส่งข้อมูลมีระยะใช้งานอยู่ในช่วงที่จำกัด โดยเฉพาะการรับส่งข้อมูลที่ความเร็วสูงนั้น คลื่นสี่เหลี่ยมจะเรียงชิดติดกัน ยิ่งทำให้มีโอกาสเกิดความผิดเพี้ยนได้ง่าย ดังนั้น มาตรฐาน RS – 232 – C จึงได้กำหนดความยาวของสายนำสัญญาณสำหรับการใช้งานที่ความเร็วต่าง ๆ ไว้

วงจรควบคุมของ RS – 232 – C นั้นมีอยู่ทั้งหมด 5 วงจร ซึ่งแต่ละวงจรจะมีหน้าที่ในการสร้างสัญญาณควบคุมต่าง ๆ ขึ้นมาเพื่อทำให้เครื่อง PC และโมเด็มทราบสถานะในการทำงานของกันและกันว่าอยู่ในสถานะใดโดยที่สัญญาณในวงจรควบคุมนั้นจะมีลักษณะทางกายภาพร่วมกันกับสัญญาณที่ปรากฏอยู่ในวงจรข้อมูลแต่โมเด็มส่วนใหญ่นั้นไม่ได้ใช้วงจรควบคุมครบทุกวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Request to Send และ Clear to Send (ขานำสัญญาณหมายเลข 4 และ 5)

วงจร RTS และ CTS นั้นเป็นวงจรที่ใช้สำหรับการควบคุมการส่งผ่านข้อมูลระหว่างเครื่อง PC และโมเด็ม โดยโมเด็มจะส่งสัญญาณ CTS สถานะ "ON" ให้แก่ PC เมื่อโมเด็มพร้อมที่จะรับข้อมูล และส่งสัญญาณ CTS สถานะ "OFF" เมื่อโมเด็มนั้นไม่พร้อมที่จะรับข้อมูล

ในทำนองเดียวกันเครื่อง PC ก็จะให้สัญญาณ RTS สถานะ "ON" เมื่อ PC พร้อมที่จะรับข้อมูล และส่งสัญญาณ RTS สถานะ "OFF" เมื่อ PC ไม่พร้อมที่จะรับข้อมูลจากโมเด็ม สัญญาณ RTS และ CTS นี้จะมีประโยชน์มากในการสื่อสารข้อมูลด้วยความเร็วสูง โดยสามารถป้องกันการล้นและการสูญหายของข้อมูลได้เป็นอย่างดี

Data Terminal Ready (ขานำสัญญาหมายเลข 20) และ Data Set Ready (ขานำสัญญาณหมายเลข 6)

สัญญาณ DTR จะใช้เป็นสิ่งที่บอกโมเด็มให้ทราบว่าเครื่อง PC นั้นกำลังอยู่ในสถานะที่พร้อมที่จะติดต่อสื่อสารกับโมเด็มหรือไม่ และในกรณีเดียวกันสัญญาณ DSR ก็จะใช้เป็นสิ่งที่บอกให้เครื่อง PC ทราบว่าโมเด็มก็พร้อมที่จะติดต่อกับ PC หรือไม่ โดยที่สัญญาณ DSR นั้นจะอยู่ในสถานะ "ON" ก็ต่อเมื่อโมเด็มได้รับสัญญาณ DTR แล้ว

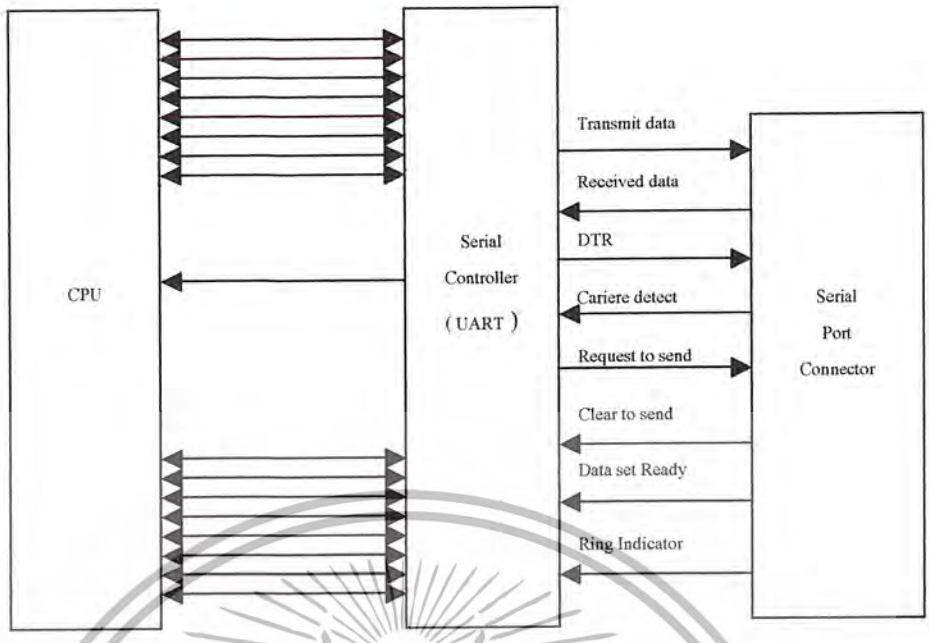
Carrier Detect (ขานำสัญญาณหมายเลข 8) และ Ring Indicator (ขานำสัญญาณหมายเลข 22)

สัญญาณ CD นี้จะใช้เป็นสิ่งที่บอก PC ให้ทราบว่าโมเด็มกำลังเชื่อมต่อกับโมเด็มเครื่องอื่น ๆ และกำลังได้รับสัญญาณพาหะจากโมเด็มปลายทาง ส่วนสัญญาณ RI นั้น จะเป็นการบอกเครื่อง PC ให้ทราบว่ามีสัญญาณกระดิ่งโทรศัพท์เรียกเข้ามาที่โมเด็ม ซึ่งโมเด็มส่วนใหญ่ในปัจจุบันก็มีวงจรที่ใช้ตอบรับโทรศัพท์โดยอัตโนมัติ (Auto – answer) อยู่ภายในตัวเองแล้ว จึงไม่จำเป็นที่จะต้องใช้งานสัญญาณ RI นี้

2.1.4 พอร์ตสื่อสารอนุกรมของเครื่อง

การอินเตอร์เฟส RS – 232 – C เข้ากับเครื่อง PC นั้นไม่ได้เฉพาะข้อกำหนดทางกายภาพของคอนเน็คเตอร์เท่านั้น ด้วยเหตุนี้เองการสื่อสารผ่านพอร์ตอนุกรมสื่อสารของ PC นั้น จำเป็นที่จะต้องมีชิพที่ใช้ในการควบคุมการสื่อสารข้อมูลโดยเฉพาะ ซึ่งชิพนี้อาจจะอยู่บนแผงวงจรหลัก หรือบนแผ่นวงจรควบคุมอุปกรณ์เอนกประสงค์ (Multi I/O Card) ก็ได้ ตัวชิพประเภทนี้จะทำหน้าที่อินเตอร์เฟสข้อมูลแบบอนุกรมเข้ากับระบบบัสข้อมูลของ PC ตามแผนภาพที่แสดงไว้ในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.3 แผนภาพการเชื่อมต่อของชิพควบคุมการสื่อสารเบอร์ 8250 บนเครื่อง PC ทั่วไป

เนื่องจากว่าชิพนี้จะเชื่อมต่ออยู่กับบัสของข้อมูลของ PC และวงจรสื่อสารแบบอนุกรมในแต่ละพอร์ตก็จะถูกกำหนดให้มีแอดเดรส I/O ประจำพอร์ต ดังนั้น จึงทำให้ CPU สามารถอ่านเขียนข้อมูลที่พอร์ตได้โดยตรงเมื่อ CPU ต้องการ

มาตรฐานของเครื่อง PC ทุกรุ่นจะกำหนดให้

- พอร์ต COM 1 ใช้แอดเดรสที่ 3F8H
- พอร์ต COM 2 ใช้แอดเดรสที่ 2F8H

ส่วนด้านสัญญาณขัดจังหวะนั้นจะกำหนดให้

- COM 1 ใช้ IRQ 4
- COM 2 ใช้ IRQ 3

2.1.5 รหัส ASCII

วิธีที่มีการกำหนดรหัสเลขฐานสองขึ้นมาใช้แทนตัวอักษรที่จะใช้งานในคอมพิวเตอร์ โดยตัวอักษรแต่ละตัวจะมีรหัสเลขฐานสองเฉพาะตัวของมัน ซึ่งมาตรฐานของรหัสเหล่านี้ที่เป็นที่นิยมใช้กันทั่วไปจะมีชื่อรหัสว่า รหัส ASCII (American Standard Code for Information Interchange) รหัส ASCII นั้นประกอบด้วยตัวอักษรและสัญลักษณ์ที่พิมพ์ได้ 96 ตัว และ 32 ตัวเป็นอักษรที่พิมพ์ไม่ได้ รวมเป็นจำนวนทั้งสิ้น 128 ตัวอักษร ซึ่งใช้ขนาดข้อมูลจำนวน 7 บิตพอดี (27) แต่ IBM ได้กำหนดชุดอักษรรหัส ASCII เพิ่มเติมขึ้นมาอีก 128 ตัว โดยใช้ข้อมูลขนาด 8 บิต การเกิดมาตรฐานของรหัส ASCII ขึ้นมานั้นทำให้เครื่องคอมพิวเตอร์และอุปกรณ์จากผู้ผลิตต่าง ๆ กัน สามารถติดต่อกันได้โดยไม่มีปัญหาที่รหัสจะไม่ตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานรหัส ASCII นั้น ถูกออกแบบขึ้นมาเพื่อใช้ทดแทนมาตรฐานเดิมที่มีชื่อว่า BAUDOT ซึ่งมีจำนวนเพียง 5 บิตต่อตัวอักษร ซึ่งใช้งานกันอยู่กับเครื่องโทรพิมพ์ แต่เนื่องจากการใช้ขนาดข้อมูลเพียง 5 บิตนี้เอง ทำให้ตัวอักษรที่สามารถใช้ได้จึงมีจำนวนได้เพียง 32 ตัวเท่านั้น

จากที่กล่าวมาแล้วว่ารหัส ASCII นั้นมีตัวอักษรที่ไม่สามารถพิมพ์ได้ 32 ตัว ซึ่งก็คือ ASCII หมายเลขตั้งแต่ 0 ถึง 31 เรียกว่า รหัสควบคุมนั่นเอง

2.2 แบบจำลองของ OSI (The OSI Reference model)

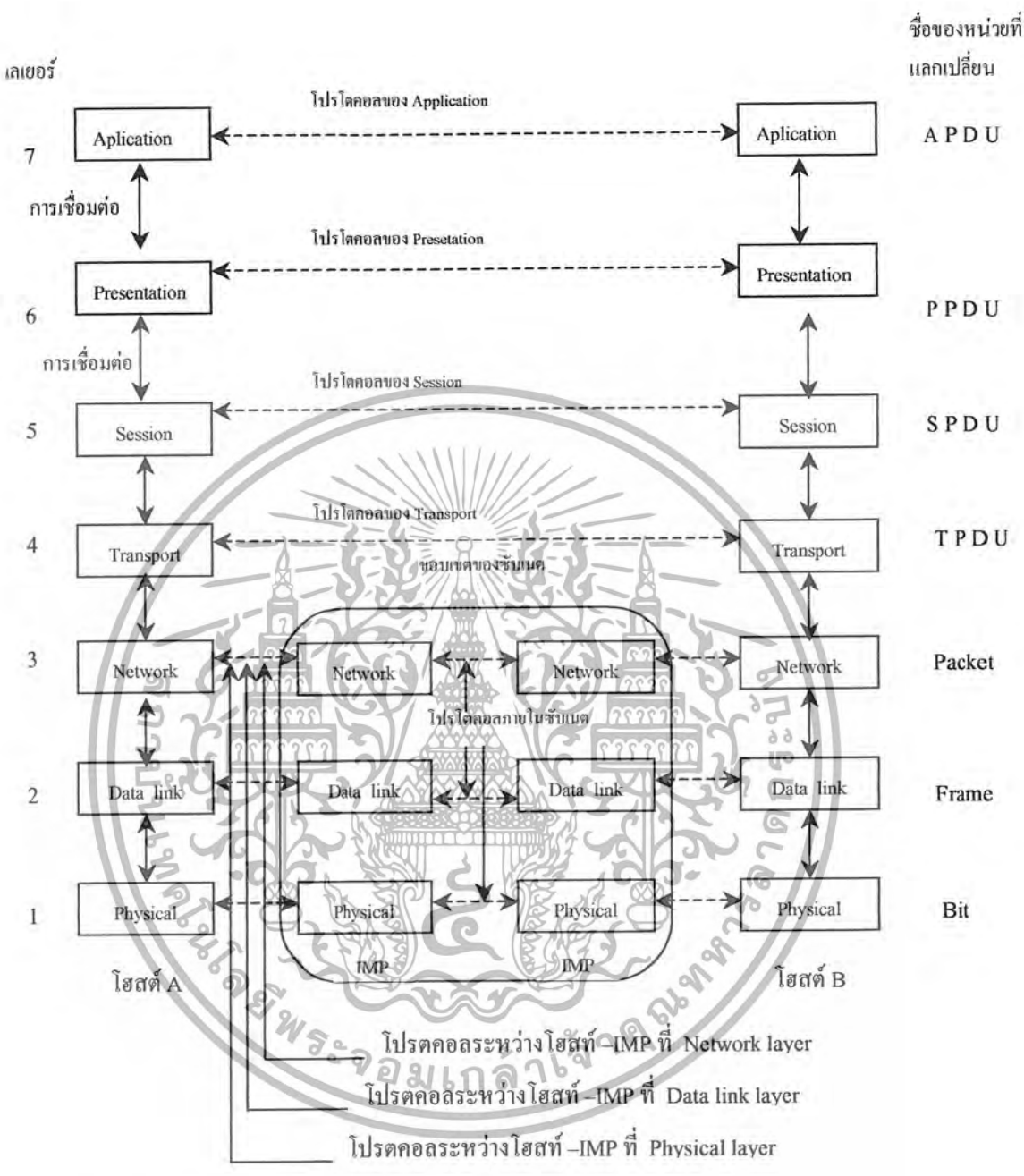
แบบจำลองนี้ถูกพัฒนาขึ้นโดย International Standard Organization (ISO) (องค์กรที่สร้างมาตรฐานระหว่างประเทศ) และแบบจำลองนี้มีชื่อเรียกว่า OSI (Open System Interconnection) ความหมายของชื่อ ก็คือ การเปิดกว้างสำหรับระบบใด ๆ ก็ตามที่ต้องการสื่อสารด้วยกัน ต่อไปจะเรียกว่า OSI เท่านั้น

OSI ประกอบด้วย 7 เลเยอร์หลักการสำหรับทั้ง 7 เลเยอร์ ก็คือ

1. เลเยอร์จะถูกกำหนดขึ้นมาเมื่อมีข้อแตกต่างด้านความนัย (abstraction)
2. แต่ละเลเยอร์จะมีการกำหนดการทำงานอย่างละเอียดและสมบูรณ์
3. แต่ละฟังก์ชั่นภายในเลเยอร์จะพยายามมุ่งไปสู่ระดับมาตรฐานของโปรโตคอล
4. ขอบเขตของเลเยอร์จะถูกเลือกและจำกัดให้มีปริมาณการเชื่อมต่อ (interface) ระหว่างเลเยอร์น้อยที่สุด
5. จำนวนของเลเยอร์จะต้องมากพอที่จะทำให้ฟังก์ชั่นที่จำเป็นและแตกต่างกันไม่อยู่ในเลเยอร์เดียวกันและเลเยอร์จะต้องไม่มากเกินไปจนทำให้โครงสร้างเทอะทะ

ในหัวข้อ 2.2.1 ถึง 2.2.7 จะกล่าวถึงรายละเอียดแต่ละเลเยอร์ ขอให้สังเกตว่าแบบจำลองของ OSI นั้นจะไม่ใช้เน็ตเวิร์คเพราะว่า OSI ไม่ได้เจาะจงในรายละเอียดที่แท้จริงของการให้บริการและโปรโตคอลที่จะใช้ในแต่ละเลเยอร์ OSI เพียงแต่บอกกว้าง ๆ ว่าแต่ละเลเยอร์ทำอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.4 โครงสร้างสถาปัตยกรรมของเน็ตเวิร์ค ที่ใช้ในหนังสือเล่มนี้วางอยู่บนแบบจำลอง OSI

2.2.1 ฟิสิกส์คัลเลเยอร์ (Physical layer)

ภายใน Physical layer จะเป็นการเกี่ยวพันกับการส่งข้อมูลดิบเป็นบิตผ่านช่องสื่อสารข้อมูล วิธีการออกแบบจะต้องทำให้แน่ใจได้ว่าข้อมูล "1" ที่ส่งออกไปและที่ปลายทางสามารถรับ "1" ได้ถูกต้อง คำถามที่เกี่ยวข้อง ก็คือ จะต้องใช้แรงดันเท่าใดสำหรับแทนเลข "1" และเท่าใดสำหรับแทนเลข "0" และแต่ละบิตจะต้องห่างกันเท่าใด การส่งข้อมูลสามารถทำแบบสองทิศทางได้หรือไม่ ก่อนจะเริ่มต้นส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลจะต้องทำให้เกิดการเชื่อมต่ออย่างไร และเมื่อส่งเสร็จแล้วจะยกเลิกการต่ออย่างไร ลึกลงไปอีกก็คือ คอนเน็คเตอร์ (Connector หรือ เป็นอุปกรณ์ที่ทำหน้าที่เป็นขั้วต่อของสายเข้ากับเครื่องคอมพิวเตอร์หรืออุปกรณ์สื่อสาร) ของเน็ตเวิร์คจะใช้กี่ขา และแต่ละขาทำหน้าที่อะไร จะเห็นว่าประเด็นที่ต้องสนใจจะเกี่ยวกับทั้งทางไฟฟ้า ทางรูปร่างภายนอกและก็ทั้งวิธีการเชื่อมต่อ รวมทั้งตัวกลางที่อยู่ใต้ Physical layer ซึ่งจะทำหน้าที่ส่งข้อมูลจริง ๆ ส่วนใหญ่แล้วขอบข่ายของ Physical layer จะเกี่ยวกับโดยตรงกับวิศวกรไฟฟ้า

2.2.2 ดาต้าลิงค์เลเยอร์ (Data link layer)

จุดประสงหลักของ Data link layer ก็คือ พยายามทำให้การส่งข้อมูลดิบเหมือนกับไม่มีความผิดพลาดเกิดขึ้น ทำให้เลเยอร์ที่สูงขึ้นไป (ซึ่งก็คือ network layer) นำไปใช้งานได้อย่างถูกต้อง วิธีการก็คือ ทางฝ่ายส่งจะทำการแตกข้อมูลออกเป็นก้อน ๆ เรียกว่า เฟรมข้อมูล (data – frame) (ปกติประมาณไม่กี่ร้อยไบต์) ทำการส่งเฟรมข้อมูลออกไปทีละชุดและรอรับการตอบรับ (acknowledge frame) ซึ่งตอบกลับมาโดยผู้รับ โดยปกติแล้ว physical layer จะไม่สนใจข้อมูลว่ามีความหมายใด จึงเป็นหน้าที่ของ Data link layer ที่จะต้องทำการสร้างและตรวจรับขอบเขตของเฟรม (frame boundary) ซึ่งสามารถทำได้โดยการเติมบิตเข้าไปในจุดเริ่มต้นและจุดสุดท้ายของเฟรม แต่จุดที่ต้องระวัง ก็คือ ถ้าหากข้อมูลที่เพิ่มเข้าไปซ้ำกับข้อมูลที่ต้องการส่งจริง ๆ

สัญญาณรบกวนจากภายนอกก็เป็นปัญหาที่อาจจะทำให้เฟรมขาดหายไปได้ ในกรณีนี้โปรแกรมที่ควบคุม Data link layer ที่เครื่องต้นทางจะส่งข้อมูลซ้ำมาใหม่ อย่างไรก็ตามการส่งเฟรมเดียวกันออกมาหลาย ๆ ครั้งก็อาจจะทำให้เกิดเฟรมซ้ำกันได้ วิธีการป้องกัน ก็คือ เฟรมที่ซ้ำกันจะส่งออกไปก็ต่อเมื่อมีการตอบรับ (acknowledgment frame) ส่งมาบอกว่าข้อมูลหายหรือถูกทำลายไปก่อน ภายใน Data link layer จะมีการเชื่อมต่อ (interface) หลาย ๆ แบบเพื่อให้ network layer สามารถเลือกใช้ได้

ข้อที่น่าสนใจอีกอย่างหนึ่ง ก็คือ ทำอย่างไรจึงจะให้ทางฝั่งที่ส่งข้อมูลซึ่งส่งเร็วกว่าทางฝั่งรับสามารถทำงานได้ วิธีการบางอย่างที่น่าสนใจก็คือ การระบายปริมาณการสื่อสารให้สม่ำเสมอ โดยการพักข้อมูลในบัฟเฟอร์ (buffer) ข้อมูลไว้ชั่วคราวแล้วค่อยส่งต่อออกไป

มีจุดน่าสนใจอีกอย่างก็คือ ถ้าหากว่าสายส่งสามารถส่งข้อมูลได้ทั้ง 2 ทางแล้ว จะก่อให้เกิดปัญหาทางด้านซอฟต์แวร์แก่ Data link layer ปัญหา ก็คือ เฟรมตอบรับจาก A ไปยัง B จะต้องแย่งชิงการใช้งานสายส่งกับเฟรมข้อมูล B ไปยัง A

2.2.3 เน็ตเวิร์คเลเยอร์ (Network layer)

ภายใน network layer จะเกี่ยวข้องกับการควบคุมการทำงานของซับเน็ต ประเด็นที่สำคัญ ก็คือ การพิจารณาว่าแพกเก็ตจะถูกส่งจากต้นทางไปยังปลายทางได้อย่างไร การหาเส้นทาง (route) อาจจะวางอยู่บนตารางที่คงที่อยู่และเชื่อมโดยตรงเข้ากับเน็ตเวิร์คและมีการเปลี่ยนตารางน้อยมาก การกำหนดเส้นทางจะกำหนดตอนเริ่มต้นติดต่อเลยก็ได้ หรืออาจจะใช้วิธีที่สามารถเปลี่ยนแปลงได้ตลอดเวลา (dynamic) ซึ่งเส้นทางที่แพกเก็ตเดินทางไปจะถูกกำหนดแบบแพกเก็ตต่อแพกเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามีจำนวนแพกเก็ตมากเกินไปภายในซับเน็ตจะทำให้เกิดปัญหาคอขวด คือ จำนวนแพกเก็ตเข้ามามากแต่ทางออกน้อย ภายใน network layer จะต้องจัดการกับปัญหาเหล่านี้ด้วย

เมื่อแพกเก็ตจะต้องเดินทางจากเน็ตเวิร์คหนึ่งไปยังอีกเน็ตเวิร์คหนึ่งจะทำให้เกิดปัญหาต่าง ๆ ขึ้น เช่น แอดเดรสของเน็ตเวิร์คที่ 2 อาจจะแตกต่างจากเน็ตเวิร์คที่ 1 ทำให้เน็ตเวิร์คที่ 2 ไม่สามารถรับแพคเก็ตนี้ได้เลย หรือโปรโตคอลที่ใช้อาจจะแตกต่างกัน network layer จะต้องจัดการกับปัญหาเหล่านี้เพื่อให้เน็ตเวิร์คทั้งหลายต่อถึงกันได้เสมือนเน็ตเวิร์คเดียวกัน

ในระบบเน็ตเวิร์คแบบกระจาย (broadcast network) วิธีการหาเส้นทางของแพกเก็ตจะง่ายมาก ทำให้ network layer มีขนาดเล็กหรืออาจจะไม่มีเลยก็ได้

2.2.4 ทรานส์พอร์ตเลเยอร์ (Transport layer)

หลักการทำงานของ transport layer ก็คือคอยติดต่อกับ session layer แยกข้อมูลให้มีขนาดพอเหมาะและส่งต่อให้ network layer พร้อมทั้งคอยตรวจสอบว่าข้อมูลได้ถูกส่งไปถึงปลายทางอย่างเรียบร้อยหรือไม่ ทั้งหมดนี้จะต้องทำให้ได้อย่างมีประสิทธิภาพเพื่อเป็นการแยกให้ session layer เป็นอิสระออกจากการเปลี่ยนแปลงทางด้านฮาร์ดแวร์

ภายใต้ภาวะปกติ transport layer จะสร้างคอนเน็คชั่นของเน็ตเวิร์คสำหรับแต่ละคู่ของ transport layer (กับอีกฝั่งหนึ่ง) แต่ถ้ามีการขอเพิ่มประสิทธิภาพการส่งข้อมูล transport layer ก็จะทำการสร้างการคอนเน็คชั่นของเน็ตเวิร์คขึ้นมาหลาย ๆ ชุดก็ได้ และทำการแบ่งข้อมูลออกส่งไปตามคอนเน็คชั่นของเน็ตเวิร์คต่าง ๆ หรืออีกแง่หนึ่งถ้าหากการสร้างคอนเน็คชั่นของเน็ตเวิร์คมีราคาแพง transport layer ก็จะใช้วิธีการมัลติเพล็กซ์หลาย ๆ คอนเน็คชั่นของ transport layer เข้าไปในคอนเน็คชั่นของเน็ตเวิร์คเดียว อย่างไรก็ตาม transport layer จะต้องทำงานไปพร้อม ๆ กันกับ session layer

นอกจากนี้ transport layer ยังต้องคอยดูว่าจะต้องให้บริการชนิดใดแก่ session layer ความต้องการของผู้ใช้เน็ตเวิร์คชนิดที่นิยมใช้กันแพร่หลาย ก็เช่น ไม่มีความผิดพลาดและจุดถึงจุด (error free point – to – point) ซึ่งจะส่งข้อมูลตามลำดับที่เข้ามาใน transport layer ชนิดอื่น ๆ ที่เป็นไปได้ก็มี เช่น ส่งข้อมูลที่เป็นอิสระต่อกันแต่ไม่มีการเรียงลำดับการรับและการส่ง แบบกระจายของข้อมูลสู่หลาย ๆ จุดปลายทาง ปกติแล้วชนิดของการให้บริการจะถูกกำหนดตอนที่สร้างคอนเน็คชั่น (connection establishment)

transport layer จะเป็นแบบจากจุดเริ่มถึงปลายทาง (source to destination) หรือปลายถึงปลาย (end – to – end) อย่างแท้จริง หรือกล่าวอีกอย่าง ซอฟท์แวร์บนเครื่องต้นทางจะต้องมีวิธีการในการใช้ส่วนหัว (header) และเมสเสจควบคุม (control message) เป็นภาษาเดียวกับเครื่องปลายทางเพื่อให้คุยกันรู้เรื่อง แต่ในระดับเลเยอร์ที่ต่ำลงมาโปรโตคอลที่ใช้ระหว่างกันและเครื่องที่อยู่ใกล้ ๆ กันจะไม่เป็นแบบจุดเริ่มถึงปลายทาง เพราะว่าถูกแยกด้วย IMP ก่อนจากรูปที่ 2.3 จะเห็นว่าเลเยอร์ 1 – 3 จะมีลักษณะเป็นห่วงคล้องกันเป็นชั้น ๆ ขณะที่เลเยอร์ 4 – 7 จะเป็นลักษณะปลายทางถึงปลายทาง (end – to – end)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเครื่องขนาดใหญ่ (host) มักจะมีลักษณะหลาย ๆ โปรแกรมใช้งานอยู่พร้อม ๆ กัน ซึ่งก็หมายความว่าจะมีหลาย ๆ คอนเน็คชั่นที่เข้ามาสู่และออกมาจากโฮสต์นี้ ทำให้จำเป็นต้องมีวิธีการในการบอกว่าข้อมูลก้อนใด ๆ เป็นของคอนเน็คชั่นใด วิธีการใส่ transport header ก็เป็นวิธีหนึ่งที่ใช้ได้

ในการมัลติเพล็กซ์หลาย ๆ ข้อมูลเข้าไปในช่องสัญญาณเดียว ภายใน transport layer จะต้องคอยรักษาการสร้างและยกเลิกคอนเน็คชั่นเอาไว้ คือ ระหว่างเน็ตเวิร์คจะต้องมีกลไกบางประการในการแจ้งชื่อเพื่อให้โปรเซสภายในเครื่องใดเครื่องหนึ่งมีทางที่จะบอกหรืออธิบายให้แก่โปรเซสใดก็ตามที่ต้องการติดต่อได้ นอกจากนี้ยังมีกลไกในการทำให้การไหลของข้อมูลมีความสม่ำเสมอ ทำให้โฮสต์ที่เร็วไม่สามารถทำงานได้เร็วกว่าเครื่องที่ช้า การควบคุมการไหลของข้อมูลระหว่างโฮสต์จะต่างจากการควบคุมการไหลของข้อมูลระหว่าง IMP ซึ่งจะกล่าวต่อไปภายหลัง

2.2.5 เซสชั่นเลเยอร์ (Session layer)

บน session layer จะยอมให้ผู้ใช้งานจัดตั้ง session ระหว่างเครื่อง กระบวนการ session นอกจากจะยอมให้มีการส่งข้อมูลเช่นเดียวกัน transport layer แล้ว ยังมีการเพิ่มการให้บริการที่ก้าวหน้ามากกว่าอีก เช่น ยอมให้ผู้ใช้งานเข้าไป (login) ยังเครื่องที่อยู่ระยะห่าง (remote system) หรือทำการถ่ายไฟล์ระหว่างเครื่อง

บริการอีกอย่างของ session layer ก็คือ การจัดการสนทนา (dialogue) session จะยอมให้การส่งผ่านได้ทั้งสองทางในเวลาเดียวกันหรือทิศทางเดียวในเวลาเดียว ถ้าเป็นการส่งในทิศทางเดียว session layer จะช่วยในการจัดทิศใดว่าถึงคราวฝ่ายใดจะส่งและรับ

มีบริการอื่นที่เกี่ยวข้องอีก ก็คือ การจัดการ โทเค้น (token) ในโปรโตคอลบางแบบมีความจำเป็นที่ทั้งสองฝั่งจะทำการสิ่งใด ๆ เหมือนกันในเวลาเดียวกันไม่ได้ จะต้องมีวิธีควบคุมการกระทำเหล่านี้ ซึ่ง session layer จะใช้โทเค้นเป็นตัวแลกเปลี่ยนระหว่างทั้งสองเครื่อง มีเพียงเครื่องที่ถือโทเค้นอยู่เท่านั้นจึงจะทำงานได้ก่อน

นอกจากนี้ยังมีบริการอีกอย่าง คือ การซิงค์โครไนซ์เซชั่น (synchronization) หรือ การทำให้ทั้งสองระบบทำงานสัมพันธ์กัน ลองพิจารณาปัญหาของการถ่ายไฟล์ระหว่าง 2 เครื่องที่ใช้เวลาร่วม 2 ชั่วโมง บนเน็ตเวิร์คที่มีค่าเฉลี่ยของการเสียหายประมาณ 1 ชั่วโมง หลังจากการถ่ายข้อมูลถูกหยุดกลางทาง กระบวนการถ่ายข้อมูลทั้งหมดจะต้องเริ่มต้นใหม่อีกครั้งและอาจจะเกิดการเสียหายของเน็ตเวิร์คได้อีก session layer จะจัดการปัญหานี้โดยการใส่จุดตรวจสอบเข้าในข้อมูล ทำให้หลังจากเกิดความเสียหายแก่เน็ตเวิร์คและกลับมาถ่ายข้อมูลใหม่ จะทำให้สามารถถ่ายข้อมูลต่อจากจุดเดิมได้ทันทีไม่ต้องไปเริ่มต้นใหม่

2.2.6 พรีเซนเตชั่นเลเยอร์ (Presentation layer)

บน presentation layer จะมีฟังก์ชั่นที่จะให้แก่ผู้ใช้สำหรับแก้ปัญหาในบางระดับ โดยที่ผู้ใช้ไม่ต้องเข้าไปแก้ปัญหาเอง ซึ่งจะต่างกับเลเยอร์ระดับล่าง ก็คือ ในระดับล่างจะสนใจเพียงการทำให้บิตของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลรับส่งได้อย่างถูกต้องเท่านั้น แต่ presentation layer จะสนใจในวากยสัมพันธ์ (syntax) และความเปลี่ยนแปลงในความหมายของคำ (semantic) ของข้อมูลที่ส่งผ่านด้วย

ตัวอย่างที่เห็นได้ง่าย ก็คือ การเข้ารหัสข้อมูล โดยปกติโปรแกรมของผู้ใช้งานจะไม่ได้ทำการแลกเปลี่ยนข้อมูลแต่เพียงข้อมูลดิบเท่านั้น จะมีข้อมูล เช่น ชื่อคน วันที่ จำนวนเงิน และใบเรียกเก็บเงิน ข้อมูลเหล่านี้จะถูกแทนด้วยข้อมูลแบบตัวอักษร แบบตัวเลข แบบเลขทศนิยม แต่ละเครื่องคอมพิวเตอร์จะมีรหัสการแทนข้อมูลแบบนี้ต่าง ๆ กัน เช่น ASCII บ้างหรือ EBCDIC เป็นต้น การที่จะทำให้คอมพิวเตอร์ที่มีการแทนรหัสที่แตกต่างกันสามารถสื่อสารกันได้ รหัสที่ใช้งานรวมทั้งโครงสร้างของข้อมูลจะต้องถูกกำหนดให้มีหลักการให้ตรงกัน การกำหนดและการเปลี่ยนไปมาจากรหัสที่แทนอยู่ภายในเครื่องไปเป็นรหัสที่มีการใช้เป็นมาตรฐานในเน็ตเวิร์คเป็นหน้าที่ของ presentation layer นั่นเอง

นอกจากนี้ยังทำหน้าที่อื่น ๆ เช่น การลดขนาดของข้อมูล (data compression) เพื่อเป็นการลดจำนวนบิตของข้อมูลที่จะใช้ส่งและยังอาจจะทำการเข้ารหัสเพื่อป้องกันไม่ให้มีการขโมยข้อมูลที่เป็นความลับ

2.2.7 แอพพลิเคชั่นเลยเยอร์ (Application layer)

ภายใน application layer จะประกอบด้วยโปรโตคอลชนิดต่าง ๆ มากมายที่มักจะเป็นที่ต้องการลองพิจารณาเทอร์มินอลชนิดต่าง ๆ ที่มีอยู่ในโลกนี้ แต่ละเครื่องจะมีลักษณะไม่เหมือนกัน มีลักษณะจอภาพขนาดไม่เท่ากัน มีการควบคุมการแสดงผลไม่เหมือนกัน (เช่น ESC sequence ใช้สำหรับแทรกหรือลบตัวอักษร เลื่อนเคอร์เซอร์ เป็นต้น)

วิธีทางหนึ่งที่จะแก้ปัญหานี้ ก็คือ กำหนดเทอร์มินอลเสมือนสำหรับเน็ตเวิร์ค (network terminal) ขึ้นมา ซึ่งเป็นตัวกลางที่โปรแกรมแก้ไขข้อความ (edit) หรือโปรแกรมอื่น ๆ จะติดต่อด้วย สำหรับเทอร์มินอลแต่ละแบบจะต้องมีโปรแกรมที่ตัวเทอร์มินอลเพื่อทำการแปลงรหัสของเทอร์มินอลเสมือน (virtual terminal) ไปเป็นการควบคุมเทอร์มินอลจริง ๆ เช่น เมื่อโปรแกรมแก้ไขข้อความ (edit) เลื่อนเคอร์เซอร์บนเทอร์มินอลเสมือน (virtual terminal) ไปยังมุมบนด้านซ้าย ตัวโปรแกรมที่เทอร์มินอลนี้จะต้องรับรู้และสามารถทำคำสั่งที่ถูกต้องเพื่อให้เคอร์เซอร์เลื่อนไปที่ตำแหน่งที่ถูกต้องจริง ๆ ลักษณะของโปรแกรมเช่นนี้จะเป็นลักษณะของ application layer

2.3 TCP/IP

ปัญหาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันคือ แต่ละเครือข่ายใช้คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อ และระบบปฏิบัติการณ์ที่ต่างกันมาตรฐานของ TCP/IP จึงถูกนำมาใช้เป็นกุญแจสำคัญในการแก้ปัญหาเหล่านี้ โดยกลายเป็นระบบที่สมบูรณ์แบบที่มีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่เครื่องพีซี จนถึงเมนเฟรม และไม่จำกัดระบบปฏิบัติการณ์ที่ใช้ TCP/IP จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน TCP/IP มากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP/IP คืออะไร

TCP/IP เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Networking Protocol)จัดทำเพื่อใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดก็ตามจะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้

TCP/IP เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดการโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่าง ๆ ที่กำหนดให้ทำเมื่อเกิดความผิดพลาดหรือต้องทำเพื่อป้องกันไม่ให้เกิดความผิดพลาด

TCP/IP เกิดจากการนำข้อกำหนดของรูปแบบต่าง ๆ กันมาใช้ร่วมกัน TCP และ IP ต่างก็เป็นรูปแบบหนึ่งของชุดนี้ (แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า TCP/IP) ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกันหรือต่างเครือข่ายก็ได้และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในตัวข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบเครือข่ายจะใช้ TCP/IP เป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ (Local Area Network) และเครือข่ายบริเวณกว้าง (Wide Area Network) ไม่ได้ใช้งานเฉพาะกับอินเตอร์เน็ตเท่านั้น

2.3.1 ส่วนประกอบของ TCP/IP

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่าง ๆ ซึ่งแบ่งเป็นกลุ่มได้ดังนี้

กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocals)

ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลระหว่างคอมพิวเตอร์สองเครื่องแบ่งย่อยออกได้เป็นสองชนิดคือ

1. TCP (Transmission Control Protocol) เป็นการบริการแบบ Connection Based Service ซึ่งคอมพิวเตอร์ด้านผู้รับและผู้ส่งต้องต่อถึงกันอยู่ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบก็คล้ายกับโทรศัพท์ที่ต้องติดต่อกันให้ได้ก่อนจะพูดคุยกันได้
2. UDP (User Datagram Protocol) เป็นการให้บริการแบบ Connectionless Oriented คอมพิวเตอร์ด้านผู้ส่งไม่จำเป็นต้องติดต่อกับด้านผู้รับก่อน เพียงรู้ที่อยู่ของด้านผู้รับแล้วใส่ที่อยู่นั้นไปกับข้อมูลที่ส่งออก ข้อมูลจะเดินทางตามเส้นทางต่าง ๆ เพื่อไปถึงปลายทางตามที่อยู่ คล้ายกับการส่งจดหมายที่ไปรษณีย์ จะส่งตามที่อยู่ที่จ่าหน้าซองจดหมาย โดยผู้ส่งและผู้รับไม่ต้องติดต่อถึงกัน

กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol)

ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือขนาดใหญ่กลุ่มข้อมูลรูปแบบนี้ก็จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสมแล้วส่งออกไป เมื่อถึงผู้รับปลายทางกลุ่มข้อมูลนี้ก็จะทำหน้าที่ตรงข้ามคือ รวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผล กลุ่มข้อกำหนดรูปแบบกลุ่มนี้ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. IP (Internet Protocol) เป็นการกำหนดรูปแบบการส่งข้อมูล
2. ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับสถานะของ IP เช่น ข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ในเครือข่าย
3. RIP (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับทำการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่เหมาะสมกับข้อมูลมากที่สุด
4. OSPF (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากเส้นทางที่สั้นที่สุดก่อน

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address)

ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและคอมพิวเตอร์ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตามเพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยไม่ว่าเครือข่ายจะใหญ่โตสักเพียงใดหรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตาม ที่อยู่จะต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบกลุ่มนี้มีดังนี้

1. ARP (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน
2. DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะการใช้งานจริงนั้นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้
3. RARP (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เช่นเดียวกับ ARP แต่ทำตรงข้ามกัน

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocol)

และสนับสนุนข้อมูลสถานะเพื่อนำไปใช้เลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1. EGP (Exterior gateway Protocol) ข้อกำหนดรูปแบบนี้จะทำการถ่ายโอนข้อมูลเส้นทางระหว่างเกตเวย์ กับเครือข่ายนอกเพื่อทำการสื่อสาร
2. GGP (Gateway-to-Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์
3. IGP (Interior Gateway Protocol) ข้อกำหนดรูปแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services)

ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรงข้อกำหนดรูปแบบนี้ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

1. BOOTP (BOOT Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายให้เริ่มทำงานข้อกำหนดรูปแบบนี้จะอ่านโปรแกรมควบคุมการทำงานจากคอมพิวเตอร์ให้บริการ(Server Computer) มาให้
2. FTP (File Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนไฟล์ข้อมูลระหว่างคอมพิวเตอร์ซึ่งอาจจะอยู่บนเครือข่ายเดียวกันหรือต่างเครือข่ายก็ได้
3. TELENET เป็นข้อกำหนดรูปแบบที่ให้บริการเกี่ยวกับการควบคุมการติดต่อระยะไกล

3.1(Remote Logic)

กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญ ๆ จัดทำไว้บนเครือข่ายที่สนใจมีดังนี้

3.1.1 NFS (Network File System) เป็นข้อกำหนดรูปแบบที่ทำให้ผู้ใช้คอมพิวเตอร์เครื่องหนึ่งสามารถเข้าไปดูไฟล์ข้อมูลและใช้งานไฟล์ข้อมูลซึ่งอยู่ในคอมพิวเตอร์เครื่องอื่นได้

3.1.2 NIS (Network Information Services) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Logins และ Password

3.1.3 RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้กับการควบคุมระยะไกล

SMTP (Simple Mail Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนจดหมายอิเล็คทรอนิกส์ (Electronic Mail) ระหว่างคอมพิวเตอร์

4. SNMP (Simple Network Management Protocol) ข้อกำหนดรูปแบบที่ให้บริการข่าวสารต่างๆ ที่แสดงสถานะของเครือข่ายและอุปกรณ์ที่ต่ออยู่บนเครือข่าย

2.3.2 โครงสร้างของชุดโปรโตคอล TCP/IP

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

ตาราง 2.1 โครงสร้างของชุดโปรโตคอล TCP/IP เปรียบเทียบกับ OSI Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54964

1. Application Layer ในชั้นนี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้ในเครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูล (File-Transfer Program) และอาจกล่าวได้ว่าโปรโตคอล TCP/IP ก็คือ โปรโตคอลในชั้นแอพพลิเคชั่นรวมกับชั้นพรีเซนเตชั่นของ OSI โมเดลนั่นเอง
2. Transport Layer ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายเปรียบเทียบได้กับชั้นเซสชั่นรวมกับทรานสปอร์ตเลเยอร์นั่นเอง โดยโปรโตคอล TCP/IP มีซ็อกเก็ต (Socket) เป็นจุดปลาย (End-Point) ในการสื่อสาร ซึ่งซ็อกเก็ตนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์ และหมายเลขพอร์ต (Port) ของเครื่องที่ต้องการส่งข้อมูลไปถึง ในชั้นนี้มีการรับรองให้ถึงที่หมาย และลำดับข้อมูลที่ส่งโดยปราศจากข้อมูลซ้ำซ้อน โดยในชั้นที่มีโปรโตคอลหลัก 2 ตัว คือ TCP และ UDP
3. Internet Layer ในชั้นนี้มีการกำหนดดาต้าแกรม (Datagram) และทำการหาเส้นทางการส่ง การทำงานในชั้นนี้จะเป็นแบบ Connectionless เนื่องจากไม่มีการเชื่อมต่อระหว่างต้นทางกับปลายทางก่อนดาต้าแกรมแต่ละตัวสามารถเลือกเส้นทางไปโดยอิสระ และไม่มีการรับประกันความถูกต้องข้อมูลหรือลำดับการส่ง
4. Network Interface Physical Layer ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็นส่วนๆ เรียกว่า เฟรม หรือ แพคเก็ตและส่งข้อมูลที่ได้ไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

2.3.3 ข้อแตกต่างระหว่างชุดโปรโตคอล TCP/IPกับ OSI Model

1. ลำดับการติดต่อสื่อสารของชั้นเลเยอร์ ในรูปแบบ OSI นั้นจะกำหนดลำดับชั้นการสื่อสารที่เป็นลำดับขั้นตอนการติดต่อที่แน่นอน โดยเฉพาะการอินเตอร์เฟซระหว่างชั้นเลเยอร์ ซึ่งทำให้รูปแบบ OSI สามารถเป็นระบบเปิดสำหรับระบบเครือข่ายคอมพิวเตอร์ทั่วไป เพราะไม่ว่าจะมีการเปลี่ยนแปลงโปรโตคอลในเลเยอร์ชั้นใดก็ตามจะไม่มีผลกระทบต่อการสื่อสารเลเยอร์ชั้นถัดไป ในขณะที่ชุดโปรโตคอล TCP/IP จะไม่มีการกำหนดรูปแบบการติดต่อที่ตายตัว เพื่อให้ผู้ออกแบบเครือข่ายอิสระสามารถเปลี่ยนแปลงโครงสร้างของเครือข่ายได้ง่ายขึ้น
2. การติดต่อสื่อสารระหว่างเครือข่ายหรือการอินเทอร์เน็ต คือ การติดต่อสื่อสารข้อมูลระหว่างระบบคอมพิวเตอร์ 2 ระบบที่ไม่สามารถติดต่อสื่อสารกันได้โดยผ่านทางเครือข่ายการสื่อสารข้อมูลเพียงเครือข่ายเดียวได้ต้องอาศัยเครือข่ายตั้งแต่ 2 เครือข่ายขึ้นไปในการติดต่อสื่อสารกัน และเครือข่ายเหล่านี้อาจจะมีลักษณะของเครือข่ายที่ต่างกันได้

1. ความแตกต่างในเรื่องของอินเทอร์เน็ตระหว่างชุดโปรโตคอล TCP/IP กับรูปแบบ OSI ก็คือในชุดโปรโตคอล TCP/IP จะใช้โปรโตคอลสำหรับอินเตอร์เน็ตที่เรียกว่า โปรโตคอล IP (Internet Protocol) ซึ่งในรูปแบบ OSI จะเรียกว่าโปรโตคอลสำหรับการอินเตอร์เน็ตว่า โปรโตคอลเน็ตเวิร์ค
2. การบริการเชื่อมต่อการสื่อสาร (Connection Service) ในชุดโปรโตคอล TCP/IP นั้นจะมีการบริการการเชื่อมต่อการสื่อสารระหว่างต้นทางและปลายทาง 2 แบบ คือการบริการแบบ Connectionless และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ Connectionless-Oriented ส่วนในรูปแบบ OSI จะให้ข้อความสำคัญเฉพาะกับการบริการแบบ Connection-Oriented เท่านั้น

3. โปรโตคอลควบคุมการจัดการสื่อสาร ในชุดโปรโตคอล TCP/IP จะใช้โปรโตคอล TCP เป็นโปรโตคอลสำหรับควบคุมการสื่อสาร กำหนดตำแหน่งต้นทางและปลายทาง และอื่นๆ กับข้อมูล ซึ่งในรูปแบบ OSI นั้นจะแบ่งแยกการควบคุมการสื่อสารออกจากกันโดยใช้โปรโตคอลเซสชั่นและโปรโตคอลทรานสปอร์ตตามลำดับ

2.3.4 ลักษณะของการติดต่อ

แบ่งออกเป็น 2 ชนิด คือ

Connection-Oriented คือ การติดต่อที่ต้องมีการเชื่อมโปรเซส (Process) ที่จะมีการส่งหรือรับข้อมูล ซึ่งใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโปรเซส ถึงแม้ว่าข้อมูลนี้อาจจะผ่าน Packet-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งขั้นตอนการทำงานออกเป็น

- ขั้นการสร้างการติดต่อ (Connection Estabishment)
- ขั้นการส่งผ่านข้อมูล (Data Transfer)
- ขั้นยกเลิกการติดต่อ (Connection Termination)

Connectionless หรือดาต้าแกรม คือ จะไม่มีขั้นการสร้างการติดต่อ และขั้นการยกเลิกการติดต่อ แต่จะมีขั้นการส่งผ่านข้อมูลเพียงอย่างเดียว โดยข้อมูลซึ่งเรียกว่า ดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

2.4 พอร์ตและซ็อกเก็ต (TCP Port and Socket)

การใช้บริการของ TCP ทำได้โดยผ่านทางพอร์ต โดยมีการกำหนดหมายเลขให้กับการบริการต่าง ๆ การบริการที่ต่างกันจะมีหมายเลขพอร์ตไม่เหมือนกัน

พอร์ตถูกกำหนดตามชนิดของการบริการ เช่น ต้องการสื่อสารกันด้วย Telnet เครื่องคอมพิวเตอร์ที่ต้องการบริการจะส่งความต้องการออกทางพอร์ตหมายเลข 23 ที่จัดไว้สำหรับบริการ Telnet ไปยังเครื่องคอมพิวเตอร์ที่ต้องการติดต่อด้วย ถ้าเครื่องคอมพิวเตอร์ปลายทางยอมรับการติดต่อ การสื่อสาร Telnet ก็จะเริ่มขึ้น ในช่วงเวลานี้ถ้ามีความต้องการการสื่อสารด้วย Telnet จากเครื่องคอมพิวเตอร์อื่นเข้ามา จะไม่สามารถทำการติดต่อได้เนื่องจากพอร์ตหมายเลข 23 สำหรับการบริการ Telnet ถูกใช้งานอยู่ TCP พอร์ตเป็นพอร์ตทาง Logical ไม่ใช่พอร์ตที่อยู่ด้านหลังของคอมพิวเตอร์ ใช้คอนเน็คเตอร์ (Connector) บนแบคโบน เพื่อให้สามารถสื่อสารบริการหลายประเภทได้พร้อม ๆ กัน

ซ็อกเก็ตเป็นหมายเลขกำหนดวงจรซึ่งทางเข้าออก ของระดับชั้น TCP ของเครื่องคอมพิวเตอร์บนเครือข่าย หมายเลขซ็อกเก็ตประกอบด้วย ไอพีแอดเดรส กับหมายเลขพอร์ตรวมกัน หมายเลขซ็อกเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้กับการสื่อสารเครื่องคอมพิวเตอร์ที่อยู่ต่างเครือข่ายได้เนื่องจาก ไอพีแอดเดรสของเครือข่ายจะไม่ซ้ำกัน ถึงแม้ว่าหมายเลขพอร์ตจะซ้ำกันได้

เครื่องคอมพิวเตอร์บนเครือข่ายจะสร้างตารางพอร์ตขึ้นมาเก็บรายการของพอร์ตที่มีอยู่เพื่อใช้ตรวจสอบว่าจะใช้บริการพอร์ตที่ต้องการได้หรือไม่ และพอร์ตมีสถานะเป็นอย่างไรการที่พอร์ตที่อยู่ในรายการตารางพอร์ตของแต่ละเครื่อง ร้องขอการสื่อสารซึ่งกันและกันเรียกว่า 'Bindig' ของพอร์ตและวิธีที่ทำให้พอร์ตที่ถูกใช้อยู่สามารถใช้สื่อสารได้อีกจากการขอใช้บริการเดียวกันเรียกว่า 'Multiplexing'

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

ตาราง 2.2 หมายเลขพอร์ตที่ให้บริการต่าง ๆ

2.4.1 ข้อกำหนดรูปแบบ IP (Internet Protocol)

IP เป็นข้อกำหนดรูปแบบหนึ่งในชุดข้อกำหนดรูปแบบ TCP/IP ลักษณะการให้บริการเป็นแบบ Connectionless เช่นเดียวกับ UDP มีหน้าที่หลักเกี่ยวกับการจัดการดาต้าแกรมระหว่างเครื่องคอมพิวเตอร์ เช่น ที่อยู่ปลายทาง วิธีการหาเส้นทางที่ดีที่สุดที่จะไปถึงปลายทาง และจัดเตรียมวิธีการแก้ปัญหาที่อาจจะเกิดขึ้นในกรณีต่าง ๆ เป็นต้น

นอกจากนี้ยังมีหน้าที่แบ่งย่อยดาต้าแกรมที่มีขนาดใหญ่เกินกว่าที่ IP Message กำหนดไว้ (ประมาณ 64 กิโลไบต์) และรวบรวมดาต้าแกรมย่อยกลับมาเป็น Message ให้เหมือนเดิมที่เครื่องปลายทาง การแบ่งย่อยและรวบรวมดาต้าแกรมมีอยู่ 4 วิธีคือ

1.Segmentation วิธีแบ่งดาต้าแกรมขนาดใหญ่ให้เป็นดาต้าแกรมย่อย

2.Reassemble วิธีรวบรวมดาต้าแกรมย่อย ๆ ให้กลับอยู่ในสภาพเดิม

3.Concatenation วิธีรวมดาต้าแกรมให้เป็นบล็อก

4.Separation วิธีแบ่งบล็อกให้เป็นดาต้าแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol)

ปัญหาหนึ่งที่เกิดขึ้นได้ง่ายกับดาต้าแกรม ก็คือ ดาต้าแกรมเดินทางผิดเส้นทาง ทำให้สูญหายหรือบางส่วนของข่าวสารเสียหาย เครื่องคอมพิวเตอร์ด้านส่งต้องรู้เงื่อนไขที่จะทำให้เกิดความผิดพลาดต่าง ๆ บนเครือข่ายเพื่อให้ดาต้าแกรมถูกส่งออกไปอย่างถูกต้อง

ICMP จะรายงานความผิดพลาดในระบบให้กับไอพีพิเศษ ICMP มี Header เหมือนกับข่าวสารไอพีอื่น ๆ โดยจะถูกส่งรวมไปกับข่าวสารอื่นสู่เครื่องปลายทาง ซึ่งเครื่องปลายทางจะส่งรายงานของความผิดพลาดกลับมาให้เครื่องด้านผู้ส่งได้รู้ว่าเกิดอะไรขึ้นบ้างในการเดินทาง ซึ่งข้อมูลในรายงานจะถูกนำมาใช้แก้ปัญหาต่อไป

2.4.3 ซ็อกเก็ต (Socket)

2.4..1 เอพีไอ (API) ข้อกำหนดในการติดต่อสื่อสาร

TCP/IP เป็น Software ในการเชื่อมโยงเครือข่ายชนิดหนึ่งที่สนับสนุนโปรแกรมประยุกต์ในการอินเตอร์เฟสสำหรับการติดต่อสื่อสารบนอินเตอร์เน็ตซึ่งจะอธิบายชนิดของ APIs ในที่นี้เพียง 2 ชนิด

Socket Systemcalls

ซ็อกเก็ตเข้าถึงข้อกำหนดในการติดต่อสื่อสารของโครงข่าย ซึ่งโดยทั่วๆ ไปจะเกี่ยวกับการเปลี่ยนแปลง BSD ของ UNIX ซึ่งเป็นอุปกรณ์อันดับแรกของซ็อกเก็ต API การอินเตอร์เฟสของซ็อกเก็ตได้จัด API สำหรับการติดต่อสื่อสารของโครงข่ายซึ่งเป็น API ที่ใกล้เคียงกับการทำงานของ I/O มาก ๆ

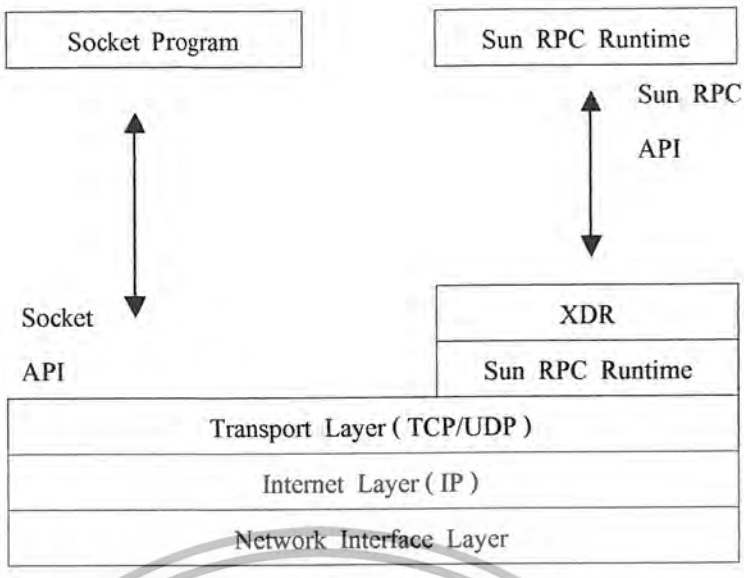
การอินเตอร์เฟสของซ็อกเก็ตเป็นเครื่องมือที่ใช้งานอย่างกว้างขวางสำหรับระบบปฏิบัติการอื่น ๆ โปรแกรมที่เข้าถึงซ็อกเก็ตสามารถที่จะใช้งานเป็นเครื่องมือสำหรับการติดต่อสื่อสารบนเครือข่ายอินเตอร์เน็ต โดยอยู่ในสภาพแวดล้อมที่แตกต่างกันได้และติดต่อได้กับโฮสทุกชนิดซ็อกเก็ต API โดยทั่วไปมักจะใช้สำหรับโครงข่ายที่ใช้งานบน TCP/IP และก็สามารถใช้งานได้ดีกับโครงข่ายแบบอื่น ๆ

Remote Procedure Call Facilities

Remote Procedure Call (RPC) เป็นชนิดหนึ่งของ API ซึ่งยอมให้ผู้เขียนโปรแกรมสำหรับโครงข่ายใช้ procedure call ที่มีโครงสร้างอย่างง่าย ๆ เพื่อที่จะควบคุมกันระหว่าง procedure ของโฮสที่มีการทำงานแตกต่างกันบนอินเตอร์เน็ต สิ่งอำนวยความสะดวกของ RPC ที่ใช้กันอย่างกว้างขวางบนสภาวะ TCP/IP ได้แก่

- Sun RPC
- Hewlett – Packard RPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.5 แสดงการเปรียบเทียบระหว่างซ็อกเก็ตและ SUN RPC application programing

ซ็อกเก็ตอินเตอร์เฟสเป็นการเริ่มพัฒนาสำหรับหลาย ๆ BSD ของระบบปฏิบัติการ LINUX โดยการใช้ ซ็อกเก็ตอินเตอร์เฟส จะมี 2 โปรแกรมประยุกต์คือ

1. การทำงานที่ Local System
2. การทำงานที่ Remote System

ซึ่งสามารถติดต่อกับโฮสต์อื่น ๆ ได้ด้วยวิธีมาตรฐาน ซ็อกเก็ตอินเตอร์เฟสที่ใช้กันอย่างกว้างขวางคือ API ที่ใช้งานบน TCP/IP

ซ็อกเก็ต API เป็นอุปกรณ์ที่ใช้สำหรับการเชื่อมต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการระหว่าง 2 ส่วนประกอบ ประยุกต์ที่ทำงานบนระบบของคอมพิวเตอร์ที่ต่างกัน

การติดต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการแต่ละครั้งนั้น จะเรียกฟังก์ชั่นขึ้นมาจัดการการเชื่อมต่อระหว่างผู้ขอใช้บริการและผู้ให้บริการนั้น ๆ การเชื่อมต่อระหว่าง 2 ส่วนประกอบ (Component) ประยุกต์นี้เป็นพื้นฐานบนการใช้โครงสร้างข้อมูลที่ถูกเรียกว่า ซ็อกเก็ตเพื่อเป็นการบริการการติดต่อสื่อสาร โดยใช้ TCP/IP

ส่วนประกอบของผู้ขอใช้บริการและผู้ให้บริการ (Client – Server component) จะเรียกฟังก์ชั่นที่ใช้ส่งและรับข้อมูลบนอินเตอร์เน็ตซึ่งคล้ายกับวิธีการเรียกฟังก์ชั่นที่ใช้กับไฟล์ I/O ทั่ว ๆ ไป

ซ็อกเก็ต API สามารถจะใช้งานได้ในโปรโตคอลอื่น ๆ ที่นอกเหนือจาก TCP/IP แต่ในที่นี้จะอธิบายเฉพาะโปรโตคอล TCP/IP เท่านั้น

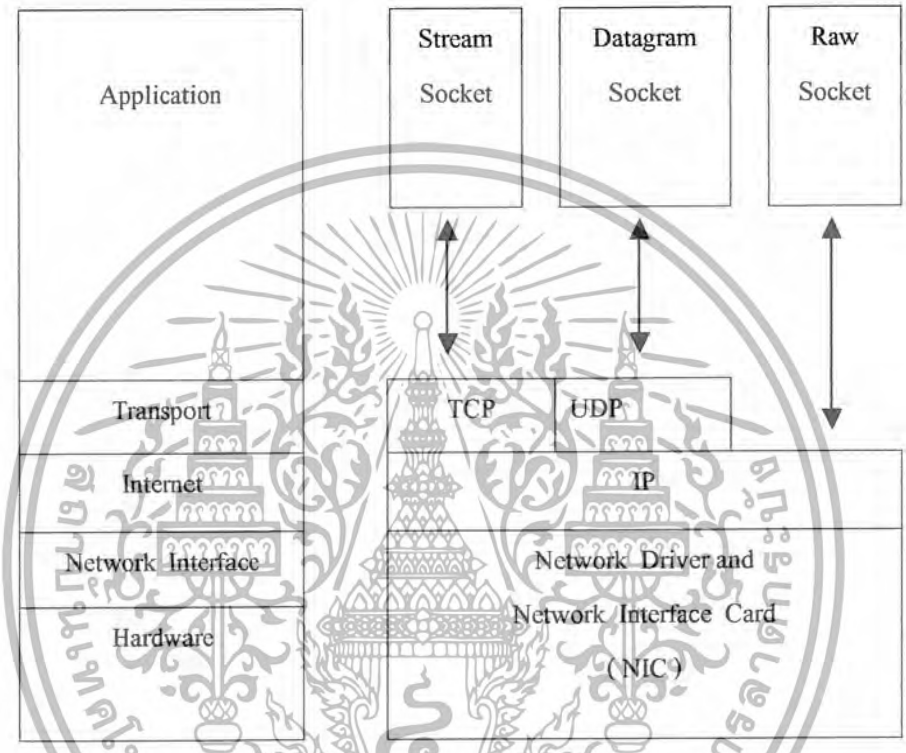
2.4.4.2 ชนิดของซ็อกเก็ต

ซ็อกเก็ตที่ใช้ในการติดต่อสื่อสารบน TCP/IP จะมีอยู่ 3 แบบด้วยกันดังนี้

1. Stream Socket คือซ็อกเก็ตที่ใช้โปรโตคอล TCP/IP ชั้นทรานสปอร์ตในการติดต่อซ็อกเก็ตชนิดนี้จะใช้การติดต่อแบบ Connection Oriented ซึ่งการรับส่งข้อมูลแบบนี้สามารถเชื่อถือได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Datagram Socket คือซ็อกเก็ตที่ใช้โปรโตคอล UDP ชั้น ทรานสปอร์ตในการติดต่อ Datagram Socket จะเชื่อถือไม่ค่อยได้รูปแบบของข้อมูลที่ส่งจะถูกแบ่งออกเป็นส่วน ๆ โดยที่แต่ละอันจะถูกส่งไปปลายทางโดยอิสระจากกัน
3. Raw Socket คือซ็อกเก็ตที่อยู่ภายใต้ชั้น IP และ ICMP ซ็อกเก็ตชนิดนี้จะใช้เฉพาะการส่งที่มีจุดหมายพิเศษ



รูป 2.6 แสดงลักษณะทั้งสามแบบของซ็อกเก็ต

Socket Address

ระบบของ ซ็อกเก็ต โดยทั่วๆ ไปจะอ้างอิงถึงโครงสร้างข้อมูลแบบพอยน์เตอร์ (pointer) ที่ประกอบไปด้วยข้อมูลต่าง ๆ ซึ่งก็คือซ็อกเก็ตแอดเดรสนั่นเอง

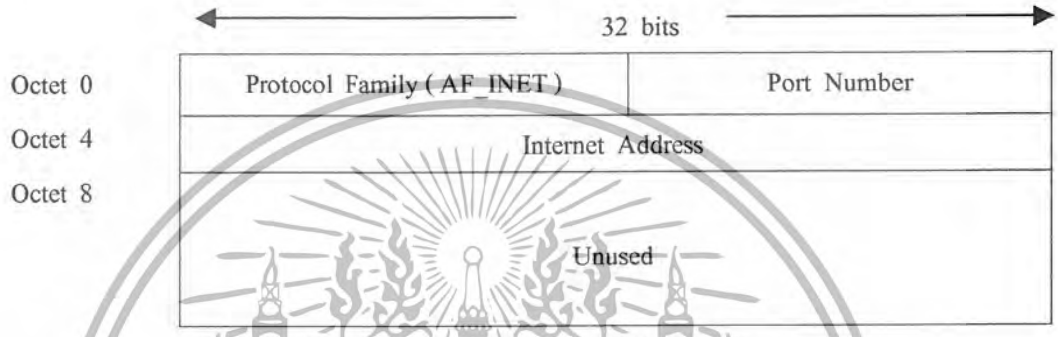
ซ็อกเก็ตแอดเดรสจะใช้เฉพาะการสื่อสารแบบ one – end

สำหรับโปรโตคอล TCP/IP จะมีโครงสร้างของซ็อกเก็ตแอดเดรสดังนี้

1. Family เป็นเลขจำนวนเต็ม 16 บิต ที่ชี้เฉพาะโปรโตคอล TCP/IP คือ AF_INTER , UNIX คือ AF_UNIX
2. พอร์ต เป็นเลขจำนวนเต็ม 16 บิต ที่ระบุหมายเลขพอร์ตสำหรับใช้ในการต่าง ๆ สำหรับผู้ให้บริการหมายเลขพอร์ตที่ผู้ใช้จะถูกกำหนดโดยตัวผู้ให้บริการเอง ซึ่งผู้ขอใช้บริการที่ต้องการติดต่อกับผู้ให้บริการ โดยใช้โปรแกรมนั้นจะต้องรู้หมายเลขพอร์ตที่ใช้ติดต่อจึงจะสามารถติดต่อกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Address เป็นเลขจำนวนเต็ม 32 บิต ซึ่งภายในคือ ไอพีแอดเดรสที่อยู่ในรูปแบบของเลขฐานสองในการทำงานเพื่อติดต่อกัน ผู้ขอใช้บริการต้องรู้ไอพีแอดเดรสของผู้ให้บริการ จึงจะสามารถทำการติดต่อกันได้ แต่ละการติดต่อของ ผู้ขอใช้บริการ – ผู้ให้บริการ ต้องมีการกำหนด ซ็อกเก็ตแอดเดรส ซึ่งซ็อกเก็ตแอดเดรสก็หมายรวมถึงซ็อกเก็ตแอดเดรสของ ผู้ให้บริการ และซ็อกเก็ตแอดเดรสของผู้ขอใช้บริการกระบวนการติดต่อของ ผู้ขอใช้บริการ และ ผู้ให้บริการ ต้องมีการชี้เฉพาะไปที่โปรโตคอลชั้นทรานสปอร์ตเพื่อที่ใช้ในการติดต่อสื่อสารรูป 2.5 จะแสดงความหมายของชนิดข้อมูลที่จะใช้ในการกำหนอซ็อกเก็ตแอดเดรส



รูป 2.7 แสดงโครงสร้างของข้อมูลที่ใช้ในการกำหนดซ็อกเก็ตแอดเดรส

2.5 การสื่อสารข้อมูลด้วยระบบโมเด็ม

เมื่อผู้ใช้สั่งให้โมเด็มเริ่มทำการโทรศัพท์ไปยังโมเด็มปลายทาง กระบวนการต่าง ๆ ระหว่างโปรแกรมสื่อสารและโมเด็มก็จะเริ่มขึ้น ถ้าหากผู้ใช้เคยใช้งานมาแล้วก็จะคุ้นเคยกับเสียงต่าง ๆ ที่ดังออกมาจากลำโพงของโมเด็ม เช่น เสียง Dial Tone ตามด้วยเสียง DTMF หรือเสียง Pulse ในขณะที่ทำการหมุนเลขหมายเลขโทรศัพท์ปลายทาง เสียง Ringing Tone และเสียงสัญญาณพาหะโต้ตอบกันระหว่างโมเด็มต้นทางและปลายทาง หลังจากนั้นโมเด็มก็จะเงียบลงพร้อมกับเข้าสู่การเชื่อมต่อ (Connection) ซึ่งกระบวนการดังกล่าวนี้นับได้ว่าเป็นกระบวนการทำงานปกติที่ผู้ใช้คุ้นเคยเป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน	ผู้ใช้	ซอฟต์แวร์	โมเด็มต้นทาง	โมเด็มปลายทาง
1.	เลือกคำสั่ง "Dial" จากซอฟต์แวร์	เปิดสัญญาณ DRT เพื่อส่งคำสั่งหมุนหมายเลขไปยังโมเด็ม	เปิดลำโพง ยกหูโทรศัพท์ รอสัญญาณให้หมุนหมายเลข	
2.		รอ result codes จากโมเด็ม	รอการตอบรับจากปลายทาง ทั้งนี้ระยะเวลาการรอขึ้นกับการกำหนดค่ารีจิสเตอร์	
3.				เสียงโทรศัพท์ดัง
4.				ตอบรับสัญญาณ
5.			รับรู้สัญญาณตอบรับและทำการส่งสัญญาณ Orginate Carrier	
6.			รับทราบวิธีการมอดูเลชั่น และความเร็วของแต่ละฝ่าย	รับทราบวิธีการมอดูเลชั่นและความเร็วของแต่ละฝ่าย
7.			โมเด็มตกลงรับรู้โปรโตคอลการควบคุมความผิดพลาดและการบีบอัดข้อมูลของแต่ละฝ่าย	
8.			ส่ง result code "Connect" ไปยัง PC ปิดลำโพงและเปิดสัญญาณ CD	
9.		รับรู้ result code และสัญญาณ CD รายงานให้ผู้ใช้ทราบว่าการติดต่อได้เกิดขึ้นแล้ว		
10.	เริ่มการติดต่อกับ	ดำเนินการสื่อสาร	สั่งและรับข้อมูล	สั่งและรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	โฮสคอมพิวเตอร์	และคอยดูสัญญาณที่ขาดหายไปจากหน้าจอสัญญาณ CD		
11.	การสื่อสารเสร็จสมบูรณ์เลือกคำสั่ง "Disconnect"	ปิดสัญญาณ DRT		
12.			วางสายโทรศัพท์	ยกเลิกสัญญาณวางสายโทรศัพท์

ตาราง 2.3 กระบวนการทำงานของโมเด็มต้นทางและปลายทางตั้งแต่ขั้นตอนที่ 1 ถึง 12

จากตาราง 2.3 จะแสดงผังลำดับขั้นตอนการทำงาน โดยเริ่มตั้งแต่ขั้นตอนที่สั่งให้โมเด็มหมุนหมายเลขโทรศัพท์ จนกระทั่งถึงการวางหูโทรศัพท์ และจากตาราง 2.3 นี้เอง ผู้ใช้สามารถสังเกตได้ว่าสิ่งที่ทำงานมากที่สุดในกระบวนการสื่อสาร ก็คือ โมเด็ม ส่วนโปรแกรมสื่อสารจะมีหน้าที่เพียงส่งชุดคำสั่ง AT ไปให้กับโมเด็มเท่านั้น การกำหนดเวลาต่าง ๆ ในกระบวนการนี้จะขึ้นอยู่กับข้อมูลที่อยู่ในรีจิสเตอร์ S ด้วย ตัวอย่างเช่น รีจิสเตอร์ S7 จะเก็บค่าของเวลาที่โมเด็มจะรอเสียงตอบจากโมเด็มปลายทาง เป็นต้น

ขั้นตอนการเชื่อมต่อระหว่างโมเด็มต้นทางและปลายทางที่ได้กล่าวไปแล้วนั้น โมเด็มทั้งคู่จำเป็นที่จะต้องมีพื้นฐานที่เหมือนกันบางประการ เช่น ความเร็ว รูปแบบของข้อมูล และโปรโตคอลถ่ายโอนไฟล์ในการรับส่งข้อมูล

2.5.1 ประเภทของชุดคำสั่ง AT

โดยพื้นฐานแล้ว เราสามารถจะแบ่งชุดคำสั่ง AT ออกเป็น 2 ประเภทใหญ่ ๆ ได้ดังนี้

1. ชุดคำสั่งที่ใช้ในการปฏิบัติงาน อย่างเช่น ATD (คำสั่งให้หมุนหมายเลขโทรศัพท์) หรือ ATH (คำสั่งให้วางสายโทรศัพท์) เป็นต้น

2. ชุดคำสั่งที่ให้กำหนดค่า หรือเปลี่ยนแปลงค่าต่าง ๆ อย่างเช่น ATS7 = 90 เป็นการกำหนดค่าให้รีจีสเตอร์ S มีค่า 90 ซึ่งก็คือ ให้โมเด็มรอการตอบรับจากโมเด็มปลายทางเป็นเวลา 90 วินาที หลังจากนั้นหากยังไม่มีสัญญาณใด ๆ ตอบกลับมาโมเด็มก็จะวางสายทันที

ตามปกติแล้ว โปรแกรมสื่อสารมักจะตั้งค่าของรีจีสเตอร์ S ต่าง ๆ ที่จำเป็นเอาไว้ตั้งแต่ตอนที่ผู้ใช้เรียกโปรแกรมนั้น ๆ ขึ้นมา ซึ่งผู้ใช้จะสามารถควบคุมและเปลี่ยนแปลงค่าของรีจีสเตอร์ S เหล่านี้ได้ในภายหลัง

2.5.2 การออนไลน์และออฟไลน์

สภาวะออฟไลน์ จะสามารถเรียกได้อีกอย่างหนึ่งว่า สภาวะคำสั่ง (Commond state) หมายถึง สภาวะที่ผู้ใช้สามารถจะส่งคำสั่งต่าง ๆ ไปยังโมเด็มได้ หรือ จะพูดอีกนัยหนึ่ง ก็คือ สภาวะที่โมเด็มจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลความหมายของข้อมูลที่ได้รับมาจาก PC ให้เป็นคำสั่งเท่านั้น ซึ่งสภาวะนี้จะไม่ได้รับ – ส่งข้อมูลกับโมเด็มปลายทาง แต่จะสื่อสารกันกับ PC เท่านั้น

สภาวะออนไลน์ หมายถึง สภาวะที่โมเด็มได้เชื่อมต่อกับโมเด็มปลายทางเป็นที่เรียบร้อยแล้ว ข้อมูลที่ส่งออกมาจาก PC ก็จะผ่านจากโมเด็มต้นทางไปยังโมเด็มปลายทางเสมอ

ถ้าหากผู้ใช้ต้องการที่จะส่งคำสั่งให้กับโมเด็ม หรือต้องการจะให้โมเด็มกลับมาอยู่ในสภาวะออฟไลน์ เพื่อรับคำสั่งจาก PC ก็สามารถทำได้โดยมีวิธีการอยู่ 2 วิธี คือ วิธีที่ 1 ให้โมเด็มวางสายแล้วกลับมาอยู่ในสภาวะออนไลน์ใหม่ และวิธีที่สอง คือ ส่งชุดอักขระ Escape Sequence เข้าไปยังโมเด็มในขณะที่ออนไลน์ ซึ่งวิธีหลังนี้จะมีความเหมาะสมกว่า เพราะการเชื่อมต่อระหว่างโมเด็มจะยังคงดำเนินไปอยู่ และหลังจากที่ได้ส่งคำสั่งต่าง ๆ ให้กับโมเด็มเป็นที่เรียบร้อยแล้ว ก็จะสามารถกลับเข้าไปอยู่ในสภาวะออนไลน์ได้เช่นเดิม วิธีการส่งชุดอักขระ Escape sequence สำหรับโมเด็มที่เข้ากันได้กับโมเด็มของ Hayes คือ ให้รอ 1 วินาที (เรียกว่า Guard Time) แล้วกดปุ่ม + ติดต่อกัน 3 ครั้ง (+++) การที่ต้องรอ 1 วินาทีก่อนที่จะกดปุ่มเครื่องหมาย + เป็นสิ่งที่จำเป็น เนื่องจากว่าโมเด็มจะได้รู้ว่าอักษร + นั้นเป็นส่วนของอักขระ Escape sequence ไม่ใช่ส่วนของข้อมูลที่จะต้องส่งไปยังโมเด็มปลายทาง

2.5.3 รูปแบบของชุดคำสั่ง AT

การใช้ชุดคำสั่ง AT จำเป็นต้องขึ้นต้นด้วยอักษร AT เสมอ และจะต้องจบลงด้วยการกด Enter หรือ Carrige return ยกเว้นเฉพาะคำสั่ง A/ ไม่ต้องกดปุ่ม Enter ซึ่งหมายถึง ให้โมเด็มกลับไปทำคำสั่งล่าสุดซ้ำอีกครั้งหนึ่ง การที่ Hayes ได้กำหนดให้คำสั่งต่าง ๆ ขึ้นต้นด้วยอักษร AT ก็เพราะต้องการให้โมเด็มรับรู้ถึงความสามารถและรูปแบบของอักษรคำสั่งที่ถูกส่งออกมาจากพอร์ตสื่อสารอนุกรมของ PC ซึ่งผู้อ่านอย่าสับสนระหว่างความเร็วของพอร์ตสื่อสารอนุกรม (ความเร็วของ DTE) และความเร็วของโมเด็ม (ความเร็วของ DCE) ในกรณีนี้จะ หมายถึง ความเร็วของ DTE

ตัวอย่างการใช้ชุดคำสั่ง AT ก็ได้แก่ คำสั่ง ATH หมายถึง คำสั่งที่ให้โมเด็มวางสายโทรศัพท์เป็นต้น แต่บางคำสั่งอาจจะต้องการพารามิเตอร์เพิ่มเติม อย่างเช่น คำสั่ง ATDT7304500 หมายถึง ให้โมเด็มหมุนหมายเลข 7304500 โดยหมุนระบบ Tone (T) แต่ถ้าต้องการให้โมเด็มหมุนแบบระบบ Pulse ก็ต้องใช้คำสั่ง ATDP7304500 เป็นต้น

2.5.4 การสนองคำสั่งของโมเด็ม

เมื่อผู้ใช้ส่งคำสั่งต่างๆไปให้กับโมเด็มแล้ว โมเด็มก็จะสนองคำสั่งโดยจะส่งข้อความที่เรียกว่า Result code กลับมายังเครื่อง PC และจะปรากฏขึ้นบนจอภาพขณะที่รันโปรแกรมสื่อสารไว้ Result code เหล่านี้จะเป็นภาษาอังกฤษ อย่างเช่น OK,ERROR,CONNECT 2400 หรืออาจจะเป็นตัวเลขอื่น ๆ ซึ่งคำสั่งที่เกี่ยวข้องกับการส่ง Result code ก็จะมีดังนี้

ATV0 เป็นคำสั่งที่กำหนดให้โมเด็ม Result code ที่จำเป็นออกมาเท่านั้น (Non – verbose)

ATV1 เป็นคำสั่งที่กำหนดให้โมเด็มส่ง Result code ออกมาทุก ๆ ครั้ง หลังจากที่ได้รับคำสั่งต่าง ๆ (Verbose)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ATQ1 เป็นคำสั่งที่กำหนดให้โมเด็มไม่ส่ง Result code ออกมา (Quiet mode)

ATQ0 เป็นคำสั่งที่ใช้ Toggle ให้โมเด็มสามารถกลับไปส่ง Result code ได้ดังเดิม นอกจากนี้ยังมีชุดคำสั่งที่เกี่ยวข้อง ซึ่งจะได้กล่าวถึงในหัวข้อสรุปชุดคำสั่ง AT ในภาคผนวกต่อไป

2.5.5 ชุดคำสั่งเพิ่มเติม

ชุดคำสั่งรุ่นแรกๆ นั้น จะใช้ตัวอักษรตั้งแต่ A จนถึง Z เช่น ATA จนถึง ATZ ซึ่งก็หมายความว่า คำสั่งจะมีให้ใช้ได้ไม่เกิน 26 คำสั่งเท่านั้น แต่โมเด็มในปัจจุบันก็ได้มีความสามารถต่างๆ ที่พิเศษเพิ่มเข้ามา ดังนั้นจึงได้กำหนดชุดคำสั่งเพิ่มเติมขึ้นมา โดยใช้ตัวอักษร & เป็นตัวเข้ามาประกอบ ดังตัวอย่าง AT & F หมายถึง ให้โมเด็มเรียกค่า Dufault Setting ขึ้นมาใช้งาน

2.5.6 การตั้งค่าในรีจิสเตอร์ S

ดังที่กล่าวมาแล้วว่ารีจิสเตอร์ S มีหน้าที่เก็บข้อมูลที่ใช้ควบคุมสถานะการทำงานของโมเด็มเอาไว้ ซึ่งผู้ใช้สามารถตั้งค่าหรือตรวจดูค่าที่เก็บเอาไว้ในรีจิสเตอร์ S ต่างๆได้ โดยใช้คำสั่งที่ขึ้นต้นด้วย ATS ในกรณีที่ต้องการตั้งค่าใช้แก่รีจิสเตอร์ S ผู้ใช้จะต้องใส่ตัวเลขที่ระบุว่าเป็น รีจิสเตอร์ S ตัวที่เท่าใดตามด้วยเครื่องหมาย = และค่าที่ต้องการใส่ลงไป ดังตัวอย่างเช่น ถ้าต้องการให้รีจิสเตอร์ S0 มีค่าเท่ากับ 9 ก็จะต้องใช้คำสั่ง ดังนี้

ATS0 = 9

ส่วนในกรณีที่ต้องการตรวจดูค่าของรีจิสเตอร์ S ต่างๆ ก็จะต้องพิมพ์คำสั่งที่ขึ้นต้นด้วย ATS ตามด้วยตัวเลขที่ระบุว่าเป็นรีจิสเตอร์ S ตัวที่เท่าไร และเครื่องหมาย ? หลังจากที่กด Enter หรือ Carriage return แล้วโมเด็มก็จะส่งค่าที่คุณต้องการทราบในรีจิสเตอร์ S นั้นๆ ออกมาพร้อมกับคำว่า "OK" ดังตัวอย่างเช่น ถ้าต้องการตรวจดูค่าของรีจิสเตอร์ S11 จะต้องใช้คำสั่ง ดังนี้

ATS11 ?

จากนั้นหน้าจอก็จะปรากฏข้อความ

095

OK

ผลลัพธ์ที่ออกมาเป็น 095 ก็คือ ค่าที่เก็บอยู่ในรีจิสเตอร์ S11 นั่นเอง

2.5.7 การส่งหลายคำสั่งในครั้งเดียว

ในกรณีที่ต้องการส่งคำสั่งไปให้โมเด็มมากกว่าหนึ่งคำสั่ง ผู้ใช้อาจจะส่งคำสั่งทั้งหมดไปในครั้งเดียวได้ ซึ่งโมเด็มจะสามารถรับส่งได้ไม่เกิน 40 ตัวอักษร ทั้งนี้ จะนับตัวอักษร AT และ Carriage Return ด้วย ผู้ใช้อาจจะพิมพ์คำสั่งทั้งหมดให้ตัวอักษรอยู่ติดกัน หรือเว้นช่องว่างระหว่างแต่ละคำสั่งได้ ดังตัวอย่าง

ATS7 = 90V1X4DT13055551234 หรือ AT S7 = 90 V1 X4 DT 1–305–555– 1234

ซึ่ง S7 = 90 หมายถึง ให้ใส่ค่า 90 ลงไปในรีจิสเตอร์ S7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V1 หมายถึง ให้โมเด็มส่ง Result Code ออกมาได้บ้างคำ อย่างเช่น OK, ERROR, CONNECT2400 เป็นต้น

DT1-305-555-1234 หมายถึง ให้โมเด็มทำการหมุนหมายเลขโทรศัพท์ DT1-305-555-1234 โดยหมุนแบบระบบ Tone

2.5.8 การเก็บค่าต่าง ๆ ที่เซตเอาไว้ใน Profile

จะเห็นได้ว่า ผู้ใช้สามารถตั้งค่ารีจิสเตอร์ S ต่าง ๆ และใช้คำสั่ง AT เพื่อควบคุมการทำงานของโมเด็มได้ในลักษณะต่าง ๆ ที่ต้องการ ซึ่ง Hayes ก็ได้ออกแบบหน่วยความจำภายในโมเด็มขึ้นมา เพื่อใช้สำหรับเก็บค่าต่าง ๆ ที่ตั้งไว้ และมีให้เลือกใช้ได้มากถึง 4 ชุด หน่วยความจำ โดยที่หน่วยความจำชุดแรก เรียกว่า แอกทีฟ คอนฟิกูเรชั่น (Active Configuration) ใช้เก็บข้อมูลการตั้งค่าต่าง ๆ ที่เซตเอาไว้ใช้งานในปัจจุบัน เมื่อผู้ใช้มีการเปลี่ยนแปลงค่าของรีจิสเตอร์ S ต่าง ๆ และได้ตั้งคำสั่ง AT บางคำสั่ง ข้อมูลการตั้งค่าต่าง ๆ เหล่านี้ก็จะถูกนำไปเก็บเอาไว้ใน แอกทีฟ คอนฟิกูเรชั่น และหลังจากปิดเครื่องไปแล้ว ข้อมูลเหล่านี้ก็จะสูญหายไป นอกจากนี้หน่วยความจำที่เหลือจะมีอยู่ 3 ชุด โดย 2 ชุดจะเป็น Nonvolatile RAM (NVRAM) และอีก 1 ชุดที่เหลือเป็น Read Only Memory (ROM) ซึ่งจะใช้เก็บข้อมูลที่เป็นค่า Default Setting ซึ่งจะอ่านออกมาได้อย่างเดียวไม่สามารถเขียนข้อมูลลงไปได้ ผู้ใช้สามารถจะเรียกเอาค่า Factory Setting ออกมาได้ โดยคำสั่ง AT & F และนำไปเก็บไว้ที่แอกทีฟ คอนฟิกูเรชั่น หลังจากที่ได้ติดตั้งและเปิดเครื่องโมเด็มเป็นครั้งแรก โมเด็มก็จะถูกกำหนดให้เรียกเอา Factory Setting ออกมาใช้งานเสมอ ซึ่งจะมีประโยชน์มากในกรณีที่ผู้ใช้อาจจะประสบกับปัญหาการเซตค่าต่าง ๆ กลับขึ้นมาใช้งานใหม่ได้ และข้อมูลที่เป็นค่าดีฟอลต์เหล่านี้จะมีอยู่ในคู่มือการใช้โมเด็มที่มาพร้อมกับโมเด็ม ส่วนหน่วยความจำที่เป็น NVRAM ทั้งสองชุดจะถูกเรียกว่า User Profiles มีไว้สำหรับให้ผู้ใช้สามารถสำรองเก็บค่าต่าง ๆ ที่เซตเอาไว้ได้ตามที่ต้องการ ซึ่งหน่วยความจำแบบ NVRAM นี้จะไม่สูญหายเมื่อปิดเครื่อง และทุกครั้งหลังจากที่ได้เปิดเครื่องแล้ว โมเด็มก็จะนำข้อมูลจากโปรไฟล์ที่ 0 มาเก็บไว้ที่แอกทีฟคอนฟิกูเรชั่น โดยอัตโนมัติ แต่ถ้าต้องการให้โมเด็มเรียกโปรไฟล์ที่ 1 ขึ้นมาทุก ๆ ครั้งที่เปิดเครื่อง ก็จะต้องใช้คำสั่ง AT&Y1 ดังจะกล่าวในหัวข้อสรุปชุดคำสั่ง AT ในภาคผนวกต่อไป

2.5.8.1 การสร้างและการเรียกใช้ User Profile

การสร้าง User Profile จะมีขั้นตอนดังต่อไปนี้

1. ให้โปรแกรมสื่อสารอยู่ในสภาวะคำสั่ง (หรือเรียกว่า Local Mode)
2. ใช้คำสั่ง AT เพื่อตั้งค่าพารามิเตอร์ต่าง ๆ และตั้งค่ารีจิสเตอร์ S ที่ต้องการ
3. ใช้คำสั่ง AT & W0 หรือ AT & W1 เพื่อเก็บข้อมูลการเซตต่าง ๆ เอาไว้ในโปรไฟล์ที่ 0 หรือ 1

หลังจากนั้นผู้ใช้สามารถเรียกใช้ User Profiles ต่าง ๆ ที่ได้จัดเก็บเอาไว้ขึ้นมาได้โดยคำสั่ง ATZ0 หรือ ATZ1 เพื่อรีเซตโมเด็ม และเรียกข้อมูลการเซตต่าง ๆ ในโปรไฟล์ที่ 0 หรือโปรไฟล์ที่ 1 ขึ้นมาใหม่อีกครั้ง โมเด็มก็จะดึงเอาโปรไฟล์ที่ 0 ขึ้นมาเก็บเอาไว้ที่แอกทีฟคอนฟิกูเรชั่นโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งผู้ใช้สามารถกำหนดให้โมเด็มอ่านไปรไฟล์ที่ 1 ขึ้นมาแทน โดยใช้คำสั่ง AT & Y1 ก่อนที่จะปิดเครื่องได้

2.5.8.2 การเรียกดูข้อมูลที่ได้ตั้งไว้ทั้งหมด

หลังจากที่ผู้ใช้ได้ตั้งค่าพารามิเตอร์ต่าง ๆ เอาไว้เรียบร้อยแล้ว ผู้ใช้จะสามารถเรียกดูข้อมูลที่ได้ตั้งเอาไว้ทั้งหมดได้ โดยใช้คำสั่ง AT & V ดังรูปที่ 2.6

AT & V
ACTIVE PROFILE :
B16 B1 B41 B60 E1 L2 M1 N1 P Q0 V1 W0 X4 Y0 &A0 &C1 &D2 &G0 &K3 &O5 &R0 &S0 &T4 &U0 &X0 &Y0
S00:000 S01.000 S02.043 S03.013 S04.010 S05.008 S06.002 S07.05 S08.002 S09.006 S10.014 S11.095 S12.050 S18.000 S25.005 S26.001 S37.000 S36.007 S38.020 S44.003 S46.002 S48.007 S49.008 S50.016 S97.030
STORED PROFILE 0 :
B16 B1 B41 B60 E1 L2 M1 N1 P Q0 V1 W0 X4 Y0 &A0 &C1 &D2 &G0 &K3 &O5 &R0 &S0 &T4 &U0 &X0
S00:000 S02.043 S06.002 S07.050 S08.002 S09.006 S10.014 S11.095 S12.050 S18.000 S25.005 S26.001 S36.007 S37.000 S38.020 S44.003 S46.002 S48.007 S49.008 S50.016 S97.030
STOTED PROFILE 1 :
B16 B1 B41 B60 E1 L2 M1 N1 P Q0 V1 W0 X4 Y0 &A0 &C1 &D2 &G0 &K3 &O5 &R0 &S0 &T4 &U0 &X0 &Y0
S00:000 S02.043 S06.002 S07.050 S08.002 S09.006 S10.014 S11.095 S12.050 S18.000 S25.005 S26.001 S36.007 S37.000 S38.020 S44.003 S46.002 S48.007 S49.008 S50.016 S97.030
TELEPHONE NUMBERS :
0
2
OK

รูป 2.8 Profile ซึ่งแสดงข้อมูลที่เก็บไว้เรียกใช้โดยคำสั่ง AT & V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.8.3 การเก็บหมายเลขโทรศัพท์ไว้ในโมเด็ม

จากรูปที่ 2.8 ผู้อ่านสามารถเห็นคำว่า Telephone Numbers บริเวณบรรทัดล่างได้ ซึ่งหมายความว่าผู้ใช้สามารถเก็บหมายเลขโทรศัพท์เอาไว้ใน NVRAM ได้ 4 หมายเลข รวมทั้งสามารถเก็บตัวอักษรที่จำเป็นในการหมุนโทรศัพท์ได้ ดังเช่น T (Tone), P (Pause), W (Wait) โดยใช้คำสั่ง

AT & Zn = xxxxxx

และสามารถสั่งให้โมเด็มหมุนโทรศัพท์ตามหมายเลขที่เก็บไว้ได้โดยใช้คำสั่ง

ATS = n

ตัวอย่างเช่น

ถ้าต้องการเก็บหมายเลข 3197707 เอาไว้ในที่เก็บตำแหน่งที่ 2 และเมื่อเรียกใช้หมายเลขนี้จะกำหนดให้โมเด็มหมุนแบบ Tone ให้ใช้คำสั่ง ดังนี้

AT & Z2 = T3197707

และเมื่อต้องการให้โมเด็มหมุนหมายเลขโทรศัพท์ที่เก็บเอาไว้นี้ ให้ใช้คำสั่ง

ATDS = 2

โดยทั่วไปแล้ว โปรแกรมสื่อสารก็มักจะมีฟังก์ชันการเก็บและหมุนหมายเลขไว้ในตัวเช่นกัน ซึ่งมักจะเรียกว่า โฟนบุ๊ค (Phone book) หรือ Dialing Directory แต่สำหรับในกรณีการเก็บหมายเลขโทรศัพท์เอาไว้ใน NVRAM เช่นนี้ มักจะใช้เกี่ยวกับกระบวนการที่ เรียกว่า Dial Backup Number ซึ่งใช้ในระบบสื่อสารที่ค่อนข้างจะอัตโนมัติและต้องการให้โมเด็มติดต่อสื่อสารอยู่ตลอดเวลา กระบวนการ Dial Backup Number นี้จะมีขั้นตอนการทำงาน คือ ขณะที่โมเด็มต้นทางติดต่อสื่อสารกับโมเด็มปลายทางแล้วเกิดปัญหาสายขาดหรือสายหลุดไม่สามารถจะดำเนินการสื่อสารได้ต่อไปได้ โมเด็มต้นทางก็จะสามารถหมุนเบอร์โทรศัพท์ที่เก็บเอาไว้ใน NVRAM ได้อัตโนมัติ เพื่อทำการโทรศัพท์กลับไปเชื่อมต่อกับปลายทางอีกครั้งหนึ่ง หรือโทรศัพท์ไปยังโมเด็มตัวอื่น ๆ ตามที่ผู้ใช้กำหนดเอาไว้

2.5.9 การเชื่อมต่อกับ RS – 232 - C

วงจร RS – 232 – C จะประกอบไปด้วยวงจรรับ (Receive Data : RD) และวงจรส่ง (Transmit Data : TD) แยกอิสระไม่ขึ้นต่อกัน ซึ่งแต่ละวงจรก็จะมีลักษณะการเคลื่อนย้ายข้อมูลแบบทางเดียว คือ ยอมให้ข้อมูลเข้าหรือออกเท่านั้น ดังนั้น จึงทำให้การเชื่อมต่อระหว่าง PC และโมเด็มมีลักษณะแบบ 2 ทาง คือ ทั้งรับข้อมูลเข้าและส่งข้อมูลออกไปพร้อม ๆ กันได้ นอกจากนี้โมเด็มก็ยังมีวงจรต่าง ๆ ที่ใช้สำหรับเชื่อมต่อกับ PC และเป็นวงจรที่อาจจำเป็นต่อโปรแกรมสื่อสารบางชนิด ซึ่งต่อไปนี้จะได้กล่าวถึงวงจรที่สำคัญและมีผลต่อการทำงานของ PC หรือมีผลต่อโปรแกรมสื่อสารรวมทั้งจะได้กล่าวถึงการควบคุมวงจรดังกล่าวด้วย

2.5.9.1 Data Terminal Ready

สัญญาณ Data Terminal Ready (DTR) คือ สัญญาณไฟฟ้าที่ขา 20 ของคอนเน็คเตอร์แบบ RS – 232 – C ซึ่งเป็นสัญญาณที่จะถูกส่งออกมาจาก PC ไปยังโมเด็ม เพื่อให้โมเด็มรับทราบว่าในขณะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้น PC พร้อมที่จะติดต่อกับโมเด็มแล้ว ถ้าสัญญาณ DTR มีค่าเป็นศูนย์ ในระหว่างที่โมเด็มกำลังติดต่อสื่อสารกับโมเด็มปลายทางอยู่ ก็จะทำให้โมเด็มหยุดการสื่อสารและวางหูโทรศัพท์ทันที ดังนั้น ถ้าผู้ใช้ต้องการจะหยุดการสื่อสาร ก็อาจจะกำหนดให้โปรแกรมสื่อสารส่งสัญญาณ DTR ในลักษณะนี้ได้ ซึ่งจะสะดวกและเร็วกว่าวิธีการใช้ Escape Sequence (+++) ตามด้วยคำสั่ง ATH มาก โมเด็มของ Hayes ในยุคต้น ๆ นั้น มักจะมีสวิตช์พิเศษที่มีไว้เพื่อใช้ควบคุมการตอบสนองต่อสัญญาณ DTR ของโมเด็ม ซึ่งถ้าสวิตช์อยู่ในตำแหน่ง ON โมเด็มก็จะทำงานโดยขึ้นอยู่กับสัญญาณ DTR แต่ถ้าสวิตช์อยู่ในตำแหน่ง OFF โมเด็มก็จะไม่สนใจสัญญาณ DTR แต่ในปัจจุบันผู้ใช้จะสามารถเลือกการตอบสนองต่อสัญญาณ DTR ของโมเด็มได้โดยตรงโดยใช้คำสั่ง AT & Dn

2.5.9.2 Carrier Detect

สัญญาณ Carrier Detect (CD) คือ สัญญาณไฟฟ้าที่ขา 8 ของคอนเน็คเตอร์แบบ RS – 232 – C ซึ่งเป็นสัญญาณที่ถูกส่งออกมาจากโมเด็มไปยัง PC มีหน้าที่ทำให้ PC รับทราบว่าในขณะนั้นโมเด็มได้เชื่อมต่อกับโมเด็มปลายทางหรือยัง ถ้าสามารถเชื่อมต่อได้เรียบร้อยแล้ว โมเด็มก็จะทำให้สัญญาณ CD มีค่าเป็น 1 พร้อม ๆ กับส่งข้อความ Result Code คำว่า "CONNECT" ออกมา ซึ่งโปรแกรมสื่อสารก็จะรับทราบสถานะของโมเด็มว่ากำลังออนไลน์ หรือ ออฟไลน์ ได้จากค่าของสัญญาณ CD และเช่นเดียวกันกับสัญญาณ DTR

2.5.10 การควบคุมการไหล

การควบคุมการไหล (Flow Control) ของข้อมูลเปรียบเสมือนได้กับวาล์วปิด – เปิดทางเดินของข้อมูล ทั้งทางด้านขาเข้าและขาออก มีหน้าที่คอยควบคุมจังหวะการไหลเข้าและออกของข้อมูลระหว่าง PC และโมเด็ม การที่จำเป็นต้องมีการควบคุมการไหล ก็เพราะว่าโมเด็มจะไม่สามารถรับส่งข้อมูลได้เร็วเท่ากับ PC หรือ PC อาจจะต้องไปทำงานอย่างอื่น จึงต้องสั่งให้โมเด็มหยุดการส่งข้อมูลให้แก่ PC ชั่วคราว การควบคุมการไหลของข้อมูลจะมีอยู่ 2 แบบ คือ แบบซอฟต์แวร์ และแบบฮาร์ดแวร์ นอกจากนี้โมเด็มที่มีโปรโตคอลการควบคุมความผิดพลาด (error control) อย่างเช่น MNP และ V.42 จะสามารถส่งสัญญาณการควบคุมการไหลของข้อมูล ไปให้กับโมเด็มปลายทางที่ไม่มีการควบคุมการไหลดังกล่าวได้เช่นกัน

2.5.10.1 การควบคุมการไหลแบบฮาร์ดแวร์ (Hardware Flow Control)

วงจรควบคุมการไหลในโมเด็มนั้น จะมีอยู่ 2 วงจร คือ RTS (Request to Send) และ CTS (Clear to Send) ซึ่งจะเป็นตัวควบคุมให้โมเด็มหรือ PC ส่งข้อมูลให้แก่กัน ซึ่งข้อมูลที่จะรับหรือส่งนั้นจะถูกหยุดเอาไว้ชั่วคราว จนกว่าสัญญาณดังกล่าวจะอยู่ในสถานะ ON ซึ่งในกรณีที่โมเด็มไม่สามารถที่จะรับข้อมูลได้จาก PC โมเด็มก็จะควบคุมให้สัญญาณ RTS อยู่ในสถานะ OFF และโปรแกรมสื่อสารก็

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะรับรู้ความเปลี่ยนแปลงของสัญญาณ RTS แล้วก็จะหยุดการส่งข้อมูลให้กับโมเด็มไว้ชั่วคราว จนกระทั่ง RTS กลับมาอยู่ในสถานะ ON อีกครั้งหนึ่ง ซึ่งการรับส่งข้อมูลก็จะดำเนินไปเช่นเดิม

2.5.10.2 การควบคุมการไหลแบบซอฟต์แวร์ (Software Flow Control)

การควบคุมการไหลแบบซอฟต์แวร์ (Software Flow Control) มีลักษณะการทำงานคล้ายกับการควบคุมการไหลแบบฮาร์ดแวร์ แต่แทนที่จะใช้วงจร RTS / CTS การควบคุมการไหลแบบซอฟต์แวร์จะใช้ตัวอักขระที่ เรียกว่า XON และ XOFF ซึ่งจะตรงกับอักขระ CTRL – Q และ CTRL – S ที่พิมพ์มาจากคีย์บอร์ดตามลำดับ เทคนิคดังกล่าวได้พัฒนามาจากการใช้งานโทรพิมพ์ในอดีต เมื่อ PC ต้องการให้โมเด็มหยุดส่งข้อมูลชั่วคราว PC ก็จะส่งอักขระ XOFF ไปยังโมเด็ม และหลังจากที่โมเด็มได้รับอักขระ XOFF แล้ว ก็จะหยุดส่งข้อมูลให้กับ PC ไว้ชั่วคราวจนกว่า PC จะพร้อมที่จะรับข้อมูลใหม่ เครื่อง PC จึงจะส่ง XON ไปยังโมเด็มและข้อมูลก็จะมีการรับส่งกันอย่างเดิม และเช่นเดียวกันโมเด็มก็สามารถส่ง XON / XOFF ไปยัง PC เพื่อที่จะควบคุมจังหวะการหยุดและไหลของข้อมูลจาก PC มายังโมเด็มได้

2.6 สัญญาณภาพ

การส่งข้อมูลมัลติมีเดียประกอบไปด้วยการส่งข้อมูลและการแสดงชุดของภาพ เมื่อภาพไม่ได้เป็น “ภาพเดี่ยว” แต่เป็นส่วนหนึ่งของภาพยนตร์ ภาพแต่ละภาพเหล่านี้จะเรียกว่า เฟรม (Frame) การบีบอัดสัญญาณภาพวีดีโอนั้นเป็นเรื่องจำเป็น อย่างน้อยก็เพื่อลดจำนวนของข้อมูลวิดีโอให้อยู่ในขนาดที่สามารถจัดการได้ การบีบอัดทางเทคนิคสามารถบีบอัดข้อมูลในเฟรมเดียว โดยไม่ต้อง อ้างอิงถึงเฟรมอื่นๆ ซึ่งเรียกว่า Intraframe Coding การบีบอัดข้อมูลวิดีโอนั้นอาศัยการลดสิ่งซ้ำซ้อนในภาพวิดีโอลง ซึ่งการซ้ำซ้อนสามารถเกิดขึ้นได้ในสามแบบคือ

1) Spatial Redundancy พิกเซล (Pixel) ต่างๆ ภายในจะมีความสัมพันธ์ในเชิงระยะซึ่งกันและกัน ตัวอย่างเช่น พิกเซลหนึ่งอยู่ติดกับอีกพิกเซลหนึ่ง การซ้ำซ้อนในเชิงระยะนี้เกิดขึ้นเมื่อพิกเซลที่อยู่ใกล้กันมีสีเหมือนกัน และบางครั้งอาจมีค่าความเข้มของสีเท่ากันอีกด้วยสิ่งนี้เกิดขึ้นค่อนข้างบ่อย แต่ด้วย Spatial Redundancy จะสามารถทำให้ลดลงโดยใช้เทคนิคแบบ Intraframe Coding ที่มีชื่อว่า DCT (Discrete Cosine Transform) ซึ่งจะได้อธิบายต่อไป

2) Spectral Redundancy ประกอบด้วยความเข้มของแสงและสี ซึ่ง Spectral Redundancy ในเฟรมภาพยนตร์จะเกิดขึ้นจากพิกเซลที่มีแนวโน้มจะมีความสว่างในทุกๆ สี แทนที่จะเป็นความสว่างที่เกิดขึ้นกับสีใดสีหนึ่งเป็นการเฉพาะ นอกจากนี้สายตาของคนทั่วไปนั้นจะไวกับความสว่างมากกว่าสี หมายความว่าสายตาของคนเราจะสังเกตเห็นความสว่างที่ผิดปกติไปได้ดีกว่าสีที่ผิดเพี้ยนไป

3) Temporal Redundancy ภาพยนตร์จะแสดงภาพด้วยอัตราความเร็วประมาณ 30 เฟรมต่อวินาที แม้แต่ในภาพยนต์แบบ “แอ็คชัน” บางส่วนของเฟรมเท่านั้นจึงจะเปลี่ยนแปลงไปจากเฟรมก่อนหน้า เมื่อแสดงเฟรมด้วยความเร็วนี้แล้วสุ่มตัวอย่างขึ้นมา เทคนิคหนึ่งที่ใช้ในการแก้ไขปัญหา Temporal redundancy คือ Motion Vector Encoding ซึ่งมีแนวคิดที่จะแบ่งพื้นที่ในเฟรมออกเป็นบล็อกย่อย สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวได้ว่าในการทำงานแบบนี้ "ทุกสิ่งจะเหมือนกับเฟรมก่อนหน้า ยกเว้นมีการเคลื่อนไหวเกิดขึ้นในบล็อกใดๆ" (ตัวอย่างเช่น ในภาพนิ่งที่มีการกระพริบตา จะมีการเปลี่ยนแปลงเฉพาะในบล็อกที่แสดงภาพตาเท่านั้น)

เฟรมแบบง่ายๆ ทั่วไปที่แสดงภาพอันสมบูรณ์จะไม่ได้รับประโยชน์จาก Temporal Redundancy ในการจัดการกับความซ้ำซ้อนแบบต่างๆ จึงได้มีการกำหนดการใช้เฟรมสามแบบ แบบแรกเป็นเฟรมต้นฉบับที่แสดงภาพอันสมบูรณ์ซึ่งเรียกว่า I-frame หรือ Intracode Frame เพื่อสื่อว่าเฟรมแบบนี้สามารถใช้แสดงผลได้โดยไม่ต้องอ้างอิงถึงเฟรมอื่นใด เฟรมแบบที่สองคือ P-frame หรือ Predicted Frame เฟรมแบบนี้จะสร้างขึ้นได้โดยอาศัย B-frame ที่อยู่ก่อนหน้า เวกเตอร์ที่เคลื่อนไหว และภาพที่ผิดพลาดที่ใช้ปรับแก้ความผิดพลาดในเวกเตอร์การเคลื่อนไหว ส่วนเฟรมประเภทสุดท้ายคือ B-frame หรือ Bi-directional Predicted Frame โดยจะเป็นเฟรมที่ประกอบด้วยเวกเตอร์ภาพเคลื่อนไหว แบบเดินหน้า, เวกเตอร์ภาพเคลื่อนไหว แบบถอยหลัง และข้อผิดพลาด
เฟรมแบบ B-frame จะถูกใช้เมื่อ P-frame มีวัตถุภาพเกิดขึ้นใหม่

เทคนิค Disrete Cosine Transform การบีบอัดสามารถใช้สมการแทนที่ข้อมูลแทนที่จะเป็นแบบสุ่มตัวอย่าง (Sampling) อย่างซ้ำๆ แล้วส่งค่าจริงออกไป ให้พิจารณากรณีที่ท่านต้องวาดเส้นตรง เริ่มจากวิชาพีชคณิตที่สอนไว้ว่าเราสามารถจะวาดเส้นตรงได้โดยใช้สมการ $y = mx+c$ นั่นหมายความว่าสิ่งที่เราต้องการก็คือค่าของ x และ c เพียงสองค่าเท่านั้น

เห็นได้ชัดว่าการบีบอัดนั้นไม่ง่าย เคล็ดลับอยู่ที่การหาค่าสมการที่ใช้แสดงแทนข้อมูลที่สนใจ และปกติสมการจะมีความซับซ้อนมากกว่าสมการเส้นตรง เทคนิคหนึ่งที่นำมาใช้ในการบีบอัดข้อมูลก็คือ DCT (Discrete Cosine Transform) โดยพยายามที่จะแสดงข้อมูลออกมาในชุดของ Cosine Wave และ Cosine Wave บางส่วนเท่านั้น (ที่สอดคล้องกับความถี่หลัก) ซึ่งจัดเก็บไว้ ส่วนที่ทิ้งไปก็ทำให้เกิดการสูญเสียของข้อมูล

DCT นั้นใช้ในมาตรฐาน MPEG (Moving Picture Experts Group) ซึ่งจะได้อธิบายต่อไป โดยถือว่า DCT เป็นเทคนิคแบบ Intraframe Coding เนื่องจากการทำงานจะดำเนินไปครั้งละเฟรมเดียวเท่านั้น

2.6.1 มาตรฐานวิดีโอ

มาตรฐานในการเข้ารหัสวิดีโอ และ Video Conferencing มีอยู่หลายแบบ โดยจะอธิบายถึงมาตรฐานสำคัญๆ โดยย่อ ดังนี้

1) MPEG (Moving Picture Experts Group) เป็นชื่ออย่างไม่เป็นทางการที่ตั้งให้กับคณะทำงานที่ได้กำหนดมาตรฐานมัลติมีเดีย ชื่อที่ตั้งอย่างเป็นทางการคือ ISO/IEC JTC1 SC29 WG11 ซึ่งย่อมาจาก "International Organization for Standardization/International Electro-Technical Commission, Joint Technical Committee 1, Subcommittee 29, Work Group 11" คณะทำงานของ MPEG นั้นจะมีการประชุมปีละสี่ครั้งเพื่อวางแผนกิจกรรมการทำงานและการดำเนินการ

MPEG มีมาตรฐานที่ได้นำมาใช้งานแล้วสามมาตรฐาน ซึ่งที่รู้จักกันในชื่อ MPEG-1, MPEG-2 และ MPEG-3 ปัจจุบันคณะทำงานกำลังในระหว่างการพัฒนามาตรฐาน MPEG-4 และ MPEG-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) MPEG-1 ถูกใช้กันเป็นหลักในการบีบอัด ภาพวิดีโอลงบนซีดีรอมมีรายละเอียดอธิบายอยู่ใน ISO/IEC Standard 11172 -2

MPEG-1 อาศัยพื้นฐานจากหลักการของความซ้ำซ้อนแบบ Spatial, Spectral และ Temporal ดังที่ได้อธิบายไว้ก่อนหน้านี้ พื้นที่ของเฟรมจะถูกแบ่งเป็นส่วนย่อยที่เรียกว่า Macroblock ขนาด 16 x 16 พิกเซล ข้อมูลที่ไม่ได้ถูกบีบอัดของแต่ละ Macroblock ประกอบด้วย 16 บล็อก โดยแต่ละกลุ่มขนาด 4 บล็อก ใช้เก็บข้อมูลของสีเขียว, สีแดง, สีน้ำเงิน และความสว่างของแต่ละพิกเซลเรียงต่อกันโดย MPEG-1 จะลด 16 บล็อกนี้ไปเป็น 4 บล็อกโดย 2 บล็อกเพื่อแทนค่าความสว่าง(Chrominance) และแต่ละบล็อคที่เหลือเก็บสีแดงและสีน้ำเงิน การบีบอัดข้อมูล MPEG-1 ใช้ DCT โดยใช้เวกเตอร์การเคลื่อนไหว และเฟรมทั้ง 3 แบบคือ I-frame, P-frame และ B-frame

ข้อมูลของ MPEG-1 จะใช้การคำนวณในช่วงบีบอัดข้อมูลสูงกว่าในช่วงขยายข้อมูลอย่างมาก อัตราการบีบอัดข้อมูลของ MPEG-1 ปกติจะอยู่ในช่วง 50:1 แม้ว่าในทางทฤษฎีจะเป็นไปได้ที่จะทำได้ถึง 200:1 ก็ตาม ในปัจจุบัน MPEG-1 ได้รับความนิยมมาก แม้ว่าต่อมาคณะกรรมการ MPEG ได้กำหนด MPEG-2 ขึ้นมาและอยู่ในระหว่างการพัฒนามาตรฐาน MPEG ตัวใหม่ขึ้นมา อีกสาเหตุหนึ่งนั้นอาจเป็นเพราะ MPEG-2 ไม่ได้รับความนิยมมากเท่าที่คาดหวังไว้ อีกเหตุผลหนึ่งคือ MPEG-1 ก้าวล้ำหน้าความสามารถในการคำนวณของเครื่องคอมพิวเตอร์อยู่เล็กน้อย จำต้องใช้เครื่องในยุคถัดมาจึงจะสามารถรองรับความต้องการของ MPEG-1 ได้ดี

3) MPEG-2 ได้นำ MPEG-1 ในส่วนที่เหมาะสมมาใช้กับโทรทัศน์แบบดิจิตอล โดยถือว่า MPEG-2 เป็น Superset ของ MPEG-1 ในลักษณะเด่นที่เห็นชัดเจนของ MPEG-2 คือสนับสนุน เฟรมแบบ Interlaced (เฟรมที่แสดงภาพเพียงครึ่งเดียว) สนับสนุนการมัลติเพล็กซ์สัญญาณวิดีโอ และอัลกอริทึมบีบอัดพื้นที่ส่วนที่มีความสว่างและเปล่งแสงออกมาแบบต่างๆ

การสนับสนุนมาตรฐาน MPEG-2 ยังไม่แน่ชัดนัก แต่อาจมีการเปลี่ยนไปในภายหลัง เนื่องจากพลังในการคำนวณที่สูงขึ้นมาก อย่างเช่น การพัฒนาของ MMX ในชิปของ Intel

4) MPEG-3 มีจุดมุ่งหมายในการปรับปรุง MPEG-2 จากเดิมที่เป็นมาตรฐานบีบอัดข้อมูลสำหรับโทรทัศน์แบบดิจิตอลให้สามารถนำไปใช้ได้กับ โทรทัศน์ความคมชัดสูง (HDTV) ต่อมาก็พบว่าหน้าที่ต่างๆ ที่กำหนดไว้ใน MPEG-3 นั้นสามารถดำเนินการได้โดยใช้ MPEG-2 ดังนั้น MPEG-3 จึงกลายเป็นเพียงเชิงอรรถหนึ่งในประวัติศาสตร์ของเทคโนโลยีมัลติมีเดีย

5) MPEG-4 การทำงานยังคงดำเนินต่อไปสำหรับ MPEG-4 ที่อาจจะกลายเป็นมาตรฐานสากลในช่วงปลายปี 1998 ปัจจุบันข้อเสนอในคุณลักษณะของ MPEG-4 ประกอบไปด้วยข้อสนับสนุนการ สื่อสารข้อมูลวิดีโอแบบสองทาง ข้อสนับสนุนในการดำเนินการสำหรับสายที่มีแบนด์วิดธ์ต่ำ และสนับสนุนให้ผู้ใช้สามารถโต้ตอบเพื่อเลือกเนื้อหาบางส่วนบางตอน และข้ามบางตอนไปได้

6) MPEG-7 ยังไม่พบคำอธิบายว่าเกิดอะไรขึ้นกับ MPEG-5 และ MPEG-6 ได้มีการพัฒนา MPEG-7 กันต่อไปโดยเป้าหมายอย่างหนึ่งคือ ต้องการสร้างมาตรฐานสำหรับ Search Engine สำหรับงานมัลติมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 Windows Media Player MPEG-4

การแสดงภาพวิดีโอ หรือภาพเคลื่อนไหวพร้อมเสียงบรรยายประกอบในโฮมเพจด้วย บราวเซอร์ในรูปแบบของ HTML ผ่านเครือข่ายอินเตอร์เน็ต แบบแสดงรายการสดในลักษณะ Real Time หรือ Live เราจะเรียกว่าเป็น Streaming Files ที่จะส่งจากเว็บเซิร์ฟเวอร์ ผ่านเครือข่ายอินเตอร์เน็ตสู่บราวเซอร์ของผู้ใช้งาน ในความเป็นจริงรูปแบบของ Streamig Files มีอยู่ด้วยกันหลายรูปแบบ เช่น Real Player หรือ Windows Media Player

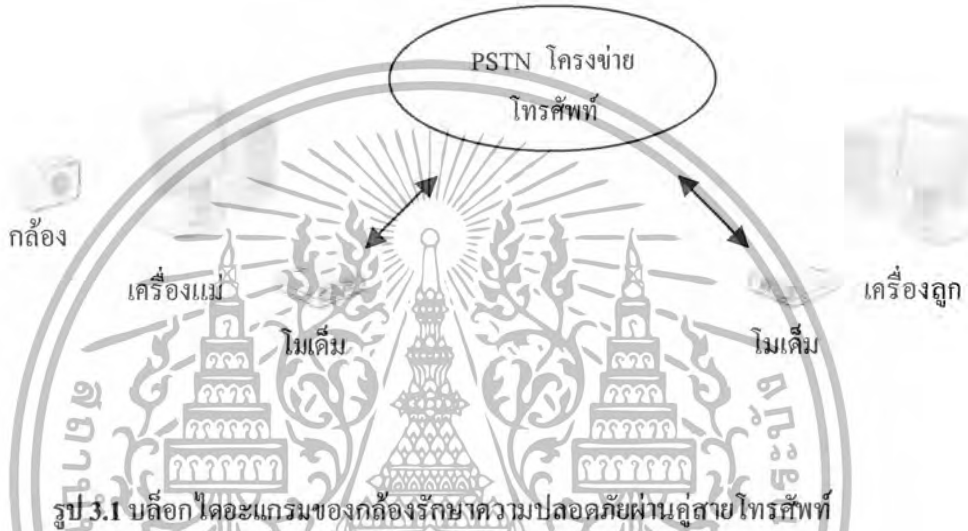
โปรแกรม Windows Media Player เป็นโปรแกรมของบริษัทไมโครซอฟท์ที่มีติดตั้งมาพร้อมกับ OS Windows ตั้งแต่ Version Windows 98 ขึ้นไป เป็นโปรแกรมที่สามารถแสดงข้อมูล Multimedia ในรูปของวีดีโอ หรือข้อมูลภาพนิ่งพร้อมสัญญาณเสียง ที่มีการบีบอัดข้อมูลในลักษณะ MPEG-4 สามารถเลือกกำหนดอัตราความเร็วของการส่งข้อมูลจากเวบเซิร์ฟเวอร์ได้หลายอัตราความเร็ว เช่นตั้งแต่ 28.8 Kb/s จนถึง 100 Mb/s จึงทำให้สามารถเข้าถึงข้อมูล Multimedia ในเว็บ เซิร์ฟเวอร์ ได้ตามความต้องการความเร็วของผู้ใช้

ดังนั้นหากเราดัดแปลงวิธีการควบคุมภาพนิ่ง เสียงบรรยายประกอบ และวีดีโอ ในลักษณะของ Streaming Files โดยให้ปรากฏผ่านโฮมเพจด้วยบราวเซอร์ ของ IE และใช้คำสั่ง Java ให้ผู้ใช้ที่เข้าผ่านอินเตอร์เน็ตสามารถควบคุมภาพที่ปรากฏ ตลอดจนทำให้เกิดการตอบโต้ระหว่างผู้ใช้กับ โฮมเพจได้แล้ว เราจะสามารถออกแบบและสร้างโฮมเพจที่ใช้ประกอบการเรียนการสอน การบรรยาย ตลอดจนเป็นระบบ Web Based Instruction สำหรับการพัฒนาบทเรียนผ่านเครือข่ายอินเตอร์เน็ต หรือจะใช้เป็นลักษณะแผ่น CD-ROM สำหรับเรียนแบบ Off-Line ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
การคำนวณและการสร้าง

โครงงานนี้มีจุดประสงค์คือการรักษาความปลอดภัยโดยการส่งข้อมูลรูปภาพจากเครื่องแม่ (Server) ไปยังเครื่องลูก (Client) ผ่านโครงข่ายโทรศัพท์ สร้างโดยการใช้โปรแกรมเดลไฟล์ เป็นตัวพัฒนา และได้ออกแบบเป็น 2 โปรแกรมคือโปรแกรมที่ติดตั้งบนเครื่องแม่ (Server) ซึ่งสร้างเป็นโปรแกรมชื่อ 'Project3TServer' และโปรแกรมที่ติดตั้งบนเครื่องลูก (Client) สร้างเป็นโปรแกรมชื่อ 'Project3TClient'



รูป 3.1 บล็อกไดอะแกรมของกล้องรักษาความปลอดภัยผ่านคู่สายโทรศัพท์

3.1 โปรแกรม Project3TServer

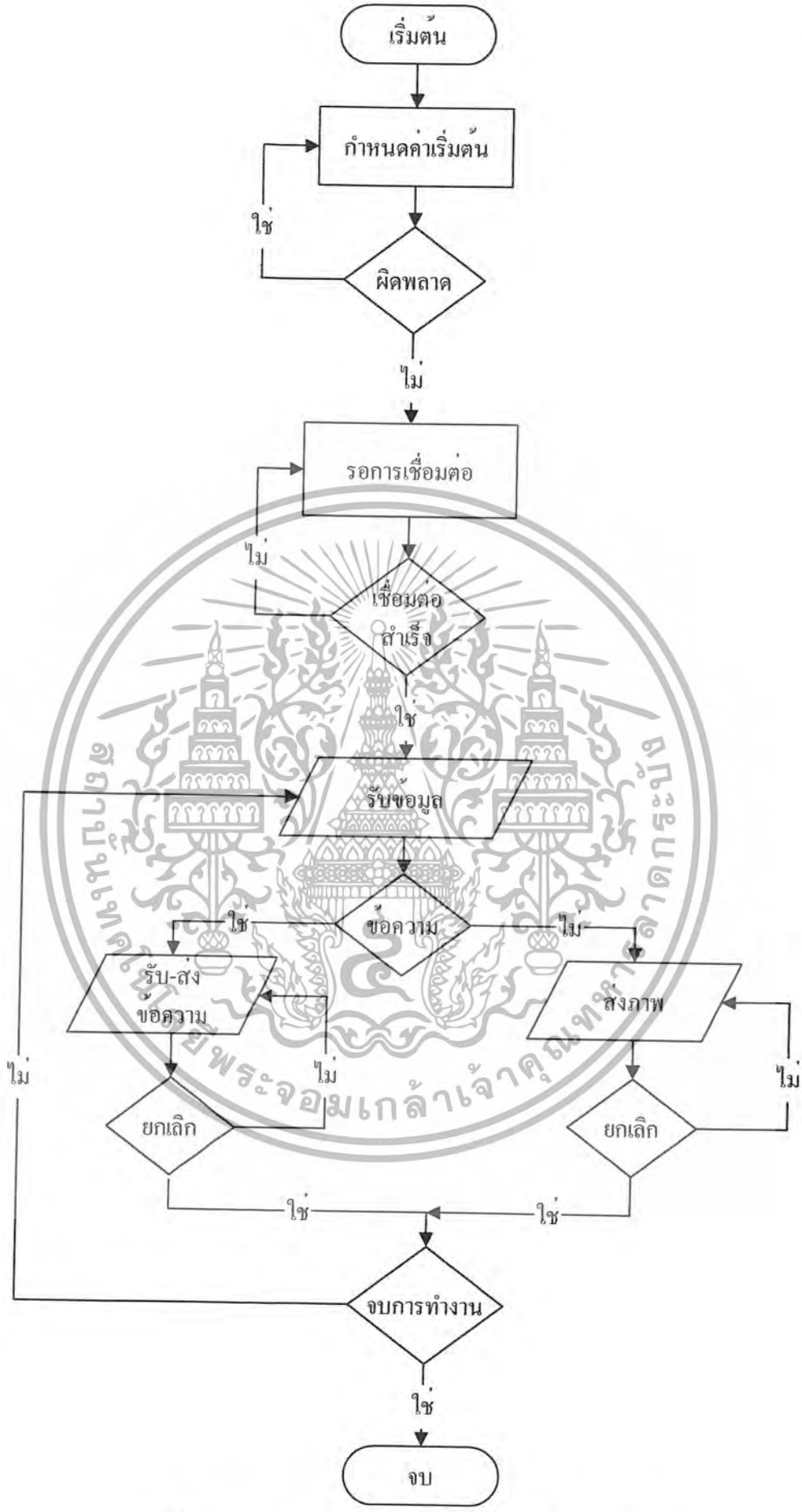
โปรแกรม Project3TServer เป็นโปรแกรมตอบรับการติดต่อจากเครื่องลูกโดยอัตโนมัติ ซึ่งตัวโปรแกรมสามารถรันพร้อมกับตอนเริ่มโปรแกรมวินโดว์ (window startup) เมื่อมีการติดต่อมาจากเครื่องลูกก็จะตอบรับและส่งภาพจากกล้องดิจิตอลไปยังเครื่องลูกโดยอัตโนมัติ การทำงานของโปรแกรมแสดงในโฟลชาร์ตดังรูปที่ 3.2

3.1.1 Component ที่สำคัญในโปรแกรม Project3TServer

1.ServerSocket เป็น Component ที่อยู่ใน Borland Delphi Visual Component Library ซึ่งปกติมีอยู่ในโปรแกรมเดลไฟล์ 5 และเป็นซ็อกเก็ตที่ทำงานบนโปรโตคอล TCP ซึ่งเป็นโปรโตคอลมาตรฐานที่ใช้สื่อสารกันระหว่างเครื่องคอมพิวเตอร์ ใน โปรแกรม Project3TServer ใช้ ServerSocket เป็นตัวสื่อสารข้อมูลที่เป็นข้อความกับ ClientSocket ในโปรแกรม Project3TClient และในการใช้งานนั้นมี Properties และ Events ที่สำคัญดังนี้

-Properties Port เป็นหมายเลขพอร์ตที่ใช้ในการติดต่อสื่อสารข้อมูลกับเครื่องลูกและจะต้องหมายเลขเดียวกันกับหมายเลขพอร์ตของ ClientSocket

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงแผนผังการทำงานของโปรแกรม Project 3Tserver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Events OnAccept เป็นกิจกรรมเมื่อเครื่องแม่ตอบรับการติดต่อจากเครื่องลูก

-Events OnClientDisconnect เป็นกิจกรรมเมื่อยกเลิกการติดต่อ

-Events OnClientError เป็นกิจกรรมเมื่อการเชื่อมต่อเกิดการผิดพลาด

-Events OnClientRead เป็นกิจกรรมเมื่อได้รับข้อมูลที่ส่งมาจากเครื่องลูก

2. ABF Component เป็นชุด Component ที่ใช้ในการควบคุมการทำงานของโปรแกรมซึ่งประกอบด้วย

- abfAutoRun เป็น Component ที่ใช้เพื่อให้โปรแกรมรันพร้อมกับโปรแกรมวินโดว์โดยมีหลักการทำงานคือการเข้าไปแก้ไขค่าใน Registry คือใส่พาร์ท(Path)ของโปรแกรมใน HKEY_CURRENT_USER , Software , Microsoft , Windows , Run

- abfShutdown เป็น Component ที่ใช้เพื่อปิดหรือรีเซ็ตเครื่องคอมพิวเตอร์เมื่อโปรแกรม Project3T Server เกิดความผิดพลาดในส่วน

ของการส่งข้อมูลรูปภาพเพื่อให้โปรแกรมพร้อมที่จะรอรับการติดต่อจากเครื่องลูกอีกครั้ง

3. JLCVideo เป็น Component ที่ใช้เพื่อแสดงภาพจากกล้องดิจิตอลโดยสามารถปรับขนาดและคุณภาพของภาพได้และใช้ในการ Capture ภาพแล้วส่งไปยังเครื่องลูกซึ่งมีหลักการทำงานคือจะตรวจสอบการติดตั้งกล้องดิจิตอลภายในเครื่องคอมพิวเตอร์จากนั้นก็จะนำภาพมาแสดงบนจอของJLCVideo โดยจะทำงานร่วมกันกับไดรเวอร์(Driver)ของกล้องดิจิตอล

4. NMStrm เป็น Component ของ Netmaster L.L.C ซึ่งมีมาพร้อมกับโปรแกรมเดลไฟ 5 อยู่แล้วโดยจะทำหน้าที่ส่งข้อมูลรูปภาพไปยังเครื่องลูกโดย

ทำงานอยู่บนโปรโตคอล TCP มีรูปแบบของคำสั่งที่ใช้ในการส่งข้อมูลคือ Server.PostIt(Data) และมี Properties และ Events ที่สำคัญดังนี้

- Properties Host เป็นชื่อของเครื่องคอมพิวเตอร์ที่เป็นเครื่องลูกเป็นข้อมูลชนิดสตริง(String) หรือจะใส่หมายเลข IP Address ของเครื่องลูกก็ได้

- Properties Port เป็นหมายเลขพอร์ตที่ใช้ในการติดต่อสื่อสารข้อมูลรูปภาพกับเครื่องลูกและจะต้องเป็นหมายเลขเดียวกันกับหมายเลขพอร์ตของ NMStrmServ

- Events OnConnect เป็นกิจกรรมเมื่อทำการติดต่อส่งข้อมูลรูปภาพไปยังเครื่องลูก

- Events OnConnectFailed เป็นกิจกรรมเมื่อการเชื่อมต่อเกิดความผิดพลาด

- Events OnDisconnect เป็นกิจกรรมเมื่อยกเลิกการส่งข้อมูลภาพไปยังเครื่องลูก

- Events OnPacketSent เป็นกิจกรรมเมื่อทำการส่งไฟล์รูปภาพไปยังเครื่องลูก

3.1.2 การทำงานของโปรแกรม Project3TServer แต่ละส่วน

- การรับส่งข้อความในการรับส่งข้อความทั้งที่เป็นข้อความและข้อความคำสั่งนั้นจะกระทำโดยใช้ Server Socket เป็นตัวดำเนินการทั้งการรับ และการส่งโดย ServerSocket จะรอรับการติดต่อโดยอัตโนมัติในตอนเริ่มรันโปรแกรม เมื่อมีการร้องขอการติดต่อจาก ClientSocket ในโปรแกรม Project3TClient ก็จะตอบรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อและพร้อมที่จะรับส่งข้อมูลที่เป็นข้อความและข้อความคำสั่งเพื่อดำเนินการตามคำสั่งเช่นการส่งข้อมูลรูปภาพหรือการรีเซ็ตเครื่องคอมพิวเตอร์ เป็นต้น

- การส่งข้อมูลภาพ ในการส่งข้อมูลภาพไปยังเครื่องลูกนั้นจะดำเนินการโดย NMStrm โดยจะต้องจะทำการส่งภาพไปเมื่อเครื่องลูกส่งข้อความมาว่า "ready"และจะหยุดส่งส่งภาพไปเมื่อเครื่องลูกส่งข้อความมาว่า "stop"

- ส่วนควบคุมโปรแกรม ในส่วนของการควบคุมโปรแกรมนั้นจะทำหน้าที่ป้องกันตัวโปรแกรมจากเหตุการณ์ผิดพลาดต่างๆ ซึ่งประกอบด้วยการสร้างรหัสเพื่อเข้าไปจัดการควบคุมคำสั่งพิเศษและรหัสที่สร้างนั้นจำเป็นต้องใช้ในการปิดโปรแกรม Project3TServer ด้วยเพราะตอนปิดโปรแกรมนั้นจะต้องใส่รหัสที่สร้างในตอนแรกให้ถูกต้องมิฉะนั้นจะไม่สามารถปิดโปรแกรมได้และในคำสั่งพิเศษภายในนั้นสามารถสั่งให้โปรแกรมล็อคคีย์บอร์ด ล็อคปุ่ม Start ของวินโดว์หรือจะทำการรีเซ็ตเครื่องคอมพิวเตอร์ก็ได้

3.2 โปรแกรม Project3TClient

โปรแกรม Project3TClient เป็นโปรแกรมที่ติดตั้งบนเครื่องลูกทำหน้าที่เชื่อมต่อไปยังเครื่องแม่โดยโมเด็ม หลังจากนั้นจะร้องขอการติดต่อทางโปรโตคอล TCP เพื่อรับส่งข้อมูลข้อความ ข้อมูลคำสั่ง ข้อมูลรูปภาพจากกล้องดิจิตอล และเก็บไฟล์ภาพที่ต้องการการ ทำงานของโปรแกรมแสดงได้ในโฟลชาร์ตดังรูปที่ 3.3

3.2.1 Component ที่สำคัญในโปรแกรม Project3TClient

1. ClientSocket เป็น Component ที่อยู่ใน Borland Delphi Visual Component Library ซึ่งปกติมีอยู่ในโปรแกรมเดลไฟ 5และเป็นซ็อกเก็ตที่ทำงานบนโปรโตคอล TCP ซึ่งเป็นโปรโตคอลมาตรฐานที่ใช้สื่อสารกันระหว่างเครื่องคอมพิวเตอร์ใน โปรแกรม Project3TClient ใช้ ClientSocket เป็นตัวสื่อสารข้อมูลที่เป็นข้อความกับ ServerSocket ในโปรแกรม Project3TServerและในการใช้งานนั้นมี Properties และ Events ที่สำคัญดังนี้

-Properties Host เป็นชื่อของเครื่องคอมพิวเตอร์ที่เป็นเครื่องแม่เป็นข้อมูลชนิดสตริง(String) หรือจะใส่หมายเลข IP Address ของเครื่องแม่ก็ได้

-Properties Port เป็นหมายเลขพอร์ตที่ใช้ในการติดต่อสื่อสารข้อมูลกับเครื่องแม่และจะต้องเป็นหมายเลขเดียวกันกับหมายเลขพอร์ตของ ServerSocket

-Events OnConnect เป็นกิจกรรมเมื่อทำการติดต่อไปยังเครื่องแม่

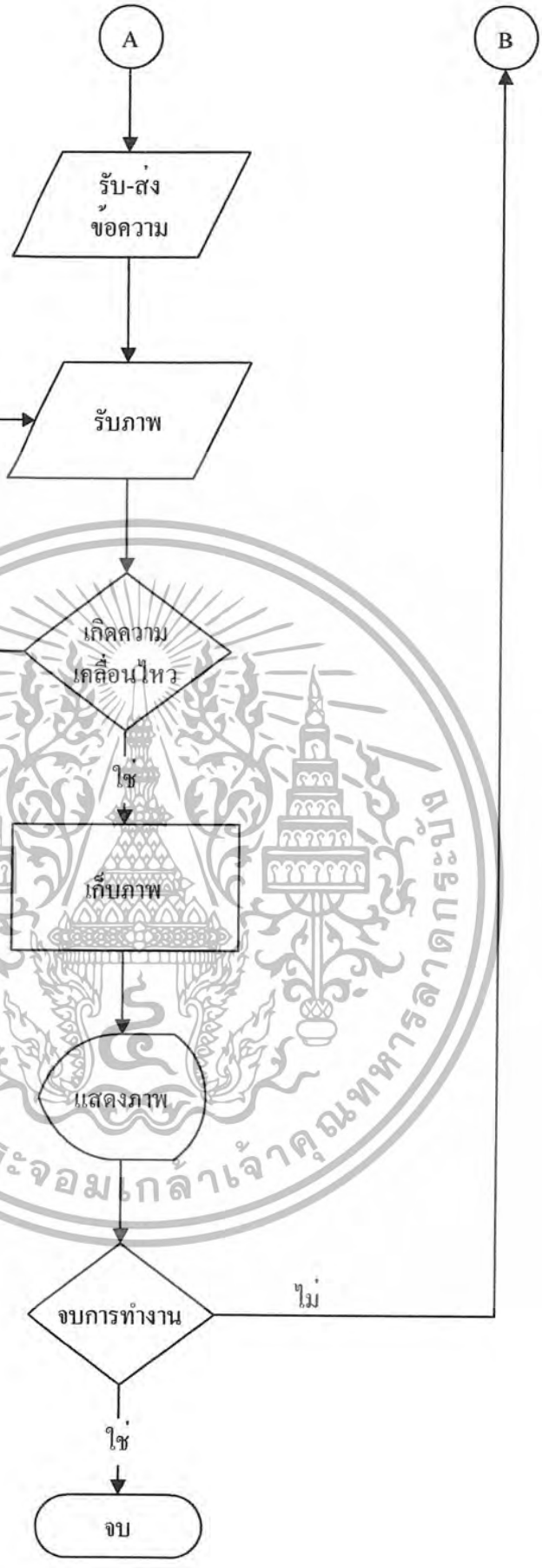
-Events OnConnecting เป็นกิจกรรมในขณะเชื่อมต่ออยู่กับเครื่องแม่

-Events OnDisconnect เป็นกิจกรรมเมื่อยกเลิกการเชื่อมต่อไปยังเครื่องแม่

-Events OnError เป็นกิจกรรมเมื่อเกิดความผิดพลาดในการเชื่อมต่อ

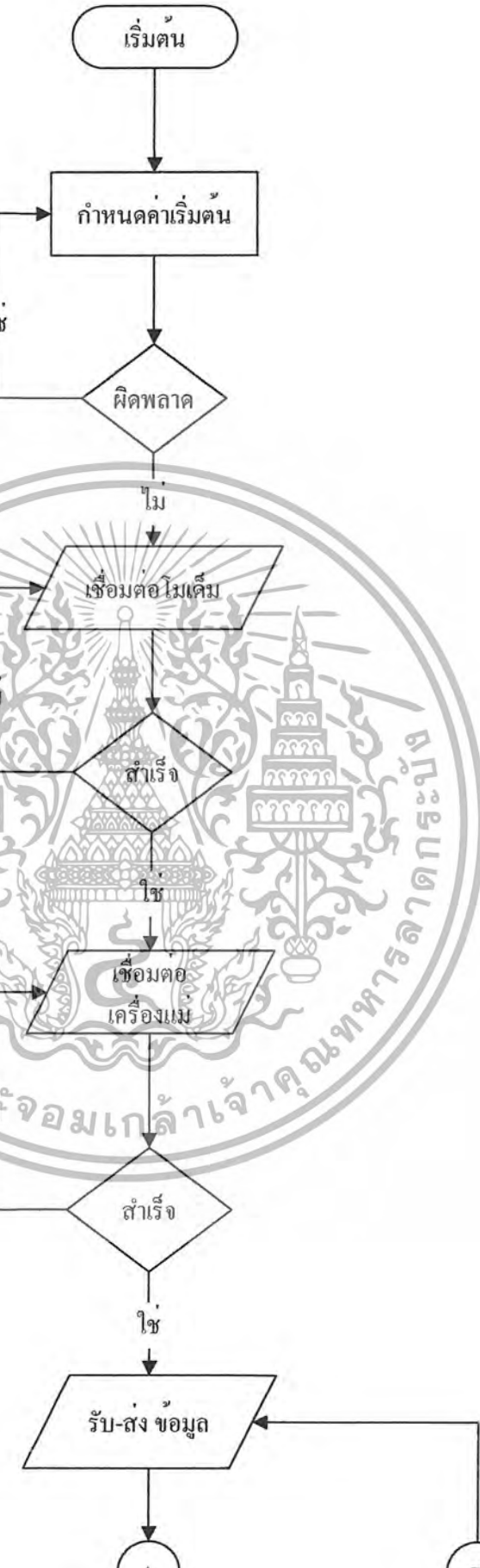
-Events OnRead เป็นกิจกรรมเมื่อได้รับข้อมูลที่ส่งมาจากเครื่องแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงแผนผังการทำงานของโปรแกรม Project 3Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. NMStrmServ เป็น Component ของ Netmaster L.L.C ซึ่งมีมาพร้อมกับโปรแกรมเดลไฟ 5 อยู่แล้วโดยจะทำหน้าที่รับข้อมูลรูปภาพที่ส่งมาจากเครื่องแม่โดยทำงานอยู่บนโปรโตคอล TCP มี Properties และ Events ที่สำคัญดังนี้

-Properties Port เป็นหมายเลขพอร์ตที่ใช้ในการรับข้อมูลรูปภาพที่ส่งมาจากเครื่องแม่และจะต้องเป็นหมายเลขเดียวกันกับหมายเลขพอร์ตของ NMStrm

-Events OnClientContact เป็นกิจกรรมเมื่อมีการตรวจพบการส่งข้อมูลมาจากเครื่องแม่

-Events OnDisconnect เป็นกิจกรรมเมื่อยกเลิกการเชื่อมต่อ

-Events OnMSG เป็นกิจกรรมเมื่อรับข้อมูลภาพที่ส่งมาจากเครื่องแม่

3. Ras.pas เป็นซอสโค้ดที่ใช้งานในส่วนของการเชื่อมต่อโดยโมเด็มจากเครื่องลูกไปยังเครื่องแม่ โดย Ras.pas จะเป็นตัวประสานการทำงานระหว่างส่วนของ Remote Access external API ของวินโดว์กับโปรแกรมเดลไฟ ทำให้โปรแกรมสามารถเซ็ตค่าต่างๆของโมเด็มและโปรแกรม Dialup Connection ของวินโดว์ได้ ทำให้สะดวกในการใช้งาน

3.2.2 การทำงานของโปรแกรม Project3TClient แต่ละส่วน

- การเชื่อมต่อโดยโมเด็ม มีหน้าที่ทำการหมุนโมเด็มเชื่อมต่อไปยังเครื่องแม่ซึ่งโปรแกรมในส่วนนี้สามารถสร้างหรือแก้ไขข้อมูลการเชื่อมต่อต่างๆที่ใช้ในการเชื่อมต่อไปยังเครื่องแม่

-พอร์ตและการควบคุม หลังจากการเชื่อมต่อโดยโมเด็มไปยังเครื่องแม่เรียบร้อยแล้ว ก็จะสามารถใช้งานในส่วนนี้คือจะทำการเชื่อมต่อไปยังเครื่องแม่โดยใช้ ClientSocket เพื่อรับส่งข้อมูลข้อความกับเครื่องแม่ และในส่วนของการควบคุมนั้นจะต้องสร้างรหัสแล้วนำรหัสนั้นไปปลดล็อคปุ่มสั่งการเพื่อสั่งให้รีเซ็ตเครื่องแม่ในกรณีที่เกิดข้อผิดพลาดของกล้องดิจิตอลหรือโปรแกรมในส่วนที่ส่งข้อมูลรูปภาพของเครื่องแม่ ซึ่งมีหลักการคือเมื่อกดปุ่ม Restart แล้วโปรแกรมจะส่งข้อความ "Restart" ไปยังเครื่องแม่ เมื่อโปรแกรมในเครื่องแม่พบว่าข้อความที่ส่งมาคือ "Restart"เครื่องแม่ก็จะทำการรีเซ็ตเครื่องเพื่อรอรับการติดต่ออีกครั้ง

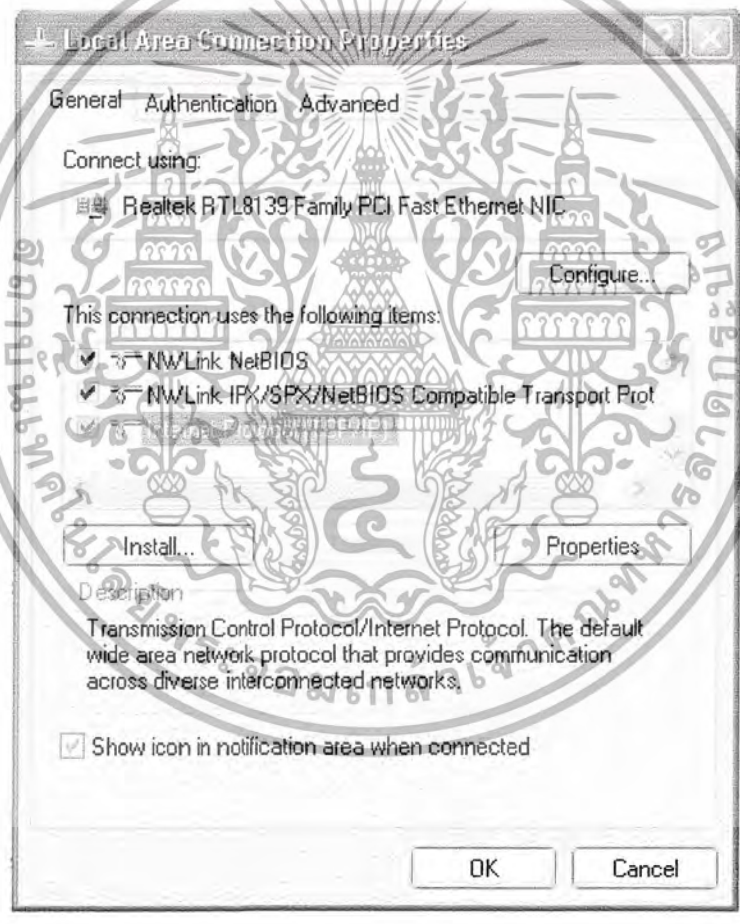
-การแสดงภาพในส่วนของการแสดงภาพจะใช้งานได้ก็ต่อเมื่อมีการเชื่อมต่อไปยังเครื่องแม่ในส่วนของพอร์ตและการควบคุมแล้วเท่านั้น เมื่อเชื่อมต่อแล้วก็จะสามารถใช้งานปุ่ม รับภาพ ได้ เมื่อเรากดปุ่ม รับภาพ โปรแกรมก็จะส่งข้อความ "ready" ไปยังเครื่องแม่เพื่อแสดงว่าพร้อมที่จะรับภาพจากนั้นเครื่องแม่ก็จะส่งข้อมูลภาพมาแล้วจะแสดงภาพในส่วนนี้ และโปรแกรมในส่วนนี้สามารถที่จะเก็บภาพที่เราต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองเชื่อมต่อและการ รับ – ส่ง ข้อมูล การใช้งานโปรแกรมการเตรียมการก่อนการใช้โปรแกรม รวมถึงการติดตั้งโปรแกรมด้วย

การเชื่อมต่อของโปรกรมเป็นการเชื่อมต่อโดยโปรโตคอล TCP/IP ดังนั้นจึงต้องทำการติดตั้งโปรโตคอล TCP/IP นี้ทั้งเครื่องแม่ (Sever) และเครื่องลูก (Client)



รูปที่ 4.1 โปรโตคอล TCP/IP

4.1 การเตรียมการใช้งานของเครื่องคอมพิวเตอร์ก่อนการใช้งานโปรแกรมที่สร้างขึ้นมา

4.1.1 การเตรียมการใช้งานของเครื่องแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

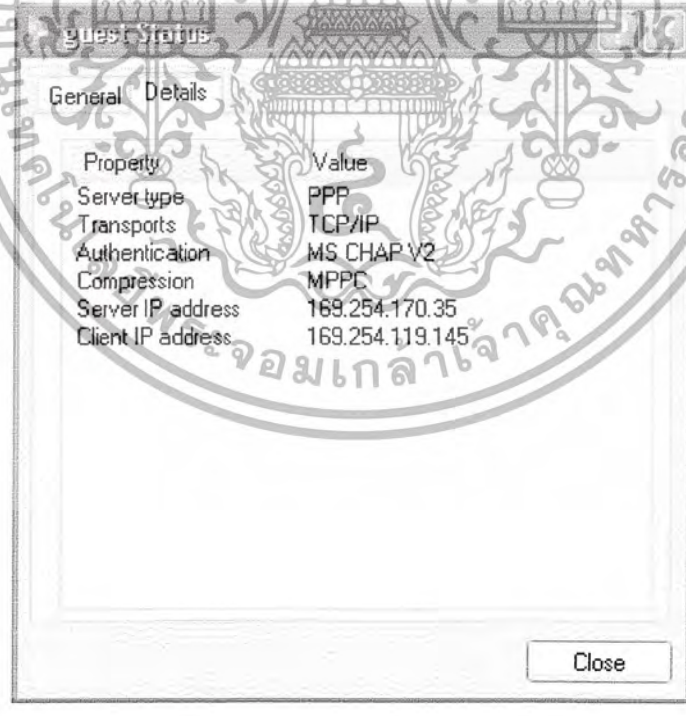
เนื่องจากการเชื่อมต่อทางโมเด็มต้องมีเครื่องที่รับการเรียกเข้าของโมเด็มโดยอัตโนมัติทางเครื่องแม่ ความต้องการที่ต้องมีคือใช้ระบบปฏิบัติการ Windows Xp Professional โดยการสร้าง Incoming Connections

รูปที่ 4.2 Incoming Connections

เมื่อมีการเชื่อมต่อเข้ามาทางโมเด็มเครื่องแม่ก็จะตอบรับอัตโนมัติโดยจะมีหน้าจอการเชื่อมต่อกันเข้ามาใน Network Connection เราสามารถทำการทุกอย่างได้เหมือนอยู่ในระบบ LAN ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 การตอบรับ Incoming Connection

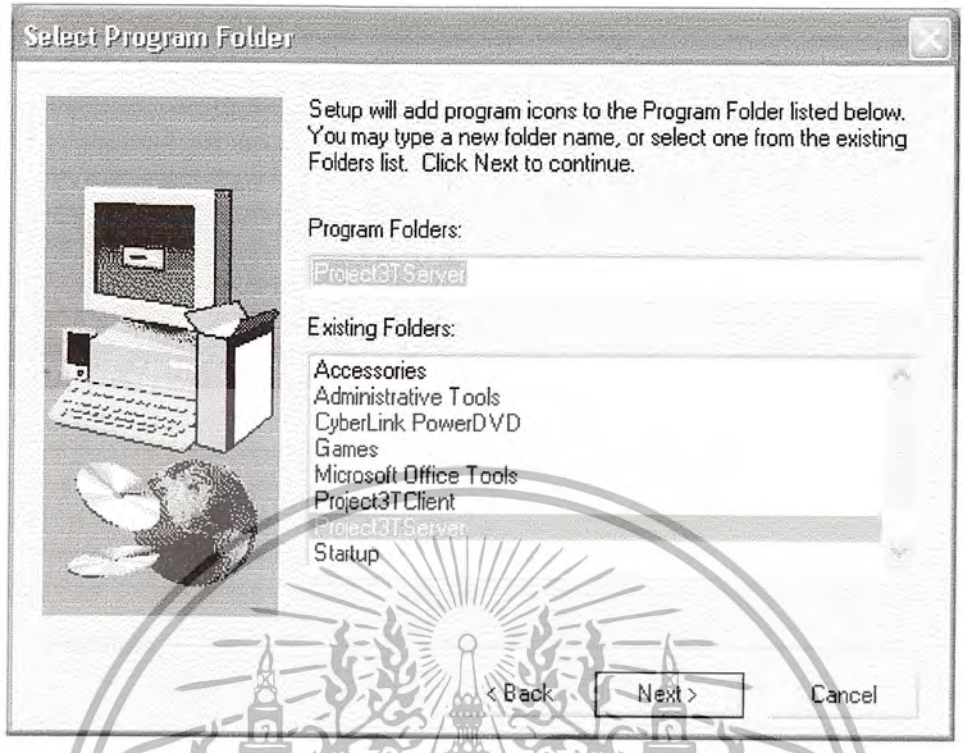


รูปที่ 4.4 Status ของ Incoming Connection

4.1.2 การติดตั้งและวิธีการใช้งานโปรแกรม Project3TSever

ในการทดลองการใช้งานโปรแกรมนี้จำเป็นต้องเริ่มจากการติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

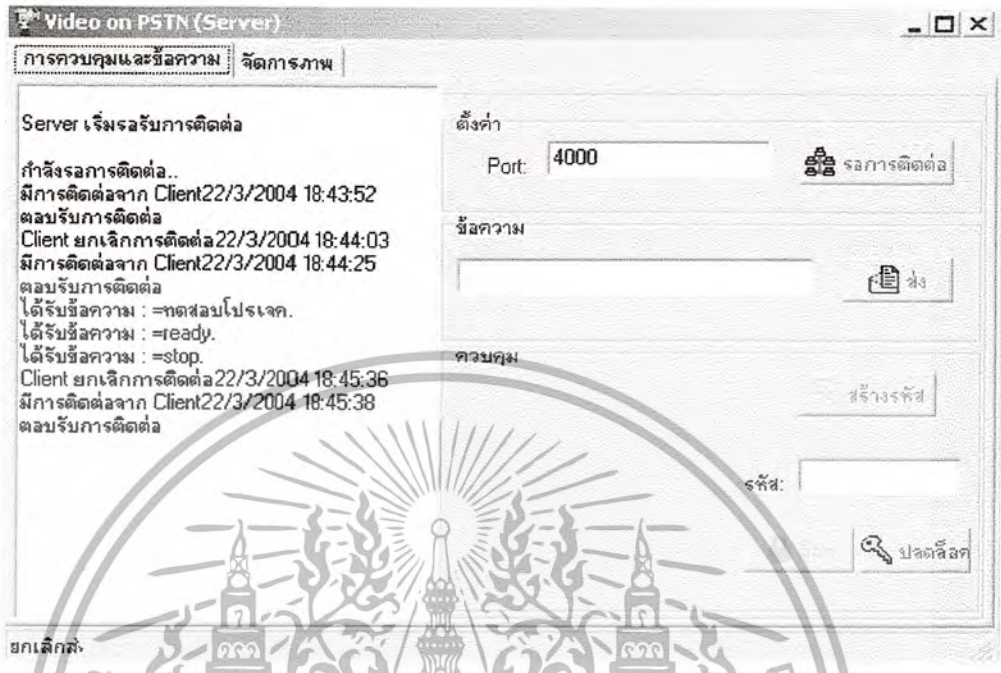


รูปที่ 4.5 การติดตั้งโปรแกรม Project3TSever

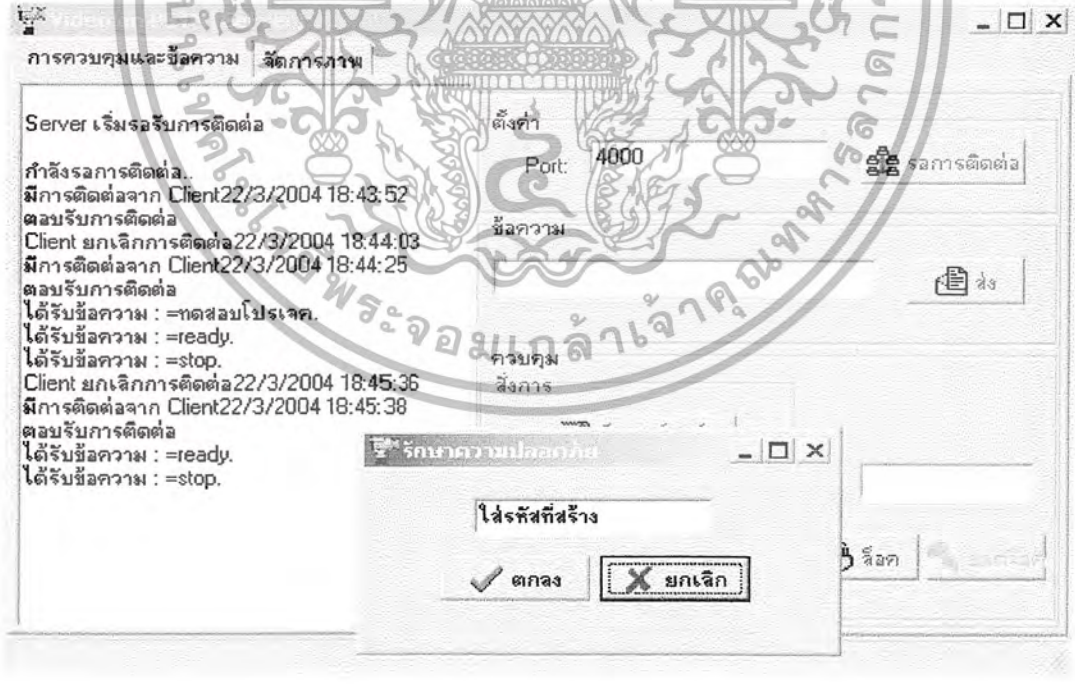
การติดตั้งโปรแกรมเราสามารถเลือกไฟล์เดอร์และไดร์ฟที่อยู่ของโปรแกรมได้ เมื่อติดตั้งโปรกรมแล้วหากมีการเปิดเครื่องขึ้นมาใหม่หรือมีการรีสตาร์ทเครื่อง โปรแกรมจะเปิดขึ้นมาและรอการเชื่อมต่อเอง โดยอัตโนมัติเพื่อว่าหากเกิดเหตุขัดข้องที่เครื่องแม่ก็จะได้ทำการรีสตาร์ทใหม่ได้จากโปรแกรมของเครื่องลูก แล้วจึงจะเชื่อมต่อเข้าไปใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 หน้าจอหลักของโปรแกรมเมื่อยังไม่มีการเชื่อมต่อ

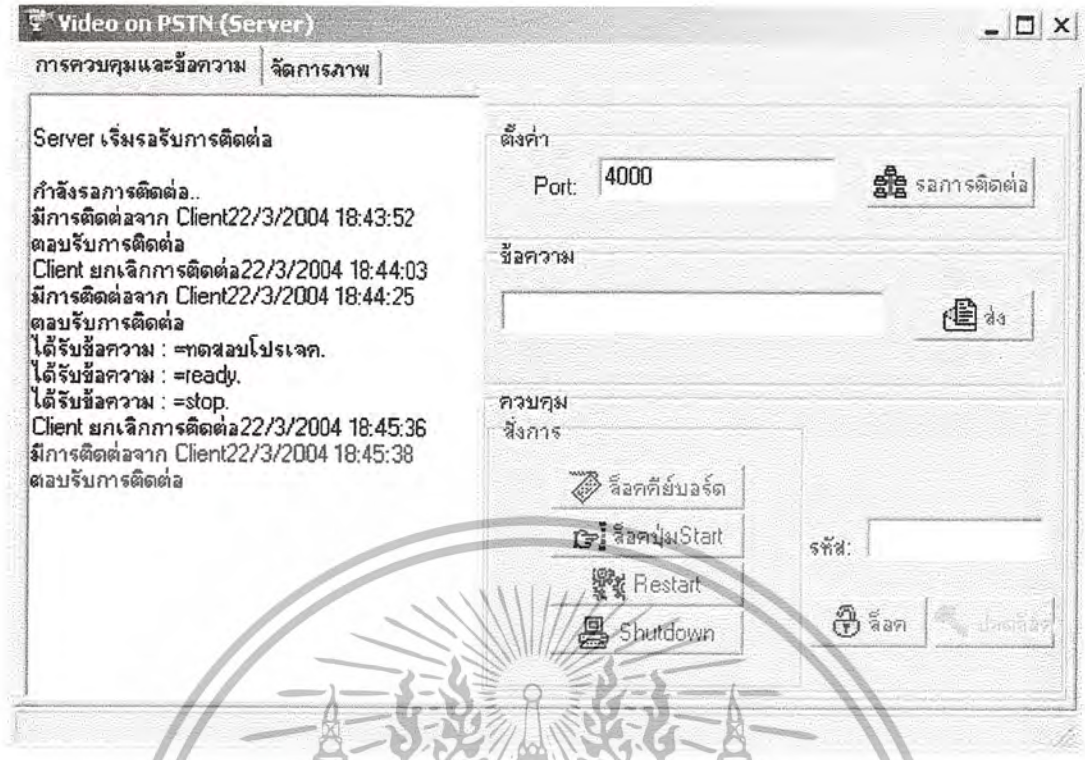


รูปที่ 4.7 หน้าจอหลักของโปรแกรมเมื่อมีการเชื่อมต่อ



รูปที่ 4.8 การสร้างรหัสเพื่อควบคุมการใช้งาน

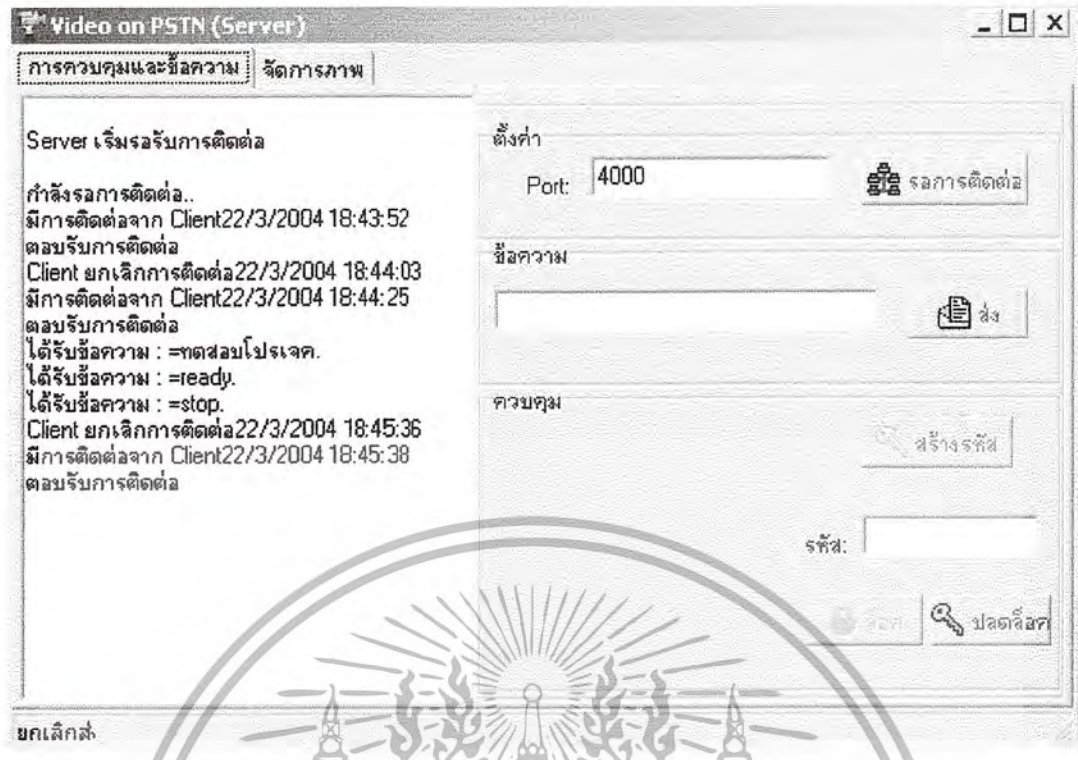
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 หน้าจอควบคุมการใช้งานหลังจากสร้างรหัสแล้ว

จากรูปที่ 4.9 เมื่อมีการสร้างรหัสมาแล้วเราสามารถที่จะควบคุมการใช้งานของ Windows ได้เพื่อป้องกันบุคคลที่จะมาทำการแก้ไขเครื่องเมื่อเราติดตั้ง เราก็สามารถป้องกันโดยการล็อกคีย์บอร์ดและปุ่ม Start ไว้เมื่อเราล็อกแล้วจะไม่สามารถใช้คีย์บอร์ดและปุ่ม Start ได้และหากต้องการจะปลดล็อกก็ต้องใส่รหัสตามที่เราสร้างไว้ในตอนแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าจอควบคุมการใช้งานเมื่อมีการล็อก

หน้าจอการใช้งาน การควบคุมและข้อความ มีการใช้งานคือหน้าจอทางซ้ายแสดงสถานะการใช้งานการติดต่อและข้อความที่มีการสื่อสารกันเมื่อมีการเชื่อมต่อก็จะแสดงเวลาการเชื่อมต่อและยกเลิกการเชื่อมต่อ

หน้าจอทางขวามี 3 ส่วนคือ

1. หน้าจอตั้งค่ามีการแสดงและกำหนดพอร์ตติดต่อและปุ่มรอการติดต่อไว้เมื่อเราจะทำการติดต่อกันครั้งใหม่เมื่อมีการยกเลิกการติดต่อ

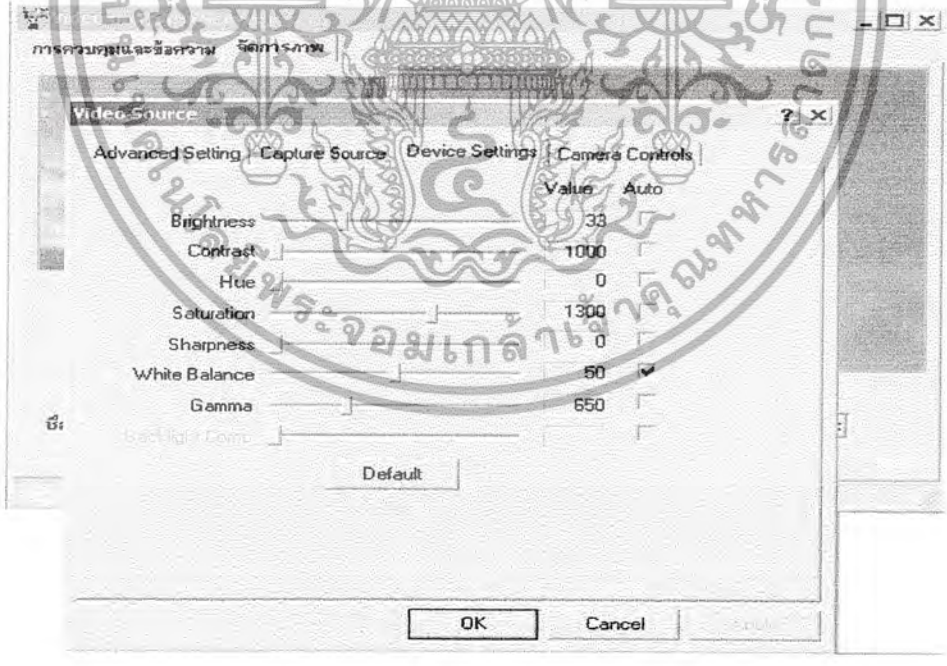
2. ช่องข้อความมีไว้เพื่อที่จะส่งข้อความหากันโดยพิมลงในช่องแล้วกดปุ่ม ส่ง ก็จะส่งข้อความหากันได้

3. ช่องควบคุมมีไว้ทำการล็อกการใช้งานและสร้างรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.11 หน้าจอควบคุมการส่งภาพ

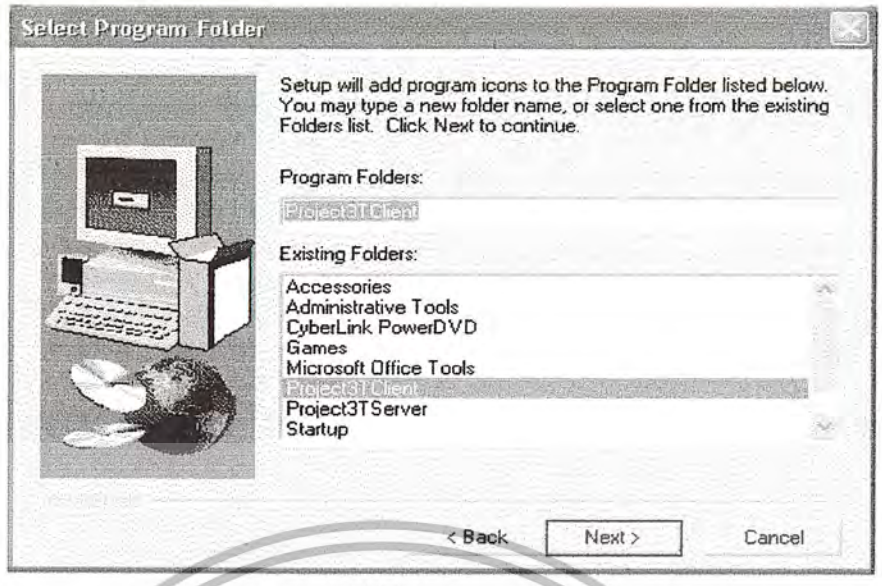
หน้าจอการใช้งาน จัดการภาพ มีการใช้งานคือจัดการกับข้อมูลภาพมีการเซ็ตค่า กำหนดขนาดภาพ การส่งภาพ กำหนดจำนวนเฟรมการส่งภาพไปให้เครื่องลูก



รูปที่ 4.12 หน้าจอการเซ็ตค่าของภาพ

4.2 การติดตั้งและการใช้งานของเครื่องลูก (Client)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



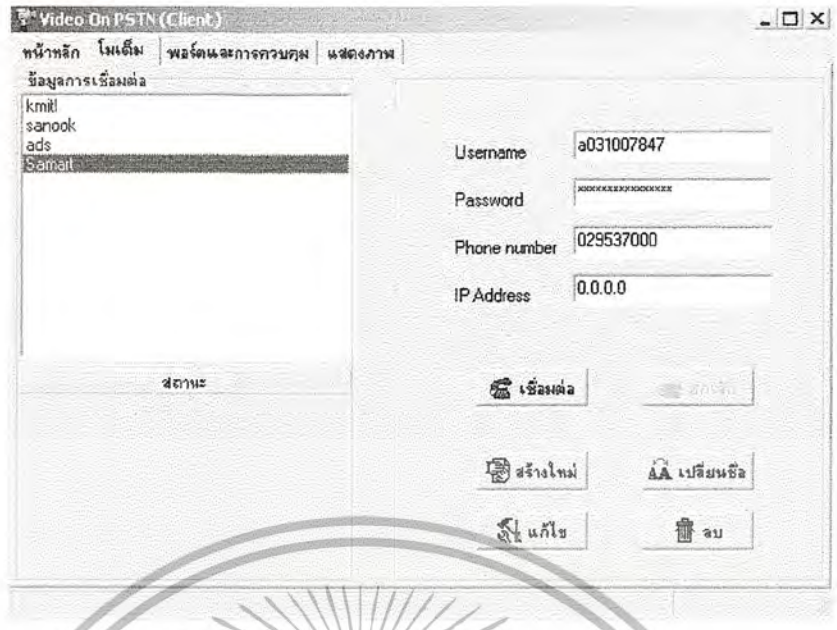
รูปที่ 4.13 การติดตั้งโปรแกรม Project3TClient

การติดตั้งโปรแกรมเราสามารถเลือกไฟล์เดอร์และไดร์ฟที่อยู่ของโปรแกรมได้เมื่อเราจะเรียกใช้เราสามารถเรียกจากปุ่มสตาร์ตได้

รูปที่ 4.14 หน้าจอหลักของเครื่องลูก

หน้าจอหลักมีหน้าอแสดงสถานะการเชื่อมต่อไปยังเคื่องแม่เวลาที่เชื่อมต่อและเวลาการยกเลิกการเชื่อมต่อ หน้าจอทางฝั่งขวามีเลขแสดงเวลาหรือนาฬิกานั่นเอง

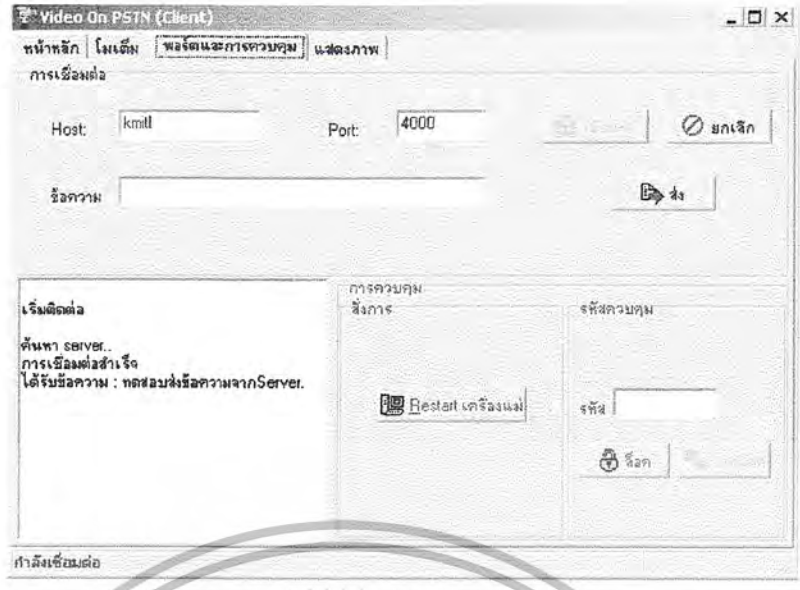
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 หน้าจอโมเด็ม

จากรูปที่ 4.15 แสดงหน้าจอของโมเด็มโดยมี Dial-up Connection ที่เราสร้างไว้ใน Network Connection อยู่ในช่องของข้อมูลการเชื่อมต่อ ทางขวาก็จะมีสถานะของ Dial-up Connection แสดงไว้นั่นคือ Username Password Phone number และส่วนของปุ่มควบคุมก็มีไว้สำหรับการเชื่อมต่อและเมื่อเชื่อมต่อสำเร็จก็จะสามารถเชื่อมต่อเข้ากับเครื่องแม่และดูภาพจากเครื่องแม่ได้ต่อไป

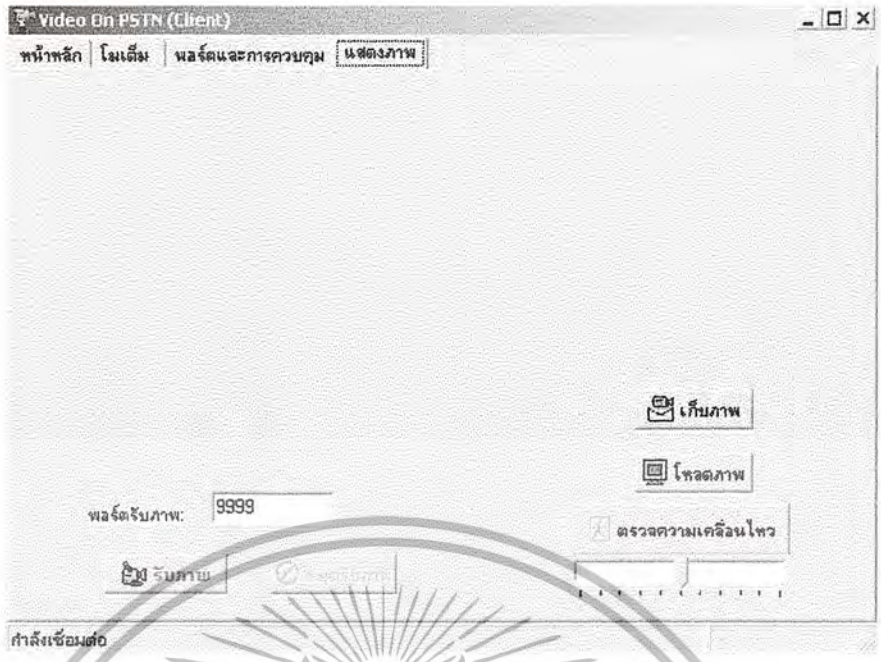
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



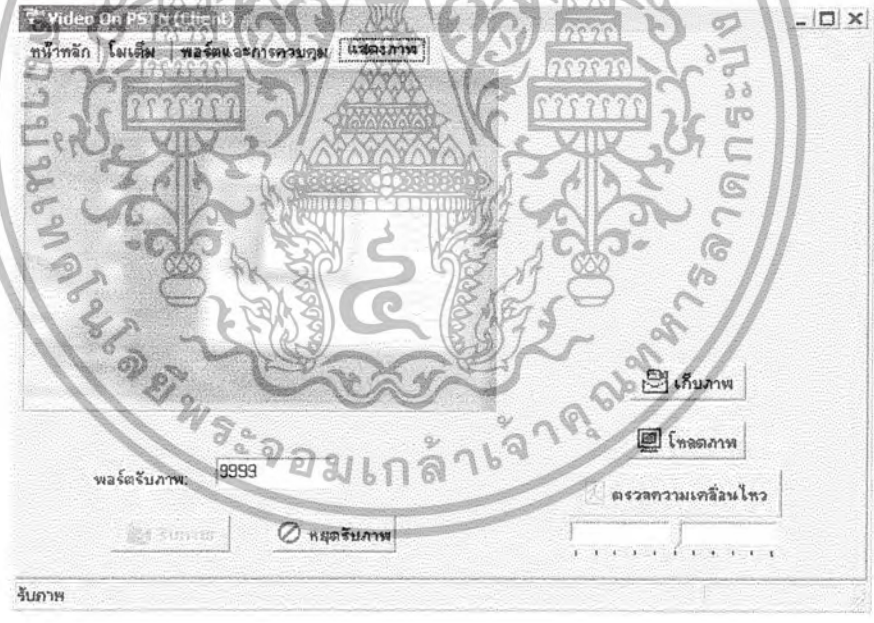
รูปที่ 4.16 หน้าจอพอร์ตและการควบคุม

สำหรับหน้าจอพอร์ตและการควบคุมจะทำการเชื่อมต่อเครื่องแม่โดยในช่อง Hose ก็จะใส่ชื่อเครื่องแม่หรือ ไอพีแอดเดรส ของเครื่องแม่แล้วกดปุ่มเชื่อมต่อก็จะเชื่อมต่อกับเครื่องแม่ได้ และยกเลิกการเชื่อมต่อโดยปุ่มยกเลิก ในช่องของข้อความก็จะมีเหมือนในเครื่องแม่ไว้คุยกันได้สามารถสื่อสารกันโดยพิมพ์ข้อความแล้วกดปุ่ม ส่ง ในส่วนของการแสดงผลก็จะมีสถานะของการเชื่อมต่อและข้อความที่รับจากเครื่องแม่ และยังส่วนของการควบคุมและการสร้างรหัสเหมือนในเครื่องแม่แต่จะมีส่วนของ ปุ่ม Restart เครื่องแม่ ไว้หากมีการผิดพลาดหรือ error ขึ้นก็ Restart เครื่องแม่ ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

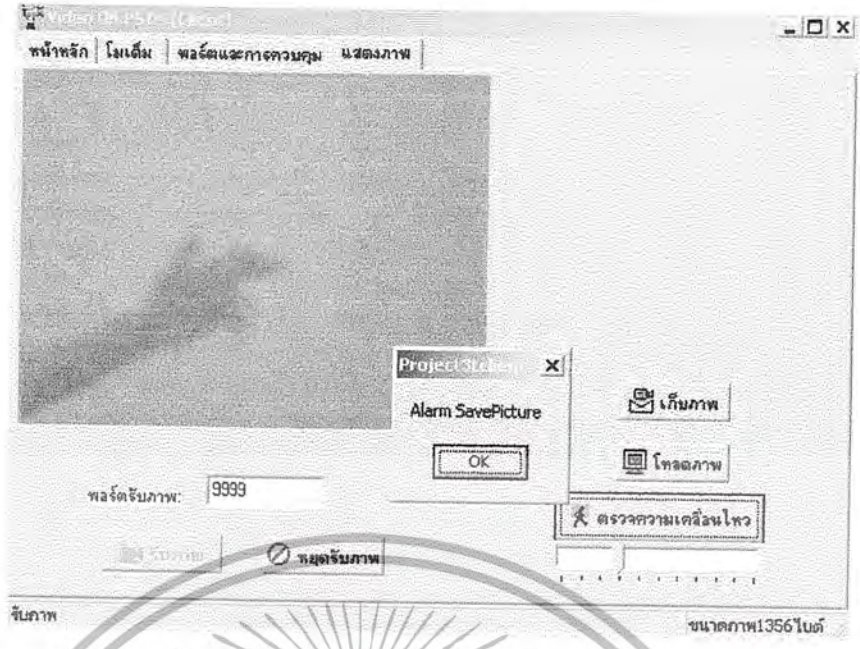


รูปที่ 4.17 หน้าจอแสดงภาพก่อนที่จะรับภาพ



รูปที่ 4.18 หน้าจอแสดงภาพเมื่อมีการรับภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 หน้าจอแสดงภาพเมื่อเกิดการเปลี่ยนแปลงของภาพ

รูปที่ 4.20 หน้าจอแสดงภาพหลังจากเก็บภาพความเคลื่อนไหวไว้ได้

การทดลองการเชื่อมต่อเครื่องลูกไปยังเครื่องแม่เมื่อเชื่อมต่อกันได้แล้วก็สามารถเชื่อมต่อข้อมูลที่เป็นภาพได้หรือว่าคือการดูภาพเครื่องแม่จากเครื่องลูกและเมื่อมีความเคลื่อนไหวโปรแกรมก็จะจับภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และบันทึกไว้เพื่อเป็นข้อมูลหรือหลักฐานได้สำหรับหน้าจอแสดงภาพนี้ก็จะมีส่วนของการแสดงภาพที่รับมาจากเครื่องแม่และส่วนของภาพที่บันทึกไว้ได้ในขณะที่มีความเคลื่อนไหว นอกจากนั้นแล้วก็ยังมีปุ่มควบคุมต่าง ๆ อันได้แก่การรับภาพ หยุดรับภาพ เก็บภาพ โหลดภาพ และตรวจความเคลื่อนไหวและก็มีการกำหนดขนาดภาพได้ด้วยแถบควบคุมด้านล่างซ้ายของหน้าจอ เราสามารถที่จะรับภาพจากเครื่องแม่และยังตรวจจับความเคลื่อนไหวแล้วบันทึกไว้ได้โดยเครื่องลูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

สรุปผลการทดลอง

จากการทดลองเครื่องแม่ (Server) และเครื่องลูก (Client) สามารถติดต่อกันได้ทางโมเด็มโดยการเซ็ตเครื่องแม่ให้ทำการรับการเรียกจากโมเด็ม ซึ่งเป็นการใช้คุณสมบัติของ Windows Xp โดยการทดลองการเชื่อมต่อจะทำการทดลองเป็นสองขั้นตอนคือ

1. การติดต่อระหว่างเครื่องแม่กับเครื่องลูกผ่านโมเด็มให้เชื่อมต่อและเห็นกันในบราวน์เซอร์
2. การเชื่อมต่อกันในทางโปรแกรมเพื่อที่จะทำการส่งภาพและข้อความไปหากัน และยังสามารถควบคุมอื่น ๆ ที่โปรแกรมได้สร้างไว้ได้

ทำการทดลองในส่วนของการจับความเคลื่อนไหวที่เกิดขึ้นโดยให้เก็บภาพขณะที่มีการเปลี่ยนแปลงให้บันทึกภาพไว้ได้เป็นไปตามจุดมุ่งหมายและในส่วนโปรแกรมได้มีส่วนที่เป็นการควบคุมการปฏิบัติการของเครื่องจากโปรแกรมที่สร้างขึ้นมาได้

ปัญหาที่พบจากการทดลอง

1. การเชื่อมต่อทางโมเด็มต้องใช้โมเด็มที่เหมือนกันไม่สามารถใช้โมเด็มที่ต่างกันได้
2. ความเร็วในการส่งข้อมูลที่มีขนาดใหญ่อย่างเช่น ภาพยังไม่เพียงพอ ภาพที่ออกมาจึงมีการติดขัดบ้าง
3. การใช้โปรแกรมยังมีการผิดพลาดหรือ error อยู่บ้างต้องทำการเซ็ตค่าต่าง ๆ ให้เหมาะสม

แนวทางในการพัฒนาโปรแกรม

1. พัฒนาเพิ่มการใช้งานในส่วนของการควบคุมเข้าไปอีก
2. พัฒนาการเตือนเมื่อเกิดการเปลี่ยนแปลงของภาพที่มีการเคลื่อนไหวเพื่อให้มีประสิทธิภาพในการรักษาความปลอดภัยขึ้นไปอีก
3. พัฒนาการส่งข้อมูลที่มีขนาดใหญ่ให้มีความเร็วมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code

Project3TSever

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit uProject3TServer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  StdCtrls, JvCtrls, ComCtrls, Psock, NMSTRM, ScktComp, ImgList,
TrayComp,
  abfComponents,JPEG,MMSystem, EDKeyb, JvSpecialProgress,
ExtCtrls,AviCaptura,
  JLCVideo, Spin;

type
  TForm1 = class(TForm)
    StatusBar1: TStatusBar;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox1: TGroupBox;
    Memo1: TMemo;
    GroupBox2: TGroupBox;
    Label1: TLabel;
    Edit1: TEdit;
    JvImgBtn1: TJvImgBtn;
    GroupBox3: TGroupBox;
    Edit2: TEdit;
    JvImgBtn2: TJvImgBtn;
    GroupBox4: TGroupBox;
    ServerSocket1: TServerSocket;
    OpenDialog1: TOpenDialog;
    Label2: TLabel;
    ImageList1: TImageList;
    Label3: TLabel;
    Edit4: TEdit;
    JvImgBtn4: TJvImgBtn;
    JvImgBtn5: TJvImgBtn;
    JvImgBtn6: TJvImgBtn;
    GroupBox5: TGroupBox;
    JvImgBtn7: TJvImgBtn;
    JvImgBtn8: TJvImgBtn;
    JvImgBtn9: TJvImgBtn;
    JvImgBtn10: TJvImgBtn;
    Keyboard1: TDisableEnableKeyboard;
    abfStart1: TabfStartButtonProperties;
    TrayComp1: TTrayComp;
    abfShutdown1: TabfShutdown;
    JvProgress1: TJvSpecialProgress;
    server: TNMStrm;
    JLCVideo: TJLCVideo;
    Image: TImage;
    IPEd: TEdit;
    PorttoEd: TEdit;
    CalidadSpin: TSpinEdit;
    Label8: TLabel;
    FPSSpin: TSpinEdit;
    beginBtn: TButton;
    setBtn: TButton;
    FormatBtn: TButton;
    sentpicBtn: TButton;
    stoppicBtn: TButton;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Label5: TLabel;
    abfAutoRun1: TabfAutoRun;
    procedure FormCreate(Sender: TObject);
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure JvImgBtn1Click(Sender: TObject);
    procedure JvImgBtn2Click(Sender: TObject);
    procedure ServerSocket1ClientConnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1Accept(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1ClientError(Sender: TObject;
      Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
      var ErrorCode: Integer);
    procedure ServerSocket1ClientRead(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1Listen(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ServerSocket1ClientDisconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure JvImgBtn4Click(Sender: TObject);
    procedure JvImgBtn5Click(Sender: TObject);
    procedure JvImgBtn6Click(Sender: TObject);
    procedure JvImgBtn7Click(Sender: TObject);
    procedure JvImgBtn8Click(Sender: TObject);
    procedure JvImgBtn9Click(Sender: TObject);
    procedure JvImgBtn10Click(Sender: TObject);
    procedure JLCVideoStatus(Sender: TObject; HndPreview: Cardinal;
      id: Integer; lpsz: PChar);
    procedure JLCVideoVideoStream(Sender: TObject; HndPreview:
Cardinal;
      lpVHdr: PVIDEOHDR);
    procedure JLCVideoYield(Sender: TObject; HndPreview: Cardinal);
    procedure serverConnect(Sender: TObject);
    procedure serverConnectionFailed(Sender: TObject);
    procedure serverDisconnect(Sender: TObject);
    procedure serverHostResolved(Sender: TComponent);
    procedure serverPacketSent(Sender: TObject);
    procedure serverStatus(Sender: TComponent; Status: String);
    procedure beginBtnClick(Sender: TObject);
    procedure CerrarBtnClick(Sender: TObject);
    procedure setBtnClick(Sender: TObject);
    procedure FormatBtnClick(Sender: TObject);
    procedure sentpicBtnClick(Sender: TObject);
    procedure stoppicBtnClick(Sender: TObject);
    procedure FPSSpinChange(Sender: TObject);
  private
      serviceType : integer;
      connectSocket : TCustomWinSocket;
      procedure InitVar;
      procedure ReleaseVar;
      procedure Echo( content : shortString );
      procedure Controles(valor:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  pwd:string;
  pwdcheck:string;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
implementation
uses upwddlg;

{$R *.DFM}
Const C_ON=TRUE;
      C_OFF=FALSE;

var BmpHead:TBITMAPINFO;
    ClientConect:Boolean;
    CALIDAD:Integer;
    datasent:packed record
                    Primero:array[0..3] of char;
                    Tam:Integer;
                    Segundo:array[8..11] of char;
                    Tercero:array[12..15] of char;
                    Tamcab:integer;
                    FormatTag:Word;
                    Channels:Word;
                    SRate:Integer;
                    BSeg :Integer;
                    BAling:Word;
                    BitSample:Word;
                    Cuarto:array[36..39] of char;
                    NumBytes:Integer;
end;

procedure TForm1.Controles(valor:boolean);
begin
     PorttoEd.Enabled:=valor;
     IPEd.Enabled:=Valor;
     CalidadSpin.Enabled:=VAlor;
     FPSSpin.Enabled:=Valor;
     sentpicBtn.Enabled:=Valor;
     beginBtn.Enabled:=Valor;
     setBtn.Enabled:=Valor;
     FormatBtn.Enabled:=Valor;
end;

procedure TForm1.InitVar;
begin
  serviceType:=0; connectSocket:=nil;
end;

procedure TForm1.ReleaseVar;
begin
end;

procedure TForm1.Echo( content : shortString );
begin
  memo1.lines.add( content );
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
   InitVar;
   JvImgBtn1.Click;
   beginBtn.Click;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose:
Boolean);
begin
   ReleaseVar;
    if(pwd<>'')then
     begin
      pwddlg.showmodal;
      if pwd=pwdcheck then
        canclose:=true
      else
      canclose:=false;
     end;
end;

procedure TForm1.JvImgBtn1Click(Sender: TObject);
var portNo : integer;
begin
  Echo( 'Server àÃÔèÁÃÍÃÑ°¡ÒÃµÔ´µèÍ'+#13#10 );
  portNo:=StrToInt( Edit1.text );
  serviceType:=1;
  ServerSocket1.port:=portNo;
  ServerSocket1.Open;
end;

procedure TForm1.JvImgBtn2Click(Sender: TObject);
var s : shortString; p : pchar;
begin
   s:=Edit2.text; p:=@s[1];
    if ( connectSocket<>nil ) then
        connectSocket.SendBuf( p^, length( s ) );
end;

procedure TForm1.ServerSocket1ClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
     JvImgBtn2.Enabled:=true;
     Echo('ÂÕ¡ÒÃµÔ´µèÍ¨Ò¡ Client'+datetimetostr(now));
end;

procedure TForm1.ServerSocket1Accept(Sender: TObject;
  Socket: TCustomWinSocket);
begin
    Echo( 'µÍ°ÃÑ°¡ÒÃµÔ´µèÍ' );
    connectSocket:=Socket;
end;

procedure TForm1.ServerSocket1ClientError(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
     Echo( '¼Ô´¾ÅÒ´' );
     Echo( ' ¼Ô´¾ÅÒ´ = '+IntToStr( errorCode ) );
end;

procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
var s : shortString; p : pchar; size : integer;
begin
    fillchar( s, sizeof( shortString ), 0 ); p:=@s[1];
    size:=socket.ReceiveBuf( p^, sizeof( shortString ) );
    s[0]:=chr(size);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Echo( 'ä´éÃÑ°¢éÍ¤ÇÒÁ : ' + '='+s+'.' );
   if (s='Restart')then
       JvImgBtn9.Click
     else if (s='ready')then
      sentpicBtn.Click
       else if (s='stop')then
        stoppicBtn.click;
end;

procedure TForm1.ServerSocket1Listen(Sender: TObject;
  Socket: TCustomWinSocket);
begin
    Echo( '¡ÓÅÑ§ÃÍ¡ÒÃµÔ´µèÍ..' );
end;

procedure TForm1.ServerSocket1ClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
    Echo('Client Â¡àÅÔ¡¡ÒÃµÔ´µèÍ'+datetimetostr(now));
end;

procedure TForm1.JvImgBtn4Click(Sender: TObject);
begin
 if(edit4.text<>'')then
   begin
    pwd:=edit4.text;
    JvImgBtn4.visible:=false;
   end
 else
   showmessage('¡ÃØ³ÒãÊèÃËÑÊ');
end;

procedure TForm1.JvImgBtn5Click(Sender: TObject);
begin
  if (groupbox5.visible=true)then
    edit4.enabled:=true;
   if(edit4.text<>'')then
    if(edit4.text=pwd)then
     begin
      groupbox5.visible:=false;
      jvimgbtn5.imageindex:=194;
      jvimgbtn5.enabled:=false;
      jvimgbtn6.enabled:=true;
      edit4.text:='';
     end
    else  showmessage('ÃËÑÊ¼èÒ¹¼Ô´')
   else
   showmessage('¡ÃØ³ÒãÊèÃËÑÊ');
end;

procedure TForm1.JvImgBtn6Click(Sender: TObject);
begin
   if(jvimgbtn4.visible=false)then
    if(edit4.text<>'')then
     if(edit4.text=pwd)then
       begin
        groupbox5.visible:=true;
        jvimgbtn6.imageindex:=172;
        edit4.text:='';
        jvimgbtn6.enabled:=false;
        jvimgbtn5.imageindex:=192;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jvimgbtn5.enabled:=true;
       end
     else
        showmessage('ÃËÑÊ¼èÒ¹¼Ô´')
    else
     showmessage('¡ÃØ³ÒãÊèÃËÑÊ')
   else
    showmessage('ÊÃéÒ§ÃËÑÊ¡èÍ¹¤ÃÑ°')
end;

procedure TForm1.JvImgBtn7Click(Sender: TObject);
begin
 if (jvimgbtn7.imageindex=174) then
     begin
       keyboard1.disable;
       jvimgbtn7.imageindex:=175;
     end
 else
     begin
       keyboard1.enable;
       jvimgbtn7.imageindex:=174;
     end;
end;

procedure TForm1.JvImgBtn8Click(Sender: TObject);
begin
  if (jvimgbtn8.imageindex=118) then
     begin
       abfstart1.enabled:=false;
       jvimgbtn8.imageindex:=119;
     end
  else
     begin
       abfstart1.enabled:=true;
       jvimgbtn8.imageindex:=118;
     end;
end;

procedure TForm1.JvImgBtn9Click(Sender: TObject);
begin
    abfShutdown1.ActionType:=aatReboot;
    abfShutdown1.force:=true;
    abfShutdown1.Execute;
end;

procedure TForm1.JvImgBtn10Click(Sender: TObject);
begin
    abfShutdown1.ActionType:=aatShutdown;
    abfShutdown1.force:=true;
    abfShutdown1.Execute;
end;

procedure TForm1.JLCVideoStatus(Sender: TObject; HndPreview:
Cardinal;
  id: Integer; lpsz: PChar);
begin
   if id=IDS_CAP_END then
      JLCVideo.Overlay:=True;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.JLCVideoVideoStream(Sender: TObject; HndPreview:
Cardinal;
  lpVHdr: PVIDEOHDR);
var
  MiStream: TFileStream;
  JP:TJPEGImage;
begin
    SetDIBits(image.canvas.handle,image.picture.bitmap.handle,0,

image.picture.bitmap.Height,lpVHdr.lpData,BmpHead,DIB_RGB_COLORS);
    Image.Repaint;
   if True then
     begin
        JP:=TJPEGImage.Create;
        JP.Assign(Image.Picture.BitMap);
        JP.CompressionQuality:=CALIDAD;
        Jp.Compress;
         If FileExists('enviar.jpg') then
            DeleteFile('enviar.jpg');
            Jp.SaveToFile('enviar.jpg');
            MiStream := TFileStream.Create('enviar.jpg', fmOpenRead);
          try
            server.PostIt(MiStream);
            finally
              MiStream.Free;
              Jp.Free;
         end;
    end;
end;


procedure TForm1.JLCVideoYield(Sender: TObject; HndPreview:
Cardinal);
begin
 Application.ProcessMessages;
end;

procedure TForm1.serverConnect(Sender: TObject);
begin
   StatusBar1.SimpleText := 'µÔ´µèí';
   ClientConect:=True;
end;

procedure TForm1.serverConnectionFailed(Sender: TObject);
begin
   ShowMessage('¡ÒÃàª×èÍÁµèÍ¼Ô´¾ÅÒ´');
   ClientConect:=False;
end;

procedure TForm1.serverDisconnect(Sender: TObject);
begin
  If StatusBar1 <> nil then
     StatusBar1.SimpleText := 'Â¡àÅÔ¡ÊèSÀÒ¾';
     ClientConect:=False;
end;

procedure TForm1.serverHostResolved(Sender: TComponent);
begin
   StatusBar1.SimpleText := 'Host resolved';
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.serverPacketSent(Sender: TObject);
begin
   StatusBar1.SimpleText := IntToStr(server.BytesSent)+' ¨Ò¡
'+IntToStr(server.BytesTotal)+' ·ÕèÊè§';
end;

procedure TForm1.serverStatus(Sender: TComponent; Status: String);
begin
 If StatusBar1 <> nil then
    StatusBar1.SimpleText := status;
end;

procedure TForm1.beginBtnClick(Sender: TObject);
begin
  if not JLCVideo.Activo then
     begin
        JLCVideo.Activo:=True;

CapGetVideoFormat(JLCvideo.handlepreview,LongInt(@BmpHead),sizeof(Bmp
Head));
        Image.picture.bitmap.width :=BmpHead.bmiHeader.biWidth ;
        Image.picture.bitmap.height:=BmpHead.bmiHeader.biHeight;
     end;
end;

procedure TForm1.CerrarBtnClick(Sender: TObject);
begin
  If JLCVideo.Activo then
     JLCVideo.Activo:=False;
end;

procedure TForm1.setBtnClick(Sender: TObject);
begin
  JLCVideo.SeleccionarFuente;
end;

procedure TForm1.FormatBtnClick(Sender: TObject);
begin
     JLCVideo.SeleccionarFormato;

CapGetVideoFormat(JLCvideo.handlepreview,LongInt(@BmpHead),sizeof(Bmp
Head));
     Image.picture.bitmap.width :=BmpHead.bmiHeader.biWidth ;
     Image.picture.bitmap.height:=BmpHead.bmiHeader.biHeight;
end;

procedure TForm1.sentpicBtnClick(Sender: TObject);
Var WAVEFORMATEX:TWAVEFORMATEX;
begin
     server.Port:=StrToInt(PorttoEd.Text);
     server.Host:=IPEd.Text;
     datasent.Primero:='RIFF';
     datasent.Tercero:='fmt ';
     datasent.Tamcab:=16;
     datasent.FormatTag:=WAVEFORMATEX.wFormatTag;
     datasent.Channels:=WAVEFORMATEX.nChannels;
     datasent.SRate:=WAVEFORMATEX.nSamplesPerSec;
     datasent.BSeg:=WAVEFORMATEX.nAvgBytesPerSec;
     datasent.BAling:=WAVEFORMATEX.nBlockAlign;
     datasent.BitSample:=WAVEFORMATEX.wBitsPerSample;
     datasent.Cuarto:='data';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Controles(C_OFF);
    CALIDAD:=CAlidadSpin.Value;
    JLCVideo.Overlay:=False;
    JLCVideo.GrabarVideoSinDisco;
end;

procedure TForm1.stoppicBtnClick(Sender: TObject);
begin
    JLCVideo.StopVideo;
    Controles(C_ON);
end;

procedure TForm1.FPSSpinChange(Sender: TObject);
begin
    JLCvideo.FramesPreview:=FPSSpin.Value;
    JLCvideo.FramesCaptura:=JLCVideo.FramesPreview;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code

Project3TClient

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit uProject3TClient;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  ExtCtrls, JvAnalogClock, StdCtrls, ComCtrls, JvClock, JvxClock,
ImgList,
  ToolWin,Ras, MPlayer, JPEG, Psock, NMSTRM,MMSystem, ScktComp,
JvCtrls ,
  JvSpecialImage;

type
  TForm1 = class(TForm)
    StatusBar1: TStatusBar;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    TabSheet4: TTabSheet;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    JvxClock1: TJvxClock;
    JvClock1: TJvClock;
    ToolBar1: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    ImageList1: TImageList;
    MediaPlayer1: TMediaPlayer;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    ListBox1: TListBox;
    Panel1: TPanel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit4: TEdit;
    Edit3: TEdit;
    JvImgBtn1: TJvImgBtn;
    JvImgBtn2: TJvImgBtn;
    JvImgBtn3: TJvImgBtn;
    JvImgBtn4: TJvImgBtn;
    JvImgBtn5: TJvImgBtn;
    JvImgBtn6: TJvImgBtn;
    ClientSocket1: TClientSocket;
    GroupBox4: TGroupBox;
    GroupBox5: TGroupBox;
    Label8: TLabel;
    Edit5: TEdit;
    Label9: TLabel;
    Edit6: TEdit;
    Edit7: TEdit;
    Label10: TLabel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    JvImgBtn7: TJvImgBtn;
    JvImgBtn8: TJvImgBtn;
    JvImgBtn9: TJvImgBtn;
    GroupBox6: TGroupBox;
    Edit8: TEdit;
    Label11: TLabel;
    JvImgBtn10: TJvImgBtn;
    JvImgBtn11: TJvImgBtn;
    JvImgBtn12: TJvImgBtn;
    TabSheet3: TTabSheet;
    GroupBox7: TGroupBox;
    JvImgBtn13: TJvImgBtn;
    Memo1: TMemo;
    Memo2: TMemo;
    JvImgBtn15: TJvImgBtn;
    Label12: TLabel;
    Image: TImage;
    PuertoEd: TEdit;
    Client: TNMStrmServ;
    JvImgBtn14: TJvImgBtn;
    JvImgBtn16: TJvImgBtn;
    Image1: TImage;
    JvImgBtn17: TJvImgBtn;
    JvImgBtn19: TJvImgBtn;
    JvImgBtn18: TJvImgBtn;
    TrackBar1: TTrackBar;
    procedure ToolButton1Click(Sender: TObject);
    procedure ToolButton2Click(Sender: TObject);
    procedure ToolButton3Click(Sender: TObject);
    procedure JvxClock1Alarm(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure JvImgBtn2Click(Sender: TObject);
    procedure JvImgBtn3Click(Sender: TObject);
    procedure JvImgBtn4Click(Sender: TObject);
    procedure JvImgBtn5Click(Sender: TObject);
    procedure JvImgBtn6Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure JvImgBtn1Click(Sender: TObject);
    procedure JvImgBtn7Click(Sender: TObject);
    procedure JvImgBtn8Click(Sender: TObject);
    procedure JvImgBtn9Click(Sender: TObject);
    procedure ClientSocket1Connect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ClientSocket1Error(Sender: TObject; Socket:
TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    procedure ClientSocket1Lookup(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ClientSocket1Read(Sender: TObject; Socket:
TCustomWinSocket);
    procedure ClientSocket1Disconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure JvImgBtn10Click(Sender: TObject);
    procedure JvImgBtn11Click(Sender: TObject);
    procedure JvImgBtn12Click(Sender: TObject);
    procedure JvImgBtn15Click(Sender: TObject);
    procedure ClientSocket1Connecting(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure ClientClientContact(Sender: TObject);
    procedure ClientDisconnect(Sender: TObject);
    procedure ClientMSG(Sender: TComponent; const sFrom: String;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      strm: TStream);
    procedure ClientStatus(Sender: TComponent; Status: String);
    procedure JvImgBtn13Click(Sender: TObject);
    procedure JvImgBtn14Click(Sender: TObject);
    procedure JvImgBtn16Click(Sender: TObject);
    procedure JvImgBtn17Click(Sender: TObject);
    procedure JvImgBtn19Click(Sender: TObject);
    procedure JvImgBtn18Click(Sender: TObject);
    procedure TrackBar1Change(Sender: TObject);

  private
      procedure RasGetPhoneBookEntries;
      function  GetParams: boolean;
      procedure Echo( content : shortString );
      procedure InitVar;
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  RasConn: THRasConn;
  RasDialParams: TRasDialParams;
  serviceType : integer;
  connectSocket : TCustomWinSocket;
  pwd:string; sizer,sizep,maxsize,slide:integer;

implementation
Var conect:Boolean;
{$R *.DFM}

procedure TForm1.ToolButton1Click(Sender: TObject);
begin
    jvxclock1.alarmenabled:=true;
    jvxclock1.alarmhour:= StrToInt(combobox1.text);
    jvxclock1.alarmminute:=strtoint(combobox2.text);
    toolbutton1.imageindex:=4;
    toolbutton2.imageindex:=2;
end;

procedure TForm1.ToolButton2Click(Sender: TObject);
begin
      jvxclock1.alarmenabled:=false;
      toolbutton1.imageindex:=5;
      toolbutton2.imageindex:=3;
end;

procedure TForm1.ToolButton3Click(Sender: TObject);
begin
    if (toolbutton3.imageindex=276)then
     begin
      toolbutton3.imageindex:=277;
     end
    else
     toolbutton3.imageindex:=276;
end;

procedure TForm1.JvxClock1Alarm(Sender: TObject);
begin
  if (toolbutton3.imageindex=276)then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   begin
    mediaplayer1.Open;
    mediaplayer1.play;
    showmessage('¶Ö§àÇÅÒ·ÕèµÑé§äÇé');
   end
 else
   showmessage('¶Ö§àÇÅÒ·ÕèµÑé§äÇé');
   toolbutton1.imageindex:=5;
   toolbutton2.imageindex:=3;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
      mediaplayer1.filename:=
ExtractFilePath(Application.ExeName)+'Trumpet3.WAV';
      RasConn := 0;
      JvImgBtn2.Enabled := false;
      RasGetPhoneBookEntries;
      InitVar;
end;

procedure RasCallBack(msg: Integer; state: TRasConnState;
    dwError: Longint); stdcall;
var
  S: string;
  cTxt: Array[0..255] of Char;
begin
  with Form1 do begin
     if (dwError <> 0) then
      begin
        RasGetErrorString(dwError, cTxt, 256);
        S := cTxt;
      end
     else
      case state of
        RASCS_OpenPort:
           S := 'à»Ô´¾ÍÃìµ';
        RASCS_PortOpened:
           S := 'à»Ô´¾ÍÃìµÊÓàÃç¨';
        RASCS_ConnectDevice:
           S := 'A device is about to be connected.';
        RASCS_DeviceConnected:
           S := 'A device has connected successfully.';
        RASCS_AllDevicesConnected:
           S := 'All devices have successfully connected.';
        RASCS_Authenticate:
           S := 'The authentication process is starting.';
        RASCS_AuthNotify:
           S := 'An authentication event has occurred.';
        RASCS_AuthRetry:
           S := 'New validation attempt requested.';
        RASCS_AuthCallback:
         S := 'The remote server has requested a callback.';
        RASCS_AuthChangePassword:
           S := 'The client has requested a password change.';
        RASCS_AuthProject:
           S := 'The projection phase is starting.';
        RASCS_AuthLinkSpeed:
           S := 'The link-speed calculation phase is starting.';
        RASCS_AuthAck:
           S := 'An authentication request is being acknowledged.';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RASCS_ReAuthenticate:
           S := 'Reauthentication (after callback) is starting.';
        RASCS_Authenticated:
           S := 'The client has successfully completed
authentication.';
        RASCS_PrepareForCallback:
           S := 'The line will disconnect in preparation for
callback.';
        RASCS_WaitForModemReset:
           S := 'Delaying; getting ready for a callback.';
        RASCS_WaitForCallback:
           S := 'Waiting for an incoming call from the remote
server.';
        RASCS_Projected:
           S := 'Projection result information is available.';
        RASCS_StartAuthentication:
           S := 'Windows 95 only: User authentication initiated.';
        RASCS_CallbackComplete:
           S := 'Windows 95 only: Client has been called back.';
        RASCS_LogonNetwork:
           S := 'Windows 95 only: Client is logging on to the
network.';
        RASCS_Connected: begin
           S := 'à³×èÍÁµèÍÊÓàÃç¨';
           Panel1.Color := clGreen;
           end;
        RASCS_Disconnected:
           S := '¡ÒÃà³×èÍÁµèí¶Ù¡Â¡ãÄÂÒ¡ÊÃ×Í¹¼Ô´¾ÅÒ¨';
     end;
      StatusBar1.panels[0].text  := S;
  end;
end;

procedure TForm1.JvImgBtn2Click(Sender: TObject);
begin
 if (RasConn <> 0) then
  RasHangUp(RasConn);
  RasConn := 0;
  StatusBar1.panels[0].text := '¡ÒÃà³×èÍÁµèÍÂ¡ãÄÂ¡';
  Panel1.Color := clRed;
  JvImgBtn1.Enabled := true;
  JvImgBtn2.Enabled := false;
end;

procedure TForm1.JvImgBtn3Click(Sender: TObject);
begin
 if (RasCreatePhonebookEntry(Handle, nil) <> 0) then
     ShowMessage('ÊÃéÒ§¡ÒÃà³×èÍÁµèÍ¹¼Ô´¾ÅÒ´!');
     RasGetPhoneBookEntries;
end;

procedure TForm1.JvImgBtn4Click(Sender: TObject);
var
  New, Old: String;
begin
// Test for a selected entry
  if (ListBox1.ItemIndex >= 0) then
   begin
    Old := ListBox1.Items[ListBox1.ItemIndex];
    New := InputBox('à»ÃÕèÂ¹ª×èÍ¡ÒÃà³×èÍÁµèÍ', 'ãÊè³×èÍãËÁè',
       ListBox1.Items[ListBox1.ItemIndex]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (New <> '') then
      if (RasRenameEntry(nil, PChar(Old), PChar(New)) <> 0) then
        ShowMessage('à»ÃÕèÂ¹ªèÍ¼Ô´¾ÅÒ´')
      else
        RasGetPhoneBookEntries;
   end
  else
    ShowMessage('àÅ×Í¡¡ÒÃà ªèÍÁµèÍ¡èÍ¹¤ÃÑ°!');
end;

procedure TForm1.JvImgBtn5Click(Sender: TObject);
begin
 if (ListBox1.ItemIndex >= 0) then
  begin
    if (RasEditPhonebookEntry(Handle, nil,
      PChar(ListBox1.Items[ListBox1.ItemIndex])) <> 0) then
      ShowMessage('á¡éä¢¼Ô´¾ÅÒ´')
    else
      RasGetPhoneBookEntries;
  end
 else
    ShowMessage('àÅ×Í¡¡ÒÃà ªèÍÁµèÍ¡èÍ¹¤ÃÑ°!');
end;

procedure TForm1.JvImgBtn6Click(Sender: TObject);
begin
 if (ListBox1.ItemIndex >= 0) then
  begin
    if
(Application.MessageBox(PChar(ListBox1.Items[ListBox1.ItemIndex] +
      ' ¡ÒÃà ªèÍÁµèÍ¹Õé¨Ð¶Ù¡Åº' + #13#10 + 'Â×¹ÂÑ¹'),
      PChar(Application.Title), MB_YESNO or MB_APPLMODAL or
MB_ICONWARNING)
      = mrYes) then
      if (RasDeleteEntry(nil,
PChar(ListBox1.Items[ListBox1.ItemIndex])) <> 0)
       then ShowMessage('¡ÒÃº¼Ô´¾ÅÒ´')
      else
        RasGetPhoneBookEntries;
  end
 else
    ShowMessage('àÅ×Í¡¡ÒÃà ªèÍÁµèÍ¡èÍ¹¤ÃÑ°!');
end;

procedure TForm1.ListBox1Click(Sender: TObject);
begin
 if not GetParams then
     JvImgBtn1.Enabled := false;
end;

procedure TForm1.RasGetPhoneBookEntries;
var
  RasEntryName: array[1..20] of TRasEntryName;
  i, x: DWord;
  BufSize, Entries: LongInt;
begin
  ListBox1.Clear;
  RasEntryName[1].dwSize := SizeOf(RasEntryName[1]);
  BufSize := SizeOf(RasEntryName);
  x := RasEnumEntries(nil, nil, @RasEntryName, BufSize, Entries);
  if (x = 0) or (x = ERROR_BUFFER_TOO_SMALL) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for i := 1 to Entries do
       if ( i < 21) and (RasEntryName[i].szEntryName[0] <> #0) then
         ListBox1.Items.Add(StrPas(RasEntryName[i].szEntryName));
end;

function TForm1.GetParams: boolean;
var
  fp: LongBool;
  ErrNo, ESize, DSize: Longint;
  Entry: TRasEntry;
  S: string;
  cTxt: Array[0..255] of Char;
begin
  Result := false;
  if (ListBox1.ItemIndex < 0) then
   begin
    ShowMessage('àÅ×Í¡¡ÒÃà¾×èÍÁµèÍ¡èÍ¹¤ÃÑº!');
    Exit;
   end;
    with RasDialParams do
     begin
      dwSize := Sizeof(TRasDialParams);
      StrLCopy(szEntryName,
PChar(ListBox1.Items[ListBox1.ItemIndex]), Ras_MaxEntryName);
     end;
       ErrNo := RasGetEntryDialParams(nil, RasDialParams, fp);
  if (ErrNo = 0) then
    with RasDialParams do
     begin
      Edit1.Text := szUserName;
      if fp then
        Edit2.Text := szPassword;
     end
  else
    begin
     RasGetErrorString(ErrNo, cTxt, 256);
     ShowMessage('¡ÒÃà¾×èÍÁµèÍ¹ÓâÁ´¾ÅÒ´ : ' + cTxt);
     Exit;
    end;
     ESize := 0;
     DSize := 0;
     Entry.dwSize := SizeOf(TRasEntry);
     RasGetEntryProperties(nil,
PChar(ListBox1.Items[ListBox1.ItemIndex]), nil,
     ESize, nil, DSize);
     ErrNo := RasGetEntryProperties(nil,
PChar(ListBox1.Items[ListBox1.ItemIndex]),
     @Entry, ESize, nil, DSize);
  if (ErrNo = 0) then with Entry do
    begin
     if (dwCountryCode <> null) and (szAreaCode <> '') then
       Edit3.Text := IntToStr(dwCountryCode) + ' (' + szAreaCode +
       ') ' + szLocalPhoneNumber
     else if (szAreaCode <> '') then
      Edit3.Text := '(' + szAreaCode + ') ' + szLocalPhoneNumber
     else
      Edit3.Text := szLocalPhoneNumber;
      with IPAddr do
        Edit4.Text := IntToStr(a) + '.' + IntToStr(b) + '.' +
        IntToStr(c) + '.' + IntToStr(d);
      Result := true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JvImgBtn1.Enabled := true;
    end
  else
    begin
     case RasGetErrorString(ErrNo, cTxt, 256) of
        0: S :=  cTxt;
        ERROR_INSUFFICIENT_BUFFER:
           S := 'ERROR_INSUFFICIENT_BUFFER';
        ERROR_INVALID_PARAMETER:
           S := 'ERROR_INVALID_PARAMETER';
     else
        case ErrNo of
           ERROR_INVALID_USER_BUFFER:
              S := 'ERROR_INVALID_USER_BUFFER';
           ERROR_INVALID_PARAMETER:
              S := 'ERROR_INVALID_PARAMETER';
           ERROR_BUFFER_INVALID:
              S := 'ERROR_BUFFER_INVALID';
           ERROR_BUFFER_TOO_SMALL:
              S := 'ERROR_BUFFER_TOO_SMALL';
           ERROR_CANNOT_OPEN_PHONEBOOK:
              S := 'ERROR_CANNOT_OPEN_PHONEBOOK';
           ERROR_CANNOT_FIND_PHONEBOOK_ENTRY:
              S := 'ERROR_CANNOT_FIND_PHONEBOOK_ENTRY';
        else
           S := 'äÁè·ÃÒ° => ' + IntToStr(ErrNo);
     end;
    end;
     ShowMessage('¤é¹ÊÒ¢éÍÁÙÅ¼Ô´¾ÅÒ´ ' + S);
 end;
end;

procedure TForm1.JvImgBtn1Click(Sender: TObject);
var
  ErrNo: longint;
  S: string;
begin
  JvImgBtn1.Enabled := false;
  JvImgBtn2.Enabled := true;
  ErrNo := RasDial(nil, nil, RasDialParams, 0,  @RasCallBack,
RasConn);
   if (ErrNo <> 0) then
    begin
     case ErrNo of
        ERROR_BUFFER_TOO_SMALL: S := 'ERROR_BUFFER_TOO_SMALL';
        ERROR_NOT_ENOUGH_MEMORY: S := 'ERROR_NOT_ENOUGH_MEMORY';
        ERROR_BUFFER_INVALID: S := 'ERROR_BUFFER_INVALID';
     else
        S := 'äÁè·ÃÒ°¤ÇòÁ¼Ô´¾ÅÒ´ (' + IntToStr(ErrNo) + ')';
     end;
     ShowMessage('¡ÒÃà×èÍÁµèÍ¼Ô´¾ÅÒ´ ' + S);
     JvImgBtn2.Click;
  end;
end;

procedure TForm1.JvImgBtn7Click(Sender: TObject);
var hostName : shortString; portNo : integer;
begin
  hostName:=Edit5.text;
   if ( hostName='' ) then
    begin Echo( 'ÃØ³ÒãÊè×èÍHost' );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    exit;
   end;
      Echo( 'àÃÔèÁµÔ´µèÍ'+#13#10 );
      portNo:=StrToInt( Edit6.text );
      serviceType:=2;
      JvImgBtn9.Enabled:=true;
      ClientSocket1.Host:=hostName;
      ClientSocket1.Port:=portNo;
      ClientSocket1.Open;
      JvImgBtn7.Enabled:=False;
      JvImgBtn8.Enabled:=True;
      JvImgBtn13.Enabled:=True;
end;

procedure TForm1.InitVar;
begin
  serviceType:=0; connectSocket:=nil;
end;

procedure TForm1.Echo( content : shortString );
begin
  memo1.lines.add( content );
end;

procedure TForm1.JvImgBtn8Click(Sender: TObject);
begin
    ClientSocket1.Close;
    JvImgBtn7.Enabled:=True;
    JvImgBtn8.Enabled:=False;
    JvImgBtn9.Enabled:=False;
    JvImgBtn14.enabled:=false;
    JvImgBtn13.enabled:=false;
end;

procedure TForm1.JvImgBtn9Click(Sender: TObject);
var  s : shortString; p : pchar;
begin
  s:=Edit7.text; p:=@s[1];
  clientSocket1.Socket.SendBuf( p^, length( s ) );
end;

procedure TForm1.ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
 Echo( '¡ÒÃà¤×èÍÁµèÍÊÓàÃç¨' );
 memo2.Lines.Add('à¤×èÍÁµèÍä»ÂѧS Server àÁ×èÍ'+datetimetostr(now));
end;

procedure TForm1.ClientSocket1Error(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
var ErrorCode: Integer);
begin
  Echo( '¼Ô´¾ÅÒ´' );
  Echo( '  ¼Ô´¾ÅÒ´ = '+IntToStr( errorCode ) );
  memo2.Lines.Add('¡ÒÃà¤×èÍÁµèÍ¼Ô´¾ÅÒ´ àÁ×èÍ'+datetimetostr(now));
end;

procedure TForm1.ClientSocket1Lookup(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  Echo('¤é¹ËÒ server..');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure TForm1.ClientSocket1Read(Sender: TObject;
  Socket: TCustomWinSocket);
var s : shortString; p : pchar; size : integer;
begin
  fillchar( s, sizeof( shortString ), 0 ); p:=@s[1];
  size:=socket.ReceiveBuf( p^, sizeof( shortString ) );
  s[0]:=chr(size);
  Echo( 'ä´éÃÑ°¢éÍ¤ÇÒÁ : ' +s+'.' );
end;

procedure TForm1.ClientSocket1Disconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
   Echo('Â¡àÅÔ¡¡ÒÃà×èÍÁµèÍ');
   memo2.Lines.Add('Â¡àÅÔ¡¡ÒÃà×èÍÁµèÍ¡Ñº Server
àÁ×èÍ'+datetimetostr(now));
   Statusbar1.Panels[0].text:='';
end;

procedure TForm1.JvImgBtn10Click(Sender: TObject);
begin
   if(edit8.text<>'')then
    begin
     pwd:=edit8.text;
     JvImgBtn10.visible:=false;
    end
   else
   showmessage('¡ÃØ³ÒãÊèÃËÑÊ');
end;

procedure TForm1.JvImgBtn11Click(Sender: TObject);
begin
  if (groupbox7.visible=true)then
    edit8.enabled:=true;
   if(edit8.text<>'')then
    if(edit8.text=pwd)then
     begin
      groupbox7.visible:=false;
      jvimgbtn11.imageindex:=194;
      jvimgbtn11.enabled:=false;
      jvimgbtn12.enabled:=true;
      edit8.text:='';
     end
    else  showmessage('ÃËÑÊ¼èÒ¹¼Ô´')
   else
   showmessage('¡ÃØ³ÒãÊèÃËÑÊ');
end;

procedure TForm1.JvImgBtn12Click(Sender: TObject);
begin
if(jvimgbtn10.visible=false)then
    if(edit8.text<>'')then
     if(edit8.text=pwd)then
       begin
        groupbox7.visible:=true;
        jvimgbtn12.imageindex:=172;
        edit8.text:='';
        jvimgbtn12.enabled:=false;
        jvimgbtn11.imageindex:=192;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jvimgbtn11.enabled:=true;
       end
     else
        showmessage('ÃËÑÊ¼èÒ¹¼Ô´')
   else
   showmessage('¡ÃØ³ÒãÊèÃËÑÊ')
  else
   showmessage('ÊÃéÒ§ÃËÑÊ¡èÍ¹¤ÃÑ°')
end;

procedure TForm1.JvImgBtn15Click(Sender: TObject);
begin
  if (JvImgBtn7.Enabled=False)then
      Edit7.text:='Restart';
      JvImgBtn9.Click;
end;

procedure TForm1.ClientSocket1Connecting(Sender: TObject;
  Socket: TCustomWinSocket);
begin
   Statusbar1.Panels[0].text:='¡ÓÃÑ§à¾×èÍÁµèÍ';
end;

procedure TForm1.ClientClientContact(Sender: TObject);
begin
  Client.ReportLevel := Status_Basic;
  client.TimeOut := 90000;
  Statusbar1.Panels[0].text:= 'ÃÑ°ÀÒ¾';
  conect:=True;
  if JvImgBtn19.imageindex=256 then
  JvImgBtn18.Click;
end;

procedure TForm1.ClientDisconnect(Sender: TObject);
begin
  Conect:=False;
end;

procedure TForm1.ClientMSG(Sender: TComponent; const sFrom: String;
  strm: TStream);
var
  MyFStream: TFileStream;
begin
  If FileExists('.\tmp.jpg') then
     DeleteFile('.\tmp.jpg');
     MyFStream := TFileStream.Create('.\tmp.jpg', fmCreate);
   try
      MyFStream.CopyFrom(strm, strm.size);
     finally
        MYFStream.Free;
   end;
    Image.Picture.LoadFromFile('.\tmp.jpg');
end;

procedure TForm1.ClientStatus(Sender: TComponent; Status: String);
begin
 If StatusBar1 <> nil then
    Statusbar1.simpletext:= status;
end;

procedure TForm1.JvImgBtn13Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  edit7.text:='ready';
  JvImgBtn9.Click;
  JvImgBtn14.enabled:=true;
  JvImgBtn13.enabled:=false;
end;

procedure TForm1.JvImgBtn14Click(Sender: TObject);
begin
  edit7.text:='stop';
  JvImgBtn9.Click;
  JvImgBtn14.enabled:=false;
  JvImgBtn13.enabled:=true;
  Statusbar1.Panels[0].text:='';
end;

procedure TForm1.JvImgBtn16Click(Sender: TObject);
begin
  image.Picture.SaveToFile('savepic.jpg');
  sizer:=sizep;
  showmessage('Alarm SavePicture');
end;

procedure TForm1.JvImgBtn17Click(Sender: TObject);
begin
    image1.Picture.LoadFromFile('savepic.jpg');
end;

procedure TForm1.JvImgBtn19Click(Sender: TObject);
begin
 sizer:=sizep;
 slide:=trackbar1.Position;
 maxsize:=sizer+slide;
  if (JvImgBtn19.imageindex=256)then
    begin
     JvImgBtn19.imageindex:=257;
    end
  else
   JvImgBtn19.imageindex:=256;
end;

procedure TForm1.JvImgBtn18Click(Sender: TObject);
var
f: file of Byte;
begin
    AssignFile(f, '.\tmp.jpg');
    Reset(f);
    sizep := FileSize(f);
    Statusbar1.Panels[1].text:= '¢¹Ò´ÀÒ¾'+inttostr(sizep)+'ä°µì';
    CloseFile(f);
    if sizep>maxsize then
    JvImgBtn16.Click;
end;

procedure TForm1.TrackBar1Change(Sender: TObject);
begin
   slide:=trackbar1.Position;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code

RAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{* Copyright (c) 1992-1996, Microsoft Corporation, all rights
reserved
**
** ras.h
** Remote Access external API
** Public header for external API clients
**
*}

{ Delphi conversion by Davide Moretti <dmoretti@iper.net> }
{ Note: All functions and structures defaults to Ansi. If you want to
use
  Unicode structs and funcs, use the names ending with 'W'
  you must define one of these for specific features: }

{.$DEFINE WINVER31}   // for Windows NT 3.5, Windows NT 3.51
{nothing}             // for Windows 95, Windows NT SUR (default)
{.$DEFINE WINVER41}   // for Windows NT SUR enhancements

unit Ras;

interface

uses Windows, Messages, SysUtils;

const

{ These are from lmcons.h }

  DNLEN = 15;  // Maximum domain name length
  UNLEN = 256; // Maximum user name length
  PWLEN = 256; // Maximum password length
  NETBIOS_NAME_LEN = 16; // NetBIOS net name (bytes)

  RAS_MaxDeviceType  = 16;
  RAS_MaxPhoneNumber = 128;
  RAS_MaxIpAddress   = 15;
  RAS_MaxIpxAddress  = 21;

{$IFDEF WINVER31}
{Version 3.x sizes }
  RAS_MaxEntryName     = 20;
  RAS_MaxDeviceName    = 32;
  RAS_MaxCallbackNumber = 48;
{$ELSE}
{Version 4.x sizes }
  RAS_MaxEntryName     = 256;
  RAS_MaxDeviceName    = 128;
  RAS_MaxCallbackNumber = RAS_MaxPhoneNumber;
{$ENDIF}

  RAS_MaxAreaCode      = 10;
  RAS_MaxPadType       = 32;
  RAS_MaxX25Address    = 200;
  RAS_MaxFacilities    = 200;
  RAS_MaxUserData      = 200;

type

  LPHRasConn = ^THRasConn;
  THRasConn = Longint;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{* Identifies an active RAS connection.  (See RasEnumConnections)
*}
  LPRasConnW = ^TRasConnW;
  TRasConnW = record
    dwSize: Longint;
    hrasconn: THRasConn;
    szEntryName: Array[0..RAS_MaxEntryName] of WideChar;
{$IFNDEF WINVER31}
    szDeviceType: Array[0..RAS_MaxDeviceType] of WideChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of WideChar;
{$ENDIF}
{$IFDEF WINVER41}
    szPhonebook: Array[0..MAX_PATH - 1] of WideChar;
    dwSubEntry: Longint;
{$ENDIF}
  end;

  LPRasConnA = ^TRasConnA;
  TRasConnA = record
    dwSize: Longint;
    hrasconn: THRasConn;
    szEntryName: Array[0..RAS_MaxEntryName] of AnsiChar;
{$IFNDEF WINVER31}
   szDeviceType: Array[0..RAS_MaxDeviceType] of AnsiChar;
   szDeviceName: Array[0..RAS_MaxDeviceName] of AnsiChar;
{$ENDIF}
{$IFDEF WINVER41}
    szPhonebook: Array[0..MAX_PATH - 1] of AnsiChar;
    dwSubEntry: Longint;
{$ENDIF}
  end;

  LPRasConn = ^TRasConn;
  TRasConn = TRasConnA;


const

{* Enumerates intermediate states to a connection.  (See RasDial)
*}
  RASCS_PAUSED = $1000;
  RASCS_DONE   = $2000;

type

  LPRasConnState = ^TRasConnState;
  TRasConnState = Integer;

const

  RASCS_OpenPort = 0;
  RASCS_PortOpened = 1;
  RASCS_ConnectDevice = 2;
  RASCS_DeviceConnected = 3;
  RASCS_AllDevicesConnected = 4;
  RASCS_Authenticate = 5;
  RASCS_AuthNotify = 6;
  RASCS_AuthRetry = 7;
  RASCS_AuthCallback = 8;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  RASCS_AuthChangePassword = 9;
  RASCS_AuthProject = 10;
  RASCS_AuthLinkSpeed = 11;
  RASCS_AuthAck = 12;
  RASCS_ReAuthenticate = 13;
  RASCS_Authenticated = 14;
  RASCS_PrepareForCallback = 15;
  RASCS_WaitForModemReset = 16;
  RASCS_WaitForCallback = 17;
  RASCS_Projected = 18;
{$IFNDEF WINVER31}
  RASCS_StartAuthentication = 19;
  RASCS_CallbackComplete = 20;
  RASCS_LogonNetwork = 21;
{$ENDIF}

  RASCS_Interactive = RASCS_PAUSED;
  RASCS_RetryAuthentication = RASCS_PAUSED + 1;
  RASCS_CallbackSetByCaller = RASCS_PAUSED + 2;
  RASCS_PasswordExpired = RASCS_PAUSED + 3;
  RASCS_Connected = RASCS_DONE;
  RASCS_Disconnected = RASCS_DONE + 1;

type

{* Describes the status of a RAS connection.  (See
RasConnectionStatus)
*}
  LPRasConnStatusW = ^TRasConnStatusW;
  TRasConnStatusW = record
    dwSize: Longint;
    rasconnstate: TRasConnState;
    dwError: LongInt;
    szDeviceType: Array[0..RAS_MaxDeviceType] of WideChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of WideChar;
{$IFDEF WINVER41}
    swPhoneNumber: Array[0..RAS_MaxPhoneNumber] of WideChar;
{$ENDIF}
  end;

  LPRasConnStatusA = ^TRasConnStatusA;
  TRasConnStatusA = record
    dwSize: Longint;
    rasconnstate: TRasConnState;
    dwError: LongInt;
    szDeviceType: Array[0..RAS_MaxDeviceType] of AnsiChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of AnsiChar;
{$IFDEF WINVER41}
    swPhoneNumber: Array[0..RAS_MaxPhoneNumber] of AnsiChar;
{$ENDIF}
  end;

  LPRasConnStatus = ^TRasConnStatus;
  TRasConnStatus = TRasConnStatusA;


{* Describes connection establishment parameters.  (See RasDial)
*}
  LPRasDialParamsW = ^TRasDialParamsW;
  TRasDialParamsW = record
    dwSize: LongInt;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    szEntryName: Array[0..RAS_MaxEntryName] of WideChar;
    szPhoneNumber: Array[0..RAS_MaxPhoneNumber] of WideChar;
    szCallbackNumber: Array[0..RAS_MaxCallbackNumber] of WideChar;
    szUserName: Array[0..UNLEN] of WideChar;
    szPassword: Array[0..PWLEN] of WideChar;
    szDomain: Array[0..DNLEN] of WideChar;
{$IFDEF WINVER41}
    dwSubEntry: Longint;
    dwCallbackId: Longint;
{$ENDIF}
  end;

  LPRasDialParamsA = ^TRasDialParamsA;
  TRasDialParamsA = record
    dwSize: LongInt;
    szEntryName: Array[0..RAS_MaxEntryName] of AnsiChar;
    szPhoneNumber: Array[0..RAS_MaxPhoneNumber] of AnsiChar;
    szCallbackNumber: Array[0..RAS_MaxCallbackNumber] of AnsiChar;
    szUserName: Array[0..UNLEN] of AnsiChar;
    szPassword: Array[0..PWLEN] of AnsiChar;
    szDomain: Array[0..DNLEN] of AnsiChar;
{$IFDEF WINVER41}
    dwSubEntry: Longint;
    dwCallbackId: Longint;
{$ENDIF}
  end;

  LPRasDialParams = ^TRasDialParams;
  TRasDialParams = TRasDialParamsA;


{* Describes extended connection establishment options.  (See
RasDial)
*}
  LPRasDialExtensions = ^TRasDialExtensions;
  TRasDialExtensions = record
    dwSize: LongInt;
    dwfOptions: LongInt;
    hwndParent: HWND;
    reserved: LongInt;
  end;

const

{* 'dwfOptions' bit flags.
*}
  RDEOPT_UsePrefixSuffix           = $00000001;
  RDEOPT_PausedStates              = $00000002;
  RDEOPT_IgnoreModemSpeaker        = $00000004;
  RDEOPT_SetModemSpeaker           = $00000008;
  RDEOPT_IgnoreSoftwareCompression = $00000010;
  RDEOPT_SetSoftwareCompression    = $00000020;
  RDEOPT_DisableConnectedUI        = $00000040;
  RDEOPT_DisableReconnectUI        = $00000080;
  RDEOPT_DisableReconnect          = $00000100;
  RDEOPT_NoUser                    = $00000200;
  RDEOPT_PauseOnScript             = $00000400;


  type
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{* Describes an enumerated RAS phone book entry name.  (See
RasEntryEnum)
*}
  LPRasEntryNameW = ^TRasEntryNameW;
  TRasEntryNameW = record
    dwSize: Longint;
    szEntryName: Array[0..RAS_MaxEntryName] of WideChar;
  end;

  LPRasEntryNameA = ^TRasEntryNameA;
  TRasEntryNameA = record
    dwSize: Longint;
    szEntryName: Array[0..RAS_MaxEntryName] of AnsiChar;
  end;

  LPRasEntryName = ^TRasEntryName;
  TRasEntryName = TRasEntryNameA;


{* Protocol code to projection data structure mapping.
*}
  LPRasProjection = ^TRasProjection;
  TRasProjection = Integer;

const

  RASP_Amb = $10000;
  RASP_PppNbf = $803F;
  RASP_PppIpx = $802B;
  RASP_PppIp = $8021;
  RASP_PppLcp = $C021;
  RASP_Slip = $20000;


type

{* Describes the result of a RAS AMB (Authentication Message Block)
** projection.  This protocol is used with NT 3.1 and OS/2 1.3
downlevel
** RAS servers.
*}
  LPRasAmbW = ^TRasAmbW;
  TRasAmbW = record
    dwSize: Longint;
    dwError: Longint;
    szNetBiosError: Array[0..NETBIOS_NAME_LEN] of WideChar;
    bLana: Byte;
  end;

  LPRasAmbA = ^TRasAmbA;
  TRasAmbA = record
    dwSize: Longint;
    dwError: Longint;
    szNetBiosError: Array[0..NETBIOS_NAME_LEN] of AnsiChar;
    bLana: Byte;
  end;

  LPRasAmb = ^TRasAmb;
  TRasAmb = TRasAmbA;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{* Describes the result of a PPP NBF (NetBEUI) projection.
*}
  LPRasPppNbfW = ^TRasPppNbfW;
  TRasPppNbfW = record
    dwSize: Longint;
    dwError: Longint;
    dwNetBiosError: Longint;
    szNetBiosError: Array[0..NETBIOS_NAME_LEN] of WideChar;
    szWorkstationName: Array[0..NETBIOS_NAME_LEN] of WideChar;
    bLana: Byte;
  end;

  LPRasPppNbfA = ^TRasPppNbfA;
  TRasPppNbfA = record
    dwSize: Longint;
    dwError: Longint;
    dwNetBiosError: Longint;
    szNetBiosError: Array[0..NETBIOS_NAME_LEN] of AnsiChar;
    szWorkstationName: Array[0..NETBIOS_NAME_LEN] of AnsiChar;
    bLana: Byte;
  end;

  LpRaspppNbf = ^TRasPppNbf;
  TRasPppNbf = TRasPppNbfA;


{* Describes the results of a PPP IPX (Internetwork Packet Exchange)
** projection.
*}
  LPRasPppIpxW = ^TRasPppIpxW;
  TRasPppIpxW = record
    dwSize: Longint;
    dwError: Longint;
    szIpxAddress: Array[0..RAS_MaxIpxAddress] of WideChar;
  end;

  LPRasPppIpxA = ^TRasPppIpxA;
  TRasPppIpxA = record
    dwSize: Longint;
    dwError: Longint;
    szIpxAddress: Array[0..RAS_MaxIpxAddress] of AnsiChar;
  end;

  LPRasPppIpx = ^TRasPppIpx;
  TRasPppIpx = TRasPppIpxA;


{* Describes the results of a PPP IP (Internet) projection.
*}
  LPRasPppIpW = ^TRasPppIpW;
  TRasPppIpW = record
    dwSize: Longint;
    dwError: Longint;
    szIpAddress: Array[0..RAS_MaxIpAddress] of WideChar;

{$IFNDEF WINNT35COMPATIBLE}

    {* This field was added between Windows NT 3.51 beta and Windows
NT 3.51
    ** final, and between Windows 95 M8 beta and Windows 95 final.
If you do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ** not require the server address and wish to retrieve PPP IP
information
    ** from Windows NT 3.5 or early Windows NT 3.51 betas, or on
early Windows
    ** 95 betas, define WINNT35COMPATIBLE.
    **
    ** The server IP address is not provided by all PPP
implementations,
    ** though Windows NT server's do provide it.
    *}
    szServerIpAddress: Array[0..RAS_MaxIpAddress] of WideChar;

{$ENDIF}
  end;

  LPRasPppIpA = ^TRasPppIpA;
  TRasPppIpA = record
    dwSize: Longint;
    dwError: Longint;
    szIpAddress: Array[0..RAS_MaxIpAddress] of AnsiChar;

{$IFNDEF WINNT35COMPATIBLE}

    {* See RASPPPIPW comment.
    *}
    szServerIpAddress: Array[0..RAS_MaxIpAddress] of AnsiChar;

{$ENDIF}
  end;

  LPRasPppIp = ^TRasPppIp;
  TRasPppIp = TRasPppIpA;


{* Describes the results of a PPP LCP/multi-link negotiation.
*}
  LpRasPppLcp = ^TRasPppLcp;
  TRasPppLcp = record
    dwSize: Longint;
    fBundled: LongBool;
  end;


{* Describes the results of a SLIP (Serial Line IP) projection.
*}
  LpRasSlipW = ^TRasSlipW;
  TRasSlipW = record
    dwSize: Longint;
    dwError: Longint;
    szIpAddress: Array[0..RAS_MaxIpAddress] of WideChar;
  end;

  LpRasSlipA = ^TRasSlipA;
  TRasSlipA = record
    dwSize: Longint;
    dwError: Longint;
    szIpAddress: Array[0..RAS_MaxIpAddress] of AnsiChar;
  end;

  LpRasSlip = ^TRasSlip;
  TRasSlip = TRasSlipA;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
const

{* If using RasDial message notifications, get the notification
message code
** by passing this string to the RegisterWindowMessageA() API.
** WM_RASDIALEVENT is used only if a unique message cannot be
registered.
*}
  RASDIALEVENT    = 'RasDialEvent';
  WM_RASDIALEVENT = $CCCD;


{* Prototypes for caller's RasDial callback handler.  Arguments are
the
** message ID (currently always WM_RASDIALEVENT), the current
RASCONNSTATE and
** the error that has occurred (or 0 if none).  Extended arguments
are the
** handle of the RAS connection and an extended error code.
**
** For RASDIALFUNC2, subsequent callback notifications for all
** subentries can be cancelled by returning FALSE.
*}
{
typedef VOID (WINAPI *RASDIALFUNC)( UINT, RASCONNSTATE, DWORD );
typedef VOID (WINAPI *RASDIALFUNC1)( HRASCONN, UINT, RASCONNSTATE,
DWORD, DWORD );
typedef DWORD (WINAPI *RASDIALFUNC2)( DWORD, DWORD, HRASCONN, UINT,
RASCONNSTATE, DWORD, DWORD );

For Delphi: Just define the callback as:

procedure RASCallback(msg: Integer; state: TRasConnState;
    dwError: Longint); stdcall;

procedure RASCallback1(hConn: THRasConn; msg: Integer;
    state: TRasConnState; dwError: Longint; dwEexterror: Longint);
stdcall;

procedure RASCallback2(dwCallbackId, dwSubEntry: Longint; hConn:
THRasConn;
    msg: Integer; state: TRasConnState; dwError: Longint;
    dwEexterror: Longint); stdcall;
}

type

{* Information describing a RAS-capable device.
*}
  LPRasDevInfoA = ^TRasDevInfoA;
  TRasDevInfoA = record
    dwSize: Longint;
    szDeviceType: Array[0..RAS_MaxDeviceType] of AnsiChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of AnsiChar;
  end;

  LPRasDevInfoW = ^TRasDevInfoW;
  TRasDevInfoW = record
    dwSize: Longint;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    szDeviceType: Array[0..RAS_MaxDeviceType] of WideChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of WideChar;
  end;

  LPRasDevInfo = ^TRasDevInfo;
  TRasDevInfo = TRasDevInfoA;


(* RAS Country Information (currently retreieved from TAPI).
*)
  LPRasCtryInfo = ^TRasCtryInfo;
  TRasCtryInfo = record
    dwSize,
    dwCountryID,
    dwNextCountryID,
    dwCountryCode,
    dwCountryNameOffset: Longint;
  end;

{* There is currently no difference between
** RASCTRYINFOA and RASCTRYINFOW.  This may
** change in the future.
*}

  LPRasCtryInfoW = ^TRasCtryInfoW;
  TRasCtryInfoW = TRasCtryInfo;
  LPRasCtryInfoA = ^TRasCtryInfoA;
  TRasCtryInfoA = TRasCtryInfo;


(* A RAS IP Address.
*)
  LPRasIPAddr = ^TRasIPAddr;
  TRasIPAddr = record
    a, b, c, d: Byte;
  end;


(* A RAS phonebook entry.
*)
  LPRasEntryA = ^TRasEntryA;
  TRasEntryA = record
    dwSize,
    dwfOptions,
    //
    // Location/phone number.
    //
    dwCountryID,
    dwCountryCode: Longint;
    szAreaCode: array[0.. RAS_MaxAreaCode] of AnsiChar;
    szLocalPhoneNumber: array[0..RAS_MaxPhoneNumber] of AnsiChar;
    dwAlternatesOffset: Longint;
    //
    // PPP/Ip
    //
    ipaddr,
    ipaddrDns,
    ipaddrDnsAlt,
    ipaddrWins,
    ipaddrWinsAlt: TRasIPAddr;
    //
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // Framing
    //
    dwFrameSize,
    dwfNetProtocols,
    dwFramingProtocol: Longint;
    //
    // Scripting
    //
    szScript: Array[0..MAX_PATH - 1] of AnsiChar;
    //
    // AutoDial
    //
    szAutodialDll: Array [0..MAX_PATH - 1] of AnsiChar;
    szAutodialFunc: Array [0..MAX_PATH - 1] of AnsiChar;
    //
    // Device
    //
    szDeviceType: Array [0..RAS_MaxDeviceType] of AnsiChar;
    szDeviceName: Array [0..RAS_MaxDeviceName] of AnsiChar;
    //
    // X.25
    //
    szX25PadType: Array [0..RAS_MaxPadType] of AnsiChar;
    szX25Address: Array [0..RAS_MaxX25Address] of AnsiChar;
    szX25Facilities: Array [0..RAS_MaxFacilities] of AnsiChar;
    szX25UserData: Array [0..RAS_MaxUserData] of AnsiChar;
    dwChannels: Longint;
    //
    // Reserved
    //
    dwReserved1,
    dwReserved2: Longint;
{$IFDEF WINVER41}
    //
    // Multilink
    //
    dwSubEntries,
    dwDialMode,
    dwDialExtraPercent,
    dwDialExtraSampleSeconds,
    dwHangUpExtraPercent,
    dwHangUpExtraSampleSeconds: Longint;
    //
    // Idle timeout
    //
    dwIdleDisconnectSeconds: Longint;
{$ENDIF}
  end;

  LPRasEntryW = ^TRasEntryW;
  TRasEntryW = record
    dwSize,
    dwfOptions,
    //
    // Location/phone number.
    //
    dwCountryID,
    dwCountryCode: Longint;
    szAreaCode: array[0.. RAS_MaxAreaCode] of WideChar;
    szLocalPhoneNumber: array[0..RAS_MaxPhoneNumber] of WideChar;
    dwAlternatesOffset: Longint;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    //
    // PPP/Ip
    //
    ipaddr,
    ipaddrDns,
    ipaddrDnsAlt,
    ipaddrWins,
    ipaddrWinsAlt: TRasIPAddr;
    //
    // Framing
    //
    dwFrameSize,
    dwfNetProtocols,
    dwFramingProtocol: Longint;
    //
    // Scripting
    //
    szScript: Array[0..MAX_PATH - 1] of WideChar;
    //
    // AutoDial
    //
    szAutodialDll: Array [0..MAX_PATH - 1] of WideChar;
    szAutodialFunc: Array [0..MAX_PATH - 1] of WideChar;
    //
    // Device
    //
    szDeviceType: Array [0..RAS_MaxDeviceType] of WideChar;
    szDeviceName: Array [0..RAS_MaxDeviceName] of WideChar;
    //
    // X.25
    //
    szX25PadType: Array [0..RAS_MaxPadType] of WideChar;
    szX25Address: Array [0..RAS_MaxX25Address] of WideChar;
    szX25Facilities: Array [0..RAS_MaxFacilities] of WideChar;
    szX25UserData: Array [0..RAS_MaxUserData] of WideChar;
    dwChannels,
    //
    // Reserved
    //
    dwReserved1,
    dwReserved2: Longint;
{$IFDEF WINVER41}
    //
    // Multilink
    //
    dwSubEntries,
    dwDialMode,
    dwDialExtraPercent,
    dwDialExtraSampleSeconds,
    dwHangUpExtraPercent,
    dwHangUpExtraSampleSeconds: Longint;
    //
    // Idle timeout
    //
    dwIdleDisconnectSeconds: Longint;
{$ENDIF}
  end;

  LPRasEntry = ^TRasEntry;
  TRasEntry = TRasEntryA;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
const

(* TRasEntry 'dwfOptions' bit flags.
*)
  RASEO_UseCountryAndAreaCodes = $00000001;
  RASEO_SpecificIpAddr         = $00000002;
  RASEO_SpecificNameServers    = $00000004;
  RASEO_IpHeaderCompression    = $00000008;
  RASEO_RemoteDefaultGateway   = $00000010;
  RASEO_DisableLcpExtensions   = $00000020;
  RASEO_TerminalBeforeDial     = $00000040;
  RASEO_TerminalAfterDial      = $00000080;
  RASEO_ModemLights            = $00000100;
  RASEO_SwCompression          = $00000200;
  RASEO_RequireEncryptedPw     = $00000400;
  RASEO_RequireMsEncryptedPw   = $00000800;
  RASEO_RequireDataEncryption  = $00001000;
  RASEO_NetworkLogon           = $00002000;
  RASEO_UseLogonCredentials    = $00004000;
  RASEO_PromoteAlternates      = $00008000;
{$IFDEF WINVER41}
  RASEO_SecureLocalFiles       = $00010000;
{$ENDIF}

(* TRasEntry 'dwfNetProtocols' bit flags. (session negotiated
protocols)
*)
  RASNP_Netbeui = $00000001; // Negotiate NetBEUI
  RASNP_Ipx     = $00000002; // Negotiate IPX
  RASNP_Ip      = $00000004; // Negotiate TCP/IP

(* TRasEntry 'dwFramingProtocols' (framing protocols used by the
server)
*)
  RASFP_Ppp  = $00000001; // Point-to-Point Protocol (PPP)
  RASFP_Slip = $00000002; // Serial Line Internet Protocol (SLIP)
  RASFP_Ras  = $00000004; // Microsoft proprietary protocol

(* TRasEntry 'szDeviceType' strings
*)
  RASDT_Modem = 'modem';      // Modem
  RASDT_Isdn  = 'isdn';       // ISDN
  RASDT_X25   = 'x25';        // X.25


{* Old AutoDial DLL function prototype.
**
** This prototype is documented for backward-compatibility
** purposes only.  It is superceded by the RASADFUNCA
** and RASADFUNCW definitions below.  DO NOT USE THIS
** PROTOTYPE IN NEW CODE.  SUPPORT FOR IT MAY BE REMOVED
** IN FUTURE VERSIONS OF RAS.
*}
{
typedef BOOL (WINAPI *ORASADFUNC)( HWND, LPSTR, DWORD, LPDWORD );

For Delphi:

function ORasAdFunc(hwndParent: THandle; szEntry: PChar; dwFlags:
Longint;
    var lpdwRetCode: Longint): Boolean; stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

{$IFDEF WINVER41}
{* Flags for RasConnectionNotification().
*}
  RASCN_Connection        = $00000001;
  RASCN_Disconnection     = $00000002;
  RASCN_BandwidthAdded    = $00000004;
  RASCN_BandwidthRemoved = $00000008;

{* TRasEntry 'dwDialMode' values.
*}
  RASEDM_DialAll      = 1;
  RASEDM_DialAsNeeded = 2;

{* TRasEntry 'dwIdleDisconnectSeconds' constants.
*}
  RASIDS_Disabled       = $ffffffff;
  RASIDS_UseGlobalValue = 0;

type

{* AutoDial DLL function parameter block.
*}
  LpRasADParams = ^TRasADParams;
  TRasADParams = record
    dwSize: Longint;
    hwndOwner: THandle;
    dwFlags: Longint;
    xDlg,
    yDlg: Longint;
  end;

const

{* AutoDial DLL function parameter block 'dwFlags.'
*}
  RASADFLG_PositionDlg = $00000001;

{* Prototype AutoDial DLL function.
*}
{
typedef BOOL (WINAPI *RASADFUNCA)( LPSTR, LPSTR, LPRASADPARAMS,
LPDWORD );
typedef BOOL (WINAPI *RASADFUNCW)( LPWSTR, LPWSTR, LPRASADPARAMS,
LPDWORD );

For Delphi:

function RasAdFuncA(lpszPhonebook, lpszEntry: PAnsiChar;
    var lpAutoDialParams: TRasADParams;
    var lpdwRetCode: Longint): Boolean; stdcall;

function RasAdFuncW(lpszPhonebook, lpszEntry: PWideChar;
    var lpAutoDialParams: TRasADParams;
    var lpdwRetCode: Longint): Boolean; stdcall;
}

type

{* A RAS phone book multilinked sub-entry.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*}
  LpRasSubEntryA = ^TRasSubEntryA;
  TRasSubEntryA = record
    dwSize,
    dwfFlags: Longint;
    //
    // Device
    //
    szDeviceType: Array[0..RAS_MaxDeviceType] of AnsiChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of AnsiChar;
    //
    // Phone numbers
    //
    szLocalPhoneNumber: Array[0..RAS_MaxPhoneNumber] of AnsiChar;
    dwAlternateOffset: Longint;
  end;

  LpRasSubEntryW = ^TRasSubEntryW;
  TRasSubEntryW = record
    dwSize,
    dwfFlags: Longint;
    //
    // Device
    //
    szDeviceType: Array[0..RAS_MaxDeviceType] of WideChar;
    szDeviceName: Array[0..RAS_MaxDeviceName] of WideChar;
    //
    // Phone numbers
    //
    szLocalPhoneNumber: Array[0..RAS_MaxPhoneNumber] of WideChar;
    dwAlternateOffset: Longint;
  end;

  LpRasSubEntry = ^TRasSubEntryA;
  TRasSubEntry = TRasSubEntryA;

{* Ras(Get,Set)Credentials structure.  These calls
** supercede Ras(Get,Set)EntryDialParams.
*}
  LpRasCredentialsA = ^TRasCredentialsA;
  TRasCredentialsA = record
    dwSize,
    dwMask: Longint;
    szUserName: Array[0..UNLEN] of AnsiChar;
    zPassword: Array[0..PWLEN ] of AnsiChar;
    szDomain: Array[0..DNLEN] of AnsiChar;
  end;

  LpRasCredentialsW = ^TRasCredentialsW;
  TRasCredentialsW = record
    dwSize,
    dwMask: Longint;
    szUserName: Array[0..UNLEN] of WideChar;
    zPassword: Array[0..PWLEN ] of WideChar;
    szDomain: Array[0..DNLEN] of WideChar;
  end;

  LPRasCredentials = ^TRasCredentials;
  TRasCredentials = TRasCredentialsA;

const
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{* TRasCredentials 'dwMask' values.
*}
  RASCM_UserName = $00000001;
  RASCM_Password = $00000002;
  RASCM_Domain   = $00000004;


type

{* AutoDial address properties.
*}
  LPRasAutoDialEntryA = ^TRasAutoDialEntryA;
  TRasAutoDialEntryA = record
    dwSize,
    dwFlags,
    dwDialingLocation: Longint;
    szEntry: Array[0..RAS_MaxEntryName] of AnsiChar;
  end;

  LPRasAutoDialEntryW = ^TRasAutoDialEntryW;
  TRasAutoDialEntryW = record
    dwSize,
    dwFlags,
    dwDialingLocation: Longint;
    szEntry: Array[0..RAS_MaxEntryName] of WideChar;
  end;

  LPRasAutoDialEntry = ^TRasAutoDialEntry;
  TRasAutoDialEntry = TRasAutoDialEntryA;

const

{* AutoDial control parameter values for
** Ras(Get,Set)AutodialParam.
*}
  RASADP_DisableConnectionQuery  = 0;
  RASADP_LoginSessionDisable     = 1;
  RASADP_SavedAddressesLimit     = 2;
  RASADP_FailedConnectionTimeout = 3;
  RASADP_ConnectionQueryTimeout  = 4;
{$ENDIF}  // WINVER41


{* External RAS API function prototypes.
*}
{Note: for Delphi the function without 'A' or 'W' is the Ansi one
  as on the other Delphi headers}

function RasDialA(
    lpRasDialExt: LPRasDialExtensions;
    lpszPhoneBook: PAnsiChar;
    var params: TRasDialParamsA;
    dwNotifierType: Longint;
    lpNotifier: Pointer;
    var rasconn: THRasConn
    ): Longint; stdcall;
function RasDialW(
    lpRasDialExt: LPRasDialExtensions;
    lpszPhoneBook: PWideChar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    var params: TRasDialParamsW;
    dwNotifierType: Longint;
    lpNotifier: Pointer;
    var rasconn: THRasConn
    ): Longint; stdcall;
function RasDial(
    lpRasDialExt: LPRasDialExtensions;
    lpszPhoneBook: PChar;
    var params: TRasDialParams;
    dwNotifierType: Longint;
    lpNotifier: Pointer;
    var rasconn: THRasConn
    ): Longint; stdcall;

function RasEnumConnectionsA(
    rasconnArray: LPRasConnA;
    var lpcb: Longint;
    var lpcConnections: Longint
    ): Longint; stdcall;
function RasEnumConnectionsW(
    rasconnArray: LPRasConnW;
    var lpcb: Longint;
    var lpcConnections: Longint
    ): Longint; stdcall;
function RasEnumConnections(
    rasconnArray: LPRasConn;
    var lpcb: Longint;
    var lpcConnections: Longint
    ): Longint; stdcall;

function RasEnumEntriesA(
    reserved: PAnsiChar;
    lpszPhoneBook: PAnsiChar;
    entrynamesArray: LPRasEntryNameA;
    var lpcb: Longint;
    var lpcEntries: Longint
    ): Longint; stdcall;
function RasEnumEntriesW(
    reserved: PWideChar;
    lpszPhoneBook: PWideChar;
    entrynamesArray: LPRasEntryNameW;
    var lpcb: Longint;
    var lpcEntries: Longint
    ): Longint; stdcall;
function RasEnumEntries(
    reserved: PChar;
    lpszPhoneBook: PChar;
    entrynamesArray: LPRasEntryName;
    var lpcb: Longint;
    var lpcEntries: Longint
    ): Longint; stdcall;

function RasGetConnectStatusA(
    hConn: THRasConn;
    var lpStatus: TRasConnStatusA
    ): Longint; stdcall;
function RasGetConnectStatusW(
    hConn: THRasConn;
    var lpStatus: TRasConnStatusW
    ): Longint; stdcall;
function RasGetConnectStatus(
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    hConn: THRasConn;
    var lpStatus: TRasConnStatus
    ): Longint; stdcall;

function RasGetErrorStringA(
    errorValue: Integer;
    erroString: PAnsiChar;
    cBufSize: Longint
    ): Longint; stdcall;
function RasGetErrorStringW(
    errorValue: Integer;
    erroString: PWideChar;
    cBufSize: Longint
    ): Longint; stdcall;
function RasGetErrorString(
    errorValue: Integer;
    erroString: PChar;
    cBufSize: Longint
    ): Longint; stdcall;

function RasHangUpA(
    hConn: THRasConn
    ): Longint; stdcall;
function RasHangUpW(
    hConn: THRasConn
    ): Longint; stdcall;
function RasHangUp(
    hConn: THRasConn
    ): Longint; stdcall;

function RasGetProjectionInfoA(
    hConn: THRasConn;
    rasproj: TRasProjection;
    lpProjection: Pointer;
    var lpcb: Longint
    ): Longint; stdcall;
function RasGetProjectionInfoW(
    hConn: THRasConn;
    rasproj: TRasProjection;
    lpProjection: Pointer;
    var lpcb: Longint
    ): Longint; stdcall;
function RasGetProjectionInfo(
    hConn: THRasConn;
    rasproj: TRasProjection;
    lpProjection: Pointer;
    var lpcb: Longint
    ): Longint; stdcall;

function RasCreatePhonebookEntryA(
    hwndParentWindow: HWND;
    lpszPhoneBook: PAnsiChar
    ): Longint; stdcall;
function RasCreatePhonebookEntryW(
    hwndParentWindow: HWND;
    lpszPhoneBook: PWideChar
    ): Longint; stdcall;
function RasCreatePhonebookEntry(
    hwndParentWindow: HWND;
    lpszPhoneBook: PChar
    ): Longint; stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function RasEditPhonebookEntryA(
    hwndParentWindow: HWND;
    lpszPhoneBook: PAnsiChar;
    lpszEntryName: PAnsiChar
    ): Longint; stdcall;
function RasEditPhonebookEntryW(
    hwndParentWindow: HWND;
    lpszPhoneBook: PWideChar;
    lpszEntryName: PWideChar
    ): Longint; stdcall;
function RasEditPhonebookEntry(
    hwndParentWindow: HWND;
    lpszPhoneBook: PChar;
    lpszEntryName: PChar
    ): Longint; stdcall;

function RasSetEntryDialParamsA(
    lpszPhoneBook: PAnsiChar;
    var lpDialParams: TRasDialParamsA;
    fRemovePassword: LongBool
    ): Longint; stdcall;
function RasSetEntryDialParamsW(
    lpszPhoneBook: PWideChar;
    var lpDialParams: TRasDialParamsW;
    fRemovePassword: LongBool
    ): Longint; stdcall;
function RasSetEntryDialParams(
    lpszPhoneBook: PChar;
    var lpDialParams: TRasDialParams;
    fRemovePassword: LongBool
    ): Longint; stdcall;

function RasGetEntryDialParamsA(
    lpszPhoneBook: PAnsiChar;
    var lpDialParams: TRasDialParamsA;
    var lpfPassword: LongBool
    ): Longint; stdcall;
function RasGetEntryDialParamsW(
    lpszPhoneBook: PWideChar;
    var lpDialParams: TRasDialParamsW;
    var lpfPassword: LongBool
    ): Longint; stdcall;
function RasGetEntryDialParams(
    lpszPhoneBook: PChar;
    var lpDialParams: TRasDialParams;
    var lpfPassword: LongBool
    ): Longint; stdcall;

function RasEnumDevicesA(
    lpBuff: LPRasDevInfoA;
    var lpcbSize: Longint;
    var lpcDevices: Longint
    ): Longint; stdcall;
function RasEnumDevicesW(
    lpBuff: LPRasDevInfoW;
    var lpcbSize: Longint;
    var lpcDevices: Longint
    ): Longint; stdcall;
function RasEnumDevices(
    lpBuff: LPRasDevInfo;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    var lpcbSize: Longint;
    var lpcDevices: Longint
    ): Longint; stdcall;

function RasGetCountryInfoA(
    var lpCtryInfo: TRasCtryInfo;
    var lpdwSize: Longint
    ): Longint; stdcall;
function RasGetCountryInfoW(
    var lpCtryInfo: TRasCtryInfo;
    var lpdwSize: Longint
    ): Longint; stdcall;
function RasGetCountryInfo(
    var lpCtryInfo: TRasCtryInfo;
    var lpdwSize: Longint
    ): Longint; stdcall;

function RasGetEntryPropertiesA(
    lpszPhonebook,
    szEntry: PAnsiChar;
        lpbEntry: Pointer;
    var lpdwEntrySize: Longint;
    lpbDeviceInfo: Pointer;
    var lpdwDeviceInfoSize: Longint
    ): Longint; stdcall;
function RasGetEntryPropertiesW(
    lpszPhonebook,
    szEntry: PWideChar;
        lpbEntry: Pointer;
    var lpdwEntrySize: Longint;
    lpbDeviceInfo: Pointer;
    var lpdwDeviceInfoSize: Longint
    ): Longint; stdcall;
function RasGetEntryProperties(
    lpszPhonebook,
    szEntry: PChar;
        lpbEntry: Pointer;
    var lpdwEntrySize: Longint;
    lpbDeviceInfo: Pointer;
    var lpdwDeviceInfoSize: Longint
    ): Longint; stdcall;

function RasSetEntryPropertiesA(
    lpszPhonebook,
    szEntry: PAnsiChar;
        lpbEntry: Pointer;
    dwEntrySize: Longint;
    lpbDeviceInfo: Pointer;
    dwDeviceInfoSize: Longint
    ): Longint; stdcall;
function RasSetEntryPropertiesW(
    lpszPhonebook,
    szEntry: PWideChar;
        lpbEntry: Pointer;
    dwEntrySize: Longint;
    lpbDeviceInfo: Pointer;
    dwDeviceInfoSize: Longint
    ): Longint; stdcall;
function RasSetEntryProperties(
    lpszPhonebook,
    szEntry: PChar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lpbEntry: Pointer;
    dwEntrySize: Longint;
    lpbDeviceInfo: Pointer;
    dwDeviceInfoSize: Longint
    ): Longint; stdcall;

function RasRenameEntryA(
    lpszPhonebook,
    szEntryOld,
    szEntryNew: PAnsiChar
    ): Longint; stdcall;
function RasRenameEntryW(
    lpszPhonebook,
    szEntryOld,
    szEntryNew: PWideChar
    ): Longint; stdcall;
function RasRenameEntry(
    lpszPhonebook,
    szEntryOld,
    szEntryNew: PChar
    ): Longint; stdcall;

function RasDeleteEntryA(
    lpszPhonebook,
    szEntry: PAnsiChar
    ): Longint; stdcall;
function RasDeleteEntryW(
    lpszPhonebook,
    szEntry: PWideChar
    ): Longint; stdcall;
function RasDeleteEntry(
    lpszPhonebook,
    szEntry: PChar
    ): Longint; stdcall;

function RasValidateEntryNameA(
    lpszPhonebook,
    szEntry: PAnsiChar
    ): Longint; stdcall;
function RasValidateEntryNameW(
    lpszPhonebook,
    szEntry: PWideChar
    ): Longint; stdcall;
function RasValidateEntryName(
    lpszPhonebook,
    szEntry: PChar
    ): Longint; stdcall;

{$IFDEF WINVER41}
function RasGetSubEntryHandleA(
    hrasconn: THRasConn;
    dwSubEntry: Longint;
    var lphrasconn: THRasConn
    ): Longint; stdcall;
function RasGetSubEntryHandleW(
    hrasconn: THRasConn;
    dwSubEntry: Longint;
    var lphrasconn: THRasConn
    ): Longint; stdcall;
function RasGetSubEntryHandle(
    hrasconn: THRasConn;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    dwSubEntry: Longint;
    var lphrasconn: THRasConn
    ): Longint; stdcall;

function RasGetCredentialsA(
    lpszPhoneBook,
    lpszEntry: PAnsiChar;
    var lpCredentials: TRasCredentialsA
    ): Longint; stdcall;
function RasGetCredentialsW(
    lpszPhoneBook,
    lpszEntry: PWideChar;
    var lpCredentials: TRasCredentialsW
    ): Longint; stdcall;
function RasGetCredentials(
    lpszPhoneBook,
    lpszEntry: PChar;
    var lpCredentials: TRasCredentials
    ): Longint; stdcall;

function RasSetCredentialsA(
    lpszPhoneBook,
    lpszEntry: PAnsiChar;
    var lpCredentials: TRasCredentialsA;
    fRemovePassword: LongBool
    ): Longint; stdcall;
function RasSetCredentialsW(
    lpszPhoneBook,
    lpszEntry: PWideChar;
    var lpCredentials: TRasCredentialsW;
    fRemovePassword: LongBool
    ): Longint; stdcall;
function RasSetCredentials(
    lpszPhoneBook,
    lpszEntry: PChar;
    var lpCredentials: TRasCredentials;
    fRemovePassword: LongBool
    ): Longint; stdcall;

function RasConnectionNotificationA(
  hrasconn: THRasConn;
  hEvent: THandle;
  dwFlags: Longint
  ): Longint; stdcall;
function RasConnectionNotificationW(
  hrasconn: THRasConn;
  hEvent: THandle;
  dwFlags: Longint
  ): Longint; stdcall;
function RasConnectionNotification(
  hrasconn: THRasConn;
  hEvent: THandle;
  dwFlags: Longint
  ): Longint; stdcall;

function RasGetSubEntryPropertiesA(
    lpszPhoneBook,
    lpszEntry: PAnsiChar;
    dwSubEntry: Longint;
    var lpRasSubEntry: TRasSubEntryA;
    var lpdwcb: Longint;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    p: Pointer;
    var lpdw: Longint
    ): Longint; stdcall;
function RasGetSubEntryPropertiesW(
    lpszPhoneBook,
    lpszEntry: PWideChar;
    dwSubEntry: Longint;
    var lpRasSubEntry: TRasSubEntryW;
    var lpdwcb: Longint;
    p: Pointer;
    var lpdw: Longint
    ): Longint; stdcall;
function RasGetSubEntryProperties(
    lpszPhoneBook,
    lpszEntry: PChar;
    dwSubEntry: Longint;
    var lpRasSubEntry: TRasSubEntry;
    var lpdwcb: Longint;
    p: Pointer;
    var lpdw: Longint
    ): Longint; stdcall;

function RasSetSubEntryPropertiesA(
    lpszPhoneBook,
    lpszEntry: PAnsiChar;
    dwSubEntry: Longint;
    var lpRasSubEntry: TRasSubEntryA;
    dwcb: Longint;
    p: Pointer;
    dw: Longint
    ): Longint; stdcall;
function RasSetSubEntryPropertiesW(
    lpszPhoneBook,
    lpszEntry: PWideChar;
    dwSubEntry: Longint;
    var lpRasSubEntry: TRasSubEntryW;
    dwcb: Longint;
    p: Pointer;
    dw: Longint
    ): Longint; stdcall;
function RasSetSubEntryProperties(
    lpszPhoneBook,
    lpszEntry: PChar;
    dwSubEntry: Longint;
    var lpRasSubEntry: TRasSubEntry;
    dwcb: Longint;
    p: Pointer;
    dw: Longint
    ): Longint; stdcall;

function RasGetAutodialAddressA(
    lpszAddress: PAnsiChar;
    lpdwReserved: Pointer;
    lpAutoDialEntries: LPRasAutoDialEntryA;
    var lpdwcbAutoDialEntries: Longint;
    var lpdwcAutoDialEntries: Longint
    ): Longint; stdcall;
function RasGetAutodialAddressW(
    lpszAddress: PWideChar;
    lpdwReserved: Pointer;
    lpAutoDialEntries: LPRasAutoDialEntryW;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    var lpdwcbAutoDialEntries: Longint;
    var lpdwcAutoDialEntries: Longint
    ): Longint; stdcall;
function RasGetAutodialAddress(
    lpszAddress: PChar;
    lpdwReserved: Pointer;
    lpAutoDialEntries: LPRasAutoDialEntry;
    var lpdwcbAutoDialEntries: Longint;
    var lpdwcAutoDialEntries: Longint
    ): Longint; stdcall;

function RasSetAutodialAddressA(
    lpszAddress: PAnsiChar;
    dwReserved: Longint;
    lpAutoDialEntries: LPRasAutoDialEntryA;
    dwcbAutoDialEntries: Longint;
    dwcAutoDialEntries: Longint
    ): Longint; stdcall;
function RasSetAutodialAddressW(
    lpszAddress: PWideChar;
    dwReserved: Longint;
    lpAutoDialEntries: LPRasAutoDialEntryW;
    dwcbAutoDialEntries: Longint;
    dwcAutoDialEntries: Longint
    ): Longint; stdcall;
function RasSetAutodialAddress(
    lpszAddress: PChar;
    dwReserved: Longint;
    lpAutoDialEntries: LPRasAutoDialEntry;
    dwcbAutoDialEntries: Longint;
    dwcAutoDialEntries: Longint
    ): Longint; stdcall;

function RasEnumAutodialAddressesA(
    lppAddresses: Pointer;
    var lpdwcbAddresses: Longint;
    var lpdwAddresses: Longint
    ): Longint; stdcall;
function RasEnumAutodialAddressesW(
    lppAddresses: Pointer;
    var lpdwcbAddresses: Longint;
    var lpdwAddresses: Longint
    ): Longint; stdcall;
function RasEnumAutodialAddresses(
    lppAddresses: Pointer;
    var lpdwcbAddresses: Longint;
    var lpdwAddresses: Longint
    ): Longint; stdcall;

function RasGetAutodialEnableA(
    dwDialingLocation: Longint;
    var lpfEnabled: LongBool
    ): Longint; stdcall;
function RasGetAutodialEnableW(
    dwDialingLocation: Longint;
    var lpfEnabled: LongBool
    ): Longint; stdcall;
function RasGetAutodialEnable(
    dwDialingLocation: Longint;
    var lpfEnabled: LongBool
    ): Longint; stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function RasSetAutodialEnableA(
    dwDialingLocation: Longint;
    fEnabled: LongBool
    ): Longint; stdcall;
function RasSetAutodialEnableW(
    dwDialingLocation: Longint;
    fEnabled: LongBool
    ): Longint; stdcall;
function RasSetAutodialEnable(
    dwDialingLocation: Longint;
    fEnabled: LongBool
    ): Longint; stdcall;

function RasGetAutodialParamA(
    dwKey: Longint;
    lpvValue: Pointer;
    var lpdwcbValue: Longint
    ): Longint; stdcall;
function RasGetAutodialParamW(
    dwKey: Longint;
    lpvValue: Pointer;
    var lpdwcbValue: Longint
    ): Longint; stdcall;
function RasGetAutodialParam(
    dwKey: Longint;
    lpvValue: Pointer;
    var lpdwcbValue: Longint
    ): Longint; stdcall;

function RasSetAutodialParamA(
    dwKey: Longint;
    lpvValue: Pointer;
    dwcbValue: Longint
    ): Longint; stdcall;
function RasSetAutodialParamW(
    dwKey: Longint;
    lpvValue: Pointer;
    dwcbValue: Longint
    ): Longint; stdcall;
function RasSetAutodialParam(
    dwKey: Longint;
    lpvValue: Pointer;
    dwcbValue: Longint
    ): Longint; stdcall;
{$ENDIF

{**
** raserror.h
** Remote Access external API
** RAS specific error codes
*}

const

  RASBASE = 600;
  SUCCESS = 0;

  PENDING                                   = (RASBASE+0);
{*
 * An operation is pending.%0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 *}
  ERROR_INVALID_PORT_HANDLE             = (RASBASE+1);
{*
 * The port handle is invalid.%0
 *}
  ERROR_PORT_ALREADY_OPEN               = (RASBASE+2);
{*
 * The port is already open.%0
 *}
  ERROR_BUFFER_TOO_SMALL                = (RASBASE+3);
{*
 * Caller's buffer is too small.%0
 *}
  ERROR_WRONG_INFO_SPECIFIED            = (RASBASE+4);
{*
 * Wrong information specified.%0
 *}
  ERROR_CANNOT_SET_PORT_INFO            = (RASBASE+5);
{*
 * Cannot set port information.%0
 *}
  ERROR_PORT_NOT_CONNECTED              = (RASBASE+6);
{*
 * The port is not connected.%0
 *}
  ERROR_EVENT_INVALID                   = (RASBASE+7);
{*
 * The event is invalid.%0
 *}
  ERROR_DEVICE_DOES_NOT_EXIST           = (RASBASE+8);
{*
 * The device does not exist.%0
 *}
  ERROR_DEVICETYPE_DOES_NOT_EXIST       = (RASBASE+9);
{*
 * The device type does not exist.%0
 *}
  ERROR_BUFFER_INVALID                  = (RASBASE+10);
{*
 * The buffer is invalid.%0
 *}
  ERROR_ROUTE_NOT_AVAILABLE             = (RASBASE+11);
{*
 * The route is not available.%0
 *}
  ERROR_ROUTE_NOT_ALLOCATED             = (RASBASE+12);
{*
 * The route is not allocated.%0
 *}
  ERROR_INVALID_COMPRESSION_SPECIFIED   = (RASBASE+13);
{*
 * Invalid compression specified.%0
 *}
  ERROR_OUT_OF_BUFFERS                  = (RASBASE+14);
{*
 * Out of buffers.%0
 *}
  ERROR_PORT_NOT_FOUND                  = (RASBASE+15);
{*
 * The port was not found.%0
 *}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ERROR_ASYNC_REQUEST_PENDING             = (RASBASE+16);
{*
 * An asynchronous request is pending.%0
 *}
  ERROR_ALREADY_DISCONNECTING             = (RASBASE+17);
{*
 * The port or device is already disconnecting.%0
 *}
  ERROR_PORT_NOT_OPEN                     = (RASBASE+18);
{*
 * The port is not open.%0
 *}
  ERROR_PORT_DISCONNECTED                 = (RASBASE+19);
{*
 * The port is disconnected.%0
 *}
  ERROR_NO_ENDPOINTS                      = (RASBASE+20);
{*
 * There are no endpoints.%0
 *}
  ERROR_CANNOT_OPEN_PHONEBOOK             = (RASBASE+21);
{*
 * Cannot open the phone book file.%0
 *}
  ERROR_CANNOT_LOAD_PHONEBOOK             = (RASBASE+22);
{*
 * Cannot load the phone book file.%0
 *}
  ERROR_CANNOT_FIND_PHONEBOOK_ENTRY       = (RASBASE+23);
{*
 * Cannot find the phone book entry.%0
 *}
  ERROR_CANNOT_WRITE_PHONEBOOK            = (RASBASE+24);
{*
 * Cannot write the phone book file.%0
 *}
  ERROR_CORRUPT_PHONEBOOK                 = (RASBASE+25);
{*
 * Invalid information found in the phone book file.%0
 *}
  ERROR_CANNOT_LOAD_STRING                = (RASBASE+26);
{*
 * Cannot load a string.%0
 *}
  ERROR_KEY_NOT_FOUND                     = (RASBASE+27);
{*
 * Cannot find key.%0
 *}
  ERROR_DISCONNECTION                     = (RASBASE+28);
{*
 * The port was disconnected.%0
 *}
  ERROR_REMOTE_DISCONNECTION              = (RASBASE+29);
{*
 * The port was disconnected by the remote machine.%0
 *}
  ERROR_HARDWARE_FAILURE                  = (RASBASE+30);
{*
 * The port was disconnected due to hardware failure.%0
 *}
  ERROR_USER_DISCONNECTION                = (RASBASE+31);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{*
 * The port was disconnected by the user.%0
 *}
  ERROR_INVALID_SIZE                    = (RASBASE+32);
{*
 * The structure size is incorrect.%0
 *}
  ERROR_PORT_NOT_AVAILABLE              = (RASBASE+33);
{*
 * The port is already in use or is not configured for Remote Access
dial out.%0
 *}
  ERROR_CANNOT_PROJECT_CLIENT           = (RASBASE+34);
{*
 * Cannot register your computer on on the remote network.%0
 *}
  ERROR_UNKNOWN                         = (RASBASE+35);
{*
 * Unknown error.%0
 *}
  ERROR_WRONG_DEVICE_ATTACHED           = (RASBASE+36);
{*
 * The wrong device is attached to the port.%0
 *}
  ERROR_BAD_STRING                      = (RASBASE+37);
{*
 * The string could not be converted.%0
 *}
  ERROR_REQUEST_TIMEOUT                 = (RASBASE+38);
{*
 * The request has timed out.%0
 *}
  ERROR_CANNOT_GET_LANA                 = (RASBASE+39);
{*
 * No asynchronous net available.%0
 *}
  ERROR_NETBIOS_ERROR                   = (RASBASE+40);
{*
 * A NetBIOS error has occurred.%0
 *}
  ERROR_SERVER_OUT_OF_RESOURCES         = (RASBASE+41);
{*
 * The server cannot allocate NetBIOS resources needed to support the
client.%0
 *}
  ERROR_NAME_EXISTS_ON_NET              = (RASBASE+42);
{*
 * One of your NetBIOS names is already registered on the remote
network.%0
 *}
  ERROR_SERVER_GENERAL_NET_FAILURE      = (RASBASE+43);
{*
 * A network adapter at the server failed.%0
 *}
  WARNING_MSG_ALIAS_NOT_ADDED           = (RASBASE+44);
{*
 * You will not receive network message popups.%0
 *}
  ERROR_AUTH_INTERNAL                   = (RASBASE+45);
{*
 * Internal authentication error.%0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 *}
  ERROR_RESTRICTED_LOGON_HOURS          = (RASBASE+46);
{*
 * The account is not permitted to logon at this time of day.%0
 *}
  ERROR_ACCT_DISABLED                   = (RASBASE+47);
{*
 * The account is disabled.%0
 *}
  ERROR_PASSWD_EXPIRED                  = (RASBASE+48);
{*
 * The password has expired.%0
 *}
  ERROR_NO_DIALIN_PERMISSION            = (RASBASE+49);
{*
 * The account does not have Remote Access permission.%0
 *}
  ERROR_SERVER_NOT_RESPONDING           = (RASBASE+50);
{*
 * The Remote Access server is not responding.%0
 *}
  ERROR_FROM_DEVICE                     = (RASBASE+51);
{*
 * Your modem (or other connecting device) has reported an error.%0
 *}
  ERROR_UNRECOGNIZED_RESPONSE           = (RASBASE+52);
{*
 * Unrecognized response from the device.%0
 *}
  ERROR_MACRO_NOT_FOUND                 = (RASBASE+53);
{*
 * A macro required by the device was not found in the device .INF
file section.%0
 *}
  ERROR_MACRO_NOT_DEFINED               = (RASBASE+54);
{*
 * A command or response in the device .INF file section refers to an
undefined macro.%0
 *}
  ERROR_MESSAGE_MACRO_NOT_FOUND         = (RASBASE+55);
{*
 * The <message> macro was not found in the device .INF file
secion.%0
 *}
  ERROR_DEFAULTOFF_MACRO_NOT_FOUND      = (RASBASE+56);
{*
 * The <defaultoff> macro in the device .INF file section contains an
undefined macro.%0
 *}
  ERROR_FILE_COULD_NOT_BE_OPENED        = (RASBASE+57);
{*
 * The device .INF file could not be opened.%0
 *}
  ERROR_DEVICENAME_TOO_LONG             = (RASBASE+58);
{*
 * The device name in the device .INF or media .INI file is too
long.%0
 *}
  ERROR_DEVICENAME_NOT_FOUND            = (RASBASE+59);
{*
 * The media .INI file refers to an unknown device name.%0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 *}
  ERROR_NO_RESPONSES                    = (RASBASE+60);
{*
 * The device .INF file contains no responses for the command.%0
 *}
  ERROR_NO_COMMAND_FOUND                = (RASBASE+61);
{*
 * The device .INF file is missing a command.%0
 *}
  ERROR_WRONG_KEY_SPECIFIED             = (RASBASE+62);
{*
 * Attempted to set a macro not listed in device .INF file section.%0
 *}
  ERROR_UNKNOWN_DEVICE_TYPE             = (RASBASE+63);
{*
 * The media .INI file refers to an unknown device type.%0
 *}
  ERROR_ALLOCATING_MEMORY               = (RASBASE+64);
{*
 * Cannot allocate memory.%0
 *}
  ERROR_PORT_NOT_CONFIGURED             = (RASBASE+65);
{*
 * The port is not configured for Remote Access.%0
 *}
  ERROR_DEVICE_NOT_READY                = (RASBASE+66);
{*
 * Your modem (or other connecting device) is not functioning.%0
 *}
  ERROR_READING_INI_FILE                = (RASBASE+67);
{*
 * Cannot read the media .INI file.%0
 *}
  ERROR_NO_CONNECTION                   = (RASBASE+68);
{*
 * The connection dropped.%0
 *}
  ERROR_BAD_USAGE_IN_INI_FILE           = (RASBASE+69);
{*
 * The usage parameter in the media .INI file is invalid.%0
 *}
  ERROR_READING_SECTIONNAME             = (RASBASE+70);
{*
 * Cannot read the section name from the media .INI file.%0
 *}
  ERROR_READING_DEVICETYPE              = (RASBASE+71);
{*
 * Cannot read the device type from the media .INI file.%0
 *}
  ERROR_READING_DEVICENAME              = (RASBASE+72);
{*
 * Cannot read the device name from the media .INI file.%0
 *}
  ERROR_READING_USAGE                   = (RASBASE+73);
{*
 * Cannot read the usage from the media .INI file.%0
 *}
  ERROR_READING_MAXCONNECTBPS           = (RASBASE+74);
{*
 * Cannot read the maximum connection BPS rate from the media .INI
file.%0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 *}
  ERROR_READING_MAXCARRIERBPS           = (RASBASE+75);
{*
 * Cannot read the maximum carrier BPS rate from the media .INI
file.%0
 *}
  ERROR_LINE_BUSY                       = (RASBASE+76);
{*
 * The line is busy.%0
 *}
  ERROR_VOICE_ANSWER                    = (RASBASE+77);
{*
 * A person answered instead of a modem.%0
 *}
  ERROR_NO_ANSWER                       = (RASBASE+78);
{*
 * There is no answer.%0
 *}
  ERROR_NO_CARRIER                      = (RASBASE+79);
{*
 * Cannot detect carrier.%0
 *}
  ERROR_NO_DIALTONE                     = (RASBASE+80);
{*
 * There is no dial tone.%0
 *}
  ERROR_IN_COMMAND                      = (RASBASE+81);
{*
 * General error reported by device.%0
 *}
  ERROR_WRITING_SECTIONNAME             = (RASBASE+82);
{*
 * ERROR_WRITING_SECTIONNAME%0
 *}
  ERROR_WRITING_DEVICETYPE              = (RASBASE+83);
{*
 * ERROR_WRITING_DEVICETYPE%0
 *}
  ERROR_WRITING_DEVICENAME              = (RASBASE+84);
{*
 * ERROR_WRITING_DEVICENAME%0
 *}
  ERROR_WRITING_MAXCONNECTBPS           = (RASBASE+85);
{*
 * ERROR_WRITING_MAXCONNECTBPS%0
 *}
  ERROR_WRITING_MAXCARRIERBPS           = (RASBASE+86);
{*
 * ERROR_WRITING_MAXCARRIERBPS%0
 *}
  ERROR_WRITING_USAGE                   = (RASBASE+87);
{*
 * ERROR_WRITING_USAGE%0
 *}
  ERROR_WRITING_DEFAULTOFF              = (RASBASE+88);
{*
 * ERROR_WRITING_DEFAULTOFF%0
 *}
  ERROR_READING_DEFAULTOFF              = (RASBASE+89);
{*
 * ERROR_READING_DEFAULTOFF%0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 *}
  ERROR_EMPTY_INI_FILE                = (RASBASE+90);
{*
 * ERROR_EMPTY_INI_FILE%0
 *}
  ERROR_AUTHENTICATION_FAILURE        = (RASBASE+91);
{*
 * Access denied because username and/or password is invalid on the
domain.%0
 *}
  ERROR_PORT_OR_DEVICE                = (RASBASE+92);
{*
 * Hardware failure in port or attached device.%0
 *}
  ERROR_NOT_BINARY_MACRO              = (RASBASE+93);
{*
 * ERROR_NOT_BINARY_MACRO%0
 *}
  ERROR_DCB_NOT_FOUND                 = (RASBASE+94);
{*
 * ERROR_DCB_NOT_FOUND%0
 *}
  ERROR_STATE_MACHINES_NOT_STARTED    = (RASBASE+95);
{*
 * ERROR_STATE_MACHINES_NOT_STARTED%0
 *}
  ERROR_STATE_MACHINES_ALREADY_STARTED = (RASBASE+96);
{*
 * ERROR_STATE_MACHINES_ALREADY_STARTED%0
 *}
  ERROR_PARTIAL_RESPONSE_LOOPING      = (RASBASE+97);
{*
 * ERROR_PARTIAL_RESPONSE_LOOPING%0
 *}
  ERROR_UNKNOWN_RESPONSE_KEY          = (RASBASE+98);
{*
 * A response keyname in the device .INF file is not in the expected
format.%0
 *}
  ERROR_RECV_BUF_FULL                 = (RASBASE+99);
{*
 * The device response caused buffer overflow.%0
 *}
  ERROR_CMD_TOO_LONG                  = (RASBASE+100);
{*
 * The expanded command in the device .INF file is too long.%0
 *}
  ERROR_UNSUPPORTED_BPS               = (RASBASE+101);
{*
 * The device moved to a BPS rate not supported by the COM driver.%0
 *}
  ERROR_UNEXPECTED_RESPONSE           = (RASBASE+102);
{*
 * Device response received when none expected.%0
 *}
  ERROR_INTERACTIVE_MODE              = (RASBASE+103);
{*
 * ERROR_INTERACTIVE_MODE%0
 *}
  ERROR_BAD_CALLBACK_NUMBER           = (RASBASE+104);
{*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 * ERROR_BAD_CALLBACK_NUMBER
 *}
  ERROR_INVALID_AUTH_STATE              = (RASBASE+105);
{*
 * ERROR_INVALID_AUTH_STATE%0
 *}
  ERROR_WRITING_INITBPS                 = (RASBASE+106);
{*
 * ERROR_WRITING_INITBPS%0
 *}
  ERROR_X25_DIAGNOSTIC                  = (RASBASE+107);
{*
 * X.25 diagnostic indication.%0
 *}
  ERROR_ACCT_EXPIRED                    = (RASBASE+108);
{*
 * The account has expired.%0
 *}
  ERROR_CHANGING_PASSWORD               = (RASBASE+109);
{*
 * Error changing password on domain.  The password may be too short
or may match a previously used password.%0
 *}
  ERROR_OVERRUN                         = (RASBASE+110);
{*
 * Serial overrun errors were detected while communicating with your
modem.%0
 *}
  ERROR_RASMAN_CANNOT_INITIALIZE        = (RASBASE+111);
{*
 * RasMan initialization failure.  Check the event log.%0
 *}
  ERROR_BIPLEX_PORT_NOT_AVAILABLE       = (RASBASE+112);
{*
 * Biplex port initializing.  Wait a few seconds and redial.%0
 *}
  ERROR_NO_ACTIVE_ISDN_LINES            = (RASBASE+113);
{*
 * No active ISDN lines are available.%0
 *}
  ERROR_NO_ISDN_CHANNELS_AVAILABLE      = (RASBASE+114);
{*
 * No ISDN channels are available to make the call.%0
 *}
  ERROR_TOO_MANY_LINE_ERRORS            = (RASBASE+115);
{*
 * Too many errors occured because of poor phone line quality.%0
 *}
  ERROR_IP_CONFIGURATION                = (RASBASE+116);
{*
 * The Remote Access IP configuration is unusable.%0
 *}
  ERROR_NO_IP_ADDRESSES                 = (RASBASE+117);
{*
 * No IP addresses are available in the static pool of Remote Access
IP addresses.%0
 *}
  ERROR_PPP_TIMEOUT                     = (RASBASE+118);
{*
 * Timed out waiting for a valid response from the remote PPP peer.%0
 *}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ERROR_PPP_REMOTE_TERMINATED           = (RASBASE+119);
{*
 * PPP terminated by remote machine.%0
 *}
  ERROR_PPP_NO_PROTOCOLS_CONFIGURED     = (RASBASE+120);
{*
 * No PPP control protocols configured.%0
 *}
  ERROR_PPP_NO_RESPONSE                 = (RASBASE+121);
{*
 * Remote PPP peer is not responding.%0
 *}
  ERROR_PPP_INVALID_PACKET              = (RASBASE+122);
{*
 * The PPP packet is invalid.%0
 *}
  ERROR_PHONE_NUMBER_TOO_LONG           = (RASBASE+123);
{*
 * The phone number including prefix and suffix is too long.%0
 *}
  ERROR_IPXCP_NO_DIALOUT_CONFIGURED     = (RASBASE+124);
{*
 * The IPX protocol cannot dial-out on the port because the machine
is an IPX router.%0
 *}
  ERROR_IPXCP_NO_DIALIN_CONFIGURED      = (RASBASE+125);
{*
 * The IPX protocol cannot dial-in on the port because the IPX router
is not installed.%0
 *}
  ERROR_IPXCP_DIALOUT_ALREADY_ACTIVE    = (RASBASE+126);
{*
 * The IPX protocol cannot be used for dial-out on more than one port
at a time.%0
 *}
  ERROR_ACCESSING_TCPCFGDLL             = (RASBASE+127);
{*
 * Cannot access TCPCFG.DLL.%0
 *}
  ERROR_NO_IP_RAS_ADAPTER               = (RASBASE+128);
{*
 * Cannot find an IP adapter bound to Remote Access.%0
 *}
  ERROR_SLIP_REQUIRES_IP                = (RASBASE+129);
{*
 * SLIP cannot be used unless the IP protocol is installed.%0
 *}
  ERROR_PROJECTION_NOT_COMPLETE         = (RASBASE+130);
{*
 * Computer registration is not complete.%0
 *}
  ERROR_PROTOCOL_NOT_CONFIGURED         = (RASBASE+131);
{*
 * The protocol is not configured.%0
 *}
  ERROR_PPP_NOT_CONVERGING              = (RASBASE+132);
{*
 * The PPP negotiation is not converging.%0
 *}
  ERROR_PPP_CP_REJECTED                 = (RASBASE+133);
{*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 * The PPP control protocol for this network protocol is not
available on the server.%0
 *}
  ERROR_PPP_LCP_TERMINATED              = (RASBASE+134);
{*
 * The PPP link control protocol terminated.%0
 *}
  ERROR_PPP_REQUIRED_ADDRESS_REJECTED  = (RASBASE+135);
{*
 * The requested address was rejected by the server.%0
 *}
  ERROR_PPP_NCP_TERMINATED              = (RASBASE+136);
{*
 * The remote computer terminated the control protocol.%0
 *}
  ERROR_PPP_LOOPBACK_DETECTED           = (RASBASE+137);
{*
 * Loopback detected.%0
 *}
  ERROR_PPP_NO_ADDRESS_ASSIGNED         = (RASBASE+138);
{*
 * The server did not assign an address.%0
 *}
  ERROR_CANNOT_USE_LOGON_CREDENTIALS    = (RASBASE+139);
{*
 * The authentication protocol required by the remote server cannot
use the Windows NT encrypted password.  Redial, entering the password
explicitly.%0
 *}
  ERROR_TAPI_CONFIGURATION              = (RASBASE+140);
{*
 * Invalid TAPI configuration.%0
 *}
  ERROR_NO_LOCAL_ENCRYPTION             = (RASBASE+141);
{*
 * The local computer does not support encryption.%0
 *}
  ERROR_NO_REMOTE_ENCRYPTION            = (RASBASE+142);
{*
 * The remote server does not support encryption.%0
 *}
  ERROR_REMOTE_REQUIRES_ENCRYPTION      = (RASBASE+143);
{*
 * The remote server requires encryption.%0
 *}
  ERROR_IPXCP_NET_NUMBER_CONFLICT       = (RASBASE+144);
{*
 * Cannot use the IPX network number assigned by remote server.
Check the event log.%0
 *}
  ERROR_INVALID_SMM                     = (RASBASE+145);
{*
 * ERROR_INVALID_SMM%0
 *}
  ERROR_SMM_UNINITIALIZED               = (RASBASE+146);
{*
 * ERROR_SMM_UNINITIALIZED%0
 *}
  ERROR_NO_MAC_FOR_PORT                 = (RASBASE+147);
{*
 * ERROR_NO_MAC_FOR_PORT%0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 *}
  ERROR_SMM_TIMEOUT                    = (RASBASE+148);
{*
 * ERROR_SMM_TIMEOUT%0
 *}
  ERROR_BAD_PHONE_NUMBER               = (RASBASE+149);
{*
 * ERROR_BAD_PHONE_NUMBER%0
 *}
  ERROR_WRONG_MODULE                   = (RASBASE+150);
{*
 * ERROR_WRONG_MODULE%0
 *}
  ERROR_INVALID_CALLBACK_NUMBER        = (RASBASE+151);
{*
 * Invalid callback number.  Only the characters 0 to 9, T, P, W, (,
), -, @, and space are allowed in the number.%0
 *}
  ERROR_SCRIPT_SYNTAX                  = (RASBASE+152);
{*
 * A syntax error was encountered while processing a script.%0
 *}
  RASBASEEND                           = (RASBASE+152);

implementation

function RasCreatePhonebookEntryA;    external 'rasapi32.dll' name
'RasCreatePhonebookEntryA';
function RasCreatePhonebookEntryW;    external 'rasapi32.dll' name
'RasCreatePhonebookEntryW';
function RasCreatePhonebookEntry;     external 'rasapi32.dll' name
'RasCreatePhonebookEntryA';
function RasDialA;                    external 'rasapi32.dll' name
'RasDialA';
function RasDialW;                    external 'rasapi32.dll' name
'RasDialW';
function RasDial;                     external 'rasapi32.dll' name
'RasDialA';
function RasEditPhonebookEntryA;      external 'rasapi32.dll' name
'RasEditPhonebookEntryA';
function RasEditPhonebookEntryW;      external 'rasapi32.dll' name
'RasEditPhonebookEntryW';
function RasEditPhonebookEntry;       external 'rasapi32.dll' name
'RasEditPhonebookEntryA';
function RasEnumConnectionsA;         external 'rasapi32.dll' name
'RasEnumConnectionsA';
function RasEnumConnectionsW;         external 'rasapi32.dll' name
'RasEnumConnectionsW';
function RasEnumConnections;          external 'rasapi32.dll' name
'RasEnumConnectionsA';
function RasEnumEntriesA;             external 'rasapi32.dll' name
'RasEnumEntriesA';
function RasEnumEntriesW;             external 'rasapi32.dll' name
'RasEnumEntriesW';
function RasEnumEntries;              external 'rasapi32.dll' name
'RasEnumEntriesA';
function RasGetConnectStatusA;        external 'rasapi32.dll' name
'RasGetConnectStatusA';
function RasGetConnectStatusW;        external 'rasapi32.dll' name
'RasGetConnectStatusW';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function RasGetConnectStatus;          external 'rasapi32.dll' name
'RasGetConnectStatusA';
function RasGetEntryDialParamsA;       external 'rasapi32.dll' name
'RasGetEntryDialParamsA';
function RasGetEntryDialParamsW;       external 'rasapi32.dll' name
'RasGetEntryDialParamsW';
function RasGetEntryDialParams;        external 'rasapi32.dll' name
'RasGetEntryDialParamsA';
function RasGetErrorStringA;           external 'rasapi32.dll' name
'RasGetErrorStringA';
function RasGetErrorStringW;           external 'rasapi32.dll' name
'RasGetErrorStringW';
function RasGetErrorString;            external 'rasapi32.dll' name
'RasGetErrorStringA';
function RasGetProjectionInfoA;        external 'rasapi32.dll' name
'RasGetProjectionInfoA';
function RasGetProjectionInfoW;        external 'rasapi32.dll' name
'RasGetProjectionInfoW';
function RasGetProjectionInfo;         external 'rasapi32.dll' name
'RasGetProjectionInfoA';
function RasHangUpA;                   external 'rasapi32.dll' name
'RasHangUpA';
function RasHangUpW;                   external 'rasapi32.dll' name
'RasHangUpW';
function RasHangUp;                    external 'rasapi32.dll' name
'RasHangUpA';
function RasSetEntryDialParamsA;       external 'rasapi32.dll' name
'RasSetEntryDialParamsA';
function RasSetEntryDialParamsW;       external 'rasapi32.dll' name
'RasSetEntryDialParamsW';
function RasSetEntryDialParams;        external 'rasapi32.dll' name
'RasSetEntryDialParamsA';

{* These are subject to change, since RNAPH.DLL will be removed.
** The 'A' and 'W' functions are present in Windows NT 4.0
*}
function RasEnumDevicesA;              external 'rasapi32.dll' name
'RasEnumDevicesA';
function RasEnumDevicesW;              external 'rasapi32.dll' name
'RasEnumDevicesW';
{$IFDEF WINVER41}
function RasEnumDevices;               external 'rasapi32.dll' name
'RasEnumDevicesA';
{$ENDIF}
function RasGetCountryInfoA;           external 'rasapi32.dll' name
'RasGetCountryInfoA';
function RasGetCountryInfoW;           external 'rasapi32.dll' name
'RasGetCountryInfoW';
{$IFDEF WINVER41}
function RasGetCountryInfo;            external 'rasapi32.dll' name
'RasGetCountryInfoA';
{$ENDIF}
function RasGetEntryPropertiesA;       external 'rasapi32.dll' name
'RasGetEntryPropertiesA';
function RasGetEntryPropertiesW;       external 'rasapi32.dll' name
'RasGetEntryPropertiesW';
{$IFDEF WINVER41}
function RasGetEntryProperties;        external 'rasapi32.dll' name
'RasGetEntryPropertiesA';
{$ENDIF}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function RasSetEntryPropertiesA;      external 'rasapi32.dll' name
'RasSetEntryPropertiesA';
function RasSetEntryPropertiesW;      external 'rasapi32.dll' name
'RasSetEntryPropertiesW';
{$IFDEF WINVER41}
function RasSetEntryProperties;       external 'rasapi32.dll' name
'RasSetEntryPropertiesA';
{$ENDIF}
function RasRenameEntryA;             external 'rasapi32.dll' name
'RasRenameEntryA';
function RasRenameEntryW;             external 'rasapi32.dll' name
'RasRenameEntryW';
{$IFDEF WINVER41}
function RasRenameEntry;              external 'rasapi32.dll' name
'RasRenameEntryA';
{$ENDIF}
function RasDeleteEntryA;             external 'rasapi32.dll' name
'RasDeleteEntryA';
function RasDeleteEntryW;             external 'rasapi32.dll' name
'RasDeleteEntryW';
{$IFDEF WINVER41}
function RasDeleteEntry;              external 'rasapi32.dll' name
'RasDeleteEntryA';
{$ENDIF}
function RasValidateEntryNameA;       external 'rasapi32.dll' name
'RasValidateEntryNameA';
function RasValidateEntryNameW;       external 'rasapi32.dll' name
'RasValidateEntryNameW';
{$IFDEF WINVER41}
function RasValidateEntryName;        external 'rasapi32.dll' name
'RasValidateEntryNameA';
{$ENDIF}
{$IFDEF WINVER41}
function RasGetSubEntryHandleA;       external 'rasapi32.dll' name
'RasGetSubEntryHandleA';
function RasGetSubEntryHandleW;       external 'rasapi32.dll' name
'RasGetSubEntryHandleW';
function RasGetSubEntryHandle;        external 'rasapi32.dll' name
'RasGetSubEntryHandleA';
function RasGetCredentialsA;          external 'rasapi32.dll' name
'RasGetCredentialsA';
function RasGetCredentialsW;          external 'rasapi32.dll' name
'RasGetCredentialsW';
function RasGetCredentials;           external 'rasapi32.dll' name
'RasGetCredentialsA';
function RasSetCredentialsA;          external 'rasapi32.dll' name
'RasSetCredentialsA';
function RasSetCredentialsW;          external 'rasapi32.dll' name
'RasSetCredentialsW';
function RasSetCredentials;           external 'rasapi32.dll' name
'RasSetCredentialsA';
function RasConnectionNotificationA; external 'rasapi32.dll' name
'RasConnectionNotificationA';
function RasConnectionNotificationW; external 'rasapi32.dll' name
'RasConnectionNotificationW';
function RasConnectionNotification;  external 'rasapi32.dll' name
'RasConnectionNotificationA';
function RasGetSubEntryPropertiesA;  external 'rasapi32.dll' name
'RasGetSubEntryPropertiesA';
function RasGetSubEntryPropertiesW;  external 'rasapi32.dll' name
'RasGetSubEntryPropertiesW';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function RasGetSubEntryProperties;   external 'rasapi32.dll' name
'RasGetSubEntryPropertiesA';
function RasSetSubEntryPropertiesA;  external 'rasapi32.dll' name
'RasSetSubEntryPropertiesA';
function RasSetSubEntryPropertiesW;  external 'rasapi32.dll' name
'RasSetSubEntryPropertiesW';
function RasSetSubEntryProperties;   external 'rasapi32.dll' name
'RasSetSubEntryPropertiesA';
function RasGetAutodialAddressA;     external 'rasapi32.dll' name
'RasGetAutodialAddressA';
function RasGetAutodialAddressW;     external 'rasapi32.dll' name
'RasGetAutodialAddressW';
function RasGetAutodialAddress;      external 'rasapi32.dll' name
'RasGetAutodialAddressA';
function RasSetAutodialAddressA;     external 'rasapi32.dll' name
'RasSetAutodialAddressA';
function RasSetAutodialAddressW;     external 'rasapi32.dll' name
'RasSetAutodialAddressW';
function RasSetAutodialAddress;      external 'rasapi32.dll' name
'RasSetAutodialAddressA';
function RasEnumAutodialAddressesA;  external 'rasapi32.dll' name
'RasEnumAutodialAddressesA';
function RasEnumAutodialAddressesW;  external 'rasapi32.dll' name
'RasEnumAutodialAddressesW';
function RasEnumAutodialAddresses;   external 'rasapi32.dll' name
'RasEnumAutodialAddressesA';
function RasGetAutodialEnableA;      external 'rasapi32.dll' name
'RasGetAutodialEnableA';
function RasGetAutodialEnableW;      external 'rasapi32.dll' name
'RasGetAutodialEnableW';
function RasGetAutodialEnable;       external 'rasapi32.dll' name
'RasGetAutodialEnableA';
function RasSetAutodialEnableA;      external 'rasapi32.dll' name
'RasSetAutodialEnableA';
function RasSetAutodialEnableW;      external 'rasapi32.dll' name
'RasSetAutodialEnableW';
function RasSetAutodialEnable;       external 'rasapi32.dll' name
'RasSetAutodialEnableA';
function RasGetAutodialParamA;       external 'rasapi32.dll' name
'RasGetAutodialParamA';
function RasGetAutodialParamW;       external 'rasapi32.dll' name
'RasGetAutodialParamW';
function RasGetAutodialParam;        external 'rasapi32.dll' name
'RasGetAutodialParamA';
function RasSetAutodialParamA;       external 'rasapi32.dll' name
'RasSetAutodialParamA';
function RasSetAutodialParamW;       external 'rasapi32.dll' name
'RasSetAutodialParamW';
function RasSetAutodialParam;        external 'rasapi32.dll' name
'RasSetAutodialParamA';
{$ENDIF}
{*
** but for now the functions without the type specifier must be
implemented
** as follows (at least until rnaph.dll disappears...
*}

{$IFNDEF WINVER41}
var
  rnaph_initialized: Boolean = False;
  is_rnaph: Boolean = False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  lib: HModule;

function rnaph_(const func: String): Pointer;
  begin
  if not rnaph_initialized then
    begin
    // Try first with RASAPI32.DLL
    lib := LoadLibrary('rasapi32.dll');
    if lib <> 0 then
      begin
      Result := GetProcAddress(lib, PChar(func + 'A'));
      if Result <> nil then
        begin
        rnaph_initialized := True;
        Exit;
        end;
      end
    else
      raise Exception.Create('Error opening rasapi.dll');
    // function not found - try rnaph.dll
    lib := LoadLibrary('rnaph.dll');
    if lib <> 0 then
      begin
      Result := GetProcAddress(lib, PChar(func));
      if Result <> nil then
        begin
        rnaph_initialized := True;
        is_rnaph := True;
        Exit;
        end
      else
        raise Exception.Create('Function ' + func + ' not found!');
      end
    else
      raise Exception.Create('Error opening rnaph.dll');
    end
  else
    begin
    if is_rnaph then
      Result := GetProcAddress(lib, PChar(func))
    else
      Result := GetProcAddress(lib, PChar(func + 'A'));
    if Result = nil then
      raise Exception.Create('Function ' + func + ' not found!');
    end;
  end;

function RasValidateEntryName(lpszPhonebook, szEntry: PAnsiChar):
Longint;
  var
    f: Function(lpszPhonebook, szEntry: PAnsiChar): Longint; stdcall;
  begin
  @f := rnaph_('RasValidateEntryName');
  Result := f(lpszPhonebook, szEntry);
  end;

function RasRenameEntry(lpszPhonebook, szEntryOld, szEntryNew:
PAnsiChar): Longint;
  var
    f: function(lpszPhonebook, szEntryOld, szEntryNew: PAnsiChar):
Longint; stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
  @f := rnaph_('RasRenameEntry');
  Result := f(lpszPhonebook, szEntryOld, szEntryNew);
  end;

function RasDeleteEntry(lpszPhonebook, szEntry: PAnsiChar): Longint;
  var
    f: function(lpszPhonebook, szEntry: PAnsiChar): Longint; stdcall;
  begin
  @f := rnaph_('RasDeleteEntry');
  Result := f(lpszPhonebook, szEntry);
  end;

function RasGetEntryProperties(lpszPhonebook, szEntry: PAnsiChar;
lpbEntry: Pointer;
    var lpdwEntrySize: Longint; lpbDeviceInfo: Pointer;
    var lpdwDeviceInfoSize: Longint): Longint;
  var
    f: function(lpszPhonebook, szEntry: PAnsiChar; lpbEntry: Pointer;
        var lpdwEntrySize: Longint; lpbDeviceInfo: Pointer;
        var lpdwDeviceInfoSize: Longint): Longint; stdcall;
  begin
  @f := rnaph_('RasGetEntryProperties');
  Result := f(lpszPhonebook, szEntry, lpbEntry, lpdwEntrySize,
lpbDeviceInfo, lpdwDeviceInfoSize);
  end;

function RasSetEntryProperties(lpszPhonebook, szEntry: PAnsiChar;
      lpbEntry: Pointer; dwEntrySize: Longint; lpbDeviceInfo:
Pointer;
    dwDeviceInfoSize: Longint): Longint;
  var
    f: function(lpszPhonebook, szEntry: PAnsiChar;
      lpbEntry: Pointer; dwEntrySize: Longint; lpbDeviceInfo:
Pointer;
      dwDeviceInfoSize: Longint): Longint; stdcall;
  begin
  @f := rnaph_('RasSetEntryProperties');
  Result := f(lpszPhonebook, szEntry, lpbEntry, dwEntrySize,
lpbDeviceInfo, dwDeviceInfoSize);
  end;

function RasGetCountryInfo(var lpCtryInfo: TRasCtryInfo;
    var lpdwSize: Longint): Longint;
  var
    f: function(var lpCtryInfo: TRasCtryInfo;
        var lpdwSize: Longint): Longint; stdcall;
  begin
  @f := rnaph_('RasGetCountryInfo');
  Result := f(lpCtryInfo, lpdwSize);
  end;

function RasEnumDevices(lpBuff: LpRasDevInfo; var lpcbSize: Longint;
    var lpcDevices: Longint): Longint;
  var
    f: function(lpBuff: LpRasDevInfo; var lpcbSize: Longint;
        var lpcDevices: Longint): Longint; stdcall;
  begin
  @f := rnaph_('RasEnumDevices');
  Result := f(lpBuff, lpcbSize, lpcDevices);
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
initialization
finalization
  if rnaph_initialized and is_rnaph then
    FreeLibrary(lib);
{$ENDIF}
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. Garry Comnell , Troy Strain , " Delphi Nut & Bolts for experienced programmers " , Berkely , CA : McGraw Hill , c1995.
2. กนก กุสุมาลย์นุกูล , ไกรวุฒิ มั่นเสถียรสิน , " คู่มือการใช้งานโปรแกรม Borland Delphi 5 " , บริษัท ซัคเซส มีเดีย จำกัด , กรุงเทพ ฯ , 2545
3. จีไอเอส – กรุ๊ป , " เครือข่ายคอมพิวเตอร์ " , หจก.สำนักพิมพ์ ฟิสิกส์ เซ็นเตอร์ , กรุงเทพ ฯ , 2536
4. ประพนธ์ อัศวภานุวัฒน์ , " Delphi Episode II " , ซีเอ็ดยูเคชั่น , กรุงเทพ ฯ , 2543
5. วสันต์ อาชาเดโชพล , ADVANCED ENGINEERING GROUP , " ระบบโทรศัพท์ดิจิตอล " , หจก.สำนักพิมพ์ ฟิสิกส์ เซ็นเตอร์ , กรุงเทพ ฯ , 2536
6. จักรพงษ์ สุขประเสริฐ , สัจจะ จรัสรุ่งรวีวร , " เริ่มต้นอย่างมืออาชีพด้วย Delphi 5 " , สำนักพิมพ์ อินโฟเพรส , กรุงเทพ ฯ , 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้