สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน

A short message system on computer via Plain Old Telephone System

โดย

นายศรัณย์พิชญ์ คุปต์กาญจนากุล

นางสาวอรุณี พิญญพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 54958
วัน,เดือน,ปี - 1 มิ.ย. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

การพัฒนาระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน

A short message system on computer via Plain Old Telephone System

โดย

นายศรัณยพิชญ์ อุปต์กาญจนากุล 43010420

นางสาวอรุณี พิญญพงษ์ 43010534

อาจารย์ที่ปรึกษา

รศ.ดร.ไกรสิน ส่งวัฒนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน

A short message system on computer via Plain Old Telephone System

ผู้จัดทำ

1. นายศรัณยพิชญ์ คุปต์กาญจนากุล 43010420

2. นางสาวอรุณี พิญญพงษ์ 43010534

.....

(รศ.ดร.ไกรสิน ส่งวัฒนา) อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน

A short message system on computer via Plain Old Telephone System

โดย นายศรัณยพิชญ์ คุปต์กาญจนากุล 43010420

นางสาวอรุณี พิญญพงษ์ 43010534

อาจารย์ที่ปรึกษา รศ.ดร. ไกรสิน ส่งวัฒนา

บทคัดย่อ

ปัจจุบันการสื่อสารข้อมูลผ่านเครือข่ายสายโทรศัพท์ได้มีการพัฒนามากขึ้น โดยสามารถส่งข้อมูลได้หลายรูปแบบโครงงานนี้จึงนำเสนอการพัฒนาระบบส่งข้อความบนคอมพิวเตอร์โดยใช้โทรศัพท์บ้านเป็นสื่อกลาง โดยใช้ Delphi ในเขียนโปรแกรมการส่งผ่านข้อมูล ซึ่งประกอบด้วย ส่วนที่เป็นลูกข่ายที่สามารถรับ - ส่งข้อความได้ และเซอฟเวอร์ที่ทำหน้าที่รับข้อความจากตัวลูกข่ายต้นทางและทำการจัดส่งข้อความไปยังลูกข่ายปลายทาง

Abstract

Nowadays communication via telephone line network has improved to transmit various type of information. This project presents a short message system (SMS) on computer which uses local telephone as a median for transmission Delphi programming is used in the software development. It consists of a client which can receive and send the message. The server receives data from input client and output to other client per request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	2
2.1 มาตรฐานการสื่อสารข้อมูลด้วยโมเด็ม	2
2.2 มาตรฐานข้อกำหนดของโมเด็ม	3
2.2.1 มาตรฐาน ITU	4
2.2.2 มาตรฐาน MNP	6
2.2.3 มาตรฐานการบีบอัดข้อมูล	7
2.2.4 มาตรฐานการควบคุมความผิดพลาด	7
2.3 องค์ประกอบในการรับส่งข้อมูล	9
2.4 การแบ่งชนิดของโมเด็ม	11
2.5 การสื่อสารข้อมูลผ่านโมเด็ม	14
2.5.1 โปรโตคอลถ่ายโอนไฟล์	16
2.5.1.1 XMODEM	16
2.5.1.2 YMODEM	17
2.5.1.3 YMODEM-g	18
2.5.1.4 ZMODEM	18
2.5.2 ประเภทของชุดคำสั่ง AT	18
2.5.3 การออนไลน์และออฟไลน์	19
2.5.4 รูปแบบของชุดคำสั่ง AT	19
2.5.5 การตอบสนองคำสั่งของโมเด็ม	19
2.5.6 ชุดคำสั่งเพิ่มเติม	20
2.5.7 การตั้งค่าในรีจิสเตอร์ S	20
2.5.8 การส่งหลายคำสั่งในครั้งเดียว	21
2.5.9 การเก็บค่าต่างๆที่เซตเอาไว้ใน Profile	21
2.5.9.1 การสร้างและการเรียกใช้ User Profile	21
2.5.9.2 การเก็บหมายเลขโทรศัพท์ไว้ในโมเด็ม	22
2.5.10 การควบคุมการไหล	22
2.5.10.1 การควบคุมการไหลแบบฮาร์ดแวร์	23
2.5.10.2 การควบคุมการไหลแบบซอฟต์แวร์	23
2.6 การพัฒนาแอพพลิเคชันบน Windows โดย Delphi 7	23
2.6.1 ความสามารถหลักๆ ของ Delphi7	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Delphi กับฐานข้อมูล 24
2.7.1ฐานข้อมูลกับความสามารถของ Delphi 7 24
2.7.2 การเข้าถึงฐานข้อมูลของ Delphi 7 25
2.7.3 Database Desktop กับงานฐานข้อมูล 25
2.8 การเรียกค้นด้วยภาษา SQL 28
2.8.1 ภาษาสำหรับการเรียกค้น 28
2.8.2 ภาษาสำหรับการจัดการข้อมูล 30
2.9 การสร้างแอพพลิเคชันฐานข้อมูลด้วยคอมโพเนนต์ 32
2.9.1 คอมโพเนนต์แหล่งข้อมูล (DataSource) 33
2.9.2 คอมโพเนนต์ต่างๆ ในเพจ BDE 34
2.9.3 คอมโพเนนต์ต่างๆ ในเพจ Data Controls 37
2.10 สร้างแอพพลิเคชันฐานข้อมูลด้วย ADO 40
2.10.1 คอมโพเนนต์ ADOConnection 41
2.10.2 คอมโพเนนต์ ADOCommand 41
2.10.3 คอมโพเนนต์ ADODataset 42
2.10.4 คอมโพเนนต์ ADOTable 42
2.10.5 คอมโพเนนต์ ADOQuery 43
2.10.6 คอมโพเนนต์ ADOStoredProc 44

บทที่ 3 การคำนวณและการสร้าง 45
3.1 คอมโพเนนต์ nrComm 45
3.1.1 ยูนิต TnrCommTAPI 46
3.1.2 ยูนิต TnrZModem 47
3.2 การทำงานในส่วนของ Client 47
3.2.1 โปรแกรมกำหนดรูปแบบการติดต่อ (Configuration) 48
3.2.2โปรแกรมในหน้า Contact 51
3.2.3โปรแกรมในหน้า Compose 55
3.2.4โปรแกรมในหน้า Inbox 58
3.2.5 Receive Message 60
3.3การทำงานในส่วนของ Server 62
3.4การออกแบบฐานข้อมูล 63

บทที่ 4 การทดลองและผลการทดลอง 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ขั้นตอนการทดลอง 68
4.2 ผลการทดลอง Client ทางด้านส่ง 68
4.3 ผลการทดลองทางด้าน Server 76
4.4 ผลการทดลองทางด้าน Client ฝั่งรับ 80

บทที่ 5 บทวิจารณ์และบทสรุป 86
ภาคผนวก
หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

บทที่ 1 บทนำ

รูปที่ 1.1 แสดงการสื่อสารข้อมูลระหว่างคอมพิวเตอร์โดยผ่านโมเด็มไปตามสายโทรศัพท์ 1

บทที่ 2 ทฤษฎีหรือหลักการ

รูปที่ 2.1 แสดงบล็อกไดอะแกรมแสดงการทำงานของโมเด็ม 2

รูปที่ 2.2 แสดงฟังก์ชั่นตามข้อกำหนด CCITT V.54 8

รูปที่ 2.3 แสดงเฟรมของข้อมูลซึ่งประกอบด้วยบิตเริ่มต้น 1 บิตข้อมูล 8 บิต
บิตพาริตี้และบิตปิดท้าย 1 บิต 9

รูปที่ 2.4 แสดงลักษณะรูปแบบการรับส่งข้อมูลของโมเด็ม 10

รูปที่ 2.5 แสดงไดอะแกรมพื้นฐานของโมเด็ม 11

รูปที่ 2.6 แสดงโมเด็มแบบต่อภายนอก 12

รูปที่ 2.7 แสดงโมเด็มแบบต่อภายใน 13

รูปที่ 2.8 แสดงฟอร์มใช้สำหรับกำหนดโครงสร้างของตารางที่ต้องการสร้าง 26

รูปที่ 2.9 แสดงชนิดต่างๆของฟิลด์ 27

รูปที่ 2.10 แสดงตารางที่ประกอบด้วยฟิลด์ต่างๆที่กำหนดขึ้นมาตามโครงสร้างของตาราง 28

รูปที่ 2.11 แสดงความสัมพันธ์ระหว่างคอมโพเนนต์ต่างๆ และการติดต่อกับฐานข้อมูล 32

รูปที่ 2.12 แสดงคอมโพเนนต์แหล่งข้อมูลในเพจ Data Access 33

รูปที่ 2.13 แสดงคอมโพเนนต์ในเพจ BDE 34

รูปที่ 2.14 แสดงคอมโพเนนต์ต่างๆ ในเพจ Data Controls 37

รูปที่ 2.15 แสดง DBNavigator เมื่อนำมาวางบนฟอร์ม 38

รูปที่ 2.16 แสดงคอมโพเนนต์ในเพจ ADO 40

บทที่ 3 การคำนวณและการสร้าง

รูปที่ 3.1 บล็อกไดอะแกรมระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน 45

รูปที่ 3.2 แสดงคอมโพเนนต์ของ nrComm 45

รูปที่ 3.3 แสดงรูปยูนิต TnrCommTAPI 46

รูปที่ 3.4 แสดงคุณสมบัติและอีเวนต์ของยูนิต TnrCommTAPI 46

รูปที่ 3.5 แสดงรูปยูนิตTnrZModem 47

รูปที่ 3.6 แสดงคุณสมบัติและอีเวนต์ของยูนิต TnrZModem 47

รูปที่ 3.7 ไดอะแกรมส่วนประกอบของโปรแกรมในส่วนของ Client 48

รูปที่ 3.8 แสดงแผนผังแสดงการทำงานของโปรแกรมกำหนดค่าพารามิเตอร์ของโมเด็ม 49

รูปที่ 3.9 แสดงส่วนที่เชื่อมต่อกับผู้ใช้ของโปรแกรมกำหนดรูปแบบการติดต่อ 50

รูปที่ 3.10 แสดงส่วนเชื่อมต่อกับผู้ใช้ในการกำหนดค่าพารามิเตอร์ของโมเด็ม 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 แสดงผังการทำงานของโปรแกรมในส่วนของ Insert Contact Data 52
รูปที่ 3.12 แสดงผังการทำงานของโปรแกรมในส่วนของ Delete Contact Data 53
รูปที่ 3.13 แสดงผังการทำงานของโปรแกรมในส่วนของ Modify Contact Data 54
รูปที่3.14 แสดงส่วนเชื่อมต่อกับผู้ใช้ในหน้า Contact 55
รูปที่ 3.15 แสดงแผนผังการทำงานของโปรแกรมในหน้า Compose 56
รูปที่ 3.16 แสดงส่วนติดต่อกับผู้ใช้ในหน้า Compose 57
รูปที่ 3.17 แสดงส่วนติดต่อกับผู้ใช้ในส่วนการยืนยันการส่งไฟล์ 57
รูปที่ 3.18 แผนผังการทำงานในหน้า Inbox 59
รูปที่ 3.19 ส่วนติดต่อกับผู้ใช้โปรแกรมในหน้า Inbox 60
รูปที่ 3.20 แสดงแผนผังการทำงานในส่วนของ Receive Message 61
รูปที่ 3.21 แสดงแผนผังการทำงานของ Server 62
รูปที่ 3.22 แสดงส่วนติดต่อกับผู้ใช้ในส่วนของ Server 63
รูปที่ 3.23 แสดงการกำหนดโครงสร้างตารางฐานข้อมูลในส่วน Contact 64
รูปที่ 3.24 แสดงตารางเพื่อใช้เก็บข้อมูลในส่วนของ Contact 64
รูปที่ 3.25 แสดงการกำหนดโครงสร้างตารางฐานข้อมูลในส่วน Inbox 65
รูปที่ 3.26 แสดงตารางเพื่อใช้เก็บข้อมูลในส่วนของ Inbox 66
รูปที่ 3.27 แสดงการกำหนดโครงสร้างตารางฐานข้อมูลในส่วน Server 66
รูปที่ 3.28 แสดงตารางเพื่อใช้เก็บข้อมูลในส่วนของ Server 67

บทที่ 4 การทดลองและผลการทดลอง
รูปที่ 4.1 แสดงฟอร์มหลักของโปรแกรม 68
รูปที่ 4.2 แสดงฟอร์มของโปรแกรมเวอร์ชั่นภาษาอังกฤษ 69
รูปที่ 4.3 แสดงฟอร์มในหน้า Contact 70
รูปที่ 4.4 แสดงฟอร์ม Contact เมื่อทำการใส่ข้อมูล ชื่อ และเบอร์โทรศัพท์แล้ว 71
รูปที่ 4.5 แสดงการแก้ไขข้อมูลในส่วน Modify 71
รูปที่ 4.6 แสดงข้อมูลใน Contact หลังจากทำการแก้ไขแล้ว 72
รูปที่ 4.7 แสดงการลบข้อมูลใน Contact 72
รูปที่ 4.8 แสดง Dialog box เพื่อขอคำยืนยันจากผู้ใช้ 72
รูปที่ 4.9 แสดงข้อมูลใน Contact หลังจากลบข้อมูล Tele4 ออกแล้ว 73
รูปที่ 4.10 แสดงฟอร์ม Configuration 73
รูปที่ 4.11 แสดงฟอร์มการกำหนดค่าพารามิเตอร์ในการสื่อสาร 74
รูปที่ 4.12 แสดงการส่งข้อความในหน้า Compose 75
รูปที่ 4.13 แสดงฟอร์ม File Attach 75
รูปที่ 4.14 แสดงฟอร์ม File Attach ขณะทำการส่งข้อความ 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.15 แสดงฟอร์ม Server เมื่อทำการเปิดโปรแกรม 77
รูปที่ 4.16 แสดงฟอร์มของ Server ขณะทำการรับไฟล์ จาก Client ทางด้านส่ง 78
รูปที่ 4.17 แสดงฟอร์ม Server ในส่วนของการแสดงสถานะของ database 79
รูปที่ 4.18 แสดงฟอร์ม Server ในส่วนของการแสดงรายละเอียดการรับส่งไฟล์ 79
รูปที่ 4.19 แสดงฟอร์ม Server ในส่วนของการจัดการข้อมูลใน database 80
รูปที่ 4.20 แสดงฟอร์ม inbox 81
รูปที่ 4.21 แสดงฟอร์ม inbox เมื่อมีการเข้ามาของข้อความ 81
รูปที่ 4.22 แสดงฟอร์ม inbox เมื่อเปิดอ่านข้อความ 82
รูปที่ 4.23 แสดง Dialog box ขอคำยืนยันจากผู้ใช้ 82
รูปที่ 4.24 แสดงฟอร์มcompose เมื่อทำการคลิก Reply จากฟอร์ม inbox 83
รูปที่ 4.25 แสดงรายละเอียดของไฟล์ที่ถูกแนบมา 84
รูปที่ 4.26 แสดงไฟล์ที่ถูกแนบมายัง Client ฝั่งรับ 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

บทที่ 2 ทฤษฎีหรือหลักการ

ตารางที่ 2.1 แสดงคุณสมบัติของโมเด็มของมาตรฐาน ITU ที่ใช้ในการสื่อสาร 6

ตารางที่ 2.2 แสดงกระบวนการทำงานของโมเด็มต้นทางและปลายทางตั้งแต่ขั้นตอน 1-12 15

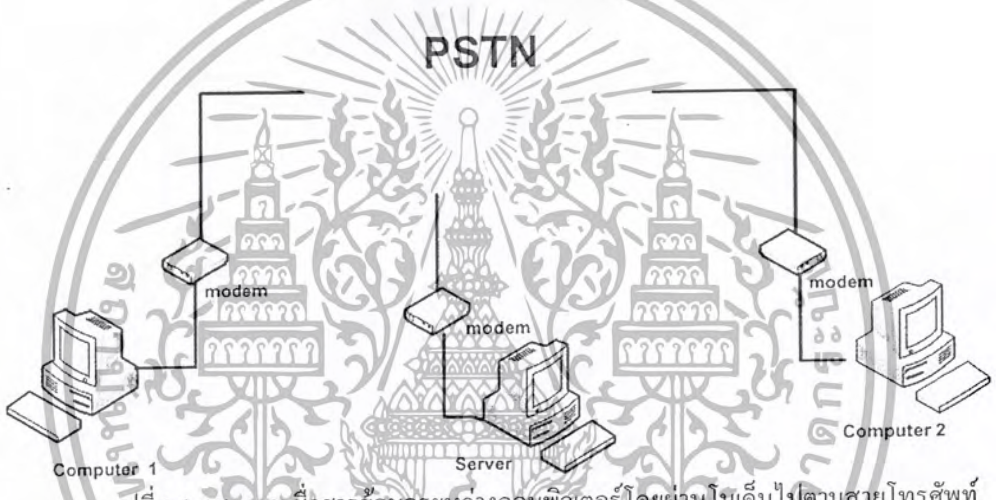
ตารางที่ 2.3 แสดงบล็อคของ XMODEM 16

ตารางที่ 2.4 แสดงรูปแบบบล็อคของ XMODEM ที่ใช้ตัวเลือก CRC 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

สังคมปัจจุบันเป็นสังคมแห่งข้อมูลข่าวสาร จะเห็นได้ว่าไม่ว่าจะเป็นตามบ้านเรือน โรงเรียนหรือสำนักงาน ได้มีการนำเทคโนโลยีของคอมพิวเตอร์ไปประยุกต์ใช้งานต่างๆ กันอย่างแพร่หลาย โดยเฉพาะอย่างยิ่งพบว่าการสื่อสารข้อมูลมีความจำเป็นอย่างมากต่อทุกระบบ เนื่องจากทุกระบบต้องการการสื่อสารข้อมูลที่มีความปลอดภัย รวดเร็วแม่นยำสูงเพื่อให้การทำงานของระบบเป็นไปอย่างราบเรียบ ดังนั้นโครงงานนี้ได้นำเสนอการพัฒนาระบบส่งข้อความสั้นๆ (Short Message Service) ระหว่างบุคคล ให้มีประสิทธิภาพสูงขึ้นโดยสามารถส่งเป็นข้อความ ไฟล์ข้อมูล และรูปภาพได้ โดยใช้คอมพิวเตอร์ส่วนบุคคลมาใช้สื่อสารข้อมูลระหว่างกันโดยผ่านโมเด็มไปตามสายโทรศัพท์ ดังรูปที่ 1.1



รูปที่ 1.1 แสดงการสื่อสารข้อมูลระหว่างคอมพิวเตอร์โดยผ่านโมเด็มไปตามสายโทรศัพท์

จากรูปที่ 1.1 เมื่อ Computer1 ทำการส่งข้อมูลไปที่ Server เมื่อ Server ได้รับก็จะทำการเก็บข้อมูลลงใน Database แล้วจึงทำการส่งข้อมูลออกไปยังปลายทางคือ Computer2 แต่ถ้าหาก Computer2 ไม่สามารถที่จะติดต่อได้ Server จะทำการส่งข้อมูลอื่นๆ ใน Database ก่อนที่จะกลับมาทำการส่งไปยัง Computer2 อีกครั้งหนึ่ง เมื่อ Computer2 รับข้อมูลได้แล้ว ข้อมูลก็จะถูกเก็บลงใน Database ซึ่งจะแสดงผลออกมาในหน้าต่าง Inbox

แนวคิดสำหรับโครงงานนี้เนื่องจากต้องการให้ทุกคนสื่อสารกันได้ง่ายขึ้น และใช้ทรัพยากรที่มีอยู่อย่างคุ้มค่า เพราะว่าโดยทั่วไปมีการใช้งานด้วยคอมพิวเตอร์กันอย่างแพร่หลาย เนื่องจากคอมพิวเตอร์ในปัจจุบันมีราคาถูก สามารถทำงานได้รวดเร็ว และสามารถเชื่อมโยงกันเป็นระบบเครือข่าย เพื่อแลกเปลี่ยนข่าวสารกันได้ และต้องการใช้สายโทรศัพท์ที่มีอยู่แล้วอย่างคุ้มค่าที่สุด

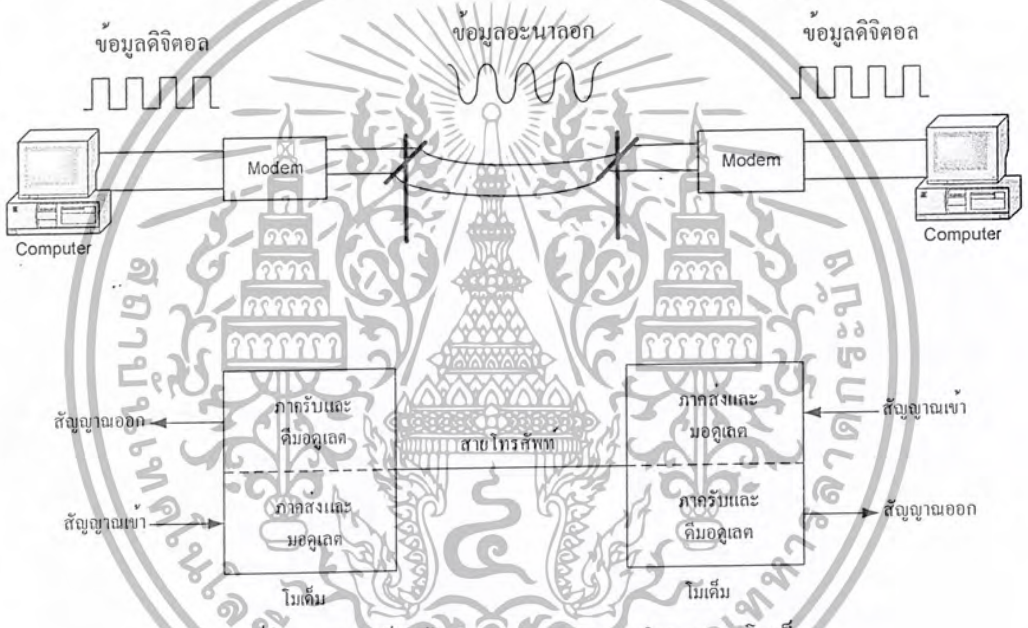
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีหรือหลักการ

2.1 มาตรฐานการสื่อสารข้อมูลด้วยโมเด็ม

คอมพิวเตอร์ทั่วไปนั้นไม่สามารถจะเชื่อมโยงกันได้โดยตรง จึงต้องมีการส่งสัญญาณผ่านสายโทรศัพท์แทนโดยมีอุปกรณ์ที่เข้ามาช่วยในการส่งสัญญาณซึ่งก็คือ โมเด็ม (Modem)

โมเด็มเป็นชื่อที่ใช้เรียกอุปกรณ์ที่ประกอบไปด้วยวงจรมอดูเลเตอร์ (Modulator) และวงจรดีมอดูเลเตอร์ (Demodulator) เอาไว้ด้วยกัน ซึ่งเป็นอุปกรณ์ที่ทำหน้าที่ในการจัดการสัญญาณที่จะส่งให้อยู่ในย่านความถี่ที่เหมาะสมกับตัวกลางที่จะส่งผ่าน โดยหลักการง่ายๆของโมเด็มจะเปลี่ยนสัญญาณเอาท์พุตที่เป็นดิจิตอลจากเครื่องพีซี ส่งออกเป็นสัญญาณอะนาลอกผ่านสายโทรศัพท์ และทำหน้าที่รับสัญญาณอะนาลอกที่ได้จากโทรศัพท์ เปลี่ยนเป็นสัญญาณดิจิตอลส่งเข้าสู่คอมพิวเตอร์ ซึ่งแสดงการทำงานดังในรูปที่ 2.1



รูปที่ 2.1 แสดงบล็อกไดอะแกรมแสดงการทำงานของโมเด็ม

โดยปกติเมื่อโมเด็มติดต่อกันได้แล้วจะไม่สามารถรับส่งข้อมูลได้ในทันที ต้องทำการตกลงรายละเอียดของวิธีการรับส่งข้อมูลกันก่อน เรียกว่าการทำ Handshaking ซึ่งมีขั้นตอนดังนี้คือ เมื่อโมเด็มปลายทางตอบรับแล้ว โมเด็มต้นทางจะทำการทดสอบสมรรถภาพสายก่อนว่าสามารถรับส่งข้อมูลได้ดีเพียงใด เพื่อที่จะได้เลือกความเร็วที่สูงที่สุดเท่าที่สายจะรับได้ จากนั้นจึงทำการทดสอบความเพี้ยนของสัญญาณต่อไป หลังจากขั้นตอนนี้โมเด็มจะตกลงกับปลายทางได้ว่าจะใช้ความเร็วในการรับส่งข้อมูลเท่าไร ใช้การผสมข้อมูลแบบไหน ที่ความถี่เท่าใด ถ้าใช้ความเร็วสูงสุดตามที่กำหนดไม่ได้โมเด็มก็จะทำการลดความเร็วลงและทดสอบสภาพสายใหม่จนกระทั่งได้ความเร็วที่สามารถรับส่งข้อมูลได้อย่างถูกต้อง ซึ่งการลดความเร็วของโมเด็มจะลดลงไปเป็นขั้นๆ ตามแต่ละมาตรฐานที่กำหนดไว้ จากนั้นโมเด็มจะตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลงกับปลายทางว่าจะใช้วิธีการตรวจสอบความผิดพลาด (Error Detection) แบบใด และใช้การลดขนาดข้อมูล (Data Compression) หรือไม่ ขั้นตอนในการทำ Handshaking ของโมเด็มทั้งหมดจะใช้เวลาหลายสิบวินาที เมื่อเสร็จเรียบร้อยแล้วจึงสามารถส่งข้อมูลกันได้

โมเด็มมีการพัฒนาขึ้นมาเรื่อยๆจนในปัจจุบันสามารถส่งข้อมูลได้ทั้งเป็นรูปภาพและเสียง การบอกคุณลักษณะหน้าที่ของโมเด็มจะบอกเป็น V.32, V.42bis ซึ่งจะทำให้เข้าใจคุณสมบัติเฉพาะของโมเด็มแต่ละรุ่น และยังทำให้ทราบว่ารุ่นใดเป็นรุ่นล่าสุดอีกด้วย

2.2 มาตรฐานข้อกำหนดของโมเด็ม

เนื่องจากแบนด์วิดท์ของวงจรโทรศัพท์มีช่วงจำกัดเท่ากับ 300-3,400 Hz จัดว่าไม่กว้างพอที่จะส่งสัญญาณดิจิตอลได้ ดังนั้นจำเป็นต้องมีการแปลงสัญญาณดิจิตอลเป็นสัญญาณอะนาลอกในช่วงความถี่เสียง เพื่อให้สามารถส่งผ่านเน็ตเวิร์คของโทรศัพท์ได้ และที่ปลายทางรูปสัญญาณจะถูกแปลงกลับเป็นสัญญาณดิจิตอลเหมือนเดิม หรือก็คือต้องมีอุปกรณ์ D/A และ A/D ในตัวอุปกรณ์โมเด็ม

ตามมาตรฐานของ CCITT นิยามให้โมเด็มหมายถึง Data Communication Equipment (DCE) และมาตรฐานของ EIA นิยามให้โมเด็มหมายถึง Data Circuit Terminating Equipment (DCE) เราติดตั้งเครื่องโมเด็มที่ปลายจุดต่อแต่ละด้านผ่านเน็ตเวิร์คโทรศัพท์ โดยใช้วงจรให้ Leased circuit หรือผ่านการหมุนหมายเลขของเครือข่ายโทรศัพท์สาธารณะ (Dial-Up Connection Via the PSTN)

ฟังก์ชั่นการหมุนหมายเลขและตอบรับอย่างอัตโนมัติมีอยู่ในโมเด็มทุกเครื่อง และบางแบบอาจมีฟังก์ชั่นให้รักษาการสื่อสารเมื่อช่องสัญญาณติดต่อล้มเหลว โดยเปลี่ยนไปหาช่องสัญญาณอื่นแทน นอกจากนี้ในโมเด็มยังมีฟังก์ชั่นตรวจสอบและแก้ไขข้อมูลที่ผิดพลาดด้วย

คอมพิวเตอร์ที่ใช้ในการสื่อสารโดยทั่วไปจะพัฒนาขึ้นโดยอาศัยมาตรฐานของ ITU (International Telecommunication Union) ซึ่งมาตรฐานนี้เป็นมาตรฐานที่ใช้กันทั่วโลก ประกอบด้วยมาตรฐานของการผลิตโมเด็ม, การแลกเปลี่ยนข้อมูลร่วมกัน และการควบคุมการใช้งาน ซึ่งตามมาตรฐานของ ITU จะบอกเป็นตัวอักษรย่อขึ้นต้นด้วยตัว V และมีตัวเลขตามหลัง หมายถึงคุณสมบัติเฉพาะของแต่ละรุ่น บางรุ่นจะมีตัวอักษร “bis” ต่อท้ายอีกซึ่งหมายถึงมีคุณสมบัติเฉพาะย่อยลงไปอีก และบางทีอาจเห็นคำว่า “Turbo” ต่อท้ายซึ่งก็หมายถึงคุณสมบัติเฉพาะลงไปอีกเป็นอย่างที่ 3

มาตรฐานการส่งสัญญาณในช่วงแรกใช้มาตรฐานของ North America Telecommunication Standard ซึ่งต่อมาแยกเป็นบริษัท AT&T และ Seven Regional Telephone Operating Companies ในปี 1984 ซึ่งก่อนหน้านี้ได้มีการสร้างมาตรฐานการใช้งานโมเด็มขึ้นมา 2 อย่างคือ BELL103 และ BELL212A

BELL103 เป็นมาตรฐานที่ได้รับการยอมรับอย่างกว้างขวางโดยใช้เทคนิคของ FSK (Frequency Shift Keying) ที่มีอัตราบอด (Baud) เท่ากับ 300 โดยเป็นแบบอัตราการส่งข้อมูลเท่ากับอัตราบอดหรือ 1 เท่า ซึ่งมาตรฐานนี้มีใช้ในปัจจุบันและจะใช้เมื่อเทคนิคของมาตรฐานอื่นใช้ได้ไม่ดี

BELL212A คือมาตรฐานที่ 2 ที่มีใช้กันอย่างกว้างขวางในอเมริกาเหนือ โดยใช้เทคนิค PSK (Phase Shift Keying) ที่มีอัตราบอดเท่ากับ 600 สามารถส่งสัญญาณได้ 1,200 บิตต่อวินาที และในยุโรปใช้ระบบนี้มาก และเรียกว่า V.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 มาตรฐาน ITU

หลังจากที่มีการสร้างมาตรฐานการส่งสัญญาณของบริษัท AT&T ขึ้นมา แต่ปรากฏว่าไม่เป็นที่แพร่หลาย ในขณะเดียวกันก็มีมาตรฐาน ITU เกิดขึ้น และเป็นที่ยอมรับมากในอเมริกา ดังนั้นโมเด็มส่วนใหญ่ในอเมริกาจึงใช้มาตรฐาน ITU ตารางที่ ก.1 แสดงคุณสมบัติของมาตรฐาน ITU ที่ใช้ในการสื่อสารแต่ละรุ่นซึ่งจะช่วยในการตัดสินใจเลือกใช้ได้ตามคุณสมบัติที่ต้องการได้ง่ายขึ้น

V.1-A	เป็นเวอร์ชั่นแรกที่สร้างขึ้น โดยการส่งสัญญาณเป็นเลขฐานสองบนแรงดันที่เหมาะสม
V.2	เริ่มมีการใช้สัญญาณเสียงส่งไปตามคู่สายโทรศัพท์
V.4	ส่งข้อมูลไปเป็นเฟรมของข้อมูล
V.5	สร้างมาตรฐานของสัญญาณที่ใช้ทางโทรศัพท์
V.6	สร้างมาตรฐานของสัญญาณที่มีการใช้ทางสายนำสัญญาณ
V.7	เริ่มใช้โมเด็มเพื่อการสื่อสารในประเทศอังกฤษ, สเปนและฝรั่งเศส
V.10	ส่งสัญญาณความเร็วสูงผ่านการเชื่อมต่อทางไฟฟ้าแบบไม่บาลานซ์ (RS-423)
V.11	ส่งสัญญาณความเร็วสูงผ่านการเชื่อมต่อทางไฟฟ้าแบบบาลานซ์ (RS-422)
V.14	เปลี่ยนรูปแบบการส่งสัญญาณจากแบบ Asynchronous ไปเป็นแบบ Synchronous
V.15	เริ่มมีการใช้สายโทรศัพท์และมีใช้กันจนถึงปัจจุบัน เพราะระบบโทรศัพท์สามารถเชื่อมต่อกันได้ง่าย และมีการใช้กันอย่างแพร่หลาย
V.17	เป็นมาตรฐานที่สร้างมาเพื่อใช้งานกับการรับส่งแฟกซ์ที่ใช้การส่งแบบฮาร์ฟดูเพล็กซ์ที่ความเร็ว 7,200, 9,600, 12,000 และ 14,400 บิตต่อวินาที ถึงแม้ว่าจะมีความเร็วค่อนข้างต่ำแต่ก็เป็นระบบพื้นฐานของมาตรฐานที่ใช้ในปัจจุบัน
V.19	เป็นมาตรฐานการส่งข้อมูลแบบ DTMF สำหรับโมเด็มความเร็วต่ำซึ่งใช้การส่งข้อมูลแบบเก่าซึ่งเป็นแบบขนาน
V.21	เป็นการส่งข้อมูลแบบ FSK ซึ่งเป็นการส่งข้อมูลแบบลำดับทีละ 300 บิตต่อวินาที
V.22	ส่งข้อมูลแบบ PSK ส่งได้ทีละ 600 หรือ 1,200 บิตต่อวินาที
V.22 bis	เป็นการส่งข้อมูลแบบ QAM สามารถส่งข้อมูลได้ 2,400 บิตต่อวินาที
V.23	ใช้การส่งสัญญาณ FM แบบ 1,200/75 บิตต่อวินาที คือตัวส่งสามารถส่งสัญญาณได้ 1,200 บิตต่อวินาทีแต่รับได้ 75 บิตต่อวินาที ตัวรับจะส่งข้อมูล 75 บิตต่อวินาที รับข้อมูลได้ 1,200 บิตต่อวินาที
V.24	เป็นเวอร์ชั่นที่ใช้ในอเมริกาซึ่งเรียกว่า EIA RS-232C ซึ่งใช้การส่งเป็นสัญญาณไฟฟ้าผ่านทางช่องสัญญาณ EIA 232 E
V.25	เป็นการรับส่งข้อมูลแบบอัตโนมัติ และสามารถแสดงเสียงเมื่อมีการส่งข้อมูลด้วย
V.25 bis	เป็นการรับส่งข้อมูลแบบอัตโนมัติ ภายใต้มาตรฐาน ITU ซึ่งใช้คำสั่งแบบ AT และใช้การสื่อสารแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V.26	เป็นการส่งข้อมูลความเร็ว 2,400 บิตต่อวินาที แบบ PSK และการส่งจะเป็นแบบฟูลดูเพล็กซ์ โดยมีอัตราบอด 1,200 บิต
V.26 bis	เป็นการส่งข้อมูลความเร็ว 2,400 บิตต่อวินาที แบบ PSK และการส่งจะเป็นแบบฮาร์ฟดูเพล็กซ์ โดยมีอัตราบอด 1,200 บิต
V.26 turbo	เป็นแบบ PSK ฟูลดูเพล็กซ์ ซึ่งสามารถเลือกสัญญาณได้ว่าจะเป็นแบบ 1,200 บิตต่อวินาที หรือ 2,400 บิตต่อวินาที
V.27	ส่งข้อมูลด้วยความเร็ว 4,800 บิตต่อวินาที ด้วยอัตราบอด 1,600 บอด
V.27 bis	เหมือนกับระบบ V.27 แต่ใช้งานได้กว้างกว่าคือ ที่อัตราบอด 1,600/1,200 สามารถส่งสัญญาณได้ 4,800/2,400 บิตต่อวินาที
V.28	เป็นการส่งสัญญาณร่วมกับแบบ V.24 โดยจะกำหนดค่าที่จำเป็นทั้งหมดสำหรับการส่งตามมาตรฐานของพอร์ตอนุกรม RS-232C
V.29	เป็นมาตรฐานที่นิยมใช้กันมากโดยที่อัตราบอด 2,400 บอดสามารถส่งข้อมูลแบบ PSK/QAM ได้ที่ความเร็ว 9,600/7,200/4,800 บิตต่อวินาที
V.32	เป็นมาตรฐานที่นิยมใช้กันมากเพราะเป็นการสื่อสารที่มีประสิทธิภาพสูง รวดเร็ว โดยส่งข้อมูลในระบบ QAM แบบฟูลดูเพล็กซ์ ที่อัตราบอด 2,400 บิต มีความเร็ว 9,600/4,500 บิตต่อวินาที
V.32 bis	เป็นการพัฒนามาจาก V.32 โดยการส่งข้อมูลแบบ TC-QAM, ฟูลดูเพล็กซ์ อัตราบอด 2,400 บอด ที่ความเร็ว 4,800/7,200/9,600/12,000/14,400 บิตต่อวินาที
V.32 turbo	พัฒนามาจาก V.32 โดยการเพิ่มอุปกรณ์และเทคนิคอื่นๆ เข้าไป โดยทำงานด้วยระบบ TC-QAM ฟูลดูเพล็กซ์ อัตราบอด 2,400 บอด ที่ความเร็ว 14,400/16,800/19,200 บิตต่อวินาที แต่ก็ไม่เป็นที่นิยมใช้เนื่องจากราคาสูงเกินไป
V.32 fast	เป็นชื่อที่ไม่เป็นทางการ โดยสามารถใช้แทน V.32 bis แต่เพิ่มความเร็วขึ้นเป็น 28,800 บิตต่อวินาที ซึ่งเป็นมาตรฐานสุดท้ายที่มีการส่งสัญญาณแบบอะนาล็อก บางทีใช้ชื่อว่า V.34
V.33	ส่งข้อมูลแบบ TC-QAM, ฟูลดูเพล็กซ์ อัตราบอด 2,400 บอด ที่ความเร็ว 14,400 บิตต่อวินาที
V.34	ซึ่งเป็นการพัฒนา V.32 fast นั่นเอง โดยมีความเร็วในการส่งข้อมูล 128 กิโลบิตต่อวินาที (kbps) ที่อัตราบอด 3,429 บอด
V.36	สามารถส่งข้อมูลได้มากถึง 45,000 บิตต่อวินาที ใช้มาในระบบที่มีการเชื่อมโยงกันเป็นกลุ่ม
V.42 bis	พัฒนามาจาก V.42 โดยเพิ่มส่วนการบีบอัดข้อมูล
V.50	มาตรฐานการควบคุมคุณภาพของการส่งข้อมูลผ่านทางสายโทรศัพท์
V.51	มาตรฐานที่มีการปรับปรุงให้เป็นแบบสากล
V.52	มาตรฐานการตรวจวัดการลดทอนและอัตราตรวจสอบความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V.54	มาตรฐานที่วัดด้วยอุปกรณ์สำหรับตรวจสอบโมเด็ม
V.55	มาตรฐานสำหรับการวัดสัญญาณรบกวนภายในคู่สาย
V.56	มาตรฐานการตรวจสภาพของโมเด็มเทียบกับมาตรฐาน
V.100	ระบบการพัฒนามาตรฐานและเทคนิคร่วมกันระหว่าง PNDS (Public Data Network) และ PSTNS (Public Switched Telephone Networks)

ตารางที่ 2.1 แสดงคุณสมบัติของโมเด็มของมาตรฐาน ITU ที่ใช้ในการสื่อสาร

2.2.2 มาตรฐาน MNP

ในช่วงกลางปี 1980 ได้มีการพัฒนาเทคโนโลยีของการสื่อสารด้วยโมเด็มขึ้นโดยบริษัท Microcom ซึ่งยังมีการใช้มาตรฐานนี้อย่างกว้างขวางจนถึงปัจจุบันโดยเรียกว่า มาตรฐาน MNP (Microcom Networking) ซึ่งได้มีการพัฒนารูปแบบการสื่อสารระหว่างโมเด็มให้ดีขึ้น ประกอบด้วยส่วนการตรวจสอบความผิดพลาด และการบีบอัดข้อมูลซึ่งช่วยให้ระบบที่ทำงานมีความถูกต้องและส่งได้รวดเร็วยิ่งขึ้นโดย MNP จะแบ่งเป็น class ต่างๆกันดังนี้

MNP Class1 (block mode) เป็นการรับส่งข้อมูลแบบทิศทางเดียว ซึ่งการส่งแบบนี้มีความเร็วเป็น 70 เปอร์เซ็นต์ของการส่งแบบไม่มีการตรวจสอบความผิดพลาด ซึ่งปัจจุบันมาตรฐานตัวนี้เลิกใช้ไปแล้ว

MNP Class2 (stream mode) เป็นการรับและส่งข้อมูลแบบ 2 ทิศทาง มีความเร็วเป็น 84 เปอร์เซ็นต์ของการส่งแบบไม่มีการตรวจสอบความผิดพลาด

MNP Class3 การส่งข้อมูลผ่านโมเด็มโดยตัด บิตเริ่มต้นและบิตปิดท้ายออก ส่วนที่โมเด็มตัวรับจะมีการเติมบิตเริ่มต้นและบิตปิดท้ายเข้าไป วิธีนี้มีความเร็วมากกว่าการส่งข้อมูลแบบไม่ตรวจสอบความผิดพลาด 8 เปอร์เซ็นต์

MNP Class4 มาตรฐานการส่งสัญญาณรูปแบบนี้จะตรวจสอบสัญญาณรบกวนภายในสายก่อน ถ้าสัญญาณรบกวนในสายมีน้อยจะส่งข้อมูลกลุ่มใหญ่ๆออกไป แต่ถ้าสัญญาณรบกวนมีมากก็จะส่งข้อมูลกลุ่มน้อยๆออกไปและมีการตรวจสอบความผิดพลาดมากขึ้น

MNP Class5 จะมีการบีบอัดข้อมูลโดยตรวจสอบข้อมูลที่มีลักษณะเหมือนกันหลายๆตัวมารวมกันเป็นรหัสทำให้มีจำนวนบิตของข้อมูลน้อยลง โมเด็มภาครับจะทำการคืนค่าข้อมูลเดิมก่อนจะส่งออกไปยังคอมพิวเตอร์ภาครับ ซึ่งการส่งข้อมูลแบบ MNP5 นี้จะมีความเร็วเป็น 2 เท่าของการส่งข้อมูลแบบไม่มีการบีบอัดข้อมูล หรือการส่งข้อมูลแบบไม่มีการตรวจสอบความผิดพลาด

MNP Class6 เป็นระบบที่ใช้กันเป็นระบบสากลที่ดึงเอาความสามารถการส่งผ่านสายโทรศัพท์มาใช้ให้เต็มที่ ช่วงแรกโมเด็มจะทำงานที่ความเร็วต่ำและเพิ่มความเร็วสูงขึ้นจนถึงความเร็วที่สูงที่สุดที่โมเด็มทั้งตัวรับและตัวส่งสามารถทำงานได้ดี

MNP Class7 เป็นลักษณะของการบีบอัดข้อมูลเช่นเดียวกับ MNP5 แต่ MNP7 สามารถเพิ่มการส่งข้อมูลได้มากขึ้น 3 เท่า

MNP Class8 ยังไม่มีระบบนี้ใช้ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MNP Class9 จะลดส่วนหัวของข้อมูลลงในทุกๆ ช่วงและแก้ไขประสิทธิภาพของส่วนตรวจสอบความผิดพลาดให้มีประสิทธิภาพดีขึ้น เพราะการส่งข้อมูลไปใหม่ เมื่อเกิดความผิดพลาดนั้นจะส่งไปเฉพาะช่วงที่เกิดผิดพลาดเท่านั้น

MNP Class10 เป็นมาตรฐานที่รู้จักกันในชื่อของ adverse channel enhancements เพื่อช่วยโมเด็มให้สามารถต่อสายโทรศัพท์ที่มีสัญญาณรบกวนมาก โดยการปรับขนาดของช่วงการส่งข้อมูลและความเร็วในการส่งข้อมูลให้ต่ำลงเมื่อมีสัญญาณรบกวน จนกระทั่งการรับข้อมูลวางใจได้จึงกลับมาทำงานเป็นปกติอีกครั้ง

2.2.3 มาตรฐานการบีบอัดข้อมูล

การส่งข้อมูลระหว่างโมเด็มทั้งหมดประกอบด้วยข้อมูลที่ซ้ำๆ กัน หรือเหมือนกันเป็นช่วงๆ ถ้าข้อมูลที่ซ้ำๆ เหล่านี้ถูกแทนที่ด้วยเครื่องหมายเล็กๆ ระหว่างการส่งข้อมูล ข้อมูลจะถูกบีบอัดให้มีขนาดเล็กลงได้

การบีบอัดข้อมูลกลายเป็นเทคนิคที่สำคัญที่ช่วยให้โมเด็มสามารถเพิ่มความเร็วในการส่งข้อมูลโดยไม่ต้องเพิ่มอัตราบอดหรือความเร็วในการส่งข้อมูล การบีบอัดข้อมูลเกิดขึ้นได้เฉพาะเมื่อโมเด็มที่ใช้ในการสื่อสารทั้งสองตัวสนับสนุนมาตรฐานการบีบอัดข้อมูลเท่านั้น

MNP Class5 เป็นมาตรฐานที่สามารถลดขนาดข้อมูลลงครึ่งหนึ่ง ซึ่งทำให้ MNP5 ที่ส่งข้อมูลได้ 2,400 บิตต่อวินาที สามารถส่งข้อมูลได้ 4,800 บิตต่อวินาที และโมเด็มที่ทำงานที่ความเร็ว 9,600 บิตต่อวินาที สามารถส่งข้อมูลได้ถึง 19,200 บิตต่อวินาที

MNP Class7 สามารถเพิ่มอัตราบีบอัดข้อมูลได้ถึง 3:1 หรือ 3 เท่า มีประสิทธิภาพมากกว่า MNP5 แต่มีโมเด็มเพียงไม่กี่รุ่นที่สนับสนุนมาตรฐานนี้

V.42 bis เป็นมาตรฐานที่รวมเอามาตรฐาน V.42LAPM (ชุดตรวจสอบความผิดพลาด) V.42 bis สามารถบีบอัดข้อมูลได้ในอัตรา 4 ต่อ 1 (เป็นอัตราการส่งข้อมูลที่ไว้ใจได้) ทำให้โมเด็มที่มีอัตราบอด 9,600 สามารถส่งข้อมูลได้ 38,400 บิตต่อวินาที เมื่อใช้มาตรฐาน V.42 bis

2.2.4 มาตรฐานการควบคุมความผิดพลาด

โมเด็มส่วนใหญ่จะมีการตรวจสอบความผิดพลาดของข้อมูลที่เกิดขึ้นในการส่งข้อมูลระหว่างโมเด็ม และจะส่งข้อมูลที่พบว่าผิดพลาดไปอีกครั้งจนกระทั่งข้อมูลฝ่ายรับนั้นถูกต้อง แต่การทำงานของส่วนการตรวจสอบความผิดพลาดโมเด็มจะต้องมีมาตรฐานในการตรวจสอบความผิดพลาดรูปแบบเดียวกัน

MNP Class4 เป็นมาตรฐานการตรวจสอบความผิดพลาดของบริษัท Microcom จะตรวจสอบคุณภาพของระบบโทรศัพท์และปรับเปลี่ยนส่วนหัวของข้อมูลที่จะส่งออกไป โดยใช้เทคนิคที่เรียกว่า adaptive packet assembly ซึ่งการส่งข้อมูลจะสัมพันธ์กับสัญญาณรบกวนในสายโทรศัพท์ ถ้าสายโทรศัพท์มีสัญญาณรบกวนน้อยก็จะส่งข้อมูลบล็อกขนาดใหญ่ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

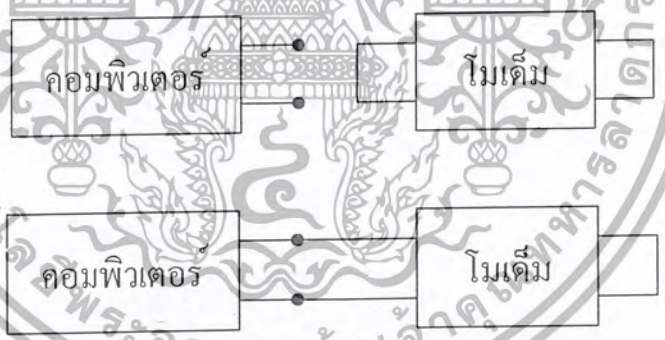
V.42 เป็นมาตรฐานการตรวจสอบความผิดพลาดของ ITU สำหรับใช้ในมาตรฐานโมเด็ม V.22, V.22 bis, V.26 turbo และ V.32 bis ITU V.42 นี้มีประสิทธิภาพสูงมากและมีความเร็วในการส่งข้อมูลสูงกว่า NMP4 ถึง 20 เปอร์เซ็นต์

เหตุผลที่ต้องใช้วิธีมอดูเลตหลายชนิด เนื่องจากแต่ละวิธีมีความเหมาะสมกับแต่ละแบบโมเด็มเมื่อพิจารณาในแง่ประสิทธิภาพและราคาที่อัตราบิตต่างกัน

ในแง่ของการจัดการของข้อมูลในเน็ตเวิร์ค โมเด็มได้ออกแบบให้มีช่องสัญญาณช่องที่ 2 (นอกจากช่องสัญญาณที่ 1 ที่ทำหน้าที่นำข้อมูล) ซึ่งเมื่อข้อมูลถูกส่งไปในช่องสัญญาณช่องที่ 1 เราจะใช้ช่องสัญญาณช่องที่ 2 ในการตรวจสอบระบบ, ส่งคำสั่งควบคุมฯ โดยช่องสัญญาณช่องที่ 1 ทำงานตามความเร็วข้างต้น ส่วนช่องสัญญาณช่องที่ 2 ทำงานที่ความเร็วต่ำระหว่าง 75 บิต/วินาที และ 150 บิต/วินาที

โมเด็มบางรุ่นถูกออกแบบให้มีความสามารถสูงในระดับประมวลผลเองได้โดยใช้อุปกรณ์ประเภทไมโครโปรเซสเซอร์ ทำให้โมเด็มสามารถตรวจสอบสถานะของตัวเองได้ เช่น ระดับแรงดันไฟฟ้าที่ใช้, จุดผิดพลาดที่อาจเกิดขึ้นได้, อัตราส่วนของสัญญาณหลักต่อสัญญาณรบกวน, รวมถึงการแจ้งกลับไปยังคอมพิวเตอร์ที่ทำหน้าที่ควบคุมเน็ตเวิร์คด้วย

โมเด็มต่ออยู่กับคอมพิวเตอร์ หรือเทอร์มินอล โดยวิธีอินเตอร์ฟิว ซึ่งประกอบด้วยปลั๊ก, ซอตเก็ต, ขาต่อ และสายเคเบิล ที่ต้องมีคุณสมบัติทั้งทางไฟฟ้าและทางกลเข้ากับอุปกรณ์ที่ต่อด้วย คุณสมบัติทางไฟฟ้า แทนด้วยรหัสเลขฐานสองคือ "0" และ "1" หรือแทนด้วยค่าแรงดันไฟฟ้าประมาณ ±6 V หรือ 0 V และ +5 V มีโมเด็มหลายชนิดที่มีฟังก์ชันการทดสอบวิเคราะห์ระบบอยู่ภายในเครื่องตามข้อกำหนด CCITT V.54 มีฟังก์ชันดังต่อไปนี้



รูปที่ 2.2 แสดงฟังก์ชั่นตามข้อกำหนด CCITT V.54

(a) เซลฟ์เทส (Self-test)

(b) โลเคิลอะนาลอกลูป (Local analog loopback)

อธิบายได้ดังนี้

(a) เซลฟ์เทส เป็นวิธีตามรูปที่ 2.1(a) ใช้ทดสอบฮาร์ดแวร์ของโมเด็มโดยจุดต่ออะนาลอกถูกลูปเข้าหากัน (หมายถึงต่อจุดต่อออกเข้าหากัน) จากนั้นโมเด็มทำการส่งบิตข้อมูลที่ได้เตรียมไว้เพื่อการทดสอบโดยเฉพาะส่งออกไปที่จุดออกด้านหนึ่งและรับบิตเข้าอีกด้านหนึ่ง เพื่อนำบิตมาเปรียบเทียบกัน ถ้า

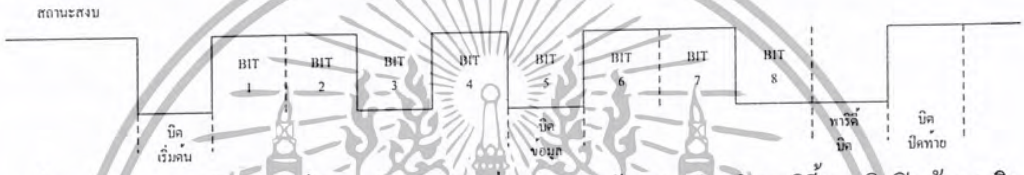
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่มีบิตผิดพลาดโมเด็มสามารถทำงานได้ตามปกติ ซึ่งวิธีนี้มีข้อดีคือ โอเปอเรเตอร์หรือผู้ใช้ไม่จำเป็นต้องคีย์ข้อมูลลงไปและคอยตรวจสอบข้อมูลจากจอแสดงผล

(b) โลเคิลอะนาลอกลูป เป็นวิธีตามรูปที่ 2.1(b) โดยลูปจุดต่ออะนาลอกเข้าหากันแต่ยังคงการติดต่อระหว่างคอมพิวเตอร์กับโมเด็ม วิธีนี้ข้อมูลส่งไปจากโมเด็มจะย้อนกลับเข้าโมเด็มเครื่องเดิม เพื่อเป็นการทดสอบการทำงานระหว่างคอมพิวเตอร์กับโมเด็ม ซึ่งเราใช้วิธีนี้ทดสอบการอินเตอร์เฟสของโมเด็มด้วย

2.3 องค์ประกอบในการรับส่งข้อมูล

โมเด็มสำหรับเครื่องคอมพิวเตอร์พีซีส่วนใหญ่จะทำงานแบบอะซิงโครนัสนั่นคือสัญญาณที่ใช้ในการรับส่งของตัวโมเด็มจะไม่ขึ้นอยู่กับสัญญาณนาฬิกา (Clock) ซึ่งการทำงานเหล่านี้จะสามารถประมวลผลได้ว่าข้อมูลใดได้รับก่อนหรือหลัง และจัดลำดับได้เป็นกลุ่มข้อมูล เรียกว่าดาต้าเฟรม (Data Frame)



รูปที่ 2.3 แสดงเฟรมของข้อมูลซึ่งประกอบด้วยบิตเริ่มต้น 1 บิตข้อมูล 8 บิต บิตพาริตี้และบิตปิดท้าย 1 บิต

ในรูปที่ 2.2 แสดงตัวอย่างของดาต้าเฟรม ซึ่งประกอบด้วย 4 ส่วนคือ Start bit, Data bit, Parity bit และ Stop bit ในสภาวะที่ไม่มีข้อมูลส่งเข้ามาสัญญาณจุดนี้จะมีสถานะลอจิกเป็น "1" เสมอ และจะมีการส่งข้อมูลครั้งแรกจะต้องส่ง Start bit เป็นลอจิก "0" ไปก่อนซึ่งทำให้ด้านรับสามารถรับรู้การส่งข้อมูลได้

ต่อมาคือ บิตข้อมูล(Data bit) ซึ่งจะมีขนาดตั้งแต่ 5-8 บิตโดยทั่วไปจะมีขนาด 7 หรือ 8 บิต โดยในการใช้โมเด็มแต่ละครั้งจะต้องตั้งจำนวนบิตให้เป็นมาตรฐานเดียวกัน

พาริตี้บิต (Parity bit) จะเป็นส่วนที่เพิ่มเข้ามาเพื่อช่วยตรวจสอบความผิดพลาดที่เกิดขึ้น ซึ่งตามปกติพาริตี้บิตจะถูกกำหนดเป็น "none" หรือลอจิก "0" โดยค่าพาริตี้จะเป็นคู่เมื่อจำนวนของบิตในการส่งข้อมูลมาเป็นเลขคู่และจะเป็นคี่เมื่อจำนวนบิต "1" ในข้อมูลเป็นเลขคี่ โมเด็มภาครับจะคำนวณจำนวนค่าข้อมูลและนำไปเปรียบเทียบกับพาริตี้บิต ถ้าได้จำนวนไม่เท่ากันจะแสดงค่าผิดพลาดออกมา

ส่วนสุดท้ายคือบิตปิดท้าย (Stop bit) ซึ่งจะมีขนาดเป็น 1 หรือ2 บิต แต่ส่วนมากนิยมใช้เป็น 1 บิต เพราะว่า 2 บิตนั้นมากเกินความจำเป็น ตามปกติบิตปิดท้ายจะกำหนดให้อยู่ในสภาวะลอจิก "1" ซึ่งจะค้างไว้จนกว่าจะมีการส่งข้อมูลครั้งต่อไป โดยปกติในโมเด็มจะตั้งจำนวนของบิตปิดท้ายให้เท่ากัน

เมื่อต้องการที่จะเชื่อมโยงคอมพิวเตอร์เข้ากับศูนย์บริการ BBS (Bulletin Board System) หรือระบบกระดานข่าว หรือระบบอื่นๆ จะต้องทำการตั้งค่ามาตรฐานการส่งข้อมูลให้เป็นมาตรฐานเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

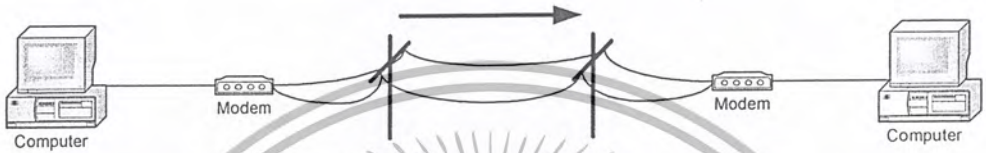
ก่อน โดยทั่วไปที่นิยมใช้จะเป็นการกำหนดดาต้าเฟรมเป็นแบบดาต้าบิต 8 บิต, ไม่มีพาริตี้บิต และมีบิตปิดท้าย 1 บิต หรือเรียกว่า "8, N, 1"

ส่วนแบบซิงโครนัสก็ยังมีการใช้งานอยู่ โดยส่วนมากจะใช้ในการสื่อสารกับเมนเฟรมคอมพิวเตอร์

ลักษณะการทำงานของโมเด็ม สามารถแบ่งได้ 2 แบบ คือ

1.แบบที่สามารถรับส่งข้อมูลได้ในเวลาเดียวกัน (Full Duplex)

2.แบบที่รับหรือส่งข้อมูลได้เพียงอย่างเดียวในขณะใดขณะหนึ่งเท่านั้น (Half Duplex)



การสื่อสารแบบ Half Duplex

การสื่อสารแบบ Full Duplex

รูปที่ 2.4 แสดงลักษณะรูปแบบการรับส่งข้อมูลของโมเด็ม

การเลือกใช้โมเด็มให้เหมาะกับวงจรข้อมูลแต่ละแบบให้พิจารณาจาก

- อัตราบิต
- ค่าใช้จ่าย

ถ้าอัตราบิตมีสูงขึ้นทำให้ราคาของโมเด็มสูงตามไป เหตุอื่นที่ทำให้โมเด็มมีอัตาเร็วสูง(ราคาสูง) เนื่องจากมีฟังก์ชั่นพิเศษดังนี้

- ลดขนาดของข้อมูลให้มีช่วงแบนด์แคบลง
- ลดผลกระทบจากสัญญาณรบกวนในสายส่ง, การลดทอนข้อมูลและการลดเสียงสะท้อนให้น้อยที่สุด
- ลดการรบกวนข้ามกันระหว่างคู่สายโทรศัพท์ที่อยู่ติดกัน

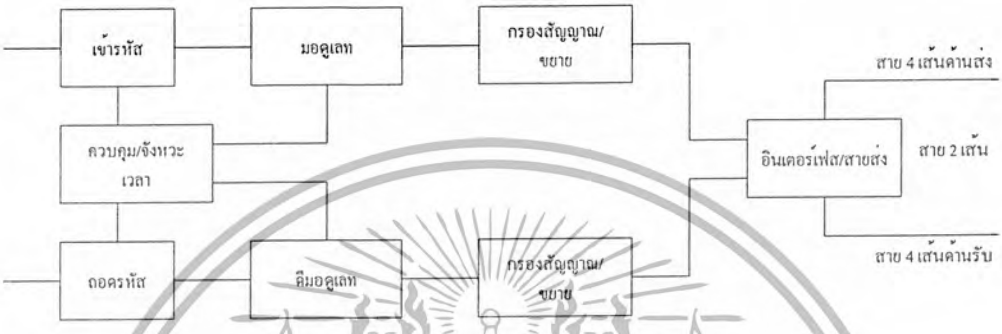
เงื่อนไขอื่นที่ยังใช้พิจารณาคือ ระยะทางระหว่างโมเด็มทั้งสองเครื่อง โดยกรณีที่มีความยาวของโทรศัพท์สั้น (ประมาณไว้ว่าไม่เกิน 25 กิโลเมตร) เราใช้อุปกรณ์ประเภทช็อตฮอลโมเด็ม (Short-haul Modem), ไลน์ไดร์เวอร์ หรือโมเด็มอีลิมิเนเตอร์ (Modem eliminator) ซึ่งไม่มีข้อกำหนดของ CCITT ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวถึงอุปกรณ์เหล่านี้ อย่างไรก็ตามก็มีการใช้งานกันเนื่องจากมีราคาถูกกว่าโมเด็มความเร็วสูงที่ส่งออกแบบให้ทำงานเฉพาะวงจรระยะไกลเท่านั้น

โมเด็มที่ทำงานที่อัตราบิตสูง เช่นที่ 9,600 บิต/วินาที หรือมากกว่านั้นอาจมีพอร์ตอินพุทดิจิตอลมากกว่าหนึ่งพอร์ต และบางชนิดได้สร้างมัลติเพล็กเซอร์แบบ TDM รวมเข้ากับเครื่องเพื่อรวมข้อมูลอินพุทจากพอร์ตดิจิตอลต่างพอร์ตเข้าด้วยกัน ก่อนที่จะส่งในรูปสัญญาณความถี่เสียงไปบนสายโทรศัพท์

ไดอะแกรมของโมเด็ม



รูปที่ 2.5 แสดงไดอะแกรมพื้นฐานของโมเด็ม

หมายเหตุ ภาคถอดรหัสไม่มีในโมเด็มชนิด FSK

จากรูปที่ 2.5 สัญญาณดิจิตอลอินพุทจากคอมพิวเตอร์หรือเทอร์มินอล ส่งเข้าส่วนเข้ารหัสซึ่งบิตข้อมูล ถูกจัดการให้เป็นข้อมูลแบบ 2 บิต, 3 บิต หรือ 4 บิต จากนั้นส่งต่อไปยังภาคมอดูเลท เพื่อสร้างสัญญาณในย่านความถี่เสียงขึ้นมาและกำจัดความถี่บางส่วนที่ไม่ต้องการออกไปด้วยวงจรกรองสัญญาณ และนำผลที่ได้มาขยายสัญญาณใหม่ เพื่อเตรียมส่งให้ภาคอินเตอร์เฟสของสายส่ง เมื่อถึงจุดนี้สัญญาณความถี่เสียงถูกเลือกให้ส่งในสายโทรศัพท์แบบใดแบบหนึ่งซึ่งมี 4 เส้นและ 2 เส้น

ในภาครับ สัญญาณจะถูกกรองเอาส่วนความถี่ที่ไม่ต้องการออกไป แล้วจึงส่งต่อให้ภาคดีมอดูเลทและภาคถอดรหัสสัญญาณให้อยู่ในรูปเดิม

2.4 การแบ่งชนิดของโมเด็ม เราสามารถแบ่งโมเด็มได้ 2 แบบ คือ

1. โมเด็มแบบติดตั้งภายนอก (External Modem) จะมีลักษณะเป็นกล่องต่ออยู่ภายนอกพีซี โดยจะมีคอนเน็กเตอร์ RS-232C ซึ่งอาจจะอยู่ติดกับตัวของโมเด็มเองหรืออาจจะมีสายสัญญาณพร้อมคอนเน็กเตอร์ที่สามารถใช้เสียบลงบนคอนเน็กเตอร์ RS-232C อีกด้านหนึ่งที่อยู่บนเครื่องพีซี และโมเด็มประเภทนี้มักจะต้องใช้ไฟฟ้าจากภายนอกพีซี ดังนั้นจึงจำเป็นที่จะต้องจัดหาเต้าเสียบปลั๊กไฟฟ้าตัวเมียเพิ่มเติมสำหรับจ่ายไฟฟ้าให้แก่โมเด็มด้วย

ข้อดีของโมเด็มแบบติดตั้งภายนอกประการหนึ่งคือ ในกรณีที่มีเครื่องพีซีหลายเครื่องวางอยู่ใกล้ๆ กัน จะสามารถใช้งานโมเด็มร่วมกันโดยใช้อุปกรณ์ตัดต่อสัญญาณ RS-232C ทำให้สามารถประหยัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าใช้จ่าย ไม่จำเป็นที่จะต้องซื้อโมเด็มหลายๆตัวเพื่อใช้ประจำกับพีซีทุกเครื่อง และทั้งยังเป็นการใช้สายโทรศัพท์ร่วมกันอีกด้วย ส่วนในกรณีที่พีซีต่ออยู่กับระบบ LAN ก็สามารถกำหนดให้ Communication Server ของ LAN จัดสรรการใช้โมเด็มร่วมกันได้

ราคาของโมเด็มแบบติดตั้งภายนอกมักจะสูงกว่าโมเด็มแบบติดตั้งภายในเล็กน้อย เพราะว่าโมเด็มแบบติดตั้งภายนอกจำเป็นที่จะต้องมีระบบจ่ายกำลังไฟฟ้า (Power Supply) เป็นของตัวเองรวมทั้งต้องมีตัวถังเพิ่มเข้ามาซึ่งไม่จำเป็นสำหรับโมเด็มแบบติดตั้งภายใน และสิ่งที่จำเป็นที่ต้องคำนึงถึงอีกประการหนึ่งคือ โมเด็มแบบติดตั้งภายนอกจะต้องมีสายนำสัญญาณและคอนเน็กเตอร์แบบ RS-232C เพื่อใช้สื่อสารกับเครื่องพีซี ซึ่งจะต่างกับโมเด็มแบบติดตั้งภายในที่ตัวการ์ดสามารถเสียบลงบนช่องเสียบภายในพีซีและรับส่งข้อมูลกับเครื่องพีซีได้โดยตรง

สำหรับผู้ผลิตโมเด็มบางรายอาจจะออกแบบให้โมเด็มแบบติดตั้งภายนอกมีลักษณะเป็นการ์ดที่มีตัวถังหุ้มอยู่ ซึ่งสามารถถอดเอาแผ่นวงจรของโมเด็มออกมาจากตัวถังนี้ได้ จุดประสงค์ของการออกแบบการใช้งานเช่นนี้ก็เพื่อสามารถนำโมเด็มมาติดตั้งอยู่ภายในแร็ค (Rack) (โครงเหล็กที่เป็นตัวถัง ใช้บรรจุโมเด็มได้ทีละหลายๆ ตัว) และบนแร็คจะมีชุดจ่ายกำลังไฟฟ้าที่คอยเลี้ยงโมเด็มทั้งหมดภายใน ลักษณะการใช้งานเช่นนี้ มักจะพบในระบบใหญ่ๆ ที่ต้องการสื่อสารข้อมูลโดยผ่านโมเด็มออกไปได้หลายๆ ทาง อย่างเช่นระบบ PABX ขนาดใหญ่ หรือระบบแพกเก็ทสวิตชิ่ง (Packet Switching) เป็นต้น นอกจากนี้ผู้ผลิตบางรายก็อาจจะผลิตโมเด็มรุ่นพิเศษที่ทำให้สามารถกำหนดค่าการเซตต่างๆได้จากหน้าปัทม์ แทนที่จะใช้ชุดคำสั่ง AT จึงทำให้งานมีความคล่องตัวมากยิ่งขึ้น

รูปที่ 2.6 แสดงโมเด็มแบบต่อภายนอก

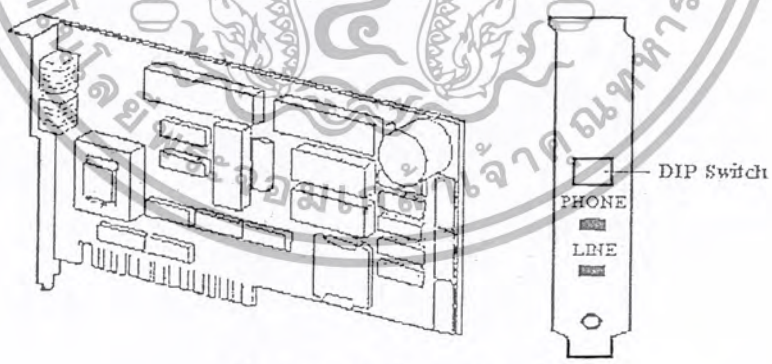
2. โมเด็มแบบติดตั้งภายใน (Internal Modem) โมเด็มชนิดนี้จะมีลักษณะเป็นการ์ด ซึ่งนำมาเสียบลงในช่องเสียบ (slot) ของเมนบอร์ดของคอมพิวเตอร์ เนื่องจากการติดตั้งการ์ดไว้ภายใน เครื่องคอมพิวเตอร์จึงทำให้ไม่ต้องมีแหล่งจ่ายไฟแยกต่างหาก รวมถึงตัดปัญหาเกี่ยวกับสายนำสัญญาณและคอนเน็กเตอร์ RS-232C แต่ก็ทำให้เสียช่องเสียบในเครื่องพีซีไป 1 ช่อง ซึ่งถ้าในกรณีที่เครื่องพีซีที่ใช้นั้นมีช่องเสียบน้อย เช่น 3 หรือ 4 ช่องจึงควรเลี่ยงไปใช้โมเด็มแบบติดตั้งภายนอก นอกจากนี้โมเด็มแบบติดตั้งภายในที่ติดอยู่ภายในเครื่องจะใช้พลังงานไฟฟ้าจากเครื่องคอมพิวเตอร์และเป็นต้นเหตุหนึ่งที่ทำให้อุณภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูงขึ้น ซึ่งถ้าเป็นพีซีรุ่นที่ค่อนข้างเก่าแล้ว ระบบจ่ายไฟฟ้าของมันนั้นจะสามารถจ่ายกำลังไฟฟ้าได้น้อย และยังมีระบบระบายความร้อนที่ไม่เพียงพอ ก็อาจก่อให้เกิดปัญหาอื่นๆตามมาในภายหลังได้

โมเด็มแบบติดตั้งภายในจะมีพอร์ตสื่อสารอนุกรม หรือพอร์ต RS-232C เพื่อใช้ติดต่อสื่อสารกับเครื่องพีซีติดตั้งมาภายในการ์ดแล้ว เราเพียงกำหนดให้วงจรพอร์ตสื่อสารแบบอนุกรมบนการ์ดโมเด็มเป็น COM1 หรือ COM2 เท่านั้น (บางกรณีอาจจะกำหนดให้เป็น COM3 หรือ COM4 ไปจนถึง COM8 ก็ได้) ถ้าเครื่องพีซีที่ใช้อยู่นั้นมีพอร์ตสื่อสารแบบอนุกรม COM1 และ COM2 มาให้เรียบร้อยแล้ว ก็ควรจะเลือกใช้โมเด็มแบบติดตั้งภายในที่สามารถกำหนดพอร์ตสื่อสารอนุกรมให้เป็น COM อื่นๆ ได้ เช่น COM3 หรือ COM4 เป็นต้น เพื่อป้องกันไม่ให้การอ้างพอร์ตบนพีซีเกิดการซ้ำซ้อนกัน

เนื่องจากโมเด็มแบบติดตั้งภายในจะต้องถูกติดตั้งลงในช่องเสียบภายในเครื่องพีซี ดังนั้นจะเป็นการไม่สะดวกนักถ้าหากต้องการเคลื่อนย้ายโมเด็มออกไปใช้งานกับเครื่องพีซีอื่นๆ รวมทั้งไม่สะดวกในการปรับตั้งสวิตซ์ต่างๆ บนตัวโมเด็ม แต่อย่างไรก็ตามโมเด็มประเภทนี้ก็มีข้อดีคือ สามารถตัดปัญหาเรื่องของสายนำสัญญาณและคอนเน็กเตอร์ RS-232C ออกไปได้ ซึ่งคนส่วนใหญ่มักจะประสบปัญหาเรื่องของการติดตั้งมาก โดยเฉพาะปัญหาเรื่องการแปลงคอนเน็กเตอร์ที่เป็นแบบตัวผู้ ตัวเมีย รวมทั้งคอนเน็กเตอร์แบบ 9 ขาหรือ 25 ขาอีกด้วย นอกจากนี้โมเด็มแบบติดตั้งภายในยังมีข้อดีอีกประการหนึ่งคือ ค่อนข้างจะปลอดภัย ไม่สามารถถอดหรือเคลื่อนย้ายออกจากเครื่องพีซีได้ง่ายนัก ส่วนข้อเสียที่สำคัญของโมเด็มประเภทนี้ก็คือ บนตัวโมเด็มมักจะไม่มีหลอดไฟ LED แสดงสถานะเหมือนกับโมเด็มแบบติดตั้งภายนอก ถึงแม้ว่าบางยี่ห้ออาจจะมีหลอดไฟดังกล่าวแต่ก็จะอยู่บนตัวการ์ดซึ่งติดตั้งอยู่ภายในพีซี ทำให้ไม่สะดวกในการมอง ซึ่งสัญญาณไฟแสดงสถานะเหล่านี้จะมีประโยชน์มากในการแก้ไขปัญหาของโมเด็มอยู่เป็นประจำ เช่น สัญญาณ CD (Carrier Detect), Off-hook, RD (Receive Data) หรือ TD (Transmitted Data) จะเป็นสิ่งที่สำคัญและอำนวยความสะดวกในการค้นหาสาเหตุ แต่อย่างไรก็ตามเมื่อเปรียบเทียบรุ่นที่มีความสามารถเท่ากันแล้ว โมเด็มแบบติดตั้งภายในมักจะมีราคาที่ต่ำกว่าโมเด็มแบบติดตั้งภายนอกเล็กน้อย



รูปที่ 2.7 แสดงโมเด็มแบบต่อภายใน

หน้าที่ของโมเด็ม

โมเด็มมีหน้าที่อยู่หลายประการด้วยกันดังต่อไปนี้

1.จัดการขั้นตอนเริ่มต้นในการติดต่อ (Handshaking)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.จัดการการส่งและรับข้อมูล

3.แปลงสัญญาณดิจิตอลให้อยู่ในรูปที่เหมาะสมในการส่งผ่านสายโทรศัพท์ และแปลงกลับมารูปเดิมเมื่อการส่งสิ้นสุดลง

4.จัดการการเรียกเองได้โดยส่งสัญญาณหมุนหมายเลขออกไป และติดตามการตอบรับจากปลายทาง

5.สามารถตอบรับการเรียกเข้ามาของอุปกรณ์อื่นได้อย่างอัตโนมัติ

6.ตรวจสอบและแก้ไขข้อผิดพลาดของข้อมูล

7.ตรวจหากรณีที่การส่งข้อมูลล้มเหลวขึ้นมาและวิเคราะห์หาข้อผิดพลาดรวมทั้งจัดการแก้ไขข้อมูลด้วย

2.5 การสื่อสารข้อมูลผ่านโมเด็ม

เมื่อผู้ใช้สั่งให้โมเด็มเริ่มทำการโทรศัพท์ไปยังปลายทาง กระบวนการต่างๆ ระหว่างโปรแกรมสื่อสารและโมเด็มก็จะเริ่มขึ้นถ้าหากผู้ใช้เคยใช้งานมาแล้วก็จะคุ้นเคยกับเสียงต่างๆ ที่ดังออกมาจากลำโพงของโมเด็ม เช่นเสียง Dial Tone ตามด้วยเสียง DTMF หรือเสียง Pulse ในขณะที่ทำการหมุนเลขหมายโทรศัพท์ปลายทาง เสียง Ringing Tone และเสียงสัญญาณพาหะโต้ตอบกันระหว่างโมเด็มต้นทางและปลายทาง หลังจากนั้นโมเด็มก็จะเงียบลงพร้อมกับเข้าสู่การเชื่อมต่อ (Connection) ซึ่งกระบวนการดังกล่าวนี้นับได้ว่าเป็นกระบวนการทำงานปกติที่ ผู้ใช้คุ้นเคยเป็นอย่างดี

ขั้นตอน	ผู้ใช้	ซอฟต์แวร์	โมเด็มต้นทาง	โมเด็มปลายทาง
1	เลือกคำสั่ง "Dial" จากซอฟต์แวร์	เปิดสัญญาณ DRT เพื่อส่งคำสั่งหมุนหมายเลขไปยังโมเด็ม โดยใช้คำสั่ง:ATDT	เปิดลำโพง ยกหู โทรศัพท์ รอสัญญาณหมุนหมายเลขโทรศัพท์	
2		รอ result codes จากโมเด็ม	รอการตอบรับจากปลายทาง ทั้งนี้ระยะเวลาการรอขึ้นกับการกำหนดค่ารีจิสเตอร์	
3				เสียงโทรศัพท์ดัง
4				ตอบรับสัญญาณ
5			รับสัญญาณตอบรับและทำการส่งสัญญาณ Originate Carrier	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6			รับทราบวิธีการมอดูเลทชั่นและความเร็วของแต่ละฝ่าย	รับทราบวิธีการมอดูเลชั่นและความเร็วของแต่ละฝ่าย
7			โมเด็มตกลงรับรู้โปรโตคอลการควบคุมความผิดพลาด และการบีบอัดข้อมูลของแต่ละฝ่าย	
8			ส่ง result code "Connect" ไปยัง PC ปิดลำโพงและเปิดสัญญาณ CD	
9		รับรู้ result code และสัญญาณ CD รายงานให้ผู้ใช้ทราบว่าการติดต่อได้เกิดขึ้นแล้ว		
10	เริ่มการติดต่อกับโฮสคอมพิวเตอร์	ดำเนินการสื่อสารและคอยดูสัญญาณที่ขาดหายไปจากหน้าจอสัญญาณ CD	ส่งและรับข้อมูล	ส่งและรับข้อมูล
11	การสื่อสารเสร็จสมบูรณ์เลือกคำสั่ง"Disconnect"	ปิดสัญญาณ DRT หรือส่ง+++ ตามด้วยคำสั่ง ATH		
12			วางสายโทรศัพท์	ยกเลิกสัญญาณวางสายโทรศัพท์

ตารางที่ 2.2 แสดงกระบวนการทำงานของโมเด็มต้นทางและปลายทางตั้งแต่ขั้นตอน 1-12

จากตารางที่ 2.2 จะแสดงผังลำดับขั้นตอนการทำงาน โดยเริ่มตั้งแต่ขั้นตอนที่สั่งให้โมเด็มหมุนหมายเลขโทรศัพท์จนกระทั่งวางหูโทรศัพท์ และจากตารางที่ 2.2 นี้เอง ผู้ใช้สามารถสังเกตได้ว่าสิ่งที่ทำงานมากที่สุดในกระบวนการสื่อสารก็คือโมเด็ม ส่วนโปรแกรมสื่อสารจะมีหน้าที่เพียงส่งชุดคำสั่ง AT ไปให้กับโมเด็มเท่านั้น การกำหนดเวลาต่างๆ ในกระบวนการนี้จะขึ้นอยู่กับข้อมูลที่อยู่ในรีจิสเตอร์ S ด้วย ตัวอย่างเช่น รีจิสเตอร์ S7 จะเก็บค่าของเวลาที่โมเด็มจะรอเสียงตอบจากโมเด็มปลายทาง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการเชื่อมต่อระหว่างโมเด็มต้นทางและปลายทางที่ได้กล่าวไปแล้วนั้น โมเด็มทั้งคู่จำเป็นที่จะต้องมีพื้นฐานที่เหมือนกันบางประการ เช่น ความเร็ว รูปแบบของข้อมูล และโปรโตคอลถ่ายโอนไฟล์ในการรับส่งข้อมูล

2.5.1 โปรโตคอลถ่ายโอนไฟล์

โปรโตคอลถ่ายโอนไฟล์จะแบ่งไฟล์ที่จะถูกส่งออกเป็นบล็อกหรือแพคเกต แต่ละแพคเกตประกอบด้วยส่วนหัว ส่วนท้ายข้อมูล รหัสตรวจสอบข้อผิดพลาดและเครื่องหมายจบแพคเกต คอมพิวเตอร์เป้าหมายต้องส่งการตอบสนองเพื่อบอกว่าได้รับแพคเกตอย่างถูกต้องหรือไม่ โปรโตคอลที่ใช้โดยทั่วไปมี 3 ตัวคือ XMODEM, YMODEM, ZMODEM เป็นโปรโตคอลที่พัฒนามาจาก XMODEM

2.5.1.1 XMODEM

XMODEM เป็นโปรโตคอลถ่ายโอนไฟล์ที่เรียบง่ายมาก แม้ว่ามันจะมีข้อจำกัด แต่ก็เป็นโปรโตคอลที่ใช้กันกว้างขวางที่สุด ข้อมูลที่ถูกส่งโดย XMODEM จะถูกแบ่งออกเป็นบล็อค แต่ละบล็อคประกอบด้วยอักขระ Start of Header (01 ฐานสิบหก) หมายเลขบล็อคหนึ่งไบต์ คอมพลีเมนต์ของหนึ่งของหมายเลขบล็อคข้อมูล 128 ไบต์และ Checksum หนึ่งไบต์ ดังในตารางที่ 2.2

ออฟเซต	ความหมาย
0	SOH (Start-of-header, ASCII 01)
1	หมายเลขบล็อค เริ่มต้นจาก 1 แต่จะกลับเป็น 0 หลังจาก FF
2	คอมพลีเมนต์ของหนึ่งของหมายเลขบล็อค (255-หมายเลขบล็อค)
3-130	ข้อมูล 128 ไบต์
131	Checksum ผลรวมของข้อมูลเท่านั้น

ตารางที่ 2.3 แสดงบล็อคของ XMODEM

ก่อนที่คอมพิวเตอร์ฝ่ายส่งจะสามารถส่งข้อมูลได้ มันต้องรับอักขระ NAK (Negative Acknowledgment) จากคอมพิวเตอร์ฝ่ายรับ โปรแกรมผู้รับจะส่ง NAK (15 ฐานสิบหก) หลังจากไทม์เอาท์ทุกๆ 10 วินาทีที่ไม่ได้รับข้อมูล NAK ตัวแรกกระตุ้นให้ผู้ส่งเริ่มทำงานส่ง เมื่อโปรแกรมผู้รับเริ่มทำการรับบล็อคมันจะรายงานข้อผิดพลาดเมื่อมีช่องว่าง 1 วินาทีหรือมากกว่า เกิดขึ้นระหว่างตัวอักษรในบล็อครวมทั้ง Checksum อย่างไรก็ตามมันต้องรอให้สายว่างก่อนที่จะส่ง NAK เพื่อแจ้งข้อผิดพลาดไทม์เอาท์ 1 วินาที ไม่เพียงพอสำหรับ การสื่อสารทางไกล จึงมักใช้เวลาคอยที่นานกว่านี้ จากนั้นผู้รับจะตรวจสอบหมายเลขบล็อคและรายงานข้อผิดพลาดถ้ามันไม่เรียงลำดับ ถ้าหมายเลขบล็อคเหมือนกับบล็อคล่าสุดแสดงถึงการส่งซ้ำซึ่งไม่ควรพิจารณาเป็นข้อผิดพลาด หลังจากการรับส่งแต่บล็อค ผู้รับจะส่ง ACK (06 ฐานสิบหก) ถ้าบล็อคที่รับมาถูกต้อง หรือ NAK ถ้าไม่ถูกต้อง ในกรณีหลังผู้ส่งจะส่งบล็อคนั้นซ้ำ หลังจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การตอบรับบล็อคนั้นแล้ว บล็อคต่อไปจึงจะถูกส่ง ในตอนจบของการส่งผู้ส่งจะส่งEOT(04 ฐานสิบหก) และรอคอย ACK มันจะส่งEOT ซ้ำถ้าไม่ได้รับ ACK

เนื่องจากจาก Checksum หนึ่งไบต์ไม่เพียงพอสำหรับตรวจสอบข้อผิดพลาดทั้งหมด ส่วนขยายของ XMODEM ที่เรียกว่า ตัวเลือก CRC จึงถูกคิดขึ้นโดยใช้ตัวเลขสองไบต์ ตัวเลขนี้เรียกว่า cyclical redundancy check (CRC-16) ซึ่งสามารถตรวจสอบข้อผิดพลาดได้อย่างน้อย 99.99 %โดยปกติผู้รับต้องบอกผู้ส่งว่าใช้ตัวเลือก CRC โดยการส่งอักขระ C แทน NAK ถ้าไม่มีการตอบสนองหลังจากการส่ง C รูปแบบของบล็อคที่ใช้ตัวเลือก CRC ถูกแสดงไว้ในตารางที่ 2.4

ออฟเซต	ความหมาย
0	SOH (Start-of-Leader, ASCII 01)
1	หมายเลขบล็อค เริ่มต้นจาก 1 แต่จะกลับเป็น 0 หลังจาก FF
2	คอมพลีเมนต์ของหนึ่งของหมายเลขบล็อค(255-หมายเลขบล็อค)
3-130	ข้อมูล 128 ไบต์
132	ไบต์บนของ CRC
133	ไบต์ล่างของ CRC

ตารางที่ 2.4 แสดงรูปแบบบล็อคของ XMODEM ที่ใช้ตัวเลือก CRC

2.5.1.2 YMODEM

YMODEM คือ XMODEM ที่มีการขยายความสามารถบางอย่าง ดังนี้

1.การตรวจสอบข้อผิดพลาดด้วย CRC-16

2.มีบล็อคขนาด 1K เป็นทางเลือก การส่ง STX (02 ฐานสิบหก) ที่จุดเริ่มต้นของแต่ละบล็อค แทน SOH (01 ฐานสิบหก) เป็นสัญลักษณ์ว่าบล็อคที่ตามมายาว 1024 ไบต์ แทน 128 ไบต์ บล็อคขนาด 1024 และ 128 ไบต์ สามารถถูกส่งผสมกันไปได้ในการส่งครั้งหนึ่ง

3.การยกเลิกด้วย CAN-CAN อักขระ CAN สองตัวติดกันบอกว่าการถ่ายโอนไฟล์ต้องถูกยกเลิก

4.การส่งชื่อไฟล์ บล็อคแรกมีหมายเลขศูนย์ประกอบด้วยชื่อไฟล์ปิดท้ายด้วย ASCII 0 ชื่อไฟล์ควรถูกแปลงเป็นตัวอักษรตัวเล็ก ถ้าคอมพิวเตอร์ฝ่ายส่งไม่สนับสนุนทั้งชื่อไฟล์ตัวเล็กและตัวใหญ่ ชื่อไฟล์สามารถรวมถึงชื่อเส้นทาง แยกโดย Slash (/) ซึ่งเป็นวิธีปกติในยูนิกซ์ ไม่ใช่ Backslash (\) ดังที่ใช้ในดอส อย่างไรก็ตามโดยวิธีปกติชื่อไฟล์แบบเต็มจะถูกส่งไป เนื่องจากคอมพิวเตอร์ฝ่ายรับอาจมีโครงสร้างไดเรกทอรี่ต่างกัน

5.การส่งไฟล์แบบแบตช์ (Batch-file transmission) ไฟล์หลายไฟล์สามารถถูกส่งในคราวเดียว ที่จุดสิ้นสุดของแต่ละไฟล์ คอมพิวเตอร์ฝ่ายส่งจะส่ง EOT หนึ่งถึงสิบตัว (ASCII 4 หรือ Ctrl-D) จนกระทั่งได้รับ ACK จากนั้นจึงสามารถส่งไฟล์อื่น หรือจบการส่ง โดยการส่งแพคเกตที่ชื่อไฟล์เป็น NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
54988

ส่วนที่เหลือของบล็อคถูกทำให้เป็น NULL ไฟล์ต่อไปถูกส่งโดยมีบล็อคชื่อไฟล์ของตัวเอง ชื่อไฟล์ที่เป็น NULL หมายถึงจบการส่งแบบแบตช์ มีข้อสังเกต คือ โฮสต์คอมพิวเตอร์และเครือข่ายจำนวนมากไม่สามารถจัดการกับบล็อคขนาด 1K ที่ต่อเนื่องกันได้

2.5.1.3 YMODEM-g

โปรโตคอล YMODEM-g คล้ายกับ YMODEM โดยมีความแตกต่างดังนี้

1. คอมพิวเตอร์ฝ่ายรับส่ง G เพื่อขอร้องให้เริ่มต้นทำการส่งแทนการส่ง C

2. คอมพิวเตอร์ฝ่ายรับไม่ส่ง ACK สำหรับแพคเกตที่ถูกต้อง ถ้ามันตรวจพบข้อผิดพลาด มันจะยกเลิกการถ่ายโอนไฟล์ โดยการส่งอักขระ CAN ติดต่อกัน

3. คอมพิวเตอร์ฝ่ายส่งไม่รอให้บล็อคหนึ่งถูกตอบรับ ก่อนจะส่งบล็อคต่อไป

2.5.1.4 ZMODEM

ZMODEM มีการปรับปรุงเพิ่มขึ้นจาก XMODEM และ YMODEM โดยมีข้อได้เปรียบหลักคือสามารถถ่ายโอนข้อมูลแปดบิตบนช่องแบบเจ็ดบิตได้ โดยใช้การเข้ารหัสอักขระควบคุมและอักขระพิเศษตัวอื่น มันสามารถถ่ายโอนไฟล์แบบอัตโนมัติหากซอฟแวร์สื่อสารรู้จัก ZMOEDM ตัวอย่างเช่น โปรแกรม ProComm Plus รู้จักการเริ่มต้นการถ่ายโอนไฟล์ด้วย ZMODEM ถ้าร้องขอให้โฮสต์ส่งไฟล์ด้วย ZMODEM ProComm Plus จะเริ่มต้นการดาวน์โหลดโดยอัตโนมัติ เมื่อมันได้รับลำดับที่เริ่มต้นการส่งของ ZMODEM

โดยในการเชื่อมต่อโมเด็มในแต่ละครั้ง อย่างน้อยที่สุด ผู้ใช้โมเด็มต้นทางจำเป็นต้องทราบว่าจะให้โมเด็มโทรศัพท์ออกด้วยวิธีใด ควรจะตั้งเวลาในการรอเสียงตอบจากโมเด็มปลายทางนานเท่าไร ระบบโทรศัพท์ของตนนั้นใช้ระบบ TONE หรือ Pulse และในขณะเดียวกันผู้ที่ใช้โมเด็มปลายทางก็จะต้องกำหนดให้โมเด็มรับทราบด้วยว่าจะให้โมเด็มรับสายโทรเข้าหลังจากมีเสียง Ringing กี่ครั้ง และจะต้องรอสัญญาณโต้ตอบนานเท่าไร เป็นต้น ซึ่งการดำเนินการทั้งหมดนี้สามารถควบคุมได้โดยชุดคำสั่ง AT

2.5.2 ประเภทของชุดคำสั่ง AT

โดยพื้นฐานแล้ว เราสามารถจะแบ่งชุดคำสั่ง AT ออกเป็นสองประเภทใหญ่ๆ ได้ดังนี้

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

2.ชุดคำสั่งที่ให้กำหนดค่า หรือเปลี่ยนแปลงค่าต่างๆ อย่างเช่น ATS=90 เป็นการกำหนดให้ค่ารีจิสเตอร์ S มีค่า 90 ซึ่งก็คือให้โมเด็มรอการตอบรับจากโมเด็มปลายทางเป็นเวลา 90 วินาที หลังจากนั้นหากยังไม่มีสัญญาณใดๆ ตอบกลับมาโมเด็มก็จะวางสายทันที หลังจากนั้นหากยังไม่มีสัญญาณใดๆ ตอบกลับมาโมเด็มก็จะวางสายทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามปกติแล้วโปรแกรมสื่อสารมักจะตั้งค่าของค่ารีจีสเตอร์ S ต่างๆ ที่จำเป็นเอาไว้ตั้งแต่ตอนที่ผู้ใช้เรียกโปรแกรมนั้นๆขึ้นมา ซึ่งผู้ใช้จะสามารถควบคุมและเปลี่ยนแปลงค่าของรีจีสเตอร์ S เหล่านี้ได้ในภายหลัง

2.5.3 การออนไลน์และออฟไลน์

สภาวะออฟไลน์ จะสามารถเรียกได้อีกอย่างหนึ่งว่าสภาวะคำสั่ง (Command State) หมายถึงสภาวะที่ผู้ใช้สามารถจะส่งคำสั่งต่างๆ ไปยังโมเด็มได้ หรือ จะพูดอีกนัยหนึ่งก็คือ สภาวะที่โมเด็มจะแปลความหมายของข้อมูลที่ได้รับมาจาก PC ให้เป็นคำสั่งเท่านั้น ซึ่งสภาวะนี้จะไม่ได้รับ – ส่งข้อมูลกับโมเด็มปลายทาง แต่จะสื่อสารกับ PC เท่านั้น

สภาวะออนไลน์ หมายถึง สภาวะที่โมเด็มได้เชื่อมต่อกับโมเด็มปลายทางเป็นที่เรียบร้อยแล้ว ข้อมูลที่ส่งออกมาจาก PC ก็จะผ่านจากโมเด็มต้นทางไปยังโมเด็มปลายทางเสมอ

ถ้าหากผู้ใช้ต้องการที่จะส่งคำสั่งให้กับโมเด็ม หรือต้องการจะให้โมเด็มกลับมาอยู่ในสภาวะออฟไลน์ เพื่อรับคำสั่งจาก PC ก็จะสามารถทำได้โดยมีวิธีการอยู่ 2 วิธีคือ วิธีที่หนึ่ง ให้โมเด็มวางสายแล้วกลับมาอยู่ในสภาวะออนไลน์ใหม่ และวิธีที่สองคือ ส่งชุดอักขระ Escape Sequence เข้าไปยังโมเด็มในขณะที่ออนไลน์ ซึ่งวิธีหลังนี้จะมีความเหมาะสมกว่า เพราะการเชื่อมต่อระหว่างโมเด็มจะยังคงดำเนินไปอยู่ และหลังจากที่ได้ส่งคำสั่งต่างๆ ให้กับโมเด็มเป็นที่เรียบร้อยแล้ว ก็จะสามารถกลับเข้าไปอยู่ในสภาวะออนไลน์ได้เช่นเดิม วิธีการส่งชุดอักขระ Escape Sequence สำหรับโมเด็มที่เข้ากันได้กับโมเด็มของ Hayes คือให้รอ 1 วินาที (เรียกว่า Guard Time) แล้วกดปุ่ม + ติดต่อกัน 3 ครั้ง (+++) การที่ต้องรอ 1 วินาทีก่อนที่จะกดปุ่มเครื่องหมาย + เป็นสิ่งที่จำเป็นเนื่องจากว่าโมเด็มจะได้รู้ว่าอักษร + นั้นเป็นส่วนของอักขระ Escape Sequence ไม่ใช่ส่วนของข้อมูลที่จะส่งไปยังปลายทาง

2.5.4 รูปแบบของชุดคำสั่ง AT

การใช้ชุดคำสั่ง AT จำเป็นจะต้องขึ้นต้นด้วยอักษร AT เสมอ และจะต้องจบลงด้วยการกด Enter หรือ Carriage return ยกเว้นคำสั่ง A/ (ไม่ต้องกดปุ่ม Enter) ซึ่งหมายถึงให้โมเด็มกลับไปทำคำสั่งชุดล่าสุดซ้ำอีกครั้งหนึ่ง การที่ Hayes ได้กำหนดให้คำสั่งต่างๆ ขึ้นต้นด้วยอักษร AT ก็เพราะต้องการให้โมเด็มรับรู้ถึงความสามารถและรูปแบบของคำสั่ง ที่ถูกส่งออกมาจากพอร์ตสื่อสารอนุกรมของ PC ซึ่งผู้อ่านอย่าสับสนระหว่างความเร็วของพอร์ตสื่อสารอนุกรม (ความเร็วของ DTE) และความเร็วของโมเด็ม (ความเร็วของ DCE) ในกรณีนี้จะหมายถึงความเร็วของ DTE

ตัวอย่างการใช้ชุดคำสั่ง AT ก็ได้แก่ คำสั่ง ATH หมายถึงคำสั่งที่ให้โมเด็มวางสายโทรศัพท์ เป็นต้น แต่บางคำสั่งอาจจะต้องการพารามิเตอร์เพิ่มเติม อย่างเช่น คำสั่ง ATDT7113022 หมายถึงให้หมุนหมายเลข 7113022 โดยหมุนระบบ TONE (T) แต่ถ้าต้องการให้โมเด็มหมุนระบบ Pulse ก็ต้องใช้คำสั่ง ATDP7113022 เป็นต้น

2.5.5 การตอบสนองคำสั่งของโมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ส่งคำสั่งต่างๆ ไปให้กับโมเด็มแล้ว โมเด็มก็จะสนองคำสั่งโดยจะส่งข้อความที่เรียกว่า Result code กลับมายังเครื่อง PC และจะปรากฏขึ้นบนจอภาพขณะที่รันโปรแกรมสื่อสารไว้ Result code เหล่านี้จะเป็นภาษาอังกฤษ อย่างเช่น OK, ERROR, CONNECT2400 หรืออาจจะเป็นตัวเลขอื่นๆ ซึ่งคำสั่งที่เกี่ยวข้องกับการส่ง Result code ก็จะมีดังนี้

ATV0 เป็นคำสั่งที่กำหนดให้โมเด็ม Result code ที่จำเป็นออกมาเท่านั้น (Non-verbose)

ATV1 เป็นคำสั่งที่กำหนดให้โมเด็มส่ง Result code ออกมาทุกๆ ครั้ง หลังจากที่ได้รับคำสั่งต่างๆ (Verbose)

ATQ1 เป็นคำสั่งที่กำหนดให้โมเด็มไม่ส่ง Result code ออกมา (Quiet mode)

ATQ0 เป็นคำสั่งที่ใช้ Toggle ให้โมเด็มสามารถกลับไปส่ง Result code ได้ดังเดิม

นอกจากนี้ยังมีชุดคำสั่งที่เกี่ยวข้องซึ่งจะได้กล่าวถึงในหัวข้อสรุปชุดคำสั่ง AT ในภาคผนวก ต่อไป

2.5.6 ชุดคำสั่งเพิ่มเติม

ชุดคำสั่งรุ่นแรกๆ ของ Hayes นั้นจะใช้ตัวอักษรตั้งแต่ A จนถึง Z เช่น ATA จนถึง ATZ ซึ่งก็หมายความว่า คำสั่งจะมีใช้ได้ไม่เกิน 26 คำสั่งเท่านั้น แต่โมเด็มในปัจจุบันได้มีความสามารถต่างๆ ที่เพิ่มมากขึ้นเข้ามา ดังนั้น Hayes จึงได้กำหนดชุดคำสั่งเพิ่มเติมขึ้นมา โดยใช้ตัวอักษร & เป็นตัวเข้ามาประกอบ ดังตัวอย่าง AT&F หมายถึงให้โมเด็มเรียกค่าจะมีให้โมเด็มเรียกค่า Default Setting ขึ้นมาใช้งาน ซึ่งโมเด็มของผู้ผลิตรายอื่นๆ ก็อาจจะเพิ่มเติมชุดคำสั่งที่ใช้ตัวอักษรอื่นๆ ประกอบ อย่างเช่น % หรือ * เป็นต้น

2.5.7 การตั้งค่าในรีจิสเตอร์ S

ดังที่กล่าวมาแล้วว่ารีจิสเตอร์ S มีหน้าที่เก็บข้อมูลที่ใช้ควบคุมสถานะการทำงานของโมเด็มเอาไว้ ซึ่งผู้ใช้สามารถตั้งค่า หรือตรวจดูค่าที่เก็บเอาไว้ในรีจิสเตอร์ S ต่างๆ ได้โดยใช้คำสั่งที่ขึ้นต้นด้วย ATS ในกรณีที่ต้องการตั้งค่าให้แก่รีจิสเตอร์ S ผู้ใช้จะต้องใส่ตัวเลขที่ระบุว่าเป็นรีจิสเตอร์ S ตัวที่เท่าใดตามด้วยเครื่องหมาย = และค่าที่ต้องการใส่ไป ดังตัวอย่างเช่น ถ้าต้องการให้รีจิสเตอร์ S0 มีค่าเท่ากับ 9 ก็จะต้องใช้คำสั่งต่อไปนี้

ATS0=9

ส่วนในกรณีที่ต้องการตรวจดูค่าของรีจิสเตอร์ S ต่างๆ ก็จะต้องพิมพ์ คำสั่งที่ขึ้นต้นด้วย ATS ตาม ด้วยตัวเลขที่ระบุว่าเป็นรีจิสเตอร์ S ตัวที่เท่าไหร่ และเครื่องหมาย? หลังจากที่กด Enter หรือ Carriage return แล้วโมเด็มก็จะส่งค่าที่คุณต้องการทราบในรีจีสเตอร์ S นั้นๆ ออกมาพร้อมกับคำว่า “OK” ดังตัวอย่างเช่น ถ้าต้องการ ตรวจดูค่าของรีจิสเตอร์ S11 จะต้องใช้คำสั่งดังนี้

ATS11?

จากนั้นหน้าจอจะปรากฏข้อความ

095

OK

ผลลัพธ์ที่ออกมาเป็น 095 ก็คือค่าที่เก็บอยู่ในรีจิสเตอร์ S11 นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.8 การส่งหลายคำสั่งในครั้งเดียว

ในบางกรณีที่ต้องการส่งคำสั่งไปให้โมเด็มมากกว่าหนึ่งคำสั่ง ผู้ใช้อาจจะส่งคำสั่งทั้งหมดไปในครั้งเดียวได้ ซึ่งโมเด็มจะสามารถรับส่งได้ไม่เกิน 40 ตัวอักษร ทั้งนี้จะนับตัวอักษร AT และ Carriage return ด้วย ผู้ใช้อาจจะพิมพ์คำสั่งทั้งหมดให้ตัวอักษรอยู่ติดกัน หรือเว้นช่องว่างแต่ละคำสั่งได้ ดังตัวอย่าง

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

ซึ่ง S7=90 หมายถึงให้ใส่ค่า 90 ลงไปในรีจิสเตอร์ S7

V1 หมายถึง ให้โมเด็มส่ง Result code ออกมาให้บางคำ อย่างเช่น OK, ERROR, CONNECT2400 เป็นต้น

DT1 – 305 – 555 -1234 หมายถึง ให้โมเด็มทำการหมุนหมายเลขโทรศัพท์ 1-305-555-1234 โดยการหมุนระบบ Tone

2.5.9 การเก็บค่าต่างๆที่เซตเอาไว้ใน Profile

จะเห็นได้ว่า ผู้ใช้สามารถตั้งค่ารีจิสเตอร์ S ต่างๆ และใช้คำสั่ง AT เพื่อควบคุมการทำงานของโมเด็มได้ในลักษณะต่างๆ ที่ต้องการ ซึ่ง Hayes ก็ได้ออกแบบหน่วยความจำภายในโมเด็มขึ้นมา เพื่อใช้สำหรับเก็บค่าต่างๆ ที่ตั้งไว้ และมีให้เลือกใช้ได้มากถึง 4 ชุดหน่วยความจำ โดยที่หน่วยความจำชุดแรกเรียกว่า แอกทีฟคอนฟิกูเรชั่น (Active Configuration) ใช้เก็บข้อมูลต่างๆ ที่เซตเอาไว้ใช้งานในปัจจุบัน เมื่อผู้ใช้มีการเปลี่ยนแปลงค่าของรีจิสเตอร์ S ต่างๆ และได้ตั้งคำสั่ง AT บางคำสั่ง ข้อมูลการตั้งค่าต่างๆ เหล่านี้ก็จะถูกนำไปเก็บเอาไว้ในแอกทีฟคอนฟิกูเรชั่น และ หลังจากปิดเครื่องไปแล้ว ข้อมูลเหล่านี้ก็จะสูญหายไป นอกจากนี้ หน่วยความจำที่เหลืออยู่จะมีอยู่ 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 และนำไปเก็บไว้ที่แอกทีฟคอนฟิกูเรชั่น หลังจากที่ได้ติดตั้งและเปิดเครื่องโมเด็มเป็นครั้งแรก โมเด็มก็จะถูกกำหนดให้เรียกเอา Factory Setting ออกมาใช้งานเสมอ ซึ่งจะมีประโยชน์มากในกรณีที่ผู้ใช้อาจจะประสบกับปัญหาการเซตค่าต่างๆ กลับขึ้นมาใช้งานใหม่ได้ และข้อมูลที่เป็นค่าดีฟอลต์เหล่านี้จะมีอยู่ในคู่มือการใช้งานโมเด็มที่มาพร้อมกับโมเด็ม ส่วนหน่วยความจำที่เป็นแบบ NVRAM ทั้งสองชุดจะถูกเรียกว่า User Profiles มีไว้สำหรับให้ผู้ใช้สามารถสำรองเก็บค่าต่างๆ ที่เซตเอาไว้ตามที่ต้องการ ซึ่งหน่วยความจำแบบ NVRAM นี้จะไม่สูญหายเมื่อปิดเครื่อง และทุกครั้งหลังจากที่ได้ปิดเครื่องแล้ว โมเด็มก็จะนำข้อมูลจากโพรไฟล์ที่ 0 มาเก็บไว้ที่แอกทีฟคอนฟิกูเรชั่น โดยอัตโนมัติ แต่ถ้าต้องการให้โมเด็มเรียกโพรไฟล์ที่ 1 ขึ้นมาทุกๆ ครั้งที่เปิดเครื่อง ก็ จะต้องใช้คำสั่ง AT&Y1 ดังจะกล่าวในหัวข้อสรุปชุดคำสั่ง AT ในภาคผนวกต่อไป

2.5.9.1 การสร้างและการเรียกใช้ User Profile

การสร้าง User Profile จะมีขั้นตอนดังต่อไปนี้

1.ให้โปรแกรมสื่อสารอยู่ในสภาวะคำสั่ง (หรือเรียกว่า Local Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ใช้คำสั่ง AT เพื่อตั้งค่าพารามิเตอร์ต่างๆ และตั้งค่ารีจิสเตอร์ S ที่ต้องการ

3.ใช้คำสั่ง AT&W0 หรือ AT&W1 เพื่อเก็บข้อมูลการเซตต่างๆ เอาไว้ในโพรไฟลล์ที่ 0 หรือ 1

หลังจากนั้นผู้ใช้สามารถเรียกใช้ User Profile ต่างๆ ที่ได้จัดเก็บเอาไว้ ขึ้นมาใช้ได้โดยคำสั่ง ATZ0 หรือ ATZ1 เพื่อรีเซ็ตโมเด็ม และเรียกข้อมูลการเซตต่างๆ ในโพรไฟล์ที่ 0 หรือโพร์ไฟล์ที่ 1 ขึ้นมาใหม่อีกครั้ง โมเด็มก็จะดึงเอาโพรไฟล์ที่ 0 ขึ้นมาเก็บไว้ที่แอกทีฟคอนฟิกูเรชั่น โดยอัตโนมัติ ซึ่งผู้ใช้สามารถกำหนดให้โมเด็มอ่านโพรไฟล์ที่ 1 ขึ้นมาแทน โดยใช้คำสั่ง AT&Y1 ก่อนที่จะปิดเครื่องได้

2.5.9.2 การเก็บหมายเลขโทรศัพท์ไว้ในโมเด็ม

ผู้ใช้สามารถเก็บหมายเลขโทรศัพท์เอาไว้ในโมเด็ม ทั้งสามารถเก็บตัวอักษรที่จำเป็นในการหมุนโทรศัพท์ได้ดังเช่น T (Tone), P (Pulse), W (Wait) โดยใช้คำสั่ง

AT&Zn = xxxxxx

และสามารถสั่งให้โมเด็มหมุนโทรศัพท์ตามหมายเลขที่เก็บไว้ได้โดยใช้คำสั่ง

ATS = n

ดังตัวอย่างเช่น

ถ้าต้องการเก็บหมายเลข 3197707 เอาไว้ในที่เก็บตำแหน่งที่ 2 และเมื่อเรียกใช้หมายเลขนี้จะกำหนดให้โมเด็มหมุนแบบ Tone ให้ใช้คำสั่งดังนี้

AT&Z2 = T3197707

และเมื่อต้องการให้โมเด็มหมุนหมายเลขโทรศัพท์ที่เก็บเอาไว้นี้ให้ใช้คำสั่ง

ATDS = 2

โดยทั่วไปแล้ว โปรแกรมสื่อสารก็มักจะมีฟังก์ชันการเก็บและหมุนหมายเลขไว้ในตัวเช่นกัน ซึ่งมักจะเรียกว่า โฟนบุ๊ก (Phone Book) หรือ Dialing Directory แต่สำหรับหรับในกรณีการเก็บหมายเลขโทรศัพท์เอาไว้ใน NVRAM เช่นนี้ มักจะใช้เกี่ยวกับ กระบวนการ Dial Backup Number ซึ่งใช้ในระบบสื่อสารที่ค่อนข้างจะอัตโนมัติและต้องการให้โมเด็มติดต่อสื่อสารอยู่ตลอดเวลา กระบวนการ Dial Backup Number นี้จะมีขั้นตอนการทำงานคือ ขณะที่โมเด็มต้นทางติดต่อสื่อสารกับโมเด็มปลายทางแล้วเกิดปัญหาสายขาด หรือสายหลุดไม่สามารถจะดำเนินการสื่อสารต่อไปได้ โมเด็มต้นทางก็จะสามารถหมุนเบอร์โทรศัพท์ที่เก็บเอาไว้ใน NVRAM ได้โดยอัตโนมัติ เพื่อทำการโทรศัพท์กลับไปเชื่อมต่อกับปลายทางอีกครั้งหนึ่ง หรือโทรศัพท์ไปยังโมเด็มตัวอื่นๆ ตามที่ผู้ใช้กำหนดไว้

2.5.10 การควบคุมการไหล

การควบคุมการไหล (Flow Control) ของข้อมูลเปรียบเทียบเสมือนได้กับวาล์วปิด – เปิด ทางเดินของข้อมูลทั้งทางด้านขาเข้าและขาออก มีหน้าที่คอยควบคุมจังหวะการไหลเข้าระหว่าง PC และ โมเด็ม การที่จำเป็นจะต้องมีการควบคุมการไหล ก็เพราะว่าโมเด็มจะไม่สามารถรับส่งข้อมูลได้เร็วเท่ากับ PC หรือ PC อาจจะต้องไปทำงานอย่างอื่น จึงต้องสั่งให้โมเด็มหยุดการส่งข้อมูลให้แก่ PC ชั่วคราว การควบคุมการไหลของข้อมูลจะมีอยู่ 2 แบบคือ แบบซอฟต์แวร์ และแบบฮาร์ดแวร์ นอกจากนี้โมเด็มที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลการควบคุมความผิดพลาด (error control) อย่างเช่น MNP และ V.42 จะสามารถส่งสัญญาณการควบคุมการไหลของข้อมูล ไปให้กับโมเด็มปลายทางที่ไม่มีการควบคุมการไหลดังกล่าวได้เช่นกัน

2.5.10.1 การควบคุมการไหลแบบฮาร์ดแวร์ (Hardware Flow Control)

วงจรการควบคุมการไหลในโมเด็มนั้น จะมีอยู่ 2 วงจรคือ RTS (Request to Send) และ CTS (Clear to Send) ซึ่งจะเป็นตัวอนุญาตให้โมเด็มหรือ PC ส่งข้อมูลให้แก่กัน ซึ่งข้อมูลที่จะรับหรือส่งนั้นจะถูกหยุดเอาไว้ชั่วคราว จนกว่าสัญญาณดังกล่าวจะอยู่ในสถานะ ON ซึ่งในกรณีที่โมเด็มไม่สามารถที่จะรับข้อมูลได้จาก PC โมเด็มก็จะควบคุมสัญญาณ RTS อยู่ในสถานะ OFF และโปรแกรมสื่อสารก็จะรับรู้ความเปลี่ยนแปลงของสัญญาณ RTS แล้วก็จะหยุดการส่งข้อมูลให้กับโมเด็มไว้ชั่วคราว จนกระทั่ง RTS กลับมาอยู่ในสถานะ ON อีกครั้งหนึ่ง และก็เช่นเดียวกัน ถ้าโปรแกรมสื่อสารต้องการที่จะหยุดการรับส่งข้อมูลจากโมเด็มเอาไว้ชั่วคราว ก็จะไปควบคุมให้สัญญาณ CTS อยู่ในสภาวะ OFF และโมเด็มก็จะหยุดการส่งข้อมูลให้แก่ PC ชั่วคราวจนกว่า CTS จะกลับมาอยู่ในสถานะ ON อีกครั้งหนึ่ง ซึ่งการรับส่งข้อมูลก็จะดำเนินไปเช่นเดิม

2.5.10.2 การควบคุมการไหลแบบซอฟต์แวร์ (Software Flow Control)

การควบคุมการไหลแบบซอฟต์แวร์ (Software Flow Control) มีลักษณะการทำงานคล้ายกับการควบคุมการไหลแบบฮาร์ดแวร์ แต่แทนที่จะใช้วงจร RTS/CTS การควบคุมการไหลแบบซอฟต์แวร์จะใช้ตัวอักขระที่เรียกว่า XON และ XOFF ซึ่ง จะตรงกับอักขระ CTRL-Q และ CTRL-S ที่พิมพ์มาจากคีย์บอร์ดตามลำดับ เทคนิคดังกล่าว ได้พัฒนามาจากการใช้งานพิมพ์ดีดในอดีต 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 เพื่อที่จะควบคุมจังหวะการหยุดและการไหลของข้อมูลจาก PC มายังโมเด็มได้

2.6 การพัฒนาแอพพลิเคชันบน Windows โดย Delphi 7

Delphi คือ ซอฟต์แวร์ที่นำมาใช้ในการเขียนโปรแกรม เพื่อสร้างและพัฒนาแอพพลิเคชันบน Windows โดยจะประกอบด้วยเครื่องมือชนิดต่างๆ ที่ใช้เพื่อให้การเขียนโปรแกรมทำได้อย่างสะดวก

Delphi จัดเป็นเครื่องมือเขียนโปรแกรมชนิด Visual Programming เช่นเดียวกับ Visual Basic หรือ Visual C++ โดยมีข้อดีคือ สามารถเขียนโปรแกรมได้ง่าย และให้ผลงานออกมาอย่างรวดเร็ว ซึ่งจะแตกต่างจากเครื่องมือเขียนโปรแกรมรุ่นเดิมๆ เช่น Turbo Pascal หรือ Borland C ที่มีความยุ่งยากในการใช้งานและการเรียนรู้ในการเขียนโปรแกรม ดังนั้นจึงจัดให้ Delphi เป็นซอฟต์แวร์ประเภท RAD (Rapid Application Development) ซึ่งหมายถึง Delphi สามารถสร้างและพัฒนาแอพพลิเคชันได้อย่างรวดเร็ว โดยใช้ภาษา Object Pascal บน Windows นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 ความสามารถหลักๆ ของ Delphi7

1.พัฒนาแอพพลิเคชันทั่วไปที่ทำงานบนระบบปฏิบัติการวินโดวส์ โดยเราสามารถสร้างโปรแกรมทางด้านกราฟฟิก โปรแกรมคำนวณต่างๆ ได้อย่างมากมายให้ตรงกับความต้องการได้ สามารถสร้างงานได้อย่างรวดเร็ว ขนาดโปรแกรมที่เล็กกะทัดรัดเมื่อเทียบกับแอพพลิเคชันที่สร้างจากเครื่องมือตัวอื่นๆ โดยทั่วไปนิยมนำไปสร้างแอพพลิเคชันทั้งเพื่อการศึกษา และแอพพลิเคชันที่ใช้ในระบบงานจริงๆ ในโลกธุรกิจ

2.พัฒนาแอพพลิเคชันด้านฐานข้อมูล ซึ่ง Delphi 7 มีจุดเด่นเป็นอย่างมากในการสร้างแอพพลิเคชันประเภทนี้ ไม่ว่าจะเป็นความง่ายในการพัฒนา ความเร็วของแอพพลิเคชันที่ได้ หรือความพร้อมของเครื่องมือที่ใช้ในการพัฒนาแอพพลิเคชัน

Delphi 7 สามารถติดต่อกับไฟล์ฐานข้อมูลทั่วไป เช่น dBASE, Paradox และ Access หรือติดต่อกับฐานข้อมูลบนเครื่อง Server ไม่ว่าจะเป็น Oracle, Interbase, SQL Servser หรือฐานข้อมูลประเภทอื่นๆ ได้

3.สร้าง ActiveX แบบต่างๆ ได้แก่ ActiveX Control และ ActiveFrom ซึ่ง ActiveX Control ก็คือเครื่องมือที่ช่วยให้เราสามารถนำส่วนของโปรแกรมที่เราได้สร้างเอาไว้ใน Delphi ไปใช้ในโปรแกรมอื่นๆ ได้ เช่น ใน Visual Basic หรือ Microsoft Excel เป็นต้น ส่วน ActiveFrom ก็คือแอพพลิเคชันที่สามารถนำไปใส่ในอินเตอร์เน็ต และเรียกใช้งานโดยใช้เว็บบราวเซอร์ เช่น Internet Explorer ได้

4.พอร์ตโปรแกรมไปใช้งานบน Linux ได้ง่าย คือ Delphi มีความสามารถในการนำโปรแกรมที่เคยพัฒนาบน Windows ไปดัดแปลงให้สอดคล้องกับสภาพแวดล้อมการทำงานของลีนุกซ์โดยใช้เครื่องมือที่มีชื่อว่า Kylix แต่ถ้าหากต้องการพัฒนาแอพพลิเคชันบนลีนุกซ์โดยตรง เราก็สามารถใช้งาน Kylix พัฒนาได้ โดย Kylix นั้นมีภาษาโปรแกรม และวิธีการใช้งานใกล้เคียงกับ Delphi มาก ทำให้ ถ้าเรามีพื้นฐานการสร้างแอพพลิเคชันกับ Delphi แล้วสามารถประยุกต์กับการสร้างแอพพลิเคชันบนลีนุกซ์ได้อย่างรวดเร็ว

2.7 Delphi กับฐานข้อมูล

ชีวิตประจำวันของเรา จะต้องมีส่วนเข้าไปเกี่ยวข้องกับแอพพลิเคชันฐานข้อมูลไม่มากก็น้อย ซึ่งเป็นสิ่งที่หลีกเลี่ยงไม่ได้ เช่น การเบิกถอนเงินสดจากเครื่อง ATM หรือการชำระเงินเมื่อเราซื้อของจากร้านค้า เป็นต้น

2.7.1ฐานข้อมูลกับความสามารถของ Delphi 7

ฐานข้อมูล เป็นที่รวบรวมข้อมูลที่มีความสัมพันธ์กัน โดยข้อมูลเหล่านี้จะถูกเก็บอยู่ในรูปแบบของตาราง ซึ่งประกอบไปด้วยฟิลด์ (Field) และเร็คคอร์ด (Record)

ฟิลด์ คือหัวข้อของข้อมูลที่จัดกลุ่มอยู่ในตาราง เช่น ชื่อ อายุ เบอร์โทรศัพท์ เป็นต้น ถ้าใส่ข้อมูลต่างๆ จนครบทุกฟิลด์ในตารางแล้ว ก็จะได้เรคคอร์ดขึ้นมา หรืออาจกล่าวได้ว่าฟิลด์และเร็คคอร์ด ก็คือคอลัมน์และแถวของตารางนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานด้านฐานข้อมูล เป็นงานที่เกี่ยวข้องกับการสร้างพื้นที่เพื่อใช้ในการเก็บข้อมูล(ตาราง) และมีระบบในการจัดเก็บข้อมูลเหล่านั้น โดยที่ฐานข้อมูลหนึ่งๆ อาจประกอบไปด้วยตารางหลายๆ ตาราง หรืออาจมีตารางเดียวก็ได้

Delphi เป็นเครื่องมือที่มีประสิทธิภาพในการสร้างแอพพลิเคชันฐานข้อมูลตัวหนึ่ง โดยมีเครื่องมือต่างๆ ที่ช่วยให้การสร้างแอพพลิเคชันฐานข้อมูลลดความซับซ้อนลง และยังช่วยเพิ่มประสิทธิภาพของแอพพลิเคชัน ไม่ว่าจะเป็นเครื่องมือที่ช่วยในการสร้างตารางขึ้นมาใช้งาน หรือเครื่องมือที่ใช้ในการติดต่อกับฐานข้อมูลเพื่อนำข้อมูลออกมาใช้งาน เป็นต้น

2.7.2 การเข้าถึงฐานข้อมูลของ Delphi 7

พื้นฐานในการสร้างแอพพลิเคชันฐานข้อมูลใดๆ ด้วย Delphi นั้น จำเป็นต้องมีตัวกลางที่ใช้ในการเข้าถึงฐานข้อมูล ซึ่งตัวกลางดังกล่าวเรียกว่า Database Engine ด้วยเหตุนี้ Delphi จึงมี Borland Database Engine หรือ BDE เป็นตัวจัดการในการเข้าถึงฐานข้อมูล เริ่มตั้งแต่เวอร์ชันแรกจนถึงเวอร์ชันปัจจุบัน

เนื่องจากเทคโนโลยีได้มีการพัฒนาขึ้นมาอย่างต่อเนื่อง ทำให้การเข้าถึงฐานข้อมูลของ Delphi ตั้งแต่เวอร์ชัน 5 เป็นต้นมา ไม่ได้มีเพียงวิธีเดียวอีกต่อไป Delphi สามารถเข้าถึงฐานข้อมูลได้โดยไม่ต้องใช้ BDE เป็นตัวกลาง โดยจะอาศัยเทคโนโลยีที่เรียกว่า ActiveX Data Objects หรือที่รู้จักกันรูปของตัวย่อว่า ADO

นอกจากวิธีการเข้าถึงข้อมูลที่กล่าวไว้ข้างต้นแล้ว เราสามารถสร้างแอพพลิเคชันฐานข้อมูลที่มีการเข้าถึงฐานข้อมูลด้วยวิธีที่แตกต่างกันออกไปได้อีก ซึ่งวิธีการต่างๆ ในการเข้าถึงฐานข้อมูลสรุปได้ดังนี้

- Borland Database Engine(BDE) เป็น Database Engine ที่มาพร้อมกับ Delphi ช่วยในการเข้าถึงฐานข้อมูล ถือได้ว่าเป็นจุดเริ่มต้นของการทำงานกับฐานข้อมูลใน Delphi
- ActiveX Data Objects(ADO) เป็นเทคโนโลยีของบริษัท Microsoft ที่รวบรวมความสามารถในการเข้าถึงฐานข้อมูลเข้าไว้ในออบเจ็กต์ ดังนั้นจึงไม่ต้องใช้ข้อมูลอื่นมาช่วยในการเข้าถึงข้อมูล
- dbExpress Library เป็นกลุ่มของไดรเวอร์ขนาดเล็กที่สามารถเข้าถึงข้อมูลในฐานข้อมูลได้อย่างรวดเร็ว สามารถนำข้อมูลออกมาแสดงได้เท่านั้น ไม่สามารถทำการแก้ไข ปรับปรุงได้ เหมาะแก่การนำเสนอในรูปแบบของรายงาน
- InterBase Express (IBX) เป็นการเข้าถึงฐานข้อมูล InterBase ซึ่งเป็นฐานข้อมูลที่เป็นผลิตภัณฑ์ของบริษัท Borland โดยการเข้าถึงข้อมูล จะมีคอมโพเนนต์เฉพาะที่เก็บไว้ในเพจ InterBase ของคอมโพเนนต์พาเลท

2.7.3 Database Desktop กับงานฐานข้อมูล

Database Desktop เป็นเครื่องมือที่มาพร้อมกับ Delphi ซึ่งมีความสามารถในการแสดง จัดเรียง แก้ไข เรียกค้นข้อมูลจากตาราง รวมทั้งสร้างตารางเพื่อใช้เก็บข้อมูล โดยตารางที่ Database Desktop สามารถสร้างได้นั้นมีหลายประเภท เช่น Paradox, dBase หรือ SQL Server เป็นต้น

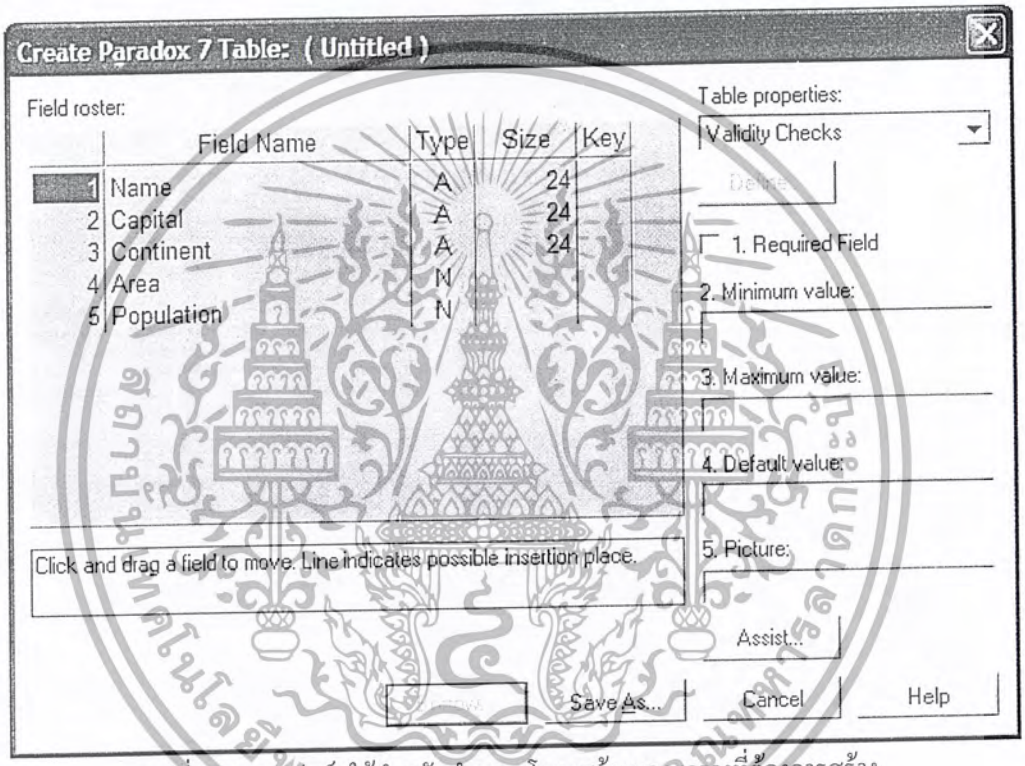
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างตารางด้วย Database Desktop

Database Desktop สามารถสร้างตารางได้หลายประเภทดังที่กล่าวไว้ข้างต้น แต่ไม่ได้หมายความว่าเราจะสามารถสร้างตารางได้ทุกประเภทจาก Database Desktop ทั้งนี้ขึ้นอยู่กับว่าเรามีไดรเวอร์ของฐานข้อมูลนั้นๆ หรือไม่ เช่น ถ้าต้องการสร้างตารางที่เป็น Microsoft Access ก็จำเป็นจะต้องมีไดรเวอร์ของ Microsoft Access ด้วย

Database Desktop มีไดรเวอร์ของฐานข้อมูลที่เตรียมไว้ให้ใช้งานได้ 2 ประเภท คือ Paradox และ dBase โดยทั้ง 2 ประเภทนี้เป็นตารางมาตรฐานที่ใช้งานใน Delphi

ในการสร้างตารางนั้นในขั้นแรกเราต้องรู้จักฟอร์มที่ใช้กำหนดโครงสร้างของตารางก่อนดังแสดงในรูปที่ 2.8



รูปที่ 2.8 แสดงฟอร์มใช้สำหรับกำหนดโครงสร้างของตารางที่ต้องการสร้าง

จากรูป 2.8 ฟอร์มที่ใช้กำหนดโครงสร้างของตารางประกอบไปด้วยหลายๆ ส่วนดังนี้

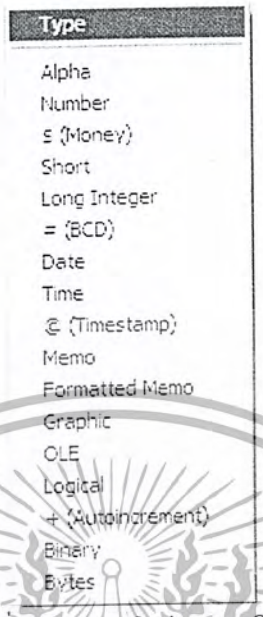
-Field Roster ใช้กำหนดฟิลด์ให้กับตาราง ประกอบไปด้วย ชื่อฟิลด์ (Field Name) ชนิดของฟิลด์ (Type) ขนาด (Size) และคีย์ (Key)

-Table properties กำหนดพร็อพเพอร์ตี้ของตาราง โดยเลือกจากรายการที่กำหนดมาให้ เช่น Validity Checks หรือ Secondary เป็นต้น

-ปุ่ม Borrow... เมื่อต้องการนำโครงสร้างของตารางที่มีอยู่แล้วมาประยุกต์ใช้งาน เราจะใช้ปุ่มนี้ในการกำหนดโครงสร้างของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

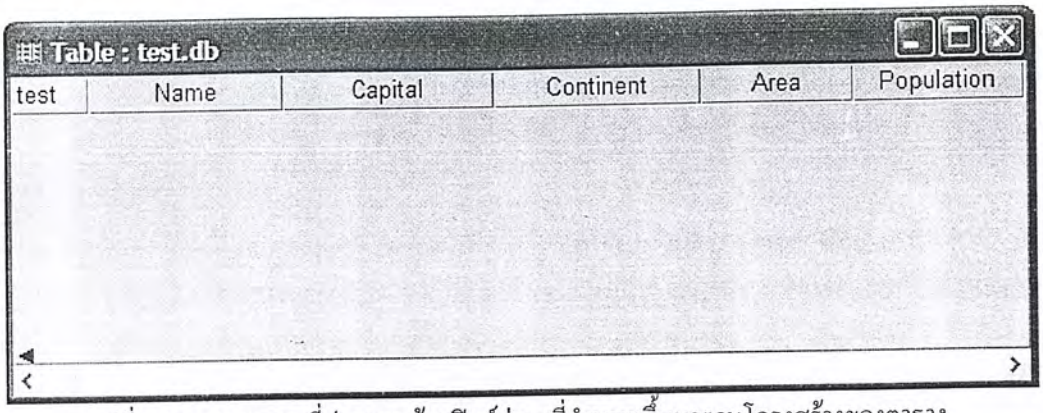
ในการสร้างตารางนั้นเราจะใส่ชื่อฟิลด์ลงไปในช่อง Field Name และทำการกำหนดชนิดของฟิลด์ในช่อง Type ซึ่งชนิดของฟิลด์แสดงได้ดังรูปที่ 2.9



รูปที่ 2.9 แสดงชนิดต่างๆของฟิลด์

จากรูปที่ 2.9 เราต้องเลือกชนิดของฟิลด์ว่าชนิดใดที่เหมาะสมกับข้อมูลที่จะใส่ในฟิลด์นั้นๆ แล้วถ้าชนิดของฟิลด์ที่เลือกเป็นชนิดที่ต้องกำหนดขนาด ให้ทำการกำหนดขนาดของฟิลด์โดยใส่ลงไปในช่อง Size จากนั้นในช่องของ Key ถ้าต้องการกำหนดให้ฟิลด์นั้นเป็น Primary Key โดยใส่เครื่องหมายดอกจัน (*) เป็นสัญลักษณ์ที่บ่งบอกว่าฟิลด์นั้นถูกกำหนดให้เป็น Primary Key ฟิลด์ที่ถูกกำหนดให้เป็น Primary Key ข้อมูลในฟิลด์นี้จะเป็นข้อมูลเฉพาะ นั่นคือจะทำให้ไม่มีเร็คคอร์ดใดซ้ำกัน รวมถึงเป็นฟิลด์ที่ถูกจัดเรียงข้อมูลโดยอัตโนมัติ จากรูปที่ 2.8 เรากำหนดให้ฟิลด์ชื่อว่า Name โดยมีชนิดของฟิลด์เป็นแบบ A (Alpha) เป็นตัวอักษรเนื่องจากชื่อประเทศต้องเก็บเป็นตัวอักษร และกำหนดให้มีขนาด 24 ตัวอักษร แล้วกำหนดให้เป็น Primary Key เนื่องจากชื่อประเทศห้ามซ้ำกัน หรือในฟิลด์ชื่อ Area เรากำหนดให้ชนิดของฟิลด์เป็น N (Number) คือมีชนิดของข้อมูลเป็นตัวเลข ซึ่งฟิลด์ชนิดนี้ไม่ต้องทำการกำหนดขนาด เป็นต้น หลังจากที่ได้กำหนดโครงสร้างของตารางเรียบร้อยแล้ว เราก็จะได้ตารางที่ใช้เก็บข้อมูลออกมาดังรูปที่ 2.10 ซึ่งเราจะนำตารางที่ได้นำไปประยุกต์ใช้งานแอพพลิเคชันฐานข้อมูลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงตารางที่ประกอบด้วยฟิลด์ต่างๆที่กำหนดขึ้นมาตามโครงสร้างของตาราง

2.8 การเรียกค้นด้วยภาษา SQL

คิวรี (Query) เป็นวิธีการเรียกค้นกลุ่มของข้อมูลที่ต้องการจากตารางออกมาแสดง โดยจะอยู่ในรูปแบบของคำถาม เช่น ดีวีดีภาพยนต์เรื่องใดที่ออกฉายในปี ค.ศ. 2003 หรือจำนวนดีวีดีภาพยนต์ที่ออกฉายในปีค.ศ. 1999 มีทั้งหมดกี่เรื่อง นอกจากจะเรียกค้นข้อมูลแล้ว ยังสามารถใช้คำสั่งในการจัดการกับกลุ่มของข้อมูลเหล่านั้นได้ด้วย เช่น การเรียงลำดับของข้อมูล หรือการคำนวณต่างๆ กับข้อมูลที่เป็นตัวเลข เช่น หาค่าสูงสุด หาค่าเฉลี่ย เป็นต้น

SQL (Structure Query Language) เป็นภาษามาตรฐานที่ใช้ในการเขียนคิวรี เพื่อใช้ในการเรียกค้นข้อมูล และจัดการกับข้อมูลในฐานข้อมูล โดยรูปประโยคของภาษา SQL นั้นจะอยู่ในรูปแบบที่สามารถเข้าใจได้ง่าย เนื่องจากมีความใกล้เคียงกับภาษาอังกฤษที่ใช้ในชีวิตประจำวัน จึงทำให้สามารถทำความเข้าใจและเขียนคิวรีโดยใช้ SQL ได้อย่างไม่ยากนัก

ภาษา SQL ใน Delphi

คำสั่งต่างๆ ในภาษา SQL เป็นคำสั่งมาตรฐาน ดังนั้นจึงสามารถนำไปใช้ได้ในหลายๆ ระบบ ไม่ว่าจะเป็นใน Delphi หรือเป็นระบบฐานข้อมูลของ ORACLE ซึ่งแต่ละระบบอาจจะมีการกำหนดรูปแบบในการใช้คำสั่งแตกต่างกันออกไป แต่ส่วนใหญ่จะมีรูปแบบที่เหมือนๆ กัน

ในการสร้างแอพพลิเคชันฐานข้อมูลใน Delphi นั้น ภาษา SQL เป็นส่วนสำคัญในการทำให้แอพพลิเคชันที่สร้างขึ้นมีความสามารถตามที่ต้องการและมีประสิทธิภาพมากขึ้นด้วย

2.8.1 ภาษาสำหรับการเรียกค้น

การเรียกค้น เป็นการค้นหาข้อมูลที่ต้องการจากตารางในฐานข้อมูล ซึ่งเป็นส่วนสำคัญในการทำงานด้านฐานข้อมูล ในกรณีที่เรามีข้อมูลอยู่จำนวนมาก การค้นหาข้อมูลโดยการเข้าถึงข้อมูลทีละเร็คคอร์ดในตารางจะทำให้เสียเวลามาก การเรียกค้นจึงเป็นวิธีที่นำมาใช้เพื่อให้การค้นหาข้อมูลจากฐานข้อมูลเป็นไปอย่างรวดเร็ว

เรียกค้นข้อมูลด้วย SELECT และ FROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปประโยคพื้นฐานของภาษา SQL สำหรับการเรียกค้น สามารถแสดงได้ดังนี้

SELECT ชื่อฟิลด์ข้อมูล, ชื่อฟิลด์ข้อมูล,...

FROM ชื่อตาราง, ชื่อตาราง,...

จากรูปประโยคข้างบนนี้ ถ้าแปลกันอย่างคร่าวๆ จะสามารถเข้าใจโดยทันทีว่าเป็นการเลือกฟิลด์ต่างๆ จากตารางที่กำหนด ดังตัวอย่างต่อไปนี้เช่น ถ้าเราจะเรียกค้นข้อมูลจากตาราง tblDvd โดยข้อมูลที่ต้องการเลือกขึ้นมาคือ ชื่อดีวีดีภาพยนต์ (DvdTitle) ปีที่ออกฉาย (Released) และระดับของดีวีดีภาพยนต์ (Rating) SQL ในการเรียกค้นข้อมูลดังกล่าวจะได้

SELECT DvdTitle, Released, Rating

FROM tblDvd

จากตัวอย่างนี้เราใช้คำสั่ง SELECT ในการกำหนดชื่อฟิลด์ข้อมูลที่เราต้องการค้นหาคือ DvdTitle, Released และ Rating ส่วนคำสั่ง FROM เป็นตัวบอกให้รู้ว่าฟิลด์ที่กำหนดนั้นมาจากตารางใด ซึ่งจากตัวอย่างก็คือ ตาราง tblDvd ผลลัพธ์ที่ได้ก็จะมีเฉพาะฟิลด์ที่เรากำหนดในส่วน SELECT เท่านั้น

เราสามารถใช้เครื่องหมายดอกจัน (*) ในการกำหนดแทนชื่อฟิลด์ทั้งหมดในตาราง แทนการพิมพ์ชื่อฟิลด์ทั้งหมด เช่น SELECT * FROM tblDvd เป็นการเรียกดูข้อมูลทุกฟิลด์จากตาราง tblDvd

SQL มีความยืดหยุ่นในการใช้งาน โดยสามารถเขียน SQL ให้อยู่ภายในบรรทัดเดียวกันทั้งหมด หรือจะแบ่งบรรทัดตามต้องการก็ได้ ซึ่งจะให้ผลลัพธ์ออกมาเหมือนกัน

คำสั่ง SELECT และ FROM สามารถช่วยให้เราเรียกค้นข้อมูลที่ต้องการได้ในระดับหนึ่งเท่านั้น ถ้าต้องการเจาะลึกเข้าไปในรายละเอียดของแต่ละเร็คคอร์ดแล้ว จะไม่สามารถเรียกค้นข้อมูลตามที่ต้องการได้เลย เช่น ต้องการเรียกค้นข้อมูลของดีวีดีภาพยนต์ที่ออกฉายในปีค.ศ. 2000 เป็นต้น เมื่อมีเงื่อนไขเพิ่มขึ้นมา SQL ก็มีส่วนที่ใช้ในการกำหนดเงื่อนไขเพื่อให้เราเรียกค้นข้อมูลได้ตามที่ต้องการคือ WHERE จากเงื่อนไขข้างต้น จะได้รูปประโยคของ SQL ดังนี้

SELECT ชื่อฟิลด์ข้อมูล, ชื่อฟิลด์ข้อมูล,...

FROM ชื่อตาราง, ชื่อตาราง,...

WHERE เงื่อนไข

การกำหนดเงื่อนไขในคิวรี เป็นวิธีการเปรียบเทียบค่า ถ้าค่าที่เราเปรียบเทียบนั้นเป็นจริงก็จะแสดงผลลัพธ์ออกมาตามเงื่อนไขที่กำหนด เครื่องหมายที่ใช้ในการเปรียบเทียบค่าในคิวรีนั้น เป็นเครื่องหมายที่คุ้นเคยกันในทางคณิตศาสตร์ เช่น =, <, >, <= และ >= ส่วนเครื่องหมายที่ไม่เท่ากับเขียนแทนด้วย != หรือ <> เช่น จากตัวอย่างข้างต้นจะได้เป็น

SELECT DvdTitle, Rating, Released

FROM tblDvd

WHERE Released = 2000

ข้อมูลที่เป็น NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เร็คคอร์ดที่ไม่มีข้อมูลอยู่ เราจะเรียกว่าฟิลด์ที่มีข้อมูลเป็น NULL แต่ถ้าข้อมูลใดมีค่าเท่ากับศูนย์ เราจะไม่เรียกว่า NULL โดยเราจะไม่ใช้เครื่องหมายทางคณิตศาสตร์ในการเปรียบเทียบข้อมูลที่เป็น NULL แต่จะใช้ IS แทน และเมื่อต้องการเปรียบเทียบข้อมูลที่ไม่ใช่ NULL ก็จะใช้ IS NOT เป็นตัวเปรียบเทียบ เช่น ต้องการเรียกค้นข้อมูลของดีวีดีภาพยนต์ที่ฟิลด์ Rating ไม่มีข้อมูล หรือมีค่าเป็น NULL จะได้

```
SELECT DvdTitle, Rating, Released
FROM tblDvd
WHERE Rating IS NULL
```

ตัวเชื่อมเงื่อนไข AND

เมื่อมีการกำหนดเงื่อนไขให้กับการเรียกค้นหลายๆ เงื่อนไข จำเป็นจะต้องกำหนดตัวเชื่อมให้กับเงื่อนไขต่างๆ เหล่านั้น โดยตัวเชื่อมเงื่อนไข AND ใช้กำหนดผลลัพธ์ที่จะถูกแสดงออกมา โดยข้อมูลนั้นจะต้องเป็นจริงตามเงื่อนไขที่กำหนดทั้งหมดตัวอย่างเช่น ต้องการเรียกค้นข้อมูลจากฟิลด์ DvdTitle, Released และ Rating ที่ออกฉายในปี ค.ศ.2000 และมี Released ไม่เป็น NULL จะได้

```
SELECT DvdTitle, Rating, Released
FROM tblDvd
WHERE Released = 2000 AND
      Rating IS NOT NULL
```

ตัวเชื่อมเงื่อนไข OR

ถ้ามีการกำหนดเงื่อนไขว่า ไม่ว่าเงื่อนไขใดเงื่อนไขหนึ่งเป็นจริง จึงจะแสดงข้อมูลออกมานั้น จะใช้ตัวเชื่อม OR ในการเชื่อมเงื่อนไขเหล่านั้น ตัวอย่างเช่น ต้องการเรียกค้นข้อมูลของดีวีดีภาพยนต์ที่ออกฉายในปี 2000 หรือดีวีดีภาพยนต์ที่มี Rating เป็น 'PG-13' จะได้

```
SELECT DvdTitle, Rating, Released
FROM tblDvd
WHERE Released = 2000 OR
      Rating = 'PG-13'
```

ผลลัพธ์ที่ได้ทั้งหมด จะมีข้อมูลที่เป็นไปตามเงื่อนไขอย่างน้อย 1 เงื่อนไข ซึ่งทำให้เงื่อนไขทั้งหมดเป็นจริง และผลลัพธ์ที่ได้ตรงกับความต้องการ

2.8.2 ภาษาสำหรับการจัดการข้อมูล

นอกจากคำสั่ง SQL ที่ใช้ในการเรียกค้นข้อมูลจากตารางแล้ว SQL ยังมีคำสั่งอีกส่วนหนึ่งที่ใช้ในการจัดการกับข้อมูลในตารางด้วย เช่น การเพิ่มข้อมูล การลบข้อมูล หรือการเปลี่ยนแปลงข้อมูลตามที่ต้องการได้ เพื่อเพิ่มประสิทธิภาพในการใช้งานด้านฐานข้อมูล

เพิ่มข้อมูลเข้าไปในตารางด้วย INSERT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง INSERT มีไว้สำหรับเพิ่มข้อมูลเข้าไปในตารางใดๆ โดยมีรูปแบบการใช้งานดังนี้

INSERT INTO ชื่อตาราง [(ชื่อฟิลด์1, ชื่อฟิลด์2, ...)]

VALUES (ค่าที่1, ค่าที่2, ...)

วิธีการใช้ INSERT เพื่อเพิ่มข้อมูลเข้าไปในตารางสามารถทำได้ 2 วิธีคือ การเพิ่มข้อมูลทุกฟิลด์ลงตาราง ตัวอย่างเช่น ต้องการเพิ่มดีวีดีภาพยนต์เรื่อง 'Air Force One' ลงในตาราง tblDvd ซึ่งเข้าฉายปี ค.ศ.1997 ระดับของดีวีดีภาพยนต์ (Rating) เป็น R จะได้

INSERT INTO tblDvd(DvdTitle, Rating, Released)

VALUES ('Air Force One', 'R', 1997)

และการเพิ่มข้อมูลเฉพาะฟิลด์ที่ต้องการ ตัวอย่างเช่น ต้องการเพิ่มดีวีดีภาพยนต์เรื่อง 'The Butterfly Effect' ลงในตาราง tblDvd ซึ่งเข้าฉายปี ค.ศ.2004 แต่เรายังไม่รู้ว่า Rating เท่าใดจึงทำการเพิ่มข้อมูลเฉพาะฟิลด์ที่รู้จะได้

INSERT INTO tblDvd(DvdTitle, Released)

VALUES ('The Butterfly Effect', 2004)

ดังนั้นผลลัพธ์ที่ได้ก็จะปรากฏข้อมูลของฟิลด์ที่เราเพิ่มขึ้นเท่านั้น โดยฟิลด์ Rating ที่เราไม่ได้กำหนดจะไม่มีข้อมูลใดๆ หรือมีค่าเท่ากับ NULL นั่นเอง

เปลี่ยนแปลงข้อมูลในตารางด้วย UPDATE

เมื่อต้องการเปลี่ยนแปลงข้อมูลในตารางจะใช้คำสั่ง UPDATE ซึ่งมีรูปแบบ ดังนี้

UPDATE ชื่อตาราง

SET ชื่อฟิลด์ = ค่าของข้อมูล, ชื่อฟิลด์ = ค่าของข้อมูล, ...

WHERE เงื่อนไข

ตัวอย่างเช่น มีข้อมูลอยู่ในตาราง tblDvd โดยต้องการแก้ไขดีวีดีภาพยนต์เรื่อง 'The Butterfly Effect' ซึ่งเข้าฉายปี ค.ศ.2004 เป็น ดีวีดีภาพยนต์เรื่อง 'Love Actually' ปี ค.ศ.2003 จะได้

UPDATE tblDvd

SET DvdTitle = 'Love Actually', Released = 2003

WHERE DvdTitle = 'The Butterfly Effect'

การใช้คำสั่ง UPDATE ในการเปลี่ยนแปลงข้อมูลในตารางนั้น จะมีประโยชน์อย่างยิ่งเมื่อต้องการแก้ไข หรือเปลี่ยนแปลงข้อมูลที่มีจำนวนมากพร้อมๆ กัน

ลบเร็คคอร์ดด้วย DELETE

เร็คคอร์ดที่เราไม่ต้องการ เราสามารถลบเร็คคอร์ดนั้นได้โดยใช้ DELETE มีรูปแบบ ดังนี้

DELETE FROM ชื่อตาราง

WHERE เงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ต้องการลบเร็คคอร์ดจากตาราง tblDvd โดยที่ค่าของฟิลด์ DvdTitle = 'The Butterfly Effect' จะได้

```
DELETE FROM tblDvd
WHERE DvdTitle = 'The Butterfly Effect'
```

2.9 การสร้างแอพพลิเคชันฐานข้อมูลด้วยคอมโพเนนต์

การสร้างแอพพลิเคชันฐานข้อมูลด้วยคอมโพเนนต์ จะเริ่มสร้างแอพพลิเคชันฐานข้อมูล โดยการใช้ Borland Database Engine (BDE) เป็นตัวกลางในการเข้าถึงฐานข้อมูล คอมโพเนนต์ที่ใช้เป็นองค์ประกอบในการสร้างแอพพลิเคชันฐานข้อมูลด้วยวิธีนี้ เป็นคอมโพเนนต์ในเพจ Data Access, เพจ BDE และเพจ Data Control โดยสามารถอธิบายความสัมพันธ์ระหว่างคอมโพเนนต์ต่างๆ และการติดต่อกับฐานข้อมูลได้ดังรูปที่ 2.11

รูปที่ 2.11 แสดงความสัมพันธ์ระหว่างคอมโพเนนต์ต่างๆ และการติดต่อกับฐานข้อมูล

จากรูปที่ 2.11 ส่วนที่ติดต่อกับฐานข้อมูลโดยตรงคือ Dataset ซึ่งก็คือคอมโพเนนต์ที่ทำหน้าที่ดึงกลุ่มของข้อมูลจากตารางในฐานข้อมูลตามที่เราต้องการ โดยคอมโพเนนต์ประเภทนี้ได้แก่

- คอมโพเนนต์ตาราง (Table)
- คอมโพเนนต์คิวรี (Query)
- คอมโพเนนต์ StoredProc
- คอมโพเนนต์ NestedTable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

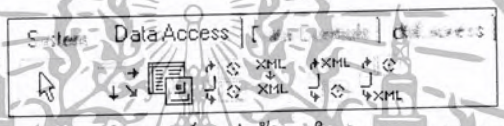
คอมโพเนนต์ทั้ง 4 ตัวดังกล่าวจะอยู่ในเพจ BDE แต่เนื่องจากคอมโพเนนต์ที่เป็น Dataset เป็นคอมโพเนนต์แบบ Non-visual จึงไม่สามารถแสดงข้อมูลจากฐานข้อมูลได้ ดังนั้นจึงต้องใช้คอมโพเนนต์ที่เรียกว่า Data-aware ในการแสดงผล

คอมโพเนนต์ที่อยู่ในเพจ Data Control ถือว่าเป็นคอมโพเนนต์แบบ Data-aware ที่สามารถนำข้อมูลจาก Dataset ออกมาแสดงผลได้ แต่การนำข้อมูลออกมาแสดงได้นั้น จำเป็นจะต้องมีตัวกลางในการเชื่อมต่อระหว่างคอมโพเนนต์จากทั้งสองกลุ่มด้วย นั่นก็คือคอมโพเนนต์แหล่งข้อมูล (DataSource) ซึ่งเป็นตัวกำหนดว่าจะนำข้อมูลจากที่ใดออกมาแสดงนั่นเอง

2.9.1 คอมโพเนนต์แหล่งข้อมูล (DataSource)

เป็นคอมโพเนนต์ที่ใช้เชื่อมความสัมพันธ์ระหว่างกลุ่มข้อมูล (DataSet) กับคอมโพเนนต์ ที่ใช้แสดงข้อมูล (Data-aware) โดยที่คอมโพเนนต์ทุกตัวในเพจ Data Control จะต้องทำการอ้างถึงคอมโพเนนต์นี้ เพื่อกำหนดแหล่งข้อมูลที่จะนำไปแสดงในคอมโพเนนต์นั้นๆ

คอมโพเนนต์แหล่งข้อมูลนี้เป็นคอมโพเนนต์ตัวแรกในเพจ Data Access แสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 แสดงคอมโพเนนต์แหล่งข้อมูลในเพจ Data Access

พร็อพเพอร์ตี้ที่สำคัญ

-AutoEdit ถ้ามีการติดต่อกับแหล่งข้อมูลแล้ว สถานะในการแก้ไขข้อมูลจะถูกกำหนดโดยอัตโนมัติ เมื่อกำหนดค่าพร็อพเพอร์ตี้นี้ให้เป็น True แต่ถ้าค่าเป็น False จะต้องเรียกใช้เมธอด Edit ก่อนถึงจะทำการแก้ไขข้อมูลนั้นได้

-DataSet กำหนด dataset ให้กับแหล่งข้อมูล ขณะที่ออกแบบจะสามารถเลือก dataset ได้จากรายการเลือกในออบเจ็กอินสเปกเตอร์ ส่วนในขณะรันสามารถเปลี่ยนค่าของ dataset ได้โดยการเขียนโค้ดขึ้นมา

-Enabled ใช้บอกว่าแหล่งข้อมูลของเราได้ทำการติดต่อกับ dataset หรือไม่ โดยที่ถ้ากำหนดค่าเป็น True แสดงว่าคอมโพเนนต์แหล่งข้อมูลได้มีการติดต่อกับ dataset แล้ว แต่ถ้าเรากำหนดค่าให้เป็น False แหล่งข้อมูลนี้ก็จะเสมือนกับว่าขาดการติดต่อกับ dataset และจะส่งผลถึงคอมโพเนนต์ต่างๆ ที่ใช้แหล่งข้อมูลนี้ด้วย

เมธอดและอีเวนต์ที่สำคัญ

- เมธอด IsLinkedTo มีรูปแบบการใช้งานดังนี้

```
Function IsLinkedTo (DataSet: TDataSet): Boolean;
```

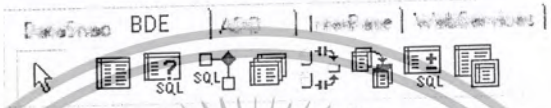
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราต้องการตรวจสอบว่าแหล่งข้อมูลมี dataset ตัวใดใช้งานอยู่เราจะใช้เมธอดนี้ โดยการส่งชื่อของ dataset ที่ต้องการตรวจสอบเข้าไป ถ้า dataset ที่ส่งเข้าไปนั้นใช้งานแหล่งข้อมูลนั้นอยู่ ก็จะส่งค่า True กลับมา แต่ถ้าไม่ได้ใช้จะส่งค่า False

- อีเวนต์ OnDataChange จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงของข้อมูลภายในเร็คคอร์ดปัจจุบัน หรือเกิดขึ้นเมื่อมีการเปลี่ยนแปลงตำแหน่งไปยังเร็คคอร์ดอื่นๆ

2.9.2 คอมโพเนนต์ต่างๆ ในเพจ BDE

คอมโพเนนต์ในเพจนี้เป็นคอมโพเนนต์ที่สามารถให้เราติดต่อกับข้อมูลภายในฐานข้อมูลผ่านทาง BDE ซึ่งแสดงได้ดังรูปที่ 2.13



รูปที่ 2.13 แสดงคอมโพเนนต์ในเพจ BDE

คอมโพเนนต์ตาราง (Table)

ใช้ในการเข้าถึงข้อมูลในตาราง และดึงกลุ่มของข้อมูลจากตารางในฐานข้อมูล โดยสามารถอ้างถึงตารางได้ 1 ตาราง

พร็อพเพอร์ตี้ที่สำคัญ

- Active ถ้าค่าเป็น True จะสามารถใช้งานข้อมูลในตารางได้ ในทางกลับกัน ถ้าค่าเป็น False ตารางเราจะไม่สามารถใช้งานข้อมูลในตารางนั้นได้
- BOF ถ้ามีค่าเป็น True แสดงว่าตัวชี้เร็คคอร์ดอยู่ที่ตำแหน่งแรกของตาราง มาจากคำว่า Begin of File
- DataSource กำหนดแหล่งข้อมูลของตารางที่เราต้องการใช้ในฐานข้อมูล
- EOF ถ้ามีค่าเป็น True แสดงว่าตัวชี้เร็คคอร์ดอยู่ที่ตำแหน่งสุดท้ายของตาราง มาจากคำว่า End of File
- FieldCount นับจำนวนฟิลด์ที่มีในตาราง
- FieldValue กำหนดค่าให้กับฟิลด์ของตาราง
- Filter กำหนดค่าเพื่อใช้ในการกรองข้อมูล
- Filtered ถ้าค่าของพร็อพเพอร์ตี้ เป็น True ข้อมูลจะถูกกรองตามเงื่อนไขที่กำหนดในพร็อพเพอร์ตี้ Filter แต่ถ้าเป็น False จะไม่มีการกรองข้อมูล
- TableName กำหนดชื่อตารางที่ต้องการนำมาใช้จากแหล่งข้อมูลที่กำหนด

เมธอดและอีเวนต์ที่สำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-เมธอด FieldByName เป็นเมธอดที่ใช้ในการดึงรายละเอียดของฟิลด์ โดยใช้ชื่อฟิลด์ในการอ้างอิงข้อมูลนั้น ซึ่งมีรูปแบบการใช้งานดังนี้

```
function FieldByName(const FieldName: string): Tfield;
```

โดยทั่วไปจะใช้เมธอดนี้ในการดึงข้อมูลหรือกำหนดค่าในฟิลด์นั้น ณ เร็คคอร์ดปัจจุบัน แสดงได้ดังตัวอย่าง

```
ตัวแปร := Table1. FieldByName('Name').AsString;   // นำค่าในฟิลด์ Name ไปใส่ตัวแปร
Table1. FieldByName('Age').AsInteger := 20;        // กำหนดค่า 20 ให้กับฟิลด์ Age
```

-AsString เป็นคุณสมบัติหนึ่งของออบเจ็กต์ TField ซึ่งเป็นออบเจ็กต์ชนิดหนึ่ง โดยเราสามารถกำหนดให้มีคุณสมบัติได้หลายรูปแบบขึ้นอยู่กับตัวแปรที่มารับค่า หรือค่าที่กำหนดไว้ให้ว่ามีชนิดเป็นอะไร จากตัวอย่างเราใช้คุณสมบัติ AsString เป็นการแปลงค่าของฟิลด์นั้นให้เป็น string ดังนั้นตัวแปรที่มารับ หรือค่าที่กำหนดให้ก็ต้องมีชนิดเป็น string ด้วย นอกจาก AsString แล้วยังมี AsInteger สำหรับจำนวนเต็ม AsFloat สำหรับข้อมูลประเภท float เป็นต้น

-เมธอด Locate เป็นเมธอดอีกตัวหนึ่งที่ใช้ในการค้นหาข้อมูลที่ต้องการ แต่ไม่จำเป็นที่ตารางนั้นจะต้องมีอินเด็กซ์ก็ได้ โดยมีรูปแบบดังนี้

```
function Locate(const KeyFields: String; const KeyValues: Variant; Options: TlocateOptions): Boolean;
```

ตัวแปร KeyFields ใช้กำหนดชื่อฟิลด์ที่เราต้องการค้นหา ส่วนค่าที่ต้องการค้นหาจะกำหนดผ่านตัวแปร KeyValues ส่วนตัวแปร Options เป็นตัวแปรที่ใช้กำหนดรูปแบบในการค้นหา โดยแบ่งเป็น 2 รูปแบบคือ

-loCaseInsensitive เป็นการกำหนดให้ค้นหาข้อมูล โดยไม่สนใจเรื่องของประเภทตัวอักษร ไม่ว่าจะเป็นตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็ก จะถือว่าเป็นตัวเดียวกัน

-loPartialKey ข้อมูลที่หาเจอจะเป็นข้อมูลที่มีบางส่วนของข้อมูลเหมือนกับค่าที่กำหนดไว้ เช่น ต้องการค้นหาข้อมูลที่มี 'BAT' ข้อมูลที่ค้นหาเจออาจจะเป็น 'BAT', 'DEBAT' หรือ 'PROBATION' เป็นต้น

การค้นหาข้อมูลโดยใช้เมธอด Locate นี้ เมื่อมีการค้นหาข้อมูลเจอ ตัวชี้เร็คคอร์ดจะชี้ไปยังเร็คคอร์ดที่ค้นหาเจอเป็นตัวแรก

เมธอดในการเลื่อนตัวชี้เร็คคอร์ดข้อมูล

ในการเลื่อนเร็คคอร์ดนั้น จะมีตัวชี้เร็คคอร์ดเป็นตัวกำหนดตำแหน่งของเร็คคอร์ด คอมโพเนนต์ตารางมีเมธอดในการเลื่อนตำแหน่งของเร็คคอร์ดอยู่ทั้งหมด 5 ตัวที่สำคัญ ดังนี้

-First เลื่อนไปยังเร็คคอร์ดแรกของตาราง

-Next เลื่อนไปยังเร็คคอร์ดถัดไป 1 เร็คคอร์ด

-Last เลื่อนไปยังเร็คคอร์ดสุดท้ายของตาราง

-Prior เลื่อนเร็คคอร์ดกลับไป 1 เร็คคอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-MoveBy เลื่อนเร็คคอร์ดไปตามจำนวนที่ส่งค่าเข้าไป เมธอดนี้มีรูปแบบดังนี้

function MoveBy(Distance: Integer): Integer;

ค่า Distance เป็นตัวเลขที่กำหนดจำนวนเร็คคอร์ดที่ต้องการเลื่อน โดยมีชนิดของข้อมูลเป็นจำนวนเต็ม การเลื่อนเร็คคอร์ดนี้สามารถเลื่อนได้ทั้งไปข้างหน้า และถอยหลัง การเลื่อนเร็คคอร์ดไปข้างหน้าค่า Distance ที่กำหนดให้ต้องเป็นเลขจำนวนบวก และถ้าต้องการให้เร็คคอร์ดเลื่อนถอยหลัง ค่า Distance จะต้องเป็นเลขจำนวนเต็มลบ

คอมโพเนนต์คิวรี (Query)

เป็นคอมโพเนนต์ที่ใช้เข้าถึงข้อมูลเช่นเดียวกับคอมโพเนนต์ตาราง แต่สามารถเข้าถึงข้อมูลได้ตั้งแต่ 1 ตารางขึ้นไป ด้วยการใช้คำสั่ง SQL ในการดึงข้อมูลที่ต้องการออกมา

พร็อพเพอร์ตี้ที่สำคัญ

- Active ถ้ากำหนดค่าเป็น True จะสามารถใช้งานข้อมูลจากคิวรีได้ แต่ถ้าเป็น False จะใช้งานไม่ได้
- DataSource กำหนดแหล่งข้อมูลที่เราต้องการค้นหาข้อมูลจากฐานข้อมูล
- Params เป็นการกำหนดค่าเพื่อที่จะส่งเข้าไปแทนที่ตัวแปรใน SQL เพื่อทำให้คำสั่ง SQL สมบูรณ์
- RecordCount นับเร็คคอร์ดที่ได้จากการดึงฐานข้อมูล
- SQL ใช้เก็บคำสั่ง SQL ที่ใช้ในการค้นหาข้อมูล โดยที่สามารถเก็บคำสั่ง SQL ได้เพียง 1 ชุดคำสั่งเท่านั้น นอกจากการเขียนคำสั่ง SQL เข้าไปเองแล้ว ยังสามารถใช้ SQL Builder ช่วยในการเขียนคำสั่งก็ได้

เมธอดที่สำคัญ

-ParamByName เราจะใช้เมธอดนี้ในการกำหนดค่าให้กับพารามิเตอร์ที่ถูกกำหนดขึ้นในคำสั่ง SQL

-ExecSQL เป็นเมธอดที่สั่งให้คำสั่ง SQL ในคิวรีทำงาน เราจะใช้เมธอดนี้ในกรณีที่ไม่มีการส่งค่ากลุ่มข้อมูลกลับมา เช่น การเพิ่มเร็คคอร์ด (INSERT), การลบเร็คคอร์ด (DELETE) หรือการแก้ไขข้อมูล (UPDATE) เป็นต้น ส่วนประโยคที่ใช้ SELECT เราจะเรียกใช้เมธอด Open แทน

คอมโพเนนต์ StoredProc

เป็นคอมโพเนนต์ที่ใช้เก็บ Stored Procedure ไว้สำหรับการใช้งานกับฐานข้อมูล, Stored Procedure ก็คือ โพรซีเจอร์ที่ประกอบด้วยคำสั่งในภาษา SQL ซึ่งมักจะเป็นงานที่ต้องทำกับฐานข้อมูลอยู่เป็นประจำ จึงทำการเขียนให้อยู่ในรูปโพรซีเจอร์เพื่อให้ง่ายต่อการเรียกใช้

คอมโพเนนต์ฐานข้อมูล (Database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้จัดการกับฐานข้อมูล มีประโยชน์ในการควบคุมการเข้ามาใช้ข้อมูลในฐานข้อมูล โดยสามารถกำหนดให้มีการป้อนชื่อผู้ใช้และรหัสผ่านได้ การเชื่อมต่อคอมโพเนนต์นี้กับฐานข้อมูลจะทำการเชื่อมต่อด้วยชื่อฐานข้อมูล หรืออาจเป็นเอเลียสที่สร้างไว้ก็ได้

พร็อพเพอร์ตี้ที่สำคัญ

- AliasName กำหนดเอเลียสที่ต้องการเชื่อมต่อกับฐานข้อมูล
- Connected ถ้ามีค่าเป็น True แสดงว่าคอมโพเนนต์ Database กำลังเชื่อมต่อกับฐานข้อมูลที่กำหนดอยู่ ในทางตรงกันข้าม ถ้าค่าเป็น False แสดงว่ายังไม่มีการติดต่อกับฐานข้อมูล
- DatabaseName กำหนดชื่อฐานข้อมูลที่ต้องการติดต่อ
- LoginPrompt กำหนดว่าจะให้แสดงไดอะล็อกบ็อกขึ้นมาถาม user name และรหัสผ่านหรือไม่ โดยถ้ามีค่าเป็น True ก็จะมีไดอะล็อกบ็อกซ์ขึ้นมาถามก่อน ถ้าเป็น False ก็จะไม่มี

คอมโพเนนต์ UpdateSQL

คอมโพเนนต์ UpdateSQL ใช้ในการเปลี่ยนแปลงคำสั่ง SQL ที่ใช้ในการคิวรีให้ทันสมัยอยู่เสมอ ซึ่งเราสามารถทำการลบ เพิ่ม และแก้ไขเปลี่ยนแปลงข้อมูลได้โดยใช้คอมโพเนนต์นี้เพียงตัวเดียวเท่านั้น

พร็อพเพอร์ตี้ที่สำคัญ

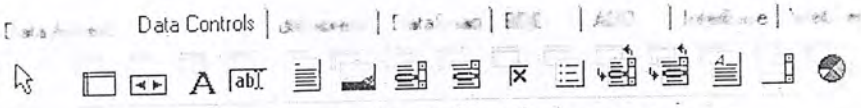
- DeleteSQL กำหนดประโยค SQL เพื่อใช้ในการลบข้อมูลด้วยคำสั่ง DELETE
- InsertSQL กำหนดประโยค SQL เพื่อใช้ในการเพิ่มข้อมูลด้วยคำสั่ง INSERT
- ModifySQL กำหนดประโยค SQL เพื่อใช้ในการเปลี่ยนแปลงข้อมูลด้วยคำสั่ง UPDATE

เมธอดที่สำคัญ

- Apply เป็นการสั่งให้ดำเนินการตามคำสั่ง SQL ที่กำหนด โดยคำสั่ง SQL ที่จะถูกดำเนินการจะขึ้นอยู่กับค่าที่เราส่งเข้าไปยังเมธอดนี้ ซึ่งมีอยู่ด้วยกัน 3 ค่า ดังนี้
  - 1.ukDelete รันคำสั่ง SQL ที่กำหนดในพร็อพเพอร์ตี้ DeleteSQL
  - 2.ukInsert รันคำสั่ง SQL ที่กำหนดในพร็อพเพอร์ตี้ InsertSQL
  - 3.ukModify รันคำสั่ง SQL ที่กำหนดในพร็อพเพอร์ตี้ ModifySQL

2.9.3 คอมโพเนนต์ต่างๆ ในเพจ Data Controls

คอมโพเนนต์ในเพจนี้ใช้แสดงข้อมูลจากฐานข้อมูล โดยข้อมูลที่นำมาแสดงนั้นได้มาจากคอมโพเนนต์ dataset จากเพจ Data Access เราจะไม่เห็นข้อมูลถ้าไม่มีคอมโพเนนต์ในเพจนี้



รูปที่ 2.14 แสดงคอมโพเนนต์ต่างๆ ในเพจ Data Controls

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโพเนนต์ DBGrid

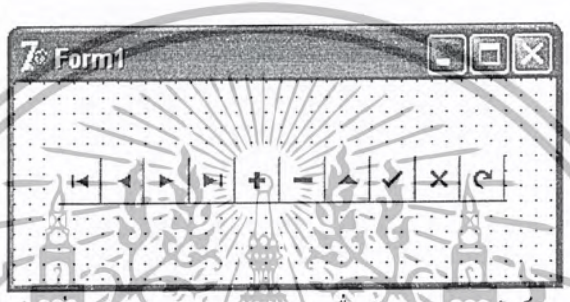
คอมโพเนนต์นี้ใช้แสดงข้อมูลในรูปแบบของตารางที่ประกอบด้วยคอลัมน์ และแถวของข้อมูลจากคอมโพเนนต์ dataset นอกจากนี้ยังสามารถทำการแก้ไขข้อมูลภายใน DBGrid ได้ด้วย

พร็อพเพอร์ตี้ที่สำคัญ

- DataSource กำหนดแหล่งข้อมูลที่ต้องการ
- Column กำหนดฟิลด์ที่ต้องการให้แสดงในกริดข้อมูล

คอมโพเนนต์ DBNavigator

เป็นคอมโพเนนต์ที่ใช้จัดการข้อมูลในตาราง ไม่ว่าจะเป็นการเลื่อนดูข้อมูล การเพิ่มข้อมูล หรือลบข้อมูล เป็นต้น เมื่อวางคอมโพเนนต์ DBNavigator ลงบนฟอร์ม จะมีรูปร่างดังนี้



รูปที่ 2.15 แสดง DBNavigator เมื่อนำมาวางบนฟอร์ม

จากรูปที่ 2.15 DBNavigator มีปุ่มต่างๆให้เลือกใช้งานอยู่เป็นจำนวนมากโดยแต่ละปุ่มมีหน้าที่ต่างกันไปดังนี้

- เลื่อนไปเร็คคอร์ดแรกของตาราง (เรียกใช้เมธอด First)
- เลื่อนไปเร็คคอร์ดก่อนหน้าเร็คคอร์ดปัจจุบัน (เรียกใช้เมธอด Prior)
- เลื่อนไปเร็คคอร์ดสุดท้ายของตาราง (เรียกใช้เมธอด Last)
- เพิ่มเร็คคอร์ดเข้าไปในตาราง (เรียกใช้เมธอด Insert)
- ลบ เร็คคอร์ดปัจจุบันออกจากตาราง (เรียกใช้เมธอด Delete)
- แก้ไขข้อมูลในตาราง (เรียกใช้เมธอด Edit)
- บันทึกการเปลี่ยนแปลงที่เกิดขึ้นกับตาราง (เรียกใช้เมธอด Post)
- ยกเลิกการเปลี่ยนแปลงที่เกิดขึ้นกับตาราง (เรียกใช้เมธอด Cancel)
- ทำให้ข้อมูลในตารางทันสมัยอยู่เสมอ (เรียกใช้เมธอด Refresh)

พร็อพเพอร์ตี้ที่สำคัญ

- ConfirmDelete ถ้ากำหนดค่าให้เป็น True เมื่อเราต้องการลบข้อมูลโดยใช้ปุ่ม Delete ของคอมโพเนนต์นี้ จะปรากฏไดอะล็อกบ็อกซ์ขึ้นมาเพื่อยืนยันการลบข้อมูลนั้น
- DataSource เป็นพร็อพเพอร์ตี้ที่สำคัญที่สุด เพราะเป็นแหล่งข้อมูลให้กับคอมโพเนนต์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Flat กำหนดลักษณะของปุ่ม ถ้ากำหนดค่าให้เป็น True เมื่อเราลากเมาส์ไปบนปุ่มจะทำให้ปุ่มนูนขึ้นมา
- VisibleButtons กำหนดปุ่มต่างๆ ว่าให้แสดงปุ่มอะไรบ้าง ถ้ากำหนดพร็อพเพอร์ตี้ให้เป็น True ที่ปุ่มใดปุ่มนั้นก็จะถูกแสดงออกมา แต่ถ้าเป็น False ปุ่มนั้นก็จะไม่แสดงให้เห็น

คอมโพเนนต์ DBText

การทำงานของ DBText นี้จะคล้ายกับการทำงานของคอมโพเนนต์ลาเบลในเพจมาตรฐาน แต่ DBText จะไม่มีพร็อพเพอร์ตี้ที่ใช้กำหนดข้อความให้กับคอมโพเนนต์ โดยข้อความที่แสดงนั้นจะเป็นข้อมูลที่ดึงมาจากฐานข้อมูล

พร็อพเพอร์ตี้ที่สำคัญ

- DataField กำหนดฟิลด์ของตารางหรือ dataset ที่เราต้องการนำข้อมูลมาแสดง
- DataSource กำหนดแหล่งข้อมูลที่ต้องการ

คอมโพเนนต์ DBEdit

การทำงานของ DBEdit จะคล้ายกับคอมโพเนนต์ Edit ในเพจมาตรฐาน แต่เป็นการแสดงข้อมูลที่ได้จากฐานข้อมูล รวมทั้งยังมีความสามารถในการแก้ไขข้อมูลในฐานข้อมูลได้ด้วย

พร็อพเพอร์ตี้ที่สำคัญ

- DataField กำหนดฟิลด์ของตารางหรือ dataset ที่เราต้องการนำข้อมูลมาแสดง
- DataSource กำหนดแหล่งข้อมูลที่ต้องการ

คอมโพเนนต์ DBMemo

ใช้แสดงข้อมูล และแก้ไขข้อมูลจำนวนหลายๆ บรรทัดจากฐานข้อมูล

พร็อพเพอร์ตี้ที่สำคัญ

- DataField กำหนดฟิลด์ของตารางหรือ dataset ที่เราต้องการนำข้อมูลมาแสดง
- DataSource กำหนดแหล่งข้อมูลที่ต้องการ

คอมโพเนนต์ DBImage

คอมโพเนนต์ DBImage ใช้ในการแสดง หรือเปลี่ยนแปลงฟิลด์ข้อมูลที่มีชนิดเป็นรูปภาพจากฐานข้อมูล คอมโพเนนต์นี้ช่วยเพิ่มความสามารถในการเก็บข้อมูลชนิดรูปภาพได้ดีทีเดียว

พร็อพเพอร์ตี้ที่สำคัญ

- DataField กำหนดฟิลด์ของตารางหรือ dataset ที่เราต้องการนำข้อมูลมาแสดง
- DataSource กำหนดแหล่งข้อมูลที่ต้องการ
- Stretch ถ้ากำหนดค่าเป็น True รูปที่แสดงออกมาจะกระจายเต็มพื้นที่ของคอมโพเนนต์ แต่ถ้าเป็นค่า False ขนาดของรูปจะเท่ากับขนาดจริง โดยไม่สนใจว่าคอมโพเนนต์จะมีขนาดเท่าใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 สร้างแอพพลิเคชันฐานข้อมูลด้วย ADO

ในการสร้างแอพพลิเคชันฐานข้อมูลทั่วๆ ไปใน Delphi จะต้องอาศัย Borland Database Engine (BDE) ในการติดต่อกับฐานข้อมูลเพื่อให้แอพพลิเคชันนั้นสามารถใช้งานได้ แต่เมื่อมีเทคโนโลยี ADO หรือ ActiveX Data Objects เข้ามา ก็ทำให้แอพพลิเคชันที่เราสร้างขึ้นโดยใช้คอมโพเนนต์ ADO สามารถติดต่อกับฐานข้อมูลได้โดยตรง โดยไม่ขึ้นอยู่กับ BDE ซึ่งเป็นประโยชน์มากในการพัฒนาแอพพลิเคชันฐานข้อมูลในปัจจุบันที่มีแหล่งข้อมูลหลากหลายรูปแบบ โดยเฉพาะอย่างยิ่งในอินเตอร์เน็ต

ADO เป็นเทคโนโลยีที่ใช้ในการติดต่อกับแหล่งข้อมูลของบริษัท Microsoft โดยจะรวมความสามารถในการเข้าถึงแหล่งข้อมูลในลักษณะที่เรียกว่า Universal Data Access นั่นคือ สามารถติดต่อกับฐานข้อมูลได้หลายรูปแบบ ไม่จำกัดว่าต้องเป็นตารางเพียงอย่างเดียว เช่น การเข้าถึงข้อมูลในอินเตอร์เน็ต อีเมล หรือเท็กซ์ไฟล์ เป็นต้น

การใช้งานแอพพลิเคชันฐานข้อมูลด้วยเทคโนโลยี ADO จะกระทำผ่าน OLE DB Provider ซึ่งเป็นรูปแบบ และวิธีการเข้าถึงแหล่งข้อมูลของเทคโนโลยี ADO ซึ่งได้เปรียบกับไดรเวอร์ที่ใช้ติดต่อกับแหล่งข้อมูลของ BDE โดย Delphi ได้รวมเทคโนโลยีนี้เข้าไว้กับคอมโพเนนต์ต่างๆ ในเพจ ADO ของคอมโพเนนต์พาเลท เพื่อให้สามารถสร้างแอพพลิเคชันฐานข้อมูลที่นำไปใช้งานได้โดยไม่ต้องใช้ BDE

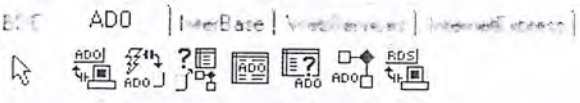
ในการสร้างแอพพลิเคชันฐานข้อมูลด้วยเทคโนโลยี ADO นั้น มีออบเจ็กต์ที่สำคัญอยู่ 3 ส่วน ดังนี้

1.Connection เป็นออบเจ็กต์ที่ใช้ในการติดต่อ ไปยังแหล่งข้อมูล โดยผ่านทาง connection string เมื่อเปรียบเทียบกับคอมโพเนนต์ BDE แล้วอาจกล่าวได้ว่าออบเจ็กต์ Connection เป็นการรวมคุณสมบัติของคอมโพเนนต์ Database และคอมโพเนนต์ Session เข้าไว้ด้วยกัน

2.Command ช่วยให้เราจัดการกับข้อมูลต่างๆ จากแหล่งข้อมูล โดยใช้คำสั่ง SQL เช่นเดียวกับการใช้คอมโพเนนต์ควิรี โดยคำสั่งต่างๆ เหล่านี้สามารถเพิ่ม ลบ แก้ไข หรือเรียกดูข้อมูลได้ตามที่เรากำหนด

3.Recordset เป็นออบเจ็กต์ที่ใช้แทนกลุ่มของข้อมูลที่ได้จากการทำตามคำสั่ง (Command) สามารถเปรียบเทียบได้กับคอมโพเนนต์ตาราง หรือควิรีจากคอมโพเนนต์ในเพจ BDE คอมโพเนนต์ต่างๆ ในเพจ ADO

เพื่อให้การสร้างแอพพลิเคชันฐานข้อมูลสนับสนุนการทำงานด้วยเทคโนโลยี ADO นั้น Delphi ได้เตรียมคอมโพเนนต์ไว้ให้ใช้งาน โดยเฉพาะ โดยคอมโพเนนต์เหล่านี้เก็บอยู่ในเพจ ADO ของคอมโพเนนต์พาเลท แสดงได้ดังรูปที่ 2.16



รูปที่ 2.16 แสดงคอมโพเนนต์ในเพจ ADO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมโพเนนต์แต่ละตัวจะทำหน้าที่ต่างกัน ไม่ว่าจะเป็น Connection, Command หรือ Recordset ซึ่งหน้าที่การทำงาน และคุณสมบัติของคอมโพเนนต์ที่สำคัญในเพจ ADO มีดังนี้

2.10.1 คอมโพเนนต์ ADOConnection

คอมโพเนนต์ ADOConnection เป็นออบเจ็กต์ Connection ทำหน้าที่ในการติดต่อกับแหล่งข้อมูล รวมทั้งควบคุมเงื่อนไข และสถานะของการติดต่อกับฐานข้อมูลนั้น เช่น กำหนดให้แสดงไดอะล็อกบ็อกซ์เพื่อรับข้อมูลของผู้ใช้ เป็นต้น

พร็อพเพอร์ตี้ที่สำคัญ

- Connected กำหนดการเชื่อมต่อระหว่างคอมโพเนนต์ ADOConnection กับแหล่งข้อมูล ถ้าค่าที่กำหนดเป็น True แสดงว่ามีการเชื่อมต่อกับแหล่งข้อมูลอยู่ แต่ถ้าเป็น False จะไม่มีการเชื่อมต่อเกิดขึ้น
- ConnectionString เก็บข้อมูลต่างๆ ที่จำเป็นในการเชื่อมต่อกับแหล่งข้อมูล
- LoginPrompt ใช้กำหนดว่าจะให้แสดงไดอะล็อกบ็อกซ์เพื่อรับข้อมูลชื่อผู้ใช้งาน และรหัสผ่านในการเชื่อมต่อกับแหล่งข้อมูลหรือไม่ ถ้าต้องการให้มีค่าที่กำหนดคือ True และในทางตรงกันข้ามกำหนดให้เป็น False

เมธอดที่สำคัญ

-Open ใช้เพื่อเปิดการเชื่อมต่อคอมโพเนนต์ ADOConnection กับแหล่งข้อมูล โดยเมื่อเรียกใช้เมธอดนี้ เท่ากับว่าเรากำหนดพร็อพเพอร์ตี้ Connected ให้เป็น True

-Close ใช้เพื่อปิดการเชื่อมต่อคอมโพเนนต์ ADOConnection กับแหล่งข้อมูล (ค่าของพร็อพเพอร์ตี้ Connected ให้เป็น False)

-GetFieldNames ใช้ดึงรายชื่อฟิลด์ทั้งหมดในตารางที่กำหนด โดยมีรูปแบบดังนี้

```
procedure GetFieldNames(const TableName: String; List: Tstrings);
```

-GetTableNames ใช้ดึงรายชื่อตารางทั้งหมดที่มีในแหล่งข้อมูลที่มีการติดต่อ มีรูปแบบดังนี้

```
procedure GetTableNames(List: Tstrings; SystemTables: Boolean = False);
```

2.10.2 คอมโพเนนต์ ADOCommand

เป็นคอมโพเนนต์ที่เก็บคำสั่งเพื่อใช้งานกับฐานข้อมูล และเนื่องจากคอมโพเนนต์นี้ไม่สามารถที่จะนำกลุ่มข้อมูลออกมาแสดงได้ จึงเหมาะกับงานที่ไม่ต้องการผลลัพธ์จากการทำตามคำสั่ง เช่น การเพิ่มข้อมูล (INSERT) หรือการลบข้อมูล (DELETE) การแก้ไขข้อมูล (UPDATE) เป็นต้น

พร็อพเพอร์ตี้ที่สำคัญ

- CommandText กำหนดคำสั่ง SQL ที่จะใช้งาน
- Connection กำหนดคอมโพเนนต์ที่ต้องการใช้ในการเชื่อมต่อข้อมูล
- ConnectionString กำหนดข้อมูลเพื่อใช้ในการเชื่อมต่อกับฐานข้อมูลโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Parameters กำหนดค่าที่ต้องการส่งเข้าไปยังคำสั่งเพื่อทำให้คำสั่งสมบูรณ์

เมธอดที่สำคัญ

-Execute รันคำสั่งที่กำหนดในพร็อพเพอร์ตี้ CommandText

-ParamByName กำหนดค่าให้กับพารามิเตอร์ใน CommandText เพื่อทำให้คำสั่งสมบูรณ์

2.10.3 คอมโพเนนต์ ADODataset

คอมโพเนนต์นี้จะมีลักษณะการใช้งานคล้ายคลึงกับคอมโพเนนต์ ADOCommand แต่ที่แตกต่างกันคือ คอมโพเนนต์นี้สามารถรองรับกลุ่มข้อมูลที่เกิดจากการรันคำสั่ง และแสดงกลุ่มข้อมูล รวมทั้งดำเนินการต่างๆ กับกลุ่มข้อมูลนั้นได้ เช่น เพิ่ม ลบ แก้ไข เป็นต้น

พร็อพเพอร์ตี้ที่สำคัญ

- CommandText กำหนดคำสั่ง SQL ที่จะใช้งาน
- Connection กำหนดคอมโพเนนต์ที่ต้องการใช้ในการเชื่อมต่อข้อมูล
- ConnectionString กำหนดข้อมูลเพื่อใช้ในการเชื่อมต่อกับฐานข้อมูลโดยตรง
- Parameters กำหนดค่าที่ต้องการส่งเข้าไปยังคำสั่งเพื่อทำให้คำสั่งสมบูรณ์
- RecordCount จำนวนเร็คคอร์ดที่มีในคอมโพเนนต์

เมธอดที่สำคัญ

-Open เปิดกลุ่มข้อมูลที่ได้จากการรันคำสั่ง SQL เพื่อนำมาใช้งาน

-ParamByName กำหนดค่าให้กับพารามิเตอร์ใน CommandText เพื่อทำให้คำสั่งสมบูรณ์

2.10.4 คอมโพเนนต์ ADOTable

ใช้เข้าถึงข้อมูลในตารางจากแหล่งข้อมูลที่มีการติดต่อ โดยจะเข้าถึงข้อมูลในตารางได้ 1 ตาราง เท่านั้นในขณะใดขณะหนึ่ง

พร็อพเพอร์ตี้ที่สำคัญ

- Active ถ้าค่าเป็น True จะสามารถใช้งานข้อมูลในตารางได้ ในทางกลับกัน ถ้าค่าเป็น False ตารางเราจะไม่สามารถใช้งานข้อมูลในตารางนั้นได้
- BOF ถ้ามีค่าเป็น True แสดงว่าตัวชี้เร็คคอร์ดอยู่ที่ตำแหน่งแรกของตาราง มาจากคำว่า Begin of File
- DataSource กำหนดแหล่งข้อมูลของตารางที่เราต้องการใช้ในฐานข้อมูล
- EOF ถ้ามีค่าเป็น True แสดงว่าตัวชี้เร็คคอร์ดอยู่ที่ตำแหน่งสุดท้ายของตาราง มาจากคำว่า End of File
- FieldCount นับจำนวนฟิลด์ที่มีในตาราง
- FieldValue กำหนดค่าให้กับฟิลด์ของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Filter กำหนดค่าเพื่อใช้ในการกรองข้อมูล
- Filtered ถ้าค่าของพร็อพเพอร์ตี้ เป็น True ข้อมูลจะถูกกรองตามเงื่อนไขที่กำหนดในพร็อพเพอร์ตี้ Filter แต่ถ้าเป็น False จะไม่มีการกรองข้อมูล
- TableName กำหนดชื่อตารางที่ต้องการนำมาใช้จากแหล่งข้อมูลที่กำหนด

เมธอดที่สำคัญ

-Close ปิดการติดต่อกับตารางจากแหล่งข้อมูล

-Open เปิดการติดต่อกับตารางจากแหล่งข้อมูล ตามที่กำหนด

-FieldByName เป็นเมธอดที่ใช้ในการดึงรายละเอียดของฟิลด์ โดยใช้ชื่อฟิลด์ในการอ้างอิงข้อมูล

-FindKey ใช้ในการค้นหาข้อมูลจากตารางที่มีการสร้างอินเด็กซ์ไว้โดยการกำหนดค่าที่ต้องการหา

-FindNearest เมธอดนี้จะคล้ายๆ กับเมธอด FindKey แต่ถ้าค้นหาข้อมูลไม่พบ ตัวชี้จะชี้ไปยังเร็คคอร์ดแรกของข้อมูลที่มีค่ามากกว่าค่าที่กำหนด

-Locate เป็นเมธอดที่ใช้ในการค้นหาข้อมูลที่ต้องการ แต่ไม่จำเป็นที่ตารางนั้นต้องมีอินเด็กซ์ก็ได้

2.10.5 คอมโพเนนต์ ADOQuery

ใช้เข้าถึงข้อมูลในตาราง จากแหล่งข้อมูลที่ทำการติดต่อผ่านพร็อพเพอร์ตี้ Connection หรือ ConnectionString โดยเราสามารถเข้าถึงข้อมูลได้ตั้งแต่ 1 ตารางขึ้นไป การเข้าถึงข้อมูล และการจัดการกับข้อมูลในตารางโดยใช้ คอมโพเนนต์ ADOQuery นี้ สามารถทำได้โดยการรันคำสั่ง SQL ที่กำหนดขึ้นมา

พร็อพเพอร์ตี้ที่สำคัญ

- Active ถ้ากำหนดค่าเป็น True จะสามารถใช้งานข้อมูลจากคิวรีได้ แต่ถ้าเป็น False จะใช้งานไม่ได้
- Connection กำหนดคอมโพเนนต์ที่ต้องการใช้ในการเชื่อมต่อกับฐานข้อมูล
- ConnectionString กำหนดข้อมูลเพื่อใช้ในการเชื่อมต่อกับฐานข้อมูลโดยตรง
- RecordCount นับเร็คคอร์ดที่ได้จากการดึงฐานข้อมูล
- SQL ใช้เก็บคำสั่ง SQL ที่ใช้ในการค้นหาข้อมูล โดยที่สามารถเก็บคำสั่ง SQL ได้เพียง 1 ชุดคำสั่งเท่านั้น นอกจากการเขียนคำสั่ง SQL เข้าไปเองแล้ว ยังสามารถใช้ SQL Builder ช่วยในการเขียนคำสั่งก็ได้

เมธอดที่สำคัญ

-ExecSQL ทำงานตามคำสั่ง SQL โดยเมธอดนี้จะใช้ในกรณีที่ไม่มีการส่งค่ากลุ่มข้อมูลกลับมา เช่น การเพิ่มเร็คคอร์ด (INSERT), การลบเร็คคอร์ด (DELETE) หรือการแก้ไขข้อมูล (UPDATE) เป็นต้น ค่าที่ส่งกลับมาเมื่อเรียกใช้เมธอดนี้คือ จำนวนเร็คคอร์ดที่เป็นไปตามเงื่อนไขที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Open ทำงานตามคำสั่ง SQL เช่นเดียวกับ ExecSQL แต่จะใช้กับคำสั่งในการดึงข้อมูล (SELECT)

-ParamByName ใช้เมธอดนี้ในการกำหนดค่าให้กับพารามิเตอร์ที่ถูกกำหนดขึ้นในคำสั่ง SQL

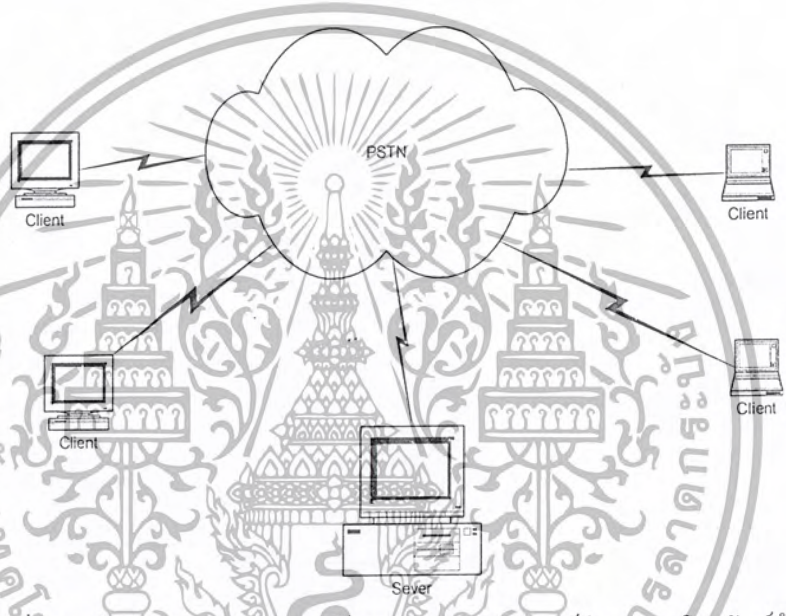
2.10.6 คอมโพเนนต์ ADOStoredProc

ใช้เข้าถึง Stored Procedure ที่ถูกจัดเก็บไว้ในแหล่งข้อมูลที่ทำการติดต่อด้วยพร็อพเพอร์ตี้ ConnectionString หรือทำการอ้างอิงการติดต่อจากคอมโพเนนต์ ADOConnection ผ่านทางพร็อพเพอร์ตี้ Connection ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวนและการสร้าง

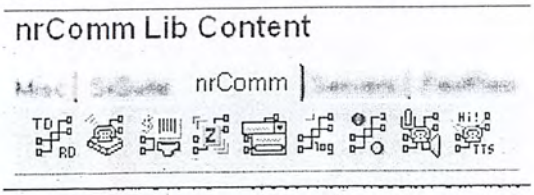
โปรแกรมที่สร้างขึ้นจากโครงการนี้ สร้างขึ้นโดยใช้เดลไฟ 7.0 ซึ่งเป็นเครื่องมือสำหรับโปรแกรมเพื่อสร้างแอพพลิเคชั่นบน Windows โดยภาษาที่ใช้ในการเขียนเป็นภาษาออบเจ็กต์ปาสคาล ซึ่งโปรแกรม จะถูกแบ่งออกเป็นสองส่วนใหญ่ๆ คือ Client กับ Server โดยในบทนี้ จะอธิบายหน้าที่การทำงาน ส่วนต่างๆ ของ Client และ Server ตลอดจน คอมโพเนนต์ที่เป็นโปรแกรมย่อย ที่ช่วยในการติดต่อกับโมเด็ม และส่งข้อมูลผ่านโมเด็มในรูปที่ 3.1 แสดงบล็อคไดอะแกรมของระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน



รูปที่ 3.1 บล็อคไดอะแกรมระบบส่งข้อความบนคอมพิวเตอร์ผ่านระบบโทรศัพท์บ้าน

3.1 คอมโพเนนต์ nrComm

คอมโพเนนต์ nrComm เป็นคอมโพเนนต์ที่ช่วยในการติดต่อสื่อสารกับโมเด็ม และส่งข้อมูลผ่านโมเด็ม โดยยูนิตที่เรานำมาใช้งานมี 2 ยูนิตคือ ยูนิต TnrCommTAPI และ ยูนิต TnrZModem โดยในรูปที่ 3.2 แสดงคอมโพเนนต์ nrComm



รูปที่ 3.2 แสดงคอมโพเนนต์ของ nrComm

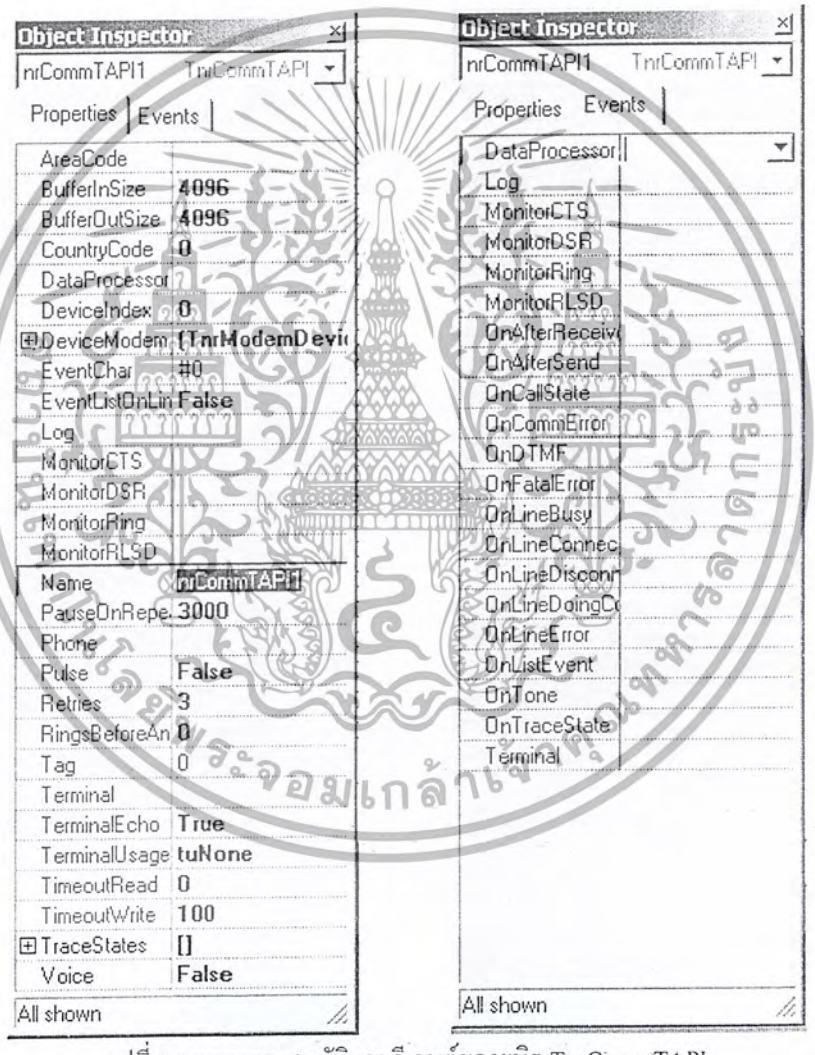
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ยูนิต TnrCommTAPI

ยูนิต TnrCommTAPI ใช้เชื่อมต่อกับ อุปกรณ์สื่อสารที่ถูกติดตั้งอยู่ในระบบปฏิบัติการ 32 บิต (WIN 32) โดยเราจะใช้ยูนิต TnrCommTAPI ในการให้โมเด็มส่งสัญญาณเรียก รอสัญญาณเรียกเข้า และตอบรับสัญญาณเรียกเข้า รูปที่ 3.3 แสดงรูปยูนิต TnrCommTAPI รูปที่ 3.4 แสดงคุณสมบัติและอีเวนต์ของยูนิต TnrCommTAPI

TnrCommTAPI i

รูปที่ 3.3 แสดงรูปยูนิต TnrCommTAPI



รูปที่ 3.4 แสดงคุณสมบัติและอีเวนต์ของยูนิต TnrCommTAPI

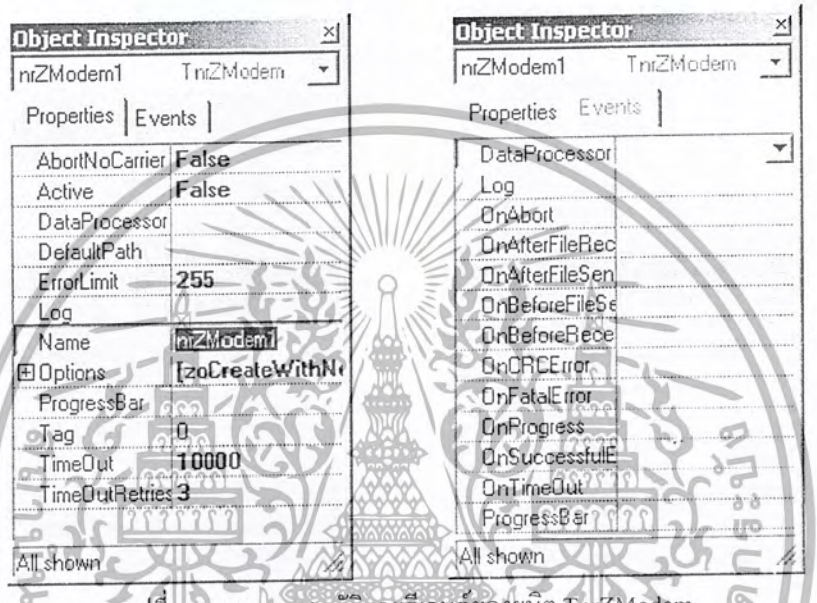
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ยูนิต TnrZModem

ยูนิต TnrZModem เป็นยูนิตที่สนับสนุนการส่งไฟล์ โดยใช้โปรโตคอล ZMODEM โดยในรูปที่ 3.5 แสดงรูปยูนิตTnrZModem และรูปที่ 3.6 แสดงคุณสมบัติและอีเวนต์ของยูนิต TnrZModem

TnrZModem

รูปที่ 3.5 แสดงรูปยูนิตTnrZModem



รูปที่ 3.6 แสดงคุณสมบัติและอีเวนต์ของยูนิต TnrZModem

3.2 การทำงานในส่วนของ Client

Client จะประกอบไปด้วย หน้าโปรแกรมหลัก คือ

- Configulation ที่ทำหน้าที่กำหนดอุปกรณ์ โมเด็มและ ค่าพารามิเตอร์ ต่างๆ ในการสื่อสาร ตลอดจนเบอร์โทรศัพท์ที่ใช้ในการติดต่อกับ Server

- Contact ทำหน้าที่เป็นส่วนบันทึก แก้ไข และแสดงชื่อ และเบอร์โทรศัพท์ ของผู้ที่เราต้องการติดต่อลงในลิสต์เพื่อนำไปใช้ในการส่งข้อความได้

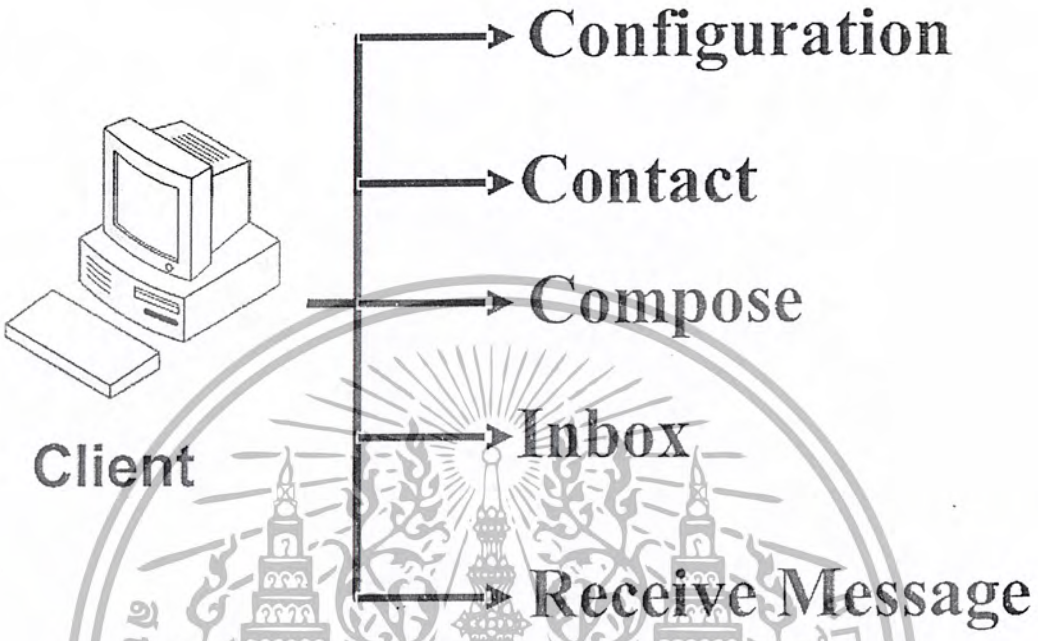
- Compose เป็นส่วนที่สามารถ เขียนข้อความที่ต้องต้องการส่ง และข้อมูลของผู้รับปลายทาง พร้อมทั้งสามารถ เลือกไฟล์ที่ต้องการแนบไปในการส่งได้

- Inbox สำหรับในหน้า Inbox จะปรากฏ กล่องรับข้อความที่จะแสดงจำนวนข้อความที่รับเข้ามาได้ทั้งหมดและสามารถเลือกเพื่อเปิดอ่านได้ เมื่อทำการเปิดอ่านแล้วสามารถที่จะลบเร็คคอร์ดนั้น หรือตอบจดหมายนั้นได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และนอกจากนี้ยังประกอบด้วยส่วน Receive Message ที่ทำหน้าที่ตอบรับการส่งสัญญาณเรียกที่มาจาก Server และทำหน้าที่รับข้อความและไฟล์ที่ได้แนบมาด้วย

โดยในรูปที่ 3.7 เป็นการแสดงบล็อคไดอะแกรมแสดงส่วนประกอบของโปรแกรมที่ใช้ในส่วนของClient



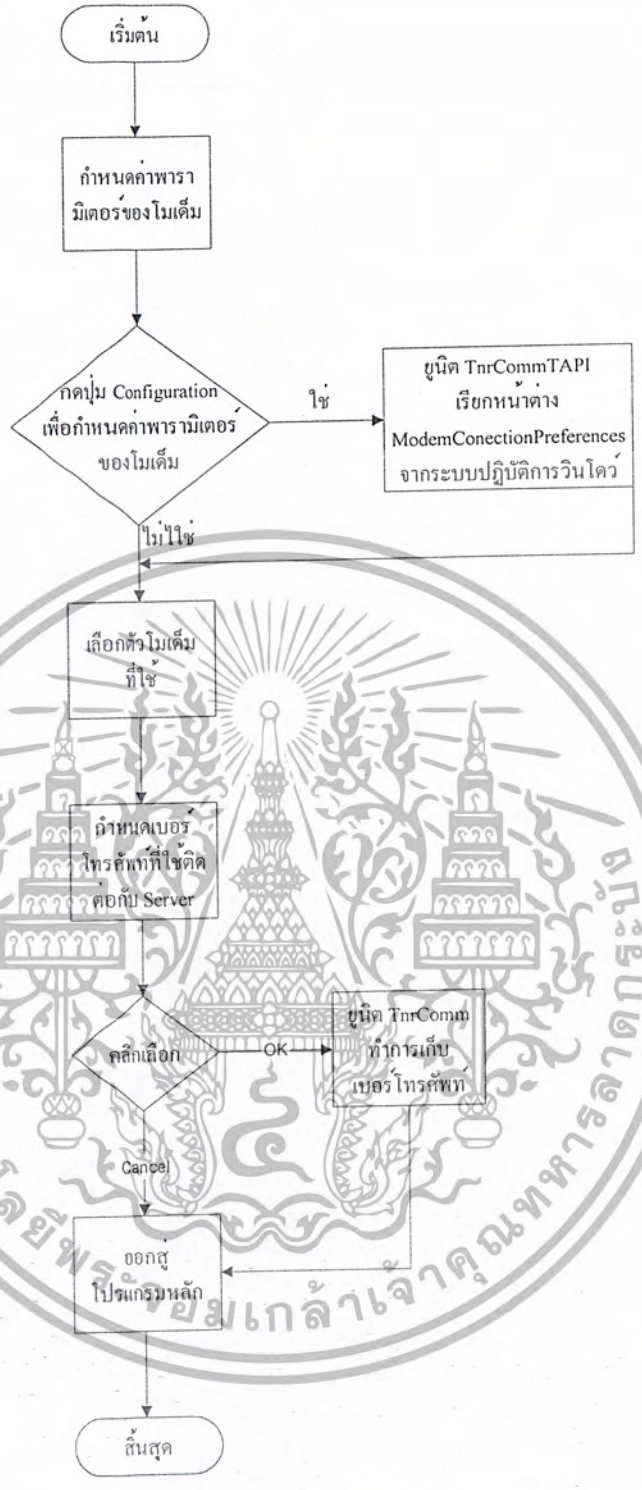
รูปที่ 3.7 ไดอะแกรมส่วนประกอบของโปรแกรมในส่วนของ Client

โดย ในรายระเอียดการทำงานของโปรแกรมต่างๆ สามารถอธิบายได้ดังต่อไปนี้

3.2.1 โปรแกรมกำหนดรูปแบบการติดต่อ (Configuration)

ทำหน้าที่กำหนดพารามิเตอร์ในการเชื่อมต่อของโมเด็ม และกำหนดเบอร์โทรศัพท์ที่ใช้ในการติดต่อระหว่าง Client กับ Server โดยที่ Client จำเป็นจะต้องทำการกำหนดก่อนการติดต่อกับ Server โดยในการกำหนดค่าพารามิเตอร์ต่างๆ นั้น ตัวโปรแกรมจะไปเรียก ยูนิต TnrCommTAPI ให้ทำหน้าที่เรียกหน้า Modem Conection Preferences จากระบบปฏิบัติการวินโดว์ออกมา ในรูปที่ 3.8 แสดงแผนผังการทำงานของโปรแกรมกำหนดรูปแบบการติดต่อ

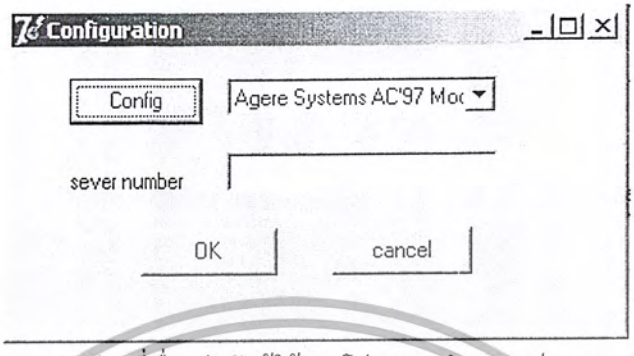
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



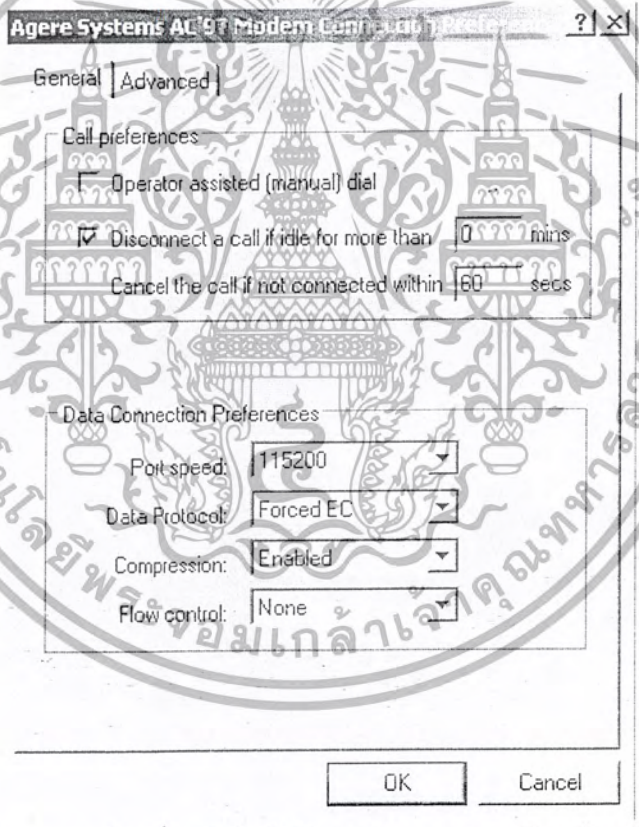
รูปที่ 3.8 แสดงแผนผังแสดงการทำงานของโปรแกรมกำหนดค่าพารามิเตอร์ของโมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.9 จะแสดงส่วนเชื่อมต่อกับผู้ใช้ของโปรแกรมกำหนดรูปแบบการติดต่อ และในรูปที่ 3.10 เป็นรูปหน้าต่าง Modem Connection Preferences ที่ยูนิต TnrCommTAPI เรียกขึ้นมา เพื่อทำการกำหนดค่าพารามิเตอร์ในการสื่อสารของโมเด็ม



รูปที่ 3.9 แสดงส่วนที่เชื่อมต่อกับผู้ใช้ของโปรแกรมกำหนดรูปแบบการติดต่อ



รูปที่ 3.10 แสดงส่วนเชื่อมต่อกับผู้ใช้ในการกำหนดค่าพารามิเตอร์ของโมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2โปรแกรมในหน้า Contact

จะประกอบไปด้วยส่วนต่างๆ ดังนี้

-Contact โดยจะทำหน้าที่แสดง รายชื่อ และเบอร์โทรศัพท์ ที่ถูกบันทึกจากส่วน Add Contact ส่วนที่ถูกแก้ไขจาก Modify Contact Data และ Delete Contact Data จากตารางฐานข้อมูล

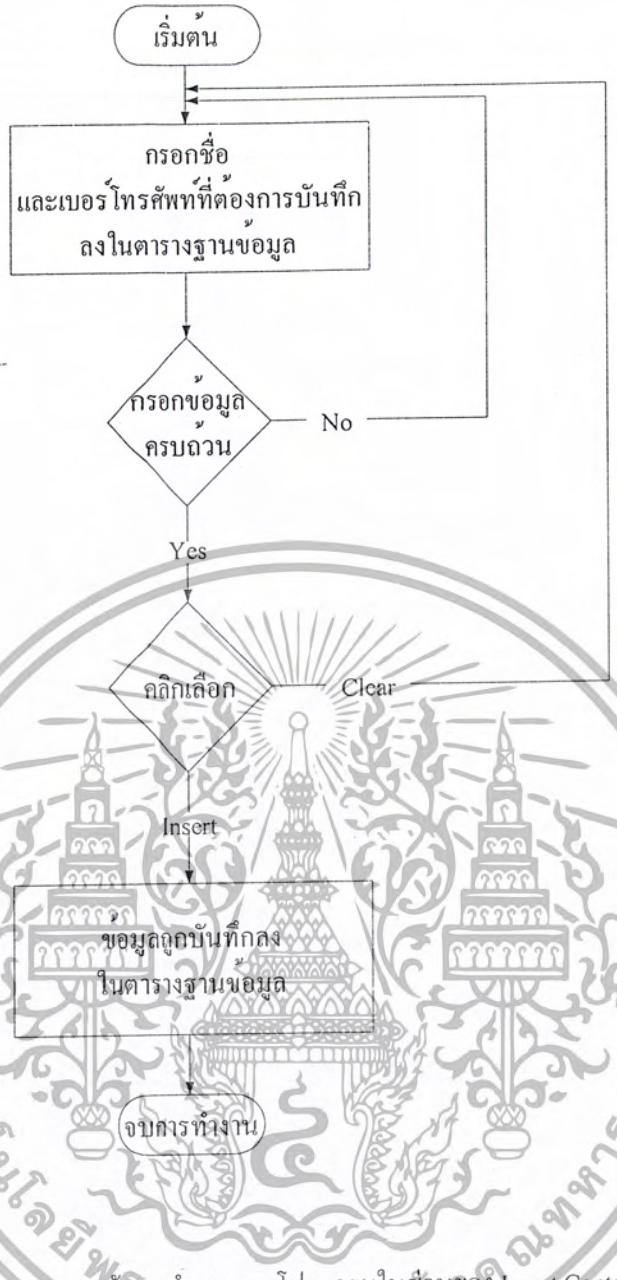
-Add Contact โดยจะทำหน้าที่รับการบันทึกรายชื่อและเบอร์โทรศัพท์ของผู้รับทางด้านปลายทาง เมื่อผู้ใช้กด Insert โปรแกรมก็จะสั่งให้มีการบันทึกรายชื่อ และเบอร์โทรศัพท์ ลงในตารางฐานข้อมูลด้วย Insert SQL แล้วนำไปแสดงอยู่ในส่วนของ Contact

-Delete Contact Data ส่วนนี้จะทำหน้าที่ในการลบรายชื่อและเบอร์โทรศัพท์ที่ได้ทำการบันทึกไว้ก่อนหน้านี้ (แสดงอยู่ในส่วนของ Contact) ออกจากตารางฐานข้อมูล ถ้าผู้ใช้ต้องการที่จะลบข้อมูลจะปรากฏ Dialog box ขึ้นมาถามก่อนว่า “คุณต้องการที่จะลบเร็คคอร์ดนี้ใช่หรือไม่” ถ้าคลิก “Yes” โปรแกรมก็จะส่งโดยใช้ Delete SQL ในการจัดการกับตารางข้อมูลในฐานข้อมูล แต่ถ้าคลิก “No” โปรแกรมจะกลับมายังหน้าเดิมไม่มีอะไรเปลี่ยนแปลง

-Modify Contact Data จะทำหน้าที่ในการแก้ไขข้อมูลรายชื่อและเบอร์โทรศัพท์ ที่ถูกเก็บอยู่ในตารางฐานข้อมูล โดยใช้ Modify SQL ในการจัดการกับตารางข้อมูลในฐานข้อมูล ผลหลังจากการแก้ไขจะถูกแสดง อยู่ในลิสต์ ในส่วนของ Contact

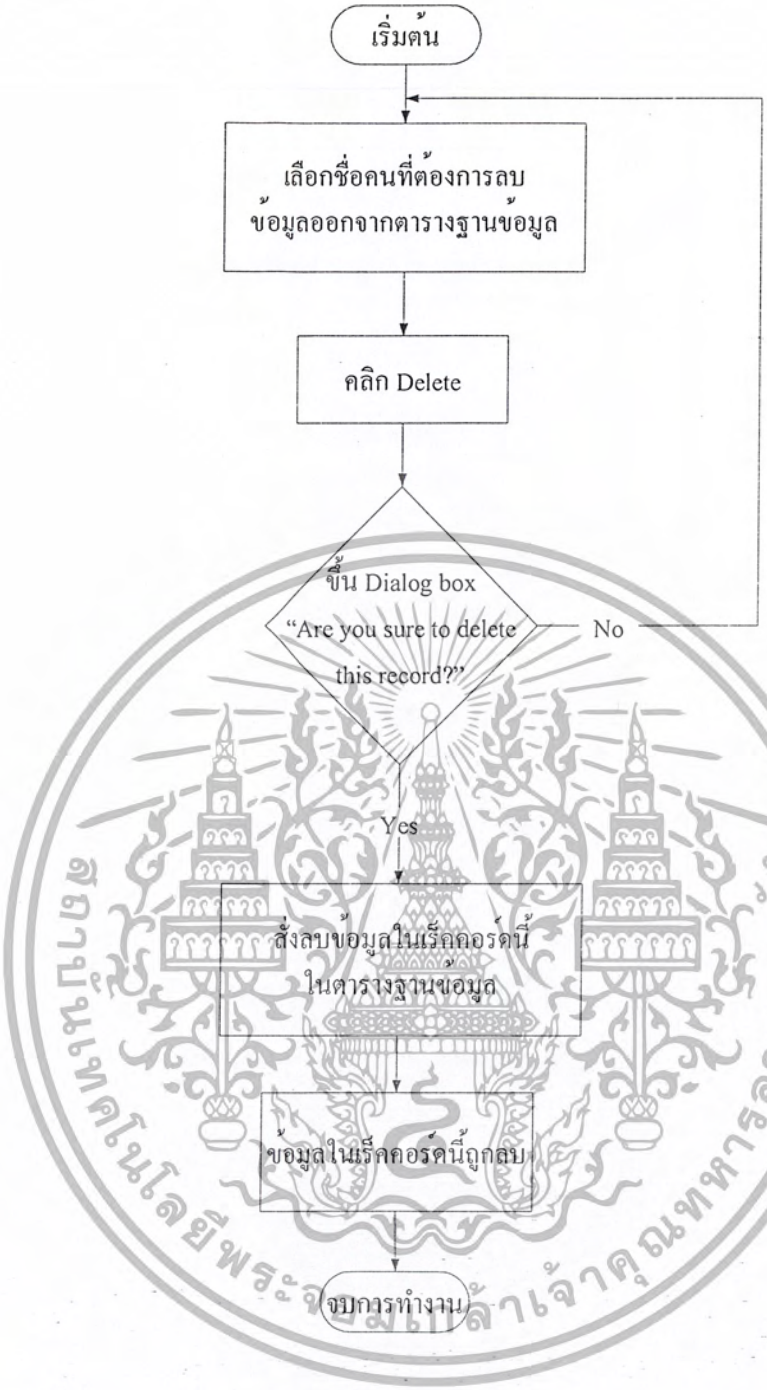
โดยมีผังการทำงานของโปรแกรมในหน้า Contact และส่วนเชื่อมต่อกับผู้ใช้ในหน้า Contact ซึ่งจะประกอบไปด้วย 3 ส่วนด้วยกัน คือส่วนของ Insert Contact Data, Delete Contact Data และ Modify Contact Data ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



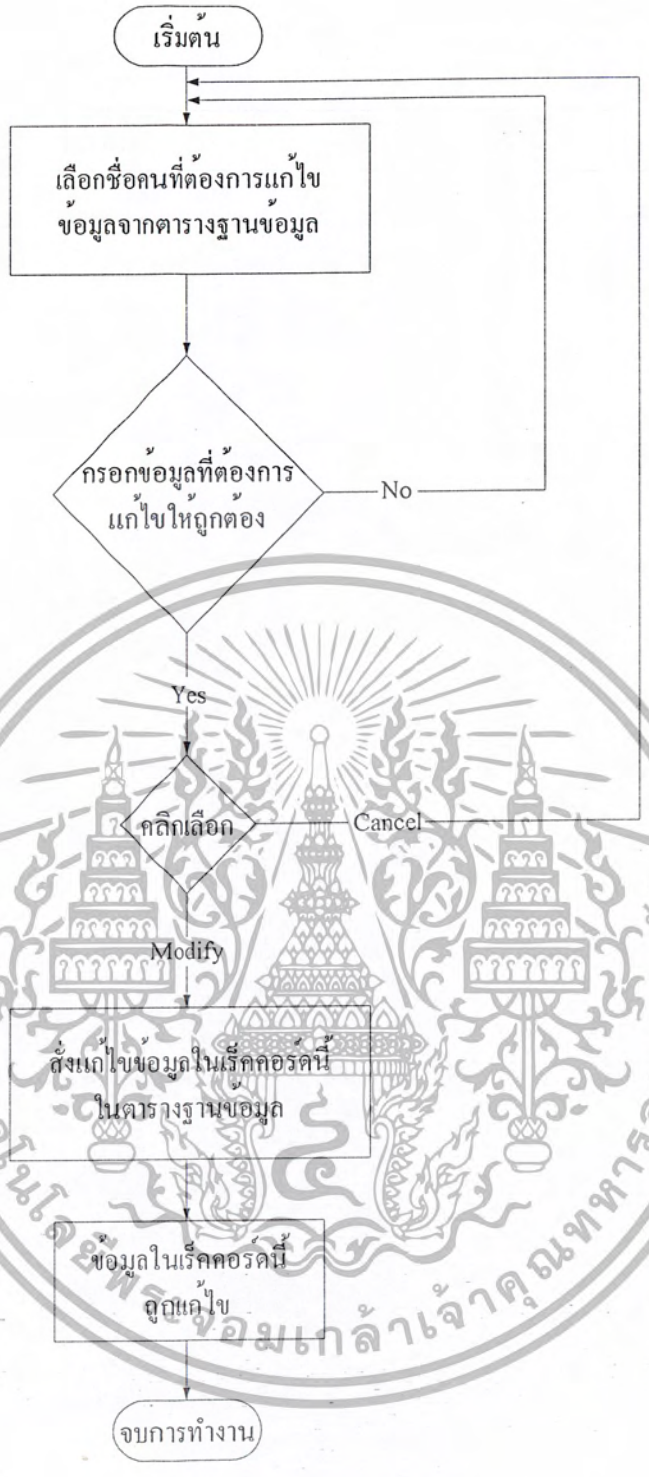
รูปที่ 3.11 แสดงผังการทำงานของโปรแกรมในส่วนของ Insert Contact Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



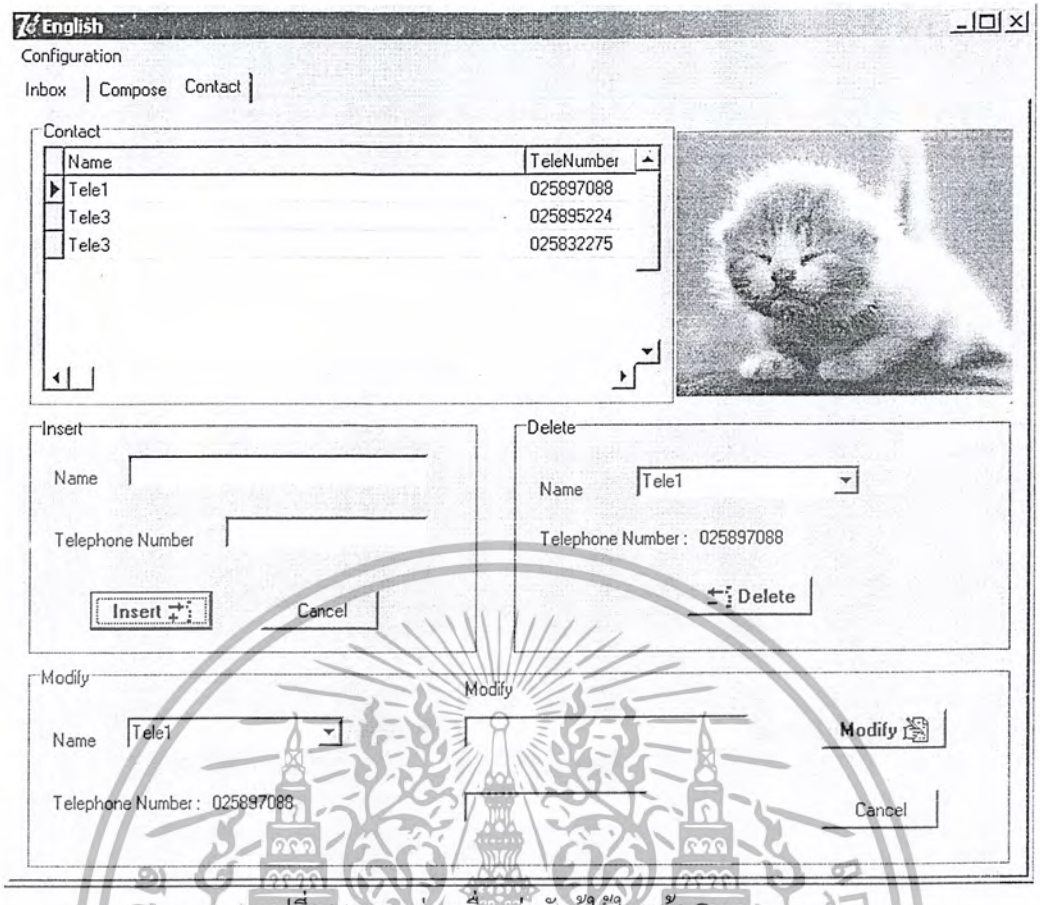
รูปที่ 3.12 แสดงผังการทำงานของโปรแกรมในส่วนของ Delete Contact Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงผังการทำงานของโปรแกรมในส่วนของ Modify Contact Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.14 แสดงส่วนเชื่อมต่อกับผู้ใช้ในหน้า Contact

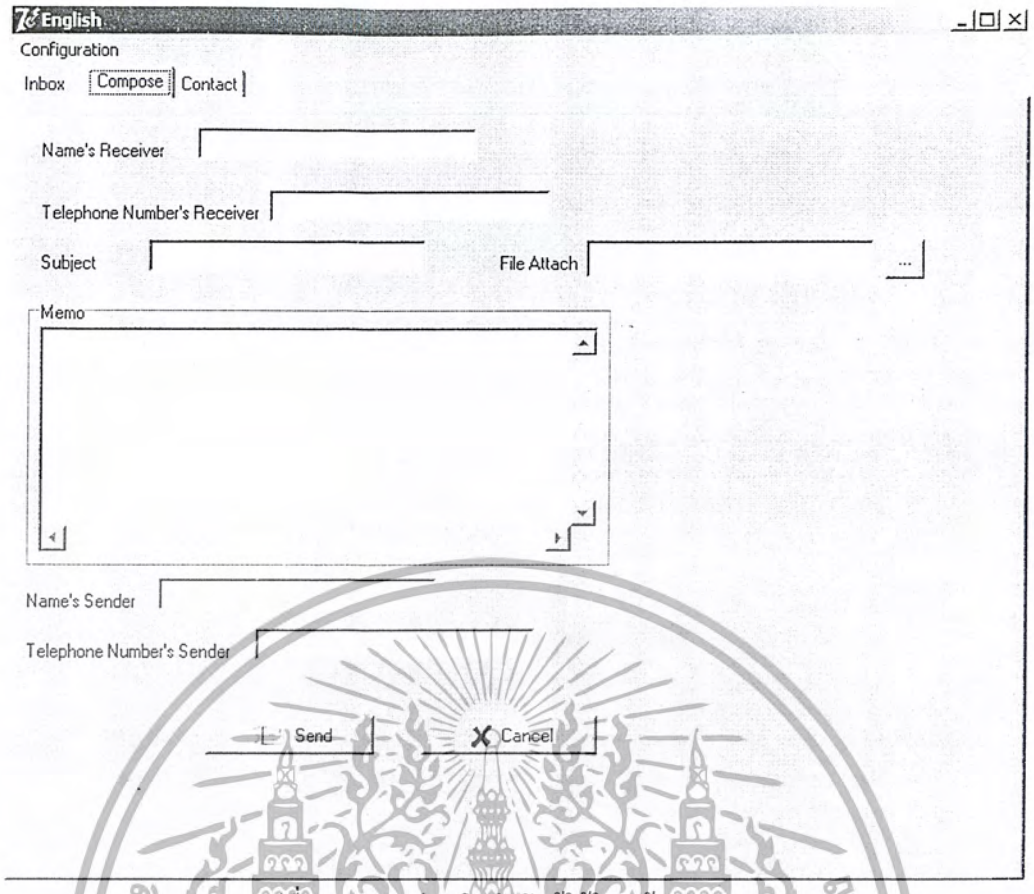
3.2.3 โปรแกรมในหน้า Compose

ในส่วนของ Compose จะประกอบไปด้วย พื้นที่เขียนข้อความที่ต้องการส่ง ที่กรอกเบอร์โทรศัพท์ทั้งของผู้รับและผู้ส่งข้อความ ชื่อของผู้ของผู้ส่งและผู้รับข้อความ หัวเรื่องของข้อความ และ ไฟล์ที่ต้องการแนบ รวมถึงส่วนของ List Name ที่จะแสดงรายชื่อที่ได้ทำการบันทึกไว้ในส่วน Contact โดยสามารถเลือกชื่อผู้รับจดหมายได้โดยไม่ต้องพิมพ์ โดยเมื่อผู้ส่งคลิก Send เพื่อที่ต้องการส่งข้อความ โปรแกรมจะทำการสร้างไฟล์ข้อความขึ้นมา และจะแสดงฟอร์ม เพื่อยืนยันการส่งข้อความและไฟล์ที่แนบว่าถูกต้องหรือไม่ ถ้าถูกต้องไฟล์ข้อความและไฟล์แนบจะถูกส่งไปยัง Server ต่อไป โดยในรูปที่ 3.15 แสดงแผนผังการทำงานของโปรแกรมในหน้า Compose และรูป 3.16 แสดงส่วนติดต่อกับผู้ใช้ในหน้า Compose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

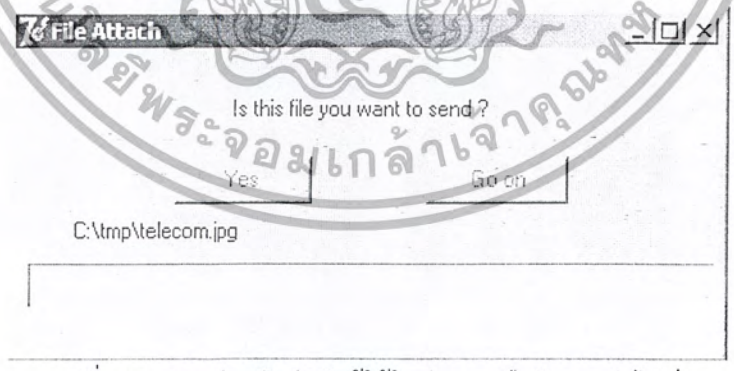
รูปที่ 3.15 แสดงแผนผังการทำงานของโปรแกรมในหน้า Compose

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงส่วนติดต่อกับผู้ใช้ในหน้า Compose

และเมื่อผู้ส่งคลิก Send เพื่อยืนยันการส่งข้อความนี้ไปยังปลายทาง โปรแกรมจะทำการสร้างไฟล์ข้อความขึ้นมาหลังจากนั้น โปรแกรมทำการเรียกยูนิต TmrCommTAPI เพื่อสั่งให้โมเด็มส่งสัญญาณเรียกไปยัง Server และปรากฏฟอร์มถามว่าไฟล์ที่แนบถูกต้องและยืนยันการส่งข้อความ ดังรูปที่ 3.17

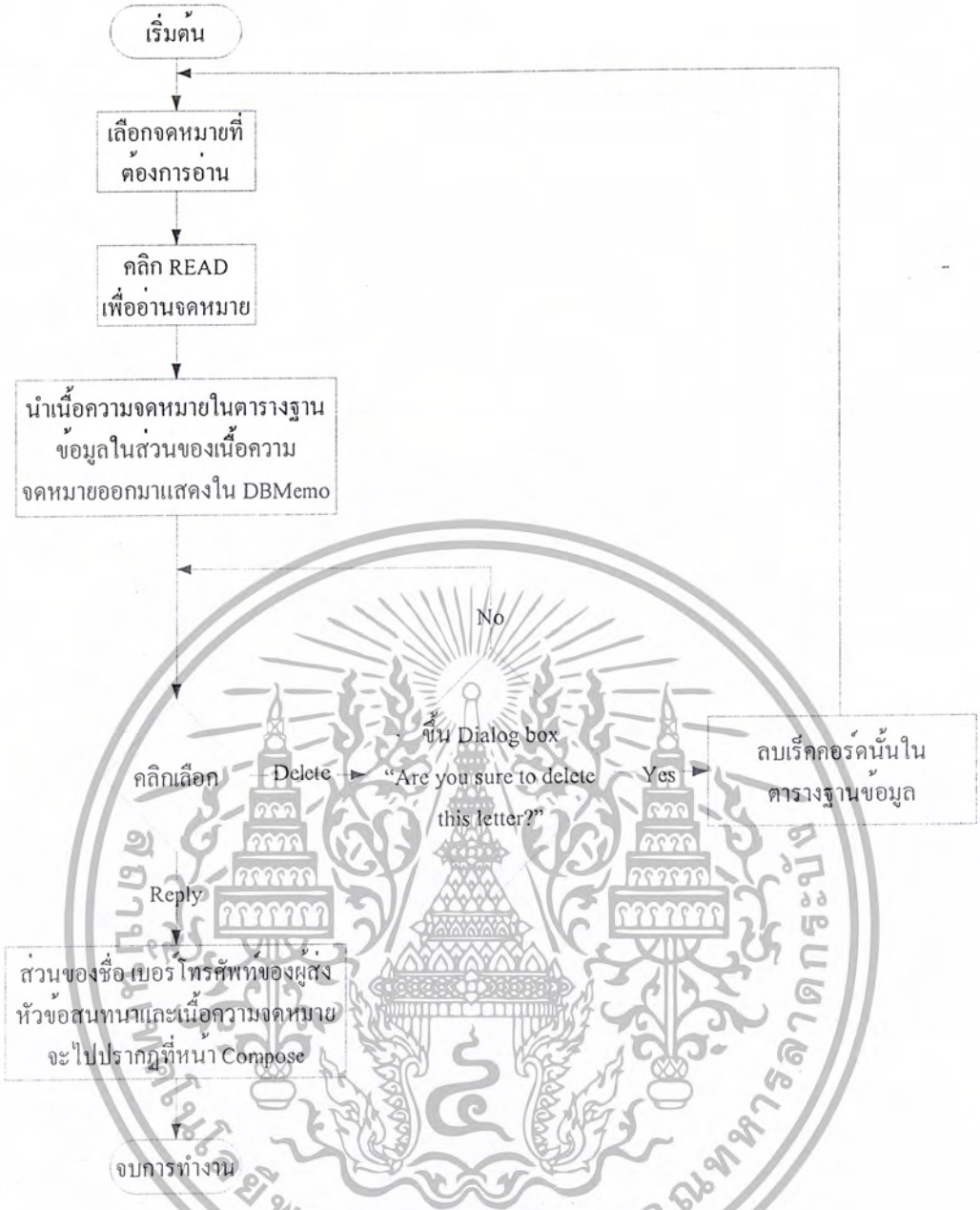


รูปที่ 3.17 แสดงส่วนติดต่อกับผู้ใช้ในส่วนการยืนยันการส่งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

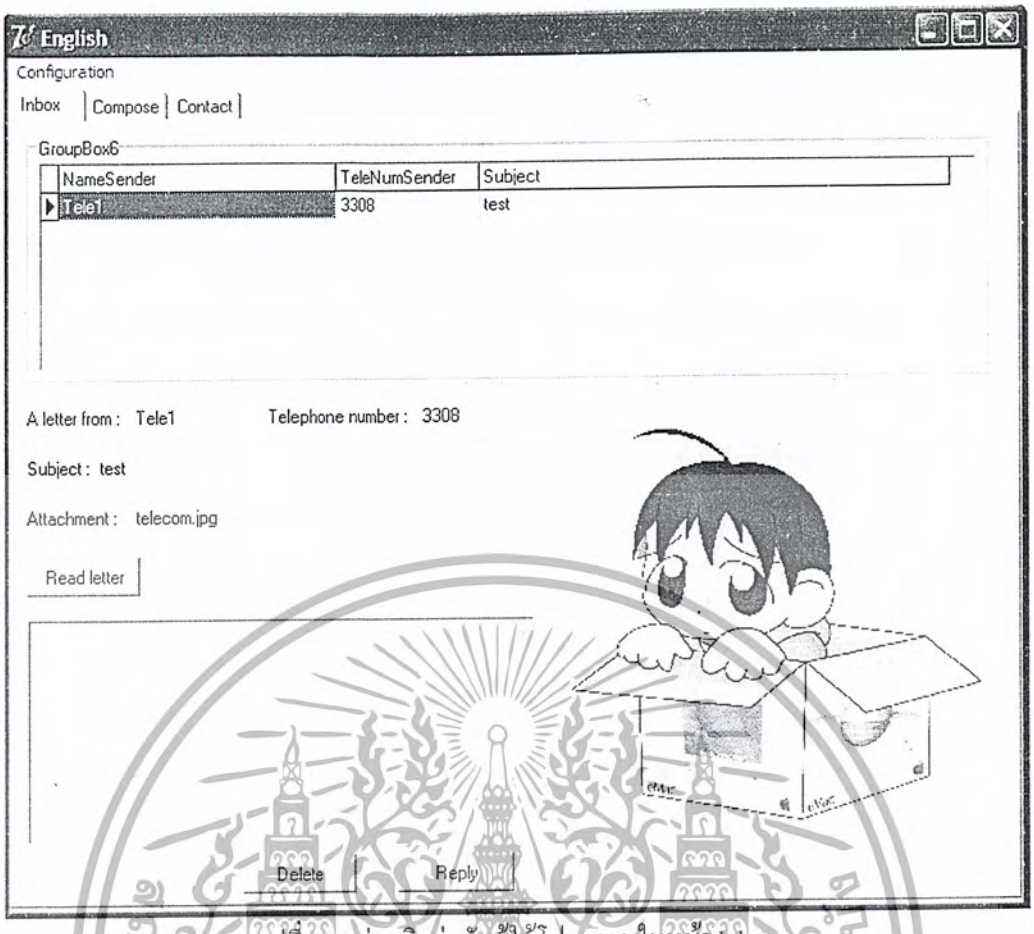
เมื่อโปรแกรมได้รับ ข้อความที่ส่งมาจาก Server แล้วจะถูกบันทึกรายละเอียดอยู่ในตารางฐานข้อมูล และจะแสดง รายการอยู่ในกล่องจดหมาย (Inbox) และเมื่อทำการเลือกข้อความที่จะอ่าน แล้วทำการคลิกปุ่ม Read โปรแกรมจะดึงข้อมูลจากฐานข้อมูล และแสดงข้อความและรายชื่อพร้อมเบอร์ผู้ส่งออกมา เมื่อผู้ใช้ได้อ่านจดหมายแล้วจะสามารถลบจดหมายนั้นทิ้งโดยคลิก Delete จะปรากฏ Dialog box ขึ้นมาถามผู้ใช้ว่า "คุณต้องการที่จะลบจดหมายฉบับนี้ใช่หรือไม่?" ถ้าคลิก Yes จดหมายก็จะถูกลบออกไปจากฐานข้อมูล โดยใช้ Delete SQL แต่ถ้าคลิก No จะกลับมายังหน้าเดิมที่ยังเปิดอ่านจดหมายอยู่ หรือเมื่ออ่านจดหมายแล้วผู้ใช้ต้องการที่จะตอบจดหมายเลยในทันทีก็ทำการคลิก Reply หรือ ตอบจดหมาย จะทำให้ชื่อผู้ส่ง เบอร์โทรศัพท์ของผู้ส่ง หัวข้อสนทนา และเนื้อความจดหมายก็จะมาแสดงในหน้า Compose ทันที โดยในรูป 3.16 แสดงแผนผังการทำงานในหน้า Inbox และรูปที่ 3.17 แสดงส่วนติดต่อกับผู้ใช้โปรแกรมในหน้า Inbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แผนผังการทำงานในหน้า Inbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 ส่วนติดต่อกับผู้ใช้โปรแกรมในหน้า Inbox

3.2.5 Receive Message

ในส่วนของ Receive Message จะประกอบอยู่ในหน้าโปรแกรมหลัก เมื่อโปรแกรมถูกเรียกเปิดใช้งาน ยูนิต TnrCommTAPI จะสั่งให้โมเด็มรอรับสัญญาณเรียกโดยอัตโนมัติ และเมื่อมีสัญญาณเรียกมาจาก Server โปรแกรมจะทำการตอบรับ และยูนิต TnrZModem จะสั่งให้โมเด็มรอรับไฟล์ ข้อความ และไฟล์ที่ถูกแนบ ที่ถูกส่งมาจาก Server และบันทึกไฟล์ข้อความลงในฐานข้อมูล ในรูปที่ 3.18 แสดงแผนผังการทำงานในส่วนของ Receive Message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.20 แสดงแผนผังการทำงานในส่วนของ Receive Message

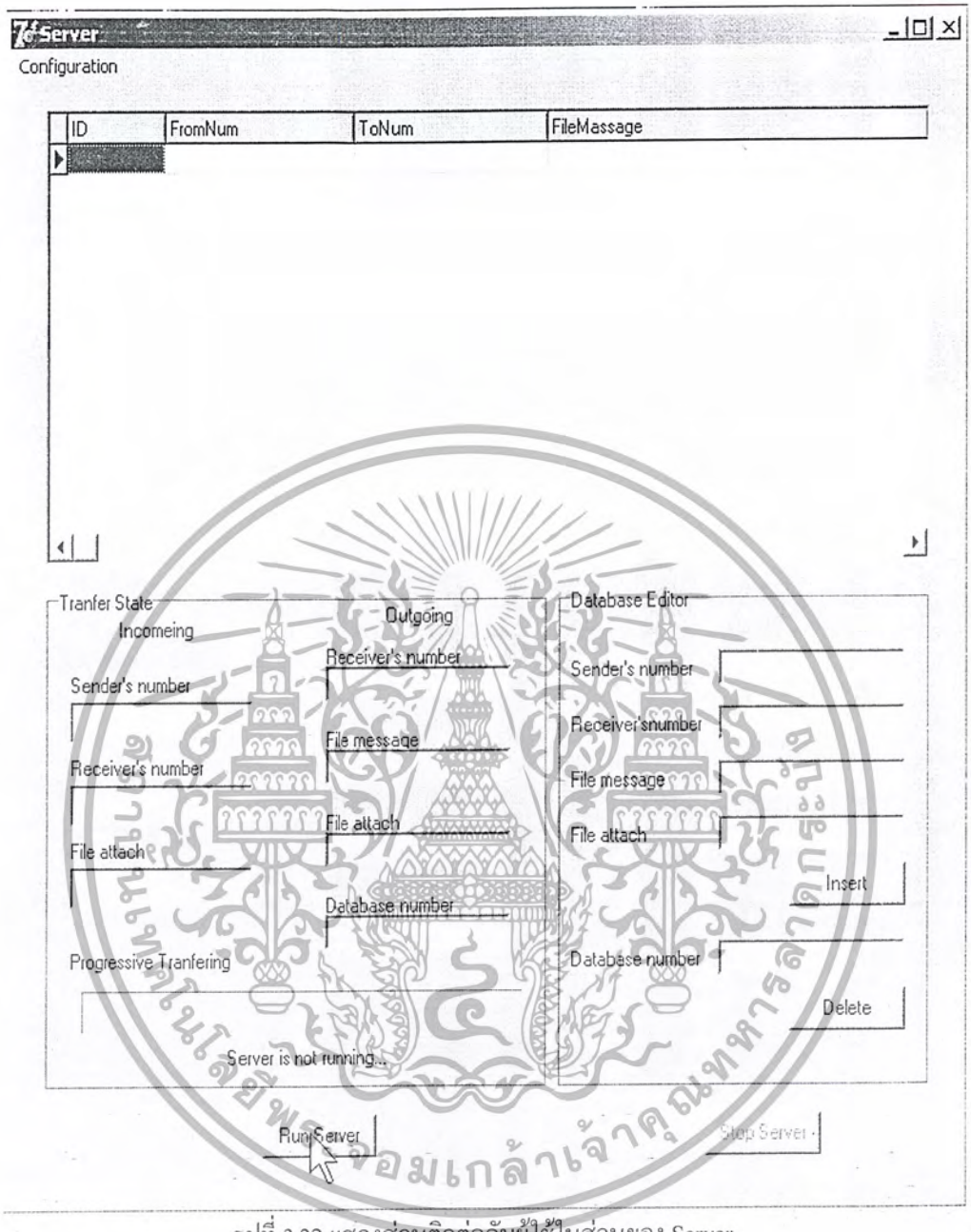
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

3.3 การทำงานในส่วนของ Sever

Sever จะทำหน้าที่รับข้อความและไฟล์ที่แนบมาจาก Client ทางต้นทางแต่ละตัว ทำการบันทึกลงฐาน และจัดลำดับก่อนที่จะทำการจัดส่งไฟล์ข้อความและไฟล์ที่แนบมาเหล่านี้ไปยัง Client ทางปลายทาง โดยถ้าหากไม่สามารถติดต่อกับ Client ทางปลายทางได้ในทันที จะทำการส่งไปเรื่อยๆ จนกว่าจะสามารถเชื่อมต่อกับทาง Client ทางปลายทางได้ โดยในรูป แสดงแผนผังการทำงานในส่วนของ Sever และรูปที่ ... แสดงส่วนติดต่อกับผู้ใช้ของ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.21 แสดงแผนผังการทำงานของ Server



รูปที่ 3.22 แสดงส่วนติดต่อกับผู้ใช้ในส่วนของ Server

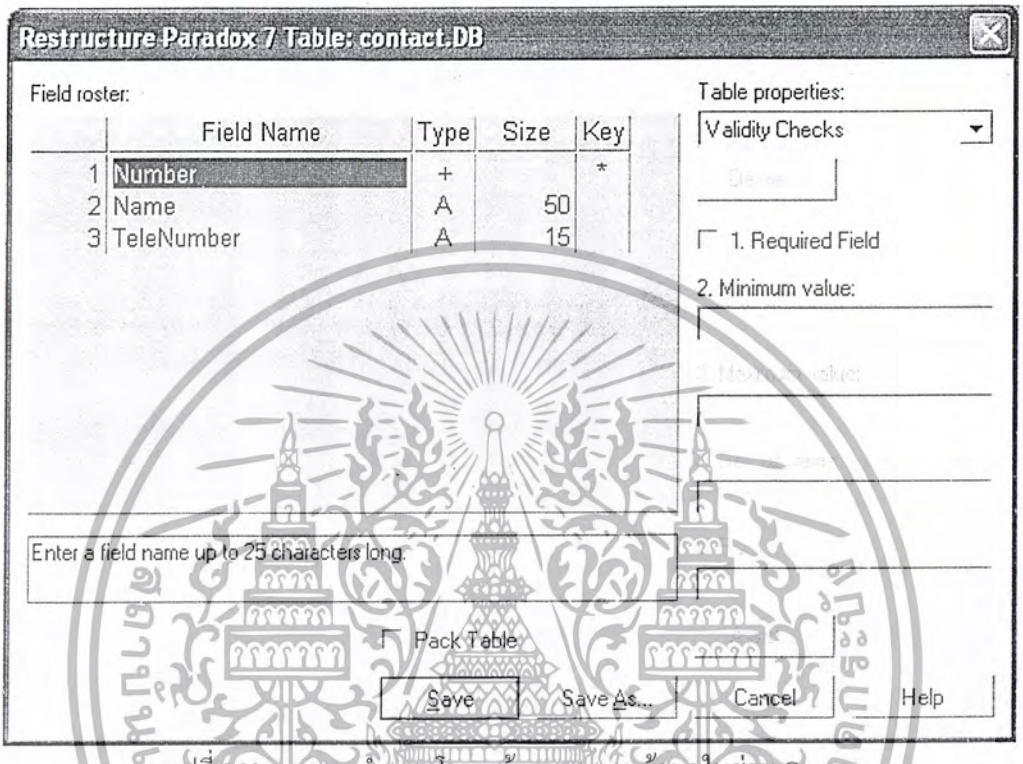
3.4การออกแบบฐานข้อมูล

ในส่วนของฐานข้อมูลนั้น ในโครงงานนี้ได้เลือกใช้ไดรเวอร์ของฐานข้อมูลเป็นประเภท Paradox ในการสร้างตาราง เนื่องจาก Paradox เป็นตารางมาตรฐานที่ใช้งานใน Delphi โดยฐานข้อมูลจะประกอบด้วยตารางทั้งหมด 3 ตารางด้วยกัน คือ ตารางในส่วนของฝั่ง Client 2 ตาราง เพื่อใช้เก็บข้อมูลใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ Contact และเก็บข้อมูลในส่วนของกล่องจดหมาย ตารางในส่วนของ Server 1 ตารางเพื่อเก็บเบอร์โทรศัพท์ของผู้รับและผู้ส่ง และเก็บข้อมูลที่ประกอบด้วยเนื้อความจดหมายและไฟล์ที่ได้แนบไป

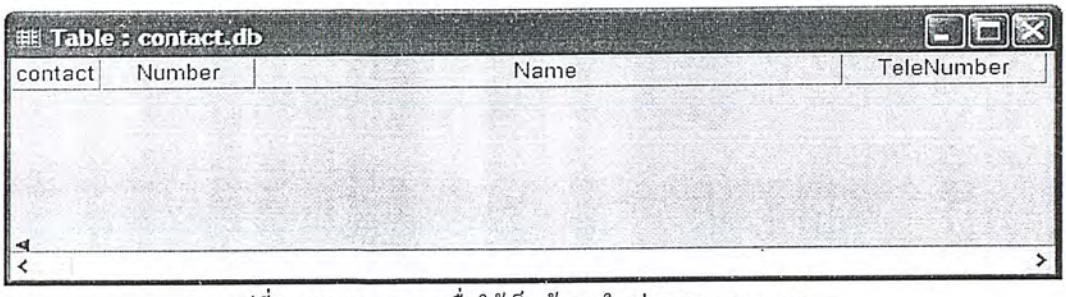
ในส่วนของ Contact จะประกอบไปด้วยชื่อและเบอร์โทรศัพท์ที่เราต้องการบันทึกไว้ แทนสมุดจด ดังนั้นในส่วนของฐานข้อมูลต้องสร้างตารางที่ประกอบด้วยฟิลด์ 3 ฟิลด์ คือ ฟิลด์ในส่วนของ Index ฟิลด์ชื่อ และฟิลด์เบอร์โทรศัพท์ซึ่งจะได้ดังรูปที่ 3.23



รูปที่ 3.23 แสดงการกำหนดโครงสร้างตารางฐานข้อมูลในส่วน Contact

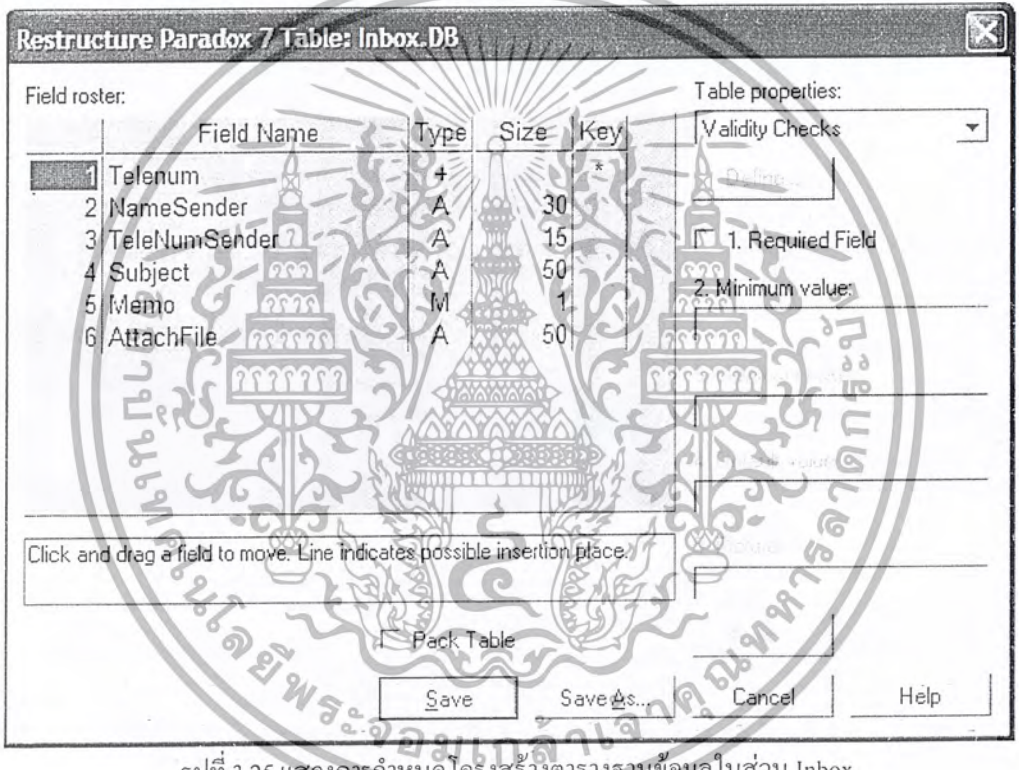
จากรูปที่ 3.23 เราทำการออกแบบตารางโดยกำหนดโครงสร้างของตารางที่ต้องการ โดยให้ชื่อของฟิลด์ที่เก็บ Index ของข้อมูลเป็น Number โดยมีชนิดของฟิลด์เป็น Autoincrement (ค่าจะเพิ่มขึ้นเรื่อยๆ) และกำหนดให้เป็น Pramary Key คือข้อมูลจะไม่ซ้ำกัน ชื่อของฟิลด์ที่เก็บชื่อที่ต้องการบันทึกเป็น Name กำหนดให้ชนิดของฟิลด์เป็น Alpha (ตัวอักษร) มีขนาดเท่ากับ 50 ตัวอักษร และชื่อของฟิลด์ที่เก็บหมายเลขโทรศัพท์เป็น TeleNumber กำหนดให้ชนิดของฟิลด์เป็น Alpha มีขนาดเท่ากับ 15 ตัวอักษร จากนั้นทำการบันทึกและเปิดตาราง Contact จะได้ดังรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงตารางเพื่อใช้เก็บข้อมูลในส่วนของ Contact

ตารางที่ใช้เก็บข้อมูลในส่วนของกล่องจดหมายหรือ Inbox นั้นเราต้องการสร้างฟิลด์เพื่อเก็บข้อมูลในส่วนของชื่อ เบอร์โทรศัพท์ของผู้ส่ง หัวข้อสนทนา เนื้อความจดหมายและชื่อไฟล์ที่ได้แนบไปซึ่งต้องออกแบบในส่วนของ Field roster ดังนี้

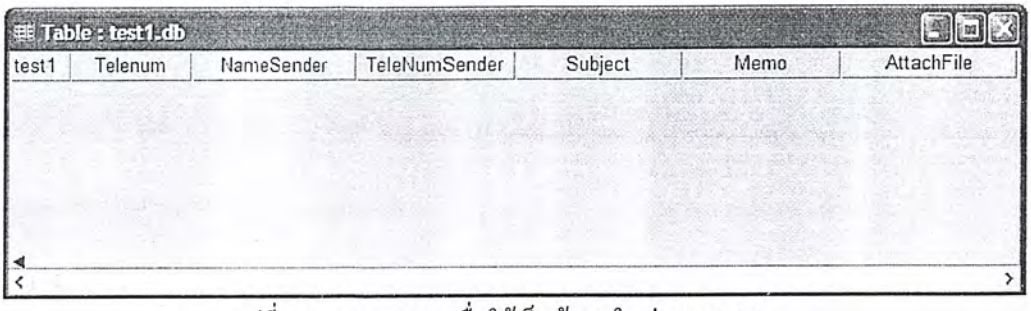


รูปที่ 3.25 แสดงการกำหนดโครงสร้างตารางฐานข้อมูลในส่วน Inbox

จากรูปที่ 3.25 เรากำหนดโครงสร้างของตารางโดยให้ฟิลด์ Telenum เป็นส่วนที่นับโดยกำหนดให้เป็น Pramary Key ฟิลด์ NameSender ใช้เก็บชื่อผู้ส่ง เป็นชนิด Alpha มีขนาดเท่ากับ 30 ตัวอักษร ฟิลด์ TeleNumSender ใช้เก็บเบอร์โทรศัพท์ของผู้ส่ง เป็นชนิด Alpha มีขนาดเท่ากับ 15 ตัวอักษร ฟิลด์ Subject ใช้เก็บหัวข้อสนทนา เป็นชนิด Alpha มีขนาดเท่ากับ 50 ตัวอักษร ฟิลด์ Memo ใช้เก็บเนื้อความจดหมาย เป็นชนิด Memo มีขนาดเท่ากับ 1 MB และฟิลด์ AttachFile ใช้เก็บชื่อไฟล์ที่แนบไป

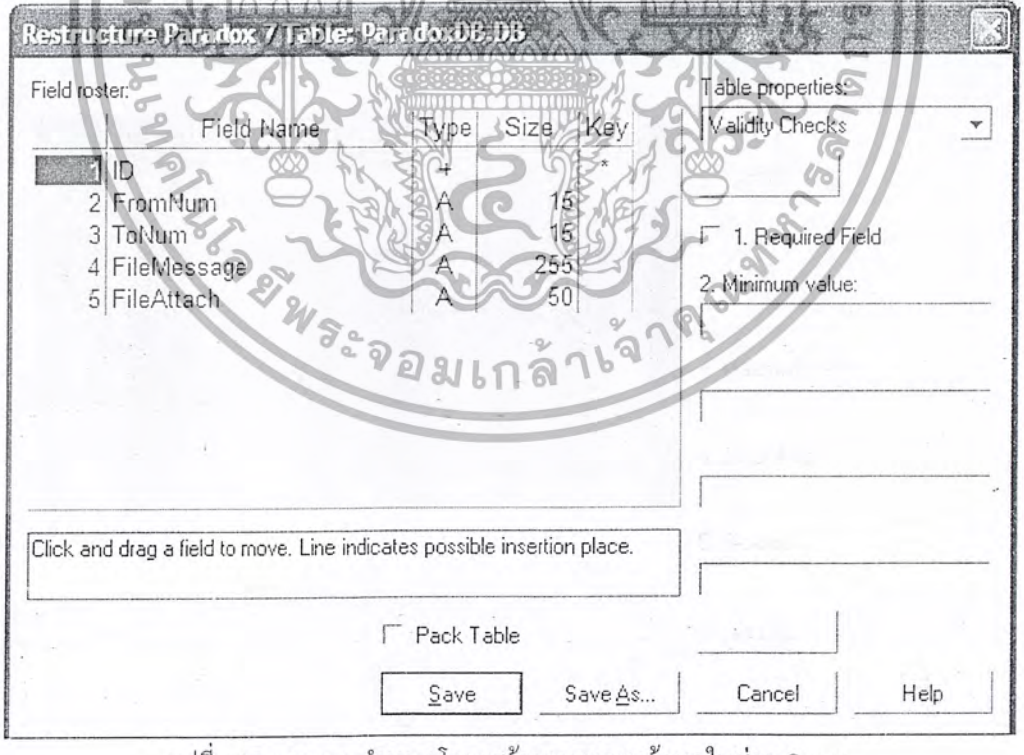
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นชนิด Alpha มีขนาดเท่ากับ 50 ตัวอักษร ซึ่งจากการออกแบบโครงสร้างตารางฐานข้อมูลจะได้ตารางฐานข้อมูลดังรูปที่ 3.26



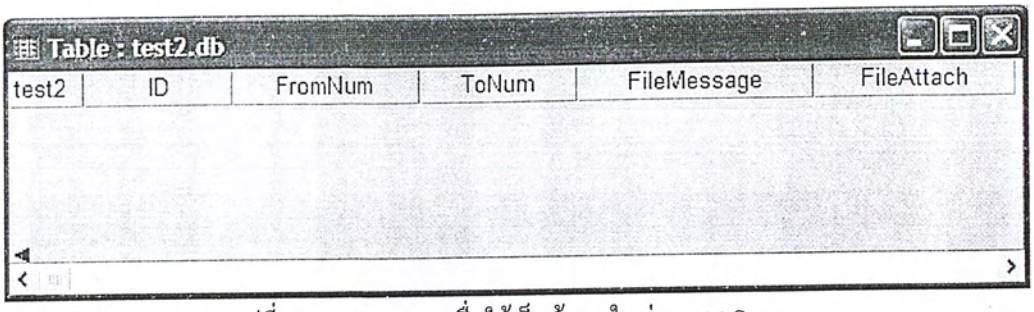
รูปที่ 3.26 แสดงตารางเพื่อใช้เก็บข้อมูลในส่วนของ Inbox

ในส่วนของ Server นั้น เราต้องการตารางฐานข้อมูลที่จะเก็บเบอร์โทรศัพท์ของผู้รับ และผู้ส่งจดหมาย ส่วนที่เก็บชื่อผู้รับและผู้ส่ง หัวข้อสนทนาและเนื้อความจดหมาย ส่วนที่เก็บชื่อไฟล์ที่แนบมา ซึ่งกำหนดให้ชื่อฟิลด์ FromNum ใช้เก็บเบอร์โทรศัพท์ของผู้ส่ง มีชนิดเป็น Alpha โดยมีขนาดเท่ากับ 15 ตัวอักษร ฟิลด์ ToNum ใช้เก็บเบอร์โทรศัพท์ของผู้รับ มีชนิดเป็น Alpha โดยมีขนาดเท่ากับ 15 ตัวอักษร ฟิลด์ FileMessage ใช้เก็บชื่อผู้รับและผู้ส่ง หัวข้อสนทนาและเนื้อความจดหมาย มีชนิดเป็น Alpha โดยมีขนาดเท่ากับ 255 ตัวอักษร และฟิลด์ FileAttach มีชนิดเป็น Alpha โดยมีขนาดเท่ากับ 50 ตัวอักษร จะได้ดังรูปที่ 3.27 และตารางที่ใช้เก็บข้อมูลต่างๆ ของ Server จะได้ดังรูปที่ 3.28 ตามลำดับ



รูปที่ 3.27 แสดงการกำหนดโครงสร้างตารางฐานข้อมูลในส่วน Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 แสดงตารางเพื่อใช้เก็บข้อมูลในส่วนของ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

ผลการทดลอง

โดยในการทดลอง จะแสดงการทำงานของโปรแกรมในส่วนของ Client ซึ่งประกอบด้วย การเพิ่มข้อมูล การแก้ไขข้อมูลและการลบข้อมูลใน Contact การส่งข้อความและไฟล์ที่ต้องการแนบใน Compose และการรับข้อความและไฟล์ที่แนบที่ถูกส่งมากจาก Server และสามารถแสดงผลออกมาได้ ในส่วนของ Server จะแสดงการรับข้อความที่รับมาจาก Client ทางด้านส่ง และทำการส่งต่อไปยัง Client ทางด้านรับ

4.1 ขั้นตอนการทดลอง

- ทำการเปิดโปรแกรม Client ทางด้านส่ง และ Server
- ให้ Client ทางด้านส่ง ทำการ ใส่ข้อมูล แก้ไข และลบ เพื่อดูว่า โปรแกรมทำงานได้อย่างถูกต้อง จากนั้นให้ทำการส่งข้อความ และไฟล์ที่ต้องการแนบ ไปยัง เครื่อง Client ทางปลายทาง
- ในส่วนของ Server ให้ตรวจเช็คการเข้ามาของไฟล์ข้อความที่ถูกส่งมาจากทางด้าน Client ด้านส่ง ว่าถูกส่งไปยัง Client ทางด้านรับได้อย่างถูกต้อง
- ทำการเปิดโปรแกรม Client ทางด้านรับ เพื่อรอรับข้อความและไฟล์ไฟล์แนบที่ถูกส่งมากจาก Client ทางด้านส่ง ผ่านมาทาง Server

4.2 ผลการทดลอง Client ทางด้านส่ง

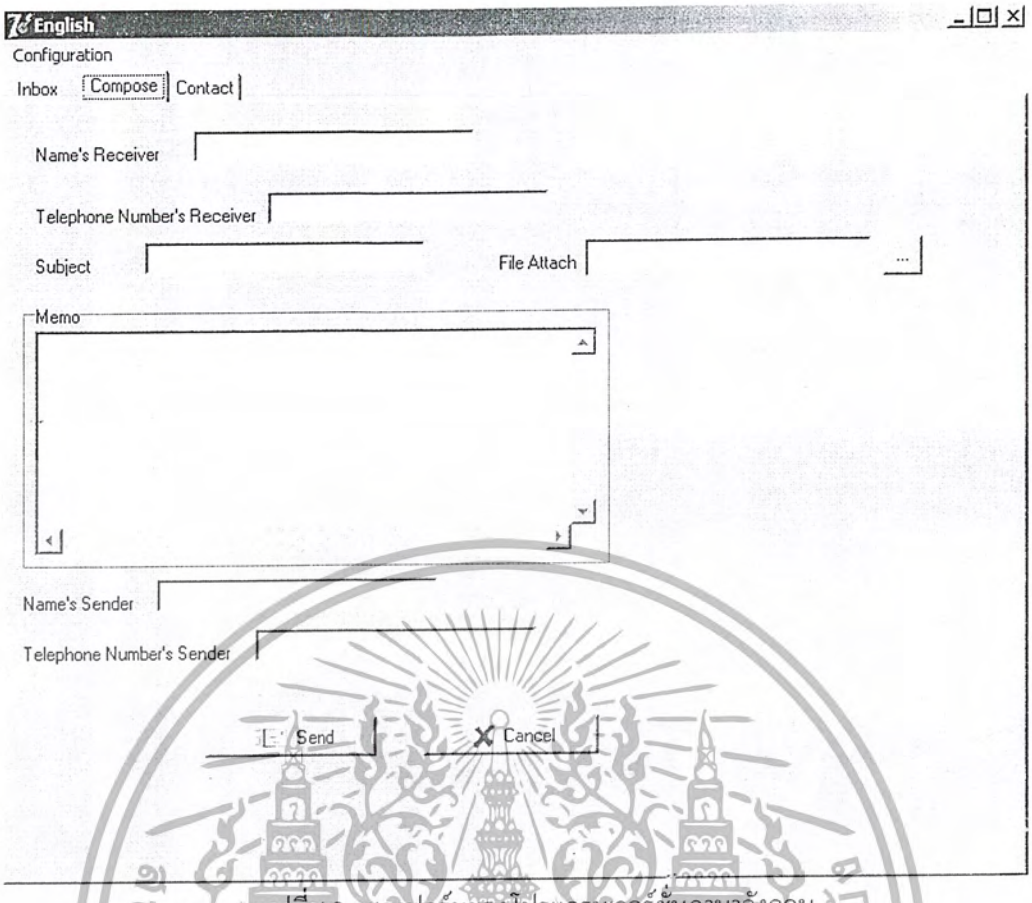
1. ทำการเปิดโปรแกรมหลักจะปรากฏฟอร์มดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 แสดงฟอร์มหลักของโปรแกรม

เมื่อทำการรันโปรแกรมจะปรากฏฟอร์มหลักเพื่อให้ผู้ใช้ได้ทำการเลือกภาษา ซึ่งในการทดลองนี้ได้เลือกภาษาอังกฤษ ซึ่งจะปรากฏฟอร์มดังรูปที่ 4.2

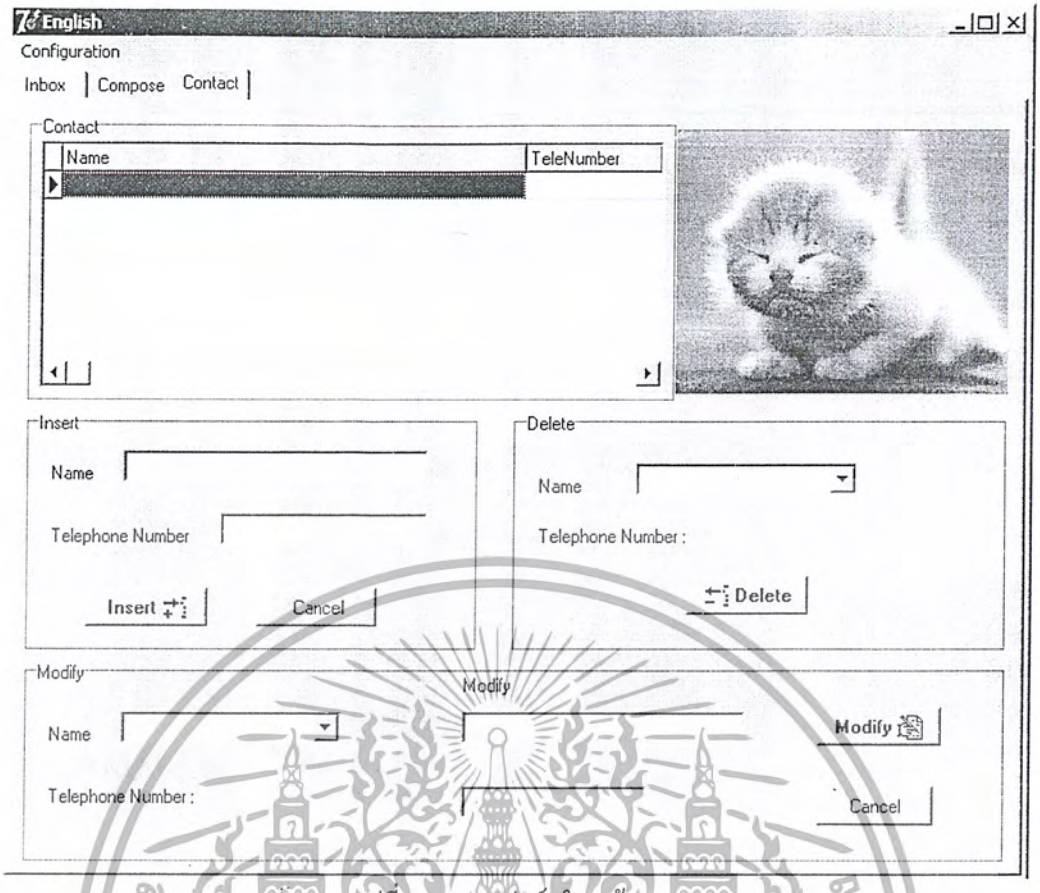
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงฟอร์มของโปรแกรมเวอร์ชั่นภาษาอังกฤษ

2. จากนั้นคลิกในแถบ Contact จะปรากฏฟอร์มดังในรูปที่ 4.3 ซึ่งเป็นส่วนที่เปรียบเสมือนสมุดโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



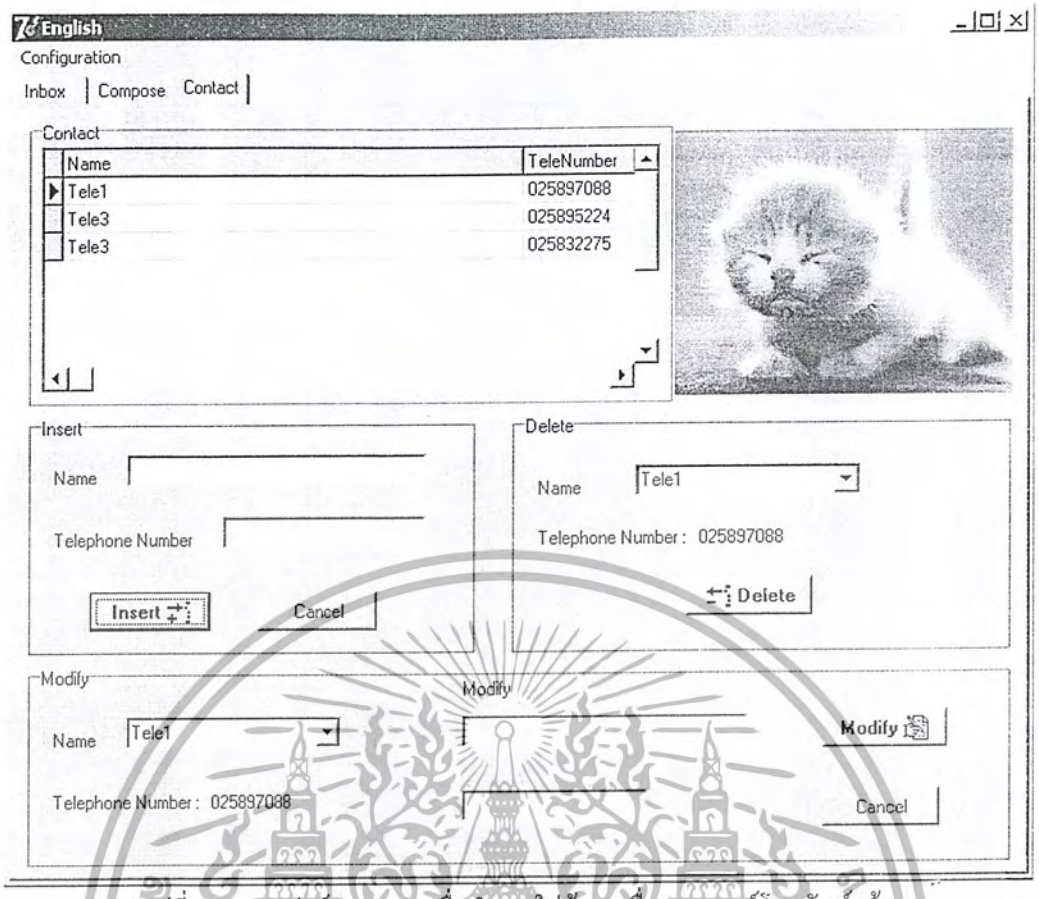
รูปที่ 4.3 แสดงฟอร์มในหน้า Contact

3. จากนั้นทำการใส่ข้อมูลต่างๆ ลงไปในฟอร์มในส่วนของ insert โดยทำการกรอกชื่อในช่อง Name และ Telephone Number แล้วทำการคลิก insert โดยกำหนดให้ข้อมูลที่ใส่ลงไปเป็นดังนี้ดังนี้

Name	Telephone Number
Tele1	025897088
Tele2	025895224
Tele3	025832275

เมื่อทำการกรอกข้อมูลจนครบถ้วนแล้วจะแสดงออกมาได้ยังฟอร์ม ในรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

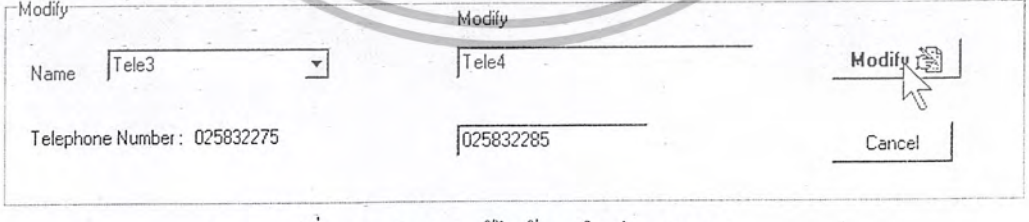


รูปที่ 4.4 แสดงฟอร์ม Contact เมื่อทำการใส่ข้อมูล ชื่อ และเบอร์โทรศัพท์แล้ว

4. ในส่วนของ Modify ให้ ทำการเลือก Name เป็น Tele3 แล้วทำการแก้ไขข้อมูลใหม่ดังนี้

Name	Telephone Number
Tele4	025832285

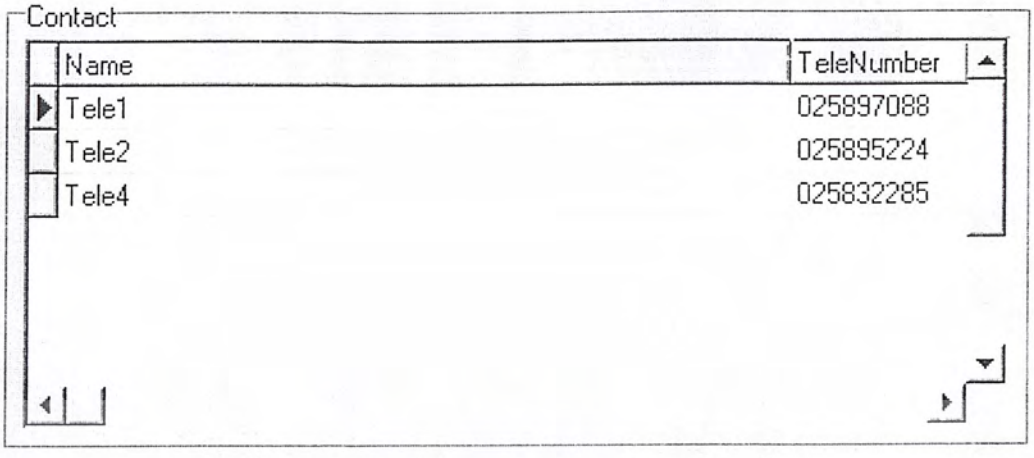
โดยวิธีการแก้ไขข้อมูลแสดงอยู่ในรูปที่ 4.5



รูปที่ 4.5 แสดงการแก้ไขข้อมูลในส่วน Modify

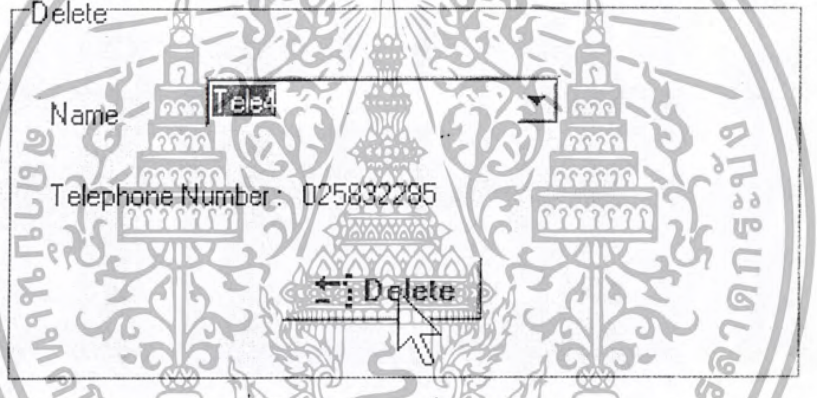
หลังจากทำการแก้ไขแล้ว ลิสต์ข้อมูลในส่วนของ Contact จะแสดงออกมาดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



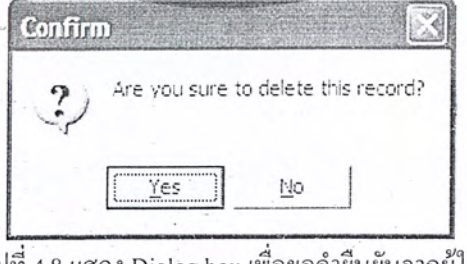
รูปที่ 4.6 แสดงข้อมูลใน Contact หลังจากทำการแก้ไขแล้ว

5. ในส่วนของ Delete ให้ทำการเลือก Tele4 แล้วทำการคลิก Delete ดังรูปที่ 4.7



รูปที่ 4.7 แสดงการลบข้อมูลใน Contact

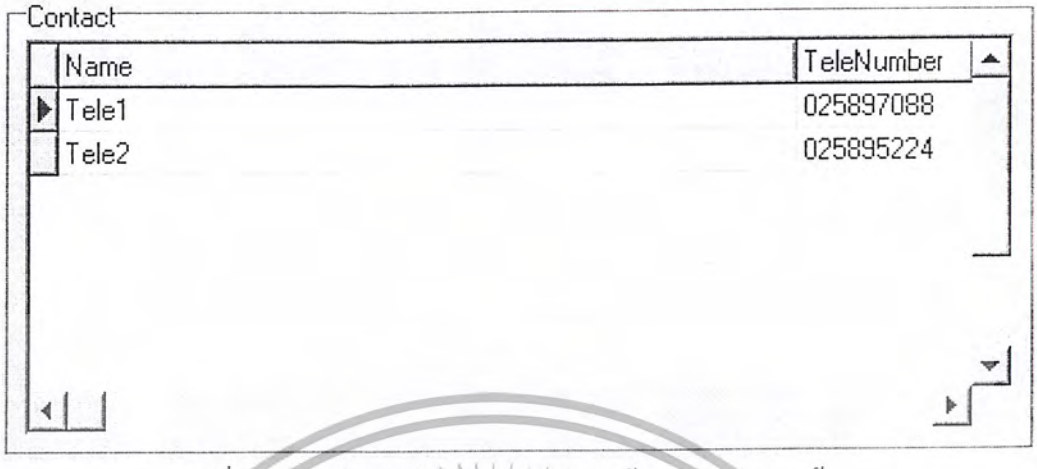
โดยหลังจากที่ทำการคลิก Delete แล้ว จะปรากฏ Dialog box ขึ้นมาถามผู้ใช้ว่าต้องการจะลบเร็คคอร์ดนี้ใช่หรือไม่ ดังรูปที่ 4.8



รูปที่ 4.8 แสดง Dialog box เพื่อขอคำยืนยันจากผู้ใช้

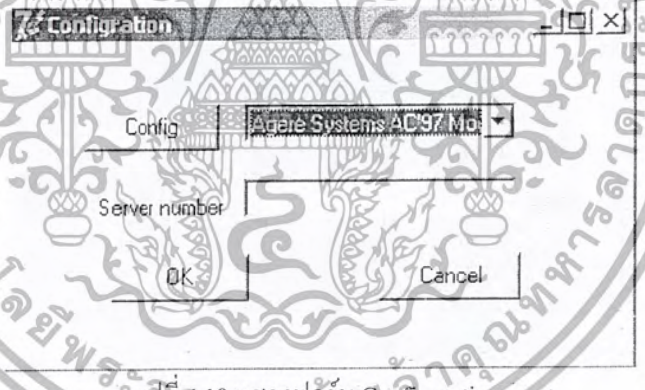
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 ถ้าผู้ใช้คลิก "Yes" ข้อมูล Tele4 ใน Contact จะถูกลบดังแสดงในรูปที่4.9 แต่ถ้าคลิก "No" โปรแกรมจะกลับมายังหน้า Contact เดิม โดยไม่มีการเปลี่ยนแปลงข้อมูลใดๆ



รูปที่ 4.9 แสดงข้อมูลใน Contact หลังจากลบข้อมูล Tele4 ออกแล้ว

6.คลิกตรงแถบ Compose จะปรากฏ โปรแกรมในส่วนของ Compose โดยก่อนที่จะทำการส่งข้อความไปยัง Server จะต้องทำการกำหนดค่าพารามิเตอร์ในการสื่อสาร โดยทำการคลิกปุ่ม Configuration โปรแกรมจะแสดงฟอร์ม Configuration ดังแสดงในรูปที่ 4.10



รูปที่ 4.10 แสดงฟอร์ม Configuration

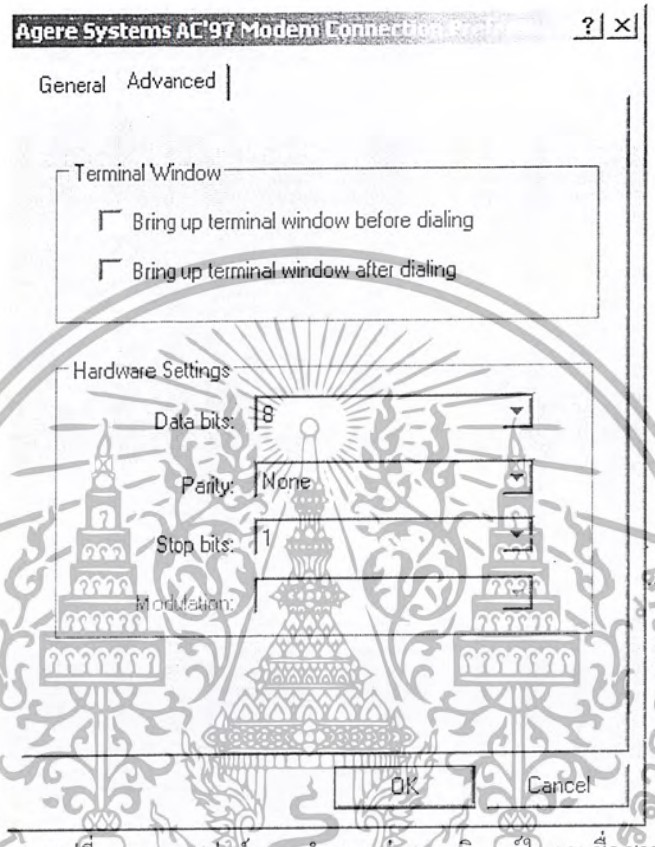
- ส่วน Tool box เป็นตัวกำหนดตัวโมเด็ม โดยในการทดลองนี้ใช้โมเด็มทางด้านส่งยี่ห้อ Agere Systems AC'97
- ส่วน Sever number เป็นตัวกำหนดเบอร์ที่ติดต่อกับ Sever โดยในที่นี้กำหนดให้เป็นเบอร์ 3303

ปุ่ม Config มีไว้เพื่อกำหนดเงื่อนไขต่างๆ ในการเชื่อมต่อ โดยจะเรียกขึ้นมาจากระบบปฏิบัติการ Windows อาทิเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Baud Rate เพื่อกำหนดความเร็วของข้อมูลที่จะส่ง
- Data Bit เพื่อกำหนดขนาดข้อมูลที่จะส่ง
- Parity เพื่อกำหนดชนิดของการตรวจสอบ Parity

เมื่อผู้ใช้ต้องการกำหนดเงื่อนไขต่างๆ ให้ผู้ใช้กำหนดปุ่ม Config โปรแกรมจะทำการแสดง ฟอร์ม ออกมาดังรูปที่ 4.11

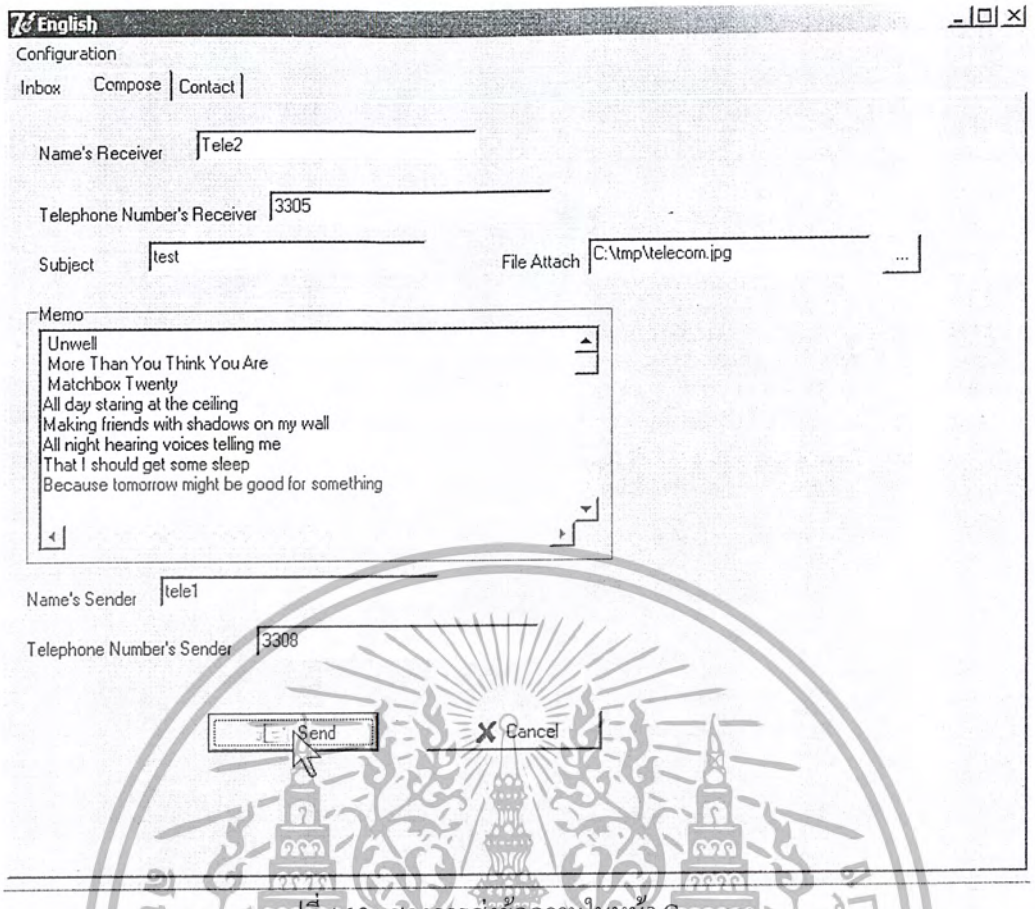


รูปที่ 4.11 แสดงฟอร์มการกำหนดค่าพารามิเตอร์ในการสื่อสาร

โดยการกำหนดเงื่อนไขต่างๆ ในรูปที่ 4.11 นั้น จำเป็นอย่างยิ่งที่ทางผู้เรียกและผู้รับจะต้องกำหนดส่วนต่างๆ ต่อไปนี้ให้ตรงกัน คือ Baud rate, Data bits, Parity เมื่อทำการกำหนดค่าต่างๆ เป็นที่เรียบร้อยแล้วให้ คลิก OK เพื่อออกสู่หน้าหลัก

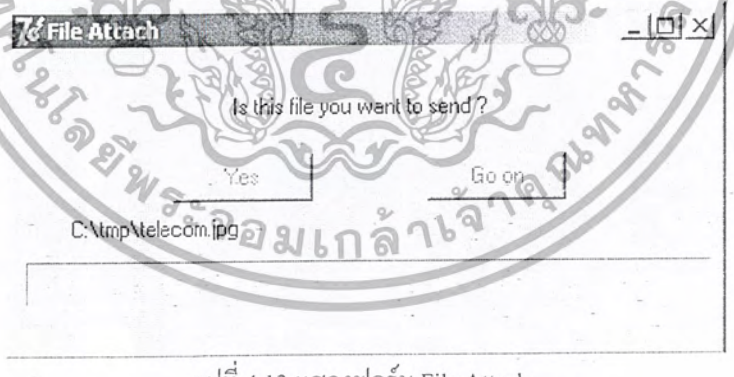
หลังจากกำหนดค่าพารามิเตอร์ในการส่งเรียบร้อยแล้ว ให้ทำการเขียนข้อความที่ต้องการส่งรายละเอียดต่างๆ ในหน้า Compose ตลอดจนไฟล์ที่ต้องการแนบ โดยการทดลองนี้จะทำการแนบไฟล์ C:\tmp\telecom.jpg ดังตัวอย่างที่แสดงในรูปที่ 4.12 หลังจากนั้นทำการคลิก OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงการส่งข้อความในหน้า Compose

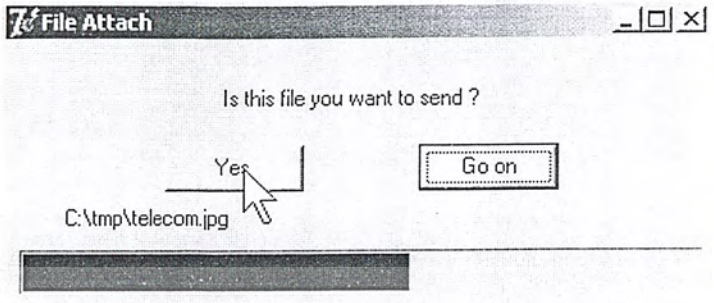
7.หลังจากคลิกปุ่ม Send จะปรากฏฟอร์ม File Attach ขึ้นมา ดังรูปที่ 4.13



รูปที่ 4.13 แสดงฟอร์ม File Attach

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.ถ้าหาก Client สามาถเชื่อมต่อไปยัง Server ได้ ปุ่ม Yes จะ enable ให้ทำการคลิก ที่ปุ่ม Yes โปรแกรมจะทำการแสดง process การส่งไฟล์ ดังรูปที่ 4.14 เมื่อการส่งไฟล์ข้อมูลถูกส่งไปยัง Server เป็นที่เรียบร้อยให้ทำการคลิกที่ปุ่ม Go on แล้วทำการปิดหน้าต่าง

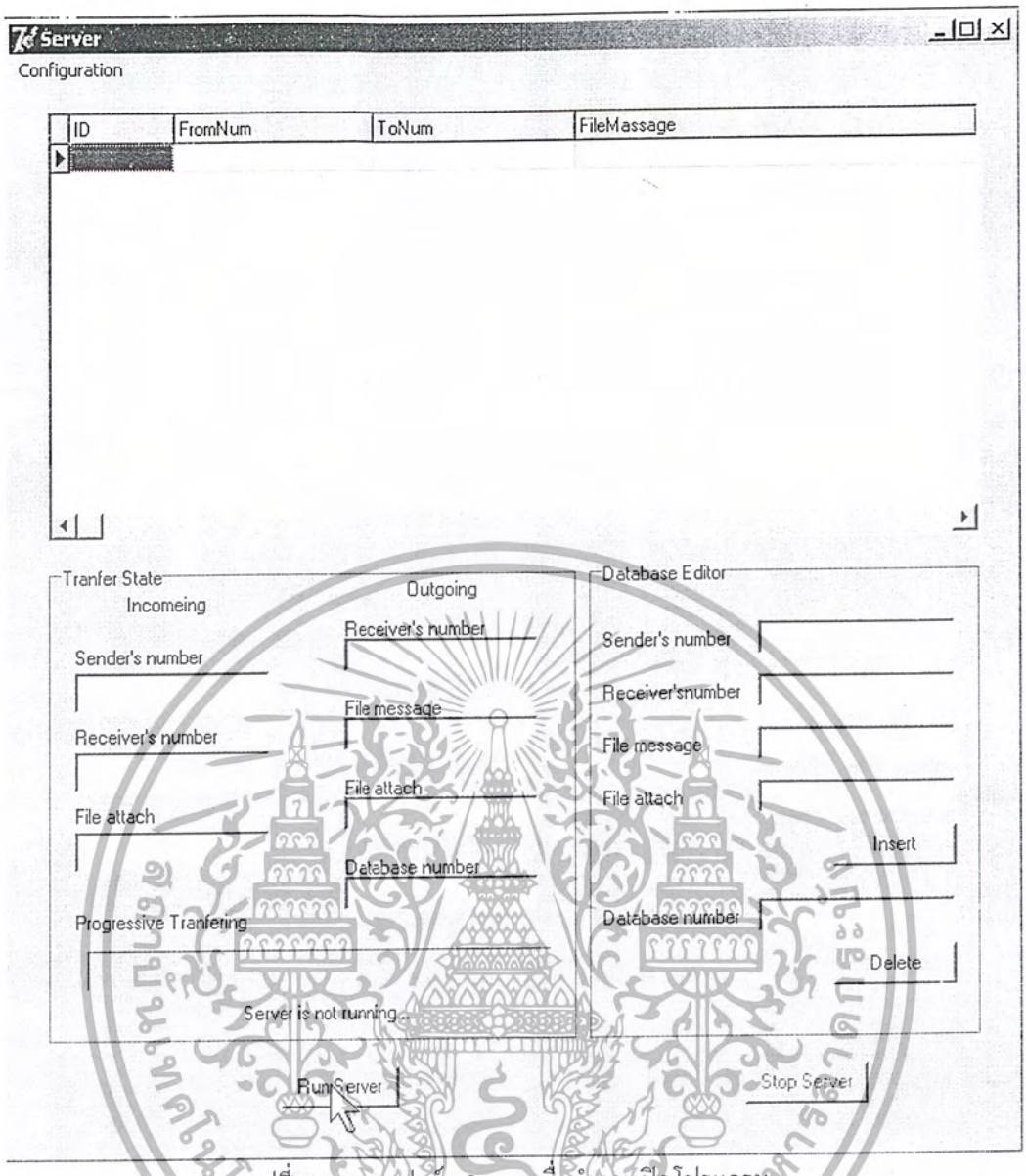


รูปที่ 4.14 แสดงฟอร์ม File Attach ขณะทำการส่งข้อความ

4.3 ผลการทดลองทางด้าน Server

1. เมื่อทำการเปิดโปรแกรมจะปรากฏฟอร์ม ออกมาดังรูป 4.15 โดยที่เมื่อฟอร์มทางฝั่ง server ถูกเปิดออกมาโปรแกรมจะถูกสั่งให้รอรับสัญญาณเข้าโดยอัตโนมัติ จากนั้น คลิกปุ่ม Run Server เพื่อเริ่มการทำงานของ Server

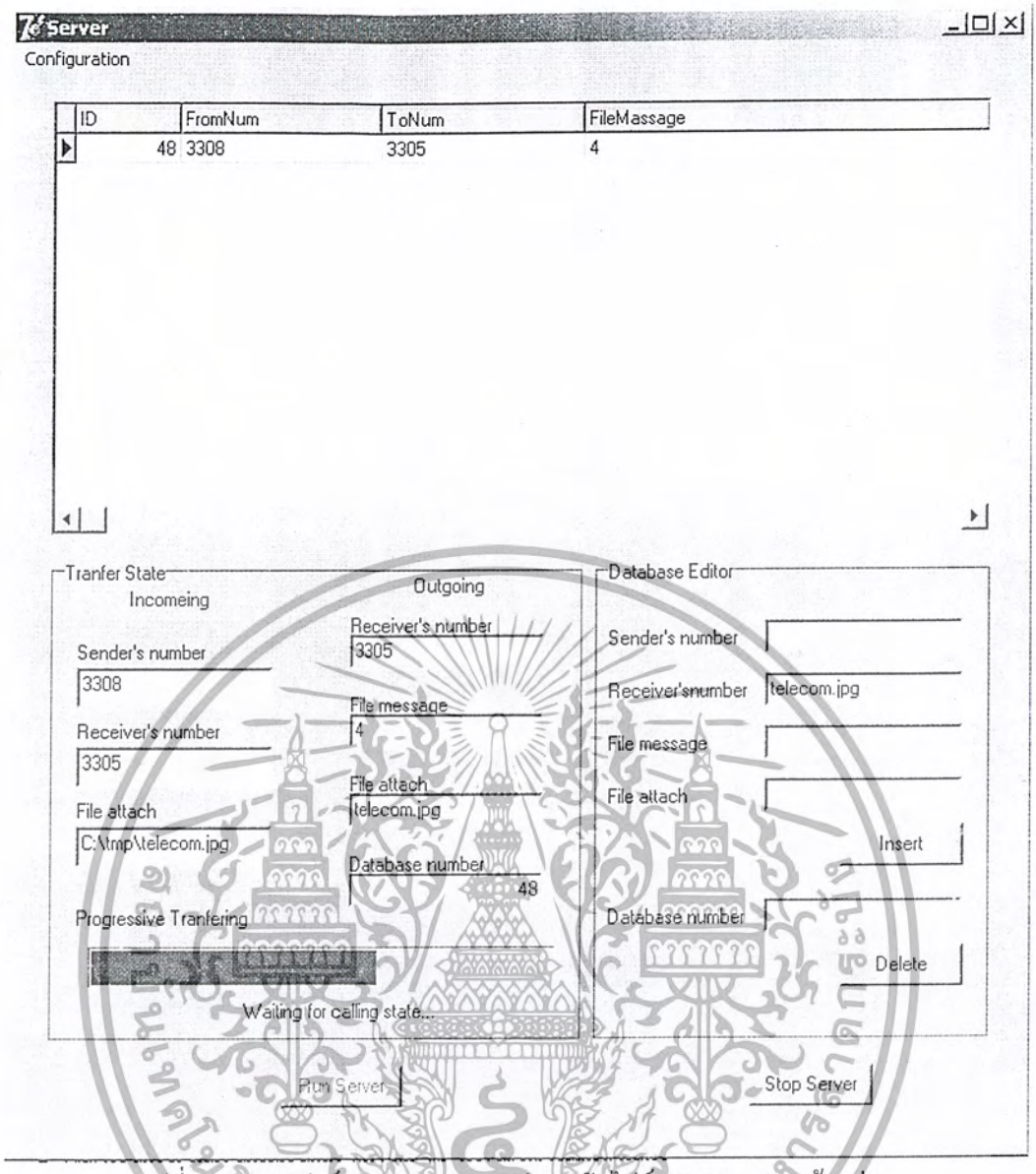
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงฟอร์ม Server เมื่อทำการเปิดโปรแกรม

2. เมื่อทาง Client ทางฝั่งส่ง ทำการติดต่อกับ Server เป็นผลสำเร็จแล้วจะทำการส่งไฟล์ข้อความและไฟล์แนบมายัง Server โดยในรูปที่ 4.16 จะเป็นการแสดงสถานะของ Server เมื่อทำการรับไฟล์ข้อความจากทาง Client ทางฝั่งรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงฟอร์มของ Server ขณะทำการรับไฟล์ จาก Client ทางด้านส่ง

โดยในส่วนของฟอร์ม ในโปรแกรม Server จะประกอบด้วยสามส่วนคือ

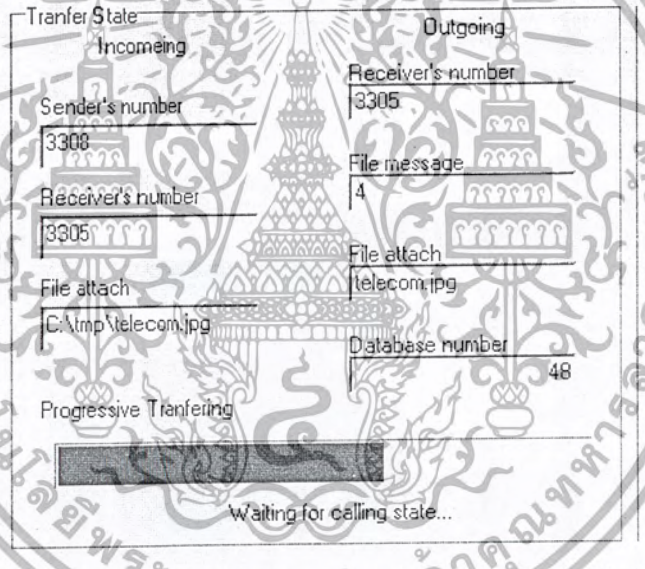
- การแสดงสถานะของ database ว่ามีรายการข้อความอะไรบ้างที่ รอทำการส่ง ไปยัง Client ทางด้านรับ ดังแสดงในรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ID	FromNum	ToNum	FileMassage
48	3308	3305	4

รูปที่ 4.17 แสดงฟอร์ม Server ในส่วนของการแสดงสถานะของ database

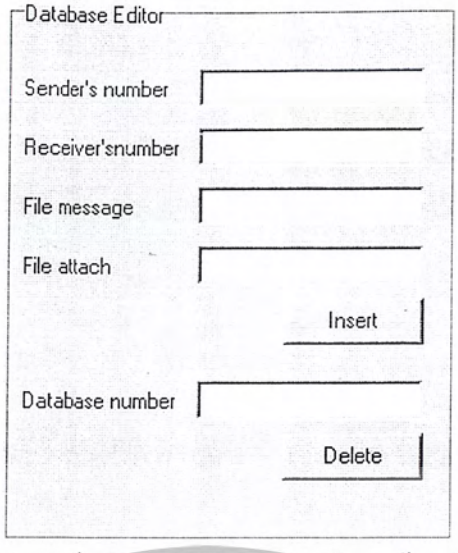
- การแสดงสถานะการรับและส่งไฟล์ โดยในแถบของ incoming จะแสดงรายละเอียดของไฟล์ข้อความที่รับเข้ามา และในแถบของ out going จะแสดงรายละเอียดของไฟล์ที่กำลังจะถูกส่งออกไปในลำดับต่อไป ดังแสดงในรูปที่ 4.18



รูปที่ 4.18 แสดงฟอร์ม Server ในส่วนของการแสดงรายละเอียดการรับส่งไฟล์

- การจัดการเกี่ยวกับข้อมูลใน database โดย Server สามารถ ที่จะจัดการข้อมูลใน database โดยการใส่ข้อมูลเพิ่มลงไปเพื่อส่งข้อความ ไปหา Client ใดก็ได้ตามต้องการ ตลอดจน สามารถลบข้อมูลใน database ออก ดังแสดงในรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงฟอร์ม Server ในส่วนของการจัดการข้อมูลใน database

3. เมื่อ Server ทำการรับไฟล์จากทางด้าน Client ฝั่งรับแล้ว จะทำข้อมูลมาเก็บไว้ใน database จากนั้นจะทำการส่งไปยัง Client ทางด้านรับต่อไป

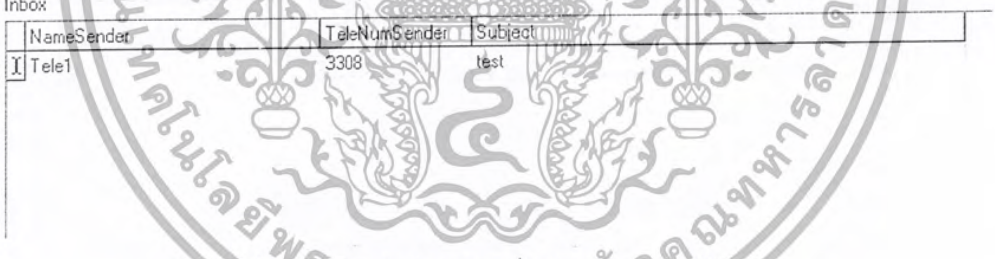
4.4 ผลการทดลองทางด้าน Client ฝั่งรับ

1. ทำการเปิดโปรแกรม Client ทางฝั่งรับ โดยโปรแกรมจะรอรับการติดต่อจาก Server โดยอัตโนมัติ โดยให้ไปคลิกที่แถบ Inbox จะปรากฏฟอร์ม Inbox ดังแสดงในรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.20 แสดงฟอร์ม inbox

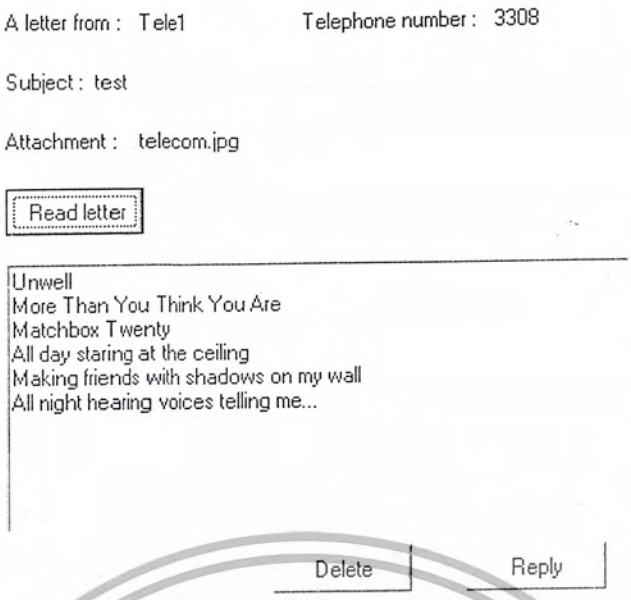
2. เมื่อมีข้อความถูกส่งเข้ามายัง Client ทางฝั่งรับ จะปรากฏรายละเอียดใน inbox ดังรูปที่ 4.21



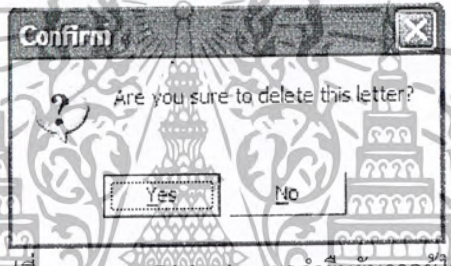
รูปที่ 4.21 แสดงฟอร์ม inbox เมื่อมีการเข้ามาของข้อความ

3. จากนั้น คลิกที่ปุ่ม Read เพื่อเปิดอ่านไฟล์ข้อความดังที่แสดงดังรูป 4.22 หรือคลิกที่ ปุ่ม Delete เพื่อลบข้อความโดยจะปรากฏ Dialog box ขึ้นมาเพื่อขอคำยืนยันจากผู้ใช้ว่าต้องการลบจดหมายฉบับนี้ใช่หรือไม่ ดังรูปที่ 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



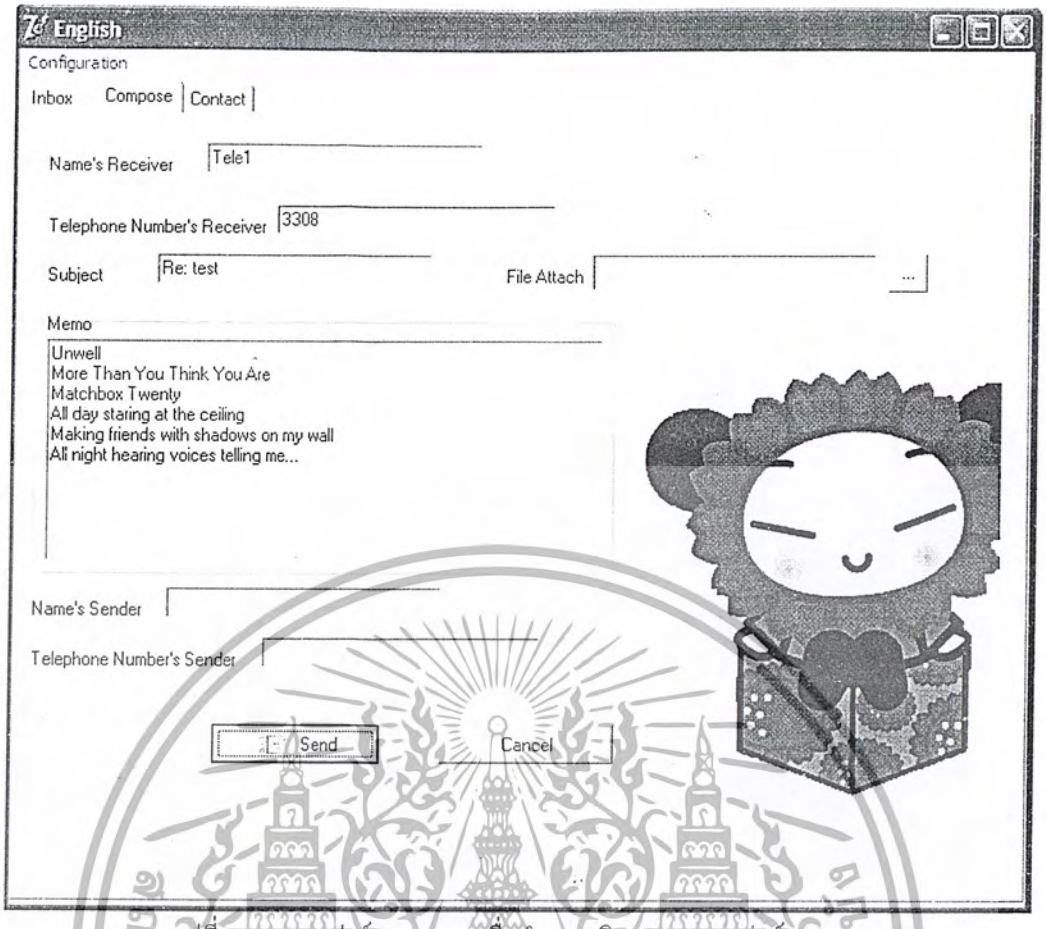
รูปที่ 4.22 แสดงฟอร์ม inbox เมื่อเปิดอ่านข้อความ



รูปที่ 4.23 แสดง Dialog box ขอคำยืนยันจากผู้ใช้

จากรูปที่ 4.23 ถ้าคลิกว่า "Yes" จดหมายจะถูกลบ แต่ถ้าคลิกว่า "No" ก็จะกลับมาสู่หน้า Inbox เดิม หรือถ้าต้องการส่งต่อ ไปให้ผู้อื่น ให้คลิก Reply ดังที่แสดงดังรูป 4.24

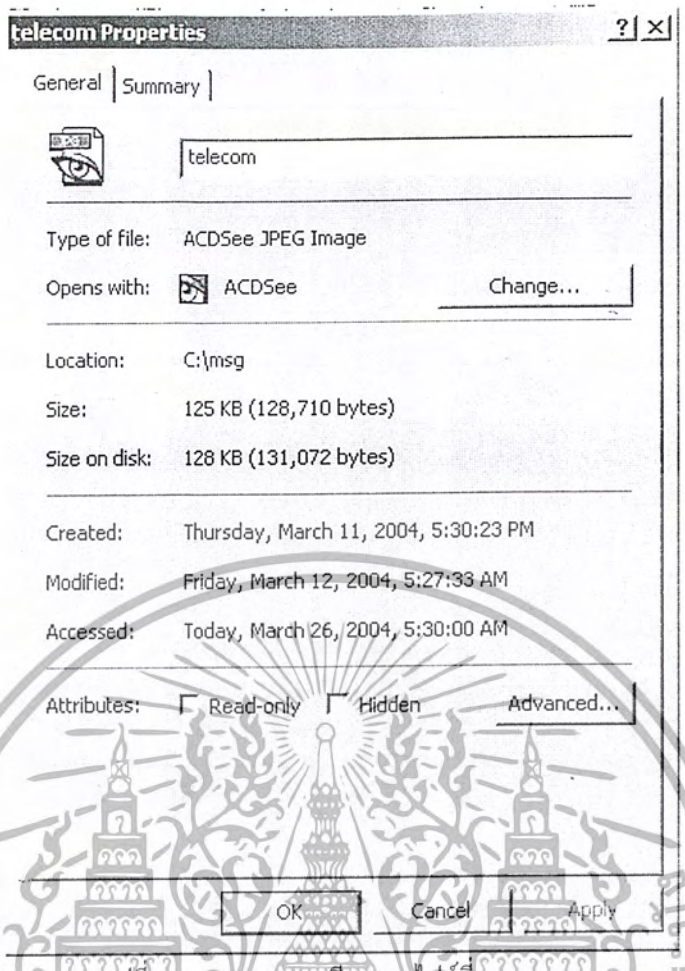
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 แสดงฟอร์มcompose เมื่อทำการคลิก Reply จากฟอร์ม inbox

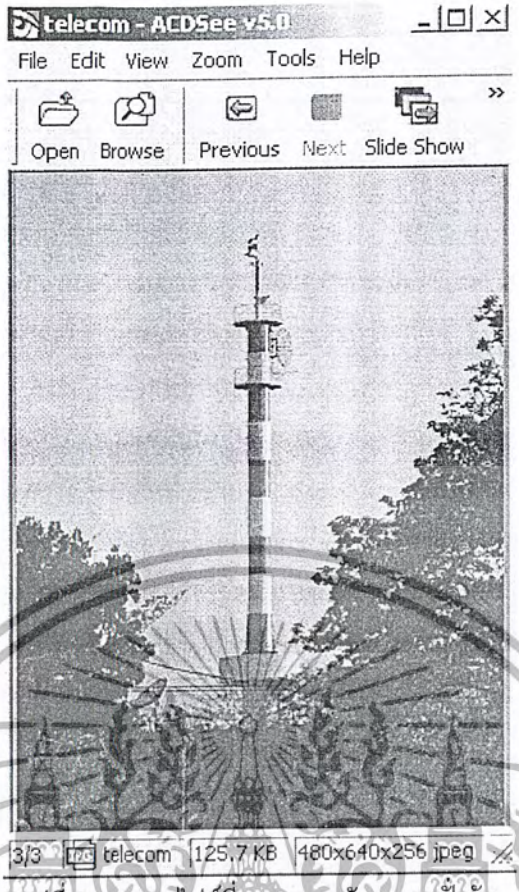
4. จากนั้นทำการตรวจสอบดูว่า ไฟล์ที่แนบมาได้รับเข้ามาครบถ้วนหรือไม่ โดยไฟล์ที่รับเข้ามาจะอยู่ในราก c:\msg ดังแสดงในรูปที่ 4.25 และรูปที่ 4.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 แสดงรายละเอียดของไฟล์ที่ถูกแบบมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 แสดงไฟล์ที่ถูกแนบมายัง Client ฝั่งรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

สรุปผลการทดลอง

จากการทดลองส่งไฟล์จากเครื่องคอมพิวเตอร์ของผู้ใช้ต้นทาง โดยเป็นไฟล์ข้อความเอกสาร ไฟล์รูปภาพ ไปยังเครื่องคอมพิวเตอร์ของ Server แล้วจากเครื่อง Serverส่งไปยังผู้ใช้ปลายทาง พบว่าโปรแกรมสามารถส่งข้อมูลและรับข้อมูลได้ โดยในฝั่งของ Server เมื่อทำการรับข้อมูลมาจากทางด้านผู้ส่ง ข้อมูลจะถูกนำไปบันทึกลงในตารางฐานข้อมูล และจากนั้น Server จึงทำการส่งไปยังด้านผู้รับ โดยเมื่อผู้รับได้รับเรียบร้อยแล้ว Server จะทำการลบข้อมูลในเร็คคอร์ดนั้น และทำการส่งข้อมูลไปยังผู้รับคนอื่นๆต่อไป ส่วนในด้านของ Inbox นั้น เมื่อได้รับข้อมูลมาจาก Server แล้วข้อมูลที่ได้จะมาเก็บไว้ในตารางฐานข้อมูลของฝั่งรับ โดยจะแสดงผลผ่านทางหน้า Inbox แล้วเราสามารถที่จะเลือกอ่านจดหมายแต่ละฉบับได้ สามารถทำการลบจดหมายออกจากตารางฐานข้อมูลได้ ตลอดจนสามารถทำการตอบจดหมายไปยังผู้ส่งได้ และในส่วนของ Contact สามารถที่จะทำการบันทึกข้อมูล ลบข้อมูล และเปลี่ยนแปลงข้อมูลได้

ปัญหาที่พบในการทดลอง

1. ระยะเวลาในการส่งข้อมูล ถ้าหากไฟล์ที่ทำการส่งมีขนาดใหญ่ อาจจะต้องใช้เวลานานในการส่ง อาจจะต้องมีการบีบอัดไฟล์ข้อมูลก่อนทำการส่งเพื่อความรวดเร็วในการส่ง
2. ถ้าหาก Server ไม่สามารถทำการติดต่อ Client ทางฝั่งรับได้ ไฟล์ข้อความที่อยู่ใน ลำดับต่อไปจะยังไม่สามารถส่งออกไปได้ โดยการแก้ไขได้ทำการเขียนโปรแกรมให้ทาง Server สามารถ ลบข้อความนั้นทิ้งไปหรือ ทำการแก้ไขให้ไปอยู่ในลำดับการส่งหลังๆ เพื่อรอการส่งต่อไป
3. ถ้าหากช่วงเวลาที่ Client ทางฝั่งส่งติดต่อไปยัง Server เป็นช่วงเวลาที่ Server กำลังส่งข้อมูลไปยัง Client ทางฝั่งรับพอดี Client ทางฝั่งส่งจะยังไม่สามารถติดต่อกับ Server ได้ทันที ต้องทำการติดต่อกับ Server อีกครั้ง
4. ความปลอดภัยของข้อมูล เนื่องจากโครงงานนี้เป็นการส่งข้อมูลโดยผ่านทางโทรศัพท์บ้าน ดังนั้นจึงทำให้ทุกคนในบ้านสามารถที่จะใช้ร่วมกัน ทำให้อาจทำการลบข้อมูลของผู้อื่นได้
5. ในด้านของฐานข้อมูลยังไม่สามารถที่จะทำการติดตั้งแล้วใช้งานได้อัตโนมัติเลยในทันที ต้องทำการติดตั้งเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

คำสั่ง AT:

คำสั่ง AT สำหรับโมเด็ม • คำสั่งพิเศษ • การอ้างถึงคำสั่ง AT • รีจีสเตอร์-S • รหัสรายงานการตอบสนองคำสั่งของโมเด็ม

คำสั่ง AT สำหรับโมเด็ม

นอกจากคำสั่งพิเศษ คำสั่งทุกคำสั่งต้องเริ่มต้นด้วยรหัสเตือน (หรือคำเติมหน้าคำสั่ง) ซึ่งได้แก่ ตัวอักษร ATและยกเลิกได้ด้วยการกดปุ่ม Enter โมเด็มจะตอบสนองคำสั่งต่าง ๆ ด้วยรหัสรายงานการตอบสนองคำสั่งด้วยคำว่า *OK* ซึ่งหมายความว่า โมเด็มเข้าใจและสามารถปฏิบัติตามคำสั่งนั้นหรือด้วยคำว่า *ERROR* ซึ่งหมายความว่า โมเด็มไม่เข้าใจคำสั่งนั้นหรือผู้ใช้พิมพ์คำสั่งผิด ตรวจสอบให้โมเด็มอยู่ในโหมดคำสั่งเสมอเมื่อป้อนคำสั่งต่าง ๆ ที่ไม่ใช่ชุดลำดับเอสเคปแบบออนไลน์ สำหรับคำสั่งที่ป้อนเข้าสู่โมเด็มในขณะที่โมเด็มอยู่ในโหมดออนไลน์ จะกลายเป็นข้อมูลและถูกส่งไปยังโมเด็มปลายทางในรูปแบบข้อมูล คำสั่งบางคำสั่งจะนำมาใช้ร่วมกับรีจีสเตอร์-S.

คำสั่งพิเศษ

+++ คือ ชุดลำดับเอสเคปแบบออนไลน์

ชุดลำดับเอสเคปนำมาใช้เพื่อสลับจากโหมดออนไลน์ ไปยังโหมดคำสั่งในระหว่างช่วงเวลาที่ติดต่อกับโมเด็มระยะไกล พิมพ์" เครื่องหมายบวก (+) "3 ตัว ไม่จำเป็นต้องป้อนคำเติมหน้าคำสั่ง AT หรือใช้ปุ่ม Enter เมื่อใช้ชุดลำดับเอสเคป การตั้งค่ารีจีสเตอร์-S เป็น S2 เท่ากับค่ารหัสกำหนดตัวอักษร ASCII ที่จะนำมาใช้ S2=43 คือค่าตั้งต้น (หรือเครื่องหมาย ("+" ค่าที่ถูกต้องคือ 0 -127 ถ้าไม่ต้องการให้คำสั่งนั้นทำงาน ให้ตั้ง S2 เป็นค่าที่สูงกว่า 127 ใช้คำสั่งATO ("AT" และตัว" O" ในภาษาอังกฤษ (เพื่อกลับมาที่โหมดออนไลน์ แทรกระยะเว้นวรรคก่อนและหลังชุดลำดับเอสเคปเพื่อป้องกันโมเด็มตีความชุดลำดับเอสเคปว่าเป็นข้อมูล ใช้รีจีสเตอร์-S เป็น S12 เพื่อกำหนดความยาวของระยะเว้นวรรค

A/ คือ ซ้ำคำสั่งสุดท้าย

สำหรับคำสั่ง A/ นั้นมีไว้สำหรับสั่งการให้โมเด็มซ้ำสตริงคำสั่งสุดท้าย โมเด็มจะปฏิบัติตามคำสั่งทันทีที่พิมพ์เครื่องหมาย / ไม่จำเป็นต้องป้อนคำเติมหน้าคำสั่ง AT หรือกดปุ่ม Enter

---

การอ้างถึงคำสั่ง AT

A คือ รับสายโทรศัพท์ การพิมพ์ ATA ทำให้โมเด็มพร้อมที่จะยกหูโทรศัพท์ขึ้น เพื่อติดต่อโมเด็มปลายทางและตอบสนองต่อสายเรียกเข้าด้วยการสร้างสัญญาณพาหะและเริ่มกระบวนการกำหนดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุม (เสียงกระชั้นถี่ของโมเด็ม) คำสั่งนี้ต้องเป็นเพียงคำสั่ง ๆ เดียวหรือคำสั่งสุดท้ายในสายคำสั่ง หาก ขั้นตอนการกำหนดสัญญาณควบคุมประสบความสำเร็จและมีการเชื่อมต่อเกิดขึ้น ข้อความ *CONNECT* จะ ปรากฏ หากไม่สามารถตรวจพบสัญญาณพาหะได้ภายในระยะเวลาที่กำหนดในรีจีสเตอร์-S ที่ตั้งเป็นS7 รหัสรายงานการตอบสนองคำสั่งด้วยคำว่า *NO CARRIER* จะปรากฏ

Bn คือ เลือกมาตรฐานการติดต่อสื่อสาร

คำสั่ง **ATBn** ทำหน้าที่เลือกมาตรฐาน การติดต่อสื่อสาร) ITU หรือ Bell) ที่โมเด็มจะนำมาใช้ ดังราย-ละเอียดต่อไปนี้:

B0	ใช้มาตรฐาน ITU V.22 ที่ 1200bps B0 เลือกมาตรฐาน ITU V.22 ที่ 1200bps และมาตรฐาน ITU V.21 ที่ 300bps
B1	ใช้มาตรฐาน Bell 212A ที่ 1200bps B1 เลือกมาตรฐาน Bell 212A ที่ 1200bps และมาตรฐาน Bell 103J ที่ 300bps
B2	ไม่เลือกช่องรีเวิร์ส V.23
B3	เลือกช่องรีเวิร์ส V.23
B15	เลือก V.21 เมื่อโมเด็มทำงานที่ 300bps (เหมือนกับ B0)
B16	เลือกมาตรฐาน Bell103J เมื่อโมเด็มทำงานที่ 300bps (เหมือนกับ B1)

กระนั้นก็ดี การตั้งค่าพารามิเตอร์คำสั่ง ATB1 และ ATB16 อาจใช้ไม่ได้ในบางประเทศ ค่าตั้งต้นเป็นค่า เฉพาะของแต่ละประเทศ

คำสั่งหยุด) ชุดลำดับเอสเคป(

ดูที่คำสั่งพิเศษ.

Cn คือ การควบคุมสัญญาณพาหะ

คำสั่ง ATCn โดยที่ n คือ 1 รับประกันถึงความเข้ากันได้ของซอฟต์แวร์การติดต่อสื่อสารที่ใช้ ป้อนคำสั่ง C1 คำสั่ง C0 ไม่ได้รับการสนับสนุน เนื่องจากคำสั่งนี้อาจกำหนดให้โมเด็มบางเครื่อง "รับ เฉพาะโหมด"

C0	สัญญาณพาหะสำหรับส่งข้อมูลปิดตลอดเวลา) *ไม่สนับสนุน*)
C1	สลับสัญญาณพาหะสำหรับส่งข้อมูลกลับมาที่ปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dn คือ หมุนหมายเลขโทรศัพท์

คำสั่ง ATDn จะนำมาใช้เพื่อหมุนหมายเลขโทรศัพท์ โดยที่ n แทนสตริงการโทรซึ่งประกอบด้วย 2 ส่วน อันได้แก่ หมายเลขที่จะหมุน/ตัวอักษรและตัวแปลงการหมุนหมายเลขโทรศัพท์ (ดูรายละเอียดด้านล่าง (ทั้งนี้ สตริงการโทรจะต้องประกอบด้วยตัวอักษรไม่เกิน 40 ตัวอักษร การเว้นวรรค เครื่องหมายยติภังค์(-) และเครื่องหมายวงเล็บอาจนำมาใช้เพื่อเหตุผลด้านความชัดเจน แต่โมเด็มจะไม่สนใจเครื่องหมายเหล่านี้ ผู้ใช้สามารถใช้คำสั่งหมุนหมายเลขโทรศัพท์ได้สำหรับการหมุนระบบพัลส์) โรตารี่(หรือการหมุนด้วยสัญญาณโทน ตัวเลขและตัวอักษรสำหรับการโทรคือ 0 - 9 A B C D # * อักษร A B C D และสัญลักษณ์ # และ* แทนคู่สัญญาณพิเศษและนำมาใช้เฉพาะเมื่อเชื่อมต่อโทรศัพท์ด้วยสัญญาณโทน แต่โมเด็มจะไม่สนใจตัวอักษรและสัญลักษณ์เหล่านี้เมื่อโทรด้วยระบบพัลส์ บางประเทศจำกัดหรือห้ามใช้ตัวอักษรเหล่านี้บางตัว

ตัวแปลงการหมุนหมายเลขโทรศัพท์

โมเด็มจะจดจำตัวแปลงการหมุนหมายเลขโทรศัพท์เฉพาะเมื่อตัวแปลงเหล่านี้เป็นส่วนหนึ่งของสตริงการโทรที่ผู้ใช้ป้อนหลังคำสั่ง ATD ตัวแปลงการหมุนหมายเลขโทรศัพท์ที่เป็นไปได้ ประกอบด้วย:

L คือ โทรซ้ำหมายเลขสุดท้าย

โทรซ้ำหมายเลขสุดท้าย หากใช้หมายเลขดังกล่าวเป็นตัวอักษรตัวแรกต่อจากคำสั่ง ATD ไม่เช่นนั้นโมเด็มจะไม่สนใจ

P คือ วิธีหมุนหมายเลขโทรศัพท์แบบพัลส์

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วย P จะนำมาใช้ร่วมกับคำสั่งหมุนหมายเลขโทรศัพท์เพื่อแนะนำให้โมเด็มหมุนหมายเลขโทรศัพท์ที่ตามมาด้วยระบบพัลส์

S=n คือ หมุนหมายเลขโทรศัพท์ที่จัดเก็บไว้

ตัวแปลง S จะแนะนำให้โมเด็มหมุนหมายเลขโทรศัพท์ที่จัดเก็บไว้ด้วยการใช้คำสั่ง AT&Zn=x คำสั่งเพื่อหมุนหมายเลขโทรศัพท์ที่จัดเก็บไว้คือ ATDS=n โดยที่ n แทนตำแหน่งการจัดเก็บหมายเลขโทรศัพท์เป็น 0 หรือ 1 ยกตัวอย่างเช่นคำสั่ง ATD P S=1 เท่ากับเป็นการหมุนหมายเลขโทรศัพท์ที่จัดเก็บไว้ด้วยระบบพัลส์มายังตำแหน่งการจัดเก็บหมายเลขโทรศัพท์ตำแหน่งที่ 1

, คือ หน่วงเวลาการหมุนตัวอักษรตัวถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรวมตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครื่องหมายจุลภาค (,) ไว้เป็นส่วนหนึ่งของสตริงการโทรต่อจากคำสั่ง ATD โมเด็มจะหยุดรอก่อนจะหมุนตัวอักษรตัวถัดไปในสตริงการโทร รีจีสเตอร์-S เป็น S8 จะกำหนดระยะเวลาในการหยุดรอ ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครือข่ายจุลภาค (,) มักถูกแทรกหลังตัวเลข (มักเป็นเลข 9) บ่อยครั้ง เพื่อรับสายนอกจากชุมสายโทรศัพท์ภายใน (PBX) เพื่อปล่อยให้เสียงสัญญาณโทนดังขึ้นช่วงระยะเวลาหนึ่งก่อนที่โมเด็มจะหมุนหมายเลขโทรศัพท์ ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วย W ใช้แทนเครื่องหมายจุลภาคได้

<u>หมายเหตุ:</u> บางประเทศมีข้อจำกัดในเรื่องระยะเวลาที่โมเด็มจะหน่วงเวลาระหว่างที่หมุนหมายเลขโทรศัพท์

T คือ วิธีหมุนหมายเลขโทรศัพท์ด้วยสัญญาณโทน

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วย T จะใช้ร่วมกับคำสั่งหมุนหมายเลขโทรศัพท์เพื่อแนะนำให้โมเด็มหมุนหมายเลขที่ตามมาด้วยสัญญาณโทน ดูรายละเอียดเพิ่มเติมได้ที่คำสั่ง ATT

W คือ รอสัญญาณโทนให้หมุนหมายเลขโทรศัพท์เป็นครั้งที่สอง

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วย W แนะนำให้โมเด็มหยุดรอเสียงสัญญาณโทนก่อนหมุนตัวอักษรที่เหลือในสตริงการโทร

! คือ วางหูโทรศัพท์ชั่วคราว

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครื่องหมาย ! ทำหน้าที่เป็นสัญญาณสลับ (หรือสัญญาณวางหูโทรศัพท์ชั่วคราว) ตัวแปลงนี้ทำให้โมเด็มวางสาย (หรือเข้าสู่สภาวะวางหูโทรศัพท์) นานประมาณ 0.5 วินาทีหลังจากนั้นจึงกลับเข้าสู่สภาวะพร้อมหมุนหมายเลขโทรศัพท์ (ระยะเวลาที่โมเด็มจะเข้าสู่สภาวะวางหูโทรศัพท์ชั่วคราวจะแตกต่างกันในแต่ละประเทศ)

@ คือ รอระยะเวลาที่ไม่มีการรับสาย

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครื่องหมาย @ ในสตริงการโทรจะแนะนำให้โมเด็มรอระยะเวลาที่ไม่มีการรับสายนาน 5 วินาทีหลังจากหมุนหมายเลขโทรศัพท์ หากไม่พบ สัญญาณเงียบ โมเด็มจะส่งรหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า NO ANSWER ให้กับผู้ใช้

; คือ กลับไปยังโหมดคำสั่งหลังจากหมุนหมายเลขโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครื่องหมายอัฒภาค (;) จะใช้ได้เฉพาะตรงส่วนท้ายสุดของสายคำสั่ง ก่อนกดปุ่ม Enter ตัวแปลงนี้แนะนำให้โมเด็มกลับไปยังโหมดคำสั่งทันทีหลังจากหมุนหมายเลขโทรศัพท์โดยไม่ตัดการเชื่อมต่อกับโมเด็มระยะไกล

^ คือ ปิดการทำงานของระบบส่งข้อมูลด้วยสัญญาณโทน

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครื่องหมาย ^ ในสตริงการโทรจะทำให้สัญญาณโทนเพื่อส่งข้อมูลสำหรับสายปัจจุบันไม่ทำงาน (สัญญาณโทนเพื่อหมุนหมายเลขโทรศัพท์จะเปิดทำงานโดยอัตโนมัติในหลายประเทศ) ตัวแปลงนี้ใช้ไม่ได้ในทุกประเทศ

S คือ การตรวจหาสัญญาณเพื่อโทรด้วยบัตรเครดิต

ตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วยเครื่องหมาย S ในสตริงการโทรแนะนำให้โมเด็มรอเสียงสัญญาณ "บอง" เพื่อโทรด้วยบัตรเครดิตก่อนหมุนตัวอักษรที่เหลือในสตริงการโทร

En คือ สะท้อนคำสั่ง

คำสั่ง ATEn โดยที่ n แทน 0 หรือ 1 ทำหน้าที่กำหนดว่า ควรแสดง (สะท้อน) คำสั่งที่คุณป้อนด้วยแป้นคีย์ให้กับโมเด็มในโหมดคำสั่งไว้บนหน้าจอคอมพิวเตอร์หรือไม่

E0	ไม่ต้องแสดงคำสั่งบนคอมพิวเตอร์
E1	แสดงคำสั่งบนคอมพิวเตอร์ (ค่าตั้งต้น(

หากคำสั่งที่คุณพิมพ์ไม่ปรากฏบนหน้าจอคอมพิวเตอร์ เป็นไปได้ว่า มีการตั้งซอฟต์แวร์ของคุณเพื่อให้รับคำสั่งที่สะท้อนมาจากระบบระยะไกล คุณแก้ไขได้ด้วยการพิมพ์คำสั่ง ATE1 หากคำสั่งที่คุณพิมพ์ปรากฏบนหน้าจอคอมพิวเตอร์พร้อมด้วยตัวอักษรที่ปรากฏซ้ำ ๆ เช่นคำว่า *LLIIKKEETTHHIISS* ให้พิมพ์คำสั่ง ATE0

Hn คือ ตัวเลือกสลับการวางหูโทรศัพท์

คำสั่ง ATHn จะทำให้โมเด็มวางหูโทรศัพท์หรือเตรียมโมเด็มให้พร้อมสำหรับการหมุนหมายเลขโทรศัพท์

ATH0	ทำให้โมเด็มวางหูโทรศัพท์) ค่าตั้งต้น(
ATH1	ทำให้โมเด็มยกหูโทรศัพท์ขึ้น (ห้ามใช้คำสั่งนี้ในบางประเทศ(

In คือ ขอข้อมูลเกี่ยวกับ ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง ATI มีตัวเลือกมากมายที่นำมาใช้เพื่อแนะนำให้โมเด็มแสดงข้อมูลเฉพาะเกี่ยวกับโมเด็มให้กับผู้ใช้

I0	แสดงข้อมูลการแก้ไขชุดคำสั่งเฟิร์มแวร์ที่ทำหน้าที่ควบคุมโมเด็ม (เหมือนกับ 13)
I1	คำนวณเช็คซัมของ ROM และแสดงข้อมูลส่วนนี้บน DTE (ตัวอย่างเช่น 12AB)
I2	ตรวจสอบ ROM พร้อมทั้งคำนวณและตรวจทานเช็คซัม ก่อนจะรายงานด้วยคำว่า OK หรือ ERROR
I3	แสดงข้อมูลการแก้ไขชุดคำสั่งเฟิร์มแวร์ของโมเด็ม
I12	ส่งคืนรหัสประเทศ) ตัวอย่างเช่น อเมริกาเหนือ(

Ln คือ ควบคุมเสียงของลำโพง

คำสั่ง ATLn โดยที่ n แทน 0, 1, 2, หรือ 3 ที่ปรากฏในข้อมูลส่วนนี้ เพื่อประโยชน์ในการอ้างอิงความเข้ากันได้เท่านั้น คอมพิวเตอร์ทำหน้าที่ควบคุมเสียงของลำโพงหน้าจอคอมพิวเตอร์ ไม่ใช่โมเด็ม

Mn คือ การเลือกเปิด/ปิดลำโพง

คำสั่ง ATMn โดยที่ n แทน 0, 1, 2, หรือ 3 เปิดการทำงานหรือปิดการทำงานของระบบส่งสัญญาณเสียงจากโมเด็มไปที่ลำโพงคอมพิวเตอร์ (เพื่อให้เกิดเสียง ลำโพงคอมพิวเตอร์ต้องเปิดทำงานด้วย)

M0	เสียงลำโพงปิดอยู่ตลอดเวลา
M1	เปิดเสียงลำโพงจนกระทั่งตรวจพบสัญญาณพาหะ
M2	เสียงลำโพงเปิดเมื่อโมเด็มอยู่สภาวะพร้อมหมุนหมายเลขโทรศัพท์
M3	ปิดเสียงลำโพงขณะที่หมุนหมายเลขโทรศัพท์ หลังจากนั้นจึงเปิดเสียงจนกระทั่งตรวจพบสัญญาณพาหะ

Nn คือ การติดต่อตัวเลือกกระบวนการเพื่อกำหนดสัญญาณควบคุม

คำสั่ง ATNn โดยที่ n แทน 0 หรือ 1 กำหนดว่าโมเด็มต้นทางควรจะแสดงให้ผู้ใช้ทราบถึงกระบวนการติดต่อเชื่อมโยงกับโมเด็มปลายทางหากความเร็วของโมเด็มปลายทางแตกต่างจากโมเด็มต้นทางหรือไม่

N0	เมื่อทำหน้าที่สร้างการเชื่อมต่อหรือรับการเชื่อมต่อ สัญญาณการติดต่อเชื่อมโยงจะปรากฏเป็นสัญญาณมาตรฐานตามที่ระบุด้วยรีจีสเตอร์-S เป็นS37 และเมื่อผู้ใช้ได้เลือกตัวเลือกคำสั่ง **ATBn** ไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N1	เมื่อทำหน้าที่สร้างการเชื่อมต่อหรือรับการเชื่อมต่อ สัญญาณการติดต่อเชื่อมโยงจะปรากฏเป็นสัญญาณมาตรฐานตามที่ระบุด้วยรีจีสเตอร์-S เป็นS37 และเมื่อผู้ใช้ได้เลือกตัวเลือกคำสั่ง **ATBn** ไว้แล้ว ในระหว่างกระบวนการกำหนดสัญญาณควบคุม โมเด็มจะถอยกลับมาที่ความเร็วต่ำลงได้หากผู้ใช้ต้องการ (ค่าตั้งต้น(

On คือ คำสั่งออนไลน์

หากมีการสลับโมเด็มมาที่โหมดคำสั่ง การพิมพ์คำสั่ง ATO0 จะทำให้โมเด็มกลับมาอยู่ที่โหมดออนไลน์ขณะที่ยังมีเชื่อมต่ออยู่

O0	แนะนำให้โมเด็มออกจากโหมดคำสั่งออนไลน์และกลับไปที่โหมดข้อมูล (ดูรายละเอียดจากชุดลำดับเอสเคป +++ ใต้หัวข้อคำสั่งพิเศษ)
O1	การตั้งค่านี้คือ การป้อนคำสั่ง retrain ก่อนย้อนกลับมาที่โหมดข้อมูลออนไลน์
O2	การตั้งค่านี้คือ การป้อนคำสั่งการติดต่อเชื่อมโยงแบบเรตก่อนย้อนกลับมาที่โหมดข้อมูลออนไลน์

P คือ เลือกหมุนหมายเลขโทรศัพท์แบบพัลส์

คำสั่ง ATP แนะนำให้โมเด็มใช้ระบบพัลส์ (โรตารี่) ในการหมุนหมายเลขโทรศัพท์ โหมดนี้ยังทำงานต่อไปสำหรับกระบวนการหมุนหมายเลขโทรศัพท์ทั้งหมด เว้นแต่จะมีการป้อนคำสั่ง ATT หรือปรากฏตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วย T ในสตริงการโทร

Qn คือ ตัวเลือกแสดงรหัสรายงานการตอบสนองคำสั่งของโมเด็ม

การตั้งค่าคำสั่ง ATQn โดยที่ n แทน 0 หรือ 1 ทำหน้าที่กำหนดว่ารหัสรายงานการตอบสนองคำสั่งของโมเด็ม (เช่น OK, CONNECT, RING, NO CARRIER, และ ERROR) ควรปรากฏบนหน้าจอคอมพิวเตอร์หรือไม่

Q0	การแสดงรหัสรายงานการตอบสนองคำสั่งของโมเด็มทำงาน
Q1	การแสดงรหัสรายงานการตอบสนองคำสั่งของโมเด็มไม่ทำงาน

คำสั่งซ้ำ

ดูที่ คำสั่งพิเศษ

T คือ เลือกหมุนหมายเลขโทรศัพท์ด้วยสัญญาณโทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง ATT แนะนำให้โมเด็มใช้สัญญาณโทนในการหมุนหมายเลขโทรศัพท์ โหมดนี้ยังทำงานต่อไปสำหรับกระบวนการหมุนหมายเลขโทรศัพท์ทั้งหมด เว้นแต่จะมีการป้อนคำสั่ง ATT หรือปรากฏตัวแปลงการหมุนหมายเลขโทรศัพท์ด้วย P ในสตริงการโทร ค่าตั้งต้นจากโรงงานคือ การหมุนหมายเลขโทรศัพท์ด้วยสัญญาณโทน

Vn คือ ตัวเลือกการจัดรูปแบบรหัสรายงานการตอบสนองคำสั่งของโมเด็ม

คำสั่ง ATVn โดยที่ n แทน 0 หรือ 1 ทำหน้าที่กำหนดว่าควรแสดงรหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยรหัสที่เป็นตัวเลข (แบบย่อ) หรือตัวอักษร (แบบเต็ม) รหัสแบบตัวเลขจะประกอบด้วยตัวเลข 1 หรือ 2 ตัว และผู้ใช้จะสามารถใช้รูปแบบนี้เมื่อโปรแกรมที่ทำหน้าที่ควบคุมโมเด็มคือ โปรแกรมอีมูลเลชั่นเทอร์มินัลซอฟต์แวร์ที่ใช้แฟ้มสคริปต์ ศึกษารายการรหัสรายงานการตอบสนองคำสั่งของโมเด็มแบบย่อและแบบเต็มได้จากรายละเอียดที่ปรากฏในตอนต้น

พิมพ์คำสั่ง ATV หรือ ATV0 เพื่อเลือกรหัสแบบตัวเลข ค่าตั้งต้นจากโรงงานจะแสดงรหัสเป็นตัวอักษร (ATV1) คุณอาจเลือกป้อนคำสั่ง ATV1 เพื่อรีเซ็ตค่าตั้งต้นจากโรงงานหลังจากเปลี่ยนแปลงการตั้งค่านี้ไปก่อนหน้าแล้วหรือเลือกรหัสแบบเต็ม (ตัวอักษร) ข้อความแสดงขั้นตอนการติดต่อเชื่อมโยง (รหัสรายงานการตอบสนองคำสั่งของโมเด็มแบบเพิ่มเติม) คือ ข้อความที่ประกอบด้วยค่าที่เป็นตัวเลขตั้งแต่ 40 ตัวขึ้นไป

คำสั่ง AT อีก 4 คำสั่ง พร้อมทั้งตัวแปลงการหมุนหมายเลขโทรศัพท์ 2 ตัวแปลงและค่ารีจีสเตอร์-S มีส่วนเกี่ยวข้อง โดยตรงในขั้นตอนการสร้างและแสดงรหัสรายงานการตอบสนองคำสั่งของโมเด็ม คำสั่ง AT เหล่านั้นได้แก่ ATQn, ATVn, ATWn, และ ATXn ตัวแปลงการหมุนหมายเลขโทรศัพท์ ATDW และ ATD@ และค่ารีจีสเตอร์-S เป็น S95

Wn คือ การเลือกข้อความแสดงขั้นตอนการติดต่อเชื่อมโยง

คำสั่ง ATWn โดยที่ n แทน 0, 1, หรือ 2 จะทำงานร่วมกับรีจีสเตอร์-S เป็น S95 เพื่อทำหน้าที่กำหนดว่าควรจะใช้รหัสย่อยของรหัสรายงานการตอบสนองคำสั่งของโมเด็ม หรือที่เรียกว่า ข้อความแสดงขั้นตอนการติดต่อเชื่อมโยงหรือรหัสรายงานการตอบสนองคำสั่งของโมเด็มแบบเพิ่มเติมอย่างไร เพื่อรายงานชนิดของการเชื่อมต่อ โปรโตคอลและเทคนิคการติดต่อสื่อสารอื่น ๆ ที่เกิดขึ้นจากกระบวนการเพื่อกำหนดสัญญาณควบคุมและการติดต่อเชื่อมโยงที่เกิดขึ้นในภายหลัง

ผู้ใช้สามารถแทน n ในคำสั่ง ATWn ได้ด้วยตัวเลือกต่าง ๆ ดังต่อไปนี้:

W0	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" CONNECT" จะรายงานความเร็วของ DTE หากตั้งค่ารีจีสเตอร์-S เป็น S95=0 จะทำให้ฟังก์ชันรหัสรายงานการตอบสนองคำสั่งของโมเด็มแบบเพิ่มเติมไม่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

W1	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" CONNECT" จะรายงานความเร็วของ DTE หากตั้งค่ารีจีสเตอร์-S เป็น S95=0 เฉพาะรหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า "CARRIER" และ" PROTOCOL" เท่านั้นที่จะทำงาน
W2	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" CONNECT" จะรายงานความเร็วของ DCE (โมเด็ม-ถึง-โมเด็ม (หากตั้งค่ารีจีสเตอร์-S เป็น S95=0 จะทำให้ฟังก์ชันรหัสรายงานการตอบสนองคำสั่งของโมเด็มแบบเพิ่มเติมทั้งหมดไม่ทำงาน

Xn คือ ตัวเลือกแสดงชนิดของรหัสรายงานการตอบสนองคำสั่งของโมเด็มเป็นเซ็ต/โทร

คำสั่ง ATXn โดยที่ n แทน 0-4 ทำหน้าที่ควบคุมวิธีที่โมเด็มจะตอบสนองต่อสัญญาณโทนและสัญญาณไม่ว่างและวิธีที่โมเด็มจะแสดงรหัสรายงานการตอบสนองคำสั่งด้วยคำว่า "CONNECT" ผู้ใช้สามารถระบุคำสั่ง ATXn ได้ด้วยตัวเลือกต่าง ๆ ดังต่อไปนี้:

X0	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" 0-4" ทำงาน การตรวจสอบสัญญาณไม่ว่างและสัญญาณโทนให้หมุนหมายเลขโทรศัพท์ไม่เปิดทำงาน
X1	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" 0-5" และ" 10" ทำงาน การตรวจสอบสัญญาณไม่ว่างและสัญญาณโทนให้หมุนหมายเลขโทรศัพท์ไม่เปิดทำงาน
X2	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" 0-6" และ" 10" การตรวจสอบสัญญาณไม่ว่างและสัญญาณโทนให้หมุนหมายเลขโทรศัพท์เปิดทำงาน
X3	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" 0-5" , "7", และ "10" เปิดทำงาน การตรวจสอบสัญญาณไม่ว่างเปิดทำงาน ขณะที่การตรวจสอบสัญญาณโทนให้หมุนหมายเลขโทรศัพท์ไม่เปิดทำงาน
X4	รหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า" 0-7" และ "10"ทำงาน การตรวจสอบสัญญาณไม่ว่างและสัญญาณโทนให้หมุนหมายเลขโทรศัพท์เปิดทำงาน

ข้อควรระวัง: บางประเทศไม่อนุญาตให้ปิดการทำงานของตัวเลือกการตรวจสอบสัญญาณไม่ว่างและสัญญาณโทนให้หมุนหมายเลขโทรศัพท์

Yn คือ ตัวเลือกวางสายหากมีระยะเว้นวรรคนาน

คำสั่ง ATYn โดยที่ n แทน 0 หรือ 1 ทำหน้าที่กำหนดว่าโมเด็มควรจะวางสายที่โทรอยู่หรือไม่เมื่อโมเด็มรับสัญญาณที่มีระยะเว้นวรรคนาน ๆ (หยุดประมาณ 1.6 วินาที) ในระหว่างการเชื่อมต่อด้วยมาตรฐาน V.22bis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Y0	ทำให้ตัวเลือกการวางสายหากมีระยะเว้นวรรคนาน ๆ ไม่เปิดทำงาน (สนับสนุนเฉพาะเมื่อต้องการอ้างอิงความเข้ากันได้เท่านั้น(
Y1	ทำให้ตัวเลือกการวางสายหากมีระยะเว้นวรรคนาน ๆ เปิดทำงาน (*ไม่สนับสนุน*)

Zn คือ เรียกใช้ฐานข้อมูลที่จัดเก็บไว้

คำสั่ง ATZn โดยที่ n แทน 0 จะวางทุกสายที่กำลังอยู่ในขั้นตอนการเชื่อมต่อและโหลดโพรไฟล์การจัดรูปแบบผู้ใช้ที่จัดเก็บไว้ในหน่วยความจำ NVRAM ในฐานะโพรไฟล์การจัดรูปแบบที่กำลังเปิดใช้งานขึ้นมาอีกครั้งหนึ่ง

Z0	วางสายและโหลดโพรไฟล์ที่อยู่ในตำแหน่งการจัดเก็บ 0 ในฐานะโพรไฟล์การจัดรูปแบบที่กำลังเปิดใช้งานขึ้นมาอีกครั้งหนึ่ง

&Bn V.32 คือ ตัวเลือกการรีเทรนโดยอัตโนมัติ

โมเด็ม Xircom มักทำการรีเทรนเสมอ ๆ ไม่สามารถปิดการทำงานของคุณสมบัติการรีเทรนโดยอัตโนมัติได้ไม่ว่ากรณีใด ๆ

&B0	ความสามารถในการรีเทรนอัตโนมัติด้วยมาตรฐาน V.32 ไม่เปิดทำงาน (ไม่สนับสนุน)
&B1	ความสามารถในการรีเทรนอัตโนมัติด้วยมาตรฐาน V.32 เปิดทำงาน (สนับสนุนเฉพาะเมื่อต้องการอ้างอิงความเข้ากันได้เท่านั้น)

&Cn คือ ตรวจสอบสัญญาณพาหะข้อมูล (DCD)

คำสั่ง AT&Cn โดยที่ n แทน 0 หรือ 1 จะทำหน้าที่เลือกวิธีที่โมเด็มจะนำมาใช้เพื่อจัดการกับสัญญาณพาหะข้อมูล

&C0	มีการบีบสัญญาณพาหะโดยไม่คำนึงถึงสถานการณ์ของสัญญาณพาหะของโมเด็มระยะไกล
&C1	มีการควบคุมสภาวะของสัญญาณพาหะที่ส่งมาจากโมเด็มระยะไกล สัญญาณ DCD ของโมเด็มท้องถิ่นจะเปิดทำงานเมื่อตรวจพบสัญญาณพาหะของโมเด็มระยะไกลและจะปิดทำงานเมื่อไม่สามารถตรวจพบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	สัญญาณดังกล่าว (ค่าตั้งต้น)

&Dn คือ ตัวเลือกเครื่องเทอร์มินัลพร้อมที่จะติดต่อกับโมเด็ม (DTR)

คำสั่ง AT&Dn โดยที่ n แทน 0-3 ทำหน้าที่ควบคุมวิธีการใช้สัญญาณเครื่องเทอร์มินัลพร้อมที่จะติดต่อกับโมเด็ม (DTR) ของโมเด็ม

&D0	ไม่สนใจสัญญาณ DTR ที่ส่งมาจากคอมพิวเตอร์และกำหนดว่าสัญญาณนี้เปิดทำงานอยู่ตลอดเวลา
&D1	ควบคุมสัญญาณ DTR และเมื่อเกิดกระบวนการส่งสัญญาณ DTR จาก "เปิดสู่ปิด" คำสั่ง AT นี้จะสลับไปยังโหมดคำสั่ง ป้อนรหัสรายงานการตอบสนองคำสั่งของโมเด็มด้วยคำว่า "OK" และทำให้โมเด็มเชื่อมต่ออยู่เช่นนั้น
&D2	ควบคุมสัญญาณ DTR และเมื่อเกิดกระบวนการส่งสัญญาณ DTR จาก "เปิดสู่ปิด" คำสั่ง AT นี้จะทำการวางสายและสลับไปยังโหมดคำสั่ง
&D3	ควบคุมสัญญาณ DTR และเมื่อเกิดกระบวนการส่งสัญญาณ DTR จาก "เปิดสู่ปิด" คำสั่ง AT นี้จะทำการวางสาย เซ็ตโมเด็มขึ้นมาอีกครั้งและสลับไปยังสภาวะเมื่อเริ่มต้นการทำงานเป็นครั้งแรก

&Fn คือ การตั้งค่าเพื่อโหลดจากโรงงาน

คำสั่ง AT&F จะโหลดพารามิเตอร์ค่าตั้งต้นของโรงงานจาก ROM เข้าสู่โพรไฟล์การจัดรูปแบบที่กำลังเปิดใช้งาน พร้อมทั้งแทนที่พารามิเตอร์ที่จัดเก็บไว้ในโพรไฟล์ดังกล่าว คำสั่งนี้สามารถป้อนลงในเครื่องได้ด้วยตัวของมันเอง หากใช้คำสั่งนี้ร่วมกับคำสั่ง AT อื่น ๆ ฟังก์ชันของคำสั่งนี้จะไม่ได้รับการสนใจ

&F0	เรียกใช้การตั้งค่าจากโรงงานในฐานะการจัดรูปแบบที่กำลังเปิดใช้งานอีกครั้งหนึ่ง
&F5	เรียกใช้การตั้งค่าจากโรงงานที่เหมาะสมสำหรับโหมด ETC อีกครั้งหนึ่งในฐานะการจัดรูปแบบที่กำลังเปิดใช้งาน คำสั่งนี้ทำให้โหมด ETC เปิดการทำงาน คำสั่งนี้จะกำหนดการตรวจสอบโทรศัพท์ระบบเซลลูล่าร์ให้โดยอัตโนมัติ ตัวเลือกต่อไปนี้จะกำหนดด้วย &F5:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งค่าจากโรงงาน

ฟังก์ชัน	เครื่องมือ MTC
การแก้ไขข้อผิดพลาดที่เป็น LAPM เพียงอย่างเดียว	\N4
ระดับการส่งที่กำหนดตายตัวสำหรับ โทรศัพท์ระบบเซลลูล่าร์	S92
รอสัญญาณพาหะ = 90 วินาที	S7=90
หน่วงเวลาการสูญเสีย CD = 10 วินาที	S10=100
ฟังก์ชัน FF/FB โดยอัตโนมัติเปิดทำงาน	N/A
เริ่มต้นที่ 9600bps	S40=2

&Gn V.22bis คือ การเลือกการ์ดโทน

ใช้ตัวเลือกนี้เฉพาะการใช้ระหว่างประเทศเท่านั้น ไม่ใช้ในอเมริกาเหนือ คำสั่ง AT&Gn ทำหน้าที่กำหนดว่าควรจะส่งการ์ดโทนใดขณะที่อยู่ในโหมดตอบรับ (ส่งด้วยความถี่สูง) ค่าของ n คือ 0, 1, หรือ 2 จะมีการกำหนดพารามิเตอร์นี้โดยอัตโนมัติสำหรับทุกประเทศที่ต้องการ

&G0	ไม่มีการกำหนดการ์ดโทน
&G1	การควบคุมการไหลของข้อมูลเป็น RTS/CTS เปิดการทำงาน (ค่าตั้งต้น)
&G2	กำหนดการ์ดโทนเป็น 1800-Hz

&Jn คือ ตัวเลือกรีเลย์สำรอง

&J0	รีเลย์สำรองจะไม่ถูกปิด
&J1	ไม่สนับสนุน (ส่งข้อความ "ERROR")

&Kn คือ ตัวเลือกการควบคุมการไหลของข้อมูลของโมเด็มท้องถิ่น

คำสั่ง AT&Kn โดยที่ n แทน 0-4 ทำหน้าที่กำหนดวิธีการควบคุมการไหลของข้อมูลระหว่างคอมพิวเตอร์และโมเด็มท้องถิ่น

&K0	การควบคุมการไหลของข้อมูลท้องถิ่นไม่เปิดการทำงาน
&K3	การควบคุมการไหลของข้อมูลเป็น RTS/CTS เปิดทำงาน (ค่าตั้งต้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

&K4	การควบคุมการไหลของข้อมูลเป็น XON/XOFF เปิดทำงาน

&Mn คือ โหมดการติดต่อสื่อสาร

คำสั่ง AT&Mn โดยที่ n แทน 0-4 ทำหน้าที่กำหนดวิธีการควบคุมการไหลของข้อมูลระหว่างคอมพิวเตอร์และโมเด็มท้องถิ่น

&M0	โหมดอะซินโครนัส (ค่าตั้งต้น สำหรับความเข้ากันได้เท่านั้น)

&Pn คือ ตัวเลือกอัตราส่วนโทร-ถึง-หยุดของการหมุนหมายเลขโทรศัพท์แบบพัลส์

คำสั่ง AT&Pn โดยที่ n แทน 0, 1, หรือ 2, ทำหน้าที่ควบคุมอัตราส่วนของสภาวะพร้อมหมุนหมายเลขโทรศัพท์ (โทร) ถึงสภาวะวางหูโทรศัพท์ (หยุด) ที่โมเด็มใช้เมื่อหมุนโทรศัพท์ด้วยระบบพัลส์

&P0	เลือกอัตราส่วนโทร/หยุดเป็น 39:61 ที่ 10 pps (ค่าตั้งต้น - สหรัฐฯ)
&P1	เลือกอัตราส่วนโทร/หยุดเป็น 33:67 ที่ 10 pps (ค่าตั้งต้น - ญี่ปุ่น)
&P2	เลือกอัตราส่วนโทร/หยุดเป็น 33:67 ที่ 20 pps (ค่าตั้งต้น - ญี่ปุ่น)

&Qn คือ โหมดการติดต่อสื่อสารแบบอะซินโครนัส

&Q0	โหมดอะซินโครนัสถูกบัฟเฟอร์ (เหมือนกับ \N0)
&Q5	โหมดการควบคุมความผิดพลาดถูกบัฟเฟอร์ (ค่าตั้งต้น เหมือนกับ \N3)
&Q6	โหมดอะซินโครนัสถูกบัฟเฟอร์ (เหมือนกับ \N0)

&Sn คือ ตัวเลือกพร้อมส่งข้อมูล (DSR)

คำสั่ง AT&Sn ทำหน้าที่ควบคุมฟังก์ชันต่าง ๆ ของวงจร DSR ในโมเด็ม

&S0	สัญญาณ DSR จะเปิดการทำงานอยู่ตลอดเวลาเมื่อโมเด็มเปิดทำงาน (ค่าตั้งต้น)
&S1	สัญญาณ DSR จะเปิดการทำงานในระหว่างกระบวนการเพื่อกำหนดสัญญาณควบคุมและปิดการทำงานเมื่อสูญเสียสัญญาณพาหะ

&Tn คือ คำสั่งทดสอบตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง AT&Tn อนุญาตให้ผู้ใช้ทำการทดสอบเพื่อตรวจหาข้อบกพร่องบนโมเด็ม

&T0	ล้มเหลว หยุดการทดสอบที่กำลังดำเนินการอยู่
&T1	วงรอบระบบอะนาล็อค การทดสอบนี้จะทำหน้าที่ตรวจสอบการทำงานของโมเด็มรวมถึงการเชื่อมต่อระหว่างโมเด็มและคอมพิวเตอร์ โมเด็มต้องอยู่ในสภาวะคำสั่ง (ออฟไลน์) เมื่อกำลังเปิดการทดสอบนี้
&T3	การทดสอบวงรอบย้อนกลับระบบอะนาล็อค
&T6	การทดสอบวงรอบย้อนกลับระบบดิจิตอลระยะไกล การทดสอบนี้สามารถตรวจสอบความพร้อมของโมเด็มต้นทาง การลิงค์ระบบการติดต่อสื่อสารและโมเด็มระยะไกล เมื่อทำการทดสอบนี้ โมเด็มต้องอยู่ในโหมดออนไลน์ขณะที่การควบคุมการไหลของข้อมูลไม่ทำงาน

&V คือ ดูโพรไฟล์การจัดรูปแบบ

คำสั่ง AT&V แสดงรายละเอียดของโพรไฟล์การจัดรูปแบบที่กำลังเปิดใช้งานอยู่

&Wn คือ เขียนโพรไฟล์ที่กำลังเปิดใช้งานลงไปที่หน่วยความจำ

คำสั่ง AT&Wn โดยที่ n แทน 0 จะช่วยให้คุณบันทึกสำเนาของโพรไฟล์การจัดรูปแบบที่กำลังเปิดใช้งานอยู่ลงใน NVRAM ผู้ใช้สามารถเรียกคืนโพรไฟล์นี้ได้ทุกเวลาด้วยการใช้คำสั่ง ATZ หรือรีเซ็ตพลังงานของโมเด็มกลับขึ้นมาใหม่

&Yn คือ เลือกโพรไฟล์ที่จัดเก็บไว้สำหรับการรีเซ็ต

คำสั่งนี้นำมารวมไว้สำหรับความเข้ากันได้กับโปรแกรมประยุกต์ที่ใช้คำสั่ง &Y0 คำสั่งนี้จะไม่ส่งผลต่อการทำงานของโมเด็ม

&Y0	เลือกโพรไฟล์ที่จัดเก็บไว้ใน 0 ขณะที่รีเซ็ตพลังงานของโมเด็มกลับขึ้นมาใหม่
&Y1	ไม่สนับสนุน ส่งข้อความ "ERROR"

&Zn=x คือ จัดเก็บหมายเลขโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง AT&Zn=x จะนำมาใช้เพื่อจัดเก็บหมายเลขโทรศัพท์เพื่อโทรติดต่อในภายหลังโดยใช้คำสั่ง ATDS=n (หมุนหมายเลขโทรศัพท์ที่จัดเก็บไว้) โดยที่ n ในคำสั่งนี้จะแทน 0 หรือ 1 ซึ่งก็คือตำแหน่งที่จัดเก็บ 2 ตำแหน่งและ x คือหมายเลขโทรศัพท์ที่จัดเก็บไว้ สตริงการโทรประกอบด้วยตัวอักษรไม่เกิน 40 ตัวอักษร

\Gn คือ การควบคุมการไหลของข้อมูลบนพอร์ตโมเด็ม

คำสั่ง AT\Gn ทำหน้าที่กำหนดว่าการควบคุมการไหลของข้อมูลเป็นควรอยู่ในสถานะ XON/XOFF

\G0	ส่งข้อความ "OK" สำหรับความเข้ากันได้ (ค่าตั้งต้น)
\G1	ไม่สนับสนุน ส่งข้อความ "ERROR"

\Jn คือ ปรับอัตราการควบคุมเป็น BPS

\J0	ปิดคุณสมบัตินี้ (ค่าตั้งต้น)
\J1	เปิดคุณสมบัตินี้

\Kn คือ ยุติการควบคุม

คำสั่ง AT\Kn กำหนดวิธีที่โมเด็มจะใช้เพื่อจัดการกับสัญญาณหยุดที่รับจาก DTE ท้องถิ่นระหว่างขั้นตอนการเชื่อมต่อ (ออนไลน์)

\K5	โมเด็มจะส่งสัญญาณหยุดให้กับโมเด็มระยะไกลเป็นลำดับพร้อมด้วยข้อมูลที่ส่ง ชนิดไม่มีข้อผิดพลาด/ไม่เร่งจังหวะ (ค่าตั้งต้น)

\Nn คือ การเลือกโหมดควบคุมความผิดพลาด

คำสั่ง AT\Nn กำหนดชนิดของการแก้ไขข้อผิดพลาดที่โมเด็มสนับสนุนเมื่อส่งหรือรับข้อมูล

\N0	โหมดบัฟเฟอร์ ไม่มีการควบคุมความผิดพลาด (เหมือนกับ &Q6)
\N1	โหมดบัฟเฟอร์ (เหมือนกับ \N0)
\N2	LAPM, MNP โหมดหยุดการเชื่อมต่อ หรือที่รู้จักกันในอีกชื่อว่าโหมดที่เชื่อถือได้
\N3	LAPM, MNP, หรือบัฟเฟอร์ (ค่าตั้งต้น) โมเด็มพยายามเชื่อมต่อในโหมดควบคุมความผิดพลาด LAPM หากล้มเหลว โมเด็มจะพยายามเชื่อมต่อใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	โหมด MNP หากยังล้มเหลวอีก โมเด็มก็จะพยายามเชื่อมต่อในโหมดบัฟเฟอร์และทำงานต่อไป หรือที่รู้จักกันในชื่อ โหมดที่เชื่อถือได้โดยอัตโนมัติด้วยมาตรฐาน V.42 (เหมือนกับ &Q5)
\N4	LAPM หรือหยุดการเชื่อมต่อ โมเด็มพยายามเชื่อมต่อในโหมดควบคุมความผิดพลาด LAPM หากล้มเหลว สายจะถูกตัด
\N5	MNP หรือโหมดหยุดการเชื่อมต่อ โมเด็มพยายามเชื่อมต่อโดยใช้ขั้นตอนการควบคุมข้อผิดพลาดแบบ MNP 2-4 หากล้มเหลว โมเด็มจะตัดการเชื่อมต่อ หรือที่รู้จักกันในชื่อ โหมดที่เชื่อถือได้ MNP

\Qn คือ การเลือกการควบคุมการไหลของข้อมูลในพื้นที่

คำสั่ง AT\Qn กำหนดประเภทของการควบคุมการไหลของข้อมูลที่ใช้บนพอร์ตซีเรียลเพื่อปรับความเร็วพอร์ตโมเด็มที่แตกต่างกัน

\Q0	ทำให้ฟังก์ชันควบคุมการไหลของข้อมูลไม่ทำงาน (เหมือนกับ &K0)
\Q1	กำหนดฟังก์ชันควบคุมการไหลของข้อมูลเป็น XON/XOFF (เหมือนกับ &K4)
\Q3	RTS/CTS เป็น DTE (ค่าตั้งต้น เหมือนกับ &K3)

\Tn คือ ตัวนับเวลาหยุดทำงานก่อนตัดการเชื่อมต่อ

คำสั่ง AT\Tn ระบุช่วงเวลา (เป็นนาที) ที่โมเด็มจะรอก่อนตัดการเชื่อมต่อเมื่อไม่มีการส่งหรือรับข้อมูล ช่วงเวลากำหนดเป็น n = 0 - 255 การตั้งค่าเป็น 0 จะทำให้ตัวนับเวลาไม่ทำงาน คุณสามารถเลือกกำหนดตัวนับเวลาในรีจิสเตอร์-S เป็น S30 ฟังก์ชันนี้จะใช้ได้เฉพาะในโหมดบัฟเฟอร์

\Xn XON/XOFF คือ การติดต่อได้

\X0	โมเด็มควบคุมการไหลของข้อมูลเป็น XON/XOFF ในพื้นที่ (ค่าตั้งต้น)
\X1	ไม่สนับสนุน ส่งข้อความ "ERROR"

%B คือ ดูหมายเลขโทรศัพท์ในรายการความผิดพลาด

หากเปิดใช้งานรายการความผิดพลาดอยู่ คำสั่ง AT%B จะแสดงหมายเลขโทรศัพท์ที่ผู้ใช้พยายามโทรในช่วง 2 ชั่วโมงสุดท้ายแต่ไม่ประสบความสำเร็จ ในประเทศที่ไม่ต้องใช้รายการความผิดพลาด คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้จะส่งข้อความ "ERROR" กลับมายังผู้ใช้

%Cn คือ ควบคุมการบีบขนาดข้อมูล

คำสั่ง AT%Cn ทำหน้าที่กำหนดการทำงานของการบีบขนาดข้อมูลมาตรฐาน V.42bis และ MNP Class 5 การเปลี่ยนแปลงออนไลน์จะไม่เริ่มทำงานจนกว่าผู้ใช้จะหยุดการเชื่อมต่อ

%C0	การบีบขนาดข้อมูลด้วยมาตรฐาน V.42bis/MNP Class 5 ไม่ทำงาน (ไม่มีการบีบขนาดข้อมูล)
%C1	การบีบขนาดข้อมูลด้วยมาตรฐาน MNP Class 5 ทำงาน (ไม่ใช่ด้วยมาตรฐาน V.42bis)
%C2	การบีบขนาดข้อมูลด้วยมาตรฐาน V.42bis ทำงาน (ไม่ใช่ด้วยมาตรฐาน MNP Class 5)
%C3	การบีบขนาดข้อมูลด้วยมาตรฐาน V.42bis/MNP Class 5 ทำงาน (ค่าตั้งต้น)

-Cn คือ สัญญาณโทนสำหรับสายข้อมูล

สัญญาณโทนสำหรับสายข้อมูลคือ สัญญาณโทนที่มีคลื่นความถี่และจังหวะที่กำหนดเป็นมาตรฐาน V.25 ที่อนุญาตให้แยกแยะสายข้อมูล/แฟกซ์/สนทนาจากระยะไกล ความถี่ 1300 Hz ด้วยจังหวะ 0.5 วินาทีและ 2 วินาทีปิดการทำงาน

-C0	สัญญาณโทนสำหรับสายข้อมูลไม่เปิดการทำงาน (ค่าตั้งต้น)
-C1	สัญญาณโทนสำหรับสายข้อมูลเปิดทำงาน

ข้อควรระวัง: บางประเทศไม่อนุญาตให้ปิดการทำงานของสัญญาณโทนสำหรับสายข้อมูล

+GCAP คือ ขอให้แสดงรายละเอียดความสามารถของโมเด็ม :

การสร้างประโยค:	AT+GCAP
	AT+GCAP=?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งนี้แสดงรายละเอียดในรูปแบบมาตรฐานซึ่งอาจเป็นหนึ่งบรรทัดหรือมากกว่านั้นเพื่ออธิบายความสามารถขั้นพื้นฐานของโมเด็ม คำสั่งนี้จะช่วยให้ซอฟต์แวร์ทั้งชุดสามารถกำหนดได้ว่าโมเด็มสนับสนุนกลุ่มคำสั่งที่ปรากฏเป็นประโยคเพิ่มเติมกลุ่มใด การตอบสนองของโมเด็มอาจปรากฏเป็นหนึ่งในรูปแบบดังต่อไปนี้:

+CGSM	คำสั่ง GSM (+C)
+DS	คำสั่งบีบขนาดข้อมูล (+D)
+ES	คำสั่งการควบคุมความผิดพลาด (+E)
+FCLASS	คำสั่งแฟกซ์ (+F)
+MS	คำสั่งควบคุมการโมดูเลชั่น (+M)
+W	คำสั่งการใช้ระบบไร้สาย (+W)

+GCAP=? อาจนำมาใช้เพื่อกำหนดว่าโมเด็มจะสนับสนุนคำสั่ง +GCAP หรือไม่: การตอบสนองด้วยคำว่า "OK" หมายถึงสนับสนุน ส่วนคำว่า "ERROR" หมายถึงทำตามคำสั่งนั้นไม่สำเร็จ

+GCI คือ การเลือกประเทศ

ประโยค:	AT+GCI=<T.35 code>
	AT+GCI?
	AT+GCI=?

คำสั่งนี้ทำหน้าที่จัดรูปแบบโมเด็มสำหรับประเทศที่ใช้ การเลือกพารามิเตอร์สำหรับทำงานและการยืนยันตามเงื่อนไขของเครือข่ายโทรศัพท์ของประเทศที่เลือก พารามิเตอร์ +GCI อาจจะเปลี่ยนแปลงเฉพาะเมื่อโมเด็มอยู่ในสภาวะว่าง

<T.35 code> อ้างถึงตัวเลขทศนิยม 6 หลักแบบ 8-bit ที่อยู่ถัดจากประเทศที่ปรากฏในรายงานดังต่อไปนี้

หมายเหตุ: แนะนำให้ใช้อรรถประโยชน์ Xircom CountrySelect สำหรับการตั้งค่าพารามิเตอร์สำหรับการโทรด้วยโมเด็ม ผู้ใช้สามารถเข้าถึงอรรถประโยชน์ CountrySelect ได้หลังจากการติดตั้ง โดยคลิกที่แถบ "เริ่ม" (Start) เลือก "โปรแกรม" (Programs) และ Xircom Utilities ทุกประเทศในรายการดังต่อไปนี้ไม่ได้รับการสนับสนุน ถ้าต้องการกำหนดประเทศที่ได้รับการสนับสนุน ให้ใช้อรรถประโยชน์ Xircom CountrySelect หรือใช้คำสั่ง AT+CGI=? คำสั่ง AT+CGI=? จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบสนองด้วยรหัส T.35 สำหรับประเทศที่ให้การสนับสนุน

ถ้าต้องการกำหนดการตั้งค่าประเทศที่ใช้อยู่ในปัจจุบัน ให้ใช้คำสั่ง AT+GCI? คำสั่งจะตอบสนองด้วยรหัส T.35 สำหรับประเทศที่เลือก ถ้าต้องการเปลี่ยนการเลือกประเทศที่ใช้อยู่ในปัจจุบัน ให้ใช้คำสั่ง AT+GCI=<T.35 code>

ประเทศ	<T.35 code>	ประเทศ	<T.35 code>
ออสเตรเลีย	09	ลักเซมเบิร์ก	69
ออสเตรีย	0A	ญี่ปุ่น	00
เกาะบาร์เบโดส	0E	เกาหลี	61
เบลเยี่ยม	0F	มาเลเซีย	6C
แคนาดา	20	เนเธอร์แลนด์	7B
สาธารณรัฐเชค	2E	นิวซีแลนด์	7E
จีน	26	นอรเวย์	82
เดนมาร์ก	31	โปแลนด์	8A
ฟินแลนด์	3C	โปรตุเกส	8B
ฝรั่งเศส	3D	สาธารณรัฐสโลวาเกีย	2E
เยอรมนี	04	แอฟริกาใต้	9F
กรีก	46	สิงคโปร์	9C
เกาะกวม	48	สเปน	A0
ฮังการี	51	สวีเดน	A5
ฮ่องกง	50	สวิตเซอร์แลนด์	A6
เกาะไอซ์แลนด์	52	ไต้หวัน	FE
อินโดนีเซีย	54	ไทย	A9
ไอร์แลนด์	57	สหราชอาณาจักร	B4
อิตาลี	59	สหรัฐฯ	B5

+GMI คือ ขอให้แสดงรายละเอียดเกี่ยวกับผู้ผลิต

ประโยค:	AT+GMI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	AT+GMI=?

คำสั่งนี้แสดงรายละเอียดเกี่ยวกับผู้ผลิตโมเด็มซึ่งอาจปรากฏเป็นหนึ่งบรรทัดหรือมากกว่านั้น +GMI=? อาจนำมาใช้เพื่อตัดสินว่า อะแด็ปเตอร์สนับสนุนคำสั่ง +GMI หรือไม่: การตอบสนองด้วยคำว่า "OK" หมายถึงสนับสนุน ส่วนคำว่า "ERROR" หมายถึงทำตามคำสั่งนั้นไม่สำเร็จ

+GMM คือ ขอให้แสดงรายละเอียดเกี่ยวกับรุ่นผลิตภัณฑ์:

ประโยค:	AT+GMM
	AT+GMM=?

คำสั่งนี้แสดงรายละเอียดเกี่ยวกับรุ่นของโมเด็มซึ่งอาจปรากฏเป็นหนึ่งบรรทัดหรือมากกว่านั้น +GMM=? อาจนำมาใช้เพื่อตัดสินว่า โมเด็มสนับสนุนคำสั่ง +GMM หรือไม่: การตอบสนองด้วยคำว่า "OK" หมายถึงสนับสนุน ส่วนคำว่า "ERROR" หมายถึงทำตามคำสั่งนั้นไม่สำเร็จ

+GMR คือ ขอให้แสดงรายละเอียดเกี่ยวกับการปรับปรุงแก้ไข:

ประโยค:	AT+GMR
	AT+GMR=?

คำสั่งนี้แสดงรายละเอียดเกี่ยวกับระดับการปรับปรุงแก้ไขเฟิร์มแวร์ของโมเด็มซึ่งอาจปรากฏเป็นหนึ่งบรรทัดหรือมากกว่านั้น +GMR=? อาจนำมาใช้เพื่อตัดสินว่า อะแด็ปเตอร์สนับสนุนคำสั่ง +GMR หรือไม่: การตอบสนองด้วยคำว่า "OK" หมายถึงสนับสนุน ส่วนคำว่า "ERROR" หมายถึงทำตามคำสั่งนั้นไม่สำเร็จ

+MS คือ การเลือกโมดูเลชั่น

ข้อควรระวัง: ผู้ใช้จะไม่สามารถใช้คำสั่ง AT+MS ได้เมื่อโมเด็มขนาด 56K ใช้มาตรฐาน V.90 ถ้าต้องการกำหนดการทำงานเป็นมาตรฐาน V.90 ให้ดูที่ รีจีสเตอร์-S เป็น S109

พารามิเตอร์ AT+MS ควบคุมการโมดูเลชั่นข้อมูลและอัตราการบิตที่อาจจะนำมาใช้สำหรับการติดต่อเพื่อสร้างการเชื่อมโยงระหว่างโมเด็มในพื้นที่และโมเด็มระยะไกล พารามิเตอร์นี้ยอมรับพารามิเตอร์ย่อยอีก 4 พารามิเตอร์ อันได้แก่

ประโยค	AT+MS=<carrier>,<automode>,<min_rate>,<max_rate>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

:	
	AT+MS?
	AT+MS=?

+MS?	รายงานการตั้งค่าปัจจุบันของพารามิเตอร์ย่อย
+MS=?	แสดงช่วงของค่าที่ยอมรับได้สำหรับพารามิเตอร์ย่อยแต่ละพารามิเตอร์
<carrier>	ระบุพารามิเตอร์ที่ต้องการเพื่อนำมาใช้สร้างการเชื่อมต่อหรือรับการเชื่อมต่อจากโมเด็มอีกเครื่องหนึ่ง พารามิเตอร์ย่อย <carrier> คือสตริงตัวอักษรที่ยังไม่ได้นำมาอ้างอิง หากมีการระบุพารามิเตอร์ <carrier> พารามิเตอร์ย่อย อื่น ๆ จะแปลงกลับเป็นค่าตั้งต้นจากโรงงาน หากยกเว้นพารามิเตอร์ <carrier> พารามิเตอร์อื่น ๆ ที่ไม่ได้ถูกระบุก็จะรักษาค่าปัจจุบันเอาไว้ (ตัวอย่างเช่น AT+MS=,0 หรือ AT+MS=,,,2400)
	ค่าที่พารามิเตอร์ <carrier> ยอมรับมีดังต่อไปนี้: B103 Bell 103 (300bps) B212 Bell 212A (1200bps) V21 ITU-T V.21 (300bps) V22 ITU-T V.22 (1200bps) V22B ITU-T V.22bis (2400bps) V23C ITU-T V.23 พร้อมด้วยการตรวจสอบสัญญาณพาหะอย่างต่อเนื่อง (1200/75 หรือ 75/1200bps) V32 ITU-T V.32 (4800 หรือ 9600bps) V32B ITU-T V.32bis (4800 - 19200bps) V34 ITU-T V.34 (2400 - 33600bps) K56 Lucent/Rockwell K56flex (32000 - 56000bps) ไม่สามารถใช้ค่านี้ได้เมื่ออยู่ภายใต้มาตรฐาน V.90 ถ้าต้องการกำหนดมาตรฐาน V.90 ให้ดูที่ส่วนรีจีสเตอร์-S เป็น S109
<automode>	ทำให้การสร้างการเชื่อมต่อของพารามิเตอร์ <carrier> ที่เลือกเปิดทำงานหรือปิดการทำงาน หากไม่สามารถใช้การโมดูเลชั่นที่ต้องการได้
	ค่าที่พารามิเตอร์ <automode> ยอมรับมีดังต่อไปนี้: 0 ไม่เปิดทำงาน โมเด็มจะหยุดการเชื่อมต่อหากไม่สามารถติดต่อเพื่อสร้างการเชื่อมต่อด้วยพารามิเตอร์ carrier> ที่ระบุ 1 เปิดทำงาน (ค่าตั้งต้น) หากไม่ปรากฏพารามิเตอร์ <carrier> ที่กำหนด โมเด็มจะพยายามติดต่อเพื่อสร้างการเชื่อมต่อกับสัญญาณพาหะที่เลือกอย่างเหมาะสม
<min_rate>	ระบุอัตราการบิตต่ำสุดที่โมเด็มอาจใช้สร้างการเชื่อมต่อ (ค่านี้กำหนดตายตัวเป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	สำหรับโมเด็ม Xircom)
<max_rate>	ระบุอัตราการบิตสูงสุดที่โมเด็มอาจใช้สร้างการเชื่อมต่อ สำหรับการโมดูเลชั่นที่สนับสนุนเฉพาะอัตราการบิตที่กำหนดตายตัว (เช่น V.22bis) พารามิเตอร์ <max_rate> จะมีค่าตายตัวที่จะใช้เป็นค่าตั้งต้น หากระบุอัตราการบิตที่กำหนดเป็นตั้งต้นหรืออัตราอื่นที่ไม่ใช่ศูนย์ โมเด็มจะตอบสนองด้วยคำว่า "ERROR" พารามิเตอร์ย่อยนี้ยอมรับค่าที่ปรากฏตามรายการต่อไปนี้: **0**, 300, 1200, 2400, 4800, 7200, 9600, 12000, 14400, 16800, 19200, 21600, 24000, 31200, 33600bps. หากไม่ได้ระบุค่าที่แน่ชัด (กำหนดเป็น 0) ค่าของพารามิเตอร์ <max_rate> จะถูกกำหนดด้วยค่าของพารามิเตอร์ <carrier> หากตั้งค่าพารามิเตอร์ <carrier> เป็น K56, พารามิเตอร์ย่อย <max_rate> จะคงอยู่ที่ค่าตั้งต้น (ศูนย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1.สัจจะ จรัสรุ่งรวีวร, จักรพงษ์ สุขประเสร็จ, เริ่มต้นอย่างมืออาชีพด้วย Delphi7 ฉบับสมบูรณ์, อินโฟเพรส, นนทบุรี, 2546

2.กนก กุสุมาลย์นุกูล, ไกรวุฒิ มั่นเสถียรสิน, คู่มือการใช้งานโปรแกรม Borland Delphi5 ฉบับเพื่อการใช้งานจริง, บริษัท ซัคเซส มีเดีย จำกัด, กรุงเทพฯ

3.Marco Cantu , Mastering Delphi 7 , SYBEX Inc. , San Francisco , 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้