

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมช่วยสอน

Assistant Training Program



โดย

นาย พิบูลย์

จารุอรียานนท์

นาย สุรัชย์

ทองอรุณศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 36909
วัน, เดือน, ปี 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โปรดแจ้งให้ทราบ หากมีข้อผิดพลาดหรือข้อสงสัยให้ติดต่อแจ้งให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

โปรแกรมช่วยสอน
Assistant Training Program

ชื่อนักศึกษา

นาย พิบูลย์ จารุอริยานนท์ 40013299
นาย สุรชัย ทองอรุณศรี 40013314

อาจารย์ที่ปรึกษา

อาจารย์ มยุรี เลิศเวชกุล
อาจารย์ พิทักษ์ ธรรมวาริน

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2542

คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอนุมัติให้
นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
(.....)

.....กรรมการ
(.....)

.....กรรมการ
(.....)

.....กรรมการ
(.....)

เอกสารนี้ ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมช่วยสอนผ่านทาง COMPUTER (Assistance Training Software)

นาย พิบูลย์ จารุอริยานนท์

นาย สุรชัย ทองอรุณศรี

2542

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการพัฒนาโปรแกรมช่วยสอนผ่านทางเครื่อง Computer ที่ทำงานผ่านระบบเครือข่ายที่ใช้โพรโทคอล TCP/IP ซึ่งในปัจจุบันนี้มีความนิยมและแพร่หลายมาก เนื่องจากในปัจจุบัน Internet ก็ใช้โพรโทคอล TCP/IP ในการติดต่อสื่อสารกันเป็นเครือข่ายขนาดใหญ่

ในการพัฒนาโปรแกรมนี้โดยใช้โปรแกรม Microsoft Visual Basic ที่ทำงานภายใต้ระบบปฏิบัติการ Microsoft Windows 95/98 ผ่านทางระบบ Network ในการติดต่อโดยใช้ Winsock ซึ่งเป็น API (Application Program Interface) ในการติดต่อสื่อสารของโพรโทคอล TCP/IP โดยโปรแกรม Microsoft Visual Basic นี้เป็นโปรแกรมเชิงวัตถุโดยสามารถกำหนดการทำงานของวัตถุ (Object) เพื่อไปสั่งงาน (Driven) หรือกำหนดให้การทำงานอย่างอื่นต่อไปตามคำสั่งที่กำหนดไว้

โปรแกรมนี้จะมีความสามารถในการจับจอภาพของเครื่องผู้สอนในขณะนั้น เพื่อจะส่งไปให้ทางเครื่องปลายทางได้เห็นภาพบนจอภาพเหมือนกับเครื่องของผู้สอนที่ทำการสอนอยู่ทำให้สามารถติดตามการเรียนการสอนได้อย่างต่อเนื่องและมีประสิทธิภาพมากโดยไม่จำเป็นต้องใช้เครื่องฉายภาพ (Overhead Projector) ซึ่งไม่ค่อยสะดวกในการใช้และการแสดงผลก็ไม่ชัดเจนบนจอภาพซึ่งขึ้นอยู่กับคุณภาพของอุปกรณ์นั้นๆ ทำให้การติดตามการเรียนการสอนดูไม่น่าสนใจหรือทำให้ไม่จำเป็นที่จะให้นักเรียนมานั่งดูที่เครื่องของผู้สอน และหวังว่าการพัฒนาโปรแกรมนี้จะทำเกิดประโยชน์ในการเรียนการสอนและสามารถพัฒนาให้มีประสิทธิภาพมากขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

1. บทนำ	1
1.1. จุดมุ่งหมาย	
1.2. แนวความคิดและที่มา	
1.3. วัตถุประสงค์	
1.4. ขอบเขต	2
1.5. แนวทางการพัฒนา	2
2. ความรู้เบื้องต้นเกี่ยวกับระบบเครือข่ายต่าง ๆ	3
2.1. ระบบเครือข่ายและการโปรแกรมบนระบบเครือข่าย	
2.1.1. ความหมายของระบบเครือข่ายคอมพิวเตอร์และอินเทอร์เน็ตเวิร์คกิ้ง	
2.1.2. รูปแบบของโปรแกรมบนระบบเครือข่าย (Network Programming Models) การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing) การประมวลผลแบบกระจาย (Distributed Computing)	
2.1.3. โมเดล OSI	4
2.1.4. เอ็นแคปซูลชัน (Encapsulation)	5
2.1.5. ลักษณะของการติดต่อ	6
Connection-oriented	
Connectionless หรือดาต้าแกรม (Datagram)	
แอดเดรส (Address)	7
2.2. ความรู้เบื้องต้นเกี่ยวกับโปรโตคอล TCP/IP	7
2.2.1. เปรียบเทียบระหว่างโปรโตคอล TCP/IP และ OSI โมเดล	
แอปพลิเคชันเลเยอร์ (Application Layer)	8
ทรานสปอร์ตเลเยอร์ (Transport Layer)	
อินเทอร์เน็ตเลเยอร์ (Internet Layer)	
ฟิสิคอลลเลเยอร์ (Physical Layer)	
2.2.2. รูปแบบการกำหนดแอดเดรสของโปรโตคอล TCP/IP	8

การทำให้เน็ตติง (Subnetting)	10
2.2.3. ชุดโพรโทคอล TCP/IP (TCP/IP Protocol Suite)	11
โพรโทคอล IP (Internet Protocol)	11
โพรโทคอล ARP (Address Resolution Protocol)	11
โพรโทคอล ICMP (Internet Control Message Protocol)	12
โพรโทคอล RARP (Reverse Address Resolution Protocol)	12
โพรโทคอล UDP (User Datagram Protocol)	12
โพรโทคอล TCP (Transmission Control Protocol)	12
2.2.4. หมายเลขพอร์ต	12
2.3. Winsock (Window Socket Application Programming Interface)	13
2.3.1. Berkeley Socket	13
2.3.2. เปรียบเทียบ Berkeley Socket กับ Winsock	13
2.3.3. ส่วนเพิ่มเติมของ Winsock จาก Berkeley Socket	14
สถาปัตยกรรมแมสเสจไดรเวนของวินโดวส์	
ฟังก์ชันการทำงานแบบอะซิงโครนัสของ Winsock	15
2.4. การใช้งาน Winsock Control	15
2.4.1. Winsock TCP Control	16
3. การออกแบบ	17
3.1. หลักในการออกแบบ	17
3.2. ขั้นตอนในการพัฒนา	18
3.2.1. ฟอร์มของส่วนติดต่อผู้ใช้	20
3.2.2. ทุลบ็อกซ์ ToolBox (กล่องเครื่องมือ)	20
3.2.3. หน้าต่างแสดงคุณลักษณะ	21
3.2.4. หน้าต่าง Project	21
3.3. ขั้นตอนในการพัฒนาโปรแกรมของ Visual Basic	22
3.4. การจัดสร้างโปรแกรมช่วยสอน	23
3.4.1. การพัฒนาในส่วนของการรับและส่งข้อมูลผ่านระบบเน็ตเวิร์ค	
3.5. อุปกรณ์และระบบที่ต้องการ (Requirement)	24
3.6. โครงสร้างโดยรวมของโปรแกรม	25

4. การทำงานเพื่อติดต่อผ่านระบบเน็ตเวิร์ค	28
4.1. การพัฒนาในส่วนของการสร้างโปรแกรมติดต่อผ่านระบบเน็ตเวิร์ค	28
4.1.1. Winsock UPD Control	33
4.2. การพัฒนาในส่วนของการโปรแกรมดักจับจอภาพ	35
4.2.1. อะไรคือดีไวซ์คอนเท็กซ์	35
4.2.2. โปรแกรมดักจับจอภาพ (Capture Screen)	36
4.2.3. เพิ่มคำสั่งที่ไม่มีใน VB	37
4.2.4. คำสั่ง BitBlt	40
4.2.5. สิ่งที่จะพัฒนาเพิ่มเติม	41
5. การทดลองและการสรุปผล	45
5.1. การทดลองพัฒนาในส่วนของการสร้างโปรแกรม	45
5.2. ปัญหาและอุปสรรคในการทำโครงการ	46
5.3. ข้อเสนอแนะเพื่อการพัฒนาโปรแกรมต่อไปในอนาคต	46
6. การติดตั้งและการใช้งานโปรแกรม	
6.1. ขั้นตอนในการติดตั้งโปรแกรมช่วยสอนมีดังต่อไปนี้	47
6.2. การใช้งานโปรแกรม Assistant Training Program V1.0	51
ภาคผนวก	57
รายละเอียดของโปรแกรม (Source Program)	58-96
กิตติกรรมประกาศ	97
เอกสารอ้างอิง (Reference Manual)	
หนังสืออ้างอิง (Reference Manual)	98
Web Site อ้างอิง (Reference Web Site)	98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

ภาพบทที่ 1

ภาพบทที่ 2

ภาพที่ 2.1 OSI โมเดลทั้ง 7 เลเยอร์	5
ภาพที่ 2.2 การเอ็นแคปซูลชั้น	5
ภาพที่ 2.3 ลำดับการเอ็นแคปซูลชั้น	6
ภาพที่ 2.4 เลเยอร์ของโพรโตคอล TCP/IP	7
ภาพที่ 2.5 หมายเลข IP แอดเดรส	9
ภาพที่ 2.6 แสดงคลาส จำนวนเครือข่าย และจำนวนโฮสต์ของแต่ละคลาส	9
ภาพที่ 2.7 แสดงรูปแบบของ IP แอดเดรสในคลาสต่าง ๆ	9
ภาพที่ 2.8 ตัวอย่างหมายเลข IP แอดเดรสทั้งสองแบบ	10
ภาพที่ 2.9 การทำซับเน็ตติ้ง	10
ภาพที่ 2.10 ความสัมพันธ์ระหว่างชุดโพรโตคอล	11

ภาพบทที่ 3

ภาพที่ 3.1 โปรแกรม Visual Basic 6.0 สำหรับวินโดวส์	18
ภาพที่ 3.2 หน้าต่างแสดงคุณลักษณะ	21
ภาพที่ 3.3 หน้าต่าง Project	22
ภาพที่ 3.4 หน้าจอหลักของโปรแกรม Assistant Training	25
ภาพที่ 3.5 หน้าจอแสดงรายละเอียดของโปรแกรม	27
ภาพที่ 3.6 หน้าจอของการดักจับจอภาพ	27

ภาพบทที่ 4

ภาพที่ 4.1 หน้าจอโปรแกรมดักจับจอภาพ	37
ภาพที่ 4.2 ภาพโครงสร้างของโปรแกรม (Program Structure)	42
ภาพที่ 4.3 ภาพไฟร์ชาร์ตรวมของโปรแกรม	43
ภาพที่ 4.4 ภาพไฟร์ชาร์ตของโปรแกรมส่วนการจับจอภาพ	44

ภาพบทที่ 5

ภาพบทที่ 6

ภาพที่ 6.1 รูปภาพการติดตั้งหลังจากการรัน Setup.exe	47
--	----

ภาพที่ 6.3 เลือกชื่อกลุ่ม โปรแกรม (Program Group)	48
ภาพที่ 6.4 เลือกเปลี่ยนชื่อกลุ่ม โปรแกรม (Program Group)	49
ภาพที่ 6.5 ภาพการติดตั้งไฟล์ต่างๆลงบนฮาร์ดดิส	49
ภาพที่ 6.6 ภาพแสดงโปรแกรมตรวจพบไฟล์ที่จะติดตั้งซ้ำกันก็ให้กดปุ่ม Yes เก็บไฟล์เดิมไว้	49
ภาพที่ 6.7 ภาพแสดงโปรแกรมตรวจพบไฟล์ที่ใช้งานอยู่ก็ให้เลือกปุ่ม Ignore เพื่อข้ามไป	50
ภาพที่ 6.8 ภาพแสดงการติดตั้งโปรแกรมสำเร็จแล้วให้กดปุ่ม OK	50
ภาพที่ 6.9 ภาพแสดงชื่อกลุ่ม โปรแกรม (Program Group) ที่สร้างในเมนูโปรแกรม	50
ภาพที่ 6.10 ภาพแสดงหน้าต่างโปรแกรม (Program Interface)	52
ภาพที่ 6.11 ภาพแสดงการเชื่อมต่อให้เป็นเครื่องแม่ข่าย	53
ภาพที่ 6.12 ภาพแสดงการเปลี่ยนภาษาที่เมนูหน้าต่าง	53
ภาพที่ 6.13 ภาพแสดงการเชื่อมต่อของเครื่องลูกข่าย	54
ภาพที่ 6.14 จอ Full Screen เป็นจอจับภาพ	55
ภาพที่ 6.15 โปรแกรมจะจับภาพจากเมมโมรี่มาเก็บไว้ในพื้นที่จับภาพของโปรแกรม	55
ภาพที่ 6.16 ภาพทางด้านลูกข่ายที่รับภาพได้จากเครื่องแม่ข่าย	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 จุดมุ่งหมาย

ในปัจจุบันนี้การติดต่อสื่อสารระหว่างเครื่อง COMPUTER ผ่านทางเน็ตเวิร์ค และ ระบบ อินเทอร์เน็ตมีบทบาทอย่างมากในการติดต่อสื่อสาร จะเห็นได้ว่าโปรแกรมที่ถูกพัฒนาให้สามารถ ทำงานได้บนระบบอินเทอร์เน็ตมีปริมาณเพิ่มขึ้นอย่างรวดเร็ว เนื่องจากประโยชน์ที่ได้รับมีมากมาย หลายด้าน เช่น ด้านการติดต่อสื่อสาร ด้านการเผยแพร่ความรู้หรือเป็นแหล่งรวบรวมความรู้ในสาขา ต่าง ๆ

ในการพัฒนาซอฟต์แวร์นี้เพื่อนำมาใช้ช่วยในการสอนวิชา COMPUTER หรือนำไปใช้ในการ บรรยายโดยผ่านทางระบบ COMPUTER ซึ่งในปัจจุบันนี้ COMPUTER มีความสำคัญมากใน ทางการศึกษาและการพัฒนาคุณภาพของบุคลากร ดังนั้นการศึกษาวិชา COMPUTER จึงเป็นพื้น ฐานทางการศึกษาที่สำคัญ ในอดีตอุปกรณ์การเรียนการสอนวิชาทาง COMPUTER จะใช้การ บรรยายผ่านทาง Overhead Projector หรือ ผ่านทางเครื่องฉายภาพนิ่ง (Slide Projector) ซึ่งไม่ค่อย สะดวกในการใช้และการแสดงผลก็ไม่ชัดเจนบนจอภาพซึ่งขึ้นอยู่กับคุณภาพของอุปกรณ์นั้นๆ ทำให้การติดตามการเรียนการสอนดูไม่น่าสนใจ

1.2 แนวความคิดและที่มา

การใช้อุปกรณ์การเรียนการสอนวิชาทาง COMPUTER จะใช้การบรรยายผ่านทาง Overhead Projector หรือ ผ่านทางเครื่องฉายภาพนิ่ง (Slide Projector) ซึ่งไม่ค่อยสะดวก ในปัจจุบัน มีผู้พัฒนาอุปกรณ์ Hardware มาใช้ช่วยในการสอนแต่เครื่องจะมีราคาแพงมากเนื่องจากจะต้องใช้ อุปกรณ์ที่จัดทำขึ้นมาเป็นพิเศษและจะต้องเดินสายจากทุกเครื่องมายังกล่องอุปกรณ์ Interface นี้แล้ว ใช้การควบคุมด้วย Manual Switch เมื่อติดตั้งแล้วไม่สามารถเคลื่อนย้ายไปใช้งานที่อื่นได้ ข้าพเจ้า เห็นว่าเราน่าจะใช้การพัฒนาซอฟต์แวร์มาทำงานในลักษณะเดียวกันกับ Hardware Interface ได้โดย ผ่านทางระบบ Network ของเครื่องที่มีอยู่แล้วให้เกิดประโยชน์สูงสุดและสามารถเคลื่อนย้ายการใช้ งานได้ง่ายและสามารถที่จะทำการพัฒนาให้ดียิ่งขึ้นได้ง่ายในอนาคตและมีราคาถูกกว่าการใช้ อุปกรณ์ Hardware และยังไม่ต้องเสียค่าบำรุงรักษาอีกด้วย

1.3 วัตถุประสงค์

- เรียนรู้การพัฒนาโปรแกรมบนระบบปฏิบัติการ Windows 95/98 ได้
- สามารถเรียนรู้การทำงานของโพรโตคอล TCP/IP และประยุกต์ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้เผยแพร่ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่แบบสงวนสิทธิ์และต้องอ้างอิงถึงชื่อของเอกสารที่นำมาใช้

1.4 ขอบเขต

- โปรแกรมประยุกต์นี้มีความสามารถที่จะดักจับจอภาพของเครื่องผู้ส่งเพื่อส่งไปยังเครื่องผู้รับที่ต้องการ และการสื่อสารระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องซึ่งต่อเครื่องคอมพิวเตอร์ของตนเข้ากับระบบเครือข่าย โดยระบบเครือข่ายนี้ต้องมีการใช้โปรโตคอล TCP/IP และเครื่องคอมพิวเตอร์ทั้งสองต้องมีหมายเลข IP Address ที่แน่นอน
- สามารถที่จะยกเลิกการรับและส่งได้
- สามารถที่จะทำการเปลี่ยน Menu และ ปุ่มต่างๆเป็นได้ทั้งภาษาไทยและภาษาอังกฤษ

1.5 แนวทางในการพัฒนา

โปรแกรมประยุกต์นี้ถูกพัฒนาโดยใช้ภาษา Visual Basic 6.0 ที่ทำงานภายใต้ระบบปฏิบัติการ Microsoft Windows 95 รวมทั้งได้ใช้ Winsock Controls ซึ่งเป็น Control ในชุดพัฒนา Microsoft Active-X ที่ใช้สำหรับการติดต่อสื่อสารโดยใช้โปรโตคอล TCP/IP โดยชุดพัฒนา Microsoft Active X นี้ทาง Microsoft ได้แจกจ่ายให้กับผู้พัฒนา Application บนอินเทอร์เน็ตฟรี นอกจากนี้เรายังได้ใช้ API (Application Program Interface) ของ Windows 95 ในส่วนของการติดต่อกับ Windows Socket Port เพื่อใช้ในการติดต่อผ่านระบบเน็ตเวิร์ค

สรุปแล้วโดยรวมหลังจากที่เราได้เลือกเครื่องมือเครื่องมือในการทำงานได้เป็นที่เรียบร้อยแล้ว เราจึงได้วางแผนและขั้นตอนในการพัฒนา ดังนี้

- ทำการเขียนแอปพลิเคชันชั้นการเข้าควบคุมและสั่งงานทางจอภาพเบื้องต้น
- ทำการเขียนแอปพลิเคชันชั้นการเข้าควบคุมจอภาพบนหน้าจอใช้งาน โดยอาศัย Device Context
- ทดลองเขียน โปรแกรมแอปพลิเคชันสำหรับใช้ทดสอบการทำงานติดต่อกับเครือข่ายคอมพิวเตอร์ซึ่งใช้โปรโตคอล TCP/IP โดยการเรียกใช้ Winsock Control
- รวบรวมโปรแกรมทั้งหมดเข้าด้วยกัน และทดสอบการทำงาน
- แก้ไขและตัดแปลงแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับระบบเครือข่ายต่าง ๆ

2.1 ระบบเครือข่ายและการโปรแกรมบนระบบเครือข่าย

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควรในหัวข้อนี้จะอธิบายกับศัพท์พื้นฐาน และหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายดังจะกล่าวต่อไป

2.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์และอินเทอร์เน็ตเวิร์กিং

(Internetworking)

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End-System) ซึ่งระบบปลายทางเป็นระบบที่เป็นอิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นได้ตั้งแต่ไมโครคอมพิวเตอร์ (Microcomputer) ไปจนถึงระบบซูเปอร์คอมพิวเตอร์ (Supercomputer) ขนาดใหญ่เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งปันทรัพยากรของระบบเช่น ไฟล์ (File) ข้อมูล, เครื่องพิมพ์ (Printer), โมเด็ม (Modem) ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing database) อินเทอร์เน็ตเวิร์กিং หรืออินเทอร์เน็ต (Internet) คือการเชื่อมต่อของระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้น คอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่ายอื่น ๆ ได้ เช่น เน็ตเวิร์กโทเคนริงค์ (Tokenring network) เชื่อมกับเน็ตเวิร์กอีเทอร์เน็ต (Ethernet network) โดยมีเกตเวย์ (Gateway) เป็นตัวเชื่อม

2.1.2 รูปแบบของโปรแกรมบนระบบเครือข่าย(Network Programming Models)

จากหัวข้อ 2.1.1 ได้ให้ความหมายของระบบเครือข่ายแสดงถึงวิธีการที่ระบบคอมพิวเตอร์ใด ๆ จะทำการเชื่อมต่อกับระบบเครือข่าย แต่ว่าคอมพิวเตอร์ที่เชื่อมต่ออยู่กับระบบเครือข่ายนั้นมีวิธีการอย่างไรในการแลกเปลี่ยนข้อมูลและแบ่งปันทรัพยากรเหล่านั้นได้ทำให้ต้องมีโปรแกรมประยุกต์ซึ่งสามารถที่จะจัดการในสิ่งที่กล่าวมาแล้วอย่างเหมาะสม ซึ่งในหัวข้อนี้จะกล่าวถึงรูปแบบของโปรแกรมบนระบบเครือข่าย 2 รูปแบบคือ โคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing) และการประมวลผลแบบกระจาย (Distributed Computing)

การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์ (Client/Server Computing)

การประมวลผลแบบไคลเอนต์เซิร์ฟเวอร์นี้ การประมวลผลของโปรแกรมประยุกต์จะแบ่งออกเป็น 2 ส่วนคือ ส่วนฟรอนต์เอนด์ (Front-end) ที่ทำงานบนไคลเอนต์ ส่วนนี้จะทำหน้าที่แสดงผลที่ได้จากการประมวลผล และรับข้อมูลจากผู้ใช้ อีกส่วนหนึ่งคือแบ็กเอนด์ (back-end) ทำงานบนเซิร์ฟเวอร์ (Server) ซึ่งทำหน้าที่ประมวลผลข้อมูลที่ได้รับจากไคลเอนต์และส่งข้อมูลกลับคืนสู่ไคลเอนต์

เซิร์ฟเวอร์มีหน้าที่ในการเก็บรวบรวม และจัดการข้อมูลจากฟรอนต์เอนด์ในรูปแบบการประมวลผล ไคลเอนด์เซิร์ฟเวอร์นี้ เครื่องเซิร์ฟเวอร์นี้มักจะเป็นเครื่องที่มีความสามารถสูงกว่าเครื่องไคลเอนด์ โดยปกติเครื่องเซิร์ฟเวอร์มักจะเป็นเครื่องเมนเฟรม หรือมินิคอมพิวเตอร์และเครื่องไคลเอนด์มักจะเป็นเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งการติดต่อกันระหว่างส่วนฟรอนต์เอนด์และแบ็คเอนด์ ทำโดยผ่านระบบเครือข่ายซึ่งในทางปฏิบัติแล้ว ในส่วนของแบ็คเอนด์ที่อยู่บนเซิร์ฟเวอร์ จะเป็นฝ่ายให้บริการแก่งานของไคลเอนด์หลายงานในเวลาเดียวกัน

การประมวลผลแบบกระจาย (Distributed Computing)

การประมวลผลของโปรแกรมประยุกต์มีรูปแบบการประมวล 2 แบบ คือ 프리คอลเล็คชัน (Precollection) และการประมวลผลแบบขนาน (parallel processing) ดังนี้

- 프리คอลเล็คชันเป็นลักษณะการทำงานที่ข้อมูลที่ต้องการในการจัดเก็บ และส่งต่อไปที่ระบบเครือข่ายอยู่ตลอดเวลาอย่างสม่ำเสมอ การทำงานในลักษณะนี้จะเหมาะสมกับงานบางงาน เช่น ต้องการเก็บสถานะของคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายหนึ่ง ๆ ทุกเครื่อง
- การประมวลผลแบบขนาน การประมวลผลลักษณะนี้งานใด ๆ จะถูกประมวลผลด้วยคอมพิวเตอร์หลาย ๆ เครื่องโดยเครื่องคอมพิวเตอร์เหล่านั้นสามารถจะติดต่อกันโดยระบบเครือข่าย เช่น การทำการพัฒนาโปรแกรมประยุกต์ขนาดใหญ่ โดยทีมพัฒนาที่มีผู้พัฒนาหลายคน สามารถลดเวลาของการแปล และการรวมโมดูล (Module) ต่าง ๆ เข้าเป็นโปรแกรมเดียวกัน โดยการแบ่งงานการแปลโมดูลเหล่านั้นแก่คอมพิวเตอร์ในระบบเครือข่ายทำการแปลในเวลาเดียวกัน

2.1.3 โมเดล OSI

OSI เป็นคำย่อที่มาจากคำว่า Open System Interconnection โดยที่เป็นมาตรฐานที่ถูกเสนอขึ้นโดย International Standards Organization ซึ่งเป็นองค์กรที่จัดตั้งขึ้นมาเพื่อดูแลและส่งเสริม ตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรม แบบระบบเปิด (Open System) เพราะมุ่งที่จะให้ระบบคอมพิวเตอร์ในหลาย ๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโปรโตคอล (protocol) ในการสื่อสารออกเป็น 7 เลเยอร์ (layer) ซึ่งโปรโตคอลคือชุดของกฎ หรือข้อตกลงในการติดต่อ ข้อสังเกตโมเดล OSI เป็นเพียงข้อเสนอแนะ มิใช่ข้อกำหนด และควรรู้ว่ายังไม่มีระบบการเชื่อมต่อใดที่สร้างเหมือนกับโมเดล OSI จริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

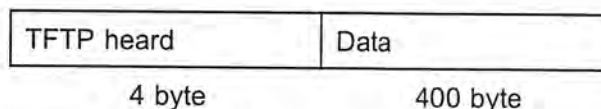
7. Application Layer
6. Presentation Layer
5. Session Layer
4. Transport Layer
3. Network Layer
2. Datalink Layer
1. Physical Layer

ภาพที่ 2.1 OSI โมเดลทั้ง 7 เลเยอร์

ในหนึ่งชั้นของเลเยอร์ ไม่ได้กำหนดว่าจะต้องมีเพียงหนึ่งโพรโทคอลเท่านั้น ที่อยู่ในระดับเลเยอร์เดียวกัน และในทางตรงข้าม ชุดของโพรโทคอลใด ๆ อาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกับเป็นข้อกำหนดของระบบเครือข่ายเรียกว่าชุดโพรโทคอล (Protocol Suite) เช่น ชุดโพรโทคอล TCP/IP (Transmission Control/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเป็นเลเยอร์ คือกำหนดการติดต่อระหว่างเลเยอร์ ทำได้โดยไม่ต้องคำนึงถึงการเปลี่ยนในเลเยอร์ใด ๆ ที่ติดกัน

2.1.4 เอ็นแคปซูลชัน (Encapsulation)

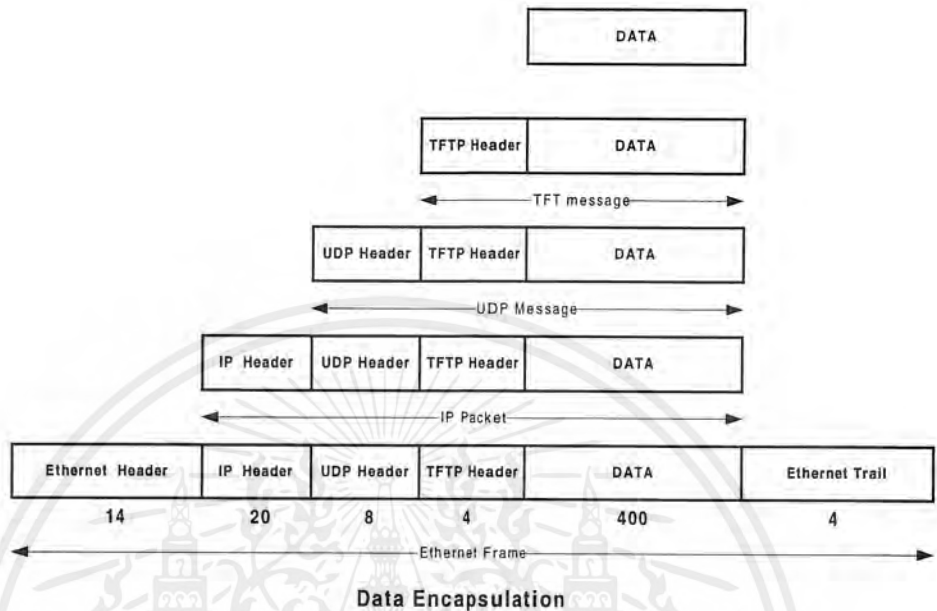
พิจารณาโปรแกรมประยุกต์ TFTP (File Transfer Protocol) ซึ่งใช้ในโพรโทคอล UDP (User Datagram Protocol) ระหว่างสองระบบซึ่งเชื่อมต่อด้วยอีเทอร์เน็ต ถ้าโปรแกรมไคลเอนต์ TFTP มีข้อมูล 400 byte ต้องการส่งไปที่โปรแกรมเซิร์ฟเวอร์ โปรแกรมไคลเอนต์ TFTP จะเพิ่มข่าวสารควบคุม 4 byte เป็นส่วนหัวของข้อมูลก่อนที่จะผ่านข้อมูลไปสู่เลเยอร์ UDP การเพิ่มของข่าวสารควบคุมไปที่ข้อมูลเรียกว่า เอ็นแคปซูลชัน ดังแสดงในภาพ 2.2



ภาพที่ 2.2 การเอ็นแคปซูลชัน

เลเยอร์ UDP จะไม่มีการตีความส่วนหัว TFTP 4 byte งานของเลเยอร์ UDP คือส่งข้อมูล 404 byte ไปสู่เลเยอร์ UDP ของโปรแกรมอีกด้านหนึ่ง จากนั้นเลเยอร์ UDP จะทำการเพิ่มส่วนหัว 8 byte แล้วส่งข้อมูล 432 byte ไปยังเลเยอร์ดาต้าลิงก์ ที่เลเยอร์นี้จะมีการเพิ่มส่วนหัวอีก 14 byte และส่วนหาง

อีก 4 byte ดังภาพ 2.3 ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.3 ลำดับการเอ็นแคปซูลชั้น

2.1.5 ลักษณะของการติดต่อแบ่งออกเป็น 2 ชนิด คือ

Connection-oriented

คือการติดต่อที่ต้องมีการเชื่อม โปรเซสที่จะทำการติดต่อก่อนที่จะมีการส่งหรือรับข้อมูลซึ่งสามารถใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโปรเซส ถึงแม้ว่าข้อมูลนี้อาจจะผ่าน Packet-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งชั้นการทำงานออกเป็น

- ชั้นการสร้างการติดต่อ (connection establishment)
- ชั้นการส่งผ่านข้อมูล (data transfer)
- ชั้นยกเลิกการติดต่อ (connection termination)

Connectionless หรือดาต้าแกรม (Datagram)

คือจะไม่มีชั้นการสร้างการติดต่อ และชั้นการยกเลิกการติดต่อ แต่จะมีชั้นการส่งผ่านข้อมูลอย่างเดียว โดยข้อมูลเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรส (Address)

การที่ระบบในระบบเครือข่ายสามารถติดต่อกันได้จำเป็นต้องมีแอดเดรสไว้คล้ายกับหมายเลขประจำตัวซึ่งลำดับของแอดเดรสสามารถพิจารณาได้คือ

- ในแต่ละเครือข่ายจะต้องมีแอดเดรสสำหรับเครือข่าย
- คอมพิวเตอร์โฮสต์ของแต่ละเครื่องในเครือข่ายจะต้องมีแอดเดรสเฉพาะของตัวเอง
- ในแต่ละโปรเซสในโฮสต์จะต้องมีหมายเลขประจำตัว

โดยทั่วไปแอดเดรสของโฮสต์จะประกอบด้วยหมายเลขเครือข่าย (Network ID) และหมายเลขของโฮสต์ (Host ID) ส่วนแอดเดรสของโปรเซสของผู้ใช้จะอยู่ในรูปจำนวนเต็มซึ่งกำหนดโดยโพรโตคอล เช่น โพรโตคอล TCP/IP จะใช้เลขจำนวนเต็มขนาด 32 บิต ในการกำหนดหมายเลขเครือข่ายและหมายเลขของโฮสต์ และทั้ง TCP และ UDP ใช้เลขจำนวนเต็มขนาด 16 บิต เป็นหมายเลขพอร์ตหรือหมายเลขของโปรเซส ชุดของโพรโตคอลส่วนใหญ่จะมีการกำหนดชุดของแอดเดรสสำหรับการบริการที่เป็นที่รู้จักกัน โดยทั่วกัน เช่นคอมพิวเตอร์ที่โพรโตคอล TCP/IP ส่วนใหญ่จะมี FTP (File Transfer Protocol) ซึ่งไคลเอนต์สามารถติดต่อได้โดยใช้หมายเลขพอร์ตคือ 21

2.2 ความรู้เบื้องต้นเกี่ยวกับโพรโตคอล TCP/IP

2.2.1 เปรียบเทียบระหว่างโพรโตคอล TCP/IP และ OSI โมเดล

การออกแบบโพรโตคอล TCP/IP นั้นไม่ได้เป็นไปตามรูปแบบของ OSI โมเดล เนื่องจากถูกออกแบบโดยองค์กรขนาดใหญ่ซึ่งใช้เวลานานในการออกแบบตลอดจนการรับรองมาตรฐานต่างกับโพรโตคอล TCP/IP ที่ถูกออกแบบด้วยความต้องการอันเร่งด่วนของรัฐบาลสหรัฐ จึงทำให้การพัฒนาโพรโตคอล TCP/IP มีเงื่อนไขของในด้านความต้องการที่ต่างจาก OSI โมเดล ซึ่งหากเรามองโดยรวมแล้วจะเห็นว่าโพรโตคอล TCP/IP มีการแบ่งเป็นเลเยอร์ที่น้อยกว่า OSI โมเดล คือ มี 4 ชั้นเท่านั้น ดังภาพ 2.4 โดยแบ่งเป็น

Application Layer
Transport Layer
Internet Layer
Physical Layer

ภาพที่ 2.4 เลเยอร์ของโพรโตคอล TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันเลเยอร์ (Application Layer)

ในเลเยอร์นี้ประกอบโปรแกรมประยุกต์ที่ใช้เครือข่าย เช่น โปรแกรมส่งถ่ายข้อมูล (File transfer protocol) และอาจกล่าวได้ว่าเลเยอร์นี้โพรโทคอล TCP/IP ก็คือ เลเยอร์ในชั้นแอปพลิเคชันเลเยอร์ ร่วมกับชั้นพรีเซนเตชันเลเยอร์ (Presentation Layer) ใน OSI โมเดลนั่นเอง และในเลเยอร์ชั้นนี้ของโพรโทคอล TCP/IP จะกลืนอยู่ภายในตัวโปรแกรมประยุกต์

ทรานสปอร์ตเลเยอร์ (Transport Layer)

ในชั้นนี้เป็นชั้นที่ทำให้การส่งข้อมูลจากจุดปลายถึงจุดปลาย หากเปรียบเทียบกับ OSI โมเดลก็สามารถเทียบได้กับชั้นเซสชันเลเยอร์ (Session Layer) ร่วมกับทรานสปอร์ตเลเยอร์นั่นเอง โดยโพรโทคอล TCP/IP มีซอกเก็ต (Socket) เป็นจุดปลาย (end-point) ในการสื่อสาร ซึ่งซอกเก็ตนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์ และหมายเลขพอร์ต (port) ของเครื่องที่ต้องการส่งข้อมูลไปถึง ในชั้นนี้มีการรับรองการถึงที่หมาย และลำดับของข้อมูลที่ส่งโดยไม่ซ้ำ และความผิดพลาดข้อมูล

อินเทอร์เน็ตเลเยอร์ (Internet Layer)

เลเยอร์นี้มีการกำหนดค่าแอดเดรส และทำการหาเส้นทางการส่ง หน้าที่ของเลเยอร์นี้เทียบเท่ากับเน็ตเวิร์คเลเยอร์ (Network Layer) และดาต้าลิงก์เลเยอร์ (Data Link Layer) ของ OSI โมเดล

ฟิสิคอลลเยอร์ (Physical Layer)

โพรโทคอล TCP/IP ไม่ได้กำหนดรูปแบบของการเชื่อมต่อในระดับนี้ไว้ใหม่ แต่ได้ใช้มาตรฐานที่มีอยู่เดิมที่กำหนดไว้ก่อน เช่น RS232, อีเทอร์เน็ต (Ethernet) เป็นต้น

2.2.2 รูปแบบการกำหนดแอดเดรสของโพรโทคอล TCP/IP

ในหัวข้อนี้จะได้อธิบายรูปแบบการกำหนดแอดเดรสของโพรโทคอล TCP/IP ซึ่งลักษณะแอดเดรสของโพรโทคอลนี้ค่าของแอดเดรสของเครื่องคอมพิวเตอร์ในระบบเครือข่ายจะไม่ซ้ำกัน โดยเลขนี้เรียกว่า IP แอดเดรส (IP Address) เป็นเลข 32 บิต ซึ่งแบ่งเป็นคลาส (Class) ตามหลักในการพิจารณาที่จะได้กล่าวต่อไปนี้

การแบ่งเน็ตเวิร์คคลาส (Network Class)

เนื่องจากหมายเลขแอดเดรสของคอมพิวเตอร์เครื่องใด ๆ นั้นจะต้องสามารถบอกถึงความแตกต่างระหว่างตัวเครื่องเอง ตลอดจนเครือข่ายที่คอมพิวเตอร์นั้นเชื่อมต่ออยู่ด้วยหมายเลข IP แอดเดรส จึงแยกออกเป็น 2 ส่วน ได้แก่ ส่วนที่แสดงหมายเลขของเครื่องคอมพิวเตอร์โฮสต์ และส่วนที่เป็นหมายเลขของเครือข่าย

การแบ่งคลาสของแอดเดรสทำได้โดยการพิจารณาจำนวนบิตของ 2 ส่วนประกอบข้างต้น ซึ่งมีการแบ่งออกเป็น 5 คลาส แต่มีการใช้เพียง 3 คลาสแรก คือ คลาส A, คลาส B และ คลาส C ส่วนคลาส D และ E ถูกสงวนไว้สำหรับจุดประสงค์พิเศษ

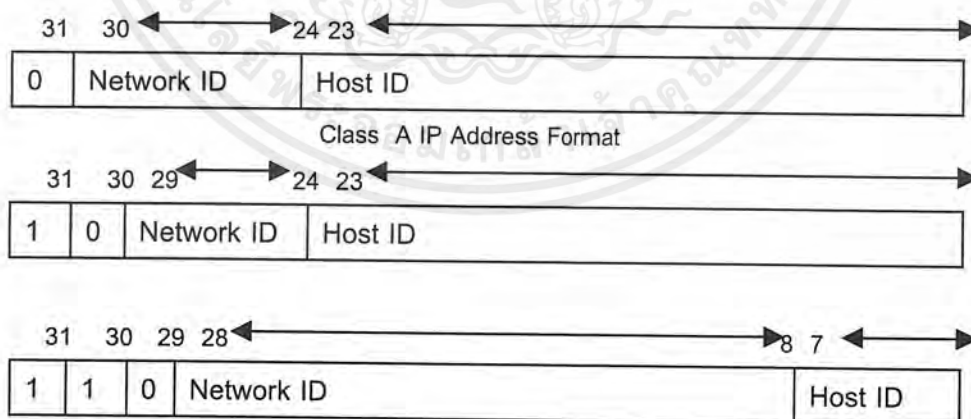
Class ID	Network ID	Host ID
----------	------------	---------

ภาพที่ 2.5 หมายเลข IP แอดเดรส

Network Class	Networks	Hosts per Network
A	126	16,777,214
B	16,382	65,534
C	2,097,150	254

ภาพที่ 2.6 แสดงคลาส จำนวนเครือข่าย และจำนวนโฮสต์ของแต่ละคลาส

โดยปกติแล้วผู้พัฒนาโปรแกรม ไม่ต้องสนใจความแตกต่างระหว่างคลาสของ IP แอดเดรส



ภาพที่ 2.7 แสดงรูปแบบของ IP แอดเดรสในคลาสต่าง ๆ

การแทนด้วยเลขฐานสิบและจุด (Dotted Decimal Notation)

เนื่องจากการแทนหมายเลข IP แอดเดรสเป็นเลขฐาน 2 ซึ่งค่อนข้างอ่านไม่สะดวกจึงมีการแทนเลขฐาน 2 เหล่านั้นในรูปเลขฐานสิบและจุด โดยเลขฐานสิบแต่ละตัวจะแทนเลขฐาน 2 จำนวน 8 บิต (ยกเว้นตัวสุดท้ายที่แทนเลขฐาน 2 จำนวน 7 บิต) อย่างไรก็ตามมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 บิต โดยระหว่างเลขฐานสิบแต่ละตัวจะแทรกด้วยจุด ดังนั้นจะต้องใช้เลขฐานสิบ 4 ตัว ในการแทนเลข 32 บิต ที่เป็น IP แอดเดรส ดังตัวอย่างตามภาพ 2.8 จะสังเกตว่าหมายเลข IP แอดเดรสนี้จัดอยู่ในคลาส B โดยหมายเลขของเครือข่ายคือ 166.78 และหมายเลขประจำเครื่องคือ 4.139

<p><u>Dotted Decimal Notation</u></p> <p>166.78.4.139</p> <p><u>Binary Representation</u></p> <p>10100110 01001110 00000100 10001011</p>
--

ภาพที่ 2.8 ตัวอย่างหมายเลข IP แอดเดรสทั้งสองแบบ

การทำซับเน็ตติง (Subnetting)

การทำซับเน็ตติงเป็นการเปลี่ยนแปลงการใช้หมายเลขของเครื่อง โฮสต์และหมายเลขของเครือข่ายในระดับท้องถิ่น โดยในทางตรรกคือ การเลื่อนเส้นแบ่งที่แยกหมายเลขเครื่อง และหมายเลขของเน็ตเวิร์กที่อยู่ในหมายเลข IP แอดเดรส โดยที่ปริมาณของหมายเลขเครื่อง โฮสต์ และหมายเลขเครือข่ายจะแปรผกผันกัน เช่น หากมีปริมาณของเน็ตเวิร์กมาก ก็จะทำให้เครื่องใด ๆ ที่จะต่อกับระบบเครือข่ายหนึ่งๆจะน้อยลงเป็นต้นในทางปฏิบัติการทำซับเน็ตติงทำโดยการนำซับเน็ตมาร์ค (Subnetmark) คือตัวเลขจำนวน 32 บิต มาทำการ กระทำตรรกและ (AND) กับหมายเลข IP แอดเดรส ดังตัวอย่างโดยกำหนดหมายเลข IP แอดเดรส คือ 166.78.4.139 และซับเน็ตมาร์ค คือ 255.255.255.0 ทำการกระทำ ตรรกและ ดังภาพ

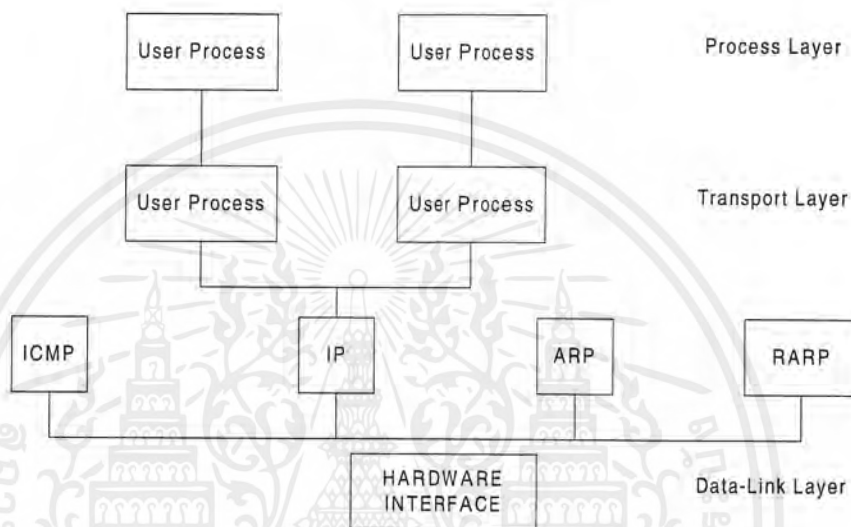
<p>166.78.4.139</p> <p>and</p> <p><u>255.255.255.0</u></p> <p>166.78.4.0</p>
--

ภาพที่ 2.9 การทำซับเน็ตติง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ชุดโพรโทคอล TCP/IP (TCP/IP Protocol Suite)

ชุดโพรโทคอล TCP/IP นอกจากมีโพรโทคอล TCP และ IP แล้วยังมีโพรโทคอลอย่างอื่น อีกดังภาพที่ 2.10 แสดงความสัมพันธ์ของชุดโพรโทคอลโดยแบ่งตามเลเยอร์



ภาพที่ 2.10 ความสัมพันธ์ระหว่างชุดโพรโทคอล

โพรโทคอล IP (Internet Protocol)

โพรโทคอล IP เป็นโพรโทคอลแบบคอนเนกชันเลส (Connectionless Protocol) ซึ่งได้กล่าวถึงลักษณะของโพรโทคอลชนิดนี้ไปแล้วในหัวข้อ 2.1.5 โดยที่โพรโทคอล IP ไม่รับประกันว่าข้อมูลที่ส่งจะไปถึงปลายทางซึ่งแพ็คเกจ (Packet) ของข้อมูลอาจไปถึงในลักษณะที่ผิดพลาด, ซ้ำกันหรือไม่ไปถึงเลย โดยความน่าเชื่อถือของการส่งจะถูกควบคุมในโพรโทคอลในเลเยอร์ต่างๆ ไป การหาเส้นทางของข้อมูลจะทำในระดับของโพรโทคอล IP นี้ โดยพิจารณาแต่ละแพ็คเกจ แยกออกจากกัน และยังมีหน้าที่ในการจัดเรียงข้อมูลใหม่ที่ปลายทางอีกด้วย

โพรโทคอล ARP (Address Resolution Protocol)

โพรโทคอลนี้ทำหน้าที่จับคู่ระหว่างหมายเลข IP แอดเดรสเข้ากับหมายเลขแอดเดรสทางฮาร์ดแวร์ โดยโพรโทคอลนี้ทำการส่งข้อความไปทั่วเครือข่ายท้องถิ่น ซึ่งข้อความนี้เป็นลักษณะข้อความที่ตรวจสอบว่ามีคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกับที่ต้องการหาหรือไม่ หากคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกันนั้นได้รับข้อความนี้ก็จะตอบกลับและเป็นหน้าที่ส่งเกตว่าโพรโทคอลนี้ ทำงานได้กับระบบเครือข่ายท้องถิ่นเท่านั้น เพราะว่าโครงสร้างหมายเลขทางฮาร์ดแวร์ของเครื่องคอมพิวเตอร์จะขึ้นอยู่กับชนิดของเครือข่ายด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโทคอล ICMP (Internet Control Message Protocol)

เป็นโพรโทคอลที่จัดการเกี่ยวกับข่าวสารความผิดพลาดและการควบคุมเกตเวย์ และเครื่องคอมพิวเตอร์ในระบบเครือข่าย

โพรโทคอล RARP (Reverse Address Resolution Protocol)

เป็นโพรโทคอลที่ทำหน้าที่จับคู่ระหว่างหมายเลขของฮาร์ดแวร์กับหมายเลข IP แอดเดรส หรือทำงานกลับกันกับโพรโทคอล ARP

โพรโทคอล UDP (User Datagram Protocol)

โพรโทคอลนี้เป็นโพรโทคอลที่อยู่ในระดับทรานสปอร์ตเลเยอร์ และมีความสำคัญเพราะว่าเป็นโพรโทคอลที่ผู้พัฒนาโปรแกรมสามารถใช้ได้โดยตรง โพรโทคอลชนิดนี้เป็นโพรโทคอลแบบคอนเนกชันเลสมีความน่าเชื่อถือต่ำ ไม่รับรองว่าข้อมูลที่ส่งไปจะไปถึงปลายทางหรือไม่ และอาจซ้ำซ้อนหรือผิดพลาดได้ แต่ข้อดีของโพรโทคอลนี้ คือค่าความสิ้นเปลือง (Overhead) ที่ต่ำ

โพรโทคอล TCP (Transmission Control Protocol)

โพรโทคอลนี้อยู่ในระดับทรานสปอร์ตเลเยอร์ เหมือนกับโพรโทคอล UDP แต่มีลักษณะที่ตรงข้ามกันคือ เป็นโพรโทคอลแบบคอนเนกชันออเรียนเต็ด โดยการติดต่อจะมีการตรวจสอบปลายทางก่อนทำการทำให้มีความน่าเชื่อถือในการรับ ส่งข้อมูลและลำดับของข้อมูลจะมีลำดับเหมือนกับต้นทางและเนื้อข้อมูลไม่ผิดพลาด จึงทำให้เกิดความสิ้นเปลืองในการเชื่อมต่อของการส่งข้อมูลมากกว่าโพรโทคอล UDP

2.2.4 หมายเลขพอร์ต

เนื่องจากในเวลาใด ๆ สามารถมีโพรเซสของผู้ใช้สามารถใช้ UDP หรือ TCP ได้พร้อมกัน ดังนั้นจึงต้องมีวิธีแยกแยะว่าข้อมูลเป็นของผู้ใช้คนใด ซึ่งวิธีที่ TCP และ UDP ใช้คือการใช้หมายเลขพอร์ต (Port Number) เมื่อโพรเซสไคลเอนต์ (client process) ต้องการที่จะติดต่อกับเซิร์ฟเวอร์ไคลเอนต์จะต้องเจาะจง เซิร์ฟเวอร์ที่ต้องการติดต่อ แต่ลำพังรู้อัดเดรสอินเทอร์เน็ต 32 บิตเพียงอย่างเดียวนั้นไม่เพียงพอ เพราะสามารถติดต่อกับโฮสต์ได้เพียงอย่างเดียวแต่ไม่สามารถเจาะจงโพรเซสที่จะทำการติดต่อได้ ดังนั้นเพื่อแก้ปัญหาที่ทั้ง TCP และ UDP ได้มีการกำหนด หมายเลขพอร์ตมาตรฐาน (well-known ports) ซึ่งเป็นที่รู้จักกันเช่น ทุก ๆ ระบบ TCP/IP ที่มีเซิร์ฟเวอร์ FTP (File Transfer Protocol) จะมีหมายเลขพอร์ตเป็น 21 เป็นต้น เมื่อ TCP หรือ UDP กำหนดหมายเลขพอร์ตที่ไม่ซ้ำกันให้โพรเซสของผู้ใช้ เราเรียกหมายเลขพอร์ตนี้ว่าหมายเลขพอร์ตชั่วคราว (Ephemeral Port Numbers) เมื่อไคลเอนต์เลิกใช้หมายเลขพอร์ตนี้แล้ว สามารถกำหนดหมายเลขพอร์ตนี้ให้ไคลเอนต์อื่นได้ โพรเซสที่ได้รับหมายเลขพอร์ตชั่วคราวนี้จะไม่สนใจว่ามูลค่าเท่าใด แต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นหน้าที่ของอีกโพรเซสหนึ่งที่ต้องสนใจ เพราะต้องส่งข้อมูลกลับมาที่พอร์ตนี้ ใน TCP และ UDP นั้นหมายเลขพอร์ตตั้งแต่ 1-1023 เป็นพอร์ตที่สงวนไว้สำหรับหมายเลขพอร์ตมาตรฐาน

2.3 Winsock (Window Socket Application Programming Interface)

Winsock คือโปรแกรมส่วนขยายที่ทำหน้าที่ฟังก์ชันด้านเน็ตเวิร์คที่มีการใช้โพรโตคอล TCP/IP โดยสำหรับในระบบของยูนิกซ์ในเริ่มแรกซึ่งยังไม่สนับสนุนโพรโตคอล TCP/IP ดังนั้นในการที่จะทำให้ยูนิกซ์สนับสนุนโพรโตคอล TCP/IP โดยไม่จำเป็นต้องดัดแปลงระบบปฏิบัติการจึงมีการเขียนโปรแกรมส่วนขยายขึ้นมาสำหรับรองรับโพรโตคอล TCP/IP ซึ่งเราจะรู้จักกันในชื่อของ *Berkeley Sockets* สำหรับบน Microsoft Windows เพื่อที่จะให้สนับสนุนโพรโตคอล TCP/IP จึงได้มีการพัฒนาโปรแกรมส่วนที่จะทำหน้าที่โพรโตคอล TCP/IP นี้ซึ่งเราจะเรียกว่า Windows Socket นั่นเอง สำหรับในหัวข้อนี้จะได้กล่าวถึง Winsock เป็นไลบรารี (Library) ของฟังก์ชันที่ทำหน้าที่เกี่ยวกับการสื่อสาร โดยติดต่อกับซ็อกเก็ต Winsock ที่มีส่วนขยายมาจาก Berkeley Socket โดยได้เพิ่มส่วนที่เป็นเมสเสจไดรเวน (Message-driven) เพื่อให้ทำงานได้ดีในสภาวะแวดล้อมของวินโดวส์และสามารถจัดการระบบเครือข่ายแบบ TCP/IP ได้ ก่อนที่จะกล่าวถึง Winsock จะได้กล่าวถึง Berkeley Socket เพื่อเป็นพื้นฐานความเข้าใจ Winsock ในอันดับต่อไป

2.3.1 Berkeley Socket

ซ็อกเก็ต (Socket) เป็นแอปพลิเคชันโปรแกรมอินเทอร์เฟซ (Application Program Interface) ในที่นี้จะขอเรียกโดยย่อว่า API โดยที่ API นี้จะเป็นส่วนที่ติดต่อกับทรานสปอร์ตเลเยอร์กับโปรแกรมประยุกต์ ทำให้ผู้พัฒนาโปรแกรมสามารถที่จะเรียกใช้บริการต่าง ๆ ของทรานสปอร์ตเลเยอร์ได้โดยการใช้ซ็อกเก็ตนั่นเอง Berkeley Socket เป็นซ็อกเก็ตที่ใช้กับทรานสปอร์ตเลเยอร์ของโพรโตคอล TCP/IP โดยที่ Berkeley Socket มีความเป็นมาพร้อม ๆ กับอินเทอร์เน็ต

2.3.2 เปรียบเทียบ Berkeley Socket กับ Winsock

ซ็อกเก็ตทั้งสองมีความแตกต่างกัน โดยแสดงเป็นข้อ ๆ ได้ดังนี้

- Winsock รุ่น 1.1 สนับสนุนการทำงานกับกลุ่มโดเมน (Domain) TCP/IP เท่านั้น ส่วนซ็อกเก็ตของยูนิกซ์ (UNIX) นั้นสามารถสนับสนุนการทำงานกับกลุ่มโดเมนของโดเมนยูนิกซ์และกลุ่มโดเมน Xerox XNS ด้วย
- ค่าที่ส่งกลับจากฟังก์ชัน (Return Value) ใน Berkeley Socket กับ Winsock จะต่างกัน เช่น ถ้าเรียกให้ฟังก์ชัน socket 0 จะส่งค่ากลับเป็น -1 หากการทำงานผิดพลาดในการทำงานของระบบปฏิบัติการยูนิกซ์ แต่ถ้าหากเป็น Winsock ที่ส่งกลับในกรณีนี้คือ INVALID_SOCKET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อของฟังก์ชันที่ทำหน้าที่เดียวกันอาจแตกต่างกันเช่นฟังก์ชัน `close 0` เป็นฟังก์ชันในการปิด การติดต่อของซ็อกเก็ตในระบบปฏิบัติการยูนิกซ์ แต่ถ้าหากเป็นชื่อใน Winsock จะใช้ชื่อ `closesocket 0`
- ในระบบการปฏิบัติการยูนิกซ์การจัดการกับซ็อกเก็ตจะเหมือนกับแฟ้มข้อมูล (File) ที่อยู่บนดิสก์ แต่ใน Winsock การจัดการกับซ็อกเก็ตจะต่างกันออกไป
- Winsock มีฟังก์ชันที่เพิ่มเติมจาก Berkeley Socket เพื่อจะสนับสนุนการทำงานแบบแมสเสจไครเวนของสถาปัตยกรรมของระบบปฏิบัติการวินโดวส์ซึ่งจะได้กล่าวในหัวข้อต่อไป

2.3.3 ส่วนเพิ่มเติมของ Winsock จาก Berkeley Socket

ส่วนที่เพิ่มขึ้นมาหลายส่วนของ Winsock เนื่องมาจากเหตุที่ระบบปฏิบัติการวินโดวส์นั้นมีสถาปัตยกรรมที่เรียกว่าแมสเสจไครเวน และเพื่อจะสนับสนุนการทำงานแบบนอนพรีเอมพ์ตีฟ (Nonpreemptive) ของระบบปฏิบัติการวินโดวส์อีกด้วย ในส่วนต่อไปจะได้กล่าวถึงสถาปัตยกรรมแมสเสจไครเวนของระบบปฏิบัติการวินโดวส์ก่อน เพื่อเป็นพื้นฐานความเข้าใจในการอธิบายต่อไป

สถาปัตยกรรมแมสเสจไครเวนของวินโดวส์

ในที่นี้จะกล่าวโดยสรุปเกี่ยวกับสถาปัตยกรรมแมสเสจไครเวน ดังนั้นทุก ๆ โปรแกรมที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์นี้จะต้องมีส่วนประกอบหลักสำคัญ 2 ส่วน ได้แก่ ลูป (Loop) ที่ทำหน้าที่รับแมสเสจและโปรแกรมย่อยอื่น ๆ ที่เป็นส่วนปฏิบัติงานจริงของโปรแกรม โดยในส่วนที่เป็นลูปคอยรับแมสเสจนั้นจะรับแมสเสจจากแมสเสจคิว (Message Queue) เมื่อโปรแกรมได้รับแมสเสจแล้วจะทำการปฏิบัติงานตามโปรแกรมย่อยที่สัมพันธ์กับแมสเสจที่ได้รับนั้น ซึ่งการทำงานแบบนี้จึงเป็นที่มาของแมสเสจไครเวน หรือ อีเวนต์ไครเวน (Event-driven) เพราะไม่มีส่วนของโปรแกรมใด ๆ ทำงานได้จนกว่าจะมีเหตุการณ์หรือแมสเสจที่เกี่ยวข้องเกิดขึ้น โดยทั่วไปตัวอย่างของแมสเสจคือ การกดคีย์บอร์ดหรือการเลื่อนเมาส์และมาจากภายในตัวระบบปฏิบัติการเอง ส่วนโปรแกรมย่อยที่เรียกใช้ในการปฏิบัติการจริงของโปรแกรมที่จะปฏิบัติตามแมสเสจที่ได้รับนั้นจะอยู่ในรูปคลาสของโปรแกรม โดยมีที่มา 2 แบบ คือมาจากการโปรแกรมของผู้เขียนโปรแกรมนั้น และบางส่วนจะเป็นคลาสที่ถูกกำหนดไว้ล่วงหน้าโดยระบบปฏิบัติการวินโดวส์

เมื่อทำงานตามโปรแกรมย่อยแล้ว โปรแกรมจะเข้าสู่การทำงานในลูปรอรับแมสเสจอีกครั้งหนึ่งเพื่อคอยแมสเสจที่จะเข้ามาในแมสเสจคิวต่อไป แต่ในระหว่างที่โปรแกรมกำลังทำงานอยู่ในโปรแกรมย่อยระบบปฏิบัติการวินโดวส์จะไม่สามารถควบคุมการทำงานของโปรแกรมนั้น ๆ ได้ เรียกลักษณะงานแบบนี้ว่านอนพรีเอมพ์ตีฟ (Nonpreemptive)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันการทำงานแบบอะซิงโครนัสของ Winsock

ในการออกแบบ Winsock นั้นครั้งแรกออกแบบมาเพื่อทำงานกับสถานะแบบนอนพรีเอมพ์ตีฟของระบบปฏิบัติการวินโดวส์ ด้วยเหตุนี้จึงต้องเพิ่มฟังก์ชันการทำงานจาก Berkeley Socket คือการทำงานแบบบล็อกกิ้ง (Blocking) และนอนบล็อกกิ้ง (Nonblocking)

ฟังก์ชันในการทำงานของ Berkeley Socket หลายฟังก์ชันเป็นฟังก์ชันที่ไม่ทราบเวลาในการส่งค่ากลับที่แน่นอน เรียกลักษณะการทำงานแบบนี้ว่าการทำงานแบบบล็อก ซึ่งในสถานะการทำงานของระบบปฏิบัติการยูนิคซ์การทำงานของฟังก์ชันแบบนี้จะไม่เป็นปัญหาที่ร้ายแรง เพราะระบบปฏิบัติการจะทำารจัดการในส่วนที่จะพรีเอมพ์ (Preempt) โปรแกรมที่บล็อก และให้โปรแกรมอื่นทำงานแทน แต่ในทางกลับกันบนระบบปฏิบัติการวินโดวส์ ระบบปฏิบัติการไม่สามารถจะทำการพรีเอมพ์โปรแกรมได้ จึงต้องรอให้โปรแกรมนั้นคืนการทำงานให้แก่ระบบปฏิบัติการเท่านั้น Winsock จึงจำเป็นต้องฟังก์ชันการทำงานแบบอะซิงโครนัส เพื่อที่จะแก้ปัญหาดังกล่าวเพราะว่าการทำงานบนระบบเครือข่ายไม่สามารถกำหนดเวลาได้แน่นอน ฟังก์ชันที่ทำงานแบบอะซิงโครนัสคือ ฟังก์ชันที่เมื่อเรียกทำงานแล้วจะส่งค่ากลับทันทีแต่อาจจะยังไม่มีการทำงานเกิดขึ้นในทันทีและเมื่อทำงานเสร็จแล้ว จะมีเมสเสจส่งกลับมาหาโปรแกรมที่เรียกใช้ฟังก์ชันนั้นเพื่อบอกถึงการทำงานที่เสร็จสิ้น และฟังก์ชันที่เป็นฟังก์ชันแบบอะซิงโครนัสจะมีคำนำหน้าว่า WSAsync นำหน้าชื่อฟังก์ชันซึ่งจะเห็นได้ว่าการทำงานแบบอะซิงโครนัสถูกออกแบบมาเพื่อให้เข้ากับการทำงานในระบบปฏิบัติการวินโดวส์เป็นอย่างดี

2.4 การใช้งาน Winsock Control

Winsock จะมีคำสั่งที่ยอมให้ผู้เขียน Program ควบคุมการติดต่อเครื่องคอมพิวเตอร์ที่อยู่ที่อื่นและควบคุมการแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์ 2 เครื่อง Internet ActiveX Control Pack จะประกอบด้วยคำสั่งของ Winsock 2 แบบ คือ Winsock TCP (Transmission Control Protocol) และ Winsock UDP (User Datagram Protocol) ลักษณะการติดต่อ 2 แบบนี้ สามารถที่จะใช้ในการสร้างโปรแกรมที่ใช้ในการติดต่อระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย

ข้อแตกต่างระหว่างโปรโตคอล 2 แบบนี้

- รูปแบบของ Winsock TCP เป็นแบบการติดต่อขั้นพื้นฐาน กล่าวคือ จะคล้ายกับการติดต่อทางโทรศัพท์คือผู้ใช้จะต้องทำการตรวจสอบก่อนที่จะเริ่มการติดต่อ
- รูปแบบของ Winsock UDP จะเป็นแบบ Connectionless คือเป็นแบบที่ไม่มีการตรวจสอบมีลักษณะการทำงานคล้ายกับสถานีส่งคลื่นวิทยุ คือ เครื่องคอมพิวเตอร์จะส่งข้อมูลแบบกระจายโดยไม่ต้องทราบข้อมูลที่ส่งถึงผู้รับหรือไม่

2.4.1 Winsock TCP Control

ลักษณะการใช้งาน :

- ใช้สร้างโปรแกรมประยุกต์ทางด้านผู้ส่ง โดยมีการเก็บข้อมูลของผู้ที่จะส่งไว้ก่อนที่จะส่งข้อมูลให้ฝ่ายผู้รับ
- สร้างโปรแกรมประยุกต์ทางด้านผู้รับที่มีฟังก์ชันในการรวบรวมข้อมูลของผู้ส่งแต่ละเครื่อง

ลักษณะการใช้งานที่ใช้ Winsock TCP ในการส่งข้อมูลไปให้ Computer ที่จะติดต่อ:

ในการใช้งาน Winsock TCP เพื่อติดต่อกับ Computer 2 เครื่องแบบ Real-time ระบบเครือข่ายนั้น เครื่อง Computer ที่จะเป็นผู้ให้บริการ (Server) จะเป็นผู้รับข้อมูล กำหนดช่องทางการติดต่อสื่อสารให้ Computer ที่จะขอใช้บริการ (Client) เมื่อ Winsock TCP บนเครื่อง Server ได้รับคำร้องขอใช้บริการ มันจะสร้าง Socket ใหม่ และทำการสร้างการติดต่อ เมื่อสามารถสร้างการติดต่อได้แล้ว Server และ Client จะสามารถส่งข้อมูลซึ่งกันและกันได้ และเมื่อการสื่อสารเสร็จสิ้น Client จะทำการยกเลิกการติดต่อ Winsock ที่ทำงานบนเครื่อง Server เมื่อนั้น Server จะทำการลบ Socket ที่ได้ทำการสร้างไว้

บทที่ 3

การออกแบบ

3.1 หลักในการออกแบบ

จุดมุ่งหมายของโปรแกรม Assistance Training ก็เพื่อที่จะให้ผู้ใช้งานคอมพิวเตอร์ที่มีการติดตั้งโปรแกรมที่ออกแบบไว้ก่อนและต่อคอมพิวเตอร์เข้ากับระบบเครือข่ายที่สามารถที่จะสื่อสารโดยผ่านทางระบบเน็ตเวิร์คที่มีอยู่ ซึ่งจะสามารถใช้งานระบบเน็ตเวิร์คที่มีอยู่ให้เกิดประสิทธิภาพและเป็นที่น่าพอใจในการสื่อสารผ่านระบบเน็ตเวิร์คจำเป็นต้องมีการใช้โพรโตคอลซึ่งเปรียบเสมือนกับภาษาที่ใช้ในการพูดคุยระหว่างเครื่องคอมพิวเตอร์ที่ต่ออยู่ในระบบเน็ตเวิร์ค ซึ่งก็มีอยู่ใช้งานกันหลายโพรโตคอล อาทิเช่น IPX/SPX, NetBEUI, TCP/IP ฯลฯ และเพื่อให้การใช้งานโปรแกรม Assistance Training สามารถที่จะใช้ได้อย่างกว้างขวางและทั่วไปเราจึงทำการออกแบบโปรแกรมโดยเลือกใช้โพรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol)

การออกแบบตัวโปรแกรมจะแบ่งออกเป็น 2 ส่วนสำคัญ ๆ คือ

1. การจัดการเกี่ยวกับภาพ
2. การจัดการและส่งข้อมูลภาพผ่านระบบเน็ตเวิร์ค

ซึ่งทั้ง 2 ส่วนหลักจะกล่าวต่อไปในรายละเอียดด้านหลัง ในการออกแบบของทั้ง 2 ส่วนหลักจะพยายามเลือกใช้รูปแบบการทำงานที่ง่ายสำหรับการพัฒนาเพื่อที่จะสามารถทำการปรับปรุงและแก้ไขได้อย่างรวดเร็ว ดังนั้นเราจึงได้ใช้โปรแกรม Visual Basic 6.0 ในการพัฒนาโปรแกรมช่วยสอนขึ้นมาโดยอาศัยหลักการของ Object ซึ่งหลักการของ Object จะทำให้เราสามารถที่จะแยกโปรแกรมแต่ละส่วนเป็นอิสระออกจากกัน แต่แต่ละ Object ก็ยังคงมีความสัมพันธ์กัน

หลังจากที่เราได้คิดออกแบบหลักการทำงานของโปรแกรมแล้วมาคิดว่าควรจะออกแบบโปรแกรมและจะต้องมีความสามารถอย่างไรบ้าง (ความสามารถพื้นฐาน) ซึ่งเราได้วางความสามารถของโปรแกรมไว้ดังนี้

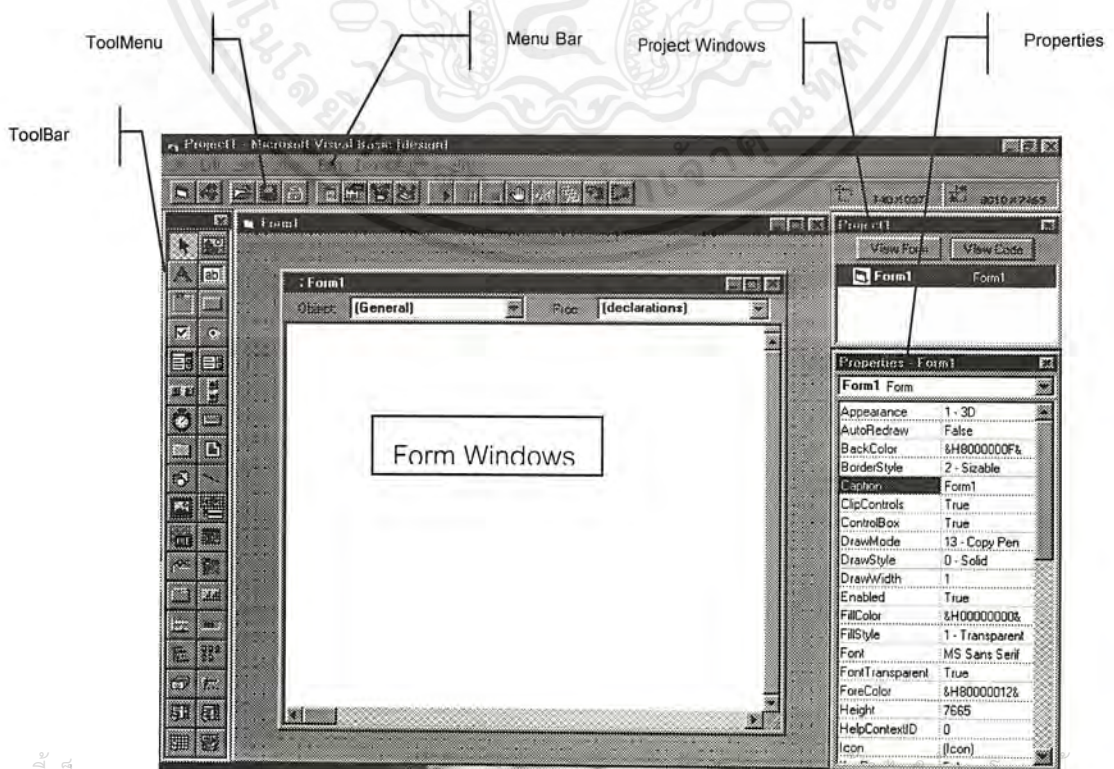
1. สามารถทำการจับจอภาพจากเครื่องหนึ่งที่เรียกว่าผู้ส่ง ไปแสดงภาพออกที่เครื่องคอมพิวเตอร์อีกเครื่องหนึ่งที่เรียกว่าผู้รับ
2. สามารถรับภาพที่ส่งมาจากเครื่องคอมพิวเตอร์ที่เรียกว่าผู้ส่งแสดงออกที่ของเครื่องตนได้
3. สามารถที่จะยกเลิกการรับและส่งภาพได้
4. โปรแกรมสามารถแสดงเมนูได้ 2 ภาษา ไทยและอังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนลิขสิทธิ์โดยสำนักหอสมุดกลางพระจอมเกล้าลาดกระบัง ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนในการพัฒนา

ในการพัฒนาเราได้เลือกใช้ Visual Basic 6.0 เป็น Tools ในการพัฒนา ซึ่งจากการศึกษา และใช้งานพบว่าเราสามารถที่จะเขียนโปรแกรมในลักษณะของ object โดยใช้ Visual Basic 6.0 ได้ และขั้นตอนในการเขียนก่อนข้างจะสามารถเข้าใจและนำไปใช้งานรวมทั้งการแก้ไขได้อย่างสะดวก และรวดเร็ว

เหตุผลที่เลือกใช้โปรแกรม Visual Basic 6.0 สำหรับวินโดวส์ ก็เนื่องมาจากในปัจจุบัน วินโดวส์เป็นระบบปฏิบัติการที่ได้รับความนิยมเป็นอย่างสูง มีข้อดีหลายประการ เช่น ทำงานใน โหมดกราฟิก ง่ายต่อการเรียนรู้และใช้งาน และอื่น ๆ อีกหลายประการ จึงทำให้วินโดวส์เป็นระบบ ปฏิบัติการที่มีการใช้ความสามารถของเครื่องระดับไมโครคอมพิวเตอร์ได้ค่อนข้างเต็มที่ ดังนั้นตัว แปลภาษาต่าง ๆ จึงถูกพัฒนาเพื่อให้นำมาเขียนโปรแกรมต่าง ๆ บนวินโดวส์ได้ ซึ่งหนึ่งในนี้ ก็คือภาษาเบสิกซึ่งเป็นภาษาชั้นสูงที่ได้รับการพัฒนาต่อมาเรื่อย ๆ ตั้งแต่ BASIC, BASICA, QuickBASIC, PDS BASIC, QBASIC, Visual BASIC for DOS จนมาถึง Visual BASIC for Windows และได้รับการพัฒนาต่อมาจนถึงเวอร์ชัน 6.0 เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ มากยิ่งขึ้น และยังมีส่วนสนับสนุนการพัฒนาแอปพลิเคชันในระดับ 32 บิต ซึ่งจะสามารถ ทำงานได้ทั้งบนวินโดวส์ 95/98 และวินโดวส์ NT



ภาพที่ 3.1 โปรแกรม Visual Basic 6.0 สำหรับวินโดวส์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำ ไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาวะแวดล้อมสำหรับการโปรแกรมของ Visual Basic ประกอบด้วยเครื่องมือที่ช่วยในการเขียนโปรแกรมบน Visual Basic เมนูบาร์เตรียมไว้สำหรับเข้าถึงคำสั่งส่วนใหญ่ซึ่งควบคุมสภาวะแวดล้อมของการเขียนโปรแกรม เมนูและคำสั่งต่าง ๆ ทำงานตามแบบมาตรฐานที่ใช้กับโปรแกรมบนวินโดวส์

โดยทั่วไปสามารถเลือกใช้เครื่องมือที่ช่วยในการเขียนโปรแกรมเหล่านี้ได้ โดยการใช้คีย์บอร์ดหรือเมาส์ ตำแหน่งได้เมนูบาร์คือทูลบาร์ (แถบแสดงเครื่องมือ) รวบรวมปุ่มต่าง ๆ ที่สามารถนำไปใช้งานเพื่อเรียกใช้คำสั่ง หรือควบคุมสภาวะแวดล้อมสำหรับการเขียนโปรแกรมของ Visual Basic การใช้งานปุ่มบนทูลบาร์ให้คลิกที่ปุ่มนั้นโดยการใช้เมาส์ ส่วนล่างของหน้าจอคือทาสก์บาร์ (แถบงาน) ของวินโดวส์ 95/98 สามารถใช้ทาสก์บาร์ในการสลับระหว่างส่วนประกอบหลาย ๆ ตัวใน Visual Basic และสามารถเรียกโปรแกรมบนวินโดวส์ที่เคยโหลดไว้แล้วให้แสดงขึ้นมาได้

นอกจากนี้ ยังมีทูลบ็อกซ์ เมนูบาร์ หน้าต่างแสดงโครงการ หน้าต่างฟอร์ม หน้าต่างโครงการ หน้าต่างปัจจุบัน (Immediate) หน้าต่างแสดงคุณสมบัติ และหน้าต่างแสดงโครงร่างฟอร์มขนาดและรูปร่างที่แท้จริงของหน้าต่างเหล่านี้ขึ้นอยู่กับว่าจะระบบจะถูกตั้งไว้อย่างไรใน Visual Basic 6.0 ซึ่งสามารถจัดเป็นแนวนอนและผูกติดกัน หรือวางตำแหน่งของหน้าต่างเพื่อทำให้องค์ประกอบที่โปรแกรมนั้นสามารถเห็นได้ (Visible) หรือสามารถเข้าถึงได้ (Accessible)

ส่วนประกอบของจอภาพ Visual Basic 6.0 มีดังนี้

ส่วนประกอบ	รายละเอียด
Form	เป็นจอภาพสำหรับพัฒนาโปรแกรม
Toolbox	ประกอบด้วย Icon ต่าง ๆ ซึ่งใน Visual Basic เรียกว่า "Control" ที่จะนำไปใช้งานโดยการนำลงไปวางบน Form
Toolbar	ประกอบด้วย Icon ต่าง ๆ ที่ใช้ร่วมในการพัฒนาโปรแกรม
Project Explorer Window	สำหรับเรียก Form ต่าง ๆ ขึ้นมาแก้ไข ในกรณีที่มี Project ประกอบด้วย Form มากกว่า 1 Form
Properties Window	ใช้สำหรับกำหนดคุณสมบัติ (Properties) ให้กับ Form และ Object ต่าง ๆ ที่อยู่บน Form
Form Layout Window	ใช้สำหรับดูตำแหน่งของ Form บนจอ ทำให้จัดตำแหน่งของ Form ได้สะดวกขึ้น

3.2.1 ฟอรัมของส่วนติดต่อผู้ใช้

ใน Visual Basic นั้น ฟอรัมคือหน้าต่างที่จัดทำขึ้นเพื่อเป็นส่วนติดต่อกับผู้ใช้ของ โปรแกรมฟอรัมสามารถประกอบไปด้วยเมนู ปุ่ม กรอบแสดงรายการ สกรอลล์บาร์ และอื่น ๆ อีกหลายอย่างที่เห็นได้ในโปรแกรมวินโดว์ทั่วไป เมื่อเริ่มเข้าสภาวะแวดล้อมการ โปรแกรมของ Visual Basic จะเห็นฟอรัมที่ชื่อ Form1 ซึ่งปรากฏพร้อมกับกริด (กลุ่มของจุดที่เรียงในระยะที่เท่ากันตลอดทั้งแนวตั้งและแนวนอน) มาตรฐานซึ่งสามารถใช้ส่วนติดต่อผู้ใช้ของโปรแกรมนี้ในการร่างองค์ประกอบต่าง ๆ ได้ ส่วนการปรับแต่งขนาดของฟอรัมทำได้โดยใช้เมาส์ ฟอรัมสามารถแสดงเต็มหน้าจอหรือบางส่วนก็ได้ และยังสามารถเพิ่มฟอรัมเข้าไปได้อีกโดยการคลิกที่คำสั่ง Add Form ในเมนู Project

ถ้าบางส่วนของฟอรัมถูกบังด้วยเครื่องมือช่วยโปรแกรมแล้ว ก็สามารถปิดหรือเปลี่ยนขนาดเครื่องมือนั้นเพื่อให้สามารถใช้พื้นที่บนฟอรัมได้เต็มที่ หรือให้คลิกที่ไทเทิลบาร์ของฟอรัม และแดรกฟอรัมเลื่อนไปจนกระทั่งเห็นส่วนที่ถูกบังได้ไม่เหมือนกับ Visual Basic เวอร์ชัน ก่อนตรงที่การเคลื่อนย้ายตำแหน่งของฟอรัมบนหน้าจอใน Visual Basic ไม่มีผลต่อตำแหน่งฟอรัมในขณะที่โปรแกรมถูกรันจริง คุณลักษณะในขณะการรันนั้นจะถูกควบคุมโดยหน้าต่างแสดงโครงร่างของฟอรัม ซึ่งสามารถกำหนดตำแหน่งเริ่มต้นของฟอรัมใหม่ด้วยวิธีการแดรกฟอรัมสมมุติเส็กๆ ในหน้าต่างแสดงโครงร่างของฟอรัมไปวางไว้ในตำแหน่งที่ต้องการ

3.2.2 ทุลบ็อกซ์ ToolBox (กล่องเครื่องมือ)

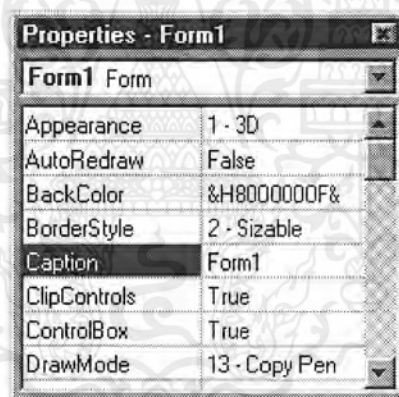
การเพิ่มองค์ประกอบของส่วนติดต่อผู้ใช้เข้าไปในฟอรัมได้โดยการใช้เครื่องมือหรือส่วนควบคุมในทุลบ็อกซ์ ให้เปิดทุลบ็อกซ์โดยการคลิกที่ปุ่มทุลบ็อกซ์บนทุลบาร์ (แถบแสดงเครื่องมือ) ทุลบ็อกซ์แสดงอยู่ทางด้านซ้ายของหน้าจอ หลังจากเพิ่มส่วนควบคุมเข้าไปในฟอรัมแล้วมันจะกลายเป็นออบเจกต์(วัตถุ) หรือองค์ประกอบของส่วนติดต่อผู้ใช้ที่สามารถโปรแกรมได้ ทุลบ็อกซ์ประกอบไปด้วยส่วนควบคุมที่สามารถใช้เพิ่มงานพิมพ์ แผ่นป้าย (เลเบล) ปุ่มกล่องแสดงรายการ สกรอลล์บาร์ เมนู และส่วนวาดรูปกราฟิกบนส่วนติดต่อผู้ใช้นี้ ผู้ใช้จะเห็นองค์ประกอบเหล่านี้ ในขณะที่ทำการรัน โปรแกรม และจะทำงานเหมือนกับออบเจกต์มาตรฐานอื่นๆ ในแอปพลิเคชันของวินโดว์ทั่วไป

ทุลบ็อกซ์ประกอบไปด้วยส่วนควบคุมที่สามารถใช้สร้างออบเจกต์ ซึ่งทำงานพิเศษโดยไม่แสดงบนหน้าจอ ออบเจกต์เหล่านี้ใช้งานได้ดีมากแต่ไม่แสดงให้ผู้ใช้เห็นในขณะที่รันโปรแกรม โดยออบเจกต์พวกนี้จะมีออบเจกต์สำหรับประมวลผลข้อมูลในฐานะข้อมูล ทำงานร่วมกับแอปพลิเคชันบนวินโดว์ตัวอื่น และตรวจสอบการทำงานตามเวลาในโปรแกรมได้ นอกจากนี้ยังสามารถแสดงชื่อของส่วนควบคุมในทุลบ็อกซ์ได้โดยการชี้เมาส์ไว้เหนือส่วนควบคุมนั้นช่วงเวลาหนึ่ง

3.2.3 หน้าต่างแสดงคุณลักษณะ

หน้าต่างแสดงคุณลักษณะช่วยให้สามารถเปลี่ยนคุณลักษณะ หรือตั้งค่าแสดงลักษณะขององค์ประกอบในส่วนติดต่อผู้ใช้บนฟอร์มได้ การตั้งค่าแสดงลักษณะเป็นคุณสมบัติภายในของออบเจกต์ในส่วนติดต่อผู้ใช้ของเรา ตัวอย่างเช่น สามารถปรับเปลี่ยนข้อความหรือขนาดตัวอักษร หรือแสดงในแนวที่เปลี่ยนไปได้ การเปลี่ยนค่าคุณสมบัตินี้ทำได้โดยการใช้หน้าต่างแสดงคุณลักษณะขณะที่กำลังสร้างส่วนติดต่อผู้ใช้ หรือเปลี่ยนค่าโดยโปรแกรมในขณะที่กำลังรัน

หน้าต่างแสดงคุณลักษณะประกอบด้วยกล่องแสดงรายการของออบเจกต์ ซึ่งแสดงรายการขององค์ประกอบ (ออบเจกต์) ที่เป็นส่วนติดต่อผู้ใช้ทั้งหมดบนฟอร์ม หน้าต่างแสดงคุณลักษณะแสดงรายการของส่วนตั้งค่าคุณลักษณะที่สามารถเปลี่ยนแปลงออบเจกต์แต่ละตัวได้ โดยสามารถดูคุณลักษณะเรียงตามตัวอักษร (Alphabetic) หรือแบ่งเป็นกลุ่ม (Categorized) ก็ได้



ภาพที่ 3.2 หน้าต่างแสดงคุณลักษณะ

3.2.4 หน้าต่าง Project

โปรแกรม Visual Basic ถูกสร้างขึ้นพร้อมกับเพิ่มข้อมูลจำนวนมาก หรือจะถูกแปลงภาษาเมื่อโปรแกรมเสร็จสมบูรณ์ เพื่อให้เข้าใจถึงองค์ประกอบต่าง ๆ เหมือนกับกำลังทำโครงการหนึ่ง ผู้ออกแบบ Visual Basic ได้รวมหน้าต่าง Project ไว้ในสถานะแวดล้อมของการโปรแกรมด้วย หน้าต่างโครงการจะแสดงไฟล์ที่ใช้งานทั้งหมดในขบวนการเขียนโปรแกรม และทำให้สามารถเข้าถึงพวกมันผ่านปุ่มพิเศษ 2 ปุ่มคือ View Code และ View Object การเพิ่มลบและบันทึกไฟล์แต่ละตัวเข้าไปในโครงการได้โดยการใช้คำสั่งบนเมนู File และ Project การเปลี่ยนแปลงใด ๆ ในโครงการหนึ่ง ๆ จะกระทบกับหน้าต่าง Project

ไฟล์โครงการซึ่งดูแลไฟล์ที่ใช้สนับสนุนทั้งหมดนี้ ในโครงการการเขียนโปรแกรมเรียกว่าไฟล์โครงการของ Visual Basic (. vbp) ใน Visual Basic 6.0 ในหน้าต่าง Project เดียวกัน สามารถไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดไฟล์โครงการเข้าไปได้มากกว่าหนึ่งไฟล์ และสามารถสลับไปมาได้โดยการคลิกที่ชื่อโครงการภายใต้ชื่อในรูปของโครงสร้างคล้ายต้นไม้ซึ่งเหมือนกับที่แสดงโดย Windows Explorer ซึ่งสามารถหดหรือขยายกิ่งก้านเหล่านี้ได้โดยการรวมฟอร์มโมดูลและองค์ประกอบอื่นๆ เข้ามาโดยการคลิกที่เครื่องหมายบวก และลบด้านหน้าของโหนดเหล่านั้น

โดยทั่วไประบบงานหนึ่ง ๆ จะประกอบด้วยหลาย ๆ จอภาพ เช่น จอภาพสำหรับป้อนข้อมูล (Data Entry) จอภาพสำหรับค้นหาข้อมูล (Data Inquiry) เป็นต้น ดังนั้นในการพัฒนาโปรแกรมจึงนิยมที่จะแยกแต่ละจอภาพออกเป็นโปรแกรมเพื่อความสะดวกต่อการแก้ไขตามหลักของการเขียนโปรแกรมแบบ Modularity เช่น ภาษาตระกูล XBase ก็แยกออกเป็นแฟ้มนามสกุล PRG หรือในภาษา BASIC ก็แยกออกเป็นนามสกุล BAS เป็นต้น แล้วจึงนำแต่ละโปรแกรมย่อยมาประกอบกันขึ้นเป็นระบบโดยการ Compile ไฟล์เหล่านั้นร่วมกันเป็น Executed Program (ไฟล์นามสกุล EXE) เพื่อนำไปใช้งาน

ใน Visual Basic ก็เช่นเดียวกัน แต่ละจอภาพที่พัฒนาขึ้นก็คือ Form ต่าง ๆ และเมื่อนำมารวมกันก็จะกลายเป็นระบบงานระบบหนึ่ง หรือที่เรียกว่า Project ดังนั้น Project และ Form จึงต้องทำงานร่วมกัน จะขาดไฟล์ใดไฟล์หนึ่งไม่ได้



ภาพที่ 3.3 หน้าต่าง Project

3.3 ขั้นตอนในการพัฒนาโปรแกรมของ Visual Basic

ขั้นตอนในการพัฒนาโปรแกรม ประกอบด้วยขั้นตอนหลัก 2 ขั้นตอน ดังนี้

ขั้นตอนที่ 1 สร้างจอภาพของโปรแกรม

ในขั้นตอนนี้ จะทำการออกแบบ Form เพื่อใช้ในการติดต่อกับผู้ใช้ หรือที่เรียกว่า การออกแบบ “User Interface” ในการพัฒนาโปรแกรมแบบเดิม ขั้นตอนนี้จะใช้เวลาและค่าใช้จ่ายสูง เนื่องจากจะต้องเขียน โปรแกรมเพื่อสร้างจอภาพต่าง ๆ จากนั้นต้อง Compile โปรแกรมนั้น แล้ว Run จึงจะเห็นจอภาพที่จัดทำขึ้น แต่สำหรับ Visual Basic ปัญหาในลักษณะนี้ได้ถูกแก้ไขโดยใช้เทคนิคของ Visualize ซึ่งเป็นความสามารถส่วนหนึ่งของ Visual Basic ขั้นตอนนี้จึงสามารถทำได้ง่าย เพียง

แต่นำเอา Control ต่าง ๆ ใน Toolbox ที่ต้องการใช้งานมาวางไว้บน Form ซึ่งทำให้ประหยัดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรในหน่วยงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
และสามารถเห็นลักษณะจอภาพที่ออกแบบได้ในขณะนั้นเลย

ไม่ว่ากรณีใดๆ ทั้งสิ้น ออกพิมพ์ใหม่ให้ทันแต่ต้องอย่าลืมแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 เขียนโปรแกรม

เมื่อทำการวาง Control ต่าง ๆ ลงบน Form เรียบร้อยแล้ว (Control ต่าง ๆ เมื่อถูกนำมาวางไว้บน Form จะเรียกว่า Object) ขั้นตอนต่อมาก็คือ การเขียนโปรแกรมเพื่อกำหนดการทำงานให้กับแต่ละออบเจกต์ ภายใต้เหตุการณ์ต่าง ๆ (Event) ที่เกิดขึ้นกับจอภาพนั้น ๆ

3.4 การจัดสร้างโปรแกรมช่วยสอน

เมื่อเข้าสู่โปรแกรม Visual Basic 6.0 สำหรับวินโดวส์แล้ว เราจะทำการสร้างหน้าจอของโปรแกรมการติดต่อหลายทางสำหรับผู้ใช้งาน โดยการสร้างโครงงานขึ้นแล้วสร้างส่วนติดต่อผู้ใช้โดยใช้ส่วนควบคุมในทูลบ็อกซ์ ในโครงงานนี้จะสร้างฟอร์ม (Form) ขึ้นมาตามลักษณะการใช้งาน ซึ่งได้ออกแบบไว้จำนวน 4 ฟอร์ม ได้แก่

- ฟอร์มของหน้าจอหลัก
- ฟอร์มของการจับจอภาพ
- ฟอร์มของการติดต่อ
- ฟอร์มเกี่ยวกับแสดง Program Version

ในการสร้างฟอร์มแต่ละฟอร์ม จะทำการเลือก Custom Control ต่าง ๆ ที่ต้องการนำมาใช้จากทูลบ็อกซ์ใส่ลงในฟอร์ม และ Control เหล่านี้ จะตอบสนองต่อเหตุการณ์ที่เกิดขึ้น โดยผ่านทางโพรซีเจอร์เหตุการณ์ (event procedure) และนอกจากนี้ยังสามารถจัดการ Control ได้โดยผ่านทางวิธี (Method) ที่เขียนขึ้นมา

3.4.1 การพัฒนาในส่วนของการรับและส่งข้อมูลผ่านระบบเน็ตเวิร์ค (Network Connection)

ดังที่ได้กล่าวข้างต้นแล้วเราได้เลือกใช้โพรโตคอล TCP/IP เป็นโพรโตคอลในการใช้รับและส่งข้อมูลในระบบเน็ตเวิร์ค ดังนั้นเราจึงจำเป็นต้องเขียนโค้ดซึ่งอยู่ในรูปแบบของ Visual Basic 6.0 ขึ้นมาเพื่อใช้เป็นตัวจัดการรับและส่งข้อมูลโดยในการใช้ Visual Basic 6.0 เขียนโปรแกรมเพื่อรับและส่งข้อมูลโดยการใช้โพรโตคอล TCP/IP นั้นเราสามารถที่จะทำได้ 2 วิธี คือ

- การใช้ไฟล์ Winsock.bas ซึ่งข้างในจะประกาศคำสั่งในการเรียกใช้ฟังก์ชันใน Winsock.dll โดยการประกาศนี้จะประกาศนี้จะประกาศออกมาให้อยู่ในรูปแบบของตัวแปรที่ Visual Basic 6.0 สนับสนุน และในการเรียกใช้ก็จะกระทำเหมือนกับการ Call Function DLL จาก API ของ Windows โดยทั่ว ๆ ไป
- การใช้ Control Winsock ซึ่งเป็น control ในชุดของ Active X Internet Control Pack ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 6.0 ได้ใช้กันอย่างง่าย โดยวิธีนี้จะช่วยให้ผู้พัฒนาแอปพลิเคชันที่เกี่ยวข้องกับการเชื่อมต่อ
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุขัดแย้งและต้องอ้างอิงถึงเจ้าของเอกสารทุกประการ

เข้าสู่เครือข่าย Internet ไม่ต้องเรียนรู้ถึงการ Control การใช้ Windows Socket อย่างลึกซึ้ง ซึ่งจะช่วยประหยัดเวลาเป็นอย่างมาก

3.5 อุปกรณ์และระบบที่ต้องการ (Requirement)

ความสามารถหรือประสิทธิภาพของโปรแกรมช่วยสอนคอมพิวเตอร์นี้ต้องอาศัยเครื่องที่มีประสิทธิภาพด้วย โดยมีคุณสมบัติของเครื่องผู้สอนและผู้เรียนดังนี้

คุณสมบัติของเครื่องผู้สอน

1. เครื่องพีซี Pentium 200 Mhz. ขึ้นไป
2. หน่วยความจำ (Memory) 32 MB.
3. เนื้อที่ใน Harddisk สำหรับการติดตั้งโปรแกรมและไวเชอร์ประมาณ 50 MB.
4. ระบบเน็ตเวิร์คการ์ด (Network Interface Card) และ Network HUB
5. สาย(Lan Cable) ที่ใช้ในการติดต่อระหว่างเครื่อง
6. จอภาพขนาด 14"
7. โปรแกรมระบบ OS Software Windows98 หรือ Windows NT แนะนำให้ใช้ NT
8. โพรโตคอล Protocol TCP/IP ซึ่งมีมาพร้อมกับ Windows98 หรือ WindowsNT แต่ต้องติดตั้งเพิ่มเติมในส่วนของระบบเน็ตเวิร์ค

คุณสมบัติของเครื่องผู้เรียน

1. เครื่องพีซี Pentium 200 Mhz. ขึ้นไป
2. หน่วยความจำ (Memory) 32 MB.
3. เนื้อที่ใน Harddisk สำหรับการติดตั้งโปรแกรมและไวเชอร์ประมาณ 30 MB.
4. ระบบเน็ตเวิร์คการ์ด (Network Interface Card)
5. สาย(Lan Cable) ที่ใช้ในการติดต่อระหว่างเครื่อง
6. จอภาพขนาด 14"
7. โปรแกรมระบบ OS Software Windows95,98 หรือ Windows NT
8. โพรโตคอล Protocol TCP/IP ซึ่งมีมาพร้อมกับ Windows95,98,NT แต่ต้องติดตั้งเพิ่มเติมในส่วนของระบบเน็ตเวิร์ค

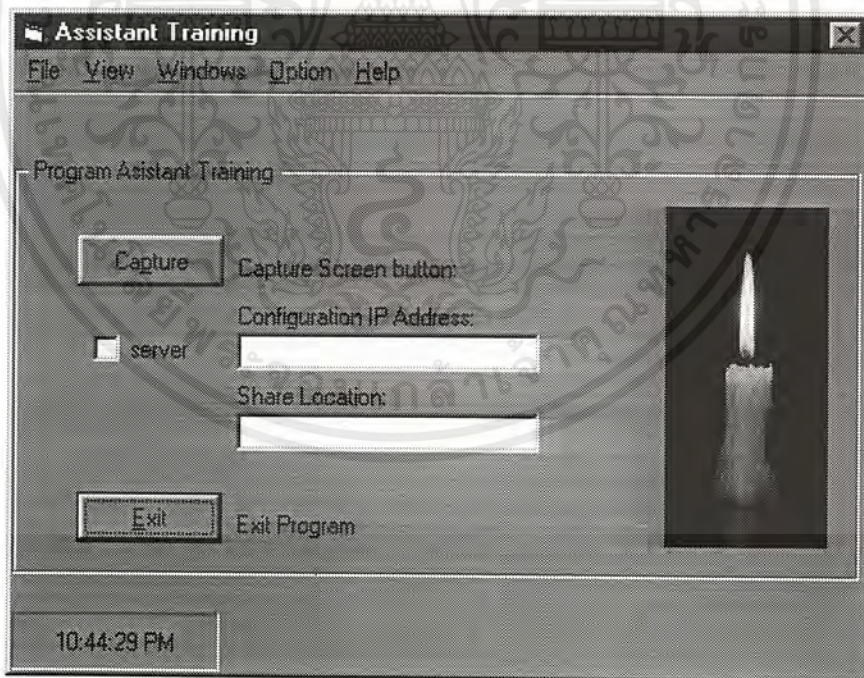
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 โครงสร้างโดยรวมของโปรแกรม

โปรแกรม Assistant Training เป็นโปรแกรมประเภท Client Server ซึ่งก็เหมือนการติดต่อจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่งโดยผ่านทางระบบเน็ตเวิร์ค การทำงานของโปรแกรมจะมีการสื่อสารด้วยโพรโทคอล TCP/IP โดยใช้ Windows Sockets Port ในการนำข้อมูลไปถึงยังผู้รับ ซึ่งการใช้งานโพรโทคอล TCP/IP นั้นกำหนดไว้ว่าเครื่องคอมพิวเตอร์ที่โพรโทคอลนี้จะมี IP Address ที่แน่นอนเป็นของตนเองและไม่ซ้ำกับใครในเครือข่ายเน็ตเวิร์ค ซึ่งเปรียบเสมือนเลขที่บ้านนั่นเอง โปรแกรมจะทำการบันทึกภาพจากจอภาพตามที่กำหนดจากนั้นก็ส่งเข้าไปแพ็คเกจข้อมูลให้อยู่ในรูปแบบของ TCP/IP แล้วจึงทำการส่งออกไป และในทางกลับกันก็จะนำ Packages ของ TCP/IP ที่ส่งมาถึงตนทำการแกะข้อมูลที่เป็นภาพออกมาแล้วทำการส่งออกไปแสดงที่จอภาพอีกครั้งหนึ่ง

3.6.1 การออกแบบส่วนติดต่อกับผู้ใช้ (User-Interface) และการใช้งานโปรแกรม

หน้าต่างการติดต่อกับผู้ใช้ของโปรแกรม Assistance Training จะมีหน้าต่างดังรูปที่ 3.4



ภาพที่ 3.4 หน้าจอหลักของโปรแกรม Assistant Training

โดยหลัก ๆ แล้วจะมีปุ่มตัวเลือกต่างๆ คือ 1.Capture 2.Server Select 3.Configuration IP Address และ Share Location 3. Exit และ นอกจากนี้ข้างล่างจะมี Status Bar สำหรับที่จะใช้แสดงเวลา ณ.ปัจจุบันที่ใช้โปรแกรมได้ และในส่วนเมนูของโปรแกรมจะประกอบด้วย 5 เมนูหลัก คือ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. File Menu
2. View Menu
3. Windows Menu (Toggle Language Thai and English)
4. Option (For Run Capture Screen)
5. Help (Program Help)

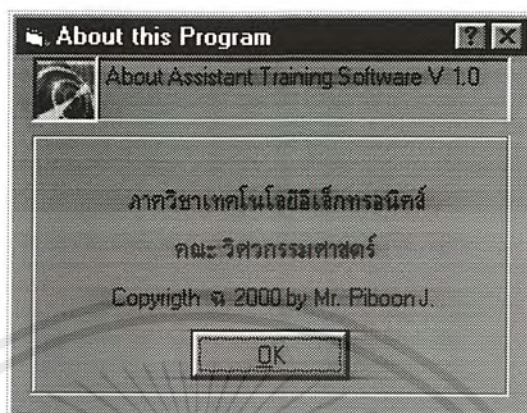
เครื่องแม่ข่ายจะเลือกที่ช่อง (Server) และต้องกำหนดตำแหน่งแชร์ (Share Location) ด้วยส่วนช่อง Configuration IP Address ไม่ต้องใส่

สำหรับเครื่องลูกข่ายจะมีช่องสำหรับให้ผู้ใช้ใส่หมายเลข IP Address ของเครื่องคอมพิวเตอร์แม่ข่ายเท่านั้น

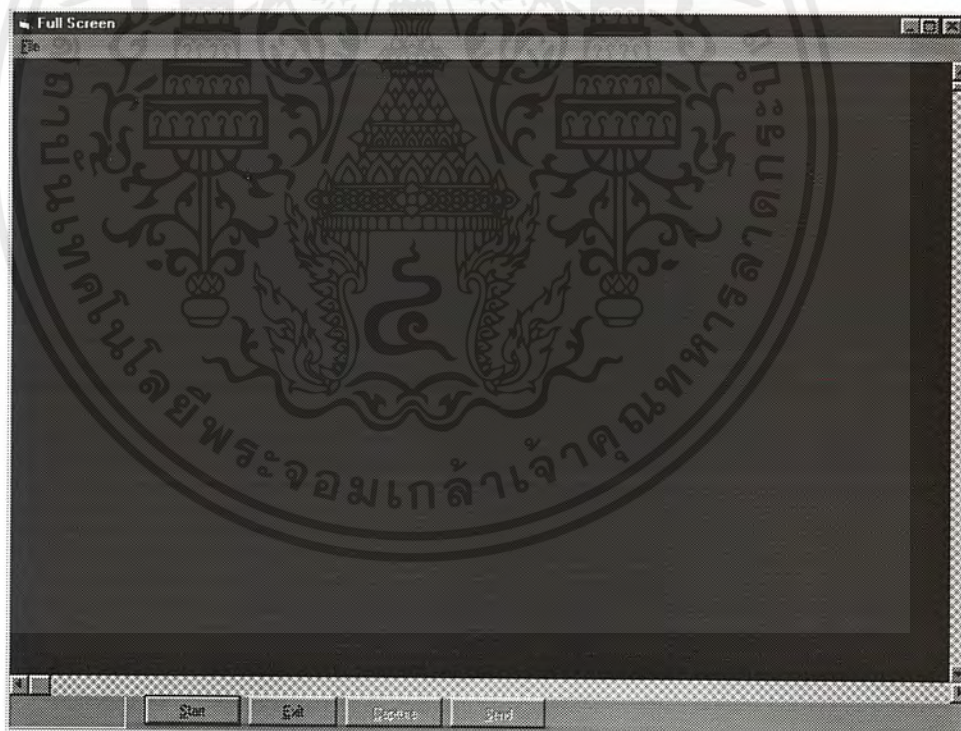
สำหรับรายละเอียดการทำงานของปุ่มต่าง ๆ มีดังนี้

- ปุ่ม Capture ปุ่มนี้จะทำงานได้เมื่อการทำ Connection สำเร็จ โดยการทำงานจะทำหน้าที่เมื่อผู้ใช้กดปุ่มนี้จะไปเปิดฟอร์มใหม่ขึ้นมาเป็นฟอร์มที่จะใช้ดักจับจอภาพที่ต้องการจะส่งไปยังอีกฝ่ายหนึ่ง ซึ่งเมื่อผู้ใช้กดปุ่มก็จะเริ่มจับภาพแล้วส่งออกไปยังเครื่องปลายทาง
- ปุ่ม Exit ใน Full Screen Form จะทำหน้าที่ยกเลิกการจับจอภาพและสิ้นสุดการทำงานของโปรแกรมกลับไปยังโปรแกรมหลัก
- ช่องเลือก Server (เครื่องแม่ข่าย) เพื่อกำหนดให้เครื่องเป็นต้นทางหรือเครื่องแม่ข่าย และช่อง Configuration IP Address จะมีไว้เพื่อใส่หมายเลข IP Address ระหว่างเครื่องคอมพิวเตอร์ที่ต้องการจะติดต่อด้วย โดยจะนำเอาหมายเลข IP Address ที่ผู้ใช้ใส่ในช่องนำไปเริ่มทำ Connection ไปยังเครื่องแม่ข่ายที่จะติดต่อด้วยโดยเมื่อทำสำเร็จแล้วทำให้ผู้ใช้สามารถติดต่อกันได้โดยผู้ใช้จะต้องใส่เป็นเลขจำนวนเต็มตามรูปแบบของ IP Address ดังนี้ XXX.XXX.XXX.XXX
- Status Bar ซึ่งจะอยู่ด้านล่างของโปรแกรมซึ่งจะแสดงเวลา ณ.ปัจจุบันที่ใช้โปรแกรม
- ปุ่ม Exit ใน Main Program จะเป็นการสิ้นสุดการทำงานของโปรแกรมหลัก
- เมนู Windows Toggle Language English-Thai ใช้ทำการสลับภาษาไทยและอังกฤษ
- เมนู Help About ใน Main Program จะเป็นการแสดงรายละเอียดของชื่อและเวอร์ชันของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.5 หน้าจอแสดงรายละเอียดของโปรแกรม



ภาพที่ 3.6 หน้าจอของการดักจับจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทำงานเพื่อติดต่อผ่านระบบเน็ตเวิร์ค

4.1 การพัฒนาในส่วนของการสร้างโปรแกรมติดต่อผ่านระบบเน็ตเวิร์ค

ในที่นี้เราได้สร้างโปรแกรม Assistant Training ในการรับและส่งข้อมูลผ่านเครือข่าย โดยใช้โปรโตคอล TCP/IP ทั้งนี้เราได้เขียนขึ้นโดยใช้โปรแกรม Visual Basic 6.0 พร้อมทั้งได้เรียกใช้ Winsock Control.

Winsock TCP Protocol

ตัวอย่างและขั้นตอนการทำงาน

บนเครื่อง Server Computer

1. กำหนดช่องที่จะใช้ในการสื่อสาร
2. รอรับการขอใช้บริการ
3. เมื่อได้รับคำขอใช้บริการจะตอบสนองต่อความต้องการนั้น ๆ
4. ทำการสื่อสารข้อมูล

บนเครื่อง Client Computer

1. กำหนดหมายเลขเครื่องปลายทางที่จะขอทำการติดต่อ
2. กำหนดช่องสื่อสารเครื่องปลายทางที่จะขอทำการติดต่อ
3. ทำการสื่อสารข้อมูล
4. แจ้งผู้ใช้งานว่าทำการติดต่อสำเร็จ
5. ทำการแลกเปลี่ยนข้อมูล
6. ขอปิดช่องทางการสื่อสาร

การเริ่มต้นใช้งาน Winsock TCP ทำโดยสร้าง Socket แรกเริ่มโดยกำหนดตำแหน่ง 0 ให้ โดยใน Visual Basic ต้องมีการประกาศตัวแปรดังนี้

```
Public gSockInstance As Integer
```

ตัวอย่าง การใช้งาน Winsock ของ Visual Basic จะมี Object ที่จะถูกใช้งานดังนี้

บนเครื่อง Server Computer

- Winsock TCP control named "SktTCPServer"
- Form named "frmServer"
- CommandButton control named "cmdSendData"

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บนเครื่อง Client Computer

- WinSock TCP control named "skt TCP Client"
- CommandButton control named "cmdConnect"
- TextBox control named "txtRecieved"
- Label control named "lblStatus"

Server: กำหนดคุณสมบัติของ PORT ต่าง ๆ

WinSock TCP ของ Server จะทำการกำหนดคุณสมบัติต่าง ๆ ให้ Port ก่อนเป็นอันดับแรก แม้ว่าเราจะสามารถใช้ Port ใด ๆ ก็ได้ แต่หมายเลข Port บางหมายเลขถูกสงวนไว้สำหรับใช้งาน เฉพาะเช่น HTML browser จะใช้ Port หมายเลข 80 Code ส่วนตัวอย่างของโปรแกรมข้างล่างจะใช้ หมายเลข 6666 สำหรับการติดต่อที่ไม่มี Protocol ใหนใช้มัน

```
SkTcpsvr.Local Port = 6666
```

Server: ทำการรอการขอเรียกใช้บริการ

จะมี Port ที่ใช้ในการรอการแยกใช้บริการว่า TCP จะต้องรอการร้องขอใช้บริการจากตัว client Computer ดังแสดงที่ Code ข้างล่างนี้

```
Private Sub frmServer_Load( )

SkTcpsvr. Local Port = 6666 ' Set the Local port.

SkTcpsvr. Listen ' Use the Listen method.

End Sub
```

Client: ต้องการติดต่อกับ Server

เริ่มการติดต่อจะต้องมีการทำการติดต่อก่อน เพื่อที่จะทำการติดต่อได้สำเร็จ Client ต้องใช้ Connect Method โดยจะต้องมีการใช้งาน RemoteHost และ RemotePort RemoteHost จะทำการระบุว่าใครต้องการทำการติดต่อด้วย โดยกำหนดเป็น IP address และ RemotePort ค่ากำหนดว่าจะทำการติดต่อกับ Remote ที่ Port ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdConnect_Click ( )

With sktTCPClient

RemoteHost = "123.123.101.201"

RemotePort = 6666

Connect

End With

End Sub

```

หรืออาจใช้รูปแบบนี้ได้

```

Private Sub cmdConnect_Click ( )

SktClient. Connect "123.123.101.201" , 6666

End Sub

```

Server: การใช้ Accept Method เพื่อตอบรับการติดต่อ

เมื่อ Server ได้รับการขอการติดต่อจาก Client Computer จะใช้ Accept Method ในการตอบสนองเพื่อบอก Client ว่าการติดต่อสำเร็จ เมื่อ Server ได้รับ Request โดยมีการขอทำ Connection Server จะทำการสร้าง Socket ใหม่แล้วให้ Socket ที่สร้างใหม่ทำการตอบสนองแทน

```

Private Sub sktServer _ ConnectionRequest_

```

```

( Index As Integer, ByVal requestID AS Long )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีารนำไปใช้

```

' Increment the global variable.

GSocketInstance = gSocketInstance + 1

Load sktServer (gSocketInstance)

SktServer (gSocketInstance). Accept request ID

```

End Sub

Client: ทำการแจ้ง User ว่า ทำการติดต่อสำเร็จแล้วดัง Code ข้างล่างจะเป็นการบอก User ว่าการติดต่อสำเร็จ

```

Private Sub sktClient Connect ( )

If sktClient. State = sckConnected Then

' Presuming a StatusBar exists, with one panel.

LblStatus. Caption = "Connection Successful !"

End If

```

End Sub

Server: ส่งข้อมูลให้ Client โดยใช้ SendData Method

เมื่อทำการติดต่อสำเร็จเราสามารถส่งข้อมูลไปให้ Computer อีกฝั่งได้โดยใช้ SendData Method ตาม Code ข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdSendData_Click ()

    SktClient. SendData "this is how we begin."

End Sub

```

Client: ใช้วิธีการ Get Data เพื่อรับข้อมูลบนเครื่อง Client เมื่อมีการส่งข้อมูลมาถึง จะใช้ Get Data Method เพื่อทำการรับข้อมูล

```

Private Sub sktServer DataArrival

    ( Index As Integer, byVal bytesTotal As Long )

    Dim vtData ' Declare a variant to hold the data.

    SktServer (Index). GetData vtData , vbString

    TxtReceived . Text = vt Data ' Display the data.

End Sub

```

Client: ปิดการติดต่อโดยใช้ Close Method หลังจากทำการส่งข้อมูลเสร็จสิ้น Client จะทำการยกเลิกการติดต่อโดยใช้ Close Method

```

Private Sub cmdCloseConnection_Click ()

    SktTCPClient . Close

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และมีการปิดการบริการจากบน Server ด้วยโดยการยกเลิกการควบคุมและลดค่าตัวแปรรวมลง

```

Private Sub sktTCPServer _ Close ( Index As Integer )

    SktTCPServer (Index) . Close

    Unload sktTCPServer ( Index ) ' Unload the instance.

    GSocketInstance = gSocketInstance - 1 ' Decrement the ' variable.

End Sub

```

4.1.1 Winsock UPD Control

WinSock UPD จะมีการทำงานเหมือน WinSock TCP ถึงอย่างไรก็ตามมันเป็น Protocol ที่ไม่มีการตรวจสอบการติดต่อเหมือน WinSock TCP การทำการส่งข้อมูลจาก Computer เครื่องหนึ่งไปยัง Computer อีกเครื่องหนึ่ง

บนเครื่องผู้ส่ง

- กำหนด RemoteHost และ RemotePort
- ส่งข้อมูล โดยใช้ Send Data Method

บนเครื่องผู้รับ

- กำหนด Port ที่จะรับข้อมูล
- รับข้อมูล โดยใช้ Get Data Method

การติดต่อ :

ตัวอย่าง การใช้งาน Visual Basic จะใช้ object ดังต่อไปนี้

บนเครื่องผู้ส่ง

- Form named “ frmSend”
- UPD control named “ udpSender”
- CommandButton control named “cmdSendData”
- TextBox control named “ txtSend”

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงในสื่อออนไลน์ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บนเครื่องผู้รับ

- Form named “frmReceiver”
- UDP control named “udpReceiver”
- TextBox control named “txtReceived”

Sender : กำหนดหมายเลขของเครื่อง Computer ที่ต้องการสื่อสารด้วยรวมถึง Port ด้วย

```
Private Sub frmSend_Load ()

    UdpSender . RemoteHost = “ 123.123.101.201”

    UdpSender . RemotePort = 6666

End Sub
```

Receiver: เครื่อง Computer เป็นผู้รับทำการกำหนด Port ที่จะทำการตอบสนองถึงผู้ส่ง

```
Private Sub frmReceiver_Load ()

    UdpReceiver . LocalPort = 6666

End Sub
```

Sender: ส่งข้อความให้ผู้รับ โดยใช้ SendData Method

```
Private Sub cmdSendData_Click ()

    UdpSender . SendData “ Calling all .....”

End Sub
```

Receiver: ใช้ Get Dada Method เพื่อทำการรับ Data ที่ผู้ส่งมาให้ โดยข้อความที่ได้รับจะถูกเขียนลงใน TextBox

```
Private Sub dpREceiver _ DataArrival ( Byval bytesTotal As Long )
```

```
    Dim vtData ' Declare a variant to hold the data.
```

```
    Udpreceiver . GetData vtData , vbString
```

```
    TxtReceived . Text = vtData ' Display the message.
```

```
End Sub
```

ผลการทดลอง

จากการทดลองเขียน โปรแกรมติดต่อผ่านเครือข่ายทั้งสองโปรแกรมโดยใช้ โพรโตคอล TCP/IP เราสามารถที่จะทำการรับและส่งข้อมูลได้จริง

4.2 การพัฒนาในส่วนของโปรแกรมดักจับจอภาพ

ในโลกของวินโดวส์จะมีไลบรารีกราฟิกที่ไม่ขึ้นอยู่กับฮาร์ดแวร์ใด ๆ เรียกว่า GDI (Graphical Device Interface) GDI จะอยู่ตรงกลางระหว่างโปรแกรมของเราและฮาร์ดแวร์ GDI ประกอบไปด้วยฟังก์ชันและออบเจกต์เกี่ยวกับกราฟิกจำนวนมากที่ช่วยให้โปรแกรมของเราสามารถค้นหาลักษณะเฉพาะของฮาร์ดแวร์แต่ละชนิดได้ GDI จะอนุญาตให้เราแสดงภาพกราฟิกในระบบพิกัดใด ๆ แล้วมันจะย่อ/ขยาย หรือแปลงภาพกราฟิกของเราให้อยู่ในระบบพิกัดของฮาร์ดแวร์นั่นเอง ที่ GDI ทำเช่นนี้ได้ เนื่องจากวินโดวส์ได้จัดเตรียมไดรเวอร์สำหรับการแสดงผลทางจอภาพ และเครื่องพิมพ์จำนวนนับร้อย ๆ ชนิดไว้แล้ว เพื่อให้ GDI แปลงคำสั่งกราฟิกของเราให้กลายเป็นคำสั่งหรือชุดคำสั่งที่ตรงกับฮาร์ดแวร์ที่เราใช้

4.2.1 อะไรคือดีไวซ์คอนเท็กซ์

การแสดงผลภาพกราฟิกภายใต้วินโดวส์นั้น พื้นที่ที่จะถูกวาดจะมีลักษณะเฉพาะหลาย ๆ อย่าง เช่น ขนาดของพื้นที่บนจอภาพหรือพื้นที่ของหน้าต่างที่จะออกเครื่องพิมพ์ สีที่ใช้ และสีพื้นของมัน ฟอนต์ ระบบพิกัด และภาพบิตแมป เป็นต้น สิ่งเหล่านี้จะช่วยให้การแสดงผลเป็นไปอย่างถูกต้อง ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่เกินขอบเขตที่ควรจะเป็น หรือสามารถนำวินโดวส์ที่ถูกบังกลับมาแสดงได้อย่างครบถ้วน คุณสมบัติเหล่านี้จะถูกเก็บรวบรวมไว้ในออบเจกต์ที่รู้จักกันว่า ดีไวซ์คอนเท็กซ์ เรียกสั้น ๆ ว่า DC (Device Context) ฟังก์ชัน API จำนวนมากที่เกี่ยวข้องกับการแสดงผล ล้วนต้องส่งผ่านพารามิเตอร์ของดีไวซ์คอนเท็กซ์ทั้งสิ้น

ภายใต้ระบบวินโดวส์จะมีดีไวซ์คอนเท็กซ์ชนิดพิเศษชนิดหนึ่ง เรียกว่า Memory Device Context หมายถึง หน่วยความจำส่วนหนึ่งที่ถูกจำลองเป็นอุปกรณ์แสดงผลซึ่งใช้สำหรับเตรียมภาพทั้งหลายไว้ในหน่วยความจำก่อนส่งออกไปยังอุปกรณ์แสดงผลจริง ๆ ลองสังเกตการสั่งพิมพ์ไฟล์ออกทางเครื่องพิมพ์ จะเห็นว่าหลังจากสั่งพิมพ์ไปแล้ว วินโดวส์จะต้องประมวลผลภาพกราฟิกทั้งหมดไว้ในหน่วยความจำก่อนแล้วจึงส่งต่อไปให้ Printer Spool (เพื่อจัดคิวการพิมพ์ในกรณีสั่งพิมพ์พร้อม ๆ กันหลาย ๆ งาน) สุดท้ายจึงทยอยส่งออกทางเครื่องพิมพ์ บางครั้งผู้ใช้จึงต้องรอนานสักหน่อยกว่าเครื่องพิมพ์จะเริ่มพิมพ์จริง ๆ

ดีไวซ์คอนเท็กซ์เทียบได้กับงานสีของจิตรกร ลองนึกถึงจิตรกรคนหนึ่งที่ต้องวาดรูปพร้อมกันหลาย ๆ รูปในเวลาเดียวกัน แต่ละรูปจะมีขนาดต่างกัน และมีสไตล์ที่ไม่เหมือนกัน เช่น ภาพหนึ่งอาจมีขนาดใหญ่ มีรายละเอียดมาก และใช้สีเข้มเป็นส่วนใหญ่ ส่วนอีกรูปหนึ่งเป็นภาพขนาดเล็กใช้สีอ่อน ๆ และอาจมีรูปอื่น ๆ อีกที่ต้องการวาดไปพร้อม ๆ กัน จิตรกรคนนี้จะต้องแยกงานสีตามจำนวนภาพที่เขาจะวาด เขาไม่ควรจะใช้งานสีเพียงงานเดียวกับทุกภาพวาดของเขา เพราะงานสีของเขาจะบรรจุสีได้จำกัด วางพักพู่กันได้ไม่มาก ซึ่งไม่เพียงพอกับทุกภาพวาด ดังนั้นเขาควรแยกงานสีตามภาพวาด และหลาย ๆ ภาพอาจใช้งานสีร่วมกันก็ได้ ถ้าเป็นภาพสไตล์เดียวกัน

วินโดวส์ได้จัดเตรียมดีไวซ์คอนเท็กซ์มาตรฐานไว้จำนวนหนึ่ง และเก็บไว้เป็นส่วนกลางคือทุกแอปพลิเคชันสามารถนำไปใช้งานได้ เมื่อแอปพลิเคชันใดนำดีไวซ์คอนเท็กซ์อันหนึ่งออกจากคลังเก็บเพื่อนำไปใช้หลังจากใช้งานเสร็จก็ต้องนำมันกลับมาเก็บที่เดิม เพื่อที่มันจะสามารถถูกใช้งานโดยแอปพลิเคชันอื่นได้

วินโดวส์ยังอนุญาตให้เราสร้างดีไวซ์คอนเท็กซ์ขึ้นมาใช้งานเองได้ ถ้าเราไม่ชอบดีไวซ์คอนเท็กซ์มาตรฐานที่มีอยู่แล้ว ในทำนองเดียวกัน คือเมื่อใช้งานดีไวซ์คอนเท็กซ์เสร็จแล้วก็ต้องลบมันทิ้ง เพื่อที่มันจะได้ไม่กลายเป็นขยะอยู่ในระบบวินโดวส์ เพราะแอปพลิเคชันอื่นไม่สามารถนำมันไปใช้งานได้ และก็ไม่สามารถนำไปเก็บไว้ในคลังของดีไวซ์คอนเท็กซ์มาตรฐานได้

4.2.2 โปรแกรมดักจับจอภาพ (Capture Screen)

โปรแกรมนี้เป็นตัวอย่างการขอใช้ดีไวซ์คอนเท็กซ์ของวินโดวส์เดสก์ทอปแล้วก็อบบี้ภาพบิตแมปของเดสก์ทอปมาเก็บไว้ใน Picture box ที่จริงวินโดวส์เดสก์ทอปก็เทียบได้กับออบเจกต์ Screen ใน Visual Basic และภาพบิตแมปของเดสก์ทอป ก็คือทุกสิ่งที่ปรากฏบนจอภาพทั้งจอ

เอง หรือพูดอีกอย่างหนึ่ง ทั้งหมดนี้ก็คือการดักจับหน้าจอ (Capture screen) มาเก็บไว้ใน Picture box ของ Visual Basic

ปกติการดักจับหน้าจอจะกระทำได้โดยผู้ใช้กดปุ่ม Print Screen และนำภาพในคลิปบอร์ดไป Paste ใส่โปรแกรมอื่น เช่น MS Word หรือ PainBrush แต่โปรแกรมนี้แสดงให้เห็นว่าการส่งจากโปรแกรม Visual Basic ก็สามารถทำได้เช่นกัน

เมื่อรันโปรแกรมดักจับจอภาพจะปรากฏหน้าจอ ดังภาพที่ 4.1 ถ้ากดปุ่ม Start Capture แล้วโปรแกรมนี้จะซ่อนตัวเองก่อนเพื่อให้มองไม่เห็นแล้วก็อปปี้น้ำจอเอาไว้ พอเสร็จแล้วก็จะแสดงตัวเหมือนเดิม พร้อมกับนำภาพที่ดักจับไว้มาแสดงใน Picture box แต่เนื่องจากเนื้อที่ที่ใช้แสดงมีขนาดเล็กกว่ารูปภาพที่ดักจับมาได้ ดังนั้น จึงมีแถบเลื่อน (Scroll bar) ตามแนวตั้งและแนวนอนใช้เลื่อนขึ้น/ลง หรือซ้าย/ขวา เพื่อมองดูรูปที่ต้องการ



ภาพที่ 4.1 หน้าจอโปรแกรมดักจับจอภาพ

4.2.3 เพิ่มคำสั่งที่ไม่มีใน VB

ปกติแล้ว Visual Basic ก็มีคำสั่งแสดงผลภาพและข้อความในฟอร์มและ Picture box อยู่แล้ว คำสั่งที่เป็นเมธอด ได้แก่ Print Circle Cls Line PainPicture Point และ Pset ส่วนพรีอเพอร์ตีที่ใช้กำหนดคุณสมบัติของการแสดงผล เช่น CurrentX CurrentY Picture FontName FontSize FontBold เป็นต้น รายละเอียดของเมธอดและพรีอเพอร์ตีเหล่านี้ ผู้อ่านสามารถศึกษาได้จาก Help ของ Visual Basic

เนื่องจากโปรแกรมดักจับจอภาพมีพีเจอร์บางอย่างที่ต้องอาศัยฟังก์ชัน API เท่านั้น การประกาศฟังก์ชัน API ดังตัวอย่างข้างล่าง จึงเปรียบเสมือนการเพิ่มคำสั่งให้กับ Visual Basic แต่การเรียกใช้คำสั่งเหล่านี้ ไม่ใช่ว่าจะเรียกใช้ได้ตามอำเภอใจ จะต้องมึลำดับก่อนหลังของการเรียกใช้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คล้าย ๆ กับการเขียนโปรแกรมจัดการกับไฟล์ จะต้องมีลำดับขั้นตอนการเปิดไฟล์ เลื่อนตัวชี้ไฟล์ เขียนอ่านไฟล์ และปิดไฟล์

```

Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Declare Function GetDesktopWindow&Lib "user32"()
Declare Function GetDC&Lib "user32" (ByVal hwnd&)
Declare Function GetWindowRect&Lib "user32" _
    (ByVal hwnd&, IpRect As RECT)
Declare Function ReleaseDC&Lib "user32" _
    (ByVal hwnd&, ByVal hdc&)
Declare Function BitBlt&Lib "gdi32" _
    (ByVal hDestDC&, _
    ByVal x&, ByVal y&, _
    ByVal nWidth&, ByVal nHeight&, _
    ByVal hSrcDC&, _
    ByVal xSrc&, ByVal ySrc&, _
    ByVal dwRop&)

```

ตัวอย่าง โปรแกรมแสดงการประกาศฟังก์ชัน API

การทำงานของรวมของโปรแกรมนี้อยู่ที่ปุ่ม Start Capture เป็นสำคัญ ดังตัวอย่างข้างล่าง เป็นซับรูทีน ที่ทำหน้าที่ตอบสนองการกดปุ่ม Start Capture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command1_Click()
    Const SRC_COPY = &HCC0020
    Dim Dhwnd&, Dhdc&, R As RECT

    Form1.Hide
    DoEvents
    Dhwnd = GetDesktopWindow
    Dhdc = GetDC(Dhwnd)
    Call GetWindowRect(Dhwnd, R)
    Picture2.Left = 0
    Picture2.Top = 0
    Picture2.Width = R.Right - R.Left
    Picture2.Height = R.Bottom - R.Top
    Call BitBlt(Picture2.hdc, 0, 0, Picture2.Width, _
        Picture2.Height, Dhdc, 0, 0, SRC_COPY)
    Call ReleaseDC(Dhwnd, Dhdc)
    Form1.Show
    Picture1.Refresh
    Picture2.Refresh
    Vscroll1.Max = Picture2.Height
    Vscroll1.Min = Picture1.ScaleHeight
    Vscroll1.Value = Vscroll1.Min
    Hscroll1.Max = Picture2.Width
    Hscroll1.Min = Picture1.ScaleWidth
    Hscroll1.Value = Hscroll1.Min
End Sub

```

ตัวอย่าง โปรแกรมแสดงการทำงาน โดยรวม แสดงความสำคัญที่ปุ่ม Start Capture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่าง ๆ อธิบายได้ดังนี้

- ซ่อนเร็นตัวเองเสียก่อน โดยใช้เมธอด Hide ของฟอร์ม
- ใช้คำสั่ง DoEvents เพื่อเปิดโอกาสให้วินโดวส์ได้อัพเดทหน้าจอที่ถูกบังโดยฟอร์มของเรา ถ้าไม่มีคำสั่งนี้ เราอาจจะดักจับหน้าจอที่วินโดวส์ยังนำส่วนที่ถูกโปรแกรมเราบังมาเติมไม่ทัน
- ใช้คำสั่ง GetDesktop Window เพื่อหาค่าแชนเดิลของวินโดวส์เดสก์ทอป วินโดวส์เดสก์ทอปก็คือ หน้าต่างบานหนึ่งหรือฟอร์มหนึ่งของระบบวินโดวส์นั่นเอง
- ใช้คำสั่ง GetDC เพื่อหาค่าแชนเดิลของดีไวซ์คอนเท็กซ์ของวินโดวส์เดสก์ทอป คำสั่งนี้จะเรียกหลังจากได้ค่าแชนเดิลจากคำสั่ง GetDesktopWindow และยังเป็นการใช้ดีไวซ์คอนเท็กซ์จากระบบวินโดวส์ด้วย
- ใช้คำสั่ง GetWindowRect เพื่อหาขนาดของจอภาพ ซึ่งก็คือความละเอียดของจอภาพที่เซตโดย Control Panel นั่นเอง
- ใช้คำสั่ง BitBlt เพื่อก๊อปปี้รูปภาพที่เก็บอยู่ในดีไวซ์คอนเท็กซ์ของเดสก์ทอปไปเก็บไว้ในดีไวซ์คอนเท็กซ์ของ Picture box ของ Visual Basic
- ใช้คำสั่ง ReleaseDC เพื่อส่งคืนดีไวซ์คอนเท็กซ์กลับสู่ระบบวินโดวส์
- แสดงฟอร์มที่ซ่อนตัวเอง โดยใช้เมธอด Show

4.2.4 คำสั่ง BitBlt

คำสั่งนี้ย่อมาจาก Bit-Block Transfer ใช้สำหรับเคลื่อนย้ายภาพจากดีไวซ์คอนเท็กซ์หนึ่งไปยังอีกดีไวซ์คอนเท็กซ์หนึ่ง โดยสามารถกำหนดขนาดของรูป และตำแหน่งที่จะวางภาพได้ รูปภาพจะประกอบไปด้วยพิกเซลที่จัดเรียงกันเป็นพื้นที่สี่เหลี่ยม ดังนั้น BitBlt สามารถกำหนดได้ว่า การเคลื่อนย้ายพิกเซลจากต้นทางมาวางที่พิกเซลปลายทาง จะกระทำในรูปแบบใด ตัวอย่างเช่น

SRCCOPY หมายถึง วางทับแทนที่พิกเซลปลายทาง

NOTSRCCOPY หมายถึง กลับสีของพิกเซล (เช่น ขาวกลายเป็นดำ) ก่อนวางทับแทนที่พิกเซลปลายทาง

SRCAND หมายถึง แทนที่พิกเซลปลายทางด้วยผลลัพธ์ที่ได้จากการ AND กันของสีของพิกเซลต้นทางและปลายทาง

PATINVERT หมายถึง แทนที่พิกเซลปลายทางด้วยผลลัพธ์ที่ได้จากการ XOR กันของสีของพิกเซลต้นทางและปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

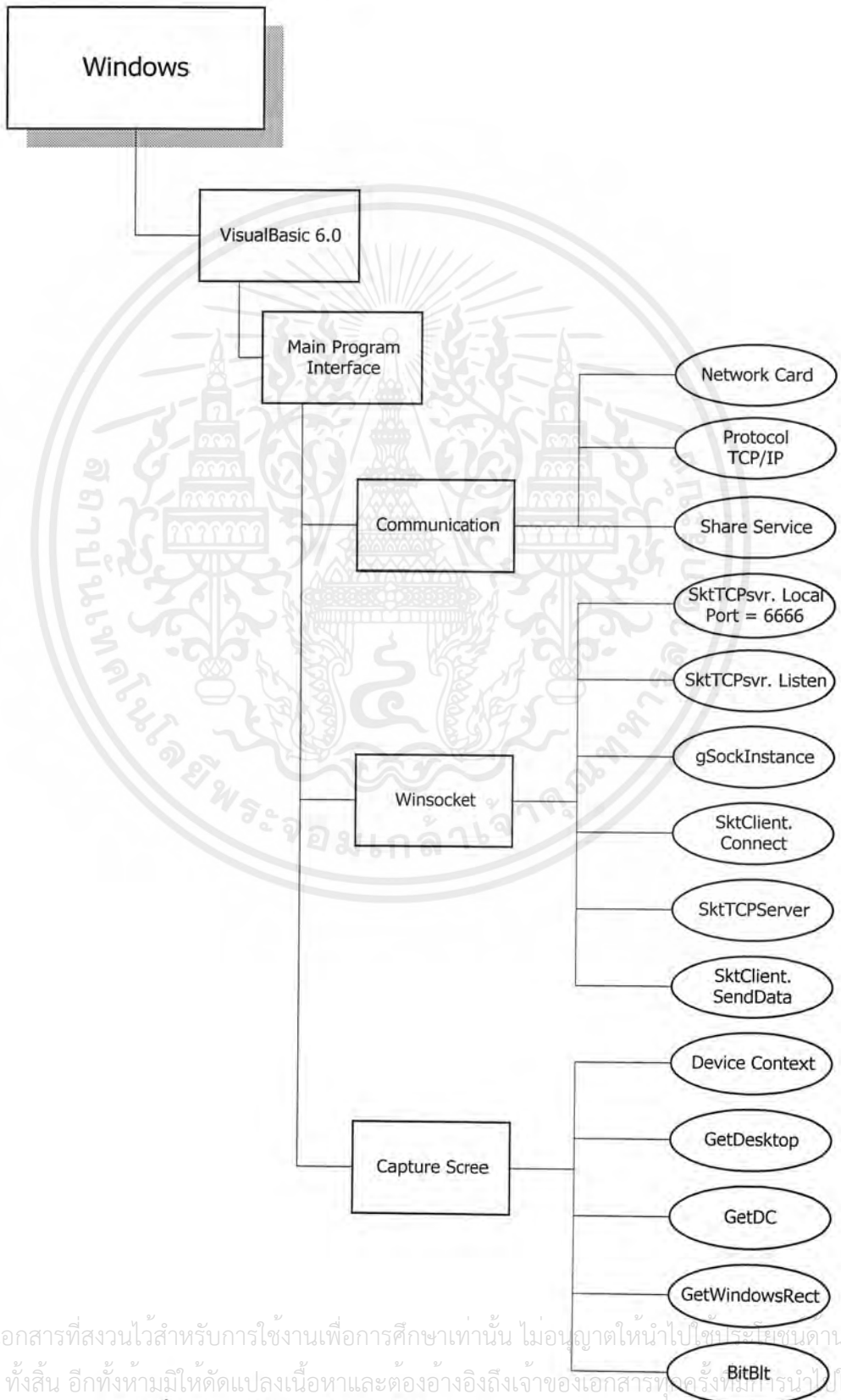
4.2.5 สิ่งที่จะพัฒนาเพิ่มเติม

โปรแกรมดักจับจอภาพนี้เพียงแต่ดักจับหน้าจอแล้วนำมาแสดงที่ฟอร์มอย่างเดียวนั้น ไม่ได้ทำ
อื่นใด ๆ เลย สำหรับท่านผู้สนใจอาจจะพัฒนาต่อเติมไปอีก เช่น เซฟรูปที่ได้ไว้ในไฟล์ (ด้วยคำสั่ง
SavePicture ของ Visual Basic) หรือดักจับหน้าจอเฉพาะบางส่วนของผู้ใช้ที่ต้องการเท่านั้น หรืออื่น ๆ
ตามแต่จินตนาการของท่าน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

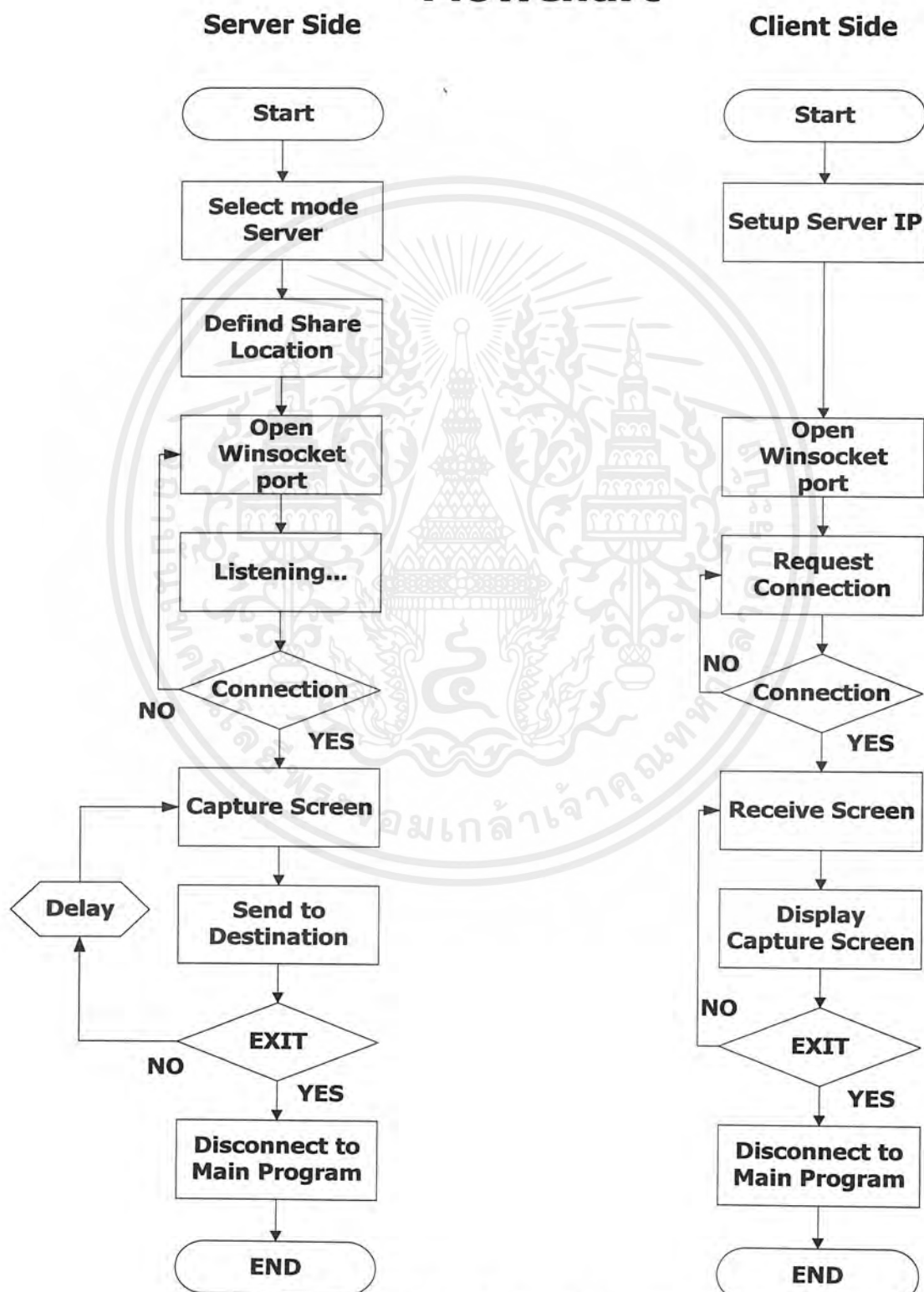
Program Structure



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

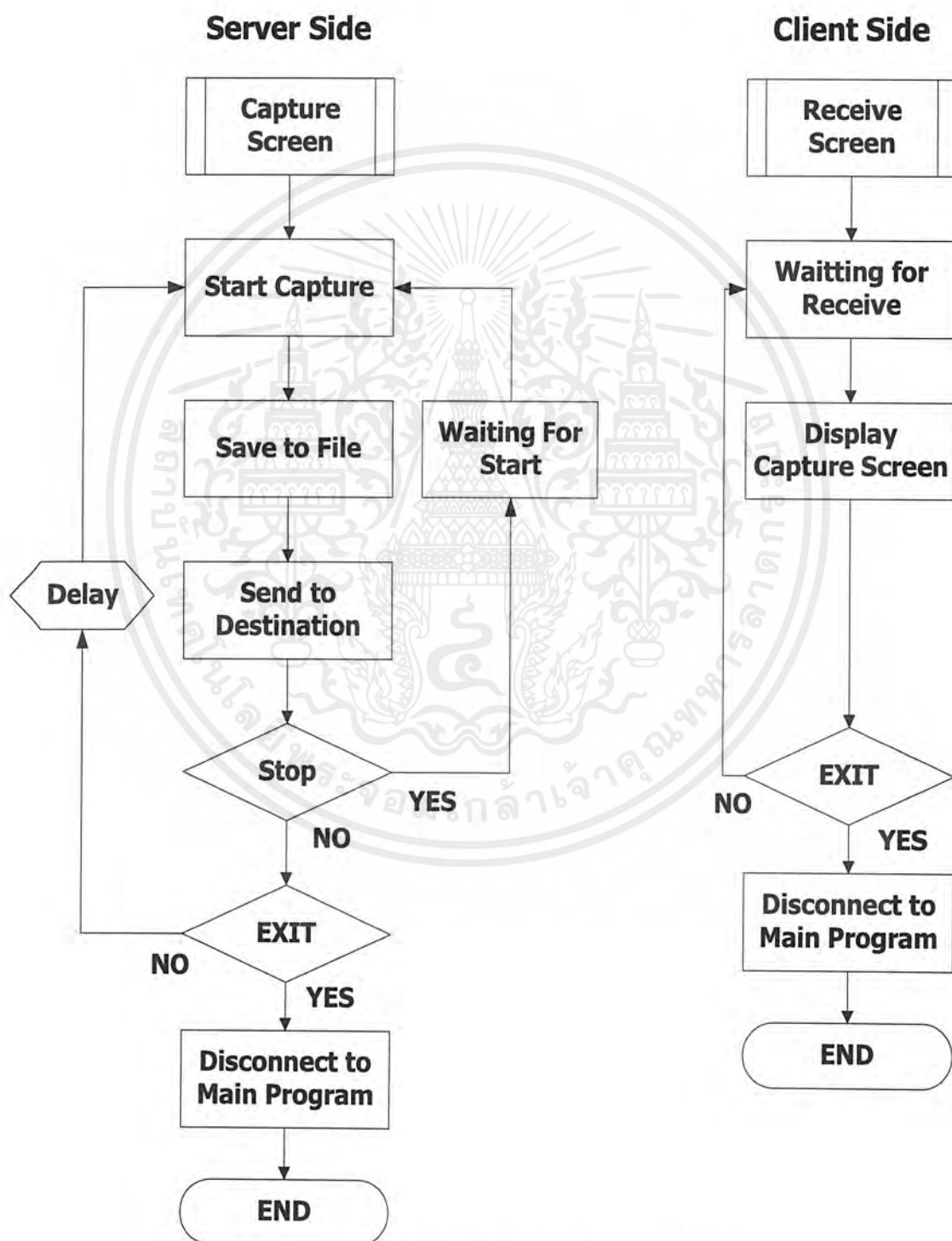
ภาพที่ 4.2 ภาพโครงสร้างของ โปรแกรม (Program Structure)

Assistant Training Program Flowchart



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่ไปนอกเขตหน้าไปใช้ประโยชน์ด้านการค้า
 ภาพที่ 4.3 ภาพไฟร์ชาร์ตรวมของโปรแกรม
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Capture Screen Flow Chart



ภาพที่ 4.4 ภาพโฟรชาร์ทของโปรแกรมส่วนการจับจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและการสรุปผล

5.1 การทดลองพัฒนาในส่วนของการสร้างโปรแกรม

ในที่นี้เราได้สร้างโปรแกรม Assistant Training ได้เขียนขึ้นโดยใช้โปรแกรม Visual Basic 6.0 สำหรับการทดลองสร้าง Form (Main Program) ขึ้นมาก่อนและกำหนดปุ่มต่างๆตามที่ได้ออกแบบไว้ซึ่งในช่วงแรกๆของการศึกษาค้นคว้าโปรแกรม Visual Basic 6.0 จะมีปัญหาในการเขียนคำสั่งต่างๆมากมายทำให้การออกแบบในช่วงแรกจะเป็นการทดลองเขียนโปรแกรมและเรียนรู้การทำงานของโปรแกรมเป็นส่วนใหญ่

ขั้นตอนในการทำโครงการ

- ในช่วงแรกจะใช้เวลาในการคิดและออกแบบตามความต้องการ
- ศึกษาค้นคว้าหาโปรแกรมที่เหมาะสมที่จะนำมาใช้ในการทำโครงการ
- ออกแบบโปรแกรมหลัก
- กำหนดการทำงานของปุ่มควบคุมต่างๆ
- ศึกษาค้นคว้าหาข้อมูลที่จะใช้ในการทำงานเพื่อติดต่อผ่านระบบเน็ตเวิร์ค
- ทดลองเขียนคำสั่งในการทำงานเกี่ยวกับระบบเน็ตเวิร์ค (ยังอยู่ในขั้นศึกษาและทดลอง)

ประโยชน์ที่ได้รับ

- โปรแกรมสามารถที่จะส่งภาพจากเครื่องต้นทางไปยังเครื่องปลายทางที่ต้องการได้
- ในการใช้งานในการสอนวิชาทางด้านคอมพิวเตอร์จะช่วยให้สะดวกมากขึ้นและทำให้ผู้เรียนสามารถติดตามการเรียนการสอนได้อย่างมีประสิทธิภาพมากขึ้นด้วย
- ไม่ต้องใช้ฮาร์ดแวร์เพิ่มในการใช้งานทำให้ประหยัดและมีความคล่องตัวมากในการติดตั้งโปรแกรม

ข้อจำกัดของโปรแกรม

- ผู้ใช้โปรแกรมในโครงการนี้สามารถใช้การติดต่อได้เพียงสองเครื่องเท่านั้นในเบื้องต้น
- ผู้ใช้ต้องทราบหมายเลข IP Address หรือ ชื่อของเครื่องที่ต้องการติดต่อด้วย Computer Name เพื่อไว้ใช้ระบุในการที่ต้องการจะติดต่อเพื่อส่งภาพไป
- ผู้ใช้จะทำการติดตั้งโปรโตคอล TCP/IP ไว้ในเครื่องที่จะใช้งานและต้องกำหนดเลข IP และ Computer Name ไว้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแสดงผลอาจไม่ Real Time มากนักเนื่องจากต้องส่งข้อมูลขนาดใหญ่ผ่านทางระบบเน็ตเวิร์ค
- ผู้ใช้ต้องเปิดโปรแกรมรอรับการติดต่อเข้ามาแต่ถ้าฝั่งปลายทางไม่เปิดโปรแกรมรอรับก็ทำให้ไม่สามารถติดต่อได้

5.2 ปัญหาและอุปสรรคในการทำ โครงการงาน

เนื่องจากในช่วงแรกของการทำโครงการงานข้าพเจ้ายังไม่มีความรู้ในการพัฒนาโปรแกรมโดยการใช้โปรแกรมสำเร็จรูปจึงต้องใช้เวลาส่วนใหญ่ในการศึกษาหาข้อมูลเกี่ยวกับโปรแกรมต่างๆที่จะนำมาใช้ในการทำโครงการที่เหมาะสมซึ่งหนังสือในท้องตลาดที่มีขายอยู่ส่วนใหญ่จะเป็นการแนะนำการเขียนโปรแกรมเบื้องต้นจึงต้องใช้เวลาในการศึกษาหาข้อมูลเพิ่มเติมและหลังจากได้ศึกษาแล้วพบว่าโปรแกรมนี้น่าจะสามารถนำมาพัฒนาในโครงการนี้ได้จึงได้ทดลองศึกษาและเขียนโปรแกรมเบื้องต้นดู

ในช่วงที่สองของโครงการได้ค้นหาข้อมูลและการเรียนรู้เพื่อที่จะใช้โปรแกรม Visual Basic ในการทำการเขียนโปรแกรมผ่านทางระบบเน็ตเวิร์คซึ่งค่อนข้างที่จะหาข้อมูลที่ใกล้เคียงกับความต้องได้ยากมากจึงต้องใช้การประยุกต์การเขียนโปรแกรมที่เขียนแบบการติดต่อแบบอื่นเข้ามาผสมผสานกันเนื่องจากการเขียนโปรแกรมผ่านระบบเน็ตเวิร์คต้องมีความเข้าใจในเรื่องโพรโตคอลและการส่งข้อมูลผ่านทางระบบเน็ตเวิร์คเป็นอย่างมากทำให้เป็นอุปสรรคในการพัฒนาโครงการอย่างมากเนื่องจากข้าพเจ้ามิได้มีพื้นฐานทางด้านการเขียนโปรแกรมทางด้านเน็ตเวิร์คมาก่อนจึงทำให้การพัฒนาเป็นไปอย่างล่าช้ากว่าแผนงานที่วางกำหนดของโครงการเอาไว้

5.3 ข้อเสนอแนะเพื่อการพัฒนาโปรแกรมต่อไปในอนาคต

- ผู้พัฒนาโปรแกรมในโครงการนี้สามารถที่จะพัฒนาให้ใช้การติดต่อได้หลายเครื่องในเวลาเดียวกันและสามารถส่งแบบ Real Time ได้
- พัฒนาการติดต่อผ่านทางระบบเน็ตเวิร์คที่ไม่ต้องมีการระบุหมายเลข IP Address ปลายทางได้โดยให้เครื่องที่เป็นเซิร์ฟเวอร์สามารถเก็บ IP Address ของเครื่องลูกข่ายมาไว้ในฐานข้อมูลได้
- ปรับปรุงและพัฒนาหน้าจอของโปรแกรมให้มีความสวยงามมากขึ้นในอนาคตเพื่อดึงดูดความสนใจของผู้ใช้
- พัฒนาให้สามารถเลือกดูจอภาพของเครื่องลูกข่าย (Client Side) ได้ด้วยในเวลาเดียวกันเพื่อการติดตามการเรียนการสอนของนักศึกษา
- ปรับปรุงและพัฒนาซอฟต์แวร์ที่จำเป็นเพิ่มเติมรวมเข้าไว้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

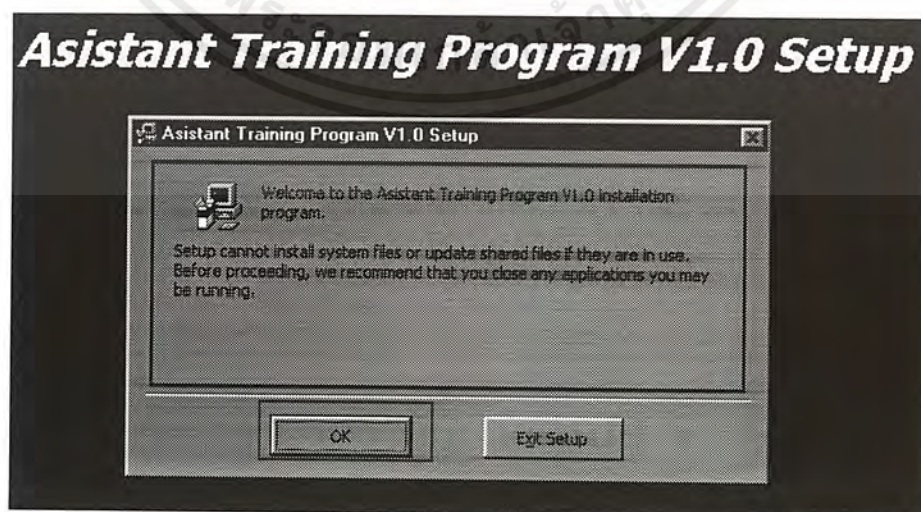
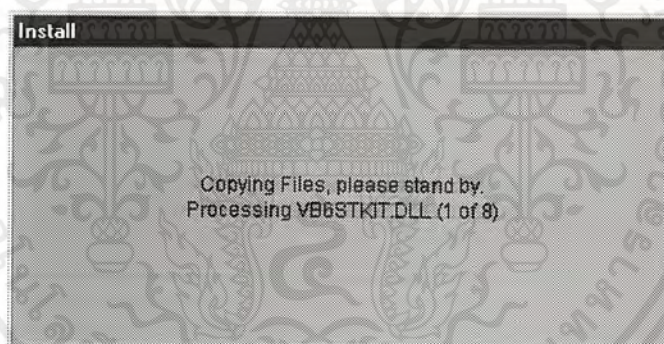
บทที่ 6

การติดตั้งและการใช้งานโปรแกรม

ในบทนี้จะอธิบายถึงขั้นตอนในการติดตั้ง การเชื่อมต่อโปรแกรมช่วยสอนกระทั่งความต้องการของอุปกรณ์และระบบที่ต้องการรวมถึงการใช้งานโปรแกรมอย่างละเอียดทุกขั้นตอนสำหรับอุปกรณ์และระบบที่ต้องการ (Hardware Requirement) ในการติดตั้งโปรแกรมจะอ้างถึงบทที่ 3 หน้า 24 หัวข้อ 3.5 หลังจากการเตรียมอุปกรณ์ให้พร้อมที่จะติดตั้ง ก็จะเริ่มเข้าสู่การติดตั้งโปรแกรมต่อไป

6.1. ขั้นตอนในการติดตั้งโปรแกรมช่วยสอนมีดังต่อไปนี้

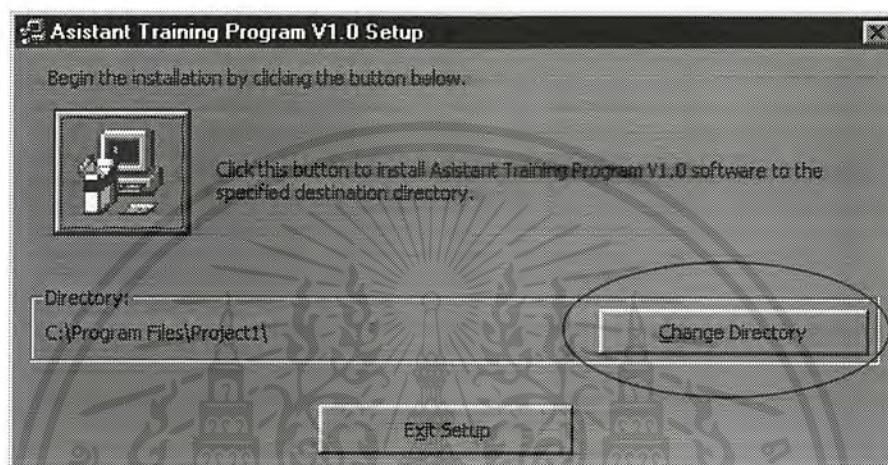
เริ่มต้นจากการรันโปรแกรม Setup.exe ก็จะเข้าสู่ขั้นตอนการติดตั้งดังภาพที่ 6.1



ภาพที่ 6.1 ภาพการติดตั้งหลังจากการรัน Setup.exe

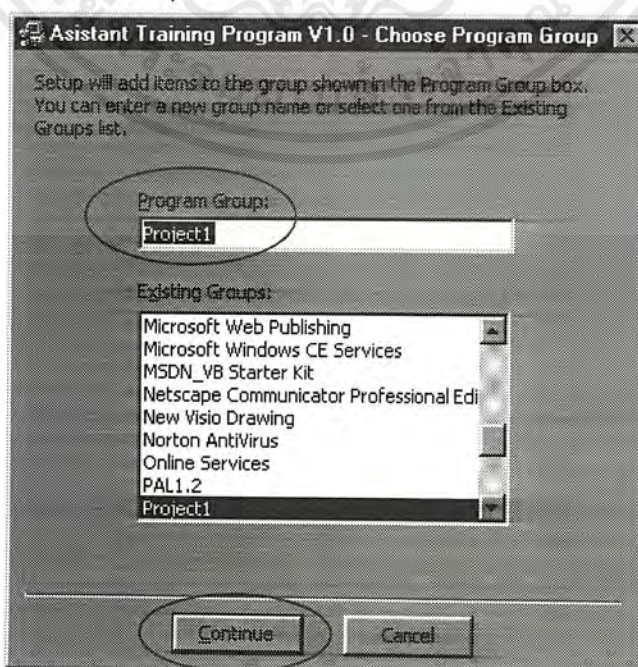
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้กดที่ปุ่ม OK ก็จะทำการทำสำเนาโปรแกรมลงบน Harddisk เพื่อการติดตั้งต่อไป จากนั้นให้เลือกโฟลเดอร์ที่ต้องการจะติดตั้งโดยการกดปุ่ม Change Directory ถ้าไม่ต้องการเปลี่ยนก็ให้กดปุ่มติดตั้ง (Install button) ได้เลย ดังภาพที่ 6.2

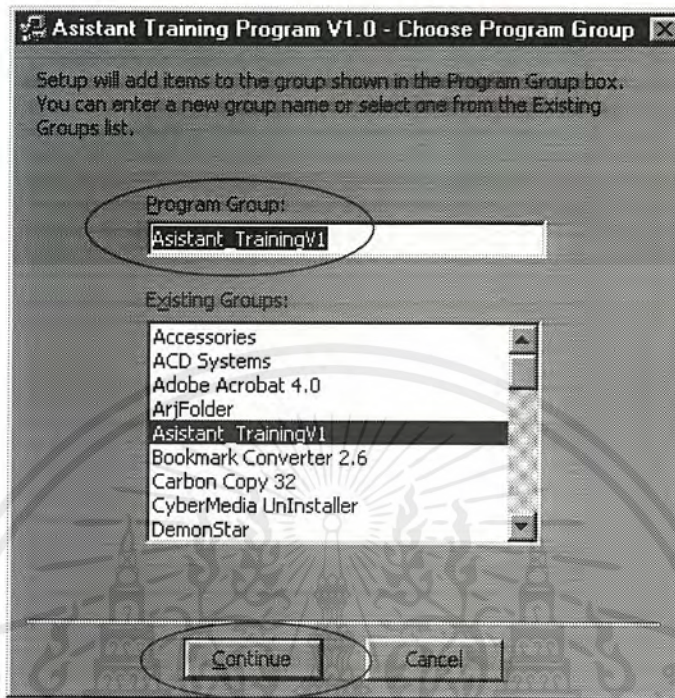


ภาพที่ 6.2 ภาพการเลือกโฟลเดอร์ที่ต้องการจะติดตั้ง

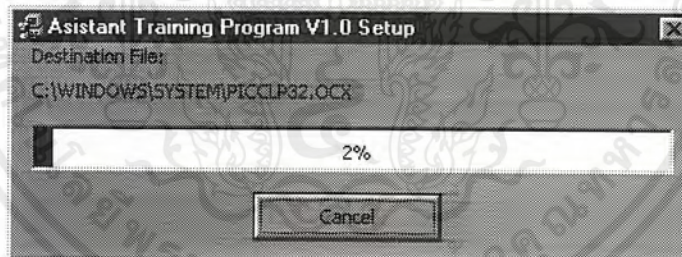
เมื่อเลือกโฟลเดอร์แล้วให้เลือกชื่อกลุ่มโปรแกรม (Program Group) ที่ต้องการที่จะติดตั้งแล้วกดปุ่มต่อเนื่อง (Continue) ดังภาพที่ 6.3 หรือถ้าต้องการเปลี่ยนชื่อกลุ่มโปรแกรมก็สามารถพิมพ์ชื่อใหม่เข้าไปได้ในช่องกลุ่มโปรแกรม ดังภาพที่ 6.4



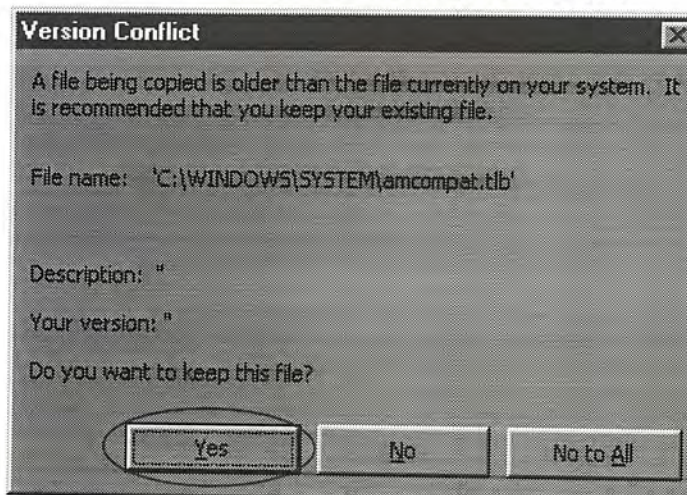
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาต
ภาพที่ 6.3 เลือกชื่อกลุ่มโปรแกรม (Program Group)
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



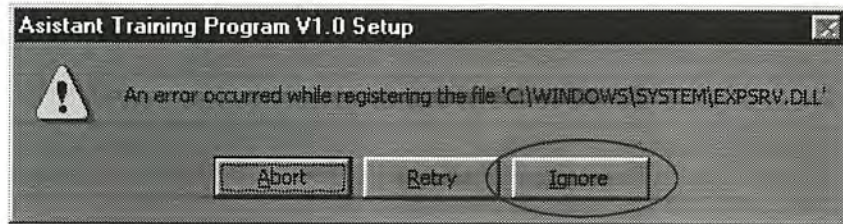
ภาพที่ 6.4 เลือกเปลี่ยนชื่อกลุ่ม โปรแกรม (Program Group)



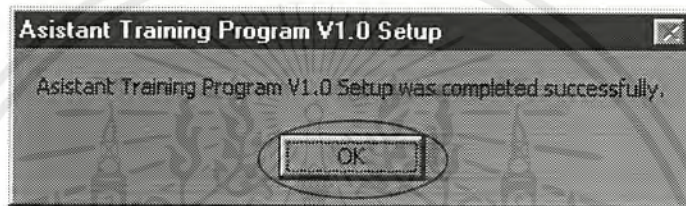
ภาพที่ 6.5 ภาพการติดตั้งไฟล์ต่างๆลงบนฮาร์ดดิส



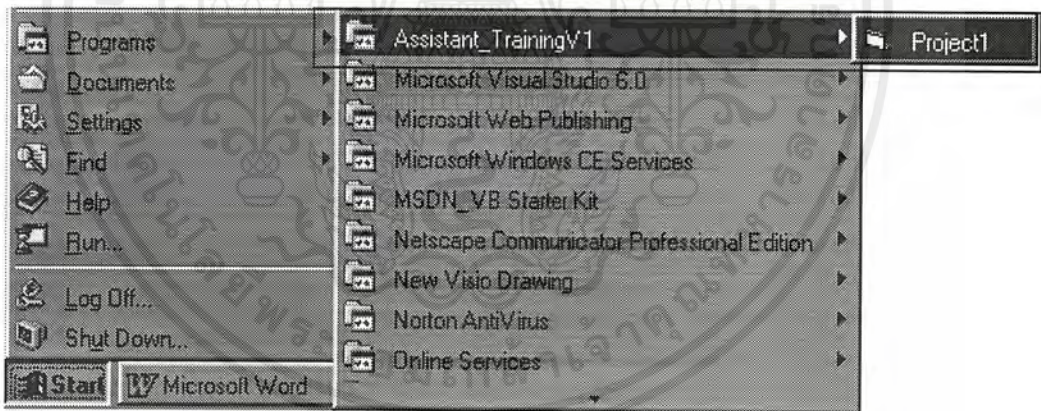
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น หากผู้ใช้ต้องการใช้ประโยชน์ด้านการค้า
 ภาพที่ 6.6 ภาพแสดงโปรแกรมตรวจพบไฟล์ที่จะติดตั้งซ้ำกันก็ให้กดปุ่ม Yes เก็บไฟล์เดิมไว้
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.7 ภาพแสดงโปรแกรมตรวจพบไฟล์ที่ใช้งานอยู่ที่ให้เลือกรุ่น Ignore เพื่อข้ามไป



ภาพที่ 6.8 ภาพแสดงการติดตั้งโปรแกรมสำเร็จแล้วให้กดปุ่ม OK



ภาพที่ 6.9 ภาพแสดงชื่อกลุ่มโปรแกรม (Program Group) ที่สร้างในเมนูโปรแกรม

เมื่อต้องการจะใช้งาน (Run Program) โปรแกรม Assistant_TrainingV1 ให้กดเลือกที่ปุ่มดังนี้

Click at Start → Programs → Assistant_TrainingV1 → Project1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2. การใช้งานโปรแกรม Assistant Training Program V1.0

ก่อนการใช้งาน โปรแกรมจะต้องทำการติดตั้งระบบเน็ตเวิร์คของเครื่องคอมพิวเตอร์ทุกเครื่องพร้อมทั้งติดตั้งโปรโตคอล TCP/IP และ IP Address ของแต่ละเครื่องให้พร้อมและตั้งชื่อของเครื่องแต่ละเครื่องให้แตกต่างกันออกไปดังตัวอย่างต่อไปนี้

Computer Name: Train1	(ชื่อเครื่องคอมพิวเตอร์ทุกเครื่องต้องไม่ซ้ำกัน)
Workgroup: Workgroup	(ชื่อกลุ่มคอมพิวเตอร์กำหนดให้เป็นกลุ่มเดียวกัน)
Share folder as C:\TEMP	(เครื่องแม่ข่ายจะต้องเปิดแชร์โฟลเดอร์ไว้)
File Name XX.BMP	(XX คือชื่อไฟล์ที่เซฟเพื่อใช้ส่งไปแสดงผล)

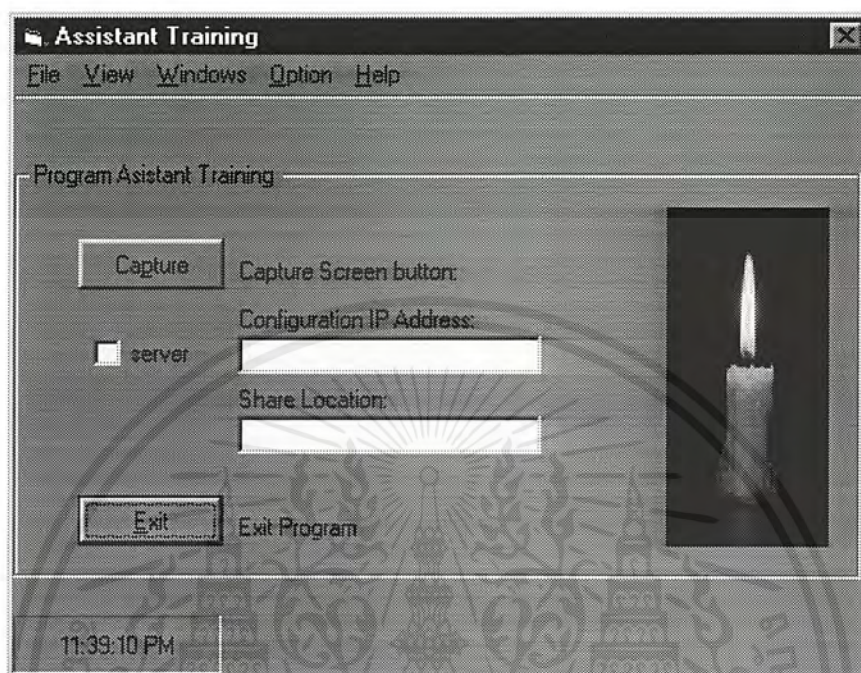
แนะนำว่าเครื่องทุกเครื่องควรใช้ความละเอียดของจอภาพที่ขนาด 800 X 600 และ 16Bit Color จะเหมาะสมในการแสดงผล

สำหรับเครื่องที่จะกำหนดให้เป็นเครื่องแม่ข่ายจะต้องเปิดแชร์โฟลเดอร์ไว้เพื่อให้เป็นที่เก็บภาพที่จะแสดงผลเพื่อส่งไปยังเครื่องปลายทางตัวอย่างเช่น

ตัวอย่าง: รูปแบบของการใช้ Share Service \\Computer Name\share name\file name
ก็จะได้ Share Location ดังนี้
\\Train1\temp\xx.bmp

การใช้งานโปรแกรมจะใช้โปรแกรมเดียวกันทั้งทางเครื่องแม่ข่ายและเครื่องลูกข่าย โดยจะทำการเลือกได้จากการเชื่อมต่อภาพที่ 6.10 เป็นหน้าต่างของโปรแกรม Assistant Training Program V1.0 จะประกอบไปด้วย Menu ต่างๆดังภาพและมีปุ่ม จับภาพ (Capture), ตัวเลือกแม่ข่าย Check Box Server, ช่องใส่เลข IP Address (Configuration IP Address), ช่องใส่ตำแหน่งแชร์โฟลเดอร์ (Share Location), ปุ่มออกจากโปรแกรม (Exit), แถบแสดงสถานะ (Status Bar) จะบอกเวลา ณ.ปัจจุบัน

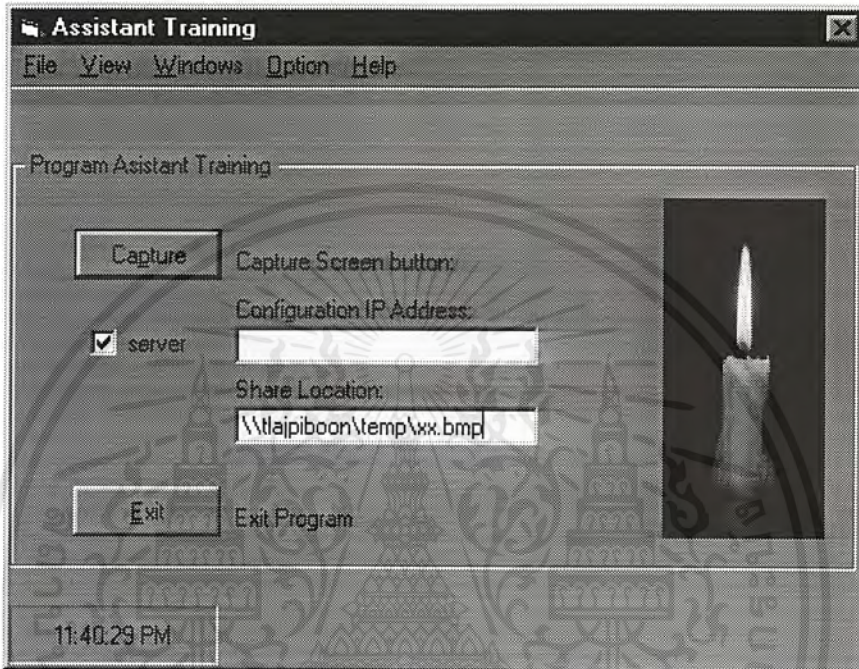
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



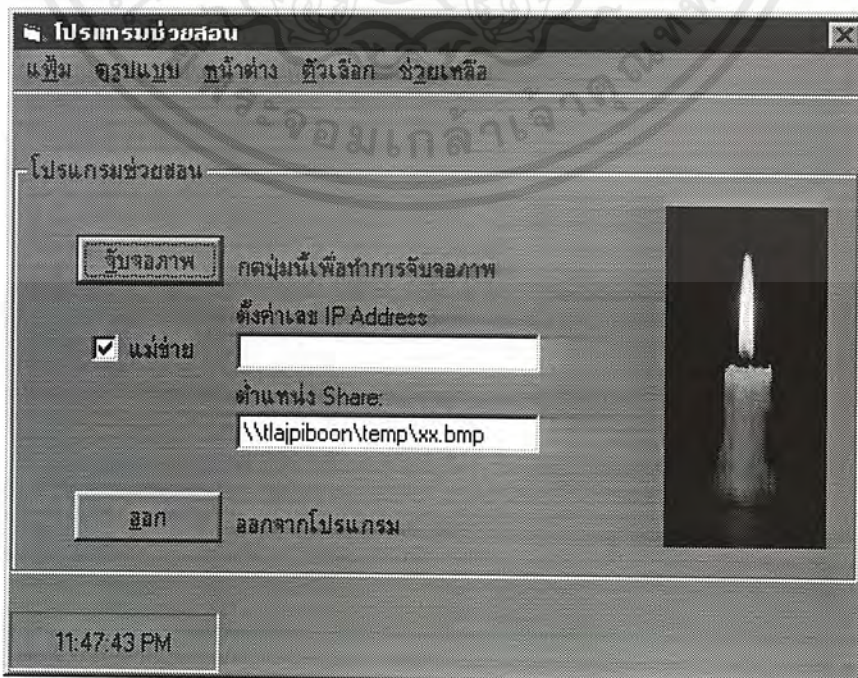
ภาพที่ 6.10 ภาพแสดงหน้าต่างโปรแกรม (Program Interface)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเครื่องแม่ข่ายจะต้องเซตอัป ดังภาพที่ 6.11 ต่อไปนี้จะต้องเลือกที่ช่องแม่ข่ายและใส่ตำแหน่งแชร์ด้วยจากนั้นก็สามารทที่จะกดปุ่มจับภาพได้เพื่อเข้าสู่จอจับภาพต่อไป



ภาพที่ 6.11 ภาพแสดงการเซตอัปให้เป็นเครื่องแม่ข่าย

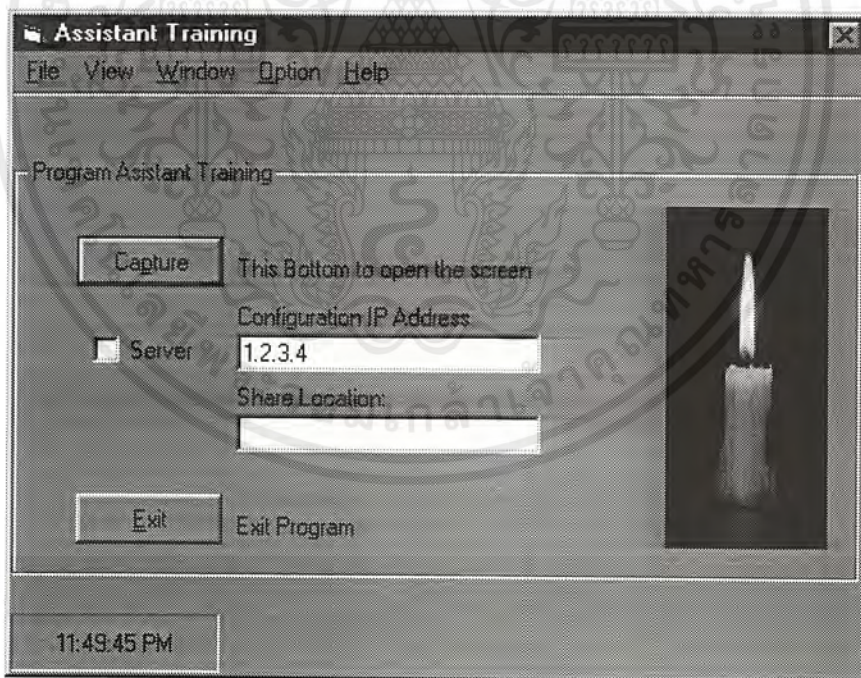


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เอกสารนี้โดยไม่ได้รับอนุญาต

ภาพที่ 6.12 ภาพแสดงการเปลี่ยนภาษาที่เมนูหน้าต่างทุกครั้งที่มีการนำไปใช้

สำหรับเครื่องแม่ข่ายเริ่มการใช้งานให้กดปุ่มจับภาพก็จะเข้าสู่จอ Full Screen เป็นจอจับภาพดังภาพที่ 6.14 เมื่อพร้อมที่จะจับภาพโดยกดที่ปุ่มเริ่ม (Start Button) โปรแกรมก็จะจับภาพจากเมมโมรี่มาเก็บไว้ในพื้นที่จับภาพของโปรแกรกดังภาพที่ 6.15 จากนั้นก็จะทำการแปลงภาพที่เป็นวัตถุ (Object Device Context) มาเซฟเป็นไฟล์ภาพแบบบิตแมป (Bitmap File) แล้วจึงทำการส่งภาพที่ได้ส่งไปยังเครื่องปลายทางผ่านทาง WinSocket port ที่ได้ทำการติดต่อไว้และในส่วนของแถบแสดงสถานะ (Status Bar) จะแสดงคำว่าส่ง (Sending...) ก็จะส่งภาพที่จับมาได้ออกไปทุกๆ 3 วินาที ถ้าต้องการจะหยุดภาพก็สามารถกดที่ปุ่มหยุด (Stop) ได้โปรแกรมก็จะหยุดส่งภาพ

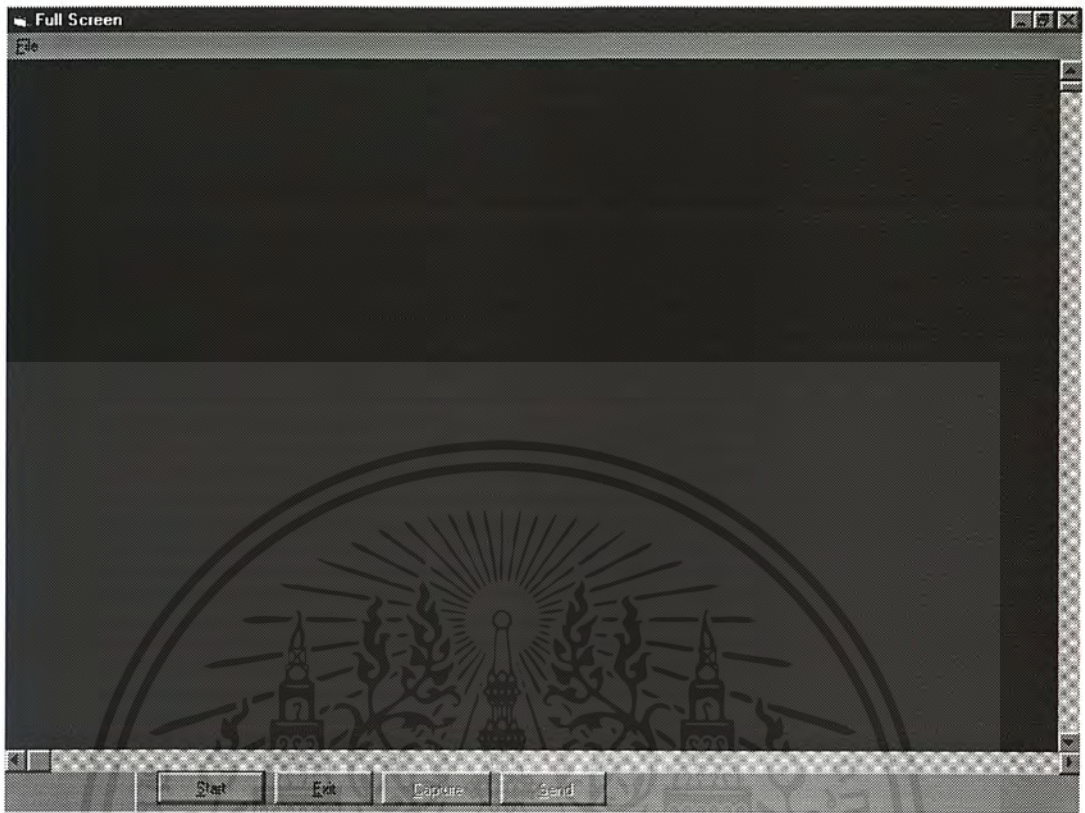
สำหรับเครื่องลูกข่ายจะต้องเซตอัปเดตภาพที่ 6.13 โดยจะต้องใส่เฉพาะหมายเลข IP ของเครื่องแม่ข่ายเพียงเท่านั้น จากนั้นก็สามารถที่จะกดปุ่มจับภาพได้เพื่อเข้าสู่หน้าจอจับภาพต่อไปจากภาพเป็นการใส่เลข IP Address (ยกตัวอย่างเช่น IP 1.2.3.4 ซึ่งเป็นเลข IP Address ของเครื่องแม่ข่าย)



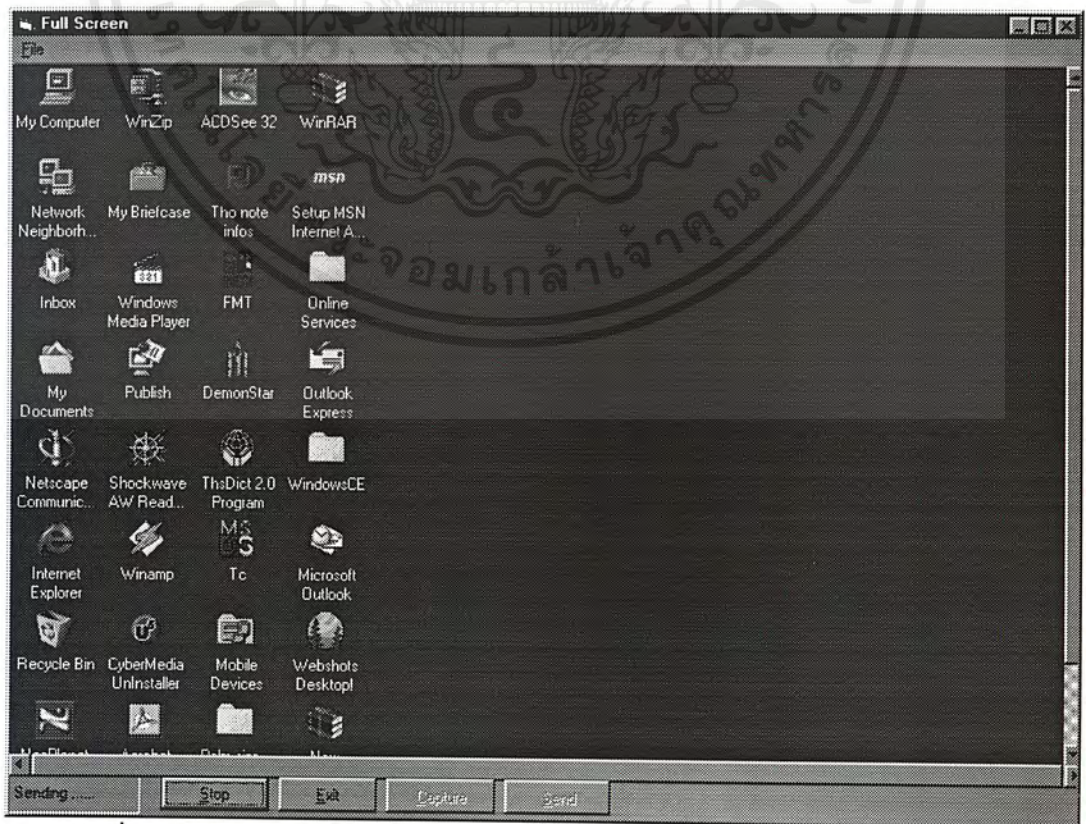
ภาพที่ 6.13 ภาพแสดงการเซตอัปเดตของเครื่องลูกข่าย

สำหรับเครื่องลูกข่ายเริ่มการใช้งานให้กดปุ่มจับภาพเช่นกันก็จะเข้าสู่จอ Full Screen เป็นจอจับภาพดังภาพที่ 6.16 แต่ที่แถบแสดงสถานะจะมีปุ่มออก (Exit) และจะแสดงคำว่ารับ (Receiving...) และพร้อมที่จะรับภาพจากเครื่องแม่ข่ายมาแสดงเหมือนกับทางด้านต้นทาง เมื่อต้องการหยุดรับภาพให้กดปุ่มออกเพื่อกลับไปสู่หน้าจอหลักของโปรแกรม

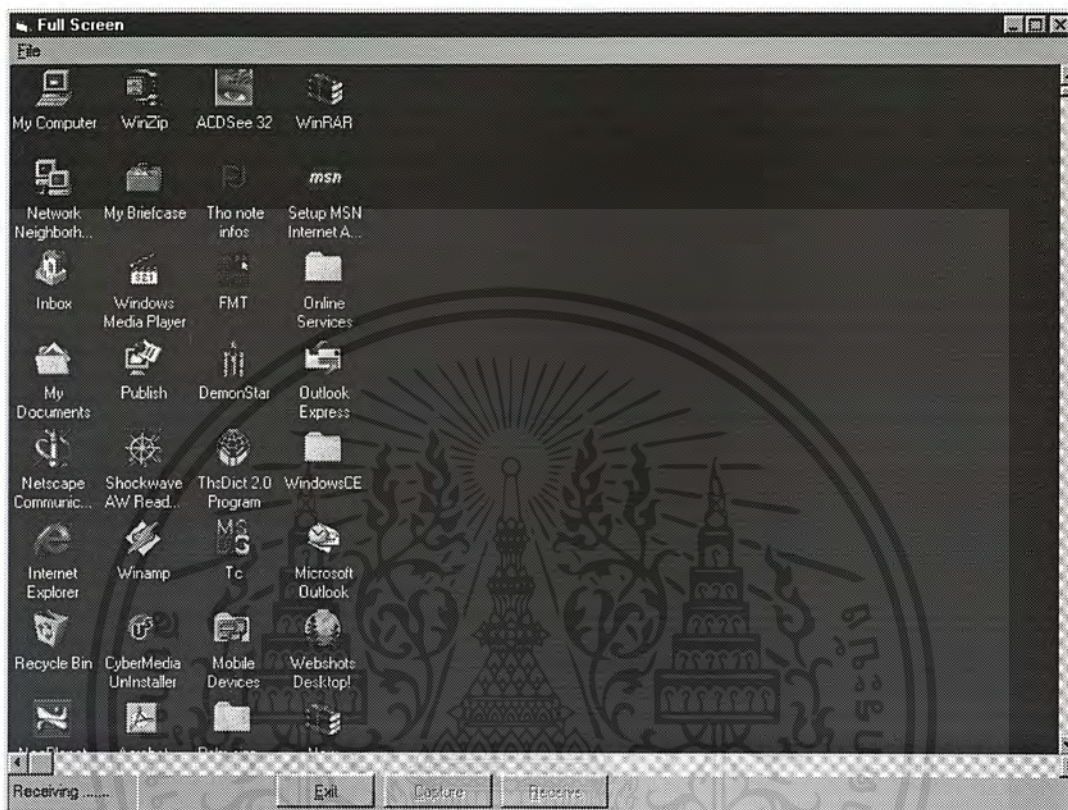
เอกสารนี้จัดทำขึ้นเพื่อแจกจ่ายฟรีโดยไม่คิดมูลค่าให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.14 จอ Full Screen เป็นจอจับภาพ



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.16 ภาพทางด้านลูกข่ายที่รับภาพได้จากเครื่องแม่ข่าย
 ชื่อนแนะนำในการใช้โปรแกรมควรจะทำกรเปิดโปรแกรมแม่ข่ายเพื่อเข้าสู่หน้าจอจับภาพ
 (Full Screen) ก่อนทางด้านเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

About - 1

```
Private Sub Command1_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Height = 3045: Width = 3975    'Fix About Form Size
```

```
    'Fix Position of About Form
```

```
    About.Left = (Screen.Width - Width) / 1 / 2
```

```
    About.Top = (Screen.Height - Height) / 1 / 2
```

```
    About.DrawMode = 4
```

```
    '-----  
    'About This Program form Caption
```

```
    Label1.Caption = "ภาควิชาเทคโนโลยีอิเล็กทรอนิกส์"
```

```
    Label2.Caption = "คณะวิศวกรรมศาสตร์"
```

```
    Label3.Caption = "Copyright © 2000 by Mr. Piboon J."
```

```
    Label5.Caption = " About Assistant Training Software V 1.0"
```

```
End Sub
```

```
Private Sub Label4_Click()
```

```
End Sub
```

```
Private Sub Image1_Click()
```

```
End Sub
```

```
Private Sub Picture1_Click()
```

```
    'Load Icon picture
```

```
    Picture1.AutoRedraw = True ' Set AutoRedraw is True.
```

```
    ' Load an icon into the Image control.
```

```
    Picture1.Picture = LoadPicture("myproject\secast.ico")
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERSION 5.00

Begin VB.Form About

Caption = "About this Program"

ClientHeight = 2625

ClientLeft = 4365

ClientTop = 2175

ClientWidth = 3855

LinkTopic = "Form1"

MaxButton = 0 'False

MinButton = 0 'False

PaletteMode = 1 'UseZOrder

ScaleHeight = 2625

ScaleWidth = 3855

WhatsThisButton = -1 'True

WhatsThisHelp = -1 'True

Begin VB.PictureBox Picture1

Height = 495

Left = 120

Picture = (Icon)

ScaleHeight = 435

ScaleWidth = 435

TabIndex = 6

Top = 0

Width = 495

End

Begin VB.CommandButton Command1

Caption = "&OK"

Height = 375

Left = 1320

TabIndex = 0

Top = 2040

Width = 1215

End

Begin VB.Frame Frame1

Height = 2055

Left = 120

TabIndex = 1

Top = 480

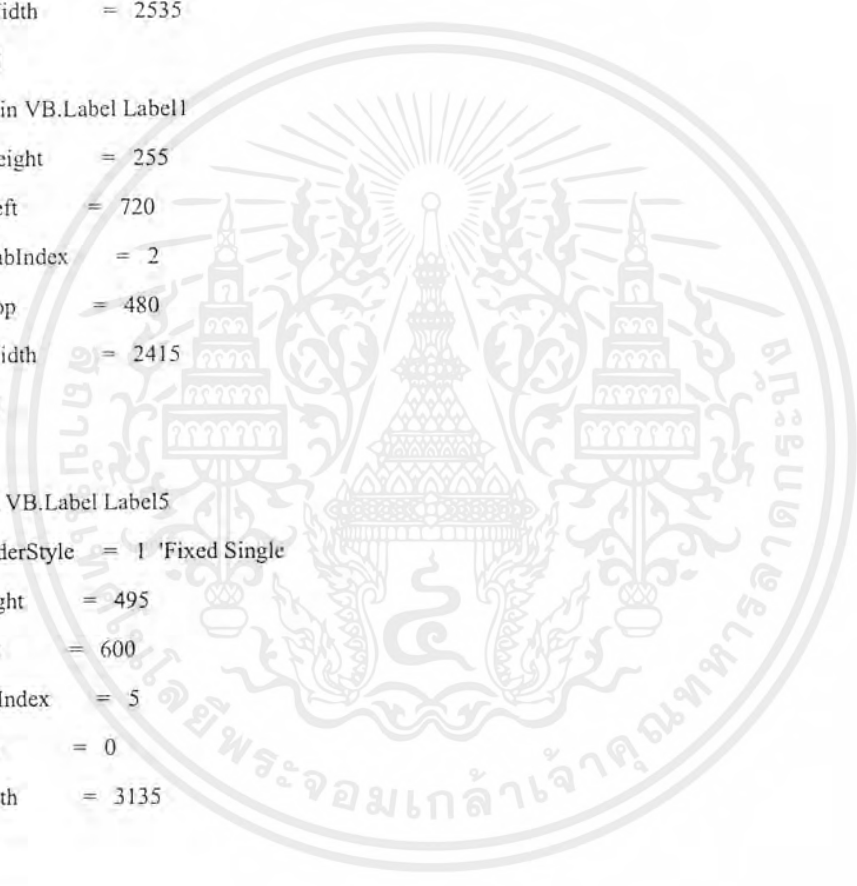
Width = 3615

Begin VB.Label Label2

Height = 255

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Left      = 1080
TabIndex  = 4
Top       = 840
Width     = 1575
End
Begin VB.Label Label3
    Height  = 375
    Left    = 600
    TabIndex = 3
    Top     = 1200
    Width   = 2535
End
Begin VB.Label Label1
    Height  = 255
    Left    = 720
    TabIndex = 2
    Top     = 480
    Width   = 2415
End
End
Begin VB.Label Label5
    BorderStyle = 1 'Fixed Single
    Height      = 495
    Left       = 600
    TabIndex   = 5
    Top        = 0
    Width      = 3135
End
End
```



capture - 1

Global up_link(50) As Integer

Global robin As Integer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form1 - 1

Option Explicit

Dim English As Boolean '

Dim intMax As Integer

Private Sub cmdCancel_Click()

 Unload Me

End Sub

Private Sub cmdCapScr_Click()

 Form1.Hide

 FullScreen.Show

End Sub

Private Sub cmdExit_Click()

 End 'Quit the Program

End Sub

Private Sub cmdFScreen_Click()

 FullScreen.Show

End Sub

Private Sub cmdIPbottom_Click()

 IPcfg.Show 'Open IP Configuration Form

End Sub

Private Sub Form_Load()

 Dim i As Integer

 rrobin = 1

 For i = 1 To 50

 up_link(i) = -1

 Next

 Form1.Left = (Screen.Width - Width) / 1 / 2

 Form1.Top = (Screen.Height - Height) / 1 / 2

 Height = 5000: Width = 6500 'Set Fix Form size

 English = True

 StatusBar1.Panels(1).Text = Now 'Show Date, Time, Now on StatusBar

 StatusBar1.Panels(1).Text = Time()

 Timer1.Interval = 1000

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Form1 - 2

```
Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Image1_Click()
```

```
Image1.Stretch = True
```

```
Image1.Picture = LoadPicture("c:\myproject\chaina.gif")
```

```
End Sub
```

```
Private Sub mnuAboutT_Click()
```

```
About.Show 'Open About Form
```

```
End Sub
```

```
Private Sub mnuCommu_Click()
```

```
IPefg.Show
```

```
End Sub
```

```
Private Sub mnuCapture_Click()
```

```
Form1.Hide
```

```
FullScreen.Show
```

```
End Sub
```

```
Private Sub mnuExit_Click()
```

```
End
```

```
End Sub
```

```
Private Sub mnuHelpTopic_Click()
```

```
Dim Msg ' Declare variable.
```

```
Hide ' Hide form.
```

```
Msg = "Under Construction Please Choose Ok Back to Main Program"
```

```
MsgBox Msg ' Display message.
```

```
Show ' Show form again.
```

```
End Sub
```

```
Private Sub StatusBar1_PanelClick(ByVal Panel As Panel)
```

```
StatusBar1.Panels(1).Text = Now 'Show Date Time on statusbar Now
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

เอกสารนี้เป็น Status Bar1.Panels(1).Text = Time() ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

Form1 - 3

```
Private Sub Winsock1_Close(index As Integer)
```

```
Dim i As Integer
```

```
Winsock1(index).Close
```

```
For i = 1 To 50
```

```
    If up_link(i) = index Then
```

```
        up_link(i) = -1
```

```
    Exit For
```

```
End If
```

```
Next
```

```
End Sub
```

```
Private Sub Winsock1_ConnectionRequest(index As Integer, ByVal requestID As Long)
```

```
Dim i As Integer
```

```
If index = 0 Then
```

```
    intMax = intMax + 1
```

```
    Load Winsock1(intMax)
```

```
    Winsock1(intMax).LocalPort = 0
```

```
    Winsock1(intMax).Accept requestID
```

```
For i = 1 To 50
```

```
    If up_link(i) = -1 Then
```

```
        up_link(i) = intMax
```

```
    Exit For
```

```
End If
```

```
Next
```

```
End If
```

```
End Sub
```

```
Private Sub Winsock1_DataArrival(index As Integer, ByVal bytesTotal As Long)
```

```
Dim buf As String
```

```
Form1.Winsock1(index).GetData buf
```

```
On Error GoTo err_loadpic
```

```
FullScreen.Picture2.Picture = LoadPicture(buf)
```

```
GoTo end_sub
```

```
err_loadpic:
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end_sub:
```

Form1 - 4

End Sub

Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)

Dim buf As String

Winsock2.GetData buf

On Error GoTo err_loadpic_client

FullScreen.Picture2.Picture = LoadPicture(buf)

Winsock2.Close

Form1.Winsock2.RemotePort = 6666

Form1.Winsock2.RemoteHost = Form1.tep_addr.Text

Winsock2.Connect

GoTo end_sub_client

err_loadpic_client:

MsgBox "Error: can't load picture"

end_sub_client:

End Sub

Private Sub winToggle_Click()

If English = True Then

English = False

Form1.Caption = "โปรแกรมช่วยสอน"

mnuFile.Caption = "&แฟ้ม"

mnuExit.Caption = "&ออก"

mnuNew.Caption = "ไ&้ใหม่"

mnuOpen.Caption = "เ&้เปิด"

mnuSave.Caption = "&จัดเก็บ"

mnuSaveAs.Caption = "จัด&เก็บเป็น..."

mnuView.Caption = "ค&ูรูป&แบบ"

mnuToolbar.Caption = "แถบ&เครื่องมือ"

mnuLarge.Caption = "ใ&้ใหญ่"

mnuSmall.Caption = "ใ&้เล็ก"

mnuNone.Caption = "ใ&้ปกติ"

mnuStatus.Caption = "ใ&้สถานะ"

mnuWindow.Caption = "ใ&้หน้าต่าง"

winToggle.Caption = "&สลับภาษา ไทย-อังกฤษ"

mnuOption.Caption = "&ตัวเลือก"

mnuCapture.Caption = "ใ&้จับจอภาพ"

mnuHelp.Caption = "ใ&้ช่วยเหลือ"

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ขอสงวนสิทธิ์ในเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mnuHelpTopic.Caption = "&หัวข้อช่วยเหลือ"
mnuUsingHelp.Caption = "&การใช้หัวข้อช่วยเหลือ"
mnuAboutT.Caption = "&เกี่ยวกับโปรแกรม..."
cmdCapScr.Caption = "&จับจอภาพ"
'cmdIPbottom.Caption = "ตั้งค่า&IP"
srvopt.Caption = "แม่ข่าย"
Label2.Caption = "ตำแหน่ง Share:"
cmdExit.Caption = "&ออก"
Frame1.Caption = "โปรแกรมช่วยสอน"
lbCapbot.Caption = "กดปุ่มนี้เพื่อทำการจับจอภาพ"
lbExit.Caption = "ออกจากโปรแกรม"
lbIPcfg.Caption = "ตั้งค่าเลข IP Address"
FullScreen.Command2.Caption = "&ออก"
'FullScreen.Command3.Caption = "&เริ่ม"

```

Else

```

English = True
Form1.Caption = "Assistant Training"
mnuFile.Caption = "&File"
mnuNew.Caption = "&New"
mnuOpen.Caption = "&Open"
mnuSave.Caption = "&Save"
mnuSaveAs.Caption = "Save &As..."
mnuExit.Caption = "&Exit"
mnuView.Caption = "View"
mnuToolbar.Caption = "Toolbar"
mnuLarge.Caption = "Large"
mnuSmall.Caption = "Small"
mnuNone.Caption = "Normal"
mnuStatus.Caption = "Status"
mnuWindow.Caption = "&Window"
winToggle.Caption = "&Toggle English-Thai"
mnuOption.Caption = "&Option"
mnuCapture.Caption = "Ca&pture"
mnuHelp.Caption = "&Help"
mnuHelpTopic.Caption = "Help &Topic"
mnuUsingHelp.Caption = "&Using Help"
mnuAboutT.Caption = "&About Training Project..."
cmdCapScr.Caption = "Ca&pture"
cmdIPbottom.Caption = "&IP Config"
srvopt.Caption = "Server"
Label2.Caption = "Share Location:"

```

เอกสารนี้เป็น **cmdIPbottom.Caption = "&IP Config"** เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ **srvopt.Caption = "Server"** ห้ามตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label2.Caption = "Share Location:"

```
cmdExit.Caption = "&Exit"  
Frame1.Caption = "Program Asistant Training"  
lbCapbot.Caption = "This Bottom to open the screen capture"  
lbExit.Caption = "Exit Program"  
lblPcfig.Caption = "Configuration IP Address"  
FullScreen.Command2.Caption = "&Exit"  
'FullScreen.Command3.Caption = "&Start"  
End If  
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form1 - 1

VERSION 5.00

Object = "{6B7E6392-850A-101B-AFC0-4210102A8DA7}#1.3#0"; "COMCTL32.OCX"

Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"

Begin VB.Form Form1

AutoRedraw = -1 'True

BorderStyle = 1 'Fixed Single

Caption = "Assistant Training"

ClientHeight = 4110

ClientLeft = 1650

ClientTop = 2025

ClientWidth = 6405

LinkTopic = "Form1"

MaxButton = 0 'False

MinButton = 0 'False

PaletteMode = 1 'UseZOrder

ScaleHeight = 4110

ScaleWidth = 6405

WhatsThisHelp = -1 'True

Begin ComctlLib.Toolbar Toolbar1

Align = 1 'Align Top

Height = 420

Left = 0

TabIndex = 3

Top = 0

Width = 6405

_ExtentX = 11298

_ExtentY = 741

Appearance = 1

_Version = 327682

End

Begin VB.Timer Timer1

Left = 6600

Top = 2280

End

Begin MSWinsockLib.Winsock Winsock2

Left = 6600

Top = 2760

_ExtentX = 741

_ExtentY = 741

End

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ Version อีกครั้ง 393216 ได้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Begin VB.TextBox tcp_addr
```

```
Height = 285
Left = 1680
TabIndex = 10
Top = 1800
Width = 2295
```

```
End
```

```
Begin VB.CheckBox srvopt
```

```
Caption = "server"
Height = 255
Left = 600
TabIndex = 9
Top = 1800
Width = 855
```

```
End
```

```
Begin MSWinsockLib.Winsock Winsock1
```

```
Index = 0
Left = 6600
Top = 3360
_ExtentX = 741
_ExtentY = 741
_Version = 393216
```

```
End
```

```
Begin VB.CommandButton cmdCapSer
```

```
Caption = "Ca&pture"
Default = -1 'True
Height = 375
Left = 480
TabIndex = 1
Top = 1080
Width = 1095
```

```
End
```

```
Begin VB.CommandButton cmdExit
```

```
Cancel = -1 'True
Caption = "&Exit"
Height = 375
Left = 480
TabIndex = 0
```

```
Top = 3000
```

```
Width = 1095
```

```
End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Begin VB.Frame Frame1
```

```
Caption = "Program Asistant Training "
```

```
Height = 3135
```

```
Left = 0
```

```
TabIndex = 4
```

```
Top = 480
```

```
Width = 6375
```

```
Begin VB.TextBox sharename
```

```
Height = 285
```

```
Left = 1680
```

```
TabIndex = 11
```

```
Top = 1920
```

```
Width = 2295
```

```
End
```

```
Begin VB.Label Label2
```

```
Caption = "Share Location:"
```

```
Height = 255
```

```
Left = 1680
```

```
TabIndex = 12
```

```
Top = 1680
```

```
Width = 1215
```

```
End
```

```
Begin VB.Label lblPcfig
```

```
Caption = "Configuration IP Address:"
```

```
Height = 255
```

```
Left = 1680
```

```
TabIndex = 7
```

```
Top = 1080
```

```
Width = 1815
```

```
End
```

```
Begin VB.Label lblExit
```

```
Caption = "Exit Program "
```

```
Height = 255
```

```
Left = 1680
```

```
TabIndex = 6
```

```
Top = 2640
```

```
Width = 1335
```

```
End
```

```
Begin VB.Image Image1
```

```
Height = 2535
```

```
Left = 4920
```

เอกสารนี้เป็นลิขสิทธิ์ของศูนย์บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Picture = (Bitmap)
Stretch = -1 'True
Top = 360
Width = 1215

```

```
End
```

```
Begin VB.Label lbCapbot
```

```

Caption = "Capture Screen button:"
Height = 255
Left = 1680
TabIndex = 5
Top = 720
Width = 2295

```

```
End
```

```
End
```

```
Begin ComctlLib.StatusBar StatusBar1
```

```

Align = 2 'Align Bottom
Height = 465
Left = 0
TabIndex = 2
Top = 3645
Width = 6405
_ExtentX = 11298
_ExtentY = 820
SimpleText = ""
_Version = 327682

```

```
BeginProperty Panels {0713E89E-850A-101B-AFC0-4210102A8DA7}
```

```
NumPanels = 1
```

```
BeginProperty Panel1 {0713E89F-850A-101B-AFC0-4210102A8DA7}
```

```

Alignment = 1
Object.Width = 2734
MinWidth = 2734
TextSave = ""
Key = ""
Object.Tag = ""

```

```
EndProperty
```

```
EndProperty
```

```
End
```

```
Begin VB.Label Label1
```

```
Height = 1215
```

```
Left = 360
```

```
TabIndex = 8
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Top      = 0
Width    = 855
End
Begin VB.Menu mnuFile
Caption  = "&File"
Begin VB.Menu mnuNew
Caption  = "&New"
End
Begin VB.Menu mnuOpen
Caption  = "&Open"
End
Begin VB.Menu mnuSplit0
Caption  = "-"
End
Begin VB.Menu mnuSave
Caption  = "&Save"
End
Begin VB.Menu mnuSaveAs
Caption  = "Save &As..."
End
Begin VB.Menu mnuSplit1
Caption  = "-"
End
Begin VB.Menu mnuExit
Caption  = "&Exit"
End
End
Begin VB.Menu mnuView
Caption  = "&View"
Begin VB.Menu mnuToolbar
Caption  = "&Toolbar"
Begin VB.Menu mnuLarge
Caption  = "&Large"
End
Begin VB.Menu mnuSmall
Caption  = "&Small"
End
Begin VB.Menu mnuNone
Caption  = "&None"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวไลยอลงกรณ์ ใช้ออกแบบและใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ก็ตาม อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End
```

```

Begin VB.Menu mnuSplit2
    Caption    = "-"
End
Begin VB.Menu mnuStatus
    Caption    = "&Status"
End
End
Begin VB.Menu mnuWindow
    Caption    = "&Windows"
Begin VB.Menu mnuSplit5
    Caption    = "-"
End
Begin VB.Menu winToggle
    Caption    = "&Toggle English-Thai"
End
End
Begin VB.Menu mnuOption
    Caption    = "&Option"
Begin VB.Menu mnuSplit6
    Caption    = "-"
End
Begin VB.Menu mnuCapture
    Caption    = "Ca&pture"
End
End
Begin VB.Menu mnuHelp
    Caption    = "&Help"
Begin VB.Menu mnuHelpTopic
    Caption    = "Help &Topic"
End
Begin VB.Menu mnuUsingHelp
    Caption    = "&Using Help"
End
Begin VB.Menu mnuSplit7
    Caption    = "-"
End
Begin VB.Menu mnuAboutT
    Caption    = "&About Training Project..."

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดก็ตาม หากมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End

' Visual Basic 4.0 16/32 Capture Routines

' This module contains several routines for capturing windows into a
' picture. All the routines work on both 16 and 32 bit Windows
' platforms.
' The routines also have palette support.

' CreateBitmapPicture - Creates a picture object from a bitmap and
' palette

' CaptureWindow - Captures any window given a window handle

' CaptureActiveWindow - Captures the active window on the desktop

' CaptureForm - Captures the entire form

' CaptureClient - Captures the client area of a form

' CaptureScreen - Captures the entire screen

' PrintPictureToFitPage - prints any picture as big as possible on
' the page

' NOTES

' - No error trapping is included in these routines

Option Explicit

Option Base 0

Private server_snd_flag As Boolean

Private Type PALETTEENTRY

peRed As Byte

peGreen As Byte

peBlue As Byte

peFlags As Byte

End Type

Private Type LOGPALETTE

palVersion As Integer

palNumEntries As Integer

palPalEntry(255) As PALETTEENTRY

Enough for 256 colors

End Type

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือเครื่องหมายการค้าเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หักข้อผิดพลาดให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Type GUID

Data1 As Long

Data2 As Integer

Data3 As Integer

Data4(7) As Byte

End Type

#If Win32 Then

Private Const RASTERCAPS As Long = 38

Private Const RC_PALETTE As Long = &H100

Private Const SIZEPALETTE As Long = 104

Private Type RECT

Left As Long

Top As Long

Right As Long

Bottom As Long

End Type

Private Declare Function CreateCompatibleDC Lib "GDI32" (_
ByVal hDC As Long) As Long

Private Declare Function CreateCompatibleBitmap Lib "GDI32" (_
ByVal hDC As Long, ByVal nWidth As Long, _
ByVal nHeight As Long) As Long

Private Declare Function GetDeviceCaps Lib "GDI32" (_
ByVal hDC As Long, ByVal iCapability As Long) As Long

Private Declare Function GetSystemPaletteEntries Lib "GDI32" (_
ByVal hDC As Long, ByVal wStartIndex As Long, _
ByVal wNumEntries As Long, lpPaletteEntries As PALETTEENTRY) _
As Long

Private Declare Function CreatePalette Lib "GDI32" (_
lpLogPalette As LOGPALETTE) As Long

Private Declare Function SelectObject Lib "GDI32" (_
ByVal hDC As Long, ByVal hObject As Long) As Long

Private Declare Function BitBlt Lib "GDI32" (_
ByVal hDCDest As Long, ByVal XDest As Long, _
ByVal YDest As Long, ByVal nWidth As Long, _
ByVal nHeight As Long, ByVal hDCSrc As Long, _
ByVal XSrc As Long, ByVal YSrc As Long, ByVal dwRop As Long) _
As Long

Private Declare Function DeleteDC Lib "GDI32" (_

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวไลยอลงกรณ์ อ่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ByVal hDC As Long) As Long
Private Declare Function GetForegroundWindow Lib "USER32" () _
    As Long
Private Declare Function SelectPalette Lib "GDI32" ( _
    ByVal hDC As Long, ByVal hPalette As Long, _
    ByVal bForceBackground As Long) As Long
Private Declare Function RealizePalette Lib "GDI32" ( _
    ByVal hDC As Long) As Long
Private Declare Function GetWindowDC Lib "USER32" ( _
    ByVal hWnd As Long) As Long
Private Declare Function GetDC Lib "USER32" ( _
    ByVal hWnd As Long) As Long
Private Declare Function GetWindowRect Lib "USER32" ( _
    ByVal hWnd As Long, lpRect As RECT) As Long
Private Declare Function ReleaseDC Lib "USER32" ( _
    ByVal hWnd As Long, ByVal hDC As Long) As Long
Private Declare Function GetDesktopWindow Lib "USER32" () As Long

Private Type PicBmp
    Size As Long
    Type As Long
    hBmp As Long
    hPal As Long
    Reserved As Long
End Type

Private Declare Function OleCreatePictureIndirect Lib "olepro32.dll" _
    (PicDesc As PicBmp, RefIID As GUID, ByVal fPictureOwnsHandle As Long, _
    IPic As IPicture) As Long

#Elseif Win16 Then
Private Const RASTERCAPS As Integer = 38
Private Const RC_PALETTE As Integer = &H100
Private Const SIZEPALETTE As Integer = 104

Private Type RECT
    Left As Integer
    Top As Integer
    Right As Integer
    Bottom As Integer
```

เอกสารนี้เป็นทรัพย์สินส่วนบุคคลของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Type

```

Private Declare Function CreateCompatibleDC Lib "GDI" ( _
    ByVal hDC As Integer) As Integer
Private Declare Function CreateCompatibleBitmap Lib "GDI" ( _
    ByVal hDC As Integer, ByVal nWidth As Integer, _
    ByVal nHeight As Integer) As Integer
Private Declare Function GetDeviceCaps Lib "GDI" ( _
    ByVal hDC As Integer, ByVal iCapability As Integer) As Integer
Private Declare Function GetSystemPaletteEntries Lib "GDI" ( _
    ByVal hDC As Integer, ByVal wStartIndex As Integer, _
    ByVal wNumEntries As Integer, _
    lpPaletteEntries As PALETTEENTRY) As Integer
Private Declare Function CreatePalette Lib "GDI" ( _
    lpLogPalette As LOGPALETTE) As Integer
Private Declare Function SelectObject Lib "GDI" ( _
    ByVal hDC As Integer, ByVal hObject As Integer) As Integer
Private Declare Function BitBlt Lib "GDI" ( _
    ByVal hDCDest As Integer, ByVal xDest As Integer, _
    ByVal yDest As Integer, ByVal nWidth As Integer, _
    ByVal nHeight As Integer, ByVal hDCSrc As Integer, _
    ByVal xSrc As Integer, ByVal ySrc As Integer, _
    ByVal dwRop As Long) As Integer
Private Declare Function DeleteDC Lib "GDI" ( _
    ByVal hDC As Integer) As Integer
Private Declare Function GetForegroundWindow Lib "USER" ( _
    Alias "GetActiveWindow" ()) As Integer
Private Declare Function SelectPalette Lib "USER" ( _
    ByVal hDC As Integer, ByVal hPalette As Integer, ByVal _
    bForceBackground As Integer) As Integer
Private Declare Function RealizePalette Lib "USER" ( _
    ByVal hDC As Integer) As Integer
Private Declare Function GetWindowDC Lib "USER" ( _
    ByVal hWnd As Integer) As Integer
Private Declare Function GetDC Lib "USER" ( _
    ByVal hWnd As Integer) As Integer
Private Declare Function GetWindowRect Lib "USER" ( _
    ByVal hWnd As Integer, lpRect As RECT) As Integer
Private Declare Function ReleaseDC Lib "USER" ( _
    ByVal hWnd As Integer, ByVal hDC As Integer) As Integer
Private Declare Function GetDesktopWindow Lib "USER" ( _
    ) As Integer
    
```

```
Private Type PicBmp
    Size As Integer
    Type As Integer
    hBmp As Integer
    hPal As Integer
    Reserved As Integer
End Type
```

```
Private Declare Function OleCreatePictureIndirect _
    Lib "oc25.dll" (PictDesc As PicBmp, RefIID As GUID, _
    ByVal fPictureOwnsHandle As Integer, IPic As IPicture) _
    As Integer
```

```
#End If
```

```
' CreateBitmapPicture
' - Creates a bitmap type Picture object from a bitmap and palette
```

```
' hBmp
' - Handle to a bitmap
'
' hPal
' - Handle to a Palette
' - Can be null if the bitmap doesn't use a palette
```

```
' Returns
' - Returns a Picture object containing the bitmap
```

```
#If Win32 Then
Public Function CreateBitmapPicture(ByVal hBmp As Long, _
    ByVal hPal As Long) As Picture
```

```
Dim r As Long
```

```
#ElseIf Win16 Then
```

```
Public Function CreateBitmapPicture(ByVal hBmp As Integer, _
    ByVal hPal As Integer) As Picture
```

```
Dim r As Integer
```

```
#End If
```

```

Dim Pic As PicBmp
' IPicture requires a reference to "Standard OLE Types"
Dim IPic As IPicture
Dim IID_IDispatch As GUID
' Fill in with IDispatch Interface ID
With IID_IDispatch
    .Data1 = &H20400
    .Data4(0) = &HC0
    .Data4(7) = &H46
End With
' Fill Pic with necessary parts
With Pic
    .Size = Len(Pic) ' Length of structure
    .Type = vbPicTypeBitmap ' Type of Picture (bitmap)
    .hBmp = hBmp ' Handle to bitmap
    .hPal = hPal ' Handle to palette (may be null)
End With ' Create Picture object

r = OleCreatePictureIndirect(Pic, IID_IDispatch, 1, IPic)
' Return the new Picture object
Set CreateBitmapPicture = IPic
End Function

.....

' CaptureWindow
' - Captures any portion of a window
'
' hWndSrc:
' - Handle to the window to be captured
'
' Client
' - If True CaptureWindow captures from the client area of the
' window
' - If False CaptureWindow captures from the entire window
'
' LeftSrc, TopSrc, WidthSrc, HeightSrc
' - Specify the portion of the window to capture

' Returns

```

เอกสารนี้เป็น **Dimensions need to be specified in pixels** ที่ออกการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

' Returns

' - Returns a Picture object containing a bitmap of the specified portion of the window that was captured

.....

#If Win32 Then

Public Function CaptureWindow(ByVal hWndSrc As Long, _
 ByVal Client As Boolean, ByVal LeftSrc As Long, _
 ByVal TopSrc As Long, ByVal WidthSrc As Long, _
 ByVal HeightSrc As Long) As Picture

Dim hDCMemory As Long

Dim hBmp As Long

Dim hBmpPrev As Long

Dim r As Long

Dim hDCSrc As Long

Dim hPal As Long

Dim hPalPrev As Long

Dim RasterCapsScrn As Long

Dim HasPaletteScrn As Long

Dim PaletteSizeScrn As Long

#ElseIf Win16 Then

Public Function CaptureWindow(ByVal hWndSrc As Integer, _
 ByVal Client As Boolean, ByVal LeftSrc As Integer, _
 ByVal TopSrc As Integer, ByVal WidthSrc As Long, _
 ByVal HeightSrc As Long) As Picture

Dim hDCMemory As Integer

Dim hBmp As Integer

Dim hBmpPrev As Integer

Dim r As Integer

Dim hDCSrc As Integer

Dim hPal As Integer

Dim hPalPrev As Integer

Dim RasterCapsScrn As Integer

Dim HasPaletteScrn As Integer

Dim PaletteSizeScrn As Integer

#End If

Dim LogPal As LOGPALETTE

เอกสารนี้เป็นลิขสิทธิ์ของ บริษัท อีเอส เอช จำกัด ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

hDCSrc = GetDC(hWndSrc) ' Get device context for client area

```

Else
    hDCSrc = GetWindowDC(hWndSrc) ' Get device context for entire
' window
End If
' Create a memory device context for the copy process
hDCMemory = CreateCompatibleDC(hDCSrc)
' Create a bitmap and place it in the memory DC
hBmp = CreateCompatibleBitmap(hDCSrc, WidthSrc, HeightSrc)
hBmpPrev = SelectObject(hDCMemory, hBmp)
' Get screen properties
RasterCapsScrn = GetDeviceCaps(hDCSrc, RASTERCAPS) ' Raster
'capabilities
HasPaletteScrn = RasterCapsScrn And RC_PALETTE ' Palette
'support
PaletteSizeScrn = GetDeviceCaps(hDCSrc, SIZEPALETTE) ' Size of
' palette
' If the screen has a palette make a copy and realize it
If HasPaletteScrn And (PaletteSizeScrn = 256) Then
' Create a copy of the system palette
LogPal.palVersion = &H300
LogPal.palNumEntries = 256
r = GetSystemPaletteEntries(hDCSrc, 0, 256, _
LogPal.palPalEntry(0))
hPal = CreatePalette(LogPal)
' Select the new palette into the memory DC and realize it
hPalPrev = SelectPalette(hDCMemory, hPal, 0)
r = RealizePalette(hDCMemory)
End If
' Copy the on-screen image into the memory DC
r = BitBlt(hDCMemory, 0, 0, WidthSrc, HeightSrc, hDCSrc, _
LeftSrc, TopSrc, vbSrcCopy)
' Remove the new copy of the on-screen image
hBmp = SelectObject(hDCMemory, hBmpPrev)
' If the screen has a palette get back the palette that was
' selected in previously
If HasPaletteScrn And (PaletteSizeScrn = 256) Then
hPal = SelectPalette(hDCMemory, hPalPrev, 0)
End If
' Release the device context resources back to the system
r = DeleteDC(hDCMemory)
r = ReleaseDC(hWndSrc, hDCSrc)

```

```
' Call CreateBitmapPicture to create a picture object from the
' bitmap and palette handles. Then return the resulting picture
' object.
```

```
Set CaptureWindow = CreateBitmapPicture(hBmp, hPal)
```

```
End Function
```

```
.....
```

```
,
```

```
' CaptureScreen
```

```
' - Captures the entire screen
```

```
,
```

```
' Returns
```

```
' - Returns a Picture object containing a bitmap of the screen
```

```
.....
```

```
,
```

```
Public Function CaptureScreen() As Picture
```

```
#If Win32 Then
```

```
Dim hWndScreen As Long
```

```
#Elseif Win16 Then
```

```
Dim hWndScreen As Integer
```

```
#End If
```

```
' Get a handle to the desktop window
```

```
hWndScreen = GetDesktopWindow()
```

```
' Call CaptureWindow to capture the entire desktop give the handle
```

```
' and return the resulting Picture object
```

```
Set CaptureScreen = CaptureWindow(hWndScreen, False, 0, 0, _
(Screen.Width \ Screen.TwipsPerPixelX), _
(Screen.Height \ Screen.TwipsPerPixelY))
```

```
End Function
```

```
Private Sub Command1_Click() 'Capture Button
```

```
Form1.WindowState = 1 ' 1 = Minimize,2=Maximize,0=Normal Form to Task bar
```

```
Const SRC_COPY = &HCC0020
```

```
Dim Dhwnd&, Dhdc&, r As RECT
```

```
Dim hdcMem As Long
```

```
FullScreen.Hide
```

```
DoEvents
```

```
Dhwnd = GetDesktopWindow
```

```
Dhdc = GetDC(Dhwnd)
```

```
Call GetWindowRect(Dhwnd, r)
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Picture2.Left = 0
Picture2.Top = 0
Picture2.Width = r.Right - r.Left
Picture2.Height = r.Bottom - r.Top
Call BitBlt(Picture2.hDC, 0, 0, Picture2.Width, _
Picture2.Height, Dhdc, 0, 0, SRC_COPY)
Picture2.Picture = CaptureScreen()
Call ReleaseDC(Dhwnd, Dhdc)
FullScreen.Show
Picture1.Refresh
Picture2.Refresh
VScroll1.Max = Picture2.Height
VScroll1.Min = Picture1.ScaleHeight
VScroll1.Value = VScroll1.Min
HScroll1.Max = Picture2.Width
HScroll1.Min = Picture1.ScaleWidth
HScroll1.Value = HScroll1.Min
SavePicture Picture2.Picture, Form1.sharename.Text & ".xx.bmp"
End Sub

Private Sub Command2_Click()
'Exit Button
Unload FullScreen 'Close this form
If Form1.srvopt.Value = 1 Then
Form1.Winsock1(0).Close
Else
Form1.Winsock2.Close
End If
Form1.Show 'Back to Form1 Show
End Sub

Private Sub Command3_Click()
If server_snd_flag = False Then
FullScreen.Timer1.Interval = 3000
FullScreen.Timer1.Enabled = True
FullScreen.Command3.Caption = "&Stop"
server_snd_flag = True
StatusBar1.Panels(1) = "Sending ....."
Else
FullScreen.Timer1.Enabled = False
FullScreen.Command3.Caption = "&Start"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นที่มามีเหตุข้อยกเว้นและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

server_snd_flag = False
StatusBar1.Panels(1) = "Waiting ....."
End If
End Sub

Private Sub Form_Load() 'FullScreen Form
Width = Screen.Width * 1 ' Set form width.
Height = Screen.Height * 1 ' Set form height.
Left = (Screen.Width - Width) / 2 ' Center form horizontally.
Top = (Screen.Height - Height) / 2 ' Center form vertically.
FullScreen.ScaleMode = 3 'Scale Mode 3=Pixel (smallest unit of monitor or printer resolution).
Picture1.ScaleMode = 3
Picture2.ScaleMode = 3
Picture1.AutoRedraw = True
Picture2.AutoRedraw = True
Picture2.Visible = True
VScroll1.LargeChange = Picture2.Height
HScroll1.LargeChange = Picture2.Width
If Form1.srvopt.Value = 1 Then ' 1 = server otherwise client
Form1.Winsock1(0).LocalPort = 6666
Form1.Winsock1(0).RemoteHost = Form1.tcp_addr.Text
Form1.Winsock1(0).Listen
FullScreen.tx.Caption = "&Send"
FullScreen.Command3.Caption = "&Start"
Else
Form1.Winsock2.RemotePort = 6666
Form1.Winsock2.RemoteHost = Form1.tcp_addr.Text
Form1.Winsock2.Connect
FullScreen.tx.Caption = "&Receive"
FullScreen.Command3.Visible = False
StatusBar1.Panels(1) = "Receiving ....."
End If
End Sub

Private Sub HScroll1_Change()
Picture2.Left = HScroll1.Min - HScroll1.Value
Picture2.SetFocus
End Sub

Private Sub HScroll1_Scroll()
HScroll1_Change

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ หากมีการนำเอกสารนี้ไปดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HScroll1_Change

FullScreen - 12

End Sub

Private Sub mnuExit_Click()

Unload Me

End Sub

Private Sub Picture1_Click() 'Capture Screen Area

Width = 12075: Height = 8955 'Max size screen 800X600

'Width = Screen.Width * 1

'Height = Screen.Height * 1

End Sub

Private Sub Picture2_Click() 'Background Red Color

'Width = Screen.Width * 1

'Height = Screen.Height * 1

End Sub

Private Sub srvopt_Click()

If Form1.srvopt.Value = 0 Then

FullScreen.tx.Caption = "&Send"

Else

FullScreen.tx.Caption = "&Receive"

FullScreen.tx.Enabled = False

End If

End Sub

Private Sub Timer1_Timer()

Dim i As Integer

Dim fn As String

Form1.WindowState = 1 ' 1 = Minimize, 2 = Maximize, 0 = Normal Form to Task bar

Const SRC_COPY = &HCC0020

Dim Dhwnd&, Dhdc&, r As RECT

Dim hdcMem As Long

'FullScreen.Hide

DoEvents

Dhwnd = GetDesktopWindow

Dhdc = GetDC(Dhwnd)

Call GetWindowRect(Dhwnd, r)

Picture2.Left = 0

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Picture2.Top = 0
Picture2.Width = r.Right - r.Left
Picture2.Height = r.Bottom - r.Top
Call BitBlt(Picture2.hDC, 0, 0, Picture2.Width, _
Picture2.Height, Dhdc, 0, 0, SRC_COPY)
Picture2.Picture = CaptureScreen()
Call ReleaseDC(Dhwnd, Dhdc)
FullScreen.Show
Picture1.Refresh
Picture2.Refresh
VScroll1.Max = Picture2.Height
VScroll1.Min = Picture1.ScaleHeight
VScroll1.Value = VScroll1.Min
HScroll1.Max = Picture2.Width
HScroll1.Min = Picture1.ScaleWidth
HScroll1.Value = HScroll1.Min
fn = Form1.sharename.Text & Trim(Str(robin))
SavePicture Picture2.Picture, fn
If robin > 10 Then
    robin = 1
Else
    robin = robin + 1
End If

For i = 1 To 50
    If up_link(i) <> -1 Then
        Form1.Winsock1(up_link(i)).SendData (fn)
    End If
Next
End Sub

Private Sub tx_Click()
Dim buf As String

If Form1.srvopt.Value = 0 Then
    Form1.Winsock2.SendData (Form1.sharename.Text)
End If
End Sub

Private Sub VScroll1_Change()
Picture2.Top = VScroll1.Min - VScroll1.Value

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งผู้จัดทำขอสงวนสิทธิ์ในเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Picture2.SetFocus

End Sub

Private Sub VScroll1_Scroll()

VScroll1_Change

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERSION 5.00

Object = "{6B7E6392-850A-101B-AFC0-4210102A8DA7}#1.3#0": "COMCTL32.OCX"

Begin VB.Form FullScreen

BackColor = &H00808080&

BorderStyle = 1 'Fixed Single

Caption = "Full Screen"

ClientHeight = 8340

ClientLeft = 255

ClientTop = 1740

ClientWidth = 8235

LinkTopic = "Form1"

PaletteMode = 1 'UseZOrder

ScaleHeight = 8340

ScaleWidth = 8235

Begin VB.CommandButton Command3

Caption = "&Start"

Default = -1 'True

Height = 375

Left = 1680

TabIndex = 8

Top = 7920

Width = 1215

End

Begin VB.Timer Timer1

Enabled = 0 'False

Left = 5760

Top = 8400

End

Begin VB.CommandButton tx

Caption = "&Send"

Enabled = 0 'False

Height = 375

Left = 5520

TabIndex = 7

Top = 7920

Width = 1215

End

Begin VB.CommandButton Command2

Cancel = 1 'True

Caption = "&Exit"

Height = 375

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Left      = 3000
TabIndex  = 5
Top       = 7920
Width     = 1095
End
Begin VB.CommandButton Command1
Caption   = "&Capture"
Enabled   = 0 'False
Height    = 375
Left      = 4200
TabIndex  = 4
Top       = 7920
Width     = 1215
End
Begin VB.HScrollBar HScroll1
Height    = 255
Left      = 0
TabIndex  = 3
Top       = 7680
Width     = 12015
End
Begin VB.VScrollBar VScroll1
Height    = 7695
Left      = 11760
TabIndex  = 2
Top       = 0
Width     = 255
End
Begin VB.PictureBox Picture1
AutoRedraw = -1 'True
AutoSize   = -1 'True
BorderStyle = 0 'None
Height     = 7935
Left       = 0
ScaleHeight = 529
ScaleMode  = 3 'Pixel
ScaleWidth  = 801
TabIndex   = 0
Top        = 0
Width      = 12015
End

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Begin VB.PictureBox Picture2

```

AutoRedraw = -1 'True
AutoSize = -1 'True
BackColor = &H00000080&
BorderStyle = 0 'None
Height = 7935
Left = 0
ScaleHeight = 529
ScaleMode = 3 'Pixel
ScaleWidth = 801
TabIndex = 1
Top = 0
Width = 12015
End
End
Begin ComctlLib.StatusBar StatusBar1
Align = 2 'Align Bottom
Height = 510
Left = 0
TabIndex = 6
Top = 7830
Width = 8235
_ExtentX = 14526
_ExtentY = 900
SimpleText = ""
_Version = 327682
BeginProperty Panels {0713E89E-850A-101B-AFC0-4210102A8DA7}
NumPanels = 1
BeginProperty Panel1 {0713E89F-850A-101B-AFC0-4210102A8DA7}
Key = ""
Object.Tag = ""
EndProperty
EndProperty
End
Begin VB.Menu mnuFile
Caption = "&File"
Begin VB.Menu mnuExit
Caption = "&Exit"
End
End

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IPcfig - 1

```
Private Sub Command1_Click()
```

```
End Sub
```

```
Private Sub cmdClose_Click()
```

```
Unload Me 'Unload this form
```

```
End Sub
```

```
Private Sub mnuExit_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub cmdConn_Click()
```

```
End Sub
```

```
Private Sub cmdDisc_Click()
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
'Set Fix IPcfig Form size
```

```
Height = 2990: Width = 3990
```

```
'Fix Position of IPCFG Form
```

```
IPcfig.Left = (Screen.Width - Width) / 1 / 2
```

```
IPcfig.Top = (Screen.Height - Height) / 1 / 2
```

```
IPcfig.DrawMode = 4
```

```
End Sub
```

```
Private Sub mnuClose_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub srvopt_Click()
```

```
If srvopt.Value = True Then
```

```
FullScreen.tx.Caption = "&Receive"
```

```
End If
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub Text1_Change()

Text1.Text = Text1.Text

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERSION 5.00

Object = "{6B7E6392-850A-101B-AFC0-4210102A8DA7}#1.3#0"; "COMCTL32.OCX"

Begin VB.Form IPcfg

Caption = "IP Configuration"

ClientHeight = 2265

ClientLeft = 2505

ClientTop = 1815

ClientWidth = 3855

LinkTopic = "Form1"

MaxButton = 0 'False

PaletteMode = 1 'UseZOrder

ScaleHeight = 2265

ScaleWidth = 3855

Begin ComctlLib.Toolbar Toolbar1

Align = 1 'Align Top

Height = 420

Left = 0

TabIndex = 6

Top = 0

Width = 3855

_ExtentX = 6800

_ExtentY = 741

Appearance = 1

_Version = 327682

End

Begin VB.CommandButton cmdClose

Caption = "Cl&ose"

Height = 375

Left = 2640

TabIndex = 3

Top = 1680

Width = 1095

End

Begin VB.CommandButton cmdDiscon

Caption = "&Disconnect"

Height = 375

Left = 1440

TabIndex = 2

Top = 1680

Width = 1095

End

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Begin VB.CommandButton cmdConnect
```

```
    Caption   = "&Connect"
```

```
    Height    = 375
```

```
    Left      = 120
```

```
    TabIndex  = 1
```

```
    Top       = 1680
```

```
    Width     = 1215
```

```
End
```

```
Begin VB.TextBox top_addr
```

```
    Height    = 375
```

```
    Left      = 1200
```

```
    TabIndex  = 0
```

```
    Top       = 840
```

```
    Width     = 2415
```

```
End
```

```
Begin VB.Frame Frame1
```

```
    Caption   = "Specify an IP Address:"
```

```
    Height    = 1695
```

```
    Left      = 0
```

```
    TabIndex  = 4
```

```
    Top       = 480
```

```
    Width     = 3855
```

```
Begin VB.Label lblIP
```

```
    Caption   = "IP Address:"
```

```
    Height    = 375
```

```
    Left      = 240
```

```
    TabIndex  = 5
```

```
    Top       = 360
```

```
    Width     = 855
```

```
End
```

```
End
```

```
Begin VB.Menu mnuOpen
```

```
    Caption   = "&Open"
```

```
Begin VB.Menu mnuConnect
```

```
    Caption   = "&Connect"
```

```
End
```

```
Begin VB.Menu mnuDiscon
```

```
    Caption   = "&Disconnect"
```

```
End
```

```
Begin VB.Menu mnuClose
```

```
    Caption   = "&Close"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ที่ลิขสิทธิ์สงวนไว้ทั้งหมดนี้ขอสงวนเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End

End

End



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำโครงการปริญญานิพนธ์นี้ สำเร็จได้ด้วยดีเกิดจากการให้คำแนะนำและการให้ข้อมูลของท่านอาจารย์ที่ปรึกษาโครงการและคณะกรรมการทุกท่านที่ได้อบรมสั่งสอนทุกท่านรวมทั้งได้สนับสนุนเครื่องมือและตำรา,เทคนิคต่างๆรวมถึง Web Site ต่างๆที่ได้ให้ข้อมูลคำแนะนำและวิธีการที่นำมาประกอบในการพัฒนาโปรแกรมที่ใช้ในการทำปริญญานิพนธ์นี้ รวมทั้งครอบครัวที่ให้การสนับสนุนในการศึกษาที่ผ่านมาจนกระทั่งถึงปัจจุบันนี้ จึงขอกราบขอบพระคุณทุกๆท่านอย่างสูง.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. การเขียนโปรแกรมบน Microsoft Visual Basic 3.0 โดย วรวิทย์ ตันติโกทินและนภดล ชาญธีระเดช
2. Microsoft Visual Basic 4.0 Learning CD Microsoft Press CDROM Media
3. Learning VB4.0 Microsoft Press Book by Michael Halvorson
4. MSDN Network Programming Microsoft MSDN CDROM
5. Computer Today Magazine โดย ชาตพล นภาวารีและปณัสย์ อลงกรณ์วรกิจ
6. ไมโครคอมพิวเตอร์ Magazine Learn VB5.0 โดย สุทธิศักดิ์ พงศ์ธนาพานิช
7. Microsoft Visual Basic 6.0 Learning CD CIS Computer Infromatiom

Reference WEB Site

http://www.microsoft.com	Microsoft Related Product
http://www.microsoft.com/visualbasic	Microsoft Visual Basic Information Page
http://support.microsoft.com/support/kb/articles	Microsoft Knowledge Base
http://www.sockets.com	Windows Sockets Information
http://msdn.microsoft.com/developer/default.htm	Visual Basic Developer Information
http://www.whatis.com	TCP/IP Protocol Information
http://www.vbtt.com	Visual Basic Tech & Tip
http://come.to/vbstreet	Visual Basic Learning information
http://www.vb-helper.com	Visual Basic developer information
http://www.developer.com	Visual Basic developer information
http://www.codeguru.com	Visual Basic developer information

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้