สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบการประชุมทางไกลด้วยภาพและเสียงผ่านอินเตอร์เน็ต

(Internet Teleconference with Image and Sound)

T042700

โดย

นายนพปฎล ประเสริฐ

นายวราสิน ม่วงเกตุ

เลขหมู่.....
เลขทะเบียน 42700
วัน, เดือน, ปี 6 มิ.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11916426

ระบบการประชุมทางไกลด้วยภาพและเสียงผ่านอินเตอร์เน็ต

(Internet Teleconference with Image and Sound)

โดย

นายนพปฎล ประเสริฐ เลขประจำตัว 40010359

นายวราสิน ม่วงเกตุ เลขประจำตัว 40010694

อาจารย์ที่ปรึกษา

รศ.ดร.รัตติกร วรากูลศิริพันธุ์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการประชุมทางไกลด้วยภาพและเสียงผ่านอินเตอร์เน็ต

(Internet Teleconference with Image and Sound)

โดย

นายนพปฎล	ประเสริฐ	เลขประจำตัว 40010359
นายวราสิน	ม่วงเกตุ	เลขประจำตัว 40010694

ปริญญานิพนธ์ฉบับนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะสอบ

.....

(รศ.ดร.รัตติกร วรากูลศิริพันธุ์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการประชุมทางไกลด้วยภาพและเสียงผ่านทางอินเตอร์เน็ต

ผู้จัดทำ

1. นายนพปฎล ประเสริฐ
2. นายวราสิน ม่วงเกตุ

.....อาจารย์ที่ปรึกษา

(รศ.ดร.รัตติกร วรากูลศิริพันธุ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการประชุมทางไกลด้วยภาพและเสียงผ่านอินเตอร์เน็ต

นายนพปฎล ประเสริฐ
นายวราสิน ม่วงเกตุ
รศ.ดร.รัตติกร วรากูลศิริพันธุ์
อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้กล่าวถึงการประชุมผ่านอินเตอร์เน็ต ว่ามีรูปแบบการทำงานเป็นอย่างไร โดยใช้โปรแกรม Delphi เวอร์ชัน 5.00 เป็นเครื่องมือในการสร้างซอฟต์แวร์นี้ ใช้การติดต่อระหว่างแต่ละเครื่องผ่าน IP Address หรือชื่อของเครื่อง ในโปรแกรมนี้สามารถส่งสัญญาณเสียงและภาพผ่านอินเตอร์เน็ตให้คู่สนทนาได้ยินแม้จะอยู่อีกสถานที่หนึ่ง โดยใช้โปรแกรม DirectX ในส่วนของ DirectSound ในการจับข้อมูลภาพและส่งไปพร้อมกับก้อนข้อมูลภาพที่ได้เตรียมไว้แล้ว และตรวจสอบผลการทำงานระหว่างการผสมข้อมูลภาพและเสียงลงไปในก้อนข้อมูลเดียวกันแล้วส่งไปในพอร์ตเดียว กับการแบ่งข้อมูลภาพและเสียงออกจากกันแล้วส่งข้อมูลไปพร้อมกัน โดยใช้พอร์ตที่ส่งข้อมูลต่างกัน สิ่งที่ใช้วัดในปริญญานิพนธ์นี้คือความสามารถที่จะส่งข้อมูลไปพร้อมๆ กันโดยมีคุณภาพของการแสดงผลเป็นตัวเสริม นอกจากนี้ เราได้พยายามที่จะพัฒนาซอฟต์แวร์นี้ให้ง่ายต่อการใช้งานของบุคคลทั่วไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Internet Teleconference with Image and Sound

Mr. Noppadon Prasert

Mr. Varasin Muangkate

ABSTRACT

This project refers to how can we done with a teleconference via Internet. We use "Delphi 5.00" which implement this software and access to any computer by IP address or host name. In this program, we can send speech signal and image to terminal computers in other place in LAN network. We use DirectX technique in sound capturing sent with asynchronous block of images data and test the result compared between mix sound and image data/port and sound and image data with separate port. The quality of result is how much we can synchronize these data , not the quality of display output. Otherwise, we try to program this software in the way that everyone can use this software easily.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เสร็จไปได้ด้วยดี ด้วยคำแนะนำและการให้แนวทางการทำงานต่างๆ จากอาจารย์รัตติกร วรากูลศิริพันธุ์ ซึ่งเป็นที่ปรึกษาในการทำงาน แม้งานของท่านจะมีอยู่มากมายก็ตาม ซึ่งผู้จัดทำรู้สึกทราบซึ้งในความอนุเคราะห์จากท่านและกราบขอบพระคุณเป็นอย่างยิ่ง

นอกจากนี้ยังมีพี่ๆปริญญาโทในห้อง Communication Network Lab บนชั้น 10 ส่วน ReCCIT ที่คอยให้การสนับสนุน และให้คำปรึกษาในการทำงาน ทั้งพี่ที่จบการศึกษาไปแล้วยังช่วยสอนการเขียนโปรแกรมละชี้แนวคิดการออกแบบโปรแกรม และที่มอบแนวความคิดของปริญญานิพนธ์นี้ ทำให้งานที่ออกมาสำเร็จลงได้ด้วยดี

สุดท้ายนี้ขอขอบคุณภาควิชาอิเล็กทรอนิกส์ที่ได้ให้การสนับสนุนในการทำปริญญานิพนธ์ครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากรายงานฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

นายนพปฎล ประเสริฐ

นายวราสิน ม่วงเกตุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 วิธีการดำเนินงาน	2
บทที่ 2 การประชุมทางไกล (Teleconference)	3
2.1 รูปแบบการประชุมทางไกล	3
2.2 รู้จักกับการทำงานของการประชุมทางไกลด้วยกล้องวิดีโอ	3
2.2.1 การใช้งานการประชุมทางไกลด้วยกล้องวิดีโอ	3
2.2.2 หัวใจสำคัญของการประชุมทางไกลด้วยกล้องวิดีโอ อยู่ที่โคเด็ก (Codec)	4
2.2.3 โครงสร้างระบบการประชุมทางไกลด้วยกล้องวิดีโอ	5
2.2.4 อนาคตของการประชุมทางไกลด้วยกล้องวิดีโอ	6
2.3 โทรศัพท์ผ่านเครือข่ายอินเตอร์เน็ต (Internet Telephony) และการประชุมทางไกล	7
2.3.1 มาตรฐาน H.323	8
2.3.2 มาตรฐาน T.120	10
2.4 ปัญหาและพัฒนาการ	12
บทที่ 3 ไดเร็กเอ็กซ์ (DirectX)	13
3.1 ไดเร็กเอ็กซ์คืออะไร	13
3.2 จุดประสงค์ของไดเร็กเอ็กซ์	13
3.3 ประโยชน์ของไดเร็กเอ็กซ์	13
3.4 ส่วนประกอบของไดเร็กเอ็กซ์	13
3.5 DirectSound	15
3.5.1 ประเภทของเสียงบนคอมพิวเตอร์	15
3.5.2 รู้จักกับ DirectSound	15
3.5.3 การทำงานของ DirectSound	17
3.5.4 การติดต่อของ DirectSound	18
3.5.5 รูปแบบเสียงใน DirectSound (โครงสร้าง WAVEFORMATEX)	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
3.5.6 การเข้าถึง DirectSound	19
3.5.6.1 การกำหนดอุปกรณ์เสียง (Enumeration)	19
3.5.6.2 การสร้างออบเจ็กต์ DirectSound	20
3.5.6.3 การกำหนดระดับ Cooperative	21
3.5.6.4 การกำหนดรูปแบบบัฟเฟอร์ primary	22
บทที่ 4 ซอกเก็ท (Socket)	23
4.1 การสื่อสารโดยผ่านซอกเก็ท (Socket)	23
4.2 Winsock	24
4.3 The Winsock API	25
4.4 Winsock Initialization	26
บทที่ 5 โปรแกรมการประชุมทางไกลด้วยกล้องวิดีโอหรือ VMeeting	27
5.1 หลักการทำงานของโปรแกรมการประชุมทางไกลด้วยกล้องวิดีโอ หรือ VMeeting	27
5.2 การเขียนโปรแกรม VMeeting	29
5.3 หลักการในการทำงานในส่วนแสดงผลด้วยภาพ	31
บทที่ 6 การใช้งานโปรแกรม VMeeting	33
บทที่ 7 ผลการทดลองและสรุปผล	37
7.1 ผลการทดลอง	37
7.2 สรุปผลการทดลอง	39
7.3 ข้อจำกัดของระบบ	39
7.4 สมรรถภาพของระบบ	39
7.5 ความผิดพลาดที่เกิดขึ้นในการทำงาน	40
7.6 สิ่งที่ได้รับจากการทำโครงงาน	40
7.7 แนวทางในการพัฒนางานเพิ่มเติม	40
บรรณานุกรม	41
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่		หน้า
รูป 2.1	รูปแบบของการประชุมทางไกลด้วยกล้องวิดีโอ	5
รูป 2.2	ส่วนประกอบของระบบการประชุมทางไกลด้วยกล้องวิดีโอ	6
รูป 2.3	แสดงการติดต่อระหว่างผู้ใช้กับเซิร์ฟเวอร์ในการประชุมทางไกลซึ่งผู้ใช้แต่ละคนจะสามารถใช้ได้ทั้งเสียง ภาพ วิดีโอ ไวท์บอร์ด และการใช้โปรแกรมร่วมกันในการสื่อสารกับคนอื่นๆ	7
รูป 2.4	แสดงขั้นตอนการติดต่อระหว่างผู้ใช้กับเซิร์ฟเวอร์ ในการทำงานตามโพรโตคอลมาตรฐาน H.323	10
รูป 3.1	ส่วนประกอบของไดเร็กเอ็กซ์	14
รูป 3.2	สถาปัตยกรรมของ DirectSound	16
รูป 3.3	การทำงานของ DirectSound	17
รูป 3.4	การติดต่อของ DirectSound	18
รูป 4.1	แสดง Socket connection for TCP	23
รูป 4.2	แสดง Socket connection for UDP	24
รูป 5.1	เซิร์ฟเวอร์ซอกเก็ท (Server Socket)	27
รูป 5.2	ไคลเอนต์ซอกเก็ท (Client Socket)	28
รูป 5.3	แสดง Transmission Protocol	29
รูป 6.1	แสดงรูปโปรแกรม	33
รูป 6.2	แสดงหน้าจอเมื่อเซิร์ฟเวอร์ได้รับการติดต่อ	34
รูป 6.3	ส่วนการปรับแต่งค่า Sound	34
รูป 6.4	ส่วนการปรับแต่งค่า Video	35
รูป 6.5	แสดงหน้าจอ Chat board	36
รูป 7.1	หน้าจอที่เกิดจากการติดต่อแล้ว	37
รูป 7.2	การทดลองในแบบต่างๆ	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ตาราง 2.1 มาตรฐาน H.323	9
ตาราง 2.2 มาตรฐาน T.120	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในวงการธุรกิจทุกวันนี้ การประชุมเป็นสิ่งที่จำเป็นในทุกองค์กรและทุกตำแหน่ง ค่าใช้จ่ายในการเดินทางและเวลาที่เสียไปในการประชุมนอกสถานที่ มีค่ามหาศาล ยิ่งผู้ที่มีหน้าที่ในการบริหารจะใช้เวลากว่า 50 เปอร์เซ็นต์ ไปกับการประชุม สิ่งหนึ่งที่สามารถทดแทนได้อย่างมีประสิทธิภาพนั่นคือ การประชุมที่ใช้การสื่อสารทางไกล หรือ Teleconference นั่นเอง

ระบบการประชุมทางไกล (Teleconference) ในปัจจุบันมีตั้งแต่การใช้โทรศัพท์ทางไกลธรรมดา จนถึงการแสดงออกทางภาพและเสียง ประกอบการใช้คอมพิวเตอร์ไปพร้อมๆ กัน ทำให้การประชุมเกิดตรงเวลา ประหยัดกว่า และสามารถดำเนินตามแผนงานธุรกิจที่วางไว้ได้อย่างไม่คลาดเคลื่อน นอกจากนี้ยังอาจเป็นตัวเสริมด้านข้อมูลก่อนหรือหลังการประชุมโดยตรงก็ได้

ข้อดีต่างๆ ของระบบการประชุมทางไกล มีดังนี้

- เพิ่มประสิทธิภาพการทำงาน
- ลดค่าใช้จ่ายที่ต้องเสียกับการประชุมนอกสถานที่ได้
- สามารถเข้าร่วมการประชุมหลายๆ ครั้งในหลายๆ ที่ได้ในวันเดียว
- แก้ปัญหาต่างๆ ได้รวดเร็วยิ่งขึ้น
- ใช้เวลาในการโต้ตอบสั้น

การประชุมทางไกล (Teleconference) เป็นทางเลือกหนึ่งที่เหมาะสมและเป็นทางเลือกใหม่นอกจากการประชุมโดยตรงและได้มีการนำไปใช้ประโยชน์ด้านอื่นๆด้วยเช่น การเรียนการสอนทางไกล (Teleteaching) และการบริการทางไกล (Teleservicing) เช่น การติดตั้งอุปกรณ์จากต่างประเทศที่ต้องให้เจ้าของมาติดตั้งให้ สามารถบอกผ่านทาง Teleservicing ได้ เป็นต้น

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาการส่งผ่านข้อมูลภาพและเสียงทางอินเตอร์เน็ต
2. ทำการปรับปรุงการส่งให้ภาพและเสียงไปพร้อมกัน
3. ทำการปรับปรุงโปรแกรมการทำงานให้มีลักษณะง่ายต่อการใช้งาน

1.3 ขอบเขตของโครงงาน

โปรแกรมสำหรับใช้ในการประชุมทางไกลผ่านอินเตอร์เน็ตนี้ ได้พัฒนาขึ้นโดยใช้ Delphi เวอร์ชัน 5 ในการควบคุมฟังก์ชันการทำงานต่างๆของโปรแกรม ได้แก่ การ login เข้าสู่ระบบ การเรียกไปยังคู่สนทนา การส่งภาพและเสียง และการแสดงผลด้วยอุปกรณ์ทางฮาร์ดแวร์ ซึ่งประเด็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักของโครงงานนี้อยู่ที่การส่งภาพและเสียง ซึ่งในการเขียนโปรแกรมในส่วนการส่งภาพนั้นจะใช้ MMTools ส่วนการส่งเสียงนั้นจะใช้ DirectSound เป็นตัวพัฒนา

ปัจจัยหลักในการเลือกเครื่องมือหลักในการเขียนโปรแกรม คือ ความเข้าใจในการทำงานและความสามารถในการทำงานร่วมกับ DirectX ได้ ซึ่ง Delphi เป็นเครื่องมือหนึ่งที่ง่ายต่อการทำความเข้าใจ และความสามารถในการพัฒนาโปรแกรมของ Delphi สามารถใช้งานร่วมกับ DirectX ได้ดี

1.4 วิธีการดำเนินงาน

โครงงานนี้เริ่มต้นจากการศึกษาทฤษฎีพื้นฐานต่างๆเกี่ยวกับวิธีการและโพรโตคอลต่างๆที่ใช้ในการติดต่อสื่อสารข้อมูลผ่านระบบเครือข่าย ต่อมาจึงได้ศึกษาทฤษฎีต่างๆที่ใช้ในการเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์ และศึกษาชุดคำสั่งต่างๆใน Delphi แล้วจึงได้ออกแบบการทำงานและส่วนอินเตอร์เฟส (Interface) ของโปรแกรม

หลังจากนั้นจึงเริ่มเขียนโปรแกรมโดยใช้คอมโพเนนท์ต่างๆ ของ Delphi และ DirectX ตามรูปแบบของโปรแกรมที่ออกแบบไว้ แล้วจึงทำการทดลองติดต่อกันระหว่างคอมพิวเตอร์ 2 เครื่อง โดยทดลองการทำงานของส่วนต่างๆ ได้แก่ การเข้าสู่ระบบ, การติดต่อระหว่างเครื่อง, การส่งข้อความ, การส่งภาพ และการส่งเสียง แล้วพัฒนาปรับปรุงคุณภาพให้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การประชุมทางไกล (Teleconference)

2.1 รูปแบบการประชุมทางไกล

ระบบการประชุมทางไกล (Teleconference) มีอยู่ด้วยกันหลายรูปแบบ ดังจะกล่าวต่อไปนี้

1. การประชุมทางไกลด้วยเสียง (Audio Teleconference) เป็นวิธีที่ง่ายที่สุด เสียค่าใช้จ่ายน้อยที่สุด และมีการใช้กันอย่างแพร่หลายที่สุดในทุกรูปแบบ วิธีการนี้อาจใช้ประชุมผ่านเสียงเพียงอย่างเดียว หรือจะมีอุปกรณ์แสดงภาพอย่างหยาบ เช่น โทรทัศน์ที่มีอัตราการแสดงผลต่ำ เป็นต้น ประกอบอยู่ด้วยก็ได้ วิธีนี้เป็นการประชุมในเวลาจริง (Real time)

2. การประชุมทางไกลด้วยกล้องวิดีโอ (Video Teleconference) จะมีการใช้ทั้งอุปกรณ์ด้านเสียงและด้านภาพไปพร้อมๆ กัน วิธีนี้จะเห็นการเคลื่อนไหวและได้ยินเสียงทั้งหมดของผู้เข้าร่วมประชุมโดยตลอด เป็นแบบเวลาจริง เช่นกัน และเป็นแบบที่สมบูรณ์ที่สุดด้วย

3. การประชุมทางไกลด้วยคอมพิวเตอร์ (Computer Teleconference) ในสมัยก่อนจะมีลักษณะเป็นการส่งข้อความเข้าหากันมากกว่า เพราะเทคโนโลยีด้านอินเตอร์เน็ตยังไม่พัฒนาดังเช่นในปัจจุบัน วิธีนี้จะไม่ใช่แบบใช้เวลาจริง เพราะสิ่งที่เป็นข้อความจะถูกเก็บไว้ในเซิร์ฟเวอร์ (server) สามารถเข้ามาดูเมื่อไหร่ก็ได้ สามารถลงความเห็น (Vote) หรือส่งข้อความเป็นการส่วนตัวก็ได้ สำหรับในปัจจุบันที่เทคโนโลยีต่างๆ พัฒนาขึ้นอย่างมากมาย ทำให้มีการพัฒนาโปรแกรมซึ่งเป็นการประชุมแบบเวลาจริงได้ ได้ยินทั้งเสียงและมองเห็นภาพได้ในเวลาเดียวกัน และจะอยู่ที่ใดในโลกก็ได้ ทั้งยังประหยัดกว่าแบบการประชุมด้วยกล้องวิดีโอที่จำเป็นต้องมีห้องประชุมโดยเฉพาะกับอุปกรณ์ราคาแพง แต่การส่งข้อมูลผ่านอินเตอร์เน็ตยังทำได้จำกัดอยู่ ผลลัพธ์จึงยังไม่ดีเท่า

2.2 รู้จักกับการทำงานของการประชุมทางไกลด้วยกล้องวิดีโอ

2.2.1 การใช้งานระบบการประชุมทางไกลด้วยกล้องวิดีโอ

การประมวลผลภาพวิดีโอค่อย ๆ พัฒนาเป็นสำคัญ การแพร่ภาพวิดีโอหรือการส่งสัญญาณโทรทัศน์มีมากกว่าห้าสิบปีแล้ว พัฒนาการของการแพร่ภาพเริ่มต้นจากการส่งภาพขาวดำ ต่อมานำสัญญาณสีมาใช้ร่วม แต่เมื่อคอมพิวเตอร์เจริญก้าวหน้าขึ้น การประมวลผลสัญญาณก็เริ่มเปลี่ยนจากอนาลอกมาเป็นดิจิตอล ภาพวิดีโอที่เห็นเป็นภาพขนาด 625 เส้น ที่จะต้องส่งให้ได้ไม่น้อยกว่า 25 เฟรมในหนึ่งวินาที และถ้าต้องการเปลี่ยนสัญญาณภาพแบบอนาลอกให้เป็นดิจิตอลจะต้องใช้แถบสัญญาณดิจิตอลถึง 90 ล้านบิตต่อวินาที การที่จะส่งสัญญาณดิจิตอลที่เป็นข้อมูลขนาด 90 เมกะบิตต่อวินาทีจึงไม่ใช่เรื่องง่าย เพราะข่ายสื่อสารส่วนใหญ่เป็นข่ายสัญญาณข้อมูลความเร็วต่ำ

วิชาการทางด้านการประมวลผลสัญญาณดิจิตอล และการสื่อสารข้อมูลจึงต้องนำมาใช้ ในสหรัฐอเมริกามีการทำโครงร่างการสื่อสารแบบ T1 ซึ่งเริ่มจากกลุ่มเล็กสุดคือ 56 กิโลบิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ทางยุโรปมีมาตรฐาน CCITT ที่ทางโครงร่างแบบ E1 คือเริ่มกลุ่มเล็กสุดคือ 64 กิโลบิต และ IE1 มีความเร็วของสัญญาณเท่ากับ 32 ช่องของ 64 กิโลบิต คือ 2048 กิโลบิตต่อวินาที

ขณะเดียวกันมาตรฐานระบบ ISDN-Integrated Service Data Network ซึ่งวางฐานของระบบบริการรวมไปบนเครือข่ายสวิตชิ่ง เช่น โทรศัพท์ โทรสาร โทรวิดีโอ หรือแม้แต่การส่งข้อมูลความเร็วสูง ก็ได้พัฒนาบนพื้นฐานของมาตรฐาน 2B+D คือมีช่องเสียงขนาด 64 กิโลบิต 2 ช่อง และ 16 กิโลบิต สำหรับข้อมูลหนึ่งของแถบกว้างที่เล็กสุดของ ISDN คือ 128 กิโลบิต + 16 กิโลบิต ระบบการประชุมทางไกลด้วยกล้องวิดีโอจะต้องมีเป้าหมายที่จะลดแถบกว้างของสัญญาณให้ลงมาเหลือขนาดที่จะส่งใน ISDN ได้ ลองนึกดูว่าจะต้องลดแถบกว้างของสัญญาณภาพทีวีขนาด 90 ล้านบิตต่อวินาทีให้เหลือ 128 กิโลบิตต่อวินาที นับว่าเป็นเรื่องที่ท้าทายมาก

การประมวลผลสัญญาณดิจิตอลและเทคนิคทางคณิตศาสตร์ ผนวกกับความก้าวหน้าในการผลิตชิพหรือ ULSI ที่ทำงานความเร็วสูง มีพัฒนาการที่รวดเร็วและก้าวหน้าจนในปัจจุบันสามารถผลิตชิพที่ทำงานตามอัลกอริทึมได้ซับซ้อนยิ่งจนระบบการประชุมทางไกลด้วยกล้องวิดีโอเป็นจริงขึ้นได้

2.2.2 หัวใจสำคัญของการประชุมทางไกลด้วยกล้องวิดีโออยู่ที่โคเด็ก (Codec)

Codec เป็นคำย่อมาจาก Code และ Decode คือ การเข้ารหัสและการถอดรหัสจากข้อมูลภาพที่มีจำนวนเส้น 625 เส้น 25 เฟรมต่อวินาที (กรณีสัญญาณ PAL) เมื่อแปลงเป็นสัญญาณดิจิตอลแล้วจะต้องเปลี่ยนกลับเป็นพิกเซลหรือจุดสี ปัญหามีอยู่ว่าจะใช้พิกเซลเท่าไรดี ตามมาตรฐาน CCITT H.261 ซึ่งเป็นมาตรฐานสำคัญที่กำหนดในเรื่องการเข้ารหัส กำหนดจำนวนเส้นใช้เพียง 288 เส้น แต่ละเส้นมีความละเอียด 352 พิกเซล นั่นหมายถึงได้ความละเอียดเท่ากับ 352 × 288 พิกเซล เรียกฟอร์แมตการแสดงผลนี้ว่า Common Intermediate format และยังยอมให้ใช้ความละเอียดแบบหนึ่งในสี่ คือลดจำนวนเส้นเหลือ 144 เส้น และพิกเซลหรือ 176 พิกเซล ซึ่งขึ้นอยู่กับขนาดของจอภาพ ถ้าใช้จอภาพขนาดเล็กจำนวนพิกเซลก็ลดลงไปได้

เมื่อจำนวนพิกเซลลดลงความละเอียดของภาพก็ลดลงโดยยังลดอัตราการแสดงภาพไว้เพียง 10-15 ภาพต่อวินาทีด้วยอัตราเหล่านี้จะทำให้ภาพเกิดการสั่นกระพริบ จึงจำเป็นต้องใช้วิธีการทางคณิตศาสตร์ ช่วยในการใช้หลักการประมาณค่าและสร้างภาพเสริมเพื่อให้ภาพนิ่ง ทฤษฎีการประมาณค่าทำให้ภาพต่อเนื่องและดูสมจริงสมจังเหมือนของเดิม

ที่สำคัญอยู่ที่หลักการการบีบอัดข้อมูลภาพ การบีบอัดข้อมูลภาพทำให้ลดขนาดข้อมูลภาพได้มาก แต่ต้องทำอย่างรวดเร็วเพื่อภาพที่ส่งจะไม่มีการหน่วงเวลา การประมวลผลภาพนี้จึงมีวิธีการทั้งทางด้านการประมวลผลขั้นต้น และการประมวลผลชดเชยไปยังด้านรับ ที่สำคัญคือใช้หลักการเปรียบเทียบภาพสองเฟรมติดกัน แยกส่วนแตกต่างแล้วจึงนำส่วนแตกต่างเข้ารหัสแล้วส่งไป การแยกส่วนแตกต่างของสองเฟรมติดกันนี้ ทำให้ลดขนาดข้อมูลภาพลงไปได้มาก เพราะภาพวิดีโอที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นภาพเคลื่อนไหว จะมีส่วนต่างของข้อมูลภาพในสองเฟรมติดกันไม่มาก และการประชุมทางไกลด้วยกล้องวิดีโอก็เป็นภาพที่ไม่ตัดต่อจากหลายกล้องมากนัก จึงทำให้วิธีการประมวลผลโดยแยกความแตกต่าง จึงเป็นสิ่งที่เหมาะสม มีการสร้างชิพเพื่อกระทำในเรื่องการเข้ารหัสเฉพาะเพื่อความรวดเร็ว

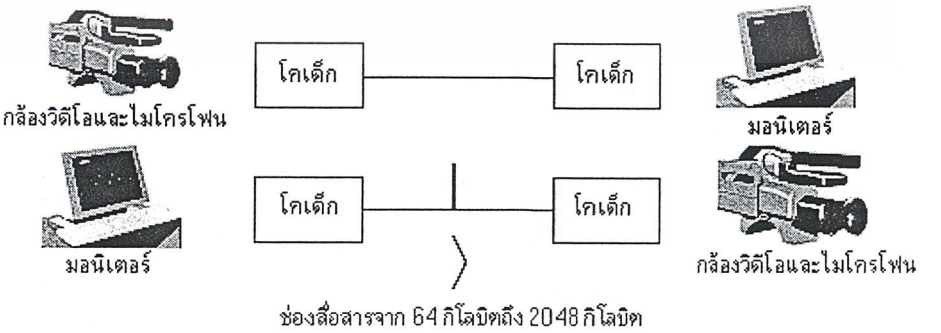
การประมวลผลสัญญาณภาพด้วยเทคนิคทางคณิตศาสตร์มีหลักการมากมาย เช่น การหาค่าของความเข้มเฉลี่ยของหลายพิกเซล การหาค่าประมาณเพื่อการชดเชยภาพเคลื่อนไหวที่อาจดูเป็นชิ้นให้มีการเคลื่อนไหวที่ต่อเนื่องดีขึ้น

นอกจากนี้ในเรื่องของเสียงก็มีการบีบอัดสัญญาณ ปกติเสียงที่ส่งในสัญญาณโทรศัพท์หนึ่งช่องเสียง ใช้อัตราสุ่ม 2 เท่าของแถบกว้างสัญญาณเสียง แถบกว้างสัญญาณเสียง 4 กิโลเฮิรตซ์ จึงใช้อัตราสุ่ม 8 กิโลเฮิรตซ์ใช้การแปลงอนาลอกเป็นดิจิตอลแบบ 8 บิต ดังนั้นช่องเสียงหนึ่งช่วงใช้แถบกว้าง 64 กิโลบิต การบีบอัดสัญญาณเสียงมีหลายเทคนิค เช่น ADPCM (Adaptive Pulse Code Modulation) การบีบอัดบางแบบ เช่น ที่ใช้ในโทรศัพท์มือถือสามารถลดแถบกว้างสัญญาณเสียงลดลงได้ถึงประมาณ 8 เท่า

การส่งสัญญาณการประชุมทางไกลด้วยกล้องวิดีโอเป็นการโต้ตอบกันสองทิศทาง ดังนั้นจะมีเสียงสะท้อนเกิดขึ้นอย่างมากมาย การสะท้อนเกิดจากการป้อนกลับของสัญญาณไปมา เช่น เสียงจากลำโพงป้อนกลับเข้าไมโครโฟนกลับไปมา คำที่เราได้ยินเสียงหอนในห้องประชุม ดังนั้นการประมวลผลสัญญาณจะมีเทคนิคพิเศษที่เรียกว่า การกำจัดเสียงสะท้อน (Echo Concellation)

2.2.3 โครงสร้างระบบการประชุมทางไกลด้วยกล้องวิดีโอ

ระบบการประชุมทางไกลด้วยกล้องวิดีโอที่ใช้กันอยู่ในขณะนี้มีหลายระดับหลายรูปแบบและหลายเทคนิค การประชุมทางไกลด้วยกล้องวิดีโอทั่วไปมีหลักการที่จะต้องลดขนาดภาพและเสียงลงให้เหลือเพียงไม่มากแล้วส่งในสายสัญญาณที่มีแถบกว้างไม่มากนัก ระบบการประชุมทางไกลด้วยกล้องวิดีโอทั่วไปมีโครงสร้างดังรูป 2.1



รูป 2.1 รูปแบบของการประชุมทางไกลด้วยกล้องวิดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องสื่อสารที่ใช้เป็นช่องสื่อสารแบบสองทิศทาง (full-duplex) ซึ่งมีความเร็วจำกัด โดยมีอุปกรณ์เข้ารหัสที่สำคัญเรียกว่าโคเด็ก เป็นตัวเข้ารหัสสัญญาณที่ส่งต่อ

อุปกรณ์ประกอบที่สำคัญของการประชุมทางไกลด้วยกล้องวิดีโอในระบบประกอบด้วย

- กล้องโทรทัศน์ปรับส่วนไปมาซูมกล้อง
- จอมอนิเตอร์แบ่งจอภาพดูปลายทางด้านใดด้านหนึ่ง
- อุปกรณ์อิเล็กทรอนิกส์ควบคุมระบบสื่อสาร ควบคุมเสียง ภาพ แหล่งจ่ายไฟและอินเตอร์เฟส
- แป้นควบคุมเพื่อควบคุมระยะไกลไปยังอีกปลายทางด้านหนึ่งได้

รูป 2.2 ส่วนประกอบของระบบการประชุมทางไกลด้วยกล้องวิดีโอ

2.2.4 อนาคตของการประชุมทางไกลด้วยกล้องวิดีโอ

การประชุมทางไกลด้วยกล้องวิดีโอยังเป็นอุปกรณ์ที่มีราคาแพง ประจวบกับช่องสื่อสารความเร็วสูงที่ใช้ในประเทศไทย เช่น ใช้สัญญาณดาวเทียม ใช้สายเช่าล้วนแล้วแต่มีราคาแพง ระบบการประชุมทางไกลด้วยวิดีโอจึงต้องใช้งานที่มีความสำคัญสูงเพื่อประโยชน์ที่คุ้มค่า

อย่างไรก็ดี การพัฒนาของระบบการประชุมทางไกลด้วยกล้องวิดีโอในปัจจุบันสามารถสร้างชิพ โคเด็กที่ลดสัญญาณลงเหลือขนาด 64 กิโลบิตได้ แต่ราคาโคเด็กมาตรฐาน H.261 ยังมีราคาแพง จึงทำให้ระบบการประชุมทางไกลด้วยกล้องวิดีโอราคาสูง

สิ่งที่น่าจับตายิ่งคือ ปัจจุบันมีการสร้างชิพ VCSI ที่ทำงานทางวิดีโอและโคเด็กได้ดี มีแนวโน้มที่ถูกลง ดังนั้นเชื่อแน่ว่าระบบการประชุมทางไกลด้วยกล้องวิดีโอจะมีราคาลดลงอีกมาก

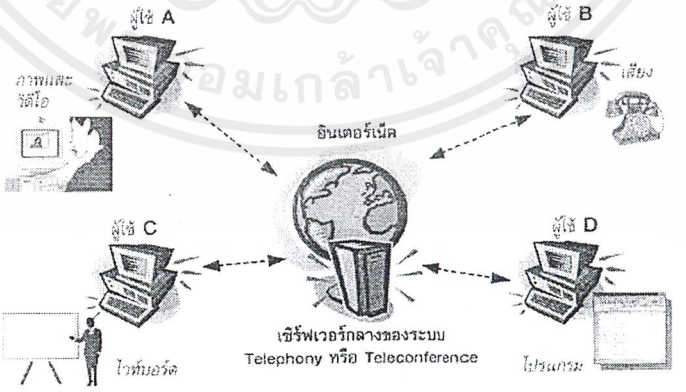
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในพีซีมีการ์ดโคเด็กตามมาตรฐาน H.261 ออกจำหน่ายแล้ว โดยใช้ช่องสื่อสารบนแลนด้วยภาพเชื่อมโยงหนึ่งช่องจาก 64 กิโลบิตขึ้นไป การ์ดตามมาตรฐาน H.261 จึงเป็นสิ่งที่น่าสนใจที่จะทำให้เครื่องคอมพิวเตอร์ส่วนบุคคลเป็นระบบการประชุมทางไกลด้วยกล้องวิดีโอที่ราคาถูกลง ปัจจุบันการ์ดโคเด็กบนพีซี ถ้าสั่งซื้อเข้ามาใช้จะตกราคาประมาณ 2 แสนบาทต่อการ์ด ในอนาคตอันใกล้นี้การ์ดโคเด็กบนพีซีจะต้องลดราคาลงอีกมาก และเชื่อว่าระบบการประชุมทางไกลด้วยกล้องวิดีโอบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่อยู่บนแลนจะมีราคาถูกลงจนมีผู้ใช้ได้อย่างแพร่หลาย

การคาดหวังทั้งหมดมีแนวโน้มที่เป็นจริงอย่างยิ่ง เชื่อแน่ว่าในไม่ช้าเราจะเริ่มเห็นโทรศัพท์ผ่านวิดีโอ การประชุมทางไกลด้วยกล้องวิดีโอ แม้กระทั่งในบ้านเรือน เพราะเป้าหมายของการบริการ ISDN ก็เป็นเป้าหมายหนึ่งที่วางไว้ให้ระบบสื่อสารเชื่อมโยงในระบบมัลติมีเดีย

2.3 โทรศัพท์ผ่านเครือข่ายอินเตอร์เน็ต (Internet Telephony) และการประชุมทางไกล

ปัจจุบันติดต่อด้วยเสียงผ่านอินเตอร์เน็ตในลักษณะคล้ายโทรศัพท์ (Internet Telephony) หรือบางทีก็เรียกว่าเป็นการส่งเสียงไปกับแพ็กเก็ต IP หรือ Voice-over-IP) และการประชุมทางไกล (teleconference) ผ่านอินเตอร์เน็ต นับเป็นแนวโน้มที่มีการใช้งานกันแพร่กลายมากขึ้น ซึ่งการประชุมทางไกลในที่นี้ไม่ได้จำกัดแค่ได้ยินเสียงหรือเห็นภาพนิ่งหรือภาพเคลื่อนไหวของแต่ละฝ่ายกันอีกต่อไป แต่ยังรวมไปถึงการแลกเปลี่ยนข้อมูลในเครื่องคอมพิวเตอร์ของผู้เข้าประชุมแต่ละคน การวาดภาพหรือสัญลักษณ์ลงบนกระดานที่เปรียบเสมือนไวท์บอร์ดในเครื่องของตนแล้วให้คนอื่นๆทั้งหมดเห็นได้พร้อมกัน รวมไปถึงการใช้งานโปรแกรมร่วมกัน เช่นแก้ไขตัวเลขในเวิร์กชีตของ Excel ที่ผู้เข้าประชุมคนอื่นนำเสนอ เพื่อให้ทุกคนเห็นว่าถ้าเพิ่มข้อมูลของเราเข้าไปแล้วจะได้ผลหรือการเปลี่ยนแปลงอย่างไร เป็นต้น



รูป 2.3 แสดงการติดต่อระหว่างผู้ใช้กับเซิร์ฟเวอร์ในการประชุมทางไกลซึ่งผู้ใช้แต่ละคนจะสามารถใช้ได้ทั้งเสียง ภาพ วิดีโอ ไวท์บอร์ด และการใช้โปรแกรมร่วมกันในการสื่อสารกับคนอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระยะแรกๆที่มีผลิตภัณฑ์และบริการเกี่ยวกับโทรศัพท์ผ่านอินเตอร์เน็ตออกมานั้น การทำงานของโปรแกรมหรือบริการต่างๆยังไม่มีมาตรฐานใดๆ ไม่ว่าจะเป็นเรื่องของโพรโตคอลที่ใช้โต้ตอบระหว่างกัน, การบีบอัดและขยายสัญญาณตลอดจนการทำงานของเซิร์ฟเวอร์ที่ให้บริการค้นหาผู้รับปลายทาง ดังนั้นโปรแกรมของคนละบริษัทจึงคุยกันไม่รู้เรื่อง ซึ่งเป็นอุปสรรคที่สำคัญในการขยายตัวของตลาดโทรศัพท์ผ่านอินเตอร์เน็ตในอนาคต ดังนั้นในเดือนพฤษภาคมปี ค.ศ. 1996 สหภาพโทรคมนาคมระหว่างประเทศ (International Telecommunications Union – ITU) จึงได้ให้สัตยาบันรับรองข้อกำหนดอันหนึ่งที่เรียกว่า H.323 ซึ่งเป็นมาตรฐานที่ระบุวิธีการส่งเสียง, ข้อมูล และภาพวิดีโอผ่านทางเครือข่ายท้องถิ่นชนิด IP และต่อมาในเวอร์ชัน 2 นี้ก็ถูกกำหนดให้ครอบคลุมถึงการสื่อสารในระยะไกลอย่างอินเตอร์เน็ตด้วย

H.323 เป็นชุดมาตรฐานที่สร้างขึ้นมาบนพื้นฐานของ Real-time Protocol และ Real-time Control Protocol (RTP/RTCP) รวมทั้งมาตรฐานสากลอื่นๆที่เกี่ยวกับการบีบอัดและขยายข้อมูล นอกจากนี้ H.323 ยังมีการทำงานที่เกี่ยวข้องกับมาตรฐานอีกชุดหนึ่งคือ T.120 ซึ่งว่าด้วยการใช้เอกสารและแอพพลิเคชันร่วมกันอันจะช่วยเสริมให้โปรแกรมโทรศัพท์ผ่านอินเตอร์เน็ตมีคุณสมบัติหลากหลายมากขึ้น ตัวอย่างเช่น โปรแกรม NetMeeting ของไมโครซอฟต์ที่ใช้มาตรฐานทั้ง 2 ชุดนี้จะสามารถสนทนาด้วยเสียงแลกเปลี่ยนข้อมูล และใช้โปรแกรมร่วมกันระหว่างเครื่องได้ด้วย ในขณะที่บางโปรแกรมที่เน้นการสื่อสารด้วยเสียงโดยเฉพาะ เช่น โทรศัพท์ผ่านอินเตอร์เน็ตสนับสนุนเฉพาะมาตรฐาน H.323 เท่านั้น

2.3.1 มาตรฐาน H.323

H.323 หรือเรียกเต็มๆว่า ITU-T H.323 นี้ไม่ใช่มาตรฐานตัวเดียวเดี่ยวๆ แต่เป็นชุดของมาตรฐานที่เกี่ยวพันกับหลายเรื่องด้วยกัน โดยจะครอบคลุมทั้งการสื่อสารแบบจุดต่อจุดและแบบหลายจุดพร้อมๆกัน สำหรับมาตรฐานย่อยในชุดนี้จะประกอบด้วย

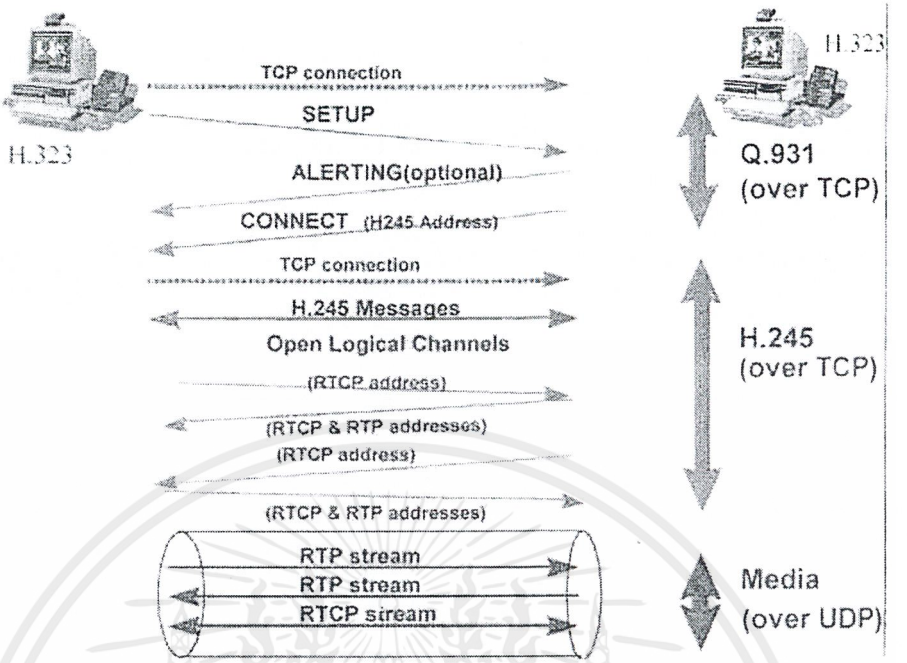
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 มาตรฐาน H.323

มาตรฐาน	คำอธิบาย
H.323	System document ที่อธิบายการทำงานของ H.323 ทั้งหมดในภาพรวม
H.225.0	กำหนด message ที่ใช้ควบคุมการเรียกหรือขอติดต่อไปยังอีกฝ่ายหนึ่ง (call control) รวมทั้งสัญญาณที่ใช้ (signaling), การตอบรับ (registration และ admission) รวมทั้งการแบ่งแพ็กเก็ตและควบคุมจังหวะการทำงานให้ตรงกันของข้อมูลที่ส่ง (packetization / synchronization of media stream)
H.245	กำหนด message ที่ใช้เปิดและปิด channel ของ media stream รวมไปถึงคำสั่ง (command) คำขอร้อง (request) และสัญญาณบอกสถานะ (indication) อื่นๆ
H.450.x	กำหนดแนวทางสำหรับบริการเสริม กระบวนการและสัญญาณที่ใช้สำหรับให้บริการในรูปแบบที่คล้ายกับโทรศัพท์ธรรมดา
H.235	กำหนดแนวทางของระบบรักษาความปลอดภัยที่ใช้ในการพิสูจน์ตัวตนของผู้ใช้ การเข้ารหัส ตลอดจนตรวจสอบความถูกต้องของข้อมูลในระบบ H.323 ทั้งหมด
H.332	กำหนดวิธีทำงานของระบบประชุมทางไกล (teleconference) ที่อาศัยมาตรฐาน H.323 อีกทีหนึ่ง
H.261	กำหนดรายละเอียดของ video codec (codec = coder / decoder หรือตัวเข้าและถอดรหัสข้อมูล) สำหรับการส่งภาพและเสียงที่ความเร็ว 64 Kbps
H.263	กำหนดรายละเอียดของ video codec สำหรับการส่งวิดีโอผ่านสายโทรศัพท์แบบเดิม (POTS-Plain Old Telephone System)
G.711	Audio codec, 3.1 KHz at 48, 56, 64 Kbps (เหมือนโทรศัพท์ธรรมดา)
G.722	Audio Codec, 7 KHz at 48, 56, and 64 Kbps
G.723	Audio Codec, for 5.3 and 6.3 Kbps modes
G.728	Audio Codec, 3.1 KHz at 16 Kbps
G.729	Audio Codec, 8 Kbps audio codec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของ H.323 จะเป็นไปดังรูป 2.4



รูป 2.4 แสดงขั้นตอนการติดต่อระหว่างผู้ใช้กับเซิร์ฟเวอร์ ในการทำงาน
ตามโพรโตคอลมาตรฐาน H.323

2.3.2 มาตรฐาน T.120

T.120 เป็นมาตรฐานอีกชุดหนึ่งของ ITU เรียกเต็มๆว่า ITU-T T.120 มาตรฐานนี้เกี่ยวข้องกับการใช้เอาสารและแอพพลิเคชันร่วมกันในระหว่างการประชุมระยะไกลแบบหลายจุดด้วยสื่อมัลติมีเดีย ตัวอย่างเช่น NetMeeting ซึ่งเป็นโปรแกรมหนึ่งที่ใช้มาตรฐานนี้เพื่อการแลกเปลี่ยนข้อมูลระหว่างผู้ร่วมสนทนา ทั้งในการสื่อสารผ่าน Whiteboard (T.126), File Transfer (T.127) และ Share Program (T.128)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.2 มาตรฐาน T.120

มาตรฐาน	คำอธิบาย
T.120	โพรโตคอลสำหรับการประชุมระยะไกลแบบมัลติมีเดียและเป็นบทนำที่สรุปข้อกำหนดในมาตรฐานชุด T.120 ทั้งหมด
T.121	ต้นแบบของแอปพลิเคชั่นที่จะพัฒนาให้ติดต่อกันผ่านทาง T.120 ทั้งหมด
T.122	ข้อกำหนดของบริการ Multipoint Communication Service (MCS) ซึ่งอธิบายว่ามีอะไรให้ผู้พัฒนาโปรแกรมเรียกใช้บ้าง
T.123	Protocol stack สำหรับแอปพลิเคชั่นการประชุมทางไกลที่ใช้ข้อมูลภาพและเสียง โดยกำหนดการทำงานของ transport protocol ที่ใช้กับเน็ตเวิร์คแบบต่างๆ
T.124	Generic Conference Control (GCC) กำหนด application protocol สำหรับจองทรัพยากรต่างๆในการประชุม (reservation) และบริการอื่นๆทั่วไปสำหรับควบคุมระบบประชุมทางไกลแบบ multipoint
T.125	ข้อกำหนดของโพรโตคอล Multpoint Communication Service (MCS) ระบุวิธีรับส่งข้อมูลในการติดต่อแบบหลายจุด (multipoint)
T.126	โพรโตคอลสำหรับการรับส่งภาพนิ่งและการทำเครื่องหมายเพิ่มเติมลงบนภาพ (annotation) ซึ่งจะกำหนดวิธีใช้ข้อมูลในการทำงานร่วมกัน รวมถึงการทำงานของไวท์บอร์ดในโปรแกรม การแสดงภาพกราฟิกและแลกเปลี่ยนข้อมูลภาพ
T.127	โพรโตคอลสำหรับรับส่งไฟล์ (Binary File Transfer Protocol) ซึ่งจะกำหนดวิธีที่แอปพลิเคชั่นจะรับส่งไฟล์ในการประชุมแบบ multipoint
T.128	โพรโตคอลสำหรับการกำหนดวิธีที่ผู้เข้าร่วมประชุมโปรแกรมร่วมกันในลักษณะที่ผู้อื่นจะสามารถเห็นผลการทำงานของโปรแกรมนั้น สามารถใช้เมาส์และคีย์บอร์ดของตนควบคุมการทำงานได้เหมือนว่าทำงานบนเครื่องของตนเอง
T.134	กำหนดวิธีทำงานของระบบพิมพ์ คุยกันด้วยข้อความ (chat)
T.135	ระบบการจองทรัพยากรในการประชุมแบบของ T.120 ซึ่งเป็นการติดต่อระหว่างแอปพลิเคชั่นบนเครื่องไคลเอนต์กับระบบจัดคิว (scheduling system) ที่ควบคุมทรัพยากรของ Multipoint Control Unit (MCU หรือ “bridge”)
T.136	กำหนดวิธีควบคุมและจัด configuration ของอุปกรณ์ซึ่งอยู่อีกที่หนึ่งโดยใช้ T.120 เป็นโพรโตคอลในระดับ Transport
T.140	กำหนดวิธีทำงานของระบบ chat ในแอปพลิเคชั่นที่เป็นมัลติมีเดีย
T.VMR	ระบบควบคุม Virtual Meeting Room โดยมีข้อกำหนดคล้ายกับในชุดอื่นๆ แต่เพิ่มรายละเอียดในส่วนของการประชุมด้วยเสียงและการแลกเปลี่ยนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ปัญหาและพัฒนาการ

ปัจจุบันผลิตภัณฑ์และบริการต่างๆที่เกียวข้องกับการสนทนาหรือการประชุมผ่านอินเตอร์เน็ต ต่างก็เป็นไปตามมาตรฐาน H.323 มากขึ้น ไม่ว่าจะเป็นซอฟต์แวร์, โมเด็ม, เกทเวย์ (Gateway) หรือเซิร์ฟเวอร์สำหรับการประชุมด้วยสื่อวิดีโอ ซึ่งในทางทฤษฎีก็หมายความว่า ผลิตภัณฑ์ทั้งหมดที่สนับสนุนมาตรฐานนี้จะสามารถทำงานร่วมกันได้ เช่น เราอาจจะใช้โปรแกรมคนละตัวเพื่อติดต่อสนทนากัน เพียงแต่อาจจะใช้คุณสมบัติร่วมกันไม่ได้ทุกอย่าง เพราะบางโปรแกรมไม่ได้นำเอามาตรฐานย่อยบางตัวไปใช้ อย่างไรก็ดีเนื่องจากมาตรฐานนี้ค่อนข้างซับซ้อน ดังนั้นในทางปฏิบัติจึงอาจจะเกิดปัญหาเวลาใช้โปรแกรม อุปกรณ์ หรือบริการจากต่างที่ต่างทางร่วมกันก็เป็นได้

นอกจากนี้เนื่องจากจุดมุ่งหมายเดิมของ H.323 คือการใช้งานบนระบบแลน หลายๆส่วนของมาตรฐานนี้จึงยังไม่เหมาะกับการใช้งานบนอินเตอร์เน็ต เช่น วิธีบีบอัดข้อมูลเสียงที่เป็นดีฟอลต์คือ G.711 นั้นต้องการช่วงกว้างของสัญญาณ (bandwidth) ถึง 64 กิโลบิตต่อวินาที ซึ่งไม่เหมาะกับการเชื่อมต่ออินเตอร์เน็ตที่มีความเร็วต่ำกว่านี้, การทำงานโดยผ่าน Proxy server และ Firewall ซึ่งทำได้แต่การติดตั้งค่อนข้างยุ่งยาก รวมทั้งข้อกำหนดของการใช้แอปพลิเคชั่นร่วมกัน ที่อนุญาตให้เครื่องอื่นสามารถควบคุมเครื่องที่ถูกแชร์ได้อย่างไม่มีขีดจำกัด ไม่ว่าจะเป็นการอ่าน เขียน และลบไฟล์ หรือการรันโปรแกรม ดังนั้นผู้ที่นำมาตรฐานนี้ไปใช้จึงต้องหาวิธีควบคุมหรือระบบรักษาความปลอดภัยที่เหมาะสมเอาเอง ซึ่งก็ทำให้การใช้ต่างโปรแกรมมาทำงานร่วมกันเป็นไปได้ยากยิ่งขึ้น

อีกประเด็นหนึ่งที่เป็นปัญหาและมีผลกระทบต่อคุณภาพของบริการ ก็เนื่องมาจากคุณสมบัติของโพรโตคอล RTP ซึ่งไม่อาจรับประกันการล่าช้าหรือสูญหายของแพ็กเก็ตที่ส่งผ่านไปมา รวมทั้งไม่มีความสามารถจัดเตรียมช่วงกว้างของสัญญาณให้กว้างพอ ทำให้โดยรวมแล้วไม่สามารถควบคุมคุณภาพ (Quality of Service หรือ Qos) ของข้อมูลที่ขึ้นกับเวลาและมีปริมาณมากๆอย่างเสียงหรือวิดีโอได้ เพื่อแก้ปัญหานี้ทาง Internet Engineering Task Force (IETF) จึงได้นำเสนอโพรโตคอลใหม่ที่เรียกว่า Resource Reservation Protocol หรือ RSVP (RFC 1633) ขึ้นมา ซึ่งจะมีวิธีการจองทรัพยากรของเครือข่ายไว้ใช้รับส่งภาพและเสียงได้อย่างเต็มที่ ทำให้สามารถรับประกันคุณภาพของบริการได้ในระดับหนึ่ง อย่างไรก็ดีกว่าที่ RSVP จะได้รับการยอมรับและมีการใช้งานในอุปกรณ์ต่างๆอย่างกว้างขวาง ก็คงต้องใช้เวลาอีกระยะหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ไดเร็กเอ็กซ์ (DirectX)

3.1 ไดเร็กเอ็กซ์คืออะไร

ไดเร็กเอ็กซ์ (DirectX) คือ กลุ่มของแอปพลิเคชั่นโปรแกรมมิ่งอินเตอร์เฟส (Application Programming Interface) หรือ เอพีไอ (API) ที่ช่วยสนับสนุนการสร้างแอปพลิเคชั่นแบบเรียลไทม์ (Real-time Applications) ด้านเกมและมัลติมีเดียที่ต้องการประสิทธิภาพการทำงานสูง

3.2 จุดประสงค์ของไดเร็กเอ็กซ์

การพัฒนาไดเร็กเอ็กซ์เพื่อเพิ่มประสิทธิภาพการทำงานด้านมัลติมีเดียของแอปพลิเคชั่นที่ทำงานอยู่บนระบบปฏิบัติการไมโครซอฟต์วินโดวส์ (Microsoft Windows) ให้มีความสามารถเทียบเท่าหรือสูงกว่าประสิทธิภาพของแอปพลิเคชั่นที่ทำงานอยู่บนระบบปฏิบัติการ MS-DOS หรือเครื่องเกมประเภทคอนโซล (Game Consoles)

3.3 ประโยชน์ของไดเร็กเอ็กซ์

1. ประโยชน์ในด้านการพัฒนา DirectX Windows Application
2. เป็นมาตรฐานสำหรับการพัฒนาฮาร์ดแวร์

3.4 ส่วนประกอบของไดเร็กเอ็กซ์

1. DirectDraw จะเป็นเทคนิคการเร่งฮาร์ดแวร์และซอฟต์แวร์แอนิเมชั่น โดยการเข้าถึงรูปภาพ bitmap ในหน่วยความจำ (ไม่ต้องผ่านระบบปฏิบัติการ Windows) ทำให้แอปพลิเคชั่นต้องการเพียงการจัดการฮาร์ดแวร์พื้นฐานง่ายๆ โดยไม่ต้องเรียก procedure ที่เจาะจงสำหรับจัดการกับสีต่างๆ รวมทั้งรองรับการจัดการกับ palettes, clipping และ animation การใช้ DirectDraw ทำให้จัดการกับหน่วยความจำได้ง่ายขึ้น

2. DirectSound ทำให้ใช้ฮาร์ดแวร์และซอฟต์แวร์ทางด้านเสียงได้ และเข้าถึงฮาร์ดแวร์ด้านเสียงได้โดยตรง DirectSound จะทำให้การทำงานของไดรเวอร์ (drivers) ต่างๆสามารถเข้ากันได้ DirectSound สามารถรับข้อมูลเสียง และเล่นได้ DirectSound ยังรองรับคุณสมบัติต่างๆที่จะทำให้ผู้พัฒนาโปรแกรมนำข้อดีต่างๆที่เพิ่มขึ้นมาจากการ์ดเสียง และ drivers ต่างๆที่เกี่ยวข้อง

3. DirectSound3D ทำหน้าที่ติดต่อให้สามารถวางตำแหน่งของเสียงในระนาบ 3 มิติได้ ทำให้มีความรู้สึกถึงเสียงรอบทิศทาง

4. DirectMusic เป็นอุปกรณ์ทางด้านเสียงเพลงของ DirectX ไม่เหมือนกับ DirectSound ตรงที่ DirectMusic จะเล่นข้อมูลที่เป็นเพลง DirectMusic รองรับมาตรฐาน Musical Instrument

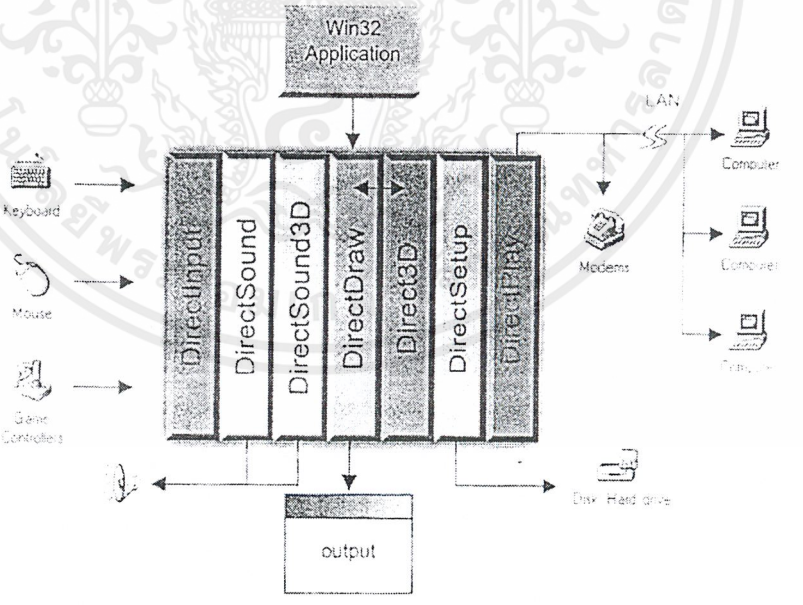
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Interface (MIDI) และ downloadable sounds (DLS), DirectMusic จะมี tools สำหรับแต่งและเล่นเพลง

5. DirectPlay ทำให้การติดต่อสำหรับเกมระบบผู้เล่นหลายๆคนง่ายขึ้น และสะดวกในการใช้งานร่วมกับโมเด็ม DirectPlay จะจัดหาการติดต่อสื่อสารที่มีประสิทธิภาพให้กับแอปพลิเคชั่นทำให้ผู้พัฒนาแอปพลิเคชั่นไม่ต้องมาพะวงกับการเชื่อต่อโปรโตคอลต่างๆเข้าด้วยกัน ทำให้พัฒนาแอปพลิเคชั่นของตนได้ง่าย

6. Direct3D ใช้ในการติดต่อฮาร์ดแวร์ในรูปแบบ 3 มิติ บนระบบปฏิบัติการวินโดวส์ ซึ่งได้เปรียบจากการที่สามารถใช้งานร่วมกับการ์ดเร่งความเร็วกราฟิกแบบ 3 มิติ ได้อย่างมีประสิทธิภาพเพราะมีฟังก์ชันคอยสนับสนุนนั่นเอง นอกจากนั้นสามารถรองรับการทำงานบนระบบปฏิบัติการ Windows NT ได้ และใช้เทคโนโลยี MMX ได้โดยตรง Direct3D มี 2 ส่วนประกอบ คือ Direct3D Retained-Mode API และ Direct3D Immediate Mode API ผู้พัฒนาส่วนมากจะใช้ Immediate Mode แทน Retained-Mode สำหรับโปรแกรมที่ต้องการความละเอียดทางกราฟิกสูง

7. DirectInput รองรับระบบการสั่งงานจากอุปกรณ์อินพุต ไม่ว่าจะเป็นจอยสติก เมาส์ หรือคีย์บอร์ด ทำได้ง่ายและมีประสิทธิภาพเป็นอย่างยิ่ง อีกทั้งยังมีฟังก์ชันที่สนับสนุนการเพิ่มเติมอุปกรณ์การเล่นที่จะมีใหม่ในอนาคต DirectInput จะทำให้สามรถเข้าถึงข้อมูลอินพุตได้เร็วกว่าเดิม เพราะมันไม่ต้องพึ่งพาการทำงานของระบบปฏิบัติการวินโดวส์ แต่จะสื่อสารโดยตรงกับฮาร์ดแวร์



รูป 3.1 ส่วนประกอบของไดเร็กเอ็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 DirectSound

การโปรแกรมเสียงเป็นงานที่มีความยาก เนื่องจากต้องเข้าใจเสียง และเสียงดนตรี ซึ่งระบบเสียงที่เขียนขึ้นนั้นต้องทำงานได้เมื่อนำไปใช้การ์ดเสียงทุกชนิด ในยุคแรกๆนั้น นักโปรแกรมเมอร์มักใช้เสียงไลบรารีเสียงสำหรับระบบดอสและวินโดวส์จากผู้ผลิตรายที่สาม ซึ่งมีราคาสูงมาก และสำหรับระบบดอส (DOS) ไลบรารีเสียงทำงานได้ดี แต่กับระบบวินโดวส์ถึงแม้ว่าจะรองรับการเล่น (playback) เสียงและมัลติมีเดีย แต่ไม่รองรับการแสดงภาพเกม (video game) แบบ real-time ปัจจุบัน DirectSound ถูกสร้างขึ้นโดยสามารถแก้ไขปัญหาดังกล่าวได้ รวมถึงการรองรับการทำงานกับการ์ดเสียงได้มากถึงล้านตัว และการแสดงเสียงแบบ 3 มิติ

3.5.1 ประเภทของเสียงบนคอมพิวเตอร์

คอมพิวเตอร์สามารถสร้างเสียงได้ 2 ประเภท คือ เสียงดิจิตอล และเสียงสังเคราะห์

เสียงดิจิตอลเป็นเสียงเอฟเฟ็กต์ที่อัดได้จากการสั่นสะเทือนของเสียงต่างๆ เช่น เสียงระเบิด เสียงผู้คนสนทนากัน จากนั้นแปลงเป็นสัญญาณไฟฟ้า หรืออนาลอก และแปลงเป็นสัญญาณดิจิตอล โดยการแซมเปิล ซึ่งอัตราการแซมเปิล หมายถึง จำนวนครั้งที่แซมเปิลในช่วงเวลาหนึ่งวินาที และสามารถแซมเปิลเสียงได้จากความถี่และแอมปลิจูด การแซมเปิลด้วยความถี่จะต้องทำที่ความถี่ 2 เท่าของความถี่เสียงที่ต้องการอัด และการแซมเปิลด้วยแอมปลิจูดขนาด 8 บิตในปัจจุบันเพียงพอกับการนำไปใช้ในเกม และขนาด 16 บิตสำหรับการสร้างเสียงและดนตรีคุณภาพสูง

เสียงสังเคราะห์เป็นการโปรแกรมเสียง หรือนำเสียงมาสร้างใหม่ โดยเสียงที่ได้จะเป็นไปตามอัลกอริทึม และคุณภาพของฮาร์ดแวร์เสียง หรือกล่าวได้ว่าเสียงสังเคราะห์ใช้เฉพาะสำหรับเป็นเสียงดนตรีเท่านั้น ซึ่งเก็บในรูปแบบไฟล์ MIDI (Musical Instrument Digital Interface) ซึ่งเสียงที่ได้จะดูเรียบง่ายเกินไปถ้าเทียบกับเสียงธรรมชาติ และขาดความรู้สึกที่จะได้รับจากเสียงดิจิตอล และต้องอาศัยเทคนิคขั้นสูงในการสร้างเสียงสังเคราะห์ให้ใกล้เคียงเสียงดิจิตอล ได้แก่ wave table และ wave guide

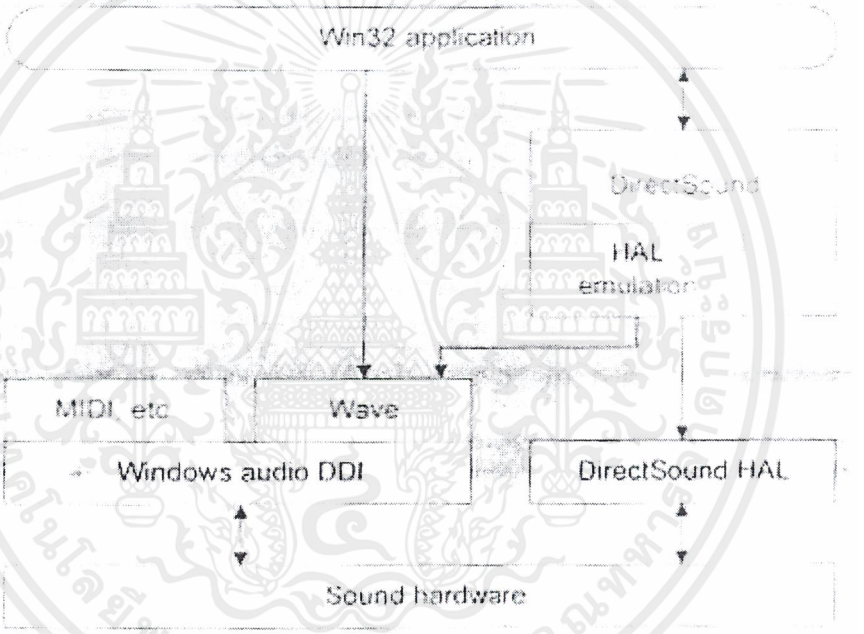
3.5.2 รู้จักกับ DirectSound

DirectSound ประกอบด้วยจำนวนคอมโพเนนท์ หรืออินเตอร์เฟสเช่นเดียวกับ DirectDraw สำหรับกรณีที่มีการ์ดเสียง สามารถใช้งานผ่านไดรเวอร์ DirectSound และถ้าไม่มีฮาร์ดแวร์เสียง DirectSound สามารถทำงานได้โดย HEL หรือการจำลองฮาร์ดแวร์ และวินโดวส์ DDI (Device Driver Interface) โดยการรวมผลงานเสียงที่สร้างเข้ากับ DirectSound COM (Component Object Model) และร้องขออินเตอร์เฟสจากออบเจ็กต์หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Win32 API บนวินโดวส์มีฟังก์ชันสร้างเสียงให้กับแอปพลิเคชั่นได้ระดับหนึ่ง แต่ในระบบของเกมที่แท้จริงต้องการ tool ที่มีประสิทธิภาพมากกว่านั้น โดย DirectSound ได้เสนอสิ่งต่างๆ เหล่านี้ ได้แก่

- การเร่งความเร็วโดยอัตโนมัติ ถ้ามีฮาร์ดแวร์เหล่านั้นอยู่
- การรวมเสียงต่างๆเข้าด้วยกันอย่างไม่จำกัด
- การกำหนดตำแหน่งของเสียง 3 มิติโดย Direct3D
- การแปลงข้อมูลอินพุตของเสียงในรูปแบบต่างๆหลายรูปแบบให้ได้เอาท์พุตที่ตรงกัน
- รองรับกำหนดคุณสมบัติใหม่ๆที่มีเข้ามาของฮาร์ดแวร์โดยไม่ต้องเปลี่ยน API
- การเล่นข้อมูลเสียงใช้ทรัพยากรต่ำ



รูป 3.2 สถาปัตยกรรมของ DirectSound

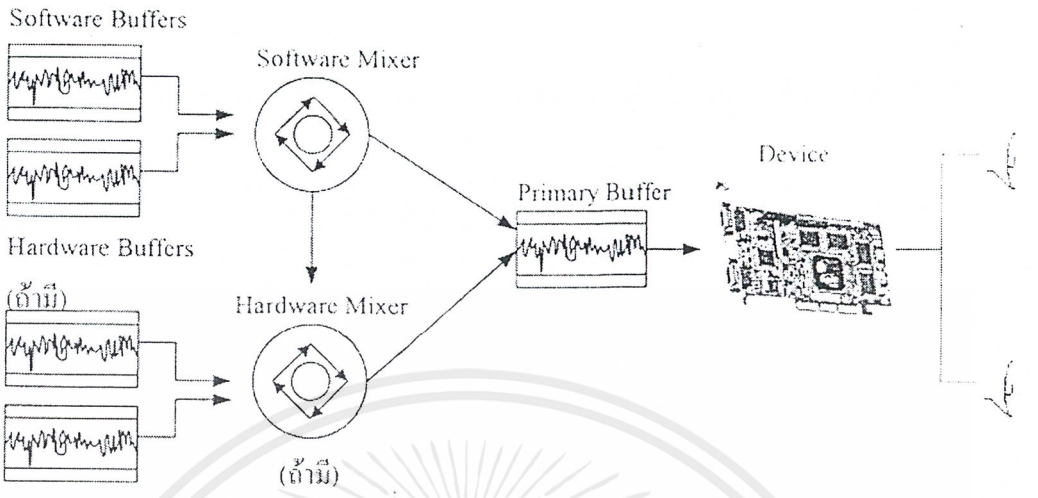
การเรียกใช้ DirectSound ประกอบด้วย 3 ส่วน

1. ขณะ run-time.DLL (Device Driver Interface) จะถูกโหลดเข้าไปเมื่อใช้ DirectSound
2. ขณะ compile-time ใช้ library DSOUND.LIB
3. เรียกไฟล์ส่วนหัวที่ชื่อ DSOUND.H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

3.5.3 การทำงานของ DirectSound



รูป 3.3 การทำงานของ DirectSound

DirectSound จะเริ่มด้วยออบเจ็กต์บัฟเฟอร์เสียง secondary ซึ่งแทนเสียงเพียงเสียงเดียว เสียงเหล่านี้อาจเป็น static sound หรือ streaming sound ก็ได้

Static sound คือ เสียงสั้นๆที่มีขนาดข้อมูลที่พอดีกับหน่วยความจำ

Streaming sound คือ ข้อมูลเสียงส่วนหนึ่งที่ย้ายจากหน่วยความจำมาเก็บไว้ในบัฟเฟอร์ช่วงขณะหนึ่ง โดยบัฟเฟอร์ทั้งหมดจะมีรูปแบบการแซมเปิลสัญญาณเสียงแบบ PCM (Pulse Code Modulate)

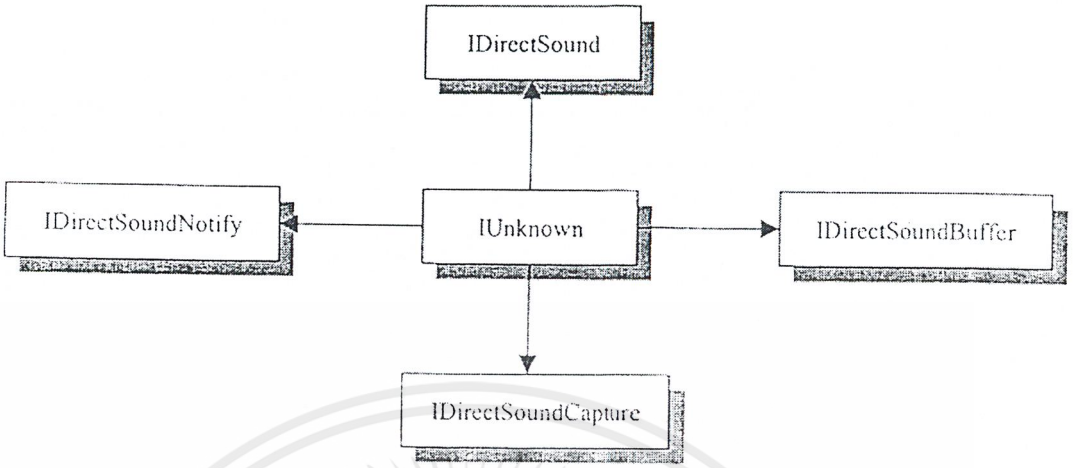
DirectSound จะนำข้อมูลจากบัฟเฟอร์ secondary แต่ละอันมาผสานเข้าด้วยกันในบัฟเฟอร์ primary โดยจะทำการแปลงรูปแบบของข้อมูลที่จำเป็น เช่น แปลงอัตราการแซมเปิล และประยุกต์ให้มีเสียงเอฟเฟ็กต์พิเศษเพิ่มเติม เช่น การจัดตำแหน่งของเสียงในระบบ 3 มิติ

ถ้าภายในฮาร์ดแวร์มีหน่วยความจำและตัวผสานเสียง DirectSound จะนำบัฟเฟอร์ที่สร้างขึ้นไปเก็บในหน่วยความจำของฮาร์ดแวร์โดยอัตโนมัติให้ได้มากที่สุดก่อน และถ้าไม่มีหรือมีไม่เพียงพอ ก็จะนำบัฟเฟอร์ไปเก็บในหน่วยความจำของระบบ หรือบัฟเฟอร์ซอฟต์แวร์ ซึ่ง DirectSound จะผสานเสียงโดยใช้บัฟเฟอร์ฮาร์ดแวร์ และส่งไปยังบัฟเฟอร์ primary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42700

3.5.4 การติดต่อของ DirectSound



รูป 3.4 การติดต่อของ DirectSound

Iunknown : คือ ออบเจ็กต์ COM หลัก

IdirectSound : คือ ออบเจ็กต์ COM หลักของ DirectSound ใช้แทนฮาร์ดแวร์เสียงของมันเอง การ์ดเสียง 1 อัน แทนออบเจ็กต์ 1 ออบเจ็กต์

IdirectSoundBuffer : คือ ฮาร์ดแวร์ผสาน (mix) เสียง และเก็บข้อมูลเสียง DirectSound มีบัฟเฟอร์อยู่ 2 ชนิด คือ บัฟเฟอร์ primary เป็นเสียงที่กำลังเล่นอยู่ และผสานเข้าด้วยกันโดยฮาร์ดแวร์ (ถ้ามี) หรือซอฟต์แวร์ ต้องเก็บไว้ในหน่วยความจำของฮาร์ดแวร์เท่านั้น และบัฟเฟอร์ secondary แทนเสียงที่ถูกเก็บไว้เพื่อจะเล่น อาจเก็บไว้ในหน่วยความจำระบบ หรือใน SRAM (Sound RAM) บนการ์ดเสียง

IdirectSoundCapture : เป็นออบเจ็กต์ COM ของ DirectX ซึ่งทำการบันทึกและจับ (capture) ข้อมูลเสียงเข้ามา เช่น voice-recognition เป็นออบเจ็กต์ที่ทำงานอิสระไม่ขึ้นกับ DirectSound และมีอินเตอร์เฟสระดับบนเป็นของตัวเอง หรือ IdirectSoundCapture ซึ่งมีส่วนที่ครอบคลุม COM (COM Wrapper) อีกทีหนึ่ง เพื่อให้สามารถใช้งานฟังก์ชันของ waveform API บน Win32 ดังนั้นระบบปฏิบัติการวินโดวส์ 98 และวินโดวส์ NT จะมีไดรเวอร์ Win32 Driver Model (WDM) ซึ่งสนับสนุนการเร่งความเร็วของฮาร์ดแวร์

IdirectSoundNotify : การส่ง message กลับไปให้ DirectSound จำเป็นสำหรับระบบเสียงที่มีความซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.5 รูปแบบเสียงใน DirectSound (โครงสร้าง WAVEFORMATEX)

- nChannels (WORD) : จำนวนช่องสัญญาณ (channel) 1 คือ mono และ 2 คือ stereo
- nSamplesPerSec (DWORD) : อัตราแซมเปิลความถี่ มีหน่วยเป็นเฮิรตซ์ หรือจำนวนครั้งที่แซมเปิลในช่วงเวลา 1 วินาที สำหรับ 1 ช่องสัญญาณ ดังนั้นเสียงแบบ stereo จะมีอัตราการแซมเปิลเป็น 2 เท่า ซึ่งในการแซมเปิลจะแทรกช่องสัญญาณทางซ้ายก่อน โดยทั่วไปฮาร์ดแวร์สำหรับพีซีจะรองรับอัตราการแซมเปิลขนาด 11025 22050 และ 44100 เฮิรตซ์
- wBitsPerSample (WORD) : เป็นจำนวนบิตสำหรับการแซมเปิลแอมปลิจูด มักใช้ 8/16 บิต
- wFormatTag (WORD) : เป็นตัวระบุว่าข้อมูลจะแปลงไปอย่างไร สำหรับ DirectSound จะใช้ WAVE_FORMAT_PCMtag ซึ่งเท่ากับ 1
- nBlockAlign (WORD) : เป็นจำนวนไบต์ในการแซมเปิลหนึ่งครั้ง หรือ nChannel * wBitsPerSample/8
- nAvgBytesPerSec (WORD) : อัตราการส่งข้อมูลเฉลี่ย หรือ nSamplesPerSec * nBlockAlign

3.5.6 การเข้าถึง DirectSound

การเข้าถึงระบบ DirectSound ต้องทำตามขั้นตอนต่อไปนี้

1. รับ GUID สำหรับอุปกรณ์ด้านเสียง
2. สร้างออบเจ็กต์ของ DirectSound (จำเป็น)
3. กำหนด Cooperative level (จำเป็น)
4. กำหนดรูปแบบของบัฟเฟอร์ primary

3.5.6.1 การกำหนดอุปกรณ์เสียง (Enumeration)

DirectSound เป็นออบเจ็กต์ COM ของ DirectX ตัวหนึ่ง ดังนั้นจะมีฟังก์ชันที่ระบุว่าจะเลือกใช้ฮาร์ดแวร์ตัวใดจากไดรเวอร์ที่มีอยู่ และส่งข้อมูลของฮาร์ดแวร์นั้นกลับไป ซึ่งการกำหนดอุปกรณ์ไม่จำเป็นต้องทำได้ เพราะในการเริ่มต้นสร้างออบเจ็กต์ DirectSound ระบบจะเลือกใช้อุปกรณ์ด้านเสียงที่เหมาะสมมากกว่าอยู่แล้ว หรือเราสามารถส่งค่า NULL ไปให้ DirectSound เพื่อกำหนดให้อุปกรณ์ของเราเป็น default ได้เช่นกัน หรือผู้ใช้สามารถเลือกอุปกรณ์ที่เหมาะสมเองได้จากรายการที่ระบบมีให้เลือก

ก่อนที่จะสร้างออบเจ็กต์ DirectSound ขึ้น เราจะเรียกฟังก์ชัน DirectSoundEnumerate ซึ่งจะต้องประสานงานร่วมกับอุปกรณ์ที่ได้ระบุไว้ และฟังก์ชันดังกล่าวประกอบด้วยตัวแปร 2 ตัว คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- lpDSEnumCallback (LPDSENUMCALLBACK) : เป็นพอยทเตอร์ชี้ไปที่แอดเดรสของฟังก์ชัน Callback ของอุปกรณ์ที่เลือก
- lpContex (LPVOID) : เป็นพอยทเตอร์ที่ชี้ไปยังข้อมูลขนาด 32 บิตซึ่งต้องการส่งหรือรับจากฟังก์ชัน Callback ของอุปกรณ์ที่เลือก

หลักการ : เมื่อเรียกใช้ฟังก์ชัน DirectSoundEnumerate และ DirectSound จะสร้างรายการของอุปกรณ์เสียงที่สามารถเลือกใช้งานได้ในระบบ ส่งข้อมูลเสียงของแต่ละอุปกรณ์กลับไปยังฟังก์ชัน Callback ของแอปพลิเคชั่น และฟังก์ชัน DSEnumCallvack จะต้องประกาศเป็น "BOOL CALLBACK" ซึ่งประกอบด้วยตัวแปร 4 ตัว คือ

- lpGuid (LPGUID) : พอยทเตอร์ชี้ไป GUID ของอุปกรณ์
- lpstrDescription (LPCSTR) : เป็นชื่อ Friendly ของอุปกรณ์
- lpcstrModule (LPCSTR) : เป็นชื่อโมดูล
- lpContext (LPVOID) : ข้อมูลเสียง

โดย DirectSound จะรับค่าจากตัวแปร 3 ตัวแรกจากไดรเวอร์ของอุปกรณ์ ส่วน lpContext เป็นค่าเดียวกันกับที่ต้องส่งไปให้ฟังก์ชัน DirectSoundEnumerate และฟังก์ชัน DSEnumCallback จะคืนค่าเป็น true/false เพื่อบอกให้ดำเนิน/สิ้นสุดการกำหนดอุปกรณ์ แต่อย่างไรก็ตาม สามารถเรียกใช้เมธอด GetCaps เพื่อตรวจสอบความสามารถของอุปกรณ์ (ภายในฟังก์ชัน Callback) ในขณะที่เริ่มต้นสร้างออบเจ็กต์ DirectSound ได้เช่นกัน แล้วจึงเลือกที่จะปลดอุปกรณ์และทำการกำหนดอุปกรณ์ครั้งใหม่ต่อไป (โดยการคืนค่าฟังก์ชันเป็น true) หรือเก็บอุปกรณ์ไว้และหยุดการกำหนดอุปกรณ์ (คืนค่าเป็น false)

3.5.6.2 การสร้างออบเจ็กต์ DirectSound

ฟังก์ชัน DirectSoundCreate มีหน้าที่ในการสร้างออบเจ็กต์ DirectSound โดยรับค่า GUID ของอุปกรณ์ตัวอย่างในรายการ หรือค่า NULL ในกรณีที่เลือกอุปกรณ์เอง ซึ่งใช้พารามิเตอร์ LPGUID lpGuid เป็นพอยทเตอร์ไปยังค่าดังกล่าว หลังจากการกำหนดอุปกรณ์

ข้อควรระวัง : การสร้างออบเจ็กต์ DirectSound สามารถสร้างหลายๆออบเจ็กต์ได้ แต่ปัญหาคือ เมื่อกำหนด Cooperative level ในออบเจ็กต์ใหม่ ค่า cooperative level จะไปทับของเดิม

ตัวอย่าง :

```
LPDIRECTSOUND lpds;
If (DirectSoundCreate(Null, &lpds, NULL)!= DS_OK) {/* error */}
Lpds -> Release( );
```

เพิ่มเติม : DSOK หรือ DirectSound OK และไม่ว่าจะทำอะไรกับออบเจ็กต์ DirectSound ต้องทำการ release ออบเจ็กต์ทุกครั้ง เพื่อลบพื้นที่บนหน่วยความจำซึ่งจองไว้สำหรับออบเจ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6.3 การกำหนดระดับ Cooperative

เมื่อสร้างออบเจ็กต์ DirectSound จะต้องกำหนดระดับ cooperative เพื่อให้ระบบวินโดวส์จัดการการทำงานของออบเจ็กต์ และการแบ่ง (share) อุปกรณ์เสียงให้กับแอปพลิเคชั่น และงานอื่นๆบนระบบและกำหนดขีดความสามารถของแอปพลิเคชั่นในการควบคุมฮาร์ดแวร์

การกำหนดระดับ cooperative แบ่งเป็น 2 กลุ่มหลัก คือ การกำหนดซึ่งทำให้นักพัฒนาเกมสามารถ/ไม่สามารถควบคุมตัวผสานเสียงของฮาร์ดแวร์ หรือบัฟเฟอร์เสียง ระบบวินโดวส์

อินเตอร์เฟส IdirectSound ใช้เมธอด SetCooperativeLevel ( ) ในการกำหนดระดับ Cooperative ซึ่งมีพารามิเตอร์ 2 ค่า คือ

1. HWND (hwnd) : เป็นส่วนจัดการวินโดวส์
2. Dwlevel (DWORD) : เป็นการกำหนดระดับ cooperative ซึ่งได้แก่

- DSSCL_NORMAL : เป็นระดับที่ดีที่สุดสำหรับแอปพลิเคชั่นทุกแอปพลิเคชั่น คือ สามารถเล่นเสียงได้ แต่ไม่ได้รับอนุญาตให้โปรแกรม และเปลี่ยนรูปแบบบัฟเฟอร์ primary ซึ่ง DirectSound จะสร้างบัฟเฟอร์ primary เป็น default ให้เลือก ตัวอย่างเช่น มีอัตราการแซมเปิลด้วยความถี่ 22 กิโลเฮิรตซ์ แอมปลิจูดขนาด 8 บิต และมี 2 channel หรือ stereo
- DSSCL_PRIORITY : เป็นระดับที่เข้าถึงฮาร์ดแวร์เสียงได้ทุกอุปกรณ์ ดังนั้นนักพัฒนาสามารถโปรแกรม และเปลี่ยนรูปแบบตัวผสานเสียง primary ได้ และยังสามารถร้องขอให้ฮาร์ดแวร์เสียงปฏิบัติการระดับสูงบนหน่วยความจำได้ เช่น การบีบอัดข้อมูล ในกรณีที่ต้องการเปลี่ยนรูปแบบข้อมูลของบัฟเฟอร์ primary
- DSSCL_EXCLUSIVE : เป็นระดับที่มีลักษณะเช่นเดียวกับ priority แต่จะไม่แสดงแอปพลิเคชั่นที่เป็น background หรือเมื่อ minimize แอปพลิเคชั่น
- DSSCL_WRITEPRIMARY : เป็นระดับ priority สูงที่สุด ซึ่งเข้าถึงและโปรแกรมตัวผสานเสียง หรือบัฟเฟอร์ primary ดังนั้นต้องมีไดรเวอร์ DirectSound ซึ่งการกำหนดระดับนี้จะไม่สร้างและใช้บัฟเฟอร์ secondary และแอปพลิเคชั่นอื่นๆ

<u>ตัวอย่าง</u> :

```
//การกำหนดระดับ cooperative
if (lpds->SetCooperativeLevel(mail_window_handle,DSSCL_NORMAL)!=DS_OK)
  { /* error setting cooperative level */ }
```

<u>เพิ่มเติม</u> : ถ้าเกิดความผิดพลาดใดๆ แอปพลิเคชั่นอื่นจะได้ใช้และควบคุมการ์ดเสียงแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6.4 การกำหนดรูปแบบบัฟเฟอร์ primary

บัฟเฟอร์ primary เป็นฮาร์ดแวร์ผสานเสียง (ตัวผสานเสียงจำลอง) บนการ์ดเสียง และทำงานตลอดการเล่นเสียง จาก default DirectSound จะสร้างบัฟเฟอร์ primary ที่มีอัตราการแซมเปิลด้วยความถี่ 22,050 กิโลเฮิรตซ์ แอมปลิจูดขนาด 8 บิต และ stereo (2 channel) หรือการกำหนดระดับ cooperative เป็น normal และถ้าต้องการเปลี่ยนรูปแบบบัฟเฟอร์ primary เพื่อให้แอปพลิเคชั่นเล่นเสียงได้อย่างมีคุณภาพมากขึ้น ตัวอย่างเช่น ให้มีอัตราการแซมเปิลแอมปลิจูดขนาด 16 บิต (เป็น default ซึ่ง DirectSound ทำงานได้ดีที่สุด) สามารถทำได้เมื่อกำหนดระดับ cooperative เป็น priority สูงที่สุด หรือระดับ DSSCL_PRIORITY เป็นต้นไป และรูปแบบบัฟเฟอร์ primary ต้องเข้ากันได้กับบัฟเฟอร์ secondary

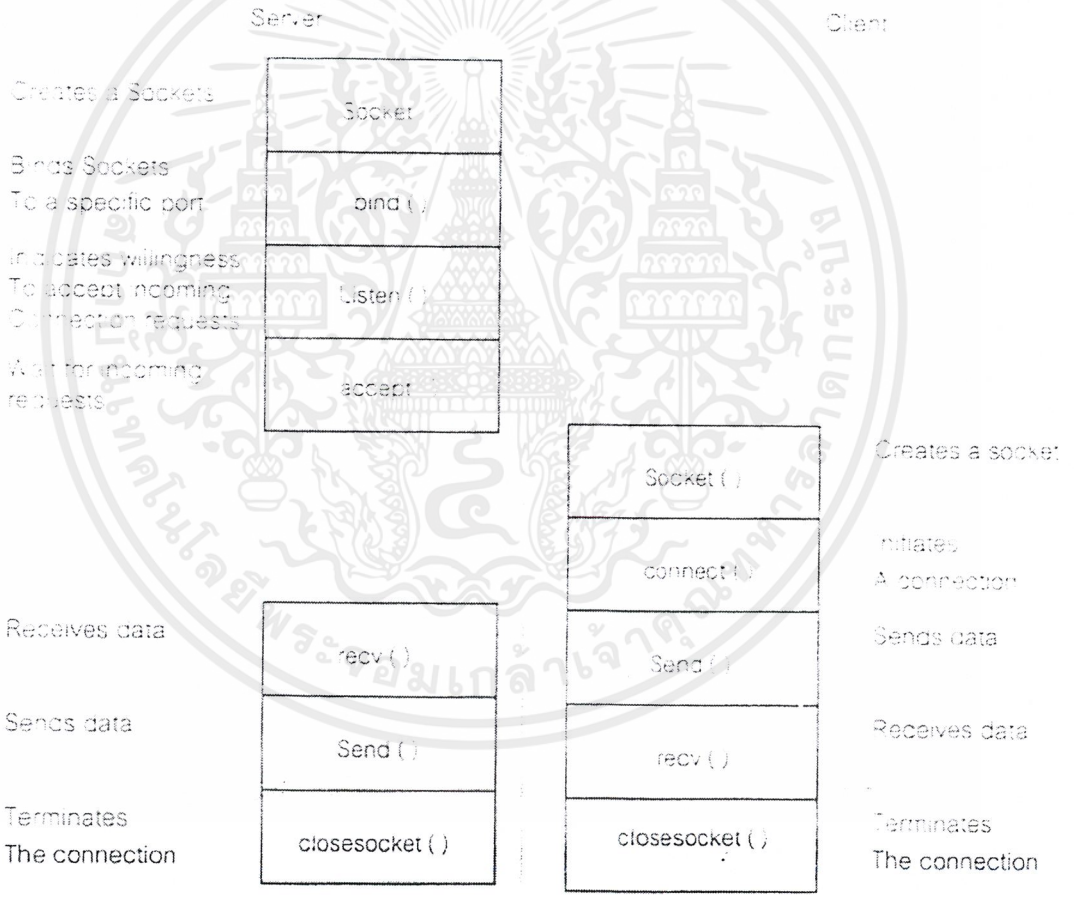
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ซอกเก็ท (Socket)

4.1 การสื่อสารโดยผ่านซอกเก็ท (Socket)

จะกล่าวถึงในกรณีของ Connection-Oriented TCP Protocol ในส่วนของเซิร์ฟเวอร์ (Server) จะมัดติดกับ TCP Port หลังจากนั้นจะใช้ Listen function เพื่อที่จะแสดงความต้องการที่จะรับข้อมูลเข้ามาจากการร้องขอการติดต่อ หลังจากที่ได้เรียก Listen function แล้วเซิร์ฟเวอร์จะเรียก Accept เพื่อที่จะรอการติดต่อเข้ามา

ในส่วนของไคลเอนต์ (Client) หลังจากได้เพิ่มซอกเก็ทแล้วก็จะเริ่มต้นการติดต่อโดยใช้ Connect และไม่มีความจำเป็นที่จะต้อง Bind Socket เพื่อที่จะกำหนดพอร์ตขึ้นมา

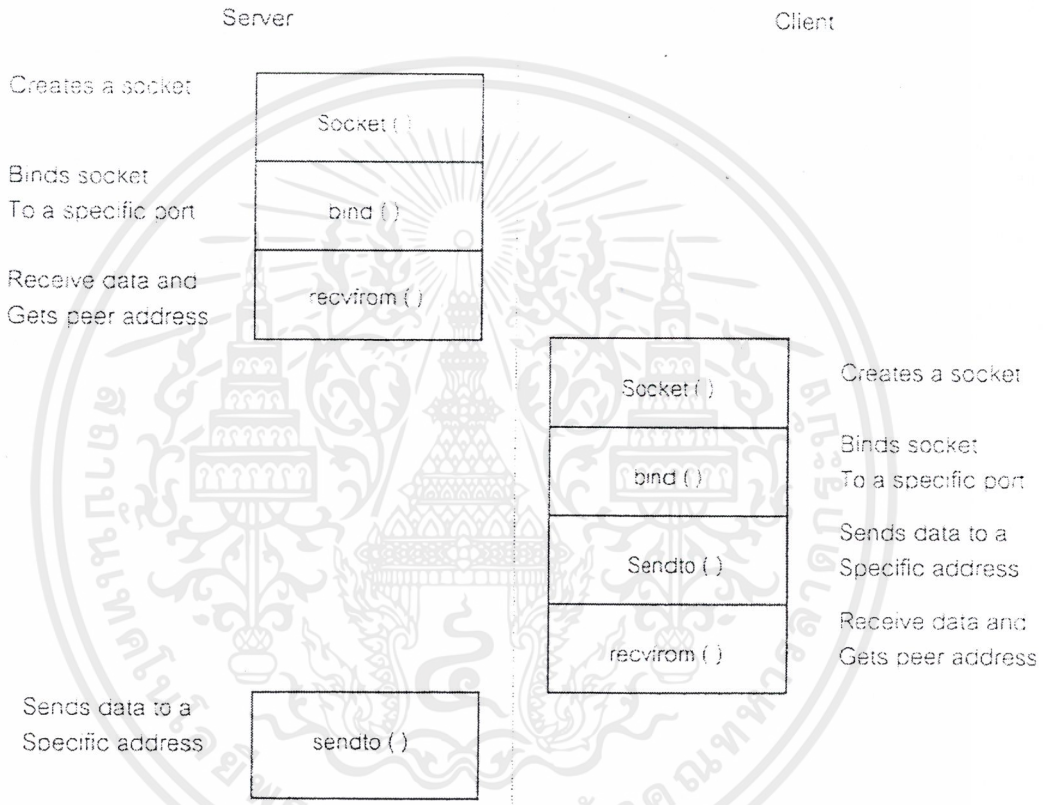


รูป 4.1 แสดง Socket connection for TCP

ในการติดต่อกันแต่ละครั้งเมื่อสำเร็จแล้ว ทั้งไคลเอนต์และเซิร์ฟเวอร์สามารถที่จะเรียกใช้ Send ในการส่งข้อมูลหรือเรียกใช้ Recv ในการรับข้อมูล และในการหยุดการติดต่อจะใช้ Close

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Socket function และถ้าเป็นกรณีของ Connectionless UDP Protocol ลำดับเหตุการณ์ที่เกิดขึ้นของไคลเอนต์ และเซิร์ฟเวอร์จะไม่เหมือนกันทุกอย่าง โดยจะเพิ่ม Respective Sockets และ bind กำหนดหมายเลขพอร์ต (Port number) ที่แตกต่างกัน คือ เซิร์ฟเวอร์ หลังจากนั้นจะเรียกไปที่ recvfrom function ซึ่งไว้สำหรับรอข้อมูลที่จะเข้ามา ไคลเอนต์จะใช้ Sendto เพื่อที่จะ Senddata ไปที่ Specific address และเมื่อข้อมูล (data) ถูกรับแล้วโดยเซิร์ฟเวอร์ และ recvfrom จะถูก return และเซิร์ฟเวอร์มักจะประกอบไปด้วยรูปแบบของที่อยู่ (address) ซึ่งจะเป็นข้อมูลที่รับเข้ามา สามารถที่จะใช้ address อันนี้ในการ Subsequent โดยการเรียก Sendto ซึ่งดูได้จากรูป 4.2



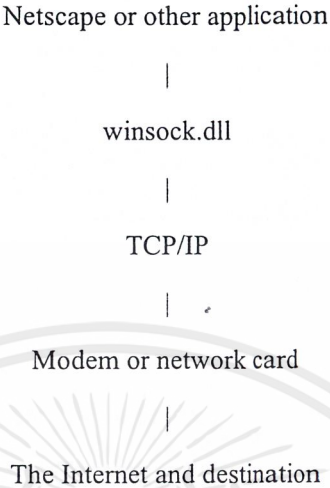
รูป 4.2 แสดง Socket connection for UDP

4.2 Winsock

เป็นเครื่องมือในการรองรับการเชื่อมต่อที่คอยรับสัญญาณอินพุตและเอาท์พุตสำหรับโปรแกรมบนอินเตอร์เน็ตบนระบบปฏิบัติการวินโดวส์ (Windows) ชื่อ Winsock เกิดขึ้นเนื่องจากเป็นการปรับปรุง Berkeley UNIX socket ให้ใช้งานได้กับวินโดวส์ ซอกเก็ท (socket) เป็นเครื่องมือที่ทำหน้าที่เป็นพอร์ต (port) ที่ใช้ในการติดต่อและแลกเปลี่ยนข้อมูลระหว่างโปรแกรม 2 โปรแกรมที่ทำงานอยู่ภายในเครื่องเดียวกันหรือผ่านระบบเน็ตเวิร์คก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Winsock จะถูกเรียกใช้งานระหว่างโปรแกรมบนเครื่องกับโปรแกรมบนระบบเน็ตเวิร์ค หรืออินเตอร์เน็ต ผ่านโพรโตคอล TCP/IP โดยมีแผนภาพการทำงานดังต่อไปนี้



Winsock สามารถจัดการกับระบบปฏิบัติการวินโดวส์ต่างเวอร์ชันกันได้ เริ่มต้นด้วยวินโดวส์ 95 ที่ Winsock ถูกบรรจุเป็นส่วนหนึ่งของระบบอยู่แล้ว ส่วนในรุ่นต่อๆ มา โปรแกรม Winsock จะต้อง install เข้าไปเอง ส่วนในระบบ UNIX ไม่ต้องการ Winsock เพราะ TCP/IP และการใช้ซอกเก็ท ถูกออกแบบมาให้ใช้งานกับโปรแกรมบน UNIX อยู่แล้ว

บริษัทต่างๆ ที่จำหน่ายและให้บริการโปรแกรม Winsock นั้นบางครั้งจะจัดเข้าเป็นชุดของโปรแกรมสำหรับโพรโตคอลบนอินเตอร์เน็ตเลย ตัวอย่างเช่น บริษัท Chameleon จำหน่ายชุดที่ประกอบด้วย Web browser, คุณสมบัติ ftp และ mail และอื่นๆ รวมทั้งโปรแกรม Winsock ด้วย นอกจากนี้ Trumpet Winsock ก็กำลังเป็นที่นิยมอยู่เช่นกัน Winsock จะทำงานเป็น dynamic link library (.DLL) บนวินโดวส์ นั่นคือเวลาที่โปรแกรมที่ใช้อยู่ต้องการเท่านั้นจึงเรียกออกมาใช้ แต่จะไม่รวมอยู่เป็นส่วนหนึ่งของโปรแกรม

เมื่อมีการติดต่อกับอินเตอร์เน็ต อาจต้องตรวจสอบดูด้วยว่าเครื่องของเรานั้นมี Winsock เวอร์ชันใดอยู่ สามารถทำงานกับโปรแกรมที่นำมาได้หรือไม่ ถ้าโปรแกรมต้องการ Winsock เวอร์ชันที่ไม่ตรงกัน จะต้องเอาเวอร์ชันใดเวอร์ชันหนึ่งออกก่อน จึงจะใช้งานได้

4.3 The Winsock API

ในการอินเตอร์เฟสด้วย Berkeley Sockets สามารถที่จะใช้ในการติดต่อสื่อสารโดยใช้ Connection-Oriented Protocol (TCP) และ Connectionless Protocol (UDP) การเขียนโปรแกรมจะเป็นแบบไคลเอนต์-เซิร์ฟเวอร์ (Client-Server); เซิร์ฟเวอร์จะรอเมื่อมีการร้องขอเข้ามา ระหว่างนั้นไคลเอนต์จะเริ่มทำการพิจารณาการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะกล่าวถึงความแตกต่างระหว่างเครื่องมือในการทำงานระหว่าง Winsock และ Unix เวอร์ชันของ Berkeley Sockets ส่วนมากจะแสดงให้เห็นว่าแท้จริงแล้วลักษณะของซอกเก็ท (Sockets) และไฟล์ (Files) ไม่สามารถที่จะใช้คุณสมบัติแทนกันได้ ในที่นี้จะทำให้เห็นผลเมื่อมีการกำหนดพอร์ตของ application ของ equivalence นี้

ความแตกต่างอีกอย่าง คือ Winsock Library ต้องการการกำหนดค่าเริ่มต้น Application ใดที่จะใช้ Winsock function ต้องทำการเรียก WSAstartup function และก็จะทำงานที่ Winsock Library และเมื่อสิ้นสุดการทำงานจะทำการเรียก WSAcleanup สำหรับการ termination อย่างสมบูรณ์

4.4 Winsock Initialization

เมื่อ Winsock Library ถูกเริ่มต้นการทำงานด้วยการเรียก WSAstartup Application จะทำการเรียก function นี้แล้วให้ address ไปที่ WSAdata Structure ซึ่งจะเป็นตัวแปรสำหรับเก็บค่า Initialization information

ตลอดเวลาที่ผ่านมาเราเรียก WSAstartup การประยุกต์ใช้งานต่างๆ จะไม่สนใจเวอร์ชันของ Winsock Library ในการ Initialization จะไม่เกิดความผิดพลาด ถ้าไม่เกิดการ Overlap ระหว่าง Version number ที่ถูก Supported โดย Application และ Version ที่ Supported โดย Winsock Library

ถ้ามีความผิดพลาดเกิดขึ้น WSAstartup จะส่งค่ากลับที่ไม่ใช่ศูนย์ ซึ่ง Application แต่ละตัวสามารถที่จะดู Information ของ Error ที่เกิดขึ้นได้จาก WSAgetlastError

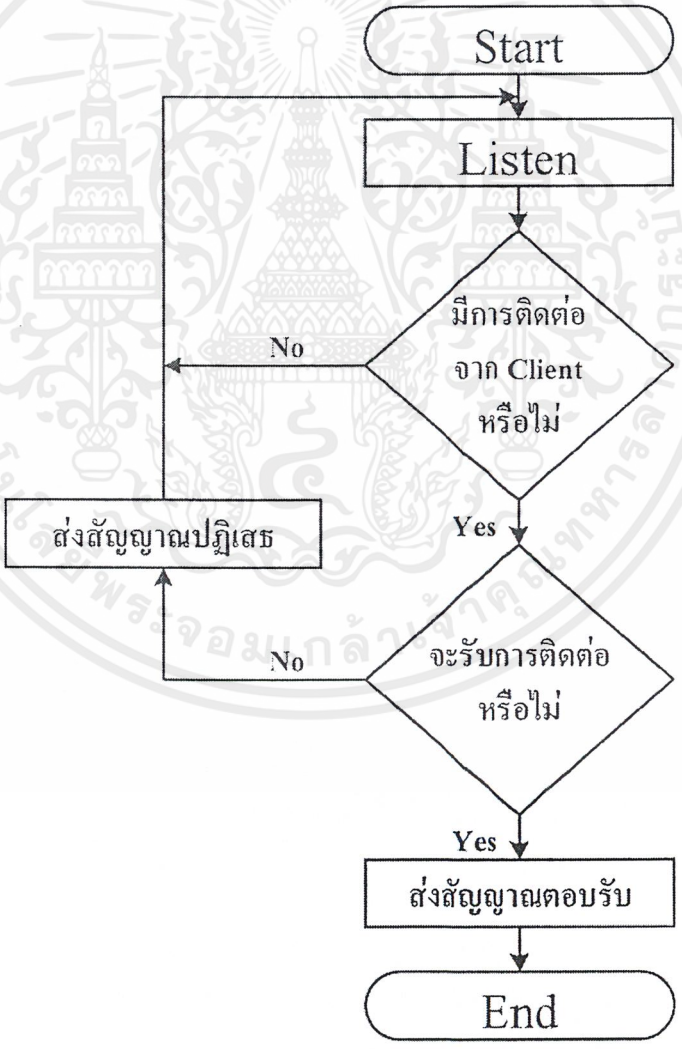
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โปรแกรมการประชุมทางไกลด้วยกล้องวิดีโอหรือ VMeeting

5.1 หลักการทำงานของโปรแกรมการประชุมทางไกลด้วยกล้องวิดีโอ หรือ VMeeting

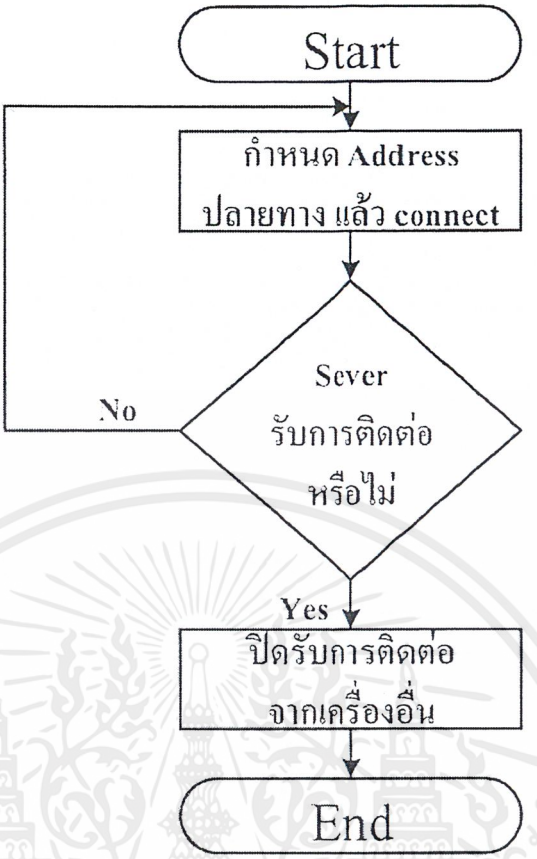
โปรแกรมการประชุมทางไกลด้วยกล้องวิดีโอ หรือ VMeeting นี้ จะติดต่อกันผ่านโพรโตคอล TCP/IP โดยมีการกำหนด address ของจุดหมายปลายทาง โดยแบ่งการติดต่อออกเป็น 2 ช่วงคือ

1. ช่วงเริ่มต้นสื่อสาร หรือในช่วงที่ผู้ใช้ทั้ง 2 เครื่องเริ่มติดต่อกัน ชนิดของการสื่อสารจะเป็นทางเดียว (half-duplex) แบบ connection-oriented communication โดยการทำงานของซอกเก็ท (socket) ในช่วงนี้ จะอธิบายได้ดังรูป 5.1 และ รูป 5.2



รูป 5.1 เซิร์ฟเวอร์ซอกเก็ท (Server Socket)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

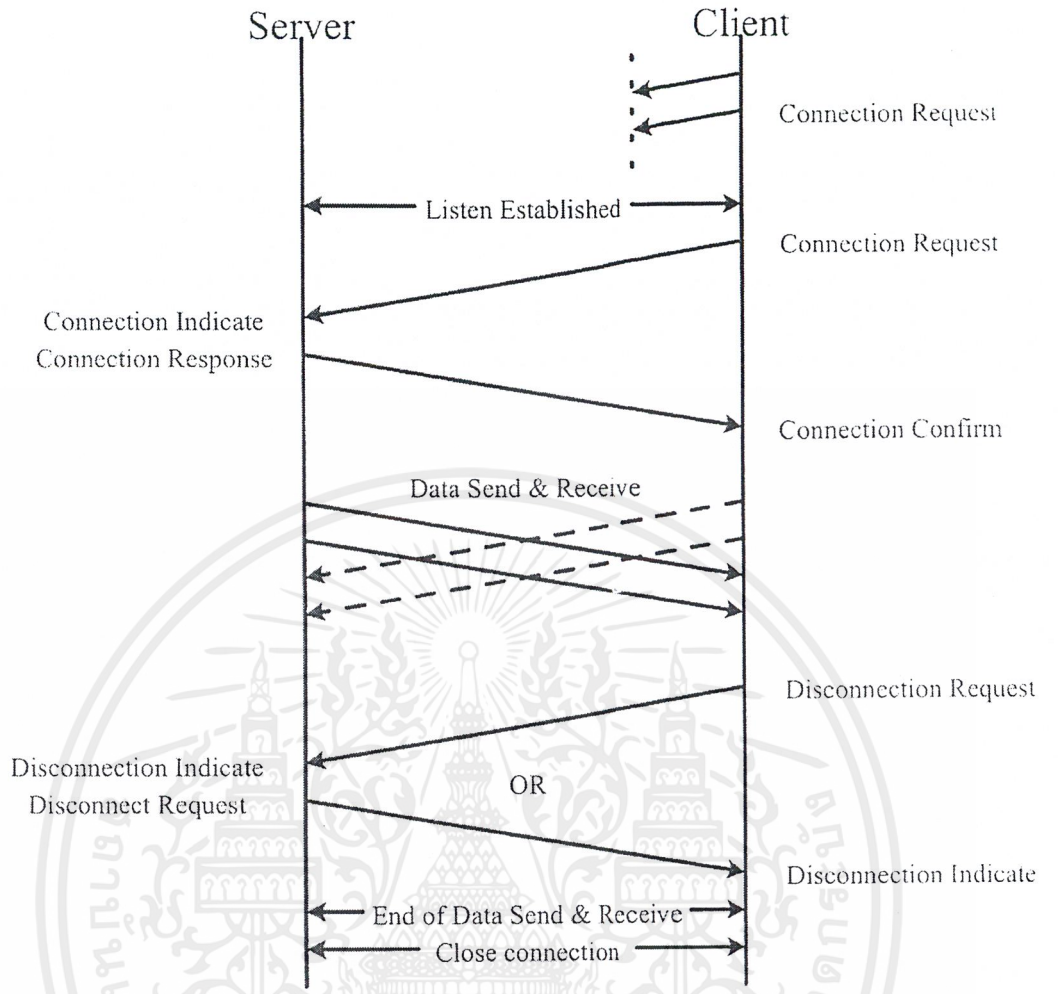


รูป 5.2 ไคลเอนต์ซอกเก็ท (Client Socket)

2. เมื่อการเชื่อมต่อเสร็จสิ้นลง ข้อมูลภาพและเสียงจะถูกส่งและรับจากเครื่องของผู้ใช้ทันที โดยแบ่งออกเป็น 3 พอร์ต ให้แต่ละพอร์ตดูแลข้อมูลแต่ละอย่าง ซึ่งได้แก่ คำสั่ง (Command) ข้อมูลภาพ(Image) และข้อมูลเสียง(Sound) โดยทั้งหมดจะทำงานแบบ full-duplex โดยใช้การส่งแบบ asynchronous ทั้งหมด และจะหยุดการส่งข้อมูลเมื่อผู้ใช้หยุดการติดต่อเท่านั้น โดยแผนภาพรวมของการทำงานทั้งหมด แสดงได้ดังรูป 5.3

การทำให้โปรแกรมตัวเดียวทำงานเป็นทั้งไคลเอนต์ (client) และ เซิร์ฟเวอร์ (server) ได้นั้น ทำได้โดยกำหนดลักษณะการทำงานของไคลเอนต์ และเซิร์ฟเวอร์ไว้ในเครื่องเดียวกันแต่ให้ทำงานสลับกัน โดยที่ฝ่ายเริ่มติดต่อก่อนจะมีการทำงานของไคลเอนต์และจะปิดการทำงานแบบเซิร์ฟเวอร์ไว้ เพื่อติดต่อไปยังอีกฟากที่เปิดเซิร์ฟเวอร์รออยู่ การทำงานทั้งหมดจะเกิดขึ้นโดยอัตโนมัติ โดยที่ผู้ใช้ไม่ต้องกำหนดเอง และเมื่อการสื่อสารปิดลงโปรแกรมจะเริ่มทำงานใหม่ โดยใช้ทั้ง 2 ลักษณะเช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 5.3 แสดง Transmission Protocol

5.2 การเขียนโปรแกรม VMeeting

ในการเขียนโปรแกรมนี้ ใช้ภาษา Delphi เวอร์ชัน 5.0 ในการเขียน โดยใช้คลาส (class) ในการเขียนดังต่อไปนี้

1. TServerSocket และ TclientSocket
2. TcustomWinSocket
3. TmemoryStream
4. TaudioStream
5. TwaveStream
6. TsoundCaptureStream

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ component อีก 1 ชุดคือ MMTools ซึ่งรายละเอียดของคลาสต่างๆมีดังนี้

1. TServerSocket และ TclientSocket

เป็นคลาสที่มี procedure และ function สำหรับการเชื่อมต่อผ่านโพรโตคอล TCP/IP โดยทำหน้าที่เป็นผู้ส่ง(client) และผู้รับ(server) แต่ละเครื่องต้องกำหนดพอร์ตให้ตรงกัน จึงจะติดต่อกันได้

สำหรับในโปรเจคนี้ โปรแกรมตัวหนึ่งจะมีการกำหนดให้เป็นทั้งไคลเอนต์ และ เซิร์ฟเวอร์ และอย่างที่ได้กล่าวไว้ว่า เมื่อเครื่องใดเครื่องหนึ่งทำตัวเป็นเซิร์ฟเวอร์ ก็จะไม่เป็นไคลเอนต์ การเริ่มติดต่อจะใช้คำสั่ง Server.Open เพื่อให้เซิร์ฟเวอร์รับ request จากไคลเอนต์ได้ เมื่อไคลเอนต์ต้องการติดต่อ ก็จะใช้คำสั่ง Client.Open

2. TcustomWinSocket

เป็นคลาสที่มี procedure และ function ที่ใช้ในการส่งข้อมูลและรับข้อมูล รวมทั้งใช้ตรวจสอบสถานะได้ สามารกำหนดเองหรือใช้การอ้างอิงมาจากคลาส TServerSocket หรือ TClientSocket ก็ได้

ในโปรเจคนี้ได้กำหนดคลาสเพื่อการรับส่งข้อมูล ทั้งหมด 3 คู่ คือ สำหรับข้อมูลเสียง ภาพ และคำสั่ง ในการส่งภาพและเสียงจะเก็บข้อมูลแบบ Stream และส่งข้อมูลโดยใช้อาร์เรย์ (array) ของข้อมูลประเภทไบต์ (byte) ส่วนคำสั่งจะใช้ข้อมูลเป็นตัวเลข โดยใช้คำสั่ง socket.sendbuf (buffer,count)

3. TmemoryStream

เป็นคลาสที่ใช้เก็บข้อมูลในรูปของ stream ในโปรเจคนี้ใช้ในการเก็บข้อมูลภาพและเสียง หลังจากเก็บข้อมูลแล้วและก่อนจะนำข้อมูลไปแสดงผล โดยตัดเป็นก้อนข้อมูลก่อนส่งไปตามสาย

อีก 3 คลาสที่จะกล่าวต่อไป เป็นคลาสที่แปลงมาจาก code ฟังก์ชันต่างๆของ DirectSound ในภาษา C++ มาแปลงเป็น code ในภาษา Delphi โดยมีการทำงานตามที่จะกล่าวต่อไปนี้

4. TaudioStream

เป็นการผลิต Wave Stream ออกมาโดยใช้ DirectSound ผ่านข้อมูลออกมาที่บัฟเฟอร์ (buffer) และไหลผ่านออกไปอย่างต่อเนื่อง และใช้คุณสมบัติที่เรียกว่า BufferLength ในการกำหนดความยาวของบัฟเฟอร์

การเล่นข้อมูลในรูปแบบ Wave Stream จะต้องกำหนดคุณสมบัติของ Wave Stream นั้นด้วย โดยอาจนำไปเทียบกับคลาส TWaveStream เลยก็ได้ การทำงานของคลาสนี้จะเป็นแบบวงกลม ดังนั้นเพื่อที่จะเล่นแบบต่อเนื่อง ต้องกำหนด property ให้ Loop:=True ด้วย

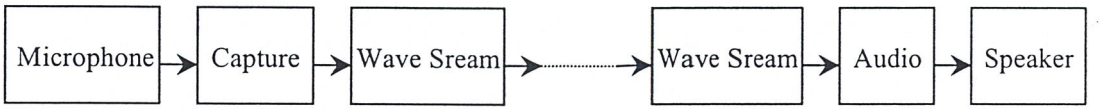
นอกจากนี้การเล่นจาก Wav.file ก็สามารถทำได้เช่นกัน

5. TwaveStream

เป็นคลาสที่ใช้จัดการกับข้อมูลประเภท Wave Stream การ create คลาสนี้ต้องอ้างถึง Stream ที่จะนำมาใช้ โดยสามารถกำหนดได้ว่าจะเป็นการบันทึกลงไปใน Stream หรือเป็นการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Stream โดยควรกำหนดคุณสมบัติต่างๆก่อนสั่งให้ทำงาน เราใช้คลาสนี้เป็นตัวกลางในการจับข้อมูลให้เป็นก้อน ข้อมูลที่สามารถตัดส่งไปได้



6. TsoundCaptureStream

เป็นการจับสัญญาณ Wave Stream เข้ามาที่บัฟเฟอร์ โดยแปลงมาจาก IDirectSoundCapture 8 ของ DirectSound จะทำการดักข้อมูลทุกจุดที่เข้ามาเก็บไว้ที่ buffer ตามขนาดที่ระบุไว้ในคุณสมบัติชื่อ Buffer Length และเก็บข้อมูลในรูป wave steam รูปแบบของข้อมูลที่ถูกดักจับได้ถูกกำหนดโดยคุณสมบัติ CaptureFormat โดยรายชื่อของรูปแบบของข้อมูลนั้นจะแสดงอยู่ที่คุณสมบัติชื่อ SupportedFormat ข้อมูลของรูปแบบจะมี 3 ส่วน คือ SamplesPerSec, BitsPerSample และ Channel ซึ่งจะเป็นตัวกำหนดขนาดของข้อมูล Wave Stream ที่จับได้ และการนำข้อมูลไปใช้ต่อนั้น ส่วนที่นำไปใช้ต้องกำหนดรูปแบบของข้อมูลให้เป็นแบบเดียวกันด้วย

ในส่วนของการแสดงผลที่เป็นภาพ เราใช้คอมโพเนนท์ที่ชื่อว่า Mmtools ซึ่งเป็น Shareware ที่มีความสามารถหลากหลายมาก สาเหตุที่เลือกใช้คอมโพเนนท์ตัวนี้เนื่องจาก

1. มี Event ที่เกิดขึ้นเมื่อกำลังเก็บภาพลงไปในบัฟเฟอร์ ก่อนจะแสดงทางหน้าจอ ทำให้สามารถใช้ event นี้ส่งข้อมูลผ่านเครือข่ายได้
2. มีระบบการจัดการภาพที่ดี คือ สามารถเลือกรูปแบบ format ของภาพได้ เลือกอุปกรณ์ที่ใช้ในการจับภาพได้ เพราะอุปกรณ์ต่างกันจะให้ภาพต่างกัน รวมทั้งยังปรับระดับสี ความคมชัด และคุณภาพของภาพได้ การปรับขนาดของภาพให้เล็กลงจะทำให้ข้อมูลที่ใช้ส่งลดลงได้ตามความเหมาะสม
3. สามารถส่งข้อมูลที่จับโดยใช้การ์ดวิดีโอได้ ซึ่งเป็นข้อเปรียบเทียบกับการส่งแบบแยกส่ง ดังที่ผู้จัดได้เขียนขึ้น
4. ใช้คำสั่งที่ง่ายต่อความเข้าใจ สามารถนำมาประยุกต์ได้ง่าย

แต่ข้อเสียที่พบ คือ คำสั่งบางอย่างใช้ง่ายเกินไป ทำให้การควบคุมทำได้ยาก แต่ก็สามารถประยุกต์มาใช้ง่ายให้ตรงกับวัตถุประสงค์ได้

5.3 หลักการในการทำงานในส่วนแสดงผลด้วยภาพ

เนื่องจากถ้าหากว่ามีการแสดงผลด้วยภาพ การทำงานโดยรวมของโปรแกรมจะช้าลงมาก จึงกำหนดให้มีเมนูที่สามารถสั่งให้แสดงผลด้านภาพอยู่ด้วย เมื่อไม่ต้องการแสดงภาพก็สามารถปิด

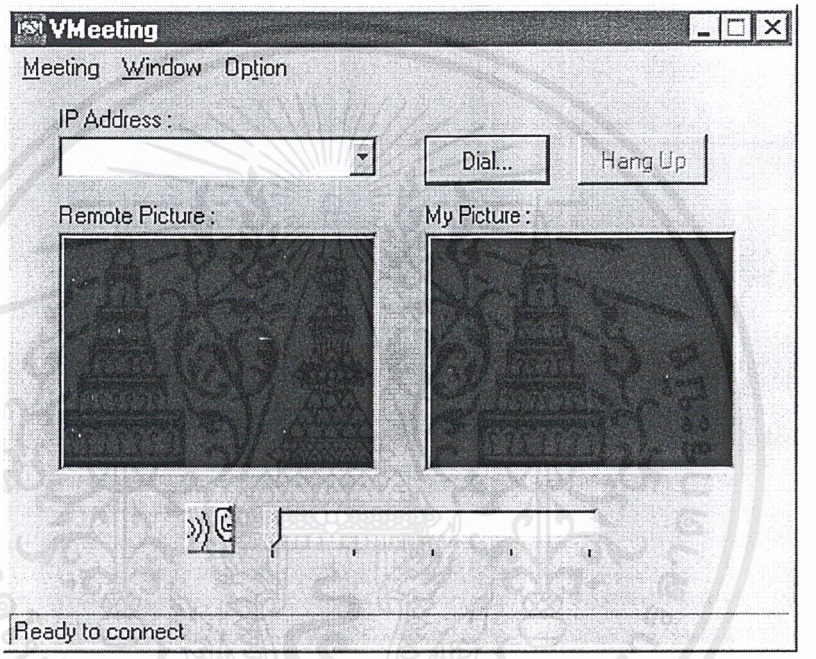
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ ความเร็วของโปรแกรมก็จะเพิ่มขึ้น การแสดงผลหรือหยุดจะใช้คำสั่ง MMServer1.OpenVideo (DriverID); และ MMServer1.Close; ส่วนในด้านของการส่งข้อมูลจะทำให้ event : MMServer1Stream โดยวิธีการเดียวกับการส่งข้อมูลเสียง และในตอนรับก็จะใช้ซอกเก็ท (Socket) รับแบบธรรมดา และเมื่อจะนำไปแสดงผลจะใช้คำสั่ง MMClient1.ProcessData (TmemoryStream)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การใช้งานโปรแกรม VMeeting

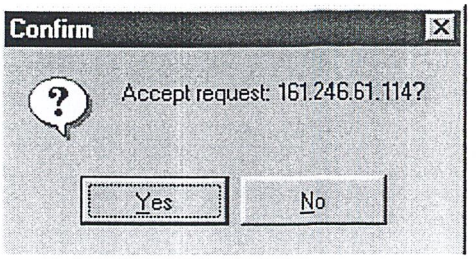
ซอฟต์แวร์ที่ถูกเขียนขึ้นมาในการทำโปรเจ็คนี้มีชื่อว่า "VMeeting" เมื่อเปิดขึ้นมาจะมีหน้าจอสำหรับแสดงภาพ 2 หน้าจอ และเมนูบาร์ (Menu bar) ด้านบน โดยมีอยู่ 3 เมนู คือ Meeting, Window และ Option แต่จะใช้งานได้เพียง Menu เดียว คือ Meeting โดยต้อง log in โดยพิมพ์ชื่อและรหัสผ่านก่อนจึงจะใช้ได้ หน้าจอแสดงผลของโปรแกรมนี้ แสดงไว้ในรูป 6.1



รูป 6.1 แสดงรูปโปรแกรม

การ log in เป็นการบอกให้ Server Socket ทำงานเพื่อรอรับ request จากไคลเอนต์ และไคลเอนต์ก็สามารถติดต่อกับเซิร์ฟเวอร์ได้โดยพิมพ์ IP Address ที่ต้องการติดต่อ แล้วกด Dial เพื่อเชื่อมต่อ จากนั้นฝั่งเซิร์ฟเวอร์เมื่อได้รับการติดต่อแล้ว จะปรากฏหน้าจอดังรูป 6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

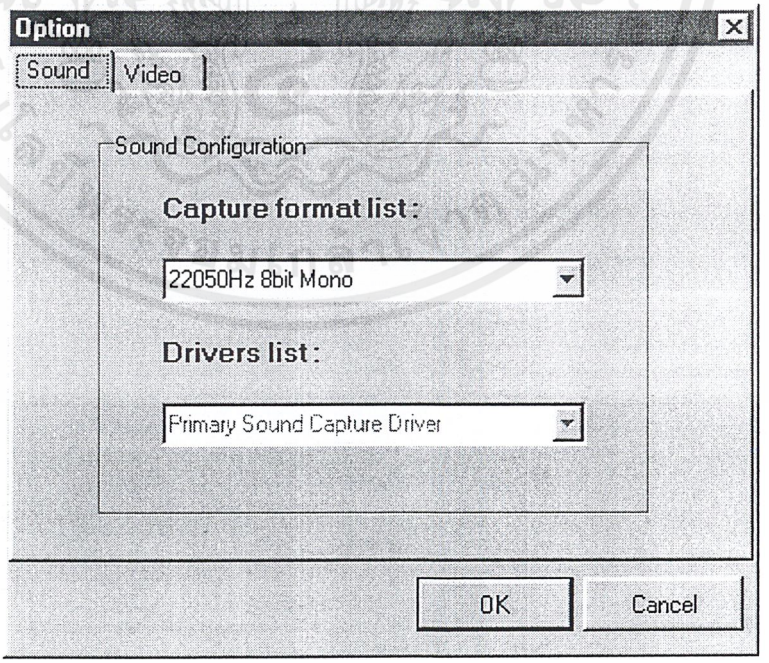


รูป 6.2 แสดงหน้าจอเมื่อเซิร์ฟเวอร์ได้รับการติดต่อ

ถ้าเซิร์ฟเวอร์รับการติดต่อ เมนูด้านบนที่เหลือก็จะให้ใช้ได้ และเมื่อต้องการจบการสนทนาก็เพียงกด Hang Up การติดต่อก็จะยุติทันที และจะสามารถติดต่อใหม่ได้เลยเช่นกัน

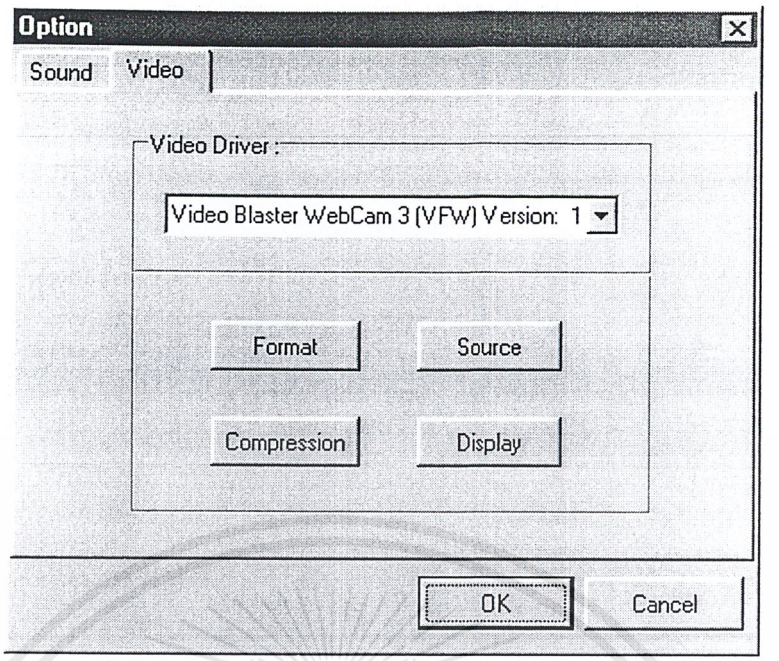
รายละเอียดภายในเมนูต่างๆมีดังนี้

1. Meeting ใช้เริ่มต้นและสิ้นสุดการสื่อสาร มี 3 เมนูย่อยดังนี้
   - Login เป็นการขอเข้าไปใช้ในโปรแกรมนี้
   - Logout ขอออกจากการใช้โปรแกรม
   - Exit ปิดโปรแกรม
2. Window ใช้แสดงผลและใช้เซ็ตค่าที่ใช้แสดง มี 2 เมนูย่อยดังนี้
   - Show Image ใช้ในการเปิด/ปิดภาพของฝ่ายตรงข้าม
   - Configuration ใช้ปรับแต่งค่าต่างๆของภาพและเสียง มีหน้าจอดังนี้



รูป 6.3 ส่วนการปรับแต่งค่า Sound

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



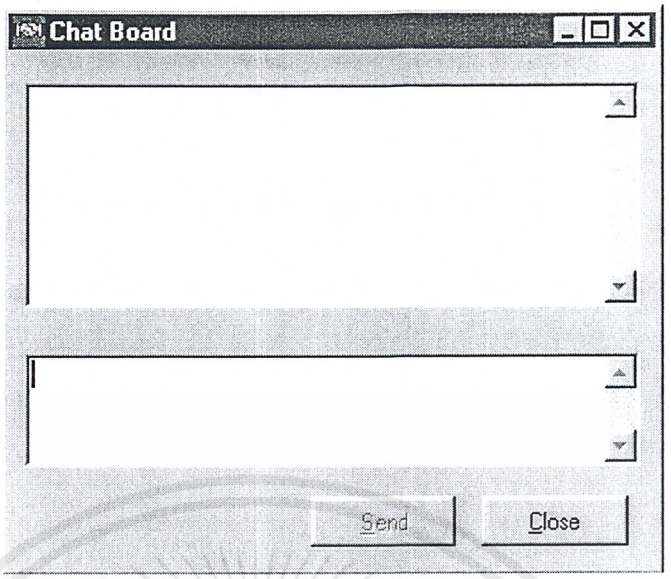
รูป 6.4 ส่วนการปรับแต่งค่า Video

ในส่วนของ Configuration นี้ แบ่งออกเป็น 2 ส่วน ได้แก่ Sound หรือเสียง โดยสามารถเลือก format ที่ต้องการได้ใน combo box แรก ส่วนใน combo box ที่ 2 จะแสดงว่าใช้การ์ดเสียงหลัก เพราะโดยทั่วไป 1 เครื่องจะมีการ์ดเสียง 1 อันเท่านั้น

อีกส่วนหนึ่งในเมนู Configuration นั้น ได้แก่ Video จะมี combo box แสดงอุปกรณ์ และปุ่มอีก 4 ปุ่ม ได้แก่ Format ที่ใช้ปรับขนาดของภาพได้, Source ใช้ปรับแต่งภาพให้เป็นตามที่ต้องการ, Compression ใช้เปลี่ยนรูปแบบการบีบอัดข้อมูล และปรับคุณภาพของภาพได้ และ Display คือ แสดงภาพจากการปรับค่าต่างๆ

3. Option เป็นส่วนบริการเพิ่มเติมนอกจากภาพและเสียง มี 1 เมนูย่อยดังนี้
   - Chat board ถ้าไม่ได้ติดตั้งการ์ดเสียง และกล้องดิจิตอลไว้ก็ยังสามารถใช้ chat board นี้ในการติดต่อสื่อสารได้ มีหน้าจอดังรูป 6.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



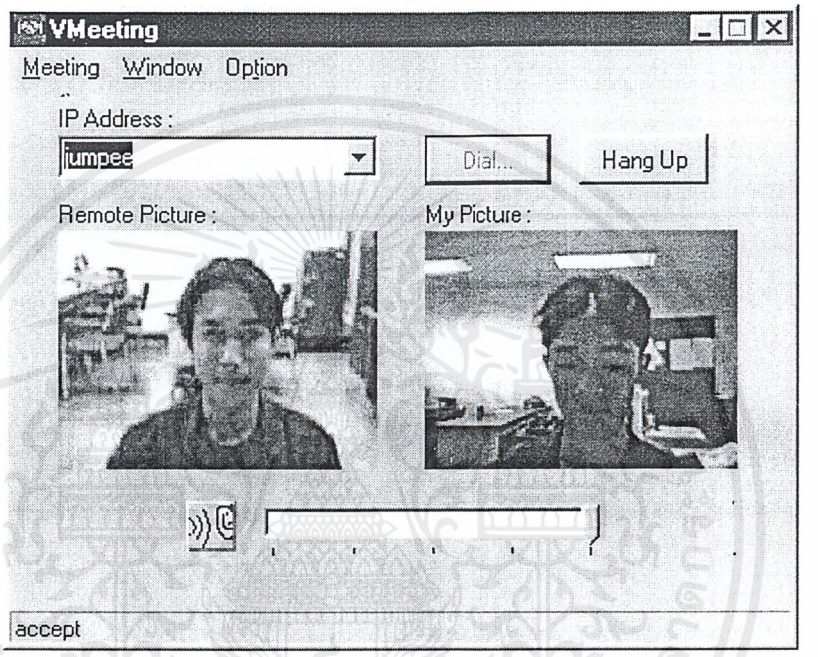
รูป 6.5 แสดงหน้าจอ Chat board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
ผลการทดลองและสรุปผล

7.1 ผลการทดลอง

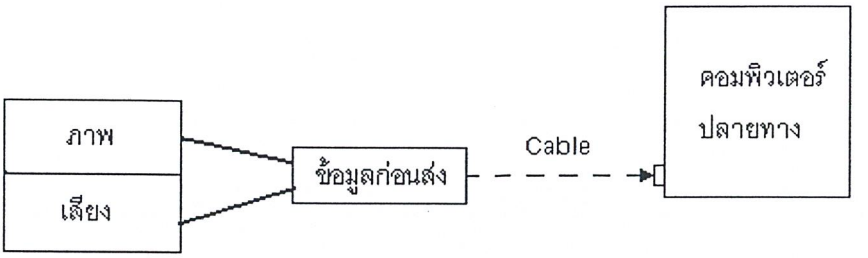
ในการทดลองเราได้นำโปรแกรมที่สำเร็จแล้วนั้นมาแก้ไขให้เป็น 2 โปรแกรมที่แตกต่างกัน โดยภาพที่ถูกจับออกมาจะใช้วิธี capture แบบเดียวกัน จะแสดงรูปของการทดลองได้ดังต่อไปนี้



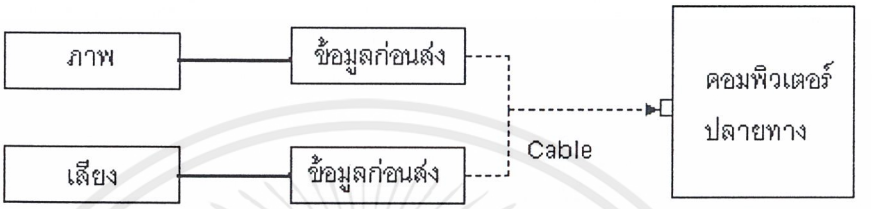
รูป 7.1 หน้าจอที่เกิดจากการติดต่อแล้ว

ส่วนการจับข้อมูลเสียงและการส่งข้อมูลจะใช้วิธีที่ต่างกัน 2 วิธี คือวิธีแรกจะใช้การจับสัญญาณภาพและเสียงลงในข้อมูลก้อนเดียวกันและส่งไปพร้อมๆกันในพอร์ตเดียว และวิธีที่ 2 ใช้การจับภาพและเสียงโดยแยกข้อมูลเป็นคนละชุดและส่งข้อมูลออกไปพร้อมกันแต่แยกพอร์ตในการส่ง อธิบายความแตกต่างให้เห็นได้ดังรูป 7.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก)



ข)

รูป 7.2 การทดลองในแบบต่างๆ

ก) ส่งข้อมูลภาพและเสียงรวมกัน ข) ส่งข้อมูลภาพและเสียงแยกกันเป็น 2 ชุด

ในการทดลองแบบ ก) เป็นการจับข้อมูลเสียงออกมาพร้อมกับข้อมูลภาพเสมือนใช้แหล่งที่มาของข้อมูลต่างกันแต่นำมารวมกันแล้วส่งข้อมูลนั้นออกไป ส่วนในแบบ ข) เป็นการจับข้อมูลภาพจากกล้องและจับข้อมูลเสียงจากการ์ดเสียงแยกจากกัน แล้วทำเป็นก้อนข้อมูล 2 ก้อนก่อนส่ง จากนั้นจึงค่อยส่งข้อมูลไปสู่ผู้ใช้ด้านตรงข้ามโดยแยกส่วนรับข้อมูลด้วย ในที่นี้ไม่ได้มีการบีบอัดข้อมูลเสียงแต่อย่างใด ผลที่ได้รับจากการทดลองมีดังต่อไปนี้

1. เมื่อทดลองให้ส่งและรับในเครื่องเดียวกันเองปรากฏว่าข้อมูลเสียงมีการหน่วงเวลาหลังจากที่พูดออกไปทั้ง 2 แบบแต่แบบ ก) จะมีการหน่วงของการแสดงผลมากกว่าแบบ ข)

2. เมื่อมีการส่งข้ามเครือข่าย LAN ผลที่ได้รับคือภาพที่ได้รับเกิดการกระตุกอยู่เล็กน้อย ส่วนเสียงที่แสดงออกมาปรากฏว่าในแบบ ก)ข้อมูลเสียงจะช้ากว่าข้อมูลภาพประมาณ 1 – 3 วินาที แต่ในแบบ ข) ข้อมูลที่ส่งมาบางครั้งช้ากว่าหรือเร็วกว่าก็มี แต่อยู่ในช่วงเวลาแตกต่างที่ไม่มากเท่าแบบ ก) อาจยอมรับได้ว่าแสดงผลข้อมูลออกมาเกือบพร้อมกัน

3. คุณภาพเสียงที่ออกมาในแบบ ก) จะดีกว่าแบบ ข) เนื่องจากการใช้ DirectSound เป็นการติดต่อกับการ์ดเสียงโดยตรง ทำให้เสียงที่จับได้แสดงผลออกมาค่อยข้างเร็วจึงเป็นผลทำให้การจัดการการแสดงผลด้านเสียงค่อนข้างทำได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 สรุปผลการทดลอง

จากผลการทดลองจะเห็นว่าการส่งข้อมูลโดยแยกกันส่งและกำหนดขนาดการรับส่งข้อมูลที่พอเหมาะ รวมทั้งสภาพแวดล้อมที่มีผลกระทบต่างๆ เช่น ระยะทางการส่งข้อมูล หรือขนาดของข้อมูลที่สายส่งสามารถรับรองได้ เป็นต้น ทำให้ข้อมูลที่ออกมาทั้งภาพและเสียงตรงกันได้ แต่คุณภาพของเสียงที่ออกมายังสู้กับที่ได้จาก Shareware หรือโปรแกรมที่ใช้กันทั่วไปอย่าง NetMeeting ไม่ได้ เนื่องจากในโปรแกรมเหล่านั้นมีการพัฒนาอย่างต่อเนื่องจนสามารถกำหนดค่าต่างๆ ได้อย่างถูกต้อง จึงได้เสียงที่มีคุณภาพดีกว่า ถ้าโปรแกรมนี้มีผู้พัฒนาต่อเนื่องอาจทำให้คุณภาพดีขึ้นกว่านี้ได้ แต่ผลที่ได้รับก็ตรงตามจุดประสงค์ในการส่งภาพและเสียงพร้อมๆ กัน นอกจากนี้ในโปรแกรมที่พัฒนาขึ้นใหม่นี้ ได้เพิ่มส่วนอื่นๆ สำหรับอำนวยความสะดวกในการใช้เข้าไปด้วย ตัวอย่างเช่น Chat Board สำหรับผู้ที่ต้องการใช้โปรแกรมแต่ไม่มีกล้องวิดีโอ หรือการ์ดเสียงที่บันทึกเสียงได้ หรือถึงแม้ใช้งานฟังก์ชันดังกล่าวได้ แต่ต้องการความแน่นอนก็สามารถพิมพ์เป็นตัวอักษรส่งถึงกันได้

7.3 ข้อจำกัดของระบบ

สำหรับโปรแกรมในครั้งนี้ ได้ใช้ความสามารถของ DirectSound ในการจับข้อมูลเสียงมาและเป็นโปรแกรมจับภาพที่ติดต่อกับกล้องผ่านการ์ดวิดีโอ จึงมีข้อกำหนดของระบบดังต่อไปนี้

1. เป็นคอมพิวเตอร์ที่ลง DirectX ไว้แล้ว หรือมีระบบปฏิบัติการที่รองรับการใช้ DirectX ได้
2. มีการ์ดวิดีโอในกรณีที่ในตัวกล้องไม่ได้ถูกบรรจุไว้แล้ว หรือใช้กล้องที่มีการ์ดวิดีโอในตัว
3. การ์ดเสียงต้องรองรับ wave table ซึ่งเป็นส่วนช่วยในการจับข้อมูลเสียงได้
4. คอมพิวเตอร์ที่ใช้ควรมีลำโพงและไมโครโฟนในกรณีที่ต้องการใช้เสียง แต่ถ้าไม่มีก็สามารถใช้การพิมพ์ข้อความถึงกันก็ได้

7.4 สมรรถภาพของระบบ

1. การติดต่อภายในวงเครือข่ายเดียวกันทำได้แน่นอน แต่ถ้าข้าม Router อาจเกิดปัญหาได้
2. ในระยะทางที่ไกลขึ้น delay ของภาพและเสียงจะต่างกันมากขึ้น ภาพและเสียงที่ได้รับจะแย่ลง
3. ช่วงความถี่ที่รับได้เป็นค่าความถี่ที่เป็นรูปแบบของ DirectSound จึงแบ่งออกเป็น 3 อย่าง คือ SamplesPerSec ตั้งแต่ 8 k ถึง 48 k, BitsPerSample ที่เป็น 8 bit ถึง 16 bit และมี Channels คือ mono กับ stereo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 ความผิดพลาดที่เกิดขึ้นในการทำงาน

ในการระหว่างการทำงานในครั้งนี้มีข้อจำกัดบางอย่างดังต่อไปนี้ เพื่อเป็นแนวทางสำหรับผู้ที่สนใจในการศึกษาต่อไป

1. เนื่องจากคอมพิวเตอร์ที่ใช้งานมีอุปกรณ์ไม่ครบทั้งหมดในเครื่องเดียวทำให้การทำงานค่อนข้างวุ่นวายเพราะต้องสลับสับเปลี่ยนคอมพิวเตอร์ใช้งานในระหว่างการเขียนโปรแกรมอยู่ตลอดเวลา

2. เนื่องจาก Object ทางด้านภาพที่นำมาใช้ เป็นShareware ทำให้ไม่สามารถควบคุมการใช้งานได้อย่างเต็มประสิทธิภาพ แต่ก็เป็นอุปกรณ์ที่มีส่วนช่วยให้เกิดการเปรียบเทียบในการทดลองได้

3. ในการเชื่อมต่อของเครือข่ายยังมีข้อผิดพลาดบางประการที่ไม่สามารถจัดการได้ ถึงแม้ว่ามีการรองรับความผิดพลาดในส่วนของ Event : OnSocketError แล้วก็ตาม

7.6 สิ่งที่ได้รับจากการทำโครงงาน

โปรแกรมที่ได้พัฒนาขึ้นมาในโครงงานครั้งนี้ มีอุปสรรคในการทำงานหลายอย่างมาก ตั้งแต่วิธีการเลือกสรรภาษาที่จะใช้ในการเขียนโปรแกรม อุปกรณ์และฟังก์ชันที่จะใช้ในการส่งข้อมูลทางเลือกต่างๆ มีอยู่มากมายแต่อุปกรณ์ที่เหมาะสมที่สุดที่จะใช้ในงานงานหนึ่งมีได้เพียงอย่างเดียว อุปกรณ์ที่เลือกใช้ได้แก่ DirectX ในเฉพาะส่วนของ DirectSound เท่านั้น การศึกษาในส่วนนี้ทำให้เข้าใจการทำงานของ DirectSound เพื่อนำไปใช้ประโยชน์ต่อไป และในส่วนของภาพก็ได้เลือกคอมโพเนนท์ที่เหมาะสมที่สุดมาใช้ ทำให้โปรแกรมนี้สามารถแสดงผลทั้งภาพและเสียงได้ เสียงที่ได้รับสามารถฟังได้รู้เรื่อง แต่เนื่องจากขนาดบัฟเฟอร์ที่แน่นอน แต่ขนาดข้อมูลที่รับได้ไม่แน่นอน ทำให้เกิดการอ่านข้อมูลซ้ำเกิดขึ้น เกิดเป็นเสียงเดิมซ้ำกัน จนกว่าข้อมูลใหม่จะรับเข้ามา ถ้าฝ่ายส่งพูดไม่รวบติดกันเกินไป ก็จะสามารถสื่อสารกันได้ นอกจากนี้ยังมีการเพิ่มเติมส่วนต่างๆ ดังที่กล่าวไว้แล้วเพื่อง่ายต่อการใช้งาน ในส่วนรายละเอียดเล็กๆ น้อยๆ ก็ได้พยายามแก้ไขจนเสร็จเป็นโปรแกรมนี้ออกมาในที่สุด

7.7 แนวทางในการพัฒนางานเพิ่มเติม

DirectX ถ้าได้มีการศึกษาอย่างละเอียดลึกซึ้งแล้วไม่ใช้เฉพาะผลิตเกมส์หรือภาพกราฟิกต่างๆ เท่านั้น ยังจะสามารถนำมาผลิตเป็นโปรแกรมประเภทนี้นี้ได้โดยไม่ต้องพึ่งคอมโพเนนท์อื่นๆ เลย ตั้งแต่ที่มีใช้ในการทดลองคือ DirectSound ที่คอยควบคุมเรื่องเสียงทุกอย่าง, DirectShow ซึ่งเป็นส่วนที่ยากมากส่วนหนึ่งใน DirectX ที่สามารถจับภาพออกมาแสดงผลได้ โดยทำงานต่างๆ คล้ายฟังก์ชัน Video for Windows ของวินโดวส์เอง คือสามารถรวมหรือแยกภาพและเสียงได้, กำหนดการบีบอัดข้อมูลได้ สามารถการจัดการข้อมูลประเภท Multimedia ทุกชนิดด้วยเทคนิคที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพ และ DirectPlay ที่ใช้ในการติดต่อสื่อสารผ่านเครือข่าย แทนการใช้ซอกเก็ทแบบทั่วๆ ไป ถึงแม้ว่าจะเหมาะสำหรับการพัฒนาเกมส์มากกว่าก็ตาม ซึ่งทั้งหมดนี้ถ้ามีการศึกษาอย่างละเอียดจนสามารถนำไปใช้งานได้ ความสามารถในการส่งข้อมูลและคุณภาพของการแสดงผลจะมีประสิทธิภาพสูงขึ้นมาก

ถ้าจะกล่าวถึงขนาดของโปรแกรม การใช้ภาษา C++ จะเป็นทางออกที่ดี เพราะไฟล์ที่ได้มีขนาดเล็ก ถ้าสามารถศึกษาวิธีการเขียนจนสามารถใช้งานได้ ก็จะทำให้โปรแกรมมีประสิทธิภาพขึ้นเช่นกัน แต่การใช้ภาษา C++ จนชำนาญนั้นเป็นเรื่องที่ยาก เพราะมีฟังก์ชันการทำงานที่อยู่ในระดับต่ำใกล้กับภาษาเครื่อง แม้จะเป็นการติดต่อกับคอมพิวเตอร์โดยตรงแต่การเขียนก็จะต้องมีความซับซ้อนมาก จะต้องใช้เวลาในการศึกษาการทำงานเป็นเวลานาน ซึ่งตรงนี้ก็แล้วแต่ความชำนาญของผู้ใช้ ว่าจะใช้ภาษาใด จนมาในปัจจุบันทางเลือกที่ดีที่สุดน่าจะเป็น Visual Basic เพราะเมื่อไม่นานมานี้ ทาง Microsoft ได้ออก DirectX 8.00 SDK สำหรับ Visual Basic มาโดยเฉพาะ แม้จะทำงานได้ไม่เท่าเทียมกับ Visual C++ แต่ศึกษาได้ง่ายกว่ามาก นับเป็นทางเลือกที่น่าสนใจมากในการพัฒนาโปรแกรมประเภทนี้ต่อไป

นอกจากนี้การเพิ่มเติมบริการเสริมต่างๆ เข้าไปในโปรแกรมก็เป็นสิ่งที่ช่วยให้โปรแกรมง่ายต่อการใช้งานมากขึ้นไม่ว่าจะเป็น การส่งไฟล์ถึงฝ่ายตรงข้าม, การใช้ Paint Board ร่วมกัน(ดังเช่นใน NetMeeting) หรือการรองรับผู้ใช้หลายคนเป็นต้น เพื่อให้โปรแกรมนี้สามารถนำไปใช้งานได้จริงต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. กมลมาศ กำจรกิจการ, "คู่มือ Borland Delphi 5 ฉบับสมบูรณ์", โปรวิชั่น,กรุงเทพ, 484 หน้า, 2543
2. สัจจะ จรัสรุ่งรวีวร, "Internet & Network Programming กับ VB 6.0 และ ASP", อินโฟเพรส, กรุงเทพ, 284 หน้า, 2542
3. สุวัฒน์ ปุณณะชัยยะ, "เปิดโลกของ TCP/IP และโปรโตคอลของอินเตอร์เน็ต", โปรวิชั่น, กรุงเทพ, 312 หน้า, 2543
4. Bradley Bargen and Peter Donnelly, "Inside DirectX", Microsoft Press, 1998
5. David Barcomb, "OFFICE AUTOMATION : A Survey of Tools and Technology Second Edition", DIGITAL PRESS, p. 194-222, 1989
6. http://community.borland.com
7. http://delphi.icm.edu.pl
8. http://msdn.microsoft.com/library/default.asp
9. http://turbo.gamedev.net/delphix.asp
10. http://web.ku.ac.th/schoolnet/f-snet1.htm
11. http://www.delphi-jedi.org
12. http://www.thaidelphi.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////////////////////////////////////////// Main form//////////////////////////////////////////////////////
unit Meeting;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, ComCtrls, ExtCtrls, StdCtrls, Option, Login, ScktComp, DXSounds,
  Wave, jpeg, MMClient, MMServer, VChat ;

type
  TfrmMeeting = class(TForm)
    mnuMain: TMainMenu;
    cmbIPAddress: TComboBox;
    btnDial: TButton;
    btnHangUp: TButton;
    trbVolume: TTrackBar;
    Image2: TImage;
    stbStatus: TStatusBar;
    mnuMeeting: TMenuItem;
    mnuLogin: TMenuItem;
    mnuLogout: TMenuItem;
    mnuSeperate: TMenuItem;
    mnuExit: TMenuItem;
    mnuWindow: TMenuItem;
    mnuShow: TMenuItem;
    Label1: TLabel;
    DXSound: TDXSound;
    socClient: TClientSocket;
    socServer: TServerSocket;
    mnuIPAddress: TMenuItem;
    mnuOption: TMenuItem;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lblVolume: TLabel;
socClData: TClientSocket;
socSvData: TServerSocket;
MMServer1: TMMServer;
MMClient1: TMMClient;
PanelCl: TPanel;
socClSound: TClientSocket;
socSvSound: TServerSocket;
PanelSv: TPanel;
Label2: TLabel;
Label3: TLabel;
mnuAddition: TMenuItem;
ChatBoard: TMenuItem;
socClChat: TClientSocket;
socSvChat: TServerSocket;
procedure mnuExitClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure mnuLoginClick(Sender: TObject);
procedure mnuLogoutClick(Sender: TObject);
procedure mnuShowClick(Sender: TObject);
procedure mnuMywinClick(Sender: TObject);
procedure OnsocServerListen(Sender: TObject; Socket: TCustomWinSocket);
procedure btnDialClick(Sender: TObject);
procedure OnsocClientConnecting(Sender: TObject;
  Socket: TCustomWinSocket);
procedure OnsocClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure OnsocServerAccept(Sender: TObject;
  Socket: TCustomWinSocket);
procedure OnsocClientLookup(Sender: TObject; Socket: TCustomWinSocket);
procedure OnsocClientError(Sender: TObject; Socket: TCustomWinSocket;
  ErrorEvent: TErrorEvent; var ErrorCode: Integer);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure OnsocServerClientError(Sender: TObject;
      Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
      var ErrorCode: Integer);
    procedure btnHangUpClick(Sender: TObject);
    procedure OnDXSoundInitialize(Sender: TObject);
    procedure OnDXSoundFinalize(Sender: TObject);
    procedure trbVolumeChange(Sender: TObject);
    procedure mnuOptionClick(Sender: TObject);
    procedure OnsocServerReceive(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure OnsocClientReceive(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure OnMMServerPicStream(Sender: TObject; Content: Byte;
      Data: TMemoryStream);
    procedure OnsocClDataReceive(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnsocClSoundRead(Sender: TObject; Socket: TCustomWinSocket);
    procedure ChatBoardClick(Sender: TObject);
    procedure socClChatRead(Sender: TObject; Socket: TCustomWinSocket);
  private
    { Private declarations }
    FUpdating: Boolean;
    WaveStream : TWaveStream;
    MySound: TMemoryStream;
    procedure CaptureFilledBuffer(Sender: TObject);
  public
    { Public declarations }
    LoginName : string;
    Capture : TSoundCaptureStream;
    Audio : TAudioStream;
    WaveBuf : TWaveStream;
    MemBuf : TMemoryStream;
    SoundPass : boolean;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ImagePass : boolean;
    TSCommand : integer;
    TRCommand : integer;
    procedure WaveInitial;

  end;

var
  frmMeeting: TfrmMeeting;
  wLogout : word;
  wShowPlay : string;

implementation

{$R *.DFM}

{Exit}
procedure TfrmMeeting.mnuExitClick(Sender: TObject);
begin
    self.mnuLogout.Click; //Automate Logout
    if mnuLogout.Enabled then exit;
    socServer.Free;  //Release memories
    socClient.Free;
    close;
end;

{Create Form}
procedure TfrmMeeting.FormCreate(Sender: TObject);
begin
    mnuLogout.Enabled    := False;
    mnuWindow.Enabled    := False;
    mnuAddition.Enabled  := False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    soundpass          := True;
    ImagePass          := True;
    trbVolume.Enabled  := False;
    btnDial.Enabled    := False;
    btnHangUp.Enabled  := False;
    cmbIPAddress.Enabled := False;

end;

{Log in}
procedure TfrmMeeting.mnuLoginClick(Sender: TObject);
begin
    frmLogin.ShowModal;

    if bPass then
    begin
        mnuLogin.Enabled    := False;
        mnuLogout.Enabled   := True;
        mnuWindow.Enabled   := True;
        mnuAddition.Enabled := True;
        btnDial.Enabled     := True;
        mnuShow.Checked     := False;
        cmbIPAddress.Enabled := True;
        stbStatus.SimpleText := 'wait for registering...';
        if MMServer1.Drivers.Count > 0 then
        begin
            PanelCl.Visible := False;
            PanelSv.Visible := False;
            mnuShow.Checked := True;
            MMServer1.OpenVideo(0);
        end else
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Showmessage('You have no driver to play.You cannot use image function.');
        ImagePass := False;
      end;
      DXSound.Initialize;
      cmbIPAddress.SetFocus;
      socServer.Open;
    end;

end;

{Log out}
procedure TfrmMeeting.mnuLogoutClick(Sender: TObject);
begin
  if btnHangup.Enabled then
  begin
    messageDlg('Hang up conference before close',mtWarning,[mbOK],0);
    exit;
  end;
  wLogout := MessageDlg('Do you want to log out now?',mtConfirmation,[mbYes,mbNo],0);
  if wLogout = mrNo then exit else
  begin
    mnuLogin.Enabled := True;
    mnuLogout.Enabled := False;
    mnuWindow.Enabled := False;
    mnuAddition.Enabled := False;
    btnDial.Enabled := False;
    cmbIPAddress.Text := '';
    cmbIPAddress.Enabled := False;
    frmLogin.edtName.Text := '';
    frmLogin.edtPassword.Text := '';
    DXSound.Finalize;
    MMServer1.CloseVideo;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if socServer.Active then socServer.Close;
        stbStatus.SimpleText := 'Log out completed';

    end;

end;

{Show Picture and Sound}
procedure TfrmMeeting.mnuShowClick(Sender: TObject);
begin
    if mnuShow.Checked then
    begin  //No Check
        mnuShow.Checked := False;
        PanelCl.Visible := True;
        PanelSv.Visible := True;

    end else
    begin  //Check
        mnuShow.Checked := True;
        PanelCl.Visible := False;
        PanelSv.Visible := False;
    end;

end;

procedure TfrmMeeting.mnuMywinClick(Sender: TObject);
begin

end;

{Dial Click}
procedure TfrmMeeting.btnDialClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var i : integer;
   CopyIP : string;
begin
   socServer.Close;
   if cmbIPAddress.Text = '' then
   begin
      MessageDlg('Please type remote address',mtWarning,[mbYes],0);
      exit;
   end;

   btnDial.Enabled := False;
   btnHangup.Enabled := True;
   CopyIP := cmbIPAddress.Text;

   cmbIPAddress.Items.Add(cmbIPAddress.Text); {Add IP to combobox}
   for i := 0 to cmbIPAddress.Items.Count-1 do
   begin
      if cmbIPAddress.Items.Strings[i] = CopyIP then
      cmbIPAddress.Items.Delete(cmbIPAddress.Items.Count-1);
   end;
{------------------Use two type of host name--------------------}
   for i := 1 to 9 do
   begin
      if (cmbIPAddress.Text[1] = IntToStr(i)) then
      begin
         socClient.Address := cmbIPAddress.Text;  {Set host name to connect}
         break;
      end
      else if not (socClient.Host = cmbIPAddress.Text) then //Set IP to connect
         socClient.Host := cmbIPAddress.Text;
   end;
{----------------------End at here----------------------------}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    socClient.Port := 2544; {Set port}
    socClient.Open;

end;

//////////////////////TCP/IP Connection////////////////////////////////

{Server Listen}
procedure TfrmMeeting.OnsocServerListen(Sender: TObject;
  Socket: TCustomWinSocket);
begin
    stbStatus.SimpleText := 'Ready to connect';
end;

{Searching for Server}
procedure TfrmMeeting.OnsocClientLookup(Sender: TObject;
  Socket: TCustomWinSocket);
begin
    stbStatus.SimpleText := 'searching for server...';

end;

{Connecting}
procedure TfrmMeeting.OnsocClientConnecting(Sender: TObject;
  Socket: TCustomWinSocket);
begin
    stbStatus.SimpleText := 'connecting...';

end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{Connect Established}
procedure TfrmMeeting.OnsocClientConnect(Sender: TObject;
 Socket: TCustomWinSocket);
begin
   stbStatus.SimpleText := 'waiting for reply...';

end;

{Server Accept}
procedure TfrmMeeting.OnsocServerAccept(Sender: TObject;
 Socket: TCustomWinSocket);
var wAccept : word;
begin
   wAccept := MessageDlg('Accept request: '+socket.RemoteAddress+'?',mtConfirmation,
[mbYes,mbNo],0);
if socServer.Socket.ActiveConnections > 0 then
begin
   if wAccept = mrYes then
   begin
      socSvData.Open;
      socSvSound.Open;
      socSvChat.Open;
      socClient.Close;
      TSCommand := 0;
      socServer.Socket.Connections[0].SendBuf(TSCommand,sizeof(TSCommand));
      btnDial.Enabled := False;
      btnHangup.Enabled := True;
      trbVolume.Enabled := True;
      stbStatus.SimpleText := 'connection complete';
   end else
   begin   // WAccept = mrNo
      TSCommand := 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      socServer.Socket.Connections[0].SendBuf(TSCommand,sizeof(TSCommand));
      btnDial.Enabled := True;
      btnHangUp.Enabled := False;

   end;
end else Showmessage('Remote user already gone.');
end;

{Client Error}
procedure TfrmMeeting.OnsocClientError(Sender: TObject;
 Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
 var ErrorCode: Integer);
begin
   ErrorCode := 0;       // How to case of ErrorCode
   stbStatus.SimpleText := 'Error('+IntToStr(ErrorCode)+'),press [HangUp] to restart';
   cmbIPAddress.SetFocus;

end;

{Server Error}
procedure TfrmMeeting.OnsocServerClientError(Sender: TObject;
 Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
 var ErrorCode: Integer);
begin
   ErrorCode := 0;       // How to case of ErrorCode
   stbStatus.SimpleText := 'Error('+IntToStr(ErrorCode)+'),press [HangUp] to restart';
   cmbIPAddress.SetFocus;

end;

{Server receive command}
procedure TfrmMeeting.OnsocServerReceive(Sender: TObject;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Socket: TCustomWinSocket);
begin
   Socket.ReceiveBuf(TRCommand,sizeof(TRCommand));
   case TRCommand of
   0 : begin  //accept and talk
         {----------For Stream receive---------}
         socClData.Port  := 2545;
         socClSound.Port := 2546;
         socClChat.Port  := 2547;
         if Socket.RemoteHost = '' then
         begin
            socClData.Address := Socket.RemoteAddress;
            socClSound.Address := Socket.RemoteAddress;
            socClChat.Address := Socket.RemoteAddress;
         end else
         begin
            socClData.Host := Socket.RemoteHost;
            socClSound.Host := Socket.RemoteHost;
            socClChat.Host := Socket.RemoteHost;
         end;
         socClData.Open;
         socClSound.Open;
         socClChat.Open;
         {----------End Stream receive---------}
       end;
   2 : begin  //disconnect
         socClData.Close;
         socSvData.Close;
         if soundpass then
            if Audio.Playing then Audio.Stop;
         socClSound.Close;
         socSvSound.Close;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        socClChat.Close;
        socSvChat.Close;

        PanelSv.Visible := True;
        PanelCl.Visible := True;
        mnuShow.Checked := False;
        socServer.Close;
        trbVolume.Enabled := False;
        cmbIPAddress.Text := '';
        btnDial.Enabled := True;
        btnHangup.Enabled := False;
        Showmessage('Finish the conference.');
        socServer.Open;
        cmbIPAddress.SetFocus;

      end;
   end;
end;

{Clinet receive command}
procedure TfrmMeeting.OnsocClientReceive(Sender: TObject;
 Socket: TCustomWinSocket);
begin
   socket.ReceiveBuf(TRCommand,sizeof(TRCommand));
   case TRCommand of
   0 : begin  //accept and talk
        {----------For Stream receive---------}
        socClData.Port   := 2545;
        socClSound.Port  := 2546;
        socClChat.Port   := 2547;
        if socClient.Host = '' then
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        socClData.Address  := socClient.Address;
        socClSound.Address := socClient.Address;
        socClChat.Address := socClient.Address;
      end else
      begin
        socClData.Host  := socClient.Host;
        socClSound.Host := socClient.Host;
        socClChat.Host := socClient.Host;
      end;
      socClData.Open;
      socClSound.Open;
      socClChat.Open;
      {----------End Stream receive---------}
      {-----------For Stream send-----------}
      socSvData.Open;
      socSvSound.Open;
      socSvChat.Open;
      {-----------End Stream send-----------}
      TSCommand := 0;
      socket.SendBuf(TSCommand,sizeof(TSCommand));

      Showmessage('You are allowed to connect.');
      trbVolume.Enabled   := True;
      btnDial.Enabled     := False;
      btnHangUp.Enabled   := True;
      stbStatus.SimpleText:= 'accept';
    end;
  1 : begin  //reject
      Showmessage('You are rejected connection.');
      btnDial.Enabled     := True;
      btnHangUp.Enabled   := False;
      socClient.Close;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        socServer.Open;
        stbStatus.SimpleText:= 'reject';
      end;
  2 : begin  //disconnect
        socClData.Close;
        socSvData.Close;
        if soundpass then
          if Audio.Playing then Audio.Stop;
        socClSound.Close;
        socSvSound.Close;
        socClChat.Close;
        socSvChat.Close;

        PanelSv.Visible := True;
        PanelCl.Visible := True;
        mnuShow.Checked := False;
        stbStatus.SimpleText := 'disconnect';
        socClient.Close;
        trbVolume.Enabled := False;
        cmbIPAddress.Text := '';
        btnDial.Enabled := True;
        btnHangup.Enabled := False;
        Showmessage('Finish the conference.');
        socServer.Open;
        cmbIPAddress.SetFocus;
      end;
    end;

end;
\
//////////////////////////End of connection////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{Hang Up}
procedure TfrmMeeting.btnHangUpClick(Sender: TObject);
begin
  try
  if socClient.Active = True then    //If client decide disconnect
  begin
    TSCommand := 2;
    socClient.Socket.SendBuf(TSCommand,sizeof(TSCommand));
    socClient.Close;
  end;
  if socServer.Active = True then        // If server decide disconnect
  begin
    TSCommand := 2;
    socServer.Socket.Connections[0].SendBuf(TSCommand,sizeof(TSCommand));
    socServer.Close;
  end;
  finally
    socClData.Close;
    socSvData.Close;
    if soundpass then
      if Audio.Playing then Audio.Stop;
    socClSound.Close;
    socSvSound.Close;
    socClChat.Close;
    socSvChat.Close;
    stbStatus.SimpleText := 'Finished';
    btnDial.Enabled := True;
    btnHangup.Enabled := False;
    trbVolume.Enabled := False;
    cmbIPAddress.Text := '';
    PanelSv.Visible := True;
    PanelCl.Visible := True;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    mnuShow.Checked := False;
    socServer.Open;
    cmbIPAddress.SetFocus;
  end;

end;

{------------------------Sound Component-------------------------------}
{Sound Initialize}
procedure TfrmMeeting.OnDXSoundInitialize(Sender: TObject);
begin
  Capture := TSoundCaptureStream.Create(nil);  {Prepare to Capture}

  if Capture.SupportedFormats.Count < 1 then
  begin
    Showmessage('You have NO sound driver.You cannot use sound function.');
    SoundPass := False;
    exit;
  end;
  Capture.BufferLength := 10000;
  Capture.CaptureFormat := 8;

  MemBuf := TMemoryStream.Create;
  MySound := TMemoryStream.Create;
  WaveBuf := TWaveStream.Create(MemBuf);
  WaveStream := TWaveStream.Create(MySound);

  with Capture.SupportedFormats[Capture.CaptureFormat] do
  begin
    WaveBuf.SetPCMFormat(SamplesPerSec,BitsPerSample,Channels);
    WaveStream.SetPCMFormat(SamplesPerSec,BitsPerSample,Channels);
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    WaveBuf.Open(True);
    wavestream.Open(True);

    Capture.OnFilledBuffer := CaptureFilledBuffer;
    Capture.Start;

    Audio := TAudioStream.Create(DXSound.DSound);
    Audio.WaveStream := WaveStream;
    Audio.BufferLength := 80;
    Audio.Looped := True;
    Audio.AutoUpdate := True;

    FUpdating := True;
    try
      trbVolume.Position := Audio.Volume;
    finally
      FUpdating := False;
    end;
end;

{Get Capture}
procedure TfrmMeeting.CaptureFilledBuffer(Sender: TObject);
var MyBuff: Array[0..100000] of byte;
begin
    if socSvSound.Socket.ActiveConnections <> 0 then
    begin
      MemBuf.Clear;
      WaveBuf.CopyFrom(Capture, Capture.FilledSize);
      MemBuf.Position := 0;
      MemBuf.Read(MyBuff, MemBuf.Size);
      socSvSound.Socket.Connections[0].SendBuf(MyBuff, MemBuf.Size);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
end;

{Auto-receive Sound}
procedure TfrmMeeting.OnsocClSoundRead(Sender: TObject;
  Socket: TCustomWinSocket);
var MyBuff: Array[0..8000] of byte;
    b : Integer;
begin
    b := socClSound.Socket.ReceiveBuf(MyBuff, sizeof(MyBuff));
    MySound.Clear;
    MySound.Write(Mybuff, b);
    if not Audio.Playing then
        Audio.Play;

end;

{Volume Change}
procedure TfrmMeeting.trbVolumeChange(Sender: TObject);
begin
    if not FUpdating then
    begin
        Audio.Volume := trbVolume.Position*100;
        if Audio <> nil then
            lblVolume.Caption := Format( '%d dB', [Audio.volume div 100]);
    end;

end;

{Wave initial}
procedure TfrmMeeting.WaveInitial;
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Capture.CaptureFormat := dlgOption.FormatCh;
    with Capture.SupportedFormats[Capture.CaptureFormat] do
       WaveBuf.SetPCMFormat(SamplesPerSec,BitsPerSample,Channels);

end;

{Sound Finalize}
procedure TfrmMeeting.OnDXSoundFinalize(Sender: TObject);
begin
    Audio.Free; Audio := nil;
    Capture.Free; Capture := nil;
    WaveStream.Free;
    MySound.Free;
    WaveBuf.Free; WaveBuf := nil;
    MemBuf.Free;

end;

{----------------------End of sound component-------------------------}

{Option Menu}
procedure TfrmMeeting.mnuOptionClick(Sender: TObject);
begin
    if soundpass then
    begin
       Audio.Stop;
       Capture.Stop;

       dlgOption.ShowModal;
       WaveInitial;
       Capture.Start;
       Audio.Play;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end else
    begin
       dlgOption.ShowModal;
    end;

end;

{-------------------------Picture Part---------------------------------}

{Auto-Send stream}
procedure TfrmMeeting.OnMMServerPicStream(Sender: TObject; Content: Byte;
  Data: TMemoryStream);
var MyBuff: array[0..65000] of Byte;
begin
    try
       Data.Position := 0;
       Data.Read(MyBuff, Data.size);
       socSvData.Socket.Connections[0].SendBuf(MyBuff, Data.Size);
    except
    end;

end;

{Auto-Receive stream}
procedure TfrmMeeting.OnsocClDataReceive(Sender: TObject;
  Socket: TCustomWinSocket);
var MyStream: TMemoryStream;
    MyBuff: Array[0..4095] of Byte;
    b : Integer;
begin
    b := Socket.ReceiveBuf(MyBuff, Sizeof(MyBuff));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MyStream := TMemoryStream.Create;
  try
    MyStream.Write(MyBuff,b);

    MMClient1.ProcessData(MyStream);
  finally
    MyStream.Free;
  end;
end;
{------------------------End of Picture part-----------------------------}

procedure TfrmMeeting.ChatBoardClick(Sender: TObject);
begin
  frmChat.Show;
end;

procedure TfrmMeeting.socClChatRead(Sender: TObject;
 Socket: TCustomWinSocket);
var Text : string;
begin
  Text := Socket.ReceiveText;
  frmChat.memoBoard.Lines.Add(Text);
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////////////////////////////////////////// Log in Form //////////////////////////////////////////////////////
unit Login;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TfrmLogin = class(TForm)
    btnOK: TButton;
    btnCancel: TButton;
    edtName: TEdit;
    edtPassword: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    procedure btnOKClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
    procedure frmLoginShow(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmLogin: TfrmLogin;
  bPass: boolean;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
implementation

uses Meeting;

{$R *.DFM}

procedure TfrmLogin.btnOKClick(Sender: TObject);

begin
    if ((edtname.text = '') or (edtpassword.text = '')) then
    begin
        messagedlg('Please Enter your information',mtWarning,[mbYes],0);
        exit;
    end;
    if edtpassword.text <> 'connect' then
    begin
        messagedlg('Incorrect Password',mtWarning,[mbYes],0);
        edtpassword.text := '';
        exit;
    end;

    bpass := True;
    frmMeeting.LoginName := edtName.Text;
    close;
end;

procedure TfrmLogin.btnCancelClick(Sender: TObject);
begin
    edtName.Text := '';
    edtPassword.Text := '';
    close;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TfrmLogin.frmLoginShow(Sender: TObject);
begin
      edtname.setfocus;
end;

end.
////////////////////////////////////////////////////// Chat Board Form ////////////////////////////////////////////////////
unit VChat;

interface

uses
 Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 StdCtrls;

type
 TfrmChat = class(TForm)
  memoBoard: TMemo;
  btnSend: TButton;
  btnClose: TButton;
  memoSend: TMemo;
  procedure btnCloseClick(Sender: TObject);
  procedure memoSendKeyPress(Sender: TObject; var Key: Char);
  procedure btnSendClick(Sender: TObject);
  procedure FormShow(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 frmChat: TfrmChat;

implementation
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
uses Meeting;

{$R *.DFM}

procedure TfrmChat.btnCloseClick(Sender: TObject);
begin
   MemoBoard.Clear;
   MemoSend.Clear;
   self.Close;

end;

procedure TfrmChat.memoSendKeyPress(Sender: TObject; var Key: Char);
begin
   memoSend.Update;
   if memoSend.Lines.Text = '' then
   begin
      btnSend.Enabled := False;
   end else
      btnSend.Enabled := True;
   case Key of
   #9 :begin
         Key := #0;
         if btnSend.Enabled then btnSend.SetFocus;
       end;
   #27:begin
         Key := #0;
         self.btnClose.Click;
      end;
   end;
end;

procedure TfrmChat.btnSendClick(Sender: TObject);
var Text,UseText: string;
   i : integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
  Text := memoSend.Text;
  i := 0;
  while (Text[i] = '') do  // Check spaces before send //
  begin
    inc(i);
    if i > Length(Text)-1 then
    begin
      MemoSend.Clear;
      btnSend.Enabled := False;
      exit;
    end;
  end;
  UseText := frmMeeting.LoginName + ': ' + memoSend.Text;
  MemoBoard.Lines.Add(UseText);
  with frmMeeting do
  begin
    if socSvChat.Socket.ActiveConnections > 0 then
    begin
      socSvChat.Socket.Connections[0].SendText(UseText);
    end else
      MemoBoard.Lines.Add(#9+'<<!!No one receive your messages!!>>'+#13);
  end;
  MemoSend.Clear;
  btnSend.Enabled := False;
  memoSend.SetFocus;
end;

procedure TfrmChat.FormShow(Sender: TObject);
begin
  btnSend.Enabled := False;
  MemoBoard.Clear;
  MemoSend.Clear;
  memoSend.SetFocus;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end.

////////////////////////////////////////////////////////////// Option Form //////////////////////////////////////////////////////////////
unit Option;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
  Buttons, ExtCtrls, DXClass, DXSounds, Wave, ComCtrls;

type
  TdlgOption = class(TForm)
    OKBtn: TButton;
    CancelBtn: TButton;
    PageControl: TPageControl;
    tabSound: TTabSheet;
    gbSound: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    cmbCaptureFormat: TComboBox;
    cmbDriver: TComboBox;
    tabVideo: TTabSheet;
    btnFormat: TButton;
    btnSource: TButton;
    btnCompress: TButton;
    btnDisplay: TButton;
    cmbVDriver: TComboBox;
    Label3: TLabel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    procedure OnOptionCreate(Sender: TObject);
    procedure OKBtnClick(Sender: TObject);
    procedure CancelBtnClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure btnDisplayClick(Sender: TObject);
    procedure btnCompressClick(Sender: TObject);
    procedure btnSourceClick(Sender: TObject);
    procedure btnFormatClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  FormatCh : integer;
  PreFocus : integer;
  end;

var
  dlgOption: TdlgOption;

implementation

uses Meeting;

{$R *.DFM}

procedure TdlgOption.OnOptionCreate(Sender: TObject);
var i: integer;
    CFormat : TSoundCaptureStream;
const ChannelText : array[1..2] of string = ('Mono','Stereo');
begin
  if frmMeeting.SoundPass then
  begin
     for i:=0 to TSoundCaptureStream.Drivers.Count-1 do
        cmbDriver.Items.Add(TSoundCaptureStream.Drivers[i].Description);
     cmbDriver.ItemIndex := 0;

     CFormat := TSoundCaptureStream.Create(nil);
     cmbCaptureFormat.Items.Clear;
     for i:=0 to CFormat.SupportedFormats.Count-1 do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        with CFormat.SupportedFormats[i] do
        cmbCaptureFormat.Items.Add(Format('%dHz %dbit %s', [SamplesPerSec, BitsPerSample,
ChannelText[Channels]]));
      cmbCaptureFormat.ItemIndex := cmbCaptureFormat.Items.Count-16;
      PreFocus := cmbCaptureFormat.ItemIndex;

      CFormat.Free;
    end;
    if frmMeeting.ImagePass = False then
    begin
      cmbVDriver.Enabled  := False;
      btnCompress.Enabled := False;
      btnFormat.Enabled   := False;
      btnSource.Enabled   := False;
      btnDisplay.Enabled  := False;
    end else
      cmbVDriver.Items := frmMeeting.MMServer1.Drivers;
      cmbVDriver.ItemIndex := 0;

end;

procedure TdlgOption.OKBtnClick(Sender: TObject);
begin
  if frmMeeting.SoundPass then
  begin
    FormatCh := cmbCaptureFormat.ItemIndex;
    PreFocus := cmbCaptureFormat.ItemIndex;
  end;
  if frmMeeting.ImagePass then
  begin
    frmMeeting.MMServer1.OpenVideo(cmbVDriver.ItemIndex);
  end;

end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TdlgOption.CancelBtnClick(Sender: TObject);
begin
    If PreFocus <> cmbCaptureFormat.ItemIndex then
        cmbCaptureFormat.ItemIndex := PreFocus;
end;

procedure TdlgOption.btnDisplayClick(Sender: TObject);
begin
    frmMeeting.MMServer1.DlgDisplay;

end;

procedure TdlgOption.btnCompressClick(Sender: TObject);
begin
    frmMeeting.MMServer1.DlgCompression;

end;

procedure TdlgOption.btnSourceClick(Sender: TObject);
begin
    frmMeeting.MMServer1.DlgSource;
end;

procedure TdlgOption.btnFormatClick(Sender: TObject);
begin
    frmMeeting.MMServer1.DlgFormat;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้