

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ต  
Voice Over Internet Protocol (VoIP)



ภาควิชา  
วิศวกรรมโทรคมนาคม

ผ่านการตรวจรูปเล่มแล้ว  
(ลงชื่อ).....ผู้ตรวจ

โดย

นายวิรัตน์

พยัคฆมาศ

นายศราวุฒิ

รุ่งเรืองวชิเอก

นายศิริวัฒน์

กรัทธิ

เลขหมู่.....

เลขทะเบียน 104182

วัน,เดือน,ปี...3.0.ค.ค.2552

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ต  
Voice Over Internet Protocol (VoIP)



ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมโทรคมนาคม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม


คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ต

**Voice Over Internet Protocol (VoIP)**

ผู้จัดทำ

- |                 |                 |          |
|-----------------|-----------------|----------|
| 1. นายวิรัตน์   | พยัคฆมาศ        | 48010833 |
| 2. นายศราวุฒิ   | รุ่งเรืองวชิเอก | 48010871 |
| 3. นายศิริวัฒน์ | กริทธิ          | 48010895 |

  
อาจารย์ที่ปรึกษา  
( ผศ.นภัทร สระเอี่ยม )



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ต  
VOICE OVER INTERNET PROTOCOL (VoIP)

โดย นายวิรัตน์ พยัคมาศ	48010833
นายศราวดี รุ่งเรืองจีเอก	48010871
นายศิริวัฒน์ กรีทวี	48010895

**บทคัดย่อ**

ในปัจจุบัน ความนิยมการใช้บริการโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ตหรือที่เรียกกันว่า VoIP นั้นได้รับความนิยมสูงขึ้น เนื่องจากมีค่าการใช้บริการที่ถูกกว่าเมื่อเทียบกับระบบโทรศัพท์แบบธรรมดา ด้วยเหตุนี้ปริญญาานิพนธ์ฉบับนี้จึงได้ทำการศึกษาและพัฒนาระบบโทรศัพท์ผ่านเครือข่ายอินเทอร์เน็ต โดยส่วนของ Server จะทำหน้าที่ในการให้บริการและจัดเก็บข้อมูลการใช้บริการ ซึ่งผู้ใช้งานสามารถลงทะเบียนสมัครใช้งานผ่านทางหน้าเว็บเพจ และส่วนของโทรศัพท์ที่ใช้ในการติดต่อสื่อสารจะใช้ระบบ สมอกลงฝังตัวเป็นตัวควบคุมการทำงาน

**ABSTRACT**

Nowadays, There are widespread users of telephone system via internet protocol or “VoIP” that become popular because of the lower fee cost than PSTN. Hence, the objective of this thesis is to study and develop a telephone system via internet protocol by server serve and storage the information of call detail record (CDR). Users can also register through web browser and IP Phone used to communication that control by embedded system.

## สารบัญ

## หน้า

บทคัดย่อ	
สารบัญ	ก
สารบัญรูป	จ
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 OSI Model	4
2.2.1 Application Layer	4
2.2.2 Presentation Layer	4
2.2.3 Session Layer	5
2.2.4 Transport Layer	5
2.2.5 Network Layer	5
2.2.6 Datalink Layer	5
2.2.7 Physical Layer	5
2.3 โพรโทคอล TCP/IP	6
2.3.1 TCP (Transmission Control Protocol)	7
2.3.2 UDP (User Datagram Protocol)	9
2.3.3 IP (Internet Protocol)	10
2.3.3.1 การกำหนดตำแหน่งที่อยู่ (Addressing)	12
2.3.3.2 ซับเน็ตมาสก์ (Subnet Mask)	13
2.3.4 โพรโทคอลอื่นๆ ในลำดับชั้นเน็ตเวิร์ก	13
2.3.4.1 โพรโทคอล ARP (Address Resolution Protocol)	13
2.3.4.2 โพรโทคอล RARP (Reverse Address Resolution Protocol)	14
2.3.4.3 โพรโทคอล ICMP (Internet Control Message Protocol)	14
2.3.4.4 โพรโทคอล IGMP (Internet Group Message Protocol)	14
2.3.5 โพรโทคอลในระดับชั้นแอปพลิเคชัน	14
2.3.5.1 DHCP (Dynamic Host Configuration Protocol)	15
2.3.5.2 DNS (Domain Name System)	15
2.3.5.3 TELNET	15
2.3.5.4 FTP (File Transfer Protocol)	15

2.3.5.5 SMTP (Simple Mail Transfer Protocol)	15
2.3.5.6 POP (Post Office Protocol)	15
2.3.5.7 HTTP (HyperText Transfer Protocol)	15
2.3.6 หมายเลขพอร์ต (Port Number)	16
2.4 มาตรฐาน IEEE 802.3	16
2.4.1 LLC (Logical Link Control)	17
2.4.2 MAC (Media Access Control)	18
2.4.3 โครงสร้างเฟรมข้อมูล	18
2.4.4 โพรโทคอล CSMA/CD	19
2.5 หลักการทำงานของเทคโนโลยี Voice over IP	19
2.5.1 Conversion to PCM (Pulse Code Modulation)	19
2.5.2 Removal of Echo	19
2.5.3 Framing	20
2.5.4 Packetisation	20
2.5.5 Address and Delivery	20
2.5.6 Conversion to Analog	21
2.6 มาตรฐาน SIP (Session Initiation Protocol)	21
2.6.1 คุณสมบัติของ SIP	21
2.6.2 องค์ประกอบสำคัญในระบบ SIP (SIP Component)	22
2.6.2.1 User Agent	22
2.6.2.2 Network Server	22
2.6.3 การทำงานของ SIP Server ในระบบ SIP	23
2.6.3.1 Registrar Server	23
2.6.3.2 Proxy Server	24
2.6.3.3 Redirect Server	25
2.6.4 การทำงานของ SIP ในระบบ Telephony	26
2.6.5 ลำดับสัญญาณ SIP ในระบบ IP Telephony	27
2.6.5.1 Request Message	27
2.6.5.2 Response Message	28
2.6.6 SIP Address	29
2.6.7 โพรโทคอล SDP (Session Description Protocol)	29
2.6.8 การติดต่อโดยใช้สัญญาณ SIP	31
2.6.8.1 การติดต่อแบบโดยตรง	31
2.6.8.2 การติดต่อแบบผ่าน Server (Bypass Server) ใช้ประโยชน์ด้านการค์	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ 2.6.8.2 การติดต่อแบบผ่าน Server (Bypass Server) ใช้ประโยชน์ด้านการค์  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.8.2.1 การติดต่อผ่าน Server แบบ Proxy	32
2.6.8.2.2 การติดต่อผ่าน Server แบบ Redirect	33
2.7 ขั้นตอนการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต	35
2.7.1 การบีบอัดเสียง (Audio Compression)	35
2.7.2 ปัจจัยที่มีผลกระทบต่อคุณภาพของเสียง	36
2.7.2.1 เวลาหน่วง (Delay)	36
2.7.2.2 จิตเตอร์ (Jitter)	37
2.7.2.3 การสูญหายของแพ็กเก็ต (Packet Loss)	37
2.7.2.4 แบนด์วิดท์ที่มีจำกัด (Limited bandwidth)	37
2.7.3 โพรโทคอลที่ใช้ในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต	37
2.7.3.1 โพรโทคอล RTP (Real-time Transport Protocol)	38
2.7.3.2 โพรโทคอล RTCP (RTP Control Protocol)	40
2.8 PJSIP API	40
2.9 โปรแกรม Asterisk	42
2.9.1 สถาปัตยกรรมของ Asterisk	42
2.9.2 ส่วนประกอบภายนอกของ Asterisk	42
2.9.3 ส่วนประกอบภายในของ Asterisk	43
2.10 ภาษา PHP	44
2.11 ระบบจัดการฐานข้อมูล MySQL	45
2.12 ระบบสมองกลฝังตัว (Embedded System)	46
2.12.1 ระบบลินุกซ์แบบฝังตัว (Embedded Linux)	46
2.13 ตัวประมวลผล ARM และบอร์ดพัฒนาระบบสมองกลฝังตัว	46
2.14 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	48
2.14.1 คุณสมบัติของ ไมโครคอนโทรลเลอร์เบอร์ P89V51RD2	48
2.15 มาตรฐานพอร์ตอนุกรมแบบ RS-232	50
2.15.1 รายละเอียดของขาคอนเน็กเตอร์ แบบ DB-9	50
2.15.2 การแปลงระดับแรงดัน	51
บทที่ 3 การออกแบบและการสร้าง	53
3.1 การออกแบบในส่วนของโทรศัพท์ที่ใช้ในการสื่อสาร	53
3.1.1 การออกแบบโปรแกรมในส่วนของบอร์ดพัฒนา JampII mini	53
3.1.1.1 การทำงานของบอร์ดพัฒนา JampII mini	53
3.1.1.2 การออกแบบโปรแกรมในส่วนของ	54
การติดต่อสื่อสารโดยใช้โปรโตคอล SIP	

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 3.1.2 การออกแบบโปรแกรมในส่วนของไมโครคอนโทรลเลอร์ ประโยชน์ด้านการค้า 64  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบในส่วนของ Server	68
3.2.1 การติดตั้ง SIP Server	68
3.2.1.1 การติดตั้งโปรแกรม Asterisk	68
3.2.1.2 Dial Plan ของระบบ	70
3.2.2 การจัดทำ Web Page ที่ใช้ในการลงทะเบียน	71
3.2.2.1 การจัดเตรียม Web Server	71
3.2.2.2 การจัดทำ Web Page ลงทะเบียน	71
3.2.3 การจัดทำระบบฐานข้อมูลที่ใช้เก็บข้อมูลผู้ลงทะเบียน และข้อมูลการใช้งานโทรศัพท์	73
3.2.3.1 การจัดเตรียมโปรแกรมฐานข้อมูล	73
3.2.3.2 การจัดทำระบบฐานข้อมูลของผู้ใช้งาน	73
3.2.3.3 การจัดทำระบบฐานข้อมูลการใช้งานโทรศัพท์	74
บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง	78
4.1 ผลการทดลองในส่วนของโทรศัพท์	78
4.1.1 ผลการทดลองในส่วนของ การ Register กับ SIP Server	79
4.1.2 ผลการทดลองในส่วนของ การโทรออก	80
4.1.3 ผลการทดลองในส่วนของ การรับสาย	83
4.1.4 ผลการทดลองในส่วนของ Audio	85
4.1.5 ผลการทดสอบความล่าช้า (Delay) ของระบบ	91
4.1.5 การใช้งานของโทรศัพท์	96
4.2 ผลการทดลองในส่วนของ Server	103
บทที่ 5 สรุปผลและวิจารณ์การทดลอง	108
5.1 สรุปผลการทดลอง	108
5.2 ปัญหาและอุปสรรคในการทำงาน	104
5.3 แนวทางในการพัฒนาต่อ	108
5.3.1 ส่วนของโทรศัพท์	108
5.3.2 ส่วนของ Server	108
เอกสารอ้างอิง	109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

## หน้า

รูปที่ 1	แผนผังการทำงานของระบบ	1
รูปที่ 2.1	การใช้งานระหว่าง PC กับ PC	3
รูปที่ 2.2	การใช้งานระหว่าง PC กับ โทรศัพท์พื้นฐาน	3
รูปที่ 2.3	การใช้งานระหว่าง โทรศัพท์กับ โทรศัพท์	3
รูปที่ 2.4	OSI Model	4
รูปที่ 2.5	โครงสร้างของ TCP/IP เปรียบเทียบกับ OSI Model	6
รูปที่ 2.6	โครงสร้างของ โพรโตคอล TCP/IP	7
รูปที่ 2.7	โครงสร้างของ TCP Segment	9
รูปที่ 2.8	โครงสร้างของ UDP Segment	10
รูปที่ 2.9	โครงสร้างของ IP Header	10
รูปที่ 2.10	โครงสร้างของ IP Address	12
รูปที่ 2.11	โครงสร้าง Class ของ IP Address	12
รูปที่ 2.12	ช่วงในการใช้งาน IP Address ในแต่ละ Class	13
รูปที่ 2.13	การใช้งาน Subnet Mask	13
รูปที่ 2.14	ตัวอย่าง Socket Address	16
รูปที่ 2.15	โครงสร้างของ Ethernet	17
รูปที่ 2.16	โครงสร้างของเฟรม	18
รูปที่ 2.17	การแปลงสัญญาณ Analog ให้เป็น Digital	19
รูปที่ 2.18	การกำจัด Echo	20
รูปที่ 2.19	การจัดรูปแบบเฟรม	20
รูปที่ 2.20	การแปลงเฟรมให้อยู่ในรูป Packet	20
รูปที่ 2.21	การใส่ค่า IP Address ให้กับ Packet	20
รูปที่ 2.22	การแปลงสัญญาณ Digital ให้กลับไปเป็น Analog	21
รูปที่ 2.23	องค์ประกอบสำคัญในโพรโตคอล SIP	22
รูปที่ 2.24	การทำงานของ Registrar Server	23
รูปที่ 2.25	การทำงานของ SIP Proxy Server	24
รูปที่ 2.26	การทำงานของ Redirect Server	25
รูปที่ 2.27	การทำงานของ SIP ในระบบ IP Telephony	27
รูปที่ 2.28	ตัวอย่างลำดับสัญญาณที่ใช้ติดต่อกัน โดยตรง	31
รูปที่ 2.29	การติดต่อผ่าน Server แบบ Proxy	32

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.31 การส่งเสียงผ่านเครือข่ายอินเทอร์เน็ต	35
รูปที่ 2.32 บทบาทของ RTP ในการสื่อสารเสียงและวิดีโอบนเครือข่ายอินเทอร์เน็ต	38
รูปที่ 2.33 โครงสร้างแพ็กเก็ตของ RTP	39
รูปที่ 2.34 โครงสร้างเฮดเดอร์ของ RTP	39
รูปที่ 2.35 สถาปัตยกรรมของ PJSIP	41
รูปที่ 2.36 สถาปัตยกรรมของ Asterisk	43
รูปที่ 2.37 หลักการทำงานของระบบฐานข้อมูลร่วมกับ PHP	45
รูปที่ 2.38 บล็อกไดอะแกรมของหน่วยประมวลผล ARM920T	47
รูปที่ 2.39 บล็อกไดอะแกรมแสดงการทำงานของหน่วยประมวลผล ARM920T	48
รูปที่ 2.40 บอร์ดพัฒนารุ่น JampII mini	48
รูปที่ 2.41 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ P89V51RD2	49
รูปที่ 2.42 รูปแสดงคอนเน็กเตอร์แบบ DB-9	50
รูปที่ 2.43 แสดงการต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์โดยใช้สัญญาณเพียง 3 เส้น	51
รูปที่ 2.44 แสดงวงจรการแปลงระดับแรงดันของการสื่อสารแบบอนุกรม	51
รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของโทรศัพท์	53
รูปที่ 3.2 การทำงานของบอร์ด JampII mini	54
รูปที่ 3.3 หน้าจอคอนโซล	54
รูปที่ 3.4 โฟลวชาร์ตเริ่มต้นการทำงานของระบบ	55
รูปที่ 3.5 โฟลวชาร์ตโปรแกรมย่อยรับข้อมูลทาง Serial Port	56
รูปที่ 3.6 โฟลวชาร์ตโปรแกรมย่อยส่งข้อมูล ออกทาง Serial Port	57
รูปที่ 3.7 โฟลวชาร์ตการตั้งค่า IP Address , Subnet Mask และ Default Gateway ให้กับ อินเทอร์เน็ตเฟซของพอร์ต Ethernet	57
รูปที่ 3.8 โฟลวชาร์ตการรับค่า User name , Password และ IP Address ของ SIP Server เพื่อใช้ในการลงทะเบียนที่ SIP Server	58
รูปที่ 3.9 การ init ค่าเริ่มต้นให้กับไลบรารี PJSUA	59
รูปที่ 3.10 การลงทะเบียนใน SIP Server โดยมีการ Authentication	60
รูปที่ 3.11 โปรแกรมย่อย Login เข้าสู่ SIP Server	61
รูปที่ 3.12 โฟลวชาร์ตแสดงเหตุการณ์เมื่อมีคนโทรเข้ามา	62
รูปที่ 3.13 โฟลวชาร์ตแสดงเหตุการณ์เมื่อต้องการโทรออก	63
รูปที่ 3.14 วงจรในส่วนของไมโครคอนโทรลเลอร์	64
รูปที่ 3.15 โปรแกรมย่อยถอดรหัสคีย์แพด	65
รูปที่ 3.16 โปรแกรมย่อยรับข้อมูลจากบอร์ด JampII mini	65
รูปที่ 3.17 โปรแกรมย่อยส่งข้อมูลไปยังบอร์ด JampII mini	66
รูปที่ 3.18 โฟลวชาร์ตการทำงานของส่วนไมโครคอนโทรลเลอร์	67

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 หน้าจอของ Asterisk CLI	69
รูปที่ 3.20 ไฟล์ที่ใช้ในการปรับแต่งค่าของระบบ	69
รูปที่ 3.21 Dial Plan ของระบบ	70
รูปที่ 3.22 หลักการทำงานของ Web Page ที่ใช้ลงทะเบียน	71
รูปที่ 3.23 โฟลว์ชาร์ตการทำงานของ Web Page ที่ใช้ลงทะเบียน	72
รูปที่ 3.24 หน้าจอ Command ของ MySQL	73
รูปที่ 3.25 โครงสร้างของตาราง member	74
รูปที่ 3.26 ไฟล์ Master.csv เมื่อใช้โปรแกรม Exel เปิด	75
รูปที่ 3.27 โครงสร้างของตารางของ cdr	76
รูปที่ 3.28 สถานะของฐานข้อมูลเมื่อเชื่อมต่อกับ Asterisk	76
รูปที่ 3.29 ข้อมูลการใช้งานโทรศัพท์เมื่อจัดเก็บอยู่ในฐานข้อมูล	77
รูปที่ 4.1 Topology ที่ใช้ในการทดลอง	78
รูปที่ 4.2 โปรแกรม Wireshark ที่ใช้ในการดักจับแพ็กเก็ต	78
รูปที่ 4.3 แพ็กเก็ตในขณะทำการ Register	79
รูปที่ 4.4 รายละเอียดของ Header SIP Request Message REGISTER	80
รูปที่ 4.5 โปรแกรม X-lite ที่ใช้ในการทดสอบ	81
รูปที่ 4.6 แพ็กเก็ตในขณะหมายเลข 5001 โทรหาหมายเลข 5003	81
รูปที่ 4.7 รายละเอียดของ Header SIP Request Message INVITE	82
รูปที่ 4.8 แพ็กเก็ตในระหว่างทำการสนทนา	83
รูปที่ 4.9 แพ็กเก็ตในขณะหมายเลข 5003 โทรเข้ามา	84
รูปที่ 4.10 รายละเอียด Header SIP Request Message INVITE ที่ส่งไปยัง SIP Server	84
รูปที่ 4.11 แพ็กเก็ตในระหว่างการสนทนา	85
รูปที่ 4.12 บล็อกไดอะแกรมการทำงานของ TLV320AIC23B	86
รูปที่ 4.13 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 500 Hz	87
รูปที่ 4.14 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 1000 Hz	87
รูปที่ 4.15 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 1500 Hz	88
รูปที่ 4.16 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 2000 Hz	88
รูปที่ 4.17 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 2500 Hz	89
รูปที่ 4.18 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 3000 Hz	89
รูปที่ 4.19 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 3500 Hz	90
รูปที่ 4.20 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 4000 Hz	90
รูปที่ 4.21 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 1	91
รูปที่ 4.22 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 2	91
รูปที่ 4.23 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 3	92

รูปที่ 4.24 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 4	92
รูปที่ 4.25 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 5	93
รูปที่ 4.26 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 6	93
รูปที่ 4.27 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 7	94
รูปที่ 4.28 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 8	94
รูปที่ 4.29 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 9	95
รูปที่ 4.30 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 10	95
รูปที่ 4.31 โทรศัพท์ที่ใช้ในการสื่อสาร	96
รูปที่ 4.32 พอร์ตที่ใช้ในการเชื่อมต่อต่างๆ	97
รูปที่ 4.33 สถานะจอ LCD เมื่อรอทำการบูตเข้าสู่ระบบ	97
รูปที่ 4.34 สถานะจอ LCD เมื่อบูตเข้าสู่ระบบเรียบร้อยแล้ว	98
รูปที่ 4.35 ทำการตั้งค่า IP Address	98
รูปที่ 4.36 ทำการตั้งค่า Subnet Mask	99
รูปที่ 4.37 ทำการตั้งค่า Default Gateway	99
รูปที่ 4.38 ใส่ค่า User ID ที่ได้รับมาจากการลงทะเบียนผ่านทางหน้าเว็บเพจ	100
รูปที่ 4.39 ใส่ค่า Password ที่ได้ทำการตั้งค่าเอาไว้	100
รูปที่ 4.40 ใส่ค่า IP Address ของ SIP Server โดยในปัจจุบันจะตั้งอยู่ที่ 161.246.18.134	101
รูปที่ 4.41 เข้าสู่โหมดรอรับหมายเลขเพื่อโทรออก	101
รูปที่ 4.42 การโทรไปหาหมายเลขภายใน	102
รูปที่ 4.43 การโทรไปหาหมายเลขภายนอก	102
รูปที่ 4.44 เว็บเพจหน้าหลัก	103
รูปที่ 4.45 เว็บเพจที่ใช้ในการลงทะเบียน	104
รูปที่ 4.46 ข้อมูลของผู้ใช้งานที่ถูกเก็บลงไปในฐานะข้อมูล	105
รูปที่ 4.47 หน้าจอคอนโซลของ Asterisk เมื่อใช้คำสั่ง “sip show peers”	105
รูปที่ 4.48 ID และ Password หลังจากลงทะเบียนเสร็จแล้ว	105
รูปที่ 4.49 ค้นหาข้อมูลหมายเลขที่ต้องการ	106
รูปที่ 4.50 ข้อมูลผู้ใช้งานหมายเลขที่ค้นหา	106
รูปที่ 4.51 ค้นหาข้อมูลการใช้งานของหมายเลขที่ต้องการ	106
รูปที่ 4.52 ข้อมูลการใช้งานโทรศัพท์ของหมายเลขที่ค้นหา	107

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

## หน้า

ตารางที่ 2.1 คุณสมบัติของ Proxy Server และ Redirect Server	26
ตารางที่ 2.2 เปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดข้อมูลเสียง	36
ตารางที่ 2.3 แสดงการจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมแบบ DB-9	50
ตารางที่ 4.1 ผลการทดสอบ Delay	96



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

## บทนำ

เทคโนโลยี “VoIP” หรือ Voice over Internet Protocol เป็นเทคโนโลยีการสื่อสารแบบใหม่ที่ทำให้เราสามารถรับ-ส่งสัญญาณเสียงผ่านทางเครือข่ายอินเทอร์เน็ตหรืออินทราเน็ตได้ โดยระบบ VoIP ถูกพัฒนาขึ้นโดย ARPANET (Advance Research Projects Agency Network) เมื่อปี ค.ศ. 1973 เพื่อให้สามารถใช้เครือข่ายได้อย่างมีประสิทธิภาพและเป็นการลดต้นทุนทางการสื่อสารได้เป็นอย่างดี ซึ่งการทำงานของระบบ VoIP นั้นจะมีการแปลงสัญญาณเสียงจากต้นทางให้อยู่ในรูปของ Packet เล็กๆ แล้วส่งไปยังผู้รับปลายทางโดยอาศัยโปรโตคอลที่เรียกว่า Internet Protocol (IP) โดยในปัจจุบันเทคโนโลยี VoIP เป็นสิ่งที่หลายองค์กรเริ่มให้ความสนใจ และนำมาใช้กันอย่างแพร่หลาย โดยมีจุดมุ่งหมายหลักเพื่อลดค่าใช้จ่ายทางโทรศัพท์และเตรียมพร้อมสำหรับเทคโนโลยีที่จะเกิดขึ้นในอนาคต

เนื่องจากในปัจจุบันภายในมหาวิทยาลัยต่างๆ ได้มีการวางเครือข่ายท้องถิ่น หรือ LAN ทั้งแบบใช้สายและไร้สาย เพื่อให้ให้นักศึกษาสามารถใช้บริการอินเทอร์เน็ตได้อย่างสะดวก ทางกลุ่มผู้จัดทำจึงมีความคิดที่จะนำระบบ VoIP มาใช้ภายในสถาบันเพื่อเป็นการประหยัดค่าโทรศัพท์มือถือที่ใช้โทรหากันเองภายในสถาบัน โดยจะใช้งานผ่านทางเครือข่ายที่สถาบันติดตั้งไว้ให้อยู่แล้ว ผู้ที่จะต้องการใช้งานสามารถลงทะเบียนผ่านทาง Web Page ซึ่งจะติดต่อกับ VoIP Server โดยจะทำหน้าที่ให้บริการและจัดเก็บข้อมูลการโทรของผู้ใช้งาน โดยจะใช้โปรโตคอล SIP ในการทำงาน ส่วนโทรศัพท์ที่ใช้ในการสื่อสารนั้นจะใช้บอร์ดพัฒนาระบบสมองกลฝังตัวรุ่น JampII mini ของบริษัท ดีไซน์ เกตเวย์ โดยจะรับหมายเลขจากคีย์แพด และแสดงผลผ่านทางจอ LCD ซึ่งจะมีแผนผังการทำงานดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 1 แผนผังการทำงานของระบบ หน้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีหรือหลักการ

#### 2.1 วิวัฒนาการการสื่อสารผ่านอินเทอร์เน็ต

ปัจจุบันการใช้อินเทอร์เน็ตมีบทบาทกับชีวิตประจำวันมากขึ้น และใช้งานกันอย่างกว้างขวางเป็น โดยเฉพาะอย่างยิ่งความจำเป็นที่จะต้องติดต่อสื่อสาร อินเทอร์เน็ตจึงได้รับการพัฒนาโครงสร้างพื้นฐาน เพื่อรองรับการสื่อสารรูปแบบต่างๆ เช่น การใช้จดหมายอิเล็กทรอนิกส์ การติดต่อด้วยเสียง ระบบ VDO Conference การใช้โทรศัพท์บนเครือข่าย ซึ่งก็มีวิวัฒนาการตามลำดับเบื้องต้นดังนี้

##### - E-mail หรือ จดหมายอิเล็กทรอนิกส์

เป็นบริการอย่างหนึ่งที่นิยมใช้กันอย่างแพร่หลายมาก จนทำให้บางคนคิดว่า E-mail คือ อินเทอร์เน็ต และอินเทอร์เน็ตคือ E-mail วิธีใช้งานอีเมลก็ง่ายและมีประโยชน์มาก การทำงานของ E-mail มีลักษณะคล้ายกับระบบไปรษณีย์ปกติ (หมายถึงระบบที่ใช้กระดาษในการเขียนจดหมาย) กล่าวคือในระบบไปรษณีย์ปกติมีหน่วยงานที่ทำหน้าที่ในการรับส่งจดหมายคือเป็นบุรุษไปรษณีย์ (ในกรณีของประเทศไทยคือการสื่อสารแห่งประเทศไทย) ถ้าเป็นในอินเทอร์เน็ตสิ่งที่ทำหน้าที่คอยรับส่งจดหมายคือบรรดาคอมพิวเตอร์ทั้งหลายที่ทำหน้าที่เป็น E-mail Server (คอมพิวเตอร์ที่ทำหน้าที่ให้บริการด้านจดหมายอิเล็กทรอนิกส์)

##### - Chat

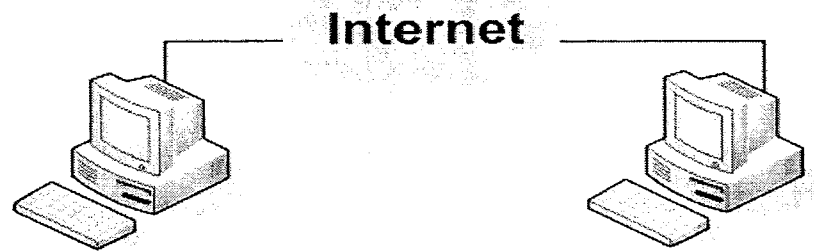
คือ การส่งข้อความสั้นๆ ระหว่างบุคคลที่อยู่หน้าเครื่องคอมพิวเตอร์ในเวลาเดียวกัน และสามารถเขียนโต้ตอบกันไปมาคล้ายกับการคุยกัน ซึ่งก็ได้มีการพัฒนาโปรแกรมสำหรับการ Chat ออกมามากมายที่เป็นที่นิยมและใช้กันอย่างแพร่หลาย เช่น ICQ, MSN Messenger เป็นต้น

และสิ่งหนึ่งที่มีการพัฒนาต่อมา คือระบบการสื่อสารด้วยเสียงผ่านเครือข่าย IP ที่เรียกว่าเทคโนโลยี Voice over IP หรือที่รู้จักกันโดยทั่วไปว่า “VoIP” จนสามารถใช้งานได้ดีขึ้น เพื่อให้ได้รับประโยชน์และมีความสะดวกมากที่สุด VoIP ถูกเริ่มต้นใช้งานกันอย่างกว้างขวาง เพื่อให้เครื่องคอมพิวเตอร์ส่วนบุคคลสามารถสนทนา ระหว่างกัน ได้ รวมถึงการสนทนากับโทรศัพท์พื้นฐานอีกด้วย โดยไม่เสียค่าบริการแต่อย่างใด และคุณภาพของบริการก็ถูกพัฒนาขึ้นมาเรื่อยๆจนเทียบเท่าระบบโทรศัพท์พื้นฐานซึ่งการใช้งาน VoIP สามารถแบ่งได้เป็น 3 ลักษณะ คือ

#### 1. คอมพิวเตอร์ส่วนบุคคล ไปยัง คอมพิวเตอร์ส่วนบุคคล (PC to PC)

PC มีการติดตั้ง sound card และไมโครโฟนที่เชื่อมต่ออยู่กับเครือข่าย IP การประยุกต์ใช้ PC และ IP-enabled telephones สามารถสื่อสารกันได้แบบจุดต่อจุด หรือ แบบจุดต่อหลายจุด โดยอาศัย software ทาง

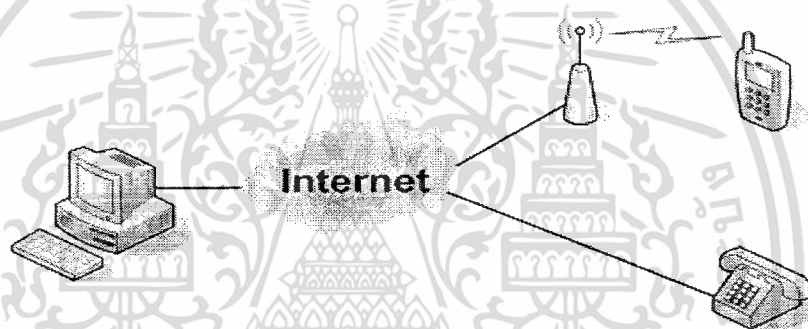
เอกสารนี้เป็นด้าน IP Telephony หรือที่เรียกกันว่า Soft Phone ศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 การใช้งานระหว่าง PC กับ PC

### 2. คอมพิวเตอร์ส่วนบุคคล ไปยัง โทรศัพท์พื้นฐาน (PC to Phone)

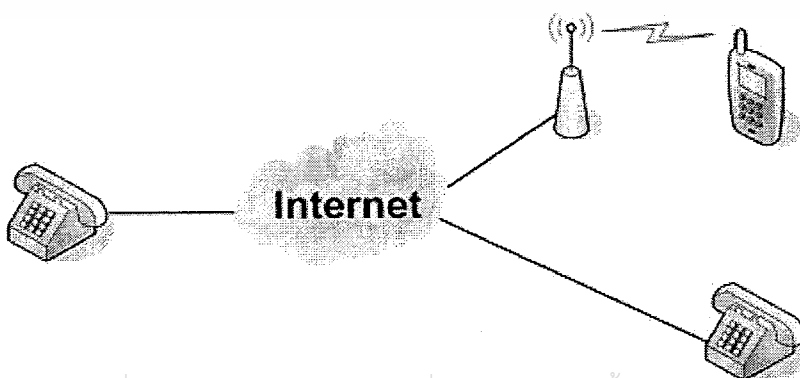
เป็นการเชื่อมต่อเครือข่ายโทรศัพท์เข้ากับ เครือข่าย IP ทำให้โดยอาศัย Voice trunks ที่สนับสนุน voice packet ทำให้สามารถใช้ PC ติดต่อกับ โทรศัพท์ระบบปกติได้



รูปที่ 2.2 การใช้งานระหว่าง PC กับ โทรศัพท์พื้นฐาน

### 3. โทรศัพท์กับ โทรศัพท์ (Telephony)

เป็นการใช้โทรศัพท์ธรรมดาติดต่อกับ โทรศัพท์ธรรมดาแต่ในกรณีนี้จริงๆแล้วประกอบด้วยขั้นตอนการส่งเสียงบนเครือข่าย Packet ประเภทต่างๆซึ่งทั้งหมดติดต่อกันระหว่างชุมสายโทรศัพท์ (PSTN) การติดต่อกับ PSTN หรือการใช้โทรศัพท์ร่วมกับเครือข่ายข้อมูลจำเป็นต้องใช้ Gateway

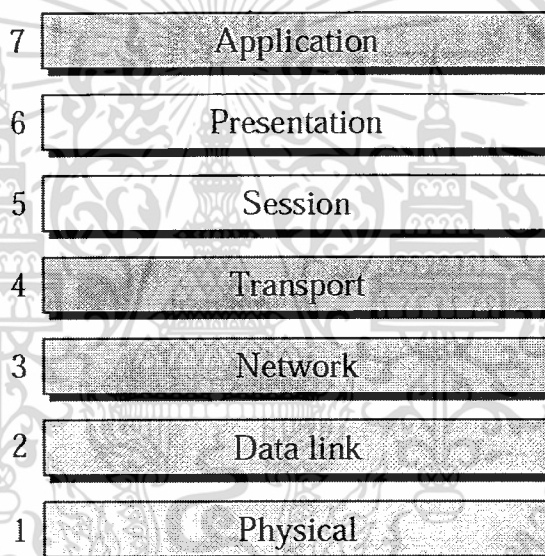


รูปที่ 2.3 การใช้งานระหว่างโทรศัพท์กับโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนที่ลงทะเบียนเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 OSI Model

เมื่อคอมพิวเตอร์เครื่องหนึ่งมีความต้องการรับส่งข้อมูลกับคอมพิวเตอร์เครื่องอื่นๆ การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์เข้าด้วยกันเป็นระบบเครือข่ายก็เกิดขึ้น ปัญหาที่ตามมาก็คือการเชื่อมต่อระหว่างคอมพิวเตอร์ระหว่างระบบที่แตกต่างกันหรือคนละผู้ผลิตเป็นสิ่งที่ทำได้ยากมากในยุคนั้นของการสื่อสารข้อมูล เนื่องจากขาดมาตรฐานส่วนกลางที่จำเป็นต้องใช้ในการรับส่งข้อมูล โดยมากผู้ผลิตแต่ละรายก็จะมีมาตรฐานของตนเองซึ่งเข้ากันไม่ได้กับผู้ผลิตรายอื่นๆ ทำให้ผู้ใช้ต้องผูกติดกับผู้ผลิตรายใดรายหนึ่ง จากปัญหานี้ทำให้หน่วยงานกำหนดมาตรฐานสากล คือ International Standards Organization หรือ ISO ทำการกำหนดโครงสร้างทั้งหมดที่จำเป็นต้องใช้ในการสื่อสารข้อมูลจากคอมพิวเตอร์ระบบหนึ่งไปยังคอมพิวเตอร์อีกระบบหนึ่งขึ้น เรียกว่า Open System Interconnection หรือ OSI Model โดยจะแบ่งฟังก์ชันการทำงานออกเป็น 7 ชั้น ดังรูปที่ 2.4



รูปที่ 2.4 OSI Model

หน้าที่ของแต่ละชั้น (Layer) มีการทำงานดังนี้

### 2.2.1 Application Layer

เป็นเลเยอร์ที่กำหนดอินเตอร์เฟซระหว่างแอปพลิเคชันที่ทำงานบนเครื่องคอมพิวเตอร์กับซอฟต์แวร์สื่อสารต่างๆ ที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ ตัวอย่างเช่น Web Browser ถือว่าเป็นแอปพลิเคชันที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ เมื่อมันต้องการรับส่งข้อมูลเว็บเพจกับเว็บเซิร์ฟเวอร์ มันจะอาศัยความสามารถของ Application Layer ในการอินเตอร์เฟซกับซอฟต์แวร์สื่อสารเลเยอร์ต่างๆ ระดับล่างเพื่อให้ได้มาซึ่งเว็บเพจที่ต้องการ

### 2.2.2 Presentation Layer

จุดประสงค์หลักของเลเยอร์นี้ก็คือ กำหนดฟอร์แมตของการสื่อสาร อย่างเช่น ASCII Text, binary และ JPEG การเข้ารหัสทั้งหมดอยู่ในเลเยอร์นี้ด้วย ตัวอย่างเช่น โปรแกรม FTP ต้องการรับส่งไบนารีไฟล์ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างเครื่องเซิร์ฟเวอร์ปลายทางกับเครื่องไคลเอนต์ โพรโทคอล FTP จะอนุญาตให้ผู้ใช้ระบุฟอร์แมตของข้อมูลที่โอนย้ายกันได้ว่าเป็นแบบ ASCII Text หรือเป็นแบบ Binary

### 2.2.3 Session Layer

เป็นเลเยอร์ที่ควบคุมการสื่อสารจากต้นทางไปยังปลายทางแบบ End to End และคอยควบคุมช่องทางการสื่อสารในกรณีที่มีหลายๆ โปรเซสต้องการรับส่งข้อมูลพร้อมๆ กันบนเครื่องเดียวกัน และยังให้อินเตอร์เฟซสำหรับแอปพลิเคชันเลเยอร์ด้านบนในการควบคุมการทำงานของโปรโตคอลในระดับ Transport / Network อย่างเช่น Socket ที่ใช้ในยูนิคส์ หรือ Window Socket ที่ใช้ในวินโดวส์ ซึ่งได้ให้ Application Programming Interface (API) แก่ผู้พัฒนาซอฟต์แวร์ในระดับบนสำหรับการเขียนโปรแกรมเพื่อควบคุมการทำงานของโปรโตคอล TCP/IP ในระดับล่าง

### 2.2.4 Transport Layer

เป็นเลเยอร์ที่มีหน้าที่หลักๆ ในการแบ่งข้อมูลในเลเยอร์บนให้พอเหมาะกับการจัดส่งไปในเลเยอร์ล่าง หรือเรียกว่า Segmentation ทำหน้าที่ประกอบรวมข้อมูลต่างๆ ที่ได้รับมาจากเลเยอร์ล่าง (Assembly) และให้บริการในการแก้ไขปัญหาเมื่อเกิดข้อผิดพลาดเกิดขึ้นในระหว่างกลางทางของการส่ง (Error Recovery) หน่วยของข้อมูลในเลเยอร์นี้มักถูกเรียกว่า Segment ตัวอย่างของโปรโตคอลในเลเยอร์นี้ที่รู้จักกันดีคือ TCP และ UDP

### 2.2.5 Network Layer

เป็นเลเยอร์ที่มีหน้าที่หลักในการส่งแพ็กเก็ตจากเครื่องต้นทางไปให้ถึงเครื่องปลายทางด้วยความพยายามที่ดีที่สุด (Best effort delivery) เลเยอร์นี้จะมีการกำหนดให้มีการตั้งลอจิกัลแอดเดรสขึ้นมานบนเครื่องคอมพิวเตอร์และอุปกรณ์สื่อสารเพื่อใช้ในการระบุตัวตน ตัวอย่างของโปรโตคอลเลเยอร์นี้ได้แก่โปรโตคอล IP และ ลอจิกัลแอดเดรสที่ใช้ก็คือ หมายเลข IP Address นั้นเอง

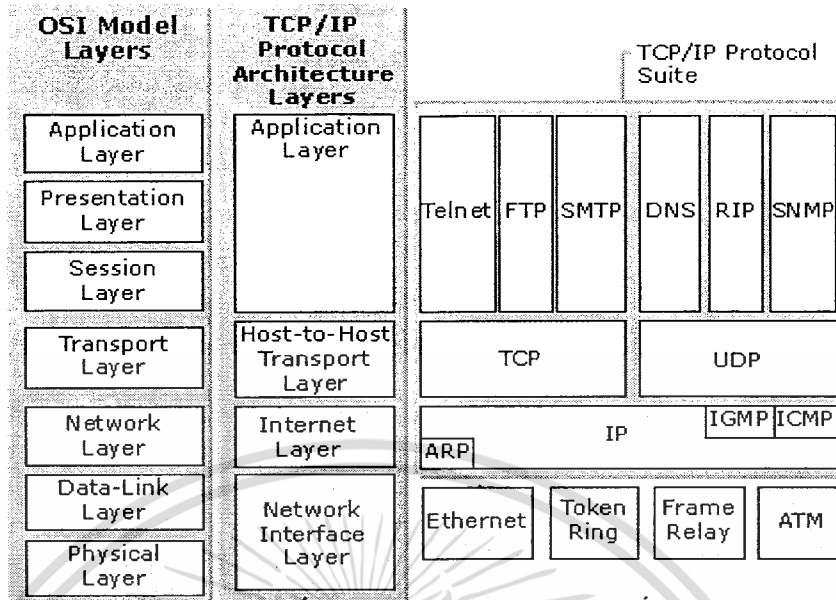
### 2.2.6 Datalink Layer

รับผิดชอบในการส่งข้อมูลบนเน็ตเวิร์คแต่ละประเภท เช่น Ethernet, Token Ring, FDDI หรือบน WAN ต่างๆ และดูแลเรื่องการท้อหุ้มข้อมูลจากเลเยอร์บน เช่น แพ็กเก็ต IP ไว้ภายใน “เฟรม” และส่งจากต้นทางไปยังอุปกรณ์ตัวถัดไป เลเยอร์นี้จะเข้าใจถึงกลไกและ อัลกอริทึมรวมทั้งฟอร์แมตของเฟรมที่ต้องใช้ในเน็ตเวิร์คประเภทต่างๆ เป็นอย่างดี

### 2.2.7 Physical Layer

เลเยอร์นี้จะกำหนดมาตรฐานของสัญญาณไฟฟ้าและมาตรฐานของคอนเน็กเตอร์ เชื่อมต่อต่างๆ รวมถึงมาตรฐานของสายเคเบิลที่จำเป็นต้องใช้ เช่น มาตรฐานสาย CAT ประเภทต่างๆ มาตรฐานของหัวที่เอกสารนี้ใช้เชื่อมต่อ V.35 ที่ใช้ใน WAN และมาตรฐาน RS232 เป็นต้น รวมทั้งแรงดันไฟฟ้าและรูปแบบการรับส่งไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 2.6 โครงสร้างของโปรโตคอล TCP/IP

การทำงานของโปรโตคอลแต่ละตัวมีการทำงานดังนี้

### 2.3.1 TCP (Transmission Control Protocol)

โปรโตคอล TCP ในเลเยอร์ของ Transport นั้นมีหน้าที่หลักๆ คือ

- จัดแบ่งข้อมูลจากระดับแอปพลิเคชันเลเยอร์ให้มีขนาดพอเหมาะที่จะส่งไปบนเน็ตเวิร์ก หน่วยของข้อมูลระดับนี้เรียกว่า TCP Segment
- เริ่มต้นสร้างคอนเน็คชันระหว่างต้นทางและปลายทางให้สำเร็จก่อน ก่อนที่ทั้งต้นทางและปลายทางจะมีการรับส่งข้อมูลกันจริงๆ การรับส่งข้อมูลโดยมีการสร้างคอนเน็คชันก่อนการส่ง เรียกว่า การสื่อสารแบบ Connection-Oriented และกระบวนการที่ใช้ในการสร้างคอนเน็คชันคือ Three Way Handshake
- มีการใส่หมายเลข Sequence Number ลงไปใน TCP Segment ที่ส่งไปเพื่อการจัดลำดับข้อมูล เมื่อปลายทางได้รับ TCP Segment นั้นๆ แล้วจะต้องมีการส่งการยืนยัน (Acknowledgement : ACK) กลับมาเพื่อให้เครื่องต้นทางทราบว่าได้รับ TCP Segment นั้นๆแล้ว
- ดังที่ได้กล่าวไปในหัวข้อที่ผ่านมาว่าเครื่องต้นทางมีการตรวจสอบว่า แพ็กเก็ตถูกส่งไปถึงเครื่องคอมพิวเตอร์ปลายทางหรือไม่ โดยการตรวจเช็คว่าได้รับ ACK กลับมาหรือยัง กรณีไม่ได้รับ ACK ยืนยันกลับมาภายในเวลารอคอยที่เหมาะสมค่าหนึ่ง มันจะเข้าใจว่า TCP Segment นั้นส่งไปไม่ถึงเครื่องปลายทาง ในกรณีนี้เครื่องต้นทางจะมีการส่งใหม่ (Retransmission) อีกครั้งและเพิ่มเวลารอคอยไปอีกระยะเวลาหนึ่งจนกว่าจะได้รับ ACK กลับมา กลไกนี้คือการทำ Error Recovery ซึ่งทำให้โปรโตคอล TCP มีความน่าเชื่อถือในการรับส่งข้อมูล (Reliability)
- ในการส่ง ACK เพื่อยืนยันว่าได้รับข้อมูลครบถ้วนนั้น เครื่องปลายทางไม่จำเป็นต้องส่ง ACK กลับทุกๆ TCP Segment ที่ได้รับแต่สามารถส่ง ACK เมื่อได้รับข้อมูลหลายๆ Segment ตามที่ตกลงกันไว้ก่อนได้โดยเครื่องต้นทางและเครื่องปลายทางต้องมีการตกลงกันแต่แรกว่าจะให้ผู้ที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้รับ TCP Segment ตอบยืนยัน ACK กลับมาเมื่อได้รับ TCP Segment ไปแล้วเป็นจำนวนเท่าไร ขนาดของ TCP Segment ที่ผู้ส่งสามารถส่งได้ครั้งหนึ่งๆ โดยที่ไม่ต้องรอคอยให้มีการตอบรับนี้เรียกว่า “Window Size” นั้นหมายความว่า ผู้รับสามารถรับข้อมูลจนครบตามขนาด Window Size แล้วจึงค่อยส่ง ACK กลับไปที่เดียว

- จัดสรรขนาดบัฟเฟอร์ข้อมูลที่พอเหมาะไว้ทั้งในขณะรับและส่งข้อมูลและช่วยประกอบรวมแพ็กเก็ต IP ที่ได้รับเข้ามาให้เป็นข้อมูลชิ้นเดียวกันก่อนส่งต่อขึ้นไปยังแอปพลิเคชันในระดับบนต่อไป

โดย TCP Segment จะมีโครงสร้างดังรูปที่ 2.7

ซึ่งข้อมูลในส่วน Header นั้นจะประกอบด้วยส่วนต่างๆ ดังนี้

**Source Port:** ส่วนนี้จะเป็นหมายเลขพอร์ตที่เป็นจุดเริ่มต้นการสื่อสารหมายเลขพอร์ตเมื่อรวมกับหมายเลข IP Address จะเป็นที่อยู่ของการส่งกลับข้อมูล

**Destination Port:** เป็นหมายเลขพอร์ตของเครื่องรับ ซึ่งพอร์ตนี้จะเป็นพอร์ตที่ใช้เชื่อมต่อกับแอปพลิเคชันที่จะนำข้อมูลส่งไปให้โปรเซสต่อไป

**Sequence Number:** เป็นหมายเลขที่บอกลำดับของแพ็กเก็ต ที่จะใช้ในฝั่งเครื่องรับในการจัดเรียงข้อมูลให้อยู่ในรูปแบบเดิม ในการส่งข้อมูลผ่านเครือข่ายที่สลับซับซ้อนนั้น แพ็กเก็ตแต่ละชุดอาจจะถูกส่งไปบนเส้นทางที่แตกต่างกัน ดังนั้นจึงเป็นไปได้ว่าที่แพ็กเก็ตจะเดินทางมาถึงปลายทางไม่เป็นไปตามลำดับที่ส่ง หมายเลขนี้จะใช้ในการเรียงลำดับของแพ็กเก็ตเหล่านี้ให้อยู่ในลำดับเดิม

**Acknowledgement Number:** เป็นหมายเลขของลำดับแพ็กเก็ตถัดไปที่ฝั่งด้านรับคาดหวัง ซึ่งเป็นการบอกเป็นนัยว่าแพ็กเก็ตที่มีหมายเลขลำดับก่อนหน้านี้ได้รับหมดแล้วนั่นเอง

**HLLEN:** Header length หรือ บางทีเรียกว่า Data Offset เป็นส่วนที่บอกขนาดข้อมูลส่วนหัว รวมไปถึงส่วนที่เป็น Option ซึ่งมีหน่วยเป็น 32 บิต Word

**Reserved:** ส่วนนี้จะกำหนดให้เป็น 0 ตลอด ซึ่งข้อมูลส่วนนี้ไม่มีความหมายอะไรเพียงแต่เป็นการสงวนไว้ใช้ในอนาคตเมื่อมีการปรับปรุงโปรโตคอล

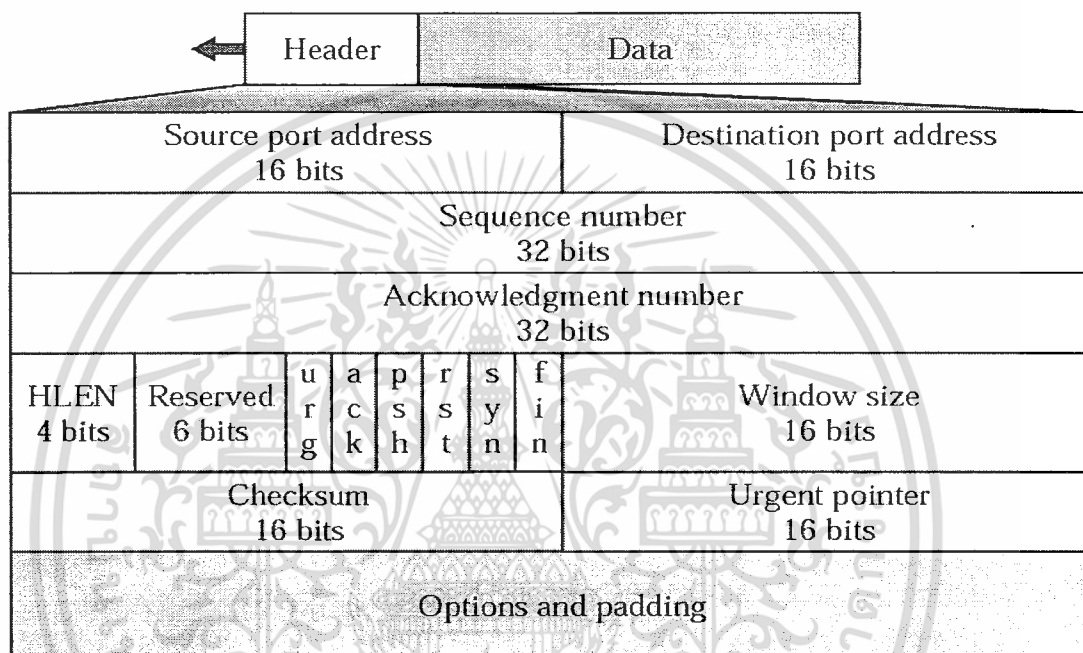
**Flags:** หรือบางทีเรียกว่า Code Bits เป็นข้อมูลที่ใช้ในการควบคุมการรับส่งแพ็กเก็ต โดยถ้าจะใช้งาน flag ใหนก็จะเซตค่าเป็น 1 ส่วน flag ใหนที่ไม่ได้ใช้งานจะเซตค่าเป็น 0 ซึ่งแต่ละ flag จะมีรายละเอียดดังนี้

URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน ซึ่งจะใช้คู่กันกับ Urgent Pointer
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledgement Number นำมาใช้งานได้
PSH	เป็นการแจ้งผู้รับทราบข้อมูลว่าควรส่งข้อมูล Segment นี้ไปยัง Application โดยเร็ว
RST	ยกเลิกการติดต่อ ในกรณีที่เกิดปัญหาขึ้น โดยจะเริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่าขอยุติการติดต่อ

**Window Size:** เป็นตัวเลขที่เครื่องปลายทางบอกให้เครื่องต้นทางทราบขนาดของวินโดวที่เครื่องปลายทางสามารถรับข้อมูลได้

**Checksum:** เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูลในส่วนของ Header โดยที่ฝ่ายส่งจะคำนวณค่า Checksum ของข้อมูลในส่วนหัว เมื่อเรื่องปลายทางได้รับข้อมูลก็จะคำนวณเช็คซ้ำด้วยวิธีเดียวกัน แล้วเปรียบเทียบค่าที่คำนวณได้กับค่าที่อยู่ในฟิลด์นี้ ถ้าเหมือนกันแสดงว่าไม่มีข้อผิดพลาดที่ได้รับ ถ้าไม่ตรงแสดงว่าเกิดข้อผิดพลาดขึ้นทางฝั่งรับจะ drop segment นั้นทิ้งไป

**Option and Padding:** เป็นส่วน Header เพิ่มเติมซึ่งจะเติมให้ Header มีจำนวนบิตที่หารด้วย 32 ลงตัว



รูปที่ 2.7 โครงสร้างของ TCP Segment

### 2.3.2 UDP (User Datagram Protocol)

โปรโตคอล UDP เป็นโปรโตคอล ในระดับ Transport Layer ที่มีความแตกต่างจากโปรโตคอล TCP เกือบๆ ทุกด้าน โปรโตคอล UDP จะทำการส่งข้อมูลโดยไม่มีการสร้างคอนเน็คชันก่อน เรียกว่า Connection-less ไม่มีการส่งยืนยันว่าได้รับข้อมูลแล้ว ไม่มีการจัดเตรียมขนาดบัฟเฟอร์สำหรับการรับส่งข้อมูล และไม่มีการจัดลำดับของข้อมูลที่ได้รับ และไม่มีการจัดลำดับของข้อมูลที่ได้รับ หน้าที่ของการยืนยันว่าได้รับข้อมูลแล้วอาจถูกผลักภาระให้กับแอปพลิเคชันในเลเยอร์บนต่อไป และไม่มีการควบคุมการไหลของการรับส่งข้อมูลด้วย

ซึ่งทำให้โปรโตคอล UDP นั้นเป็นโปรโตคอลที่ไม่มีที่น่าเชื่อถือ Unreliable ส่วนข้อได้เปรียบของโปรโตคอล UDP คือความเร็ว และไม่สิ้นเปลืองเวลาและทรัพยากรที่ต้องใช้ในการติดตามสถานะในการส่งข้อมูลต่างๆ โดยจะมีโครงสร้างดังรูปที่ 2.8

โดยข้อมูลในส่วนของ Header นั้นจะประกอบด้วยส่วนต่างๆ ดังนี้

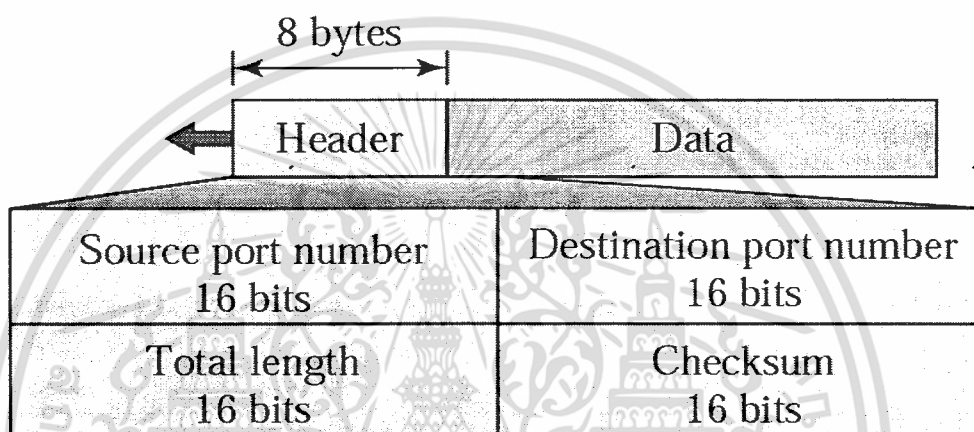
**Source Port:** เป็นหมายเลขพอร์ตของเครื่องส่ง เมื่อรวมหมายเลขพอร์ตนี้กับหมายเลข IP ก็จะ

เอกสารนี้เป็นที่อยู่ของเครื่องรับ ในการตอบกลับข้อมูลศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Destination Port:** เป็นหมายเลขพอร์ตของทางฝั่งเครื่องรับที่ใช้ในการส่งผ่านข้อมูลไปยังแอปพลิเคชันที่ต้องการติดต่อด้วย

**Checksum:** เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูล ซึ่งสามารถมีก็ได้ ไม่มีก็ได้ ถ้าไม่ต้องการใช้ Checksum จะเซตทุกบิตเป็น 0 หหมด โดยที่เครื่องทางฝั่งรับจะคำนวณหมายเลขนี้ด้วยวิธีเดียวกัน แล้วเปรียบเทียบกับค่าที่ส่งมา ถ้าเท่ากันแสดงว่า ไม่มีข้อผิดพลาด

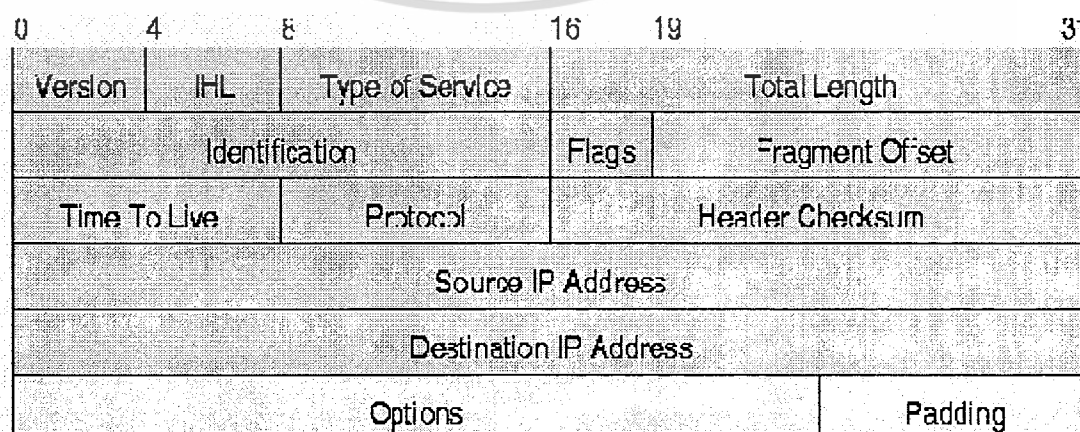
**Total Length:** เป็นข้อมูลที่บอกความยาวของข้อมูลทั้งหมด ซึ่งจะเป็นข้อมูลที่ช่วยให้ทางฝั่งรับทราบว่าข้อมูลควรมีขนาดเท่าไร



รูปที่ 2.8 โครงสร้างของ UDP Segment

### 2.3.3 IP (Internet Protocol)

โปรโตคอล IP เป็นโปรโตคอลในระดับ Network Layer โดยจะทำหน้าที่จัดการเกี่ยวกับการรับส่งแพ็กเก็ต ซึ่งจะรับข้อมูลมาจากเลขอร์ที่สูงกว่าเช่น TCP, UDP ถ้าโฮสต์ปลายทางอยู่บนละเครือข่ายกับโฮสต์ที่ส่งข้อมูล IP จะรับผิดชอบในการจัดเส้นทาง ให้แพ็กเก็ตส่งไปยังเครือข่ายที่โฮสต์นั้นอยู่ ซึ่งในการจัดส่งแพ็กเก็ตข้ามเครือข่ายนั้น IP จะใช้เราท์เตอร์ (Router) ในการเชื่อมต่อเครือข่ายเหล่านั้น ซึ่งโปรโตคอล IP จะมีโครงสร้างดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างของ IP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยข้อมูลในส่วนของ Header จะประกอบไปด้วยส่วนต่างๆ ดังนี้

**Version:** สำหรับฟิลด์แรกนี้ คือ หมายเลขเวอร์ชันของไอพี ซึ่งปัจจุบันเวอร์ชันที่ใช้กัน คือ เวอร์ชัน 4 (IPv4) โดยมีค่าไบแนรีที่กำหนดไว้เท่ากับ 0100 (สำหรับแนวโน้มอนาคตอันใกล้นี้ จะมีการนำ IPv6 มาใช้งานเพื่อรองรับอัตราการเจริญเติบโตของผู้ใช้อินเทอร์เน็ต ที่นับวันจะเพิ่มขึ้นและเติบโตอย่างต่อเนื่อง)

**IHL (Internet Header Length):** เป็นฟิลด์ที่ใช้ระบุความกว้างของ Header ซึ่งมีขนาด 4 บิต ทำให้สามารถแทนค่าตัวเลขระหว่าง 0 ถึง 15 และเนื่องจากหน่วยนับความยาวจะเป็น 4 เท่าของไบต์ หรือ 32 บิต ดังนั้น เมื่อมีการนำมาคูณด้วย 4 จึงทำให้มีค่าสูงสุดเท่ากับ 60 ไบต์

**Type of Service:** เป็นฟิลด์ขนาด 8 บิต ที่ใช้กำหนดรูปแบบการบริการให้กับฝ่ายส่ง เช่น อัตราการส่งผ่าน (Throughput), ความล่าช้า (Delay), ความสำคัญ (Precedence) และความน่าเชื่อถือของข้อมูลในแพ็กเก็ตนี้

**Total Length:** เป็นฟิลด์ขนาด 16 บิต หรือ 2 ไบต์ ที่ใช้ระบุความยาวทั้งหมด (รวมทั้ง Header) ของไอพีดาต้าแกรม ซึ่งสามารถระบุความยาวได้สูงสุด 64 กิโลไบต์ หรือ 65,535 ไบต์

**Identification:** สำหรับฟิลด์นี้ จะใช้สำหรับการแฟรกเมนต์ (Fragmentation) โดยการแตกเป็นแพ็กเก็ตย่อยๆ โดยในขณะที่แพ็กเก็ตได้มีการส่งผ่านไปยังเครือข่ายต่างชนิดกัน อาจจำเป็นต้องมีการแบ่งเป็นแฟรกเมนต์ เพื่อให้เข้ากับขนาดเฟรมของเครือข่ายนั้นๆ โดยแต่ละแฟรกเมนต์จะถูกระบุหมายเลขลำดับ (Sequence Number) ในฟิลด์นี้

**Flags:** ในส่วนของฟิลด์ขนาด 3 บิตนี้ จะใช้สำหรับกำหนดค่าด้านแกรมว่า สามารถที่จะทำการแฟรกเมนต์ได้หรือไม่ รวมถึงเป็นตัวที่ระบุว่าแฟรกเมนต์นั้นเป็นแฟรกเมนต์แรก กลาง หรือเป็นแฟรกเมนต์สุดท้าย โดยจะมีลักษณะดังนี้

บิต 0	reserved ไว้เป็น 0 เสมอ
บิต 1 (DF)	0 = มีการ Fragment 1 = ไม่มีการ Fragment
บิต 2 (MF)	0 = เป็นชิ้นสุดท้ายของการ Fragment 1 = ไม่ใช่ชิ้นสุดท้ายของการ Fragment

**Fragmentation Offset:** เป็นส่วนที่ใช้สำหรับเป็นพอยท์เตอร์หรือตัวชี้ตำแหน่งออฟเซตของข้อมูล

**Time To Live (TTL):** เป็นฟิลด์ที่ใช้สำหรับกำหนดอายุขัยของดาต้าแกรม โดยโฮสต์ฝ่ายส่งจะมีการกำหนดค่าเริ่มต้นของอายุขัยให้กับดาต้าแกรม (โดยทั่วไปค่าที่กำหนดไว้คือ 32 หรือ 64) และเมื่อดาต้าแกรมนั้นได้เดินทางผ่านเครือข่ายอินเทอร์เน็ต จากเราท์เตอร์ไปยังเราท์เตอร์ เราท์เตอร์แต่ละตัวก็จะทำการลดค่านี้ลงทีละหน่วย และหากค่าดังกล่าวได้ถูกลดค่าลง จนกระทั่งมีค่าเป็นศูนย์ก่อนที่ดาต้าแกรมเหล่านั้นจะเดินทางไปถึงปลายทางสุดท้าย ดาต้าแกรมทั้งหมดอายุขัยนี้จะถูกทิ้งไป ซึ่งกระบวนการดังกล่าวเป็นการป้องกันมิให้ดาต้าแกรมที่มีปัญหา ส่งข้อมูลวนเวียนอยู่ในเครือข่ายนั่นเอง

**Protocol:** เป็นฟิลด์ที่ใช้ระบุชนิดของโปรโตคอลในลำดับชั้นส่วนบน (Upper-Layer) เพื่อจะได้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวกำหนดว่าจะส่งให้กับ TCP หรือ UDP

**Header Checksum:** เป็นฟิลด์ขนาด 16 บิตที่ใช้สำหรับตรวจสอบความถูกต้องสมบูรณ์ของ เฮดเดอร์ โดยการตรวจสอบในที่นี้จะทำเฉพาะในส่วนของเฮดเดอร์เท่านั้น ไม่ได้รวมแพ็กเก็ตข้อมูล

**Source Address** คือ หมายเลข IP ของโฮสต์ต้นทาง

**Destination Address** คือ หมายเลข IP ของโฮสต์ปลายทาง

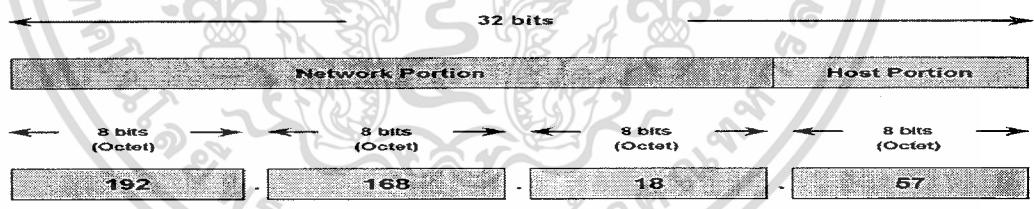
**Options** เป็นส่วนเพิ่มเติม ในกรณีที่ต้องการให้กำหนดหน้าที่เพิ่มเติมให้กับไอพีดาต้าแกรม โดยอาจมีหน้าที่เกี่ยวกับการควบคุมเส้นทาง เวลา การจัดการ และวางแผนทาง เป็นต้น

**Padding:** เป็นการเติม 0 ที่เพิ่มให้กับส่วนหัวของแพ็กเก็ต เพื่อให้ส่วนหัวมีความยาวที่หารด้วย 32 บิตลงตัว

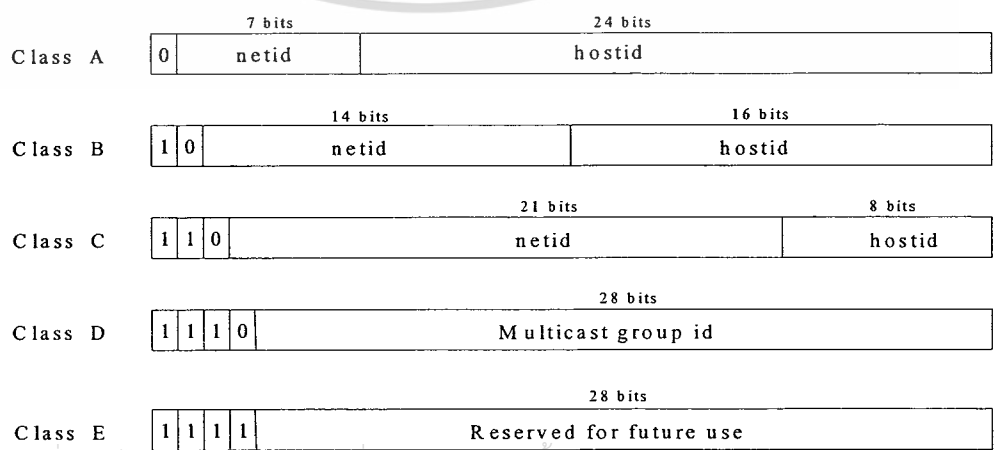
**2.3.3.1 การกำหนดตำแหน่งที่อยู่ (Addressing)**

หน้าที่นี้หมายถึง การให้บริการในการตั้ง “ลอจิกัลแอดเดรส (Logical Address)” ให้กับเครื่องคอมพิวเตอร์ต่างๆ ที่โปรโตคอล IP เนื่องจาก ลอจิกัลแอดเดรสนี้จะไม่ได้ถูกกำหนดมาตายตัวหรือฝังมากับเน็ตเวิร์กการ์ดดังนั้นมันจึงเป็นแอดเดรสที่ผู้ออกแบบหรือผู้บริหารเครือข่ายเป็นผู้ตั้งขึ้นมาเองและสามารถเปลี่ยนแปลงแก้ไขได้ตามใจชอบ

ไอพีแอดเดรสจะประกอบด้วย 4 ไบต์ (32 บิต) โดยปกติจะประกอบด้วย 2 ส่วนหลักๆ คือ ส่วนของหมายเลขเครือข่าย (NetID) และส่วนของหมายเลขโฮสต์ (HostID) โดยจะใช้เลขฐานสิบแทนด้วยเลขฐานสองทั้งหมด 8 ตัวแล้วกันด้วยจุด ตามรูปที่ 2.10



รูปที่ 2.10 โครงสร้างของ IP Address



รูปที่ 2.11 โครงสร้าง Class ของ IP Address

โดยในช่วงแรกๆของการเริ่มใช้งาน จะแบ่ง IP Address ออกเป็นคลาสด้วยกันทั้งหมด 5 คลาสด้วยกันดังรูปที่ 2.11

ปัจจุบันมีรูปแบบของคลาสที่ใช้งานอยู่ 5 ชนิดด้วยกัน ซึ่งแต่ละคลาสได้ออกแบบมาเพื่อรองรับความต้องการที่แตกต่างกันตามแต่ละองค์กร โดยคลาส A และคลาส B ได้ถูกนำมาใช้งานเต็มหมดแล้ว ดังนั้นในปัจจุบันจึงมีเพียงแต่คลาส C ที่ใช้งานอยู่ทั่วไปเพียงอย่างเดียว ในขณะที่ คลาส D และคลาส E ได้ถูกสงวนการใช้งานไว้ โดยคลาส D สงวนไว้สำหรับ มัลติคาสต์แอดเดรสซึ่งจะมีช่วง Address ในการใช้งานดังรูปที่ 2.12

Class	High Order Bits	Start	End
Class A	0	0.0.0.0	127.255.255.255
Class B	10	128.0.0.0	191.255.255.255
Class C	110	192.0.0.0	223.255.255.255
Multicast	1110	224.0.0.0	239.255.255.255
Experimental	1111	240.0.0.0	255.255.255.255

รูปที่ 2.12 ช่วงในการใช้งาน IP Address ในแต่ละ Class

### 2.3.3.2 ชั้นเน็ตมาส์ (Subnet Mask)

Subnet Mask เป็นพารามิเตอร์อีกตัวหนึ่งที่ต้องระบุควบคู่กับหมายเลข IP Address หน้าที่ของ Subnet Mask ก็คือ การช่วยในการแยกแยะว่าส่วนใดภายในหมายเลข IP Address เป็น Network ID และส่วนใดเป็น Host ID โดยจะใช้บิต 1 แทนส่วนที่เป็น Network ID และ บิต 0 แทนส่วนที่เป็น Host ID

	1st Octet	2st Octet	3st Octet	4st Octet	Subnet Mask
Class A	Network	Host	Host	Host	255.0.0.0 or /8
Class B	Network	Network	Host	Host	255.255.0.0 or /16
Class C	Network	Network	Network	Host	255.255.255.0 or /24

รูปที่ 2.13 การใช้งาน Subnet Mask

### 2.3.4 โพรโทคอลอื่นๆ ในลำดับชั้นเน็ตเวิร์ก

นอกจาก IP ซึ่งจัดเป็นโพรโทคอลสำคัญในลำดับชั้นเน็ตเวิร์กแล้ว ในลำดับชั้นนี้ยังสนับสนุนโพรโทคอลอีก 4 ชนิดด้วยกัน คือ โพรโทคอล ARP, RARP, ICMP และ IGMP

#### 2.3.4.1 โพรโทคอล ARP (Address Resolution Protocol)

ARP เป็นโพรโทคอลที่ใช้ในการค้นหาหมายเลข MAC Address ของเครื่องโฮสต์ปลายทางที่เครื่องโฮสต์ต้นทางต้องการส่งแพ็กเก็ตไปหาซึ่งโพรโทคอล ARP นี้จะทำงานร่วมกับกับโพรโทคอล IP ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยการทำหน้าที่เปลี่ยนหมายเลขไอพีให้เป็นหมายเลข MAC Address เมื่อใดก็ตามที่โฮสต์ต่างๆ หรือเราท์เตอร์ ต้องการค้นหาฟิสิคัลแอดเดรสบนเครือข่าย ก็จะมีการตรวจสอบข้อมูลดังกล่าวในหน่วยความจำแคชของ ARP ก่อนว่า ชุดหมายเลข IP นี้ตรงกับหมายเลขแมคแอดเดรสของเครื่องปลายทางเครื่องใด และหากไม่มีหมายเลขไอพีในหน่วยความจำแคช ก็จะมีกลไกที่เรียกว่า ARP Request ซึ่งจะใช้หมายเลข IP ของปลายทาง บรอดคาสต์ (Broadcast) ไปบนเครือข่าย

#### 2.3.4.2 โพรโตคอล RARP (Reverse Address Resolution Protocol)

โพรโตคอล RARP จะมีการทำงานที่คล้ายคลึงกับโพรโตคอล ARP โดยจะทำงานในลักษณะตรงกันข้าม ด้วยการแปลงหมายเลข MAC Address ให้เป็นหมายเลข IP ซึ่งโพรโตคอล RARP นี้ออกแบบมาเพื่อใช้งานกับคอมพิวเตอร์ที่ปราศจากดิสก์หรือฮาร์ดดิสก์ (Diskless Computer) ดังนั้นเวลาบูทเครื่องจึงจำเป็นต้องบูทจากระบบปฏิบัติการเครือข่ายของเซิร์ฟเวอร์บนเครือข่าย โดยเซิร์ฟเวอร์บนเครือข่ายจะทำการจัดเก็บตารางความสัมพันธ์ระหว่าง MAC Address กับหมายเลข IP โฮสต์ที่ต้องการหมายเลข IP จะทำการบรอดคาสต์ RARP Query Packet ที่บรรจุฟิสิคัลแอดเดรสไปยังทุกๆ โฮสต์บนเครือข่าย จากนั้นเครื่องเซิร์ฟเวอร์บนเครือข่ายก็จะจัดการเก็บ RARP Packet ด้วยการตอบกลับไปด้วยหมายเลข IP ไปยังโฮสต์นั้นๆ

#### 2.3.4.3 โพรโตคอล ICMP (Internet Control Message Protocol)

โพรโตคอล ICMP เป็นโพรโตคอลที่ช่วยในการแจ้งสถานะหรือรายงานข้อผิดพลาดที่เกิดขึ้นกับการทำงานของโพรโตคอล TCP/IP ให้กับโฮสต์ต้นทางหรือเราท์เตอร์ทราบ อย่างไรก็ตาม โพรโตคอล ICMP นั้นเพียงแต่แจ้งรายงานปัญหาหรือสถานะต่างๆ เท่านั้น แต่ไม่ได้ช่วยแก้ปัญหาให้โดยตรง แต่ข้อความต่างๆ ที่มันแจ้งให้ทราบจะมีประโยชน์ต่อผู้ดูแลระบบเครือข่ายในการวิเคราะห์ปัญหาต่างๆ ที่เกิดขึ้นได้ง่ายขึ้น โดยทั้งโฮสต์ต้นทาง ปลายทาง หรือเราท์เตอร์มีสิทธิ์ส่งแพ็กเก็ต ICMP ออกมาได้ทั้งสิ้น

#### 2.3.4.4 โพรโตคอล IGMP (Internet Group Message Protocol)

โพรโตคอล IP ยังสามารถแบ่งการสื่อสารออกเป็นสองชนิดด้วยกันคือ แบบยูนิคาสติง (Unicasting) และ แบบมัลติคาสติง (Multicasting) โดยยูนิคาสติงเป็นการสื่อสารระหว่างฝ่ายส่งและฝ่ายรับ ซึ่งเป็นการสื่อสารแบบหนึ่งต่อหนึ่ง (One-to-One Communication) อย่างไรก็ตาม ในบางครั้งอาจจำเป็นต้องมีการส่งแมสเสจเดียวกันนี้ไปยังผู้รับหลายๆ คน หรือหลายๆ กลุ่ม จึงจำเป็นต้องใช้วิธีการส่งแบบมัลติคาสติง ซึ่งเรียกว่าการสื่อสารแบบหนึ่งต่อกลุ่ม (One-to-Many Communication) โดยที่โพรโตคอล IGMP ทำหน้าที่ในการส่ง UDP คาต้าแกรมไปยังกลุ่มของโฮสต์ หรือโฮสต์หลายๆ โฮสต์พร้อมกัน เช่น การประชุมผ่าน Video Conference

#### 2.3.5 โพรโตคอลในระดับชั้นแอปพลิเคชัน

โพรโตคอลในลำดับชั้นเน็ตเวิร์คและทรานสปอร์ต เป็นเรื่องภายในของกระบวนการในด้านของการส่งและลำเลียงข้อมูลไปยังปลายทาง ผู้ใช้งานไม่ได้เข้าไปข้องเกี่ยวแต่อย่างใด ในลำดับชั้นแอปพลิเคชันไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นหรือชั้นประยุกต์ต่างหาก ที่ผู้ใช้งานต่างๆ จะใช้สำหรับติดต่อ หรือเพื่อการอินเทอร์เน็ตเฟซกับคอมพิวเตอร์ ซึ่งในลำดับชั้นแอปพลิเคชันนั้น ได้มีการจัดเตรียมโปรโตคอลต่างๆ มากมายเพื่อการสนับสนุนการบริการให้กับผู้ใช้งาน โปรโตคอลที่สำคัญๆ ดังรายละเอียดต่อไปนี้

#### 2.3.5.1 DHCP (Dynamic Host Configuration Protocol)

เป็นโปรโตคอลที่ใช้กำหนดหรือจัดสรรหมายเลขไอพีแอดเดรสให้แก่โฮสต์บนเครือข่ายแบบไดนามิก ซึ่งเป็นแบบไม่ตายตัว กล่าวคือ ในการติดต่อของโฮสต์เดียวกันในแต่ละครั้ง โฮสต์นั้นอาจได้หมายเลขไอพีแอดเดรสที่แตกต่างกันได้ โดยเครือข่ายที่ต้องการบริการ DHCP จำเป็นต้องมี DHCP เซิร์ฟเวอร์ เพื่อให้เครื่องไคลเอนต์ขอใช้บริการ

#### 2.3.5.2 DNS (Domain Name System)

เป็นบริการแปลงชื่อโดเมนให้เป็นหมายเลขไอพีแอดเดรส ซึ่งปกติเครื่องคอมพิวเตอร์จะมีหมายเลข IP เป็นหมายเลขประจำตัวที่ใช้ในการติดต่อ แต่หมายเลขดังกล่าวยากต่อการจดจำ ดังนั้นจึงนิยมตั้งเป็นชื่อที่เป็นตัวอักษรขึ้นมาแทนเพื่อง่ายต่อการจดจำ และ DNS ก็คือบริการที่จะทำหน้าที่แปลงชื่อนี้ให้เป็นหมายเลขไอพีแอดเดรส

#### 2.3.5.3 TELNET

เป็นโปรโตคอลที่ใช้สำหรับการล็อกอินเพื่อเชื่อมต่อเครือข่ายแบบระยะไกล (Remote Login) เช่น ยูสเซอร์ที่มีบัญชีหรือ Account อยู่บนเซิร์ฟเวอร์ โดยขณะนั้นอยู่ต่างสถานที่ แต่ต้องการที่จะรีโมทล็อกอินเข้ามาเพื่อใช้งานโดยไม่ต้องเดินทางมายังศูนย์ ก็สามารถทำได้ด้วยการเทลเน็ตเข้ามาเพื่อเชื่อมต่อใช้งานกับโฮสต์หรือเซิร์ฟเวอร์ซึ่งเปรียบเสมือนเป็นเทอร์มินัลหนึ่งที่เชื่อมโยงเข้ากับเซิร์ฟเวอร์โดยตรง

#### 2.3.5.4 FTP (File Transfer Protocol)

เป็นโปรโตคอลที่บริการด้านการโอนถ่ายแฟ้มข้อมูลระหว่างโฮสต์ เช่น เครื่องไคลเอนต์ได้มีการขอใช้บริการเพื่อการถ่ายโอนข้อมูลจากเซิร์ฟเวอร์มายังเครื่องของตน เป็นต้น

#### 2.3.5.5 SMTP (Simple Mail Transfer Protocol)

เป็นโปรโตคอลที่ใช้สำหรับบริการไปรษณีย์อิเล็กทรอนิกส์บนอินเทอร์เน็ต

#### 2.3.5.6 POP (Post Office Protocol)

เป็นโปรโตคอลแบบไคลเอนต์เซิร์ฟเวอร์ ระหว่างผู้ใช้งานบนเวิร์กสเตชันและเมลเซิร์ฟเวอร์

#### 2.3.5.7 HTTP (Hyper Text Transfer Protocol)

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนวิชาคอมพิวเตอร์กราฟิก ซึ่งจัดเป็นตัวอย่างในการรับส่งข้อมูล ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

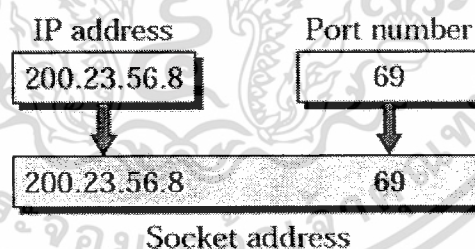
ระหว่างโปรแกรม บราวเซอร์และเว็บเซิร์ฟเวอร์

### 2.3.6 หมายเลขพอร์ต (Port Number)

เป็นค่าตัวเลขที่บ่งบอกให้ทราบถึงชื่อของแอปพลิเคชัน หรือชื่อของเซิร์ฟเวอร์ที่ทำงานอยู่ในระดับแอปพลิเคชันเลเยอร์ หมายเลขพอร์ตจะเก็บอยู่ในทั้งฟิลด์ TCP/UDP Source Port และ TCP/UDP Destination Port ซึ่งอยู่ในส่วนของ TCP และ UDP Header โดยที่ในระหว่างติดต่อกันจะใช้ IP Address และเมื่อมีการรวม IP Address กับหมายเลขช่องสื่อสารหรือพอร์ตแล้ว จะเรียกว่า ซ็อกเก็ต (Socket) พอร์ต แต่ละหมายเลขจะถูกกำหนดด้วยหมายเลขแอดเดรสที่เป็นเลขจำนวนเต็มที่มีขนาด 16 บิตจำนวนบิตดังกล่าว ถือว่ามีเพียงพอต่อการนำไปใช้งาน เนื่องจากขนาด 16 บิตนี้สามารถสนับสนุนพอร์ตได้มากถึง 65,536 พอร์ตด้วยกัน (0 – 65535) โดยหมายเลขพอร์ตตั้งแต่ 0 – 1023 นี้จะถูกสงวนเพื่อการใช้งานมาตรฐานของโฮสต์ที่เป็นเซิร์ฟเวอร์ ที่เรียกว่า Well-Know Ports ในขณะที่เครื่องไคลเอนต์ที่ต้องการขอติดต่อกับเซิร์ฟเวอร์นั้น จะใช้หมายเลขพอร์ตที่มีค่าตั้งแต่ 1024 เป็นต้นไป โดยตัวซอฟต์แวร์ TCP จะเป็นตัวกำหนดหมายเลขพอร์ตที่ยังว่างอยู่ให้ซึ่งจะต้องไม่ซ้ำกันกับที่มีการใช้งานอยู่ของเครื่องไคลเอนต์นั้นๆ

สำหรับการมัลติเพล็กซ์ในลำดับชั้นนี้จะใช้แนวคิดที่เรียกว่า ซ็อกเก็ต ซึ่งเป็นตัวที่ใช้เพื่อแยกแยะโปรเซสของฝ่ายต้นทางกับฝ่ายปลายทางที่ติดต่อกันและทำให้สามารถเปิดเซสชันหลายเซสชันพร้อมกันได้โดยซ็อกเก็ตจะประกอบด้วย 3 สิ่งด้วยกัน คือ [หมายเลข IP, โพรโตคอลทรานสปอร์ต, หมายเลขพอร์ต]

### Socket address



รูปที่ 2.14 ตัวอย่าง Socket Address

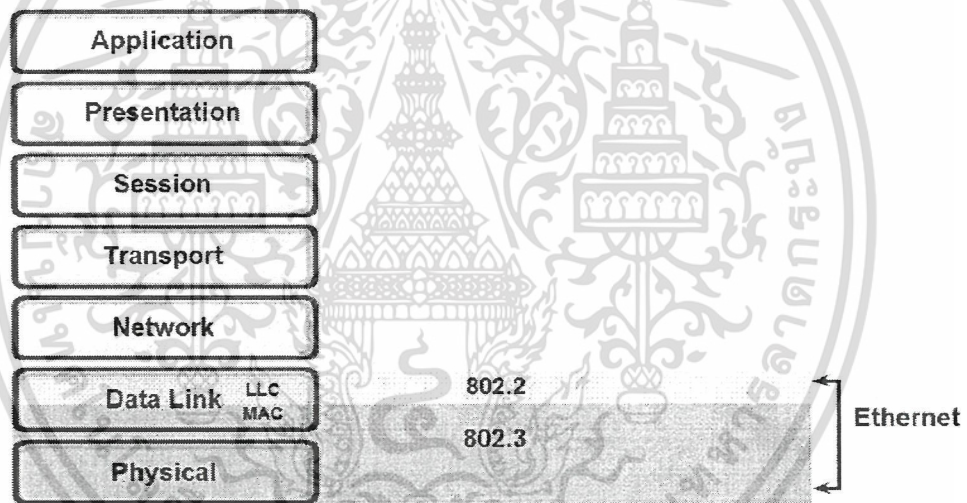
### 2.4 มาตรฐาน IEEE 802.3

ในปีค.ศ.1973 Bob Metcalfe ได้คิดค้นระบบอีเทอร์เน็ตในการรับส่งข้อมูลระหว่างคอมพิวเตอร์ และสามารถส่งข้อมูล ไปยังเครื่องพิมพ์ได้ หลังจากนั้นอีเทอร์เน็ตได้ถูกพัฒนาต่อที่ศูนย์วิจัยของบริษัทซีรอกซ์ (Xerox) ในช่วงแรกนั้นอีเทอร์เน็ตเป็นลิขสิทธิ์ของบริษัทซีรอกซ์ บริษัทเดียวเท่านั้น ต่อมา มาตรฐานอีเทอร์เน็ตที่มีความเร็ว 10 Mbps ได้ประกาศใช้เมื่อปี ค.ศ.1980 โดยความร่วมมือของ 3 บริษัท คือ DEC , Intel , Xerox หรือเรียกสั้นๆว่า DIX และในขณะเดียวกัน IEEE ก็ได้พัฒนามาตรฐานอีเทอร์เน็ตเช่นกัน คือ มาตรฐาน IEEE 802.3 ซึ่งได้พัฒนามาจากมาตรฐานอีเทอร์เน็ตของ DIX อีกทีหนึ่ง มาตรฐาน IEEE ถูกตีพิมพ์ครั้งแรกในปี 1985 และต่อมา ISO (International Standard for Organization) ก็ได้ยอมรับ

เอกสารนี้เป็นเอกสารต้นฉบับที่สงวนลิขสิทธิ์ไว้การใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่สามารถนำไปใช้ประโยชน์อื่นใดได้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร หากมีการนำออกไปใช้โดยไม่ได้รับอนุญาต เจ้าของเอกสารขอสงวนสิทธิ์ในการฟ้องดำเนินคดีตามกฎหมายที่เกี่ยวข้อง

เอามาตรฐาน IEEE 802.3 นี้เป็นมาตรฐานอีเธอร์เน็ตนานาชาติ ทำให้บริษัทใดก็ได้สามารถผลิตอุปกรณ์อีเธอร์เน็ตได้โดยไม่ต้องเสียค่าลิขสิทธิ์ให้กับใคร หลังจากนั้นทำให้การใช้งานอีเธอร์เน็ตแพร่ไปทั่วโลกอย่างรวดเร็ว จนกลายมาเป็นเทคโนโลยีเครือข่ายยอดนิยมในปัจจุบัน

IEEE ได้พัฒนามาตรฐานระบบ LAN (Local area network) ต่างๆ ที่ทำงานในระดับชั้น Physical และ Data Link Layers ตามที่ได้กำหนดไว้ในแบบจำลอง OSI Model สำหรับมาตรฐาน 802.3 นั้น Data Link Layer ได้ถูกแบ่งออกเป็น 2 ชั้นย่อย (Sub Layer) ซึ่งได้แก่ Logical Link Control (LLC) และ Medium Access Control (MAC) โดยชั้นย่อย LLC จะทำหน้าที่ควบคุมให้โพรโทคอลต่างๆที่อยู่ในระดับชั้นที่สูงกว่าสามารถใช้ Data Link Layer ร่วมกันได้ ส่วนชั้นย่อย MAC นั้นจะทำหน้าที่ควบคุมการลำเลียงเฟรม (frame) จากชั้นย่อย LLC ที่เครื่องท้องถิ่น (Local) ให้ถึงชั้นย่อย LLC ของเครื่องที่อยู่ห่างไกล (Remote) โดยได้มีการกำหนดรูปแบบเฟรมและวิธีการควบคุมป้องกันการรับข้อมูลที่ผิดพลาด (Error Control) ด้วย ดังโครงสร้างรูปที่ 2.15



รูปที่ 2.15 โครงสร้างของ Ethernet

#### 2.4.1 LLC (Logical Link Control)

LLC (Logical Link Control) เป็นเลเยอร์ที่อยู่ด้านบนของดาต้าลิงก์เลเยอร์ ซึ่งจะใช้ให้บริการกับโพรโทคอลของเลเยอร์บนในการเข้าใช้สื่อกลางหรือสายสัญญาณในการรับส่งข้อมูล ตามมาตรฐาน IEEE 802.2 แล้วจะอนุญาตให้สถาปัตยกรรมของ LAN ที่ต่างกันสามารถทำงานร่วมกันได้ กล่าวคือโพรโทคอลเลเยอร์บนไม่จำเป็นต้องทราบว่ามีสื่อกลางเลเยอร์ใช้สายสัญญาณประเภทใดในการรับส่งข้อมูล เพราะ LLC จะรับผิดชอบแทนในการปรับเฟรมข้อมูลให้สามารถส่งไปได้ในสายสัญญาณประเภทนั้น ๆ LLC เป็นเลเยอร์ที่แยกชั้นเครือข่าย (Network Layer) ออกจากการเปลี่ยนแปลงบ่อย ๆ ของสถาปัตยกรรมของ LAN โดยโพรโทคอลของเลเยอร์ที่สูงกว่าไม่จำเป็นต้องสนใจว่าแพ็กเก็ตจะส่งผ่านเครือข่ายแบบอีเธอร์เน็ต โทเคนริง หรือ ATM และไม่จำเป็นต้องรู้ว่าการส่งผ่านข้อมูลในชั้นกายภาพจะใช้การรับส่งข้อมูลแบบใด ชั้น LLC จะจัดการเรื่องเหล่านี้ได้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4.2 MAC (Media Access Control)**

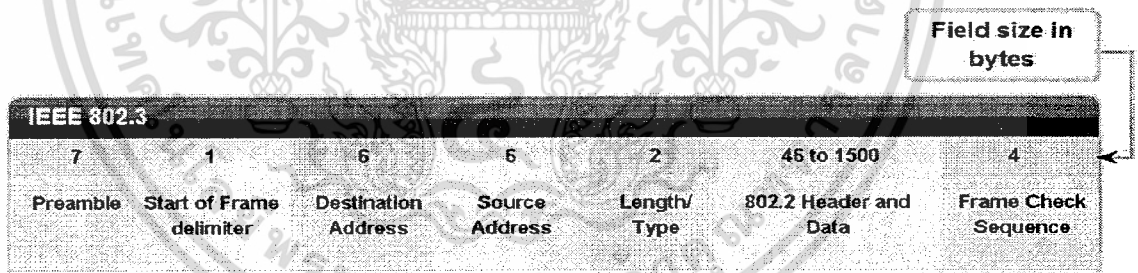
MAC (Media Access Control) เป็นเลเยอร์ย่อยที่อยู่ล่างสุดของดาต้าลิงก์เลเยอร์ ซึ่งจะทำหน้าที่เชื่อมต่อกับฟิสิคอลละเยอร์ และรับผิดชอบในการรับส่งข้อมูลให้สำเร็จและถูกต้อง โดยจะแบ่งหน้าที่ออกเป็นสองส่วนคือ การส่งข้อมูลและการรับข้อมูล

MAC จะทำหน้าที่ห่อหุ้มข้อมูลที่ส่งผ่านจากชั้น LLC และทำให้อยู่ในรูปแบบเฟรมข้อมูล ซึ่งเฟรมข้อมูลนี้จะประกอบด้วยที่อยู่ (Address) และข้อมูลต่าง ๆ ที่จำเป็นสำหรับการส่งข้อมูลให้ถึงปลายทาง ชั้น MAC ยังรับผิดชอบในการสร้างกลไกสำหรับตรวจสอบข้อผิดพลาดของข้อมูลในเฟรมนั้น ๆ ในระหว่างการรับส่งเฟรมด้วย นอกจากนี้ MAC ยังต้องตรวจสอบชั้นกายภาพว่าช่องสัญญาณพร้อมสำหรับการส่งข้อมูลหรือไม่ ถ้าพร้อมเฟรมก็จะถูกส่งต่อไปยังชั้นกายภาพเพื่อทำการส่งไปตามสายสัญญาณต่อไป แต่ถ้ายังไม่พร้อมชั้น MAC ก็จะรอจนกว่าจะว่าง แล้วค่อยทำการส่งข้อมูล

หน้าที่สุดท้ายของชั้น MAC คือ การตรวจสอบสถานะภาพของเฟรมที่กำลังส่ง ว่ามีการชนกันของข้อมูลเกิดขึ้นหรือไม่ ถ้าหากมีการชนกันเกิดขึ้นก็หยุดการส่งข้อมูล และเข้าสู่กลไกการรอด้วยช่วงเวลาที่เป็นเลขสุ่มเพื่อการส่งข้อมูลใหม่อีกครั้ง ซึ่งจะทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะทำการส่งข้อมูลได้สำเร็จ

**2.4.3 โครงสร้างเฟรมข้อมูล**

ข้อมูลที่อยู่ในเลเยอร์ที่สองนั้นจะจัดอยู่ในรูปแบบของเฟรมข้อมูล ซึ่งตามมาตรฐาน IEEE 802.3 แล้วฟอร์แมตของเฟรมข้อมูลจะแสดงในรูปที่ 2.16



รูปที่ 2.16 โครงสร้างของเฟรม

รายละเอียดของฟิลด์ต่าง ๆ ของอีเธอร์เน็ตที่มีรายละเอียดดังนี้

**Preamble:** เป็นฟิลด์ที่มีบิตสลับกันระหว่าง 1 กับ 0 ซึ่งเป็นสัญญาณบอกฝ่ายรับว่ากำลังมีข้อมูลส่งมา ฟิลด์นี้มีความยาว 8 ไบต์ โดยรวมเอาไบต์ของ SOF เข้าด้วย

**Start of Frame delimiter :** เป็นไบต์สุดท้ายพรีแอมเบิล ซึ่งไบต์จะแตกต่างจากไบต์อื่น ๆ คือ 2 บิตสุดท้ายจะเป็นเลข 1 ทั้งคู่ เพื่อเป็นสัญญาณสำหรับบอกจุดเริ่มต้นของเฟรมจริง ๆ

**Destination/Source Address:** หมายเลขหรือที่อยู่ของสถานีปลายทางและต้นทางซึ่งมีความยาว 48 บิต ส่วนใหญ่จะเรียกว่า "แมคแอดเดรส (MAC address)" การกำหนดหมายเลขนี้จะควบคุมโดย IEEE ซึ่งมีเกณฑ์ คือ 24 บิตเป็นหมายเลขที่กำหนดให้กับบริษัทผู้ผลิต NIC (Network Interface Card) และบริษัทนั้นจะเป็นคนที่กำหนดอีก 24 บิตที่เหลือ การทำเช่นนี้เพื่อให้แน่ใจว่าในเครือข่ายหนึ่ง ๆ จะไม่มี

เอกสารนี้เป็นเอกสารที่วางแปลสำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้เพื่อประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขนี้ซ้ำกัน

**Type/Length:** เป็นฟิลด์ที่บอกประเภทของโปรโตคอลของเลขที่อยู่เหนือกว่า เช่น IP, IPX, IPv6, ARP, Apple Talk เป็นต้น

**Data:** ส่วนนี้จะเป็นฟิลด์ที่เก็บข้อมูลซึ่งความยาวอย่างน้อยต้องไม่ต่ำกว่า 46 ไบต์ ถ้าต่ำกว่านี้จะต้องเสริมฟิลด์ Padding เพื่อให้ข้อมูลมีขนาดอย่างน้อย 46 ไบต์ เหตุที่ต้องกำหนดความยาวขั้นต่ำระดับนี้ก็เพื่อสำหรับการตรวจเช็คการชนกันของข้อมูล (Collision) ในระหว่างการรับส่งข้อมูลส่วนความยาวสูงสุดคือ 1,500 ไบต์

**FCS (Frame Check Sequence):** ฟิลด์นี้มีความยาว 4 ไบต์ ซึ่งเป็นโค้ดสำหรับตรวจสอบข้อผิดพลาดแบบ CRC (Cyclic Redundancy Check) ของข้อมูลในเฟรม

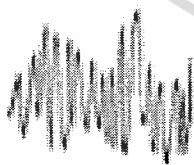
#### 2.4.4 โปรโตคอล CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ซึ่งเป็นโปรโตคอลที่รับส่งข้อมูลแบบฮาล์ฟดูเพล็กซ์ (Half-Duplex) โปรโตคอลนี้ใช้สำหรับการเข้าใช้สื่อกลางที่แชร์กันในการส่งสัญญาณระหว่างเครื่องในเครือข่าย ซึ่งมีขั้นตอนดังนี้ เมื่อเครื่องใด ๆ ต้องการที่จะส่งข้อมูลจะต้องคอยฟังก่อน (Carrier Sense) ว่ามีเครื่องอื่นกำลังส่งข้อมูลหรือไม่ ถ้ามีให้รอจนกว่าเครื่องอื่นส่งข้อมูลเสร็จก่อนแล้วค่อยเริ่มส่งข้อมูล และในขณะที่กำลังส่งข้อมูลอยู่นั้นต้องตรวจสอบว่ามีการชนกันของข้อมูลเกิดขึ้นหรือไม่ (Collision Detection) ถ้ามีการชนกันของข้อมูลเกิดขึ้นให้หยุดทำการส่งข้อมูลทันที แล้วค่อยเริ่มกระบวนการส่งข้อมูลใหม่อีกโดยใช้การสุ่มเวลาขึ้นมาในการรอการส่งใหม่ของแต่ละเครื่อง เมื่อครบตามที่กำหนดเวลาไว้ก็เริ่มทำการส่งข้อมูลไปใหม่อีกครั้ง

### 2.5 หลักการทำงานของเทคโนโลยี Voice over IP

การทำงานของระบบ VoIP แบ่งเป็นขั้นตอนหลักๆ ได้ดังนี้

#### 2.5.1 Conversion to PCM (Pulse Code Modulation)



01101110001010010001010110110010

รูปที่ 2.17 การแปลงสัญญาณ Analog ให้เป็น Digital

ในขั้นตอนแรกจะเป็นการแปลงสัญญาณ Analog ให้ไปอยู่ในรูปแบบสัญญาณ Digital หรือที่เรียกว่า PCM

#### 2.5.2 Removal of Echo

ขั้นตอนต่อไปนี้จะเป็นการมีการแยกสัญญาณออกเป็นส่วนๆ เพื่อทำการตัดสัญญาณ Echo ออก ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะเพื่อการศึกษานานาชาติเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กระบวนการนี้จะถูกจัดการโดย DSP (Digital Signal Processors) ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0110111000101001000101011011001001101001001011

รูปที่ 2.18 การกำจัด Echo

### 2.5.3 Framing

0110111000101001000101011011001001101001001

รูปที่ 2.19 การจัดรูปแบบเฟรม

ในส่วนของสัญญาณที่เหลือนั้น ก็จะถูกแบ่งและจัดรูปแบบขึ้นมาใหม่ในรูปของ Frame ซึ่งกระบวนการนี้จะถูกจัดการโดยรูปแบบการบีบอัดที่เรียกว่า CODEC หลังจากเสร็จกระบวนการนี้แล้ว Frame ของสัญญาณเสียงจะถูกสร้างขึ้น

### 2.5.4 Packetisation

RTP 0110111000101001000101011011001001101001001

รูปที่ 2.20 การแปลงเฟรมให้อยู่ในรูป Packet

ในกระบวนการนี้จะเป็นการแปลง Frame ของสัญญาณให้มาอยู่ในรูปของ Packet ซึ่งจะมีการเพิ่ม Header เข้าไปใน Packet โดยในส่วนของ Header นั้น ก็จะประกอบไปด้วยข้อมูลที่เรียกว่า Sequence Number และ Time Stamp หลังจากนั้น Packet นี้จะถูกส่งต่อไปที่ Host Processor

### 2.5.5 Address and Delivery

IP UDP RTP 0110111000101001000101011011001001101001001

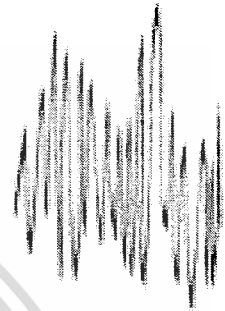
รูปที่ 2.21 การใส่ค่า IP Address ให้กับ Packet

หลังจากที่ได้แปลงสัญญาณให้อยู่ในรูปของ Packet แล้ว ข้อมูลนั้นจะถูกนำมาวิเคราะห์และใส่ค่า IP-Address ปลายทาง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.6 Conversion to Analog

หลังจากที่ได้ทำการใส่ค่าของ IP Address ปลายทางไปใน Header ของ Packet แล้วนั้นเมื่อ Packet เหล่านั้นไปถึงด้านปลายทาง ข้อมูล Header เหล่านี้จะถูกแยกออกเพื่อให้เหลือแค่ Voice Frame หลังจากนั้นก็จะทำการแปลงสัญญาณ Digital PCM ให้กลับมาเป็นสัญญาณรูปแบบ Analog ที่เป็นสัญญาณเสียงที่เราได้ยินกันอีกครั้งหนึ่ง

011011100010100100010101101



รูปที่ 2.22 การแปลงสัญญาณ Digital ให้กลับไปเป็น Analog

## 2.6 มาตรฐาน SIP (Session Initiation Protocol)

โพรโทคอล SIP เป็นโพรโทคอลระดับ Application ถูกพัฒนาโดย IETF (Internet Engineering Task Force) โดยออกแบบให้ใช้ในการตกลงกันระหว่างคู่สนทนาหรือระหว่างผู้ใช้บริการเพื่อทำการเริ่มต้นการโทร (Establish Call), เปลี่ยนแปลงรูปแบบการโทร (Modify Call) และสิ้นสุดการโทรหรือจบการสนทนา (End Call/Terminate Call) ดังนั้น โพรโทคอล SIP มีหน้าที่รับผิดชอบในส่วนการส่งสัญญาณควบคุม (Control Signaling) ก่อนและจบการสนทนาเท่านั้น นอกจากนี้การทำงานอย่างเป็นระบบตามมาตรฐาน SIP ทำให้เกิดความยืดหยุ่นและความสะดวกในการใช้งานแก่ผู้ใช้บริการมากยิ่งขึ้น

### 2.6.1 คุณสมบัติของ SIP

โพรโทคอล SIP เป็นโพรโทคอลระดับ Application ซึ่งอยู่เหนือโพรโทคอลระดับ Transport โดยอนุญาตให้สามารถทำการส่งสัญญาณโดยใช้โพรโทคอลระดับ Transport ได้ทั้ง TCP (Transmission Control Protocol) และ UDP (User Datagram Protocol)

รูปแบบสัญญาณที่นิยามตามมาตรฐาน SIP มีลักษณะเป็นข้อความ (Text-based) ซึ่งถูกเรียกว่า SIP Message โดยรูปแบบและไวยากรณ์ของสัญญาณมีลักษณะคล้ายกับรูปแบบสัญญาณของโพรโทคอล HTTP (Hypertext Transfer Protocol) ทำให้ผู้พัฒนาสามารถพัฒนาได้ง่าย และเหตุที่มีการใช้สัญญาณ SIP Message ตลอดขั้นตอนการส่งสัญญาณควบคุมรวมถึงลำดับขั้นตอนการส่งสัญญาณตามมาตรฐาน SIP นั้นเพราะไม่ซับซ้อนจึงทำให้โพรโทคอล SIP สามารถทำงานได้รวดเร็วกว่ามาตรฐาน H.323

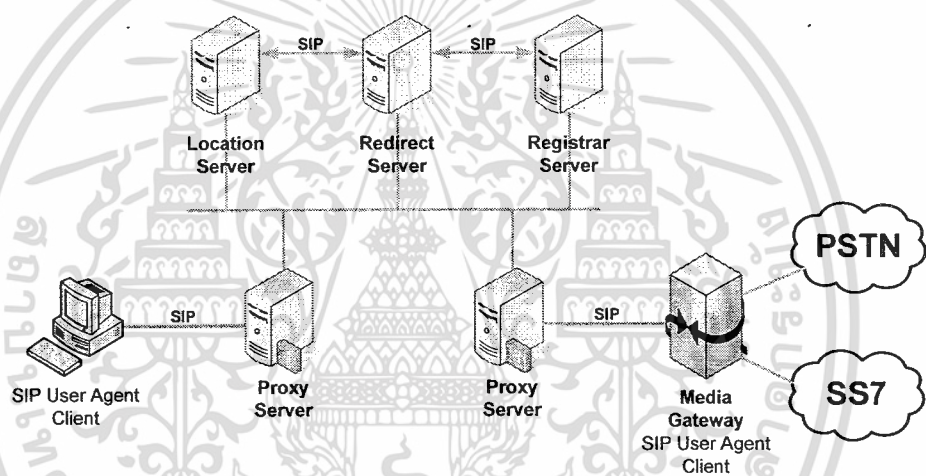
ตามมาตรฐานของ SIP จะรับผิดชอบขั้นตอนการส่งสัญญาณควบคุมในส่วนก่อนและหลังการสนทนาเท่านั้น โดยในขั้นตอนการส่งข้อมูลสื่อประสม (Media Stream) จะใช้โพรโทคอล RTP (Real

Time Transfer Protocol) และในขั้นตอนการแลกเปลี่ยนความสามารถในการส่งข้อมูลสื่อประสม (Media Capabilities Exchange) จะใช้โปรโตคอล SDP (Session Description Protocol) ทำงานร่วมด้วย

สถาปัตยกรรมตามมาตรฐาน SIP เป็นแบบ Client/Server โดยมีการอ้างอิงตัว SIP Client โดยใช้ชื่ออ้างอิงถึงที่อยู่ของตัว SIP Client นั้นๆ ซึ่งถูกเรียกว่า SIP URL (Uniform Resource Locators) ทำให้เกิดความสับสนและความยืดหยุ่นในการใช้บริการเพิ่มมากขึ้น

**2.6.2 องค์ประกอบสำคัญในระบบ SIP (SIP Component)**

จากรูปที่ 2.23 แสดงส่วนต่างๆ ที่ทำงานร่วมกันอย่างเป็นระบบ เนื่องจากโปรโตคอล SIP มีการทำงานแบบ Client/Server ดังนั้นจึงสามารถแบ่งออกเป็น 2 ส่วนหลัก คือ User Agent (SIP Client) และ Network Server (SIP Server)



รูปที่ 2.23 องค์ประกอบสำคัญในโปรโตคอล SIP

**2.6.2.1 User Agent**

เป็นนิยามของ SIP Client ภายในระบบ SIP โดยสามารถแยกเป็น 2 รูปแบบการทำงาน คือ

1. UAC (User Agent Client) ทำหน้าที่ในการสร้างหรือเริ่มต้นการโทรโดยส่งสัญญาณร้องขอ (Request Signal) ไปยังปลายทาง
2. UAS (User Agent Server) ทำหน้าที่ในการตอบรับการโทรและส่งสัญญาณตอบกลับ (Response Signal)

โดยปกติแล้ว SIP Client จะทำหน้าที่เป็นได้ทั้ง UAC และ UAS เพื่อให้สามารถทำงานได้ทั้งเป็นผู้โทร (Caller) และผู้ถูกเรียก (Callee)

**2.6.2.2 Network Server**

เป็นนิยามของ SIP Server ที่มีรูปแบบการทำงานแบบต่างๆ ได้แก่

1. Registrar Server ทำหน้าที่รับการลงทะเบียนจากผู้ใช้บริการ (SIP Client) เพื่อเก็บข้อมูลที่อยู่ที่ SIP URL และข้อมูลสำคัญของผู้ใช้บริการ

2. Proxy Server ทำหน้าที่เป็นตัวกลางในการส่งสัญญาณระหว่างผู้โทร (Caller) และผู้ถูกเรียก (Callee) โดยรับผิดชอบการส่งสัญญาณตลอดขั้นตอนการส่งสัญญาณตามมาตรฐาน SIP

3. Redirect Server ทำหน้าที่ระบุที่อยู่ของผู้ถูกเรียก (Callee) ที่ได้ทำการลงทะเบียนไว้ โดยจะส่งที่อยู่ที่ค้นหาได้กลับไปยังผู้โทร (Caller)

นอกจาก User Agent และ Network Server แล้วภายในระบบ SIP ยังมีส่วนการทำงานอื่นๆ ได้แก่

1. Location Server ทำหน้าที่ในการเก็บข้อมูลหรือเป็นฐานข้อมูลผู้ใช้บริการให้กับตัว Network Server ได้ ทำให้ป้องกันปัญหาเรื่องของขนาดฐานข้อมูลไม่เพียงพอและความปลอดภัยของข้อมูลได้ ตามมาตรฐาน SIP อนุญาตให้สามารถพัฒนาตัว Location Server ไว้เป็นตัวเดียวกันกับตัว SIP Server ได้

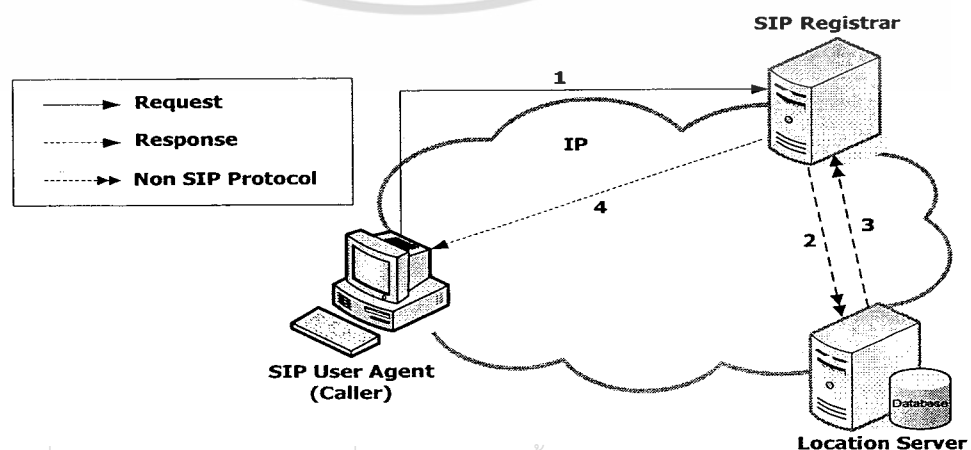
2. SIP Gateway ทำหน้าที่ในการแปลงสัญญาณและข้อมูลสื่อประสมระหว่างระบบเครือข่ายที่แตกต่างกัน เช่น การเชื่อมต่อไปยังระบบ PSTN (Public Switching Telephone Network), ระบบ SS7 (Signaling System 7) เป็นต้น

### 2.6.3 การทำงานของ SIP Server ในระบบ SIP

ลักษณะการทำงานของ SIP Server สามารถทำงานได้หลายรูปแบบ ดังนี้

#### 2.6.3.1 Registrar Server

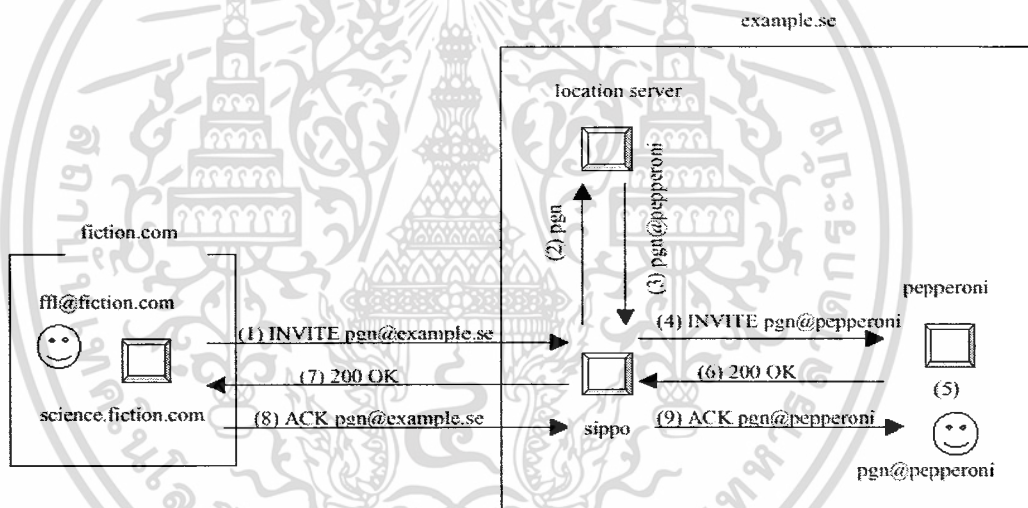
เป็น SIP Server ที่ทำหน้าที่ในการรับการลงทะเบียนจาก SIP Client โดยจะทำการเก็บข้อมูลทั้งหมดไว้ใน Location Server ซึ่งโดยทั่วไป Registrar Server ก็คือ SIP Server เมื่อทำหน้าที่รับการลงทะเบียนนั่นเอง ดังแสดงดังรูปที่ 2.24 เมื่อผู้ร้องขอการติดต่อทำการส่ง Request Message ไปยัง SIP Registrar Server ตัว SIP Registrar Server จะทำการติดต่อไปยัง Location Service ของ Location Server เพื่อถามหาที่อยู่ของผู้ถูกร้องขอการติดต่อโดยสัญญาณที่ใช้นี้ตามมาตรฐานของ SIP ไม่มีการกำหนดไว้ ดังนั้น SIP Registrar Server สามารถกำหนดรูปแบบของสัญญาณที่ใช้ติดต่อระหว่าง SIP Registrar Server กับ Location Server ได้เอง ต่อมาเมื่อ Location Server สามารถหาที่อยู่ได้แล้วก็จะส่งมาบอกกับ SIP Registrar Server สุดท้าย SIP Registrar Server จะส่ง Response Message ไปให้ผู้ร้องขอการติดต่อเพื่อบอกที่อยู่ของผู้ถูกร้องขอการติดต่อตามที่ผู้ร้องขอการติดต่อต้องการ ซึ่งแสดงได้ดังรูปที่ 2.24



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 2.24 การทำงานของ Registrar Server  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.2 Proxy Server

เป็น SIP Server ที่ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ร้องขอการติดต่อและผู้ถูกร้องขอการติดต่อ โดยการทำงานในขั้นตอน Call Setup, Media & Channel Control และ End Call การส่งสัญญาณทุกอย่างในขั้นตอนทั้งหมดนี้จะต้องทำผ่าน SIP Server โดย SIP Server จะทำการตรวจสอบ SIP Message ที่ได้รับทุกครั้งและจะทำการส่ง SIP Message นั้นไปยังปลายทางที่ต้องการ โดยอาจมีการเปลี่ยนแปลงรูปแบบของ SIP Message ตามความเหมาะสม สำหรับการทำงานในขั้นตอนการส่งข้อมูลเสียง (Media Stream) จะไม่ผ่าน SIP Server นั่นคือ SIP Client ทั้ง 2 ฝ่ายจะทำการส่งข้อมูลเสียงกันโดยตรง เมื่อมองในระดับ Logical แต่หากมองในระดับ Physical แล้ว SIP Client ทั้ง 2 ฝ่ายอาจมีการส่งข้อมูลผ่าน Server หรือ Gateway ก็ได้ เนื่องจากการทำงานทั้งหมดมีพื้นฐานการทำงานอยู่บนระบบเครือข่าย Internet ดังนั้นการส่งข้อมูลสามารถส่งผ่านแต่ละ โหนด (Node) เพื่อให้สามารถส่งข้อมูลไปยังปลายทางที่ต้องการได้



รูปที่ 2.25 การทำงานของ SIP Proxy Server

การทำงานของ SIP Proxy Server สามารถอธิบาย โดยพิจารณาตัวอย่างการทำงาน จากรูปที่ 2.25 ผู้ร้องขอการติดต่อมี SIP URL คือ ffl@fiction.com โดยมีที่อยู่จริงที่ science.fiction.com ซึ่งอยู่ในเครือข่ายของ fiction.com สำหรับผู้ถูกร้องขอการติดต่อมี SIP URL คือ pgn@example.se และมีที่อยู่คือ pgn@pepperoni อยู่ในเครือข่ายของ example.se ซึ่งมี SIP Server เป็นแบบ Proxy Server โดยมีชื่อว่า คือ sippo เมื่อผู้ร้องขอการติดต่อ ffl@fiction.com ต้องการติดต่อไปยัง pgn@pepperoni โดยผู้ถูกร้องขอการติดต่อนี้ได้ทำการ Register ไว้กับ SIP Server sippo แล้ว เริ่มต้นผู้ร้องขอการติดต่อจะทำการส่ง INVITE Message (1) ไปยัง SIP Server เพื่อบอกว่าต้องการติดต่อไปยังผู้ถูกร้องขอการติดต่อใด โดยทำการระบุ SIP URL ของผู้ถูกร้องขอการติดต่อด้วย นั่นคือ pgn@example.se เมื่อ SIP Server ได้รับ INVITE Message นี้แล้ว SIP Server จะทำการหาที่อยู่โดยตามจาก Location Server โดยส่งชื่อ (2) ของผู้ถูกร้องขอการติดต่อไปให้จากนั้น Location Server จะหาที่อยู่แล้วตอบกลับมาในที่นี้คือ pgn@pepperoni (3) จากนั้น

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาดูงานนี้ ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้วยการดัดแปลงหรือแก้ไขในลักษณะใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

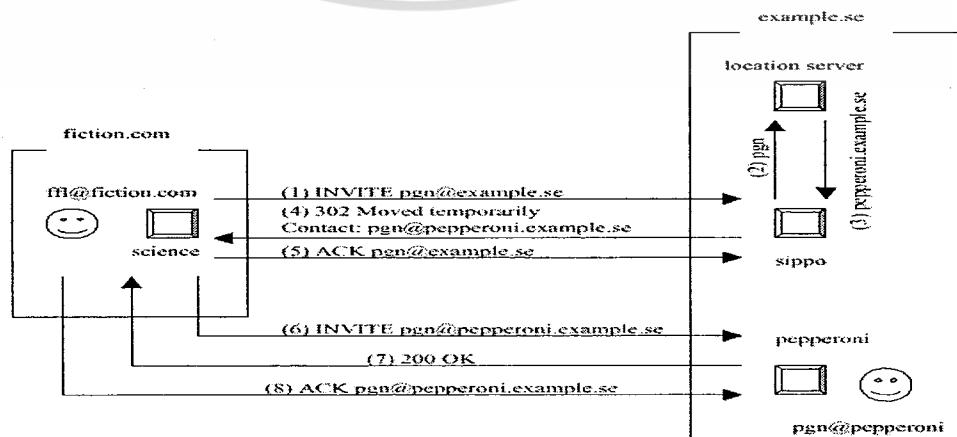
SIP Server จะทำการเปลี่ยนแปลง SIP Message บางอย่างเพื่อให้สามารถส่งไปยังปลายทางที่ pgn@pepperoni ได้ นั่นคือจะส่ง INVITE Message pgn@pepperoni (4) ไป เมื่อ pgn (5) ได้รับแล้วจึงตอบรับการติดต่อโดยส่ง Response Message เป็น Status Code 200 OK (6) กลับมายัง SIP Server จากนั้น SIP Server ก็ทำการส่งต่อ(7)ไปให้ ffn@fiction.com เพื่อให้รับรู้ว่าได้รับ Response Message นี้แล้วจึงต้องมีการส่ง ACK Message (8)ไปบอก

ด้วย SIP Server เมื่อได้รับ ACK Message ก็ทำการส่งต่อไปยัง pgn โดยทำการเปลี่ยนแปลง SIP Message เพื่อระบุที่อยู่ของ pgn นั่นคือ ACK pgn@pepperoni (9) หลังจากขั้นตอนนี้ SIP Client ทั้ง 2 ฝ่าย จะทำการส่งข้อมูลเสียงโดยไม่ผ่าน SIP Server

ดังจะเห็นได้ว่า Proxy Server จะทำหน้าที่ในการรับภาระหรือรับผิดชอบการติดต่อระหว่างผู้ร้องขอการติดต่อและผู้ถูกร้องขอการติดต่อ ทำให้ Proxy Server สามารถทำการเพิ่มเติมความสามารถต่างๆ เพื่อให้สามารถให้บริการ SIP Client ได้ดียิ่งขึ้น แต่อย่างไรก็ตามการทำงานในลักษณะนี้ทำให้ต้องสูญเสียเวลาในช่วงการติดต่อที่ทำโดยตัว Proxy Server และ Proxy Server สามารถให้บริการ SIP Client ได้จำกัด ขึ้นอยู่กับความสามารถของ Proxy Server นั้นๆ ที่จะสามารถรองรับการบริการกับ SIP Client ที่ทำการติดต่อได้มากน้อยเพียงไร

**2.6.3.3 Redirect Server**

เป็น SIP Server ที่ทำหน้าที่ในการระบุที่อยู่ของผู้รับการติดต่อที่ได้ทำการลงทะเบียนไว้ โดยจะทำการหาที่อยู่จาก Location Server ซึ่งเมื่อหาที่อยู่ได้แล้วจะส่งที่อยู่ทั้งหมดที่หาได้ให้กับผู้ร้องขอการติดต่อ ดังนั้นหลังจากที่ SIP Client ได้ที่อยู่ในการติดต่อแล้วจะต้องทำการติดต่อไปยังปลายทางที่ต้องการตามที่อยู่ที่ได้นั่นเอง โดยหากผู้ร้องขอการติดต่อได้ที่อยู่มากกว่า 1 ที่อยู่ ผู้ร้องขอการติดต่อจะต้องทำการติดต่อไปยังทุกๆ ที่อยู่เพื่อหาที่อยู่จริงๆ ของผู้ถูกร้องขอการติดต่อเอง ดังจะเห็นได้ว่าการทำงานในลักษณะนี้จะทำให้ภาระในการติดต่อทั้งหมดตกอยู่ที่ตัว SIP Client แต่การทำงานลักษณะนี้จะทำให้ SIP Client สามารถทำการติดต่อได้อย่างรวดเร็วในกรณีที่ได้ที่อยู่เดียวจาก SIP Server และตัว SIP Server สามารถให้บริการกับ SIP Client ได้จำนวนมาก ซึ่งแสดงได้ดังรูปที่ 2.26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 2.26 การทำงานของ Redirect Server  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Redirect Server สามารถอธิบายโดยพิจารณาตัวอย่างการทำงานจากรูปที่ 2.26 ผู้ร้องขอการติดต่อมี SIP URL คือ ffl@fiction.com อยู่ในเครือข่ายของ fiction.com ผู้ถูกร้องขอการติดต่อชื่อ pgn มีที่อยู่คือ pgn@pepperoni โดยมี SIP URL คือ pgn@example.se อยู่ในเครือข่าย example.se ซึ่งมี SIP Server แบบ Redirect Server ชื่อ sippo เมื่อผู้ร้องขอการติดต่อ ffl@fiction.com ต้องการติดต่อไปยัง pgn ผู้ร้องขอการติดต่อจะทำการติดต่อไปยัง Redirect Server เพื่อถามหาที่อยู่ของ pgn ดังนั้นจึงส่ง INVITE Message (1) ไปยัง Redirect Server จากนั้น Server จะทำการถามที่อยู่จาก Location Server โดยส่งชื่อ pgn (2) ไป เมื่อ Location Server สามารถหาที่อยู่ได้แล้วจะส่งที่อยู่ที่ได้กลับมาคือ pepperoni.example.se (3) จากนั้น Redirect Server จะส่งที่อยู่นี้ไปให้ผู้ร้องขอการติดต่อ ffl@fiction.com โดยส่งเป็น Response Message ซึ่งมี Status Code คือ 302 Moved temporarily (4) ซึ่งจะใช้สำหรับ SIP Server แบบ Redirect Server และเพื่อเป็นการยืนยันว่าได้รับที่อยู่แล้วผู้ร้องขอการติดต่อจึงทำการส่ง ACK Message (5) ไปยัง Redirect Server โดยขั้นตอนการติดต่อหลังจากนี้ผู้ร้องขอการติดต่อจะทำการติดต่อกับปลายทางที่ต้องการเอง โดยเริ่มตั้งแต่การส่ง INVITE Message (6) ไปยัง pgn ตามที่อยู่ที่ได้จาก Redirect Server หลังจากนั้น เมื่อ pgn ได้รับการติดต่อก็จะส่ง Response Message (7) กลับมา และเพื่อให้รู้ว่าได้รับการตอบรับแล้วจึงส่ง ACK Message (8) บอกไปยัง pgn หลังจากนั้นทั้ง 2 ฝ่ายก็จะทำการส่งข้อมูลเสียงตามลำดับ

ตารางที่ 2.1 คุณสมบัติของ Proxy Server และ Redirect Server

Proxy Server	Redirect Server
ทำงานได้ทั้งแบบ Stateless และ Stateful	สามารถรองรับการทำงานหรือให้บริการกับ SIP Client ได้จำนวนมาก
มีความยืดหยุ่นในการทำงานสูง และสามารถเพิ่มเติมการบริการได้ต้องการ	หน้าที่หรือความรับผิดชอบในการติดต่อจะขึ้นอยู่กับ SIP Client
รับผิดชอบการติดต่อในทุกขั้นตอนการติดต่อทั้งหมด	รับผิดชอบในการส่งที่อยู่ให้ผู้ร้องขอ จากนั้นผู้ร้องขอจะทำการติดต่อกับปลายทางเอง

2.6.4 การทำงานของ SIP ในระบบ Telephony

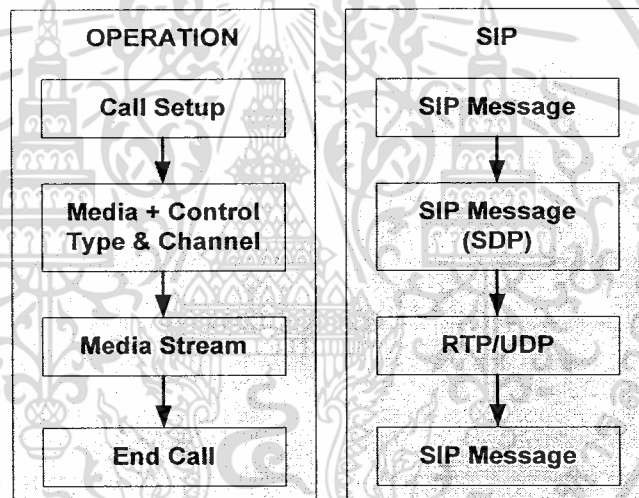
จากรูปที่ 2.27 แสดงการเปรียบเทียบขั้นตอนการทำงานของ SIP ในระบบ IP Telephony ซึ่งมีรายละเอียดดังนี้

1. Call Setup เป็นการเริ่มต้นการติดต่อ โดยผู้ร้องขอการติดต่อทำการติดต่อไปยังผู้ถูกร้องขอการติดต่อครั้งแรกเพื่อเรียกการสนทนา ในส่วนนี้ จะใช้ SIP Message ที่ชื่อว่า Message INVITE ทำการร้องขอการติดต่อไปยังปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Media & Channel Control เป็นการตกลงหรือทำการระบุข้อตกลงต่างๆ ที่ต้องใช้ในการส่งข้อมูลเสียง ได้แก่ มาตรฐานการบีบอัดข้อมูลเสียง, หมายเลข IP Address และหมายเลขพอร์ต (Port) เป็นต้น โดยในส่วนนี้ จะนำโปรโตคอล SDP (Session Description Protocol) มาช่วยในการตกลงรูปแบบของการติดต่อ

3. Media Stream เป็นขั้นตอนของการส่งข้อมูลเสียงโดยใช้ตามข้อตกลงที่ได้ตกลงกันในขั้นตอนที่ 2 Media & Channel Control โดยทั่วไปจะนิยมใช้ Real-Time Transport Protocol หรือ RTP ในการส่งข้อมูลเสียงในขั้นตอนนี้ ซึ่งขั้นตอนนี้ไม่ได้อยู่ในความรับผิดชอบของ SIP End Call เป็นขั้นตอนการส่งสัญญาณระหว่างผู้ร้องขอการติดต่อและผู้ถูกร้องขอการติดต่อเพื่อขอจบการติดต่อ โดยในส่วนนี้จะใช้ SIP Message ที่ชื่อว่า Message BYE เป็นสัญญาณในการสิ้นสุดการติดต่อ หลังจากจบขั้นตอนนี้ช่องทางของสัญญาณที่สร้างขึ้นทั้งหมดจะถูกปิด หากต้องการเริ่มสนทนาใหม่ จะต้องทำการติดต่อตามขั้นตอนเริ่มตั้งแต่ Call Setup ใหม่ทั้งหมด



รูปที่ 2.27 การทำงานของ SIP ในระบบ IP Telephony

### 2.6.5 ลำดับสัญญาณ SIP ในระบบ IP Telephony

ในหัวข้อนี้เราจะมารู้จักรูปแบบของลำดับสัญญาณของ SIP ที่จะใช้ในการทำการติดต่อกัน โดยจะมีการยกตัวอย่างการใช้งาน สัญญาณรูปแบบต่างๆ ที่สำคัญในการทำการติดต่อ

สัญญาณ SIP จะอยู่ในรูปของข้อความ (Message) โดย แบ่งออกเป็น 2 ประเภท คือ

#### 2.6.5.1 Request Message

เป็นสัญญาณ ใช้เพื่อทำการร้องขอ (Request) มีทั้งหมด 6 สัญญาณ คือ

- **INVITE** เป็นสัญญาณที่เฉพาะเจาะจงถึงผู้ที่ต้องการเชิญให้เข้าร่วมการสนทนา สำหรับการติดต่อที่มีเพียง 2 ฝ่าย ผู้ร้องขอการติดต่อจะบอกรายละเอียด เช่น ชนิดของมัลติมีเดียที่เฉพาะเจาะจงที่ตนเองมีความสามารถรองรับได้ พร้อมทั้ง Address และพอร์ตที่จะใช้

- **ACK** เป็นการยืนยันว่า Client ได้รับข้อความที่ Server ส่งมาให้แล้ว โดยจะใช้ควบคู่กับคำสั่ง INVITE รหัส 2xx จะเป็นรหัสที่ตอบกลับมาเพื่อให้ Client ยืนยันว่าได้รับแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **OPTION** เป็นคำสั่งเพื่อที่จะทำการสอบถามข้อมูลความสามารถของผู้ที่ต้องจะให้เข้าร่วมสนทนา แต่ไม่ใช่เป็นการ INVITE โดยผู้ที่รับคำสั่งนี้จะทำการตอบกลับมาโดยอธิบายถึงความสามารถของตนที่จะสามารถสนับสนุนการทำงานอะไรได้บ้าง

- **BYE** เป็นคำสั่งที่ใช้เพื่อที่จะขอจบการติดต่อระหว่างกัน ซึ่งคำสั่งนี้ทั้งผู้ที่ร้องขอการติดต่อและผู้ที่ถูกร้องขอการติดต่อ สามารถที่จะใช้คำสั่งนี้ได้ทั้งสอง ซึ่งถ้าฝ่ายใดเป็นฝ่ายที่ใช้คำสั่ง BYE ก่อนอีกฝ่ายก็จะต้องตอบกลับมาเพื่อยืนยันว่าได้รับข้อความเพื่อขอจบการติดต่อ

- **CANCEL** เป็นคำสั่งที่ใช้เพื่อที่จะขอยกเลิกข้อความบางข้อความที่ได้ส่งไปก่อนหน้านั้น แต่สำหรับข้อมูลที่มีความสมบูรณ์แล้ว ซึ่งก็คือมีการตอบสนองการการ Request เรียบร้อยแล้ว คำสั่ง CANCEL จะไม่มีผล ไม่ใช่แต่เพียง Client เท่านั้นที่จะใช้หรือรับคำสั่ง CANCEL ด้าน Server ก็มีการใช้คำสั่ง CANCEL ได้เช่นเดียวกัน เช่น ในกระบวนการที่ทำการติดต่อผู้ที่ถูกร้องขอการติดต่อในหลายๆ ที่แต่มีบางที่ที่ตอบกลับมาแล้ว Server ก็จะส่งคำสั่ง CANCEL เพื่อยกเลิกการติดต่อกับตัวอื่นที่ยังไม่ได้รับการตอบกลับมา

- **REGISTER** ใช้โดย Client เพื่อทำการลงทะเบียนโดย Address โดยจะลงทะเบียนต่อ SIP Server ที่ต้องการ

#### 2.6.5.2 Response Message

เป็นสัญญาณที่ใช้เพื่อตอบสนองต่อ Request Message ที่ได้รับมา ซึ่งลักษณะของสัญญาณ จะเป็น Status Code ดังต่อไปนี้

1xx เป็นรหัสการตอบสนองของ Server ที่จะเป็นข้อมูลเพื่อบอกรายละเอียดให้ Client ทราบถึงสถานะของข้อความที่ส่งไปว่าตอนนี้ถึงขั้นตอนไหนแล้ว โดยจะส่งรหัสแบบนี้มาเฉพาะในช่วงที่ Server กำลังทำงานกับข้อความนั้นเท่านั้น

- 100 บอกถึงว่าตอนนี้กำลังพยายามติดต่ออยู่
- 180 บอกถึงว่าตอนนี้กำลังรอการตอบรับจากผู้ที่ Server ติดต่อไป
- 182 บอกว่าตอนนี้ข้อความยังรออยู่ในคิวเพื่อรอการทำงานต่อไป

2xx รหัสการตอบสนองระหว่าง Client กันเองว่าคำสั่งที่ให้มีการทำงานตามข้อความที่ได้ส่งไปนั้นเรียบร้อยแล้ว

- 200 บอกถึงการยกเลิกการติดต่อด้วยคำสั่ง BYE เรียบร้อยแล้ว หรือ การยกเลิกข้อความบางข้อความด้วยคำสั่ง CANCEL เรียบร้อยแล้ว หรือ การเชิญให้มีการติดต่อด้วยคำสั่ง INVITE เรียบร้อยแล้ว หรือ การลงทะเบียนต่อ Server ด้วยคำสั่ง REGISTER เรียบร้อยแล้ว

3xx รหัสการตอบสนองสำหรับการติดต่อผ่าน Server ในโหมดการทำงานแบบ Redirection

- 302 บอก Address ของผู้ที่ต้องการติดต่อว่าผู้นั้นได้เปลี่ยนไปใช้ Address ใด

4xx รหัสการตอบสนองหากมีข้อผิดพลาดเกิดขึ้นในระหว่างที่มีการติดต่อโดยเป็นข้อผิดพลาดที่เกิดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าจากฝั่ง Client ที่ทำการติดต่อ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 400 บอกว่าการ Request นั้นไม่เหมาะสม ถูกต้องตามรูปแบบที่ควรจะเป็น
- 401 บอกว่าการ Request นั้นจะต้องมีการระบุรหัสผ่านด้วย
- 404 บอกว่า Address ที่ต้องการติดต่อมันไม่สามารถหาได้ คือไม่ได้ลงทะเบียนกับ Server ไว้

5xx รหัสการตอบสนองหากมีข้อผิดพลาดเกิดขึ้นในระหว่างที่มีการติดต่อกัน โดยเป็นข้อผิดพลาดที่เกิดที่ฝั่ง Server

- 500 เป็นข้อผิดพลาดที่เป็นข้อผิดพลาดภายในของ Server เอง
- 501 เป็นข้อผิดพลาดเนื่องจากในส่วนที่ต้องการทำงานนั้นยังไม่ได้มีการสร้าง
- 505 เป็นข้อผิดพลาดเนื่องจากเวอร์ชันที่ใช้ไม่สนับสนุนการทำงาน

6xx รหัสการตอบสนองหากมีข้อผิดพลาดเกิดขึ้นระหว่างที่มีการติดต่อกัน โดยเป็นข้อผิดพลาดทั่วไป

- 600 เป็นข้อผิดพลาดที่ทุกส่วนมีการทำงานที่อยู่แยกอยู่ไม่พร้อมจะทำงาน

#### 2.6.6 SIP Address

ในระบบ SIP การส่งข้อมูลระหว่างกันจะต้องระบุแอดเดรส โดยที่ SIP จะใช้แอดเดรสในลักษณะที่คล้ายกับอีเมลซึ่งวิธีการใช้แอดเดรสแบบนี้ ก็เป็นส่วนหนึ่งของตระกูลอินเทอร์เน็ตแอดเดรสที่เรียกกันว่า URIs เพื่อใช้อ้างอิงถึงผู้ใช้ SIP URIs จะประกอบด้วยแอดเดรส รูปแบบของแอดเดรสจะอยู่ในรูปแบบของ name@domain โดยอาจจะเป็น user@domain, user@address, phone-number@gateway หรือ user@host แอดเดรสนี้จะถูกใช้อ้างอิงถึงผู้ใช้ทั้งผู้เรียกและผู้ถูกเรียกในการส่งข้อความ ตัวอย่างของ SIP URI เช่น SIP:5001@192.168.1.5

#### 2.6.7 โพรโทคอล SDP (Session Description Protocol)

SDP เป็นโพรโทคอลที่บรรยายถึงลักษณะของการเชื่อมต่อ อธิบายถึงลักษณะของเซสชันที่ต้องการเปิด ซึ่งจำเป็นสำหรับ SIP เพราะ SIP ไม่ได้กำหนดชนิดของสื่อที่จะส่ง ดังนั้นผู้ส่งต้องบอกลักษณะของสื่อว่าเป็นชนิดไหน เช่น เสียง วิดีโอ หรือข้อมูล โดยใช้การส่งข้อมูลในรูปแบบของตัวอักษร (Text-Based) เช่นเดียวกับ SIP

การใช้ SDP ใน SIP นั้น จะใช้เมื่อต้องการสร้างการเชื่อมต่อโดยส่งไปพร้อมกับ Invite Message หรือ ACK Message ซึ่ง SDP จะระบุชนิดของสื่อที่ยอมรับได้และจะต้องยอมรับได้ด้วยกันทั้งสองฝ่ายถึงจะสร้างการเชื่อมต่อกันได้ นอกจากนี้ยังระบุหมายเลขพอร์ตและไอพีแอดเดรสที่จะใช้ในการสร้างการเชื่อมต่อกันอีกด้วย โดยแต่ละฟิลด์มีรายละเอียดดังนี้

v: จะแสดงถึง Protocol Version ของ SDP ซึ่งในปัจจุบันอยู่ที่ version 0

o: จะประกอบด้วยข้อมูลเกี่ยวกับผู้สร้างเซสชัน ซึ่งมีรูปแบบดังนี้

o = [username] [session-id] [version] [network-type] [address-type] [address]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่างเช่น o=johnston 2890844526 2890844526 IN IP4 43.32.1.5

username ชื่อผู้ใช้หรือโฮสต์ หรือ – ถ้าไม่มี  
 session-id หมายเลขอ้างอิงของแต่ละเซสชัน โดยจะใช้การสุ่มขึ้นมา โดยจะต้องไม่ซ้ำกัน  
 version ตัวเลขที่เพิ่มขึ้นเมื่อมีการเปลี่ยนแปลงเซสชัน  
 network-type เป็น IN เสมอสำหรับอินเทอร์เน็ต  
 address-type ใช้ IP4 สำหรับ IPv4 ส่วน IPv6 จะใช้ IP6  
 address ไอพีแอดเดรสหรือชื่อโฮสต์ก็ได้

s: จะประกอบด้วยชื่อสำหรับเซสชัน โดยสามารถมี options i แสดงรายละเอียดเกี่ยวกับเซสชันนั้นๆ

c: จะประกอบด้วยข้อมูลเกี่ยวกับการติดต่อสื่อสาร ซึ่งมีรูปแบบดังนี้

c = [network-type] [address-type] [connection-address]

ยกตัวอย่างเช่น c=IN IP4 170.30.130.133

network-type เป็น IN เสมอสำหรับอินเทอร์เน็ต  
 address-type ใช้ IP4 สำหรับ IPv4, ใช้ IP6 สำหรับ IPv6  
 connection-address ไอพีแอดเดรสที่ต้องการจะติดต่อ

t: จะประกอบด้วยเวลาเริ่มต้น และเวลาหยุดของเซสชัน

t = [start-time] [stop-time]

เวลาจะถูกระบุโดยใช้ NTP (Network Time Protocol) timestamp ถ้าเวลาหยุดเป็น 0 หมายความว่าเซสชันที่ทำไปเรื่อยๆ โดยไม่มีกำหนด ถ้าทั้งเวลาเริ่มต้น และเวลาหยุดเป็น 0 หมายความว่าเซสชันถาวร

m: ประกอบด้วยข้อมูลเกี่ยวกับชนิดของสื่อ

m = [media] [port] [transport] [format-list]

ยกตัวอย่างเช่น m=application 49152 TCP Warship

media รูปแบบของสื่อที่ใช้ เช่น audio, video, application, data, telephone-event, control เป็นต้น  
 port หมายเลขพอร์ตที่ใช้งาน  
 transport โพรโตคอลที่ใช้สื่อสาร เช่น RTP, UDP  
 format-list ข้อมูลอื่นๆ เกี่ยวกับสื่อที่ใช้

a: บอกลักษณะ Attributes ของสื่อที่ใช้ เช่น

a=rtpmap: 0 PCMU/8000

a=rtpmap: 6 DVI4/16000

a=rtpmap: 8 PCMA/8000

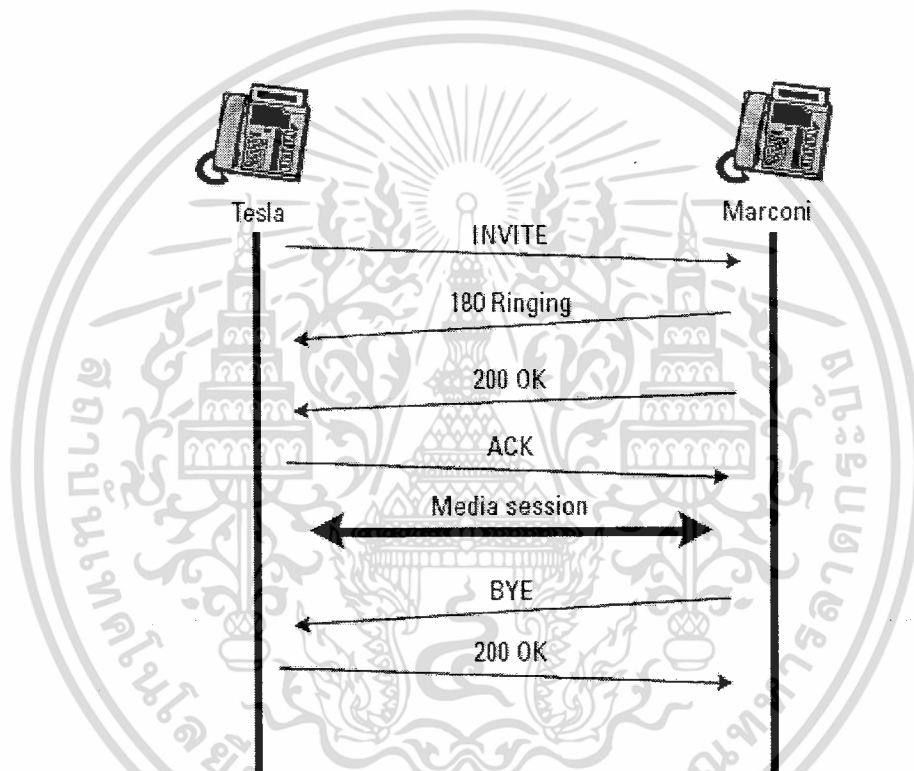
a=rtpmap: 99 iLBC

โดยที่ rtpmap: คือ RTP/AVP list มี media codec เป็น PCM u-law และ sampling rate เท่ากับ 8000 Hz

## 2.6.8 การติดต่อโดยใช้สัญญาณ SIP

### 2.6.8.1 การติดต่อแบบโดยตรง

การติดต่อแบบโดยตรงเป็นการติดต่อระหว่าง Client 2 ตัว โดยมีได้ผ่าน SIP component อื่นอีก ลักษณะลำดับสัญญาณในการติดต่อเป็นดังนี้



รูปที่ 2.28 ตัวอย่างลำดับสัญญาณที่ใช้ติดต่อกันโดยตรง

จากรูปที่ 2.28 แสดงลำดับสัญญาณ SIP ที่ใช้ในระบบ IP Telephony อธิบายได้ดังนี้

- **INVITE** เป็น Message Request ที่ผู้ร้องขอการติดต่อ (Tesla) ส่งไปยังผู้ถูกร้องขอการติดต่อ (Marconi) เพื่อเชิญการติดต่อ โดยภายใน Message INVITE นี้จะมีการใส่โปรโตคอล SDP ไปด้วยเพื่อบอกข้อมูลรูปแบบการรับ-ส่ง Media ที่ฝั่งผู้ร้องขอการติดต่อยอมรับได้ สามารถรับได้

- **180 Ringing** เป็น Message Response ที่ฝั่ง Marconi จะต้องส่งกลับมายัง Tesla เพื่อบอกว่ากำลังรอการรับสายจากอยู่

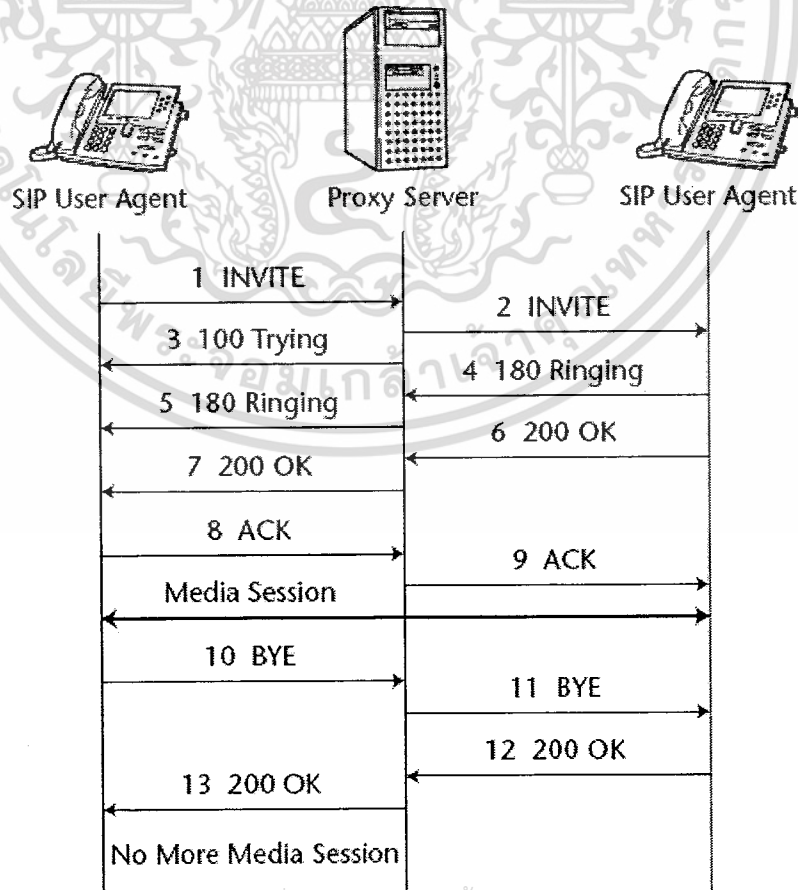
- **200 OK** เป็น Message Response ที่ฝั่ง Marconi จะส่งไปบอก Tesla เพื่อตอบตกลงการติดต่อ โดยภายใน Message 200 OK นี้จะมีการใส่โปรโตคอล SDP ไปด้วยเพื่อบอกข้อมูลรูปแบบการรับ-ส่งข้อมูลสื่อประสมที่ฝั่ง Marconi สามารถรับได้

- ACK เป็น Message ที่ Tesla จะส่งกลับไปเพื่อยืนยันว่าได้รับ Message 200 OK แล้ว
- Media session จะเป็นขั้นตอนการส่งสัญญาณเสียงผ่านทางช่องทาง media ที่ได้ตกลงผ่านทางโปรโตคอล SDP โดยจะมีขั้นตอนหลักๆ คือ ขั้นแรก SESSAION ESTABLISHED จะเป็นขั้นตอนการสร้างช่องสัญญาณการติดต่อ โดยใช้รูปแบบการรับ-ส่งข้อมูลสื่อประสมที่ได้ตกลงกันจาก SIP Message ส่วนที่สองคือ TRANSFER MEDIA STREAM เป็นขั้นตอนของการรับและส่ง ข้อมูล ที่ได้ตกลงกัน
- BYE เมื่อผู้ใช้ต้องการยกเลิกการติดต่อกัน ก็จะส่งสัญญาณนี้ไปบอกฝั่งตรงข้าม
- 200 OK เมื่อฝั่งตรงข้ามได้รับสัญญาณ BYE ก็จะตอบตกลงด้วยการส่ง Message 200 OK กลับไป แล้วทั้งสองฝั่งก็จะทำการปิดช่องสัญญาณการส่งข้อมูล

**2.6.8.2 การติดต่อแบบผ่าน Server (Bypass Server)**

การติดต่อแบบผ่าน Server เป็นความสามารถที่จะช่วยให้ SIP Client สามารถค้นหาผู้ใช้ปลายทางได้โดยการให้ SIP Server เป็นตัวช่วยค้นหาเส้นทางของผู้ใช้ปลายทางที่ต้องการได้ โดย Client จะต้องทำการลงทะเบียน (Register) ไว้ที่ Server ก่อนการติดต่อกัน ซึ่งการติดต่อแบบผ่าน Server ก็ยังแบ่งย่อยได้อีก 2 แบบ ตามมาตรฐาน คือ

**2.6.8.2.1 การติดต่อผ่าน Server แบบ Proxy**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่อาคารสื่อสารวิทยุ โทรคมนาคมในวังไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 2.29 การติดต่อผ่าน Server แบบ Proxy  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อผ่าน Server แบบ Proxy คือ การติดต่อโดยลำดับสัญญาณสัญญาณทั้งหมดจะต้องถูกส่งผ่าน SIP Server ก่อนไปถึงเครื่องปลายทาง โดยการทำงานในส่วนนี้จะเป็นหน้าที่ของ Proxy Server เริ่มตั้งแต่การร้องขอการติดต่อไปยัง Server เพื่อให้ Server ทำการค้นหา ที่อยู่ปลายทางให้ ไปจนกระทั่งได้รับการตอบตกลงจากปลายทางลักษณะของลำดับสัญญาณเป็นดังนี้

1. **Invite** เป็นสัญญาณที่ผู้ต้องการร้องขอส่งไปยัง Server เพื่อร้องขอการติดต่อ เมื่อ Server ได้รับแล้วก็จะไปค้นหา IP Address ของผู้ที่ถูกร้องขอใน Location Server (อยู่ใน Server ตัวเดียวกันแต่ทำงานคนละหน้าที่) ซึ่งจะเก็บหมายเลข IP Address ของ SIP User Agent ที่ลงทะเบียนไว้

2. **Invite** เมื่อค้นหา IP Address ได้แล้วก็จะทำการส่งต่อสัญญาณร้องขอออกไปยังปลายทางนั้น

3. **100 Trying** เป็นสัญญาณที่ Server ส่งไปบอกฝั่งผู้ร้องขอการติดต่อ ว่าตอนนี้กำลังพยายามติดต่อไป

4. **180 Ringing** เป็นสัญญาณที่ผู้ถูกร้องขอ ส่งกลับไปบอก Server ว่า ขณะนี้ได้รับสัญญาณ INVITE แล้ว กำลังรอการรับสายจากผู้ใช้

5. **180 Ringing** เป็นสัญญาณที่ Server ส่งกลับ ไปบอกผู้ร้องขอการติดต่อ ว่า ขณะนี้ ผู้ถูกร้องขอการติดต่อ ได้รับสัญญาณ INVITE แล้ว กำลังรอการรับสายจากผู้ใช้

6. **200 OK** เป็นสัญญาณที่ ผู้ถูกร้องขอการติดต่อ ส่งไปบอก Server ว่าตอนนี้ผู้ใช้ตกลงรับการติดต่อ

7. **200 OK** เป็นสัญญาณที่ Server ส่งไปบอกผู้ร้องขอการติดต่อ ว่าตอนนี้ รับการติดต่อแล้ว

8. **ACK** เป็นสัญญาณที่ผู้ร้องขอการติดต่อ ส่งกลับไปบอก Server ว่าได้รับสัญญาณ 200 OK แล้ว

9. **ACK** เป็นสัญญาณที่ Server ส่งกลับ ไปบอกผู้ถูกร้องขอการติดต่อ ว่า ผู้ร้องขอ ได้รับสัญญาณ 200 OK แล้ว

**Media Session** จะเป็นขั้นตอนการส่งสัญญาณเสียงผ่านทางช่องทาง media ที่ได้ตกลงผ่านทางโปรโตคอล SDP โดยจะมีขั้นตอนคือ

- **SESSION ESTABLISHED** เป็นขั้นตอนการสร้างช่องสัญญาณเพื่อเตรียมพร้อมที่จะส่งข้อมูลมัลติมีเดียระหว่างกัน

- **TRANSFER MEDIA STREAM** เป็นขั้นตอนการส่งข้อมูลมัลติมีเดียระหว่างผู้เรียก และ ผู้ถูกเรียก

10. **BYE** เป็นสัญญาณที่ส่งไปบอก Server ว่าต้องการยกเลิกการติดต่อ

11. **BYE** เป็นสัญญาณที่ Server ส่งไปบอกปลายทางว่าต้นทางต้องการยกเลิกการติดต่อ

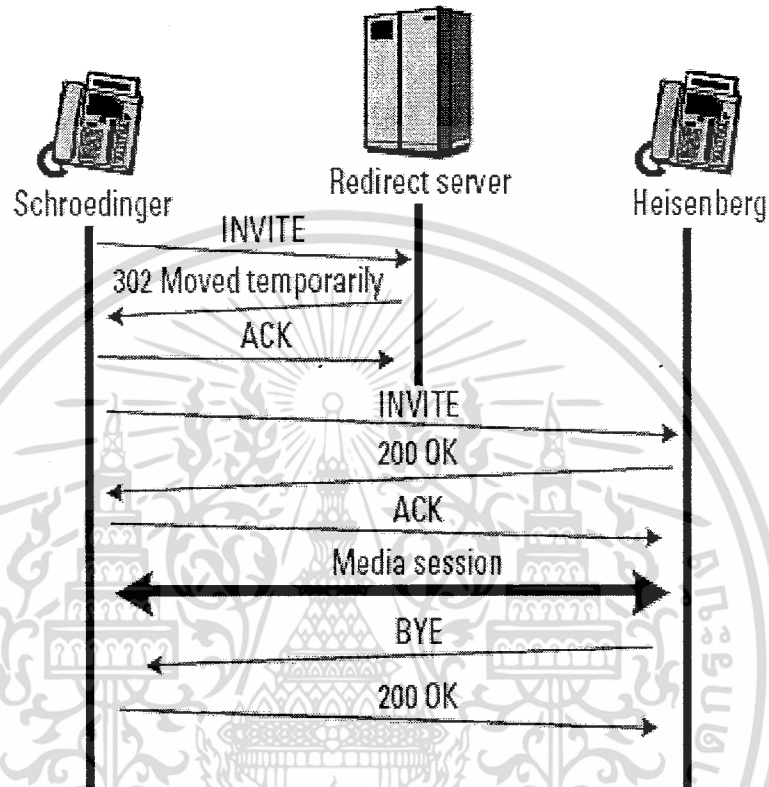
12. **200 OK** เป็นสัญญาณที่ปลายทางตอบตกลงยกเลิกการติดต่อ และปิดช่องทางการส่งข้อมูล

13. **200 OK** เป็นสัญญาณที่ Server ส่งไปบอกต้นทางว่าปลายทางตกลงยกเลิกการติดต่อและปิดช่องทางการส่งข้อมูล

#### 2.6.8.2.2 การติดต่อผ่าน Server แบบ Redirect

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อวัตถุประสงค์เฉพาะเท่านั้น ไม่สามารถนำออกให้ผู้อื่นได้โดยไม่ได้รับอนุญาต  
การติดต่อผ่าน Server แบบ Redirect คือ การติดต่อไปยังปลายทาง โดยทำการร้องขอเพื่อทำการ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหา IP Address ของ เครื่องปลายทางไปยัง SIP Server หลังจากที่ ได้ IP Address แล้ว ผู้ร้องขอการติดต่อ จะทำการติดต่อไปยังเครื่องปลายทางเอง โดยไม่ต้องติดต่อผ่าน Server อีก ลักษณะของลำดับสัญญาณเป็น ดังรูปที่ 2.30



รูปที่ 2.30 การติดต่อผ่าน Server แบบ Redirect

จากรูปที่ 2.30 แสดงลำดับสัญญาณในการติดต่อผ่าน Server แบบ Redirect อธิบายลำดับสัญญาณได้ ดังนี้

- INVITE เป็นสัญญาณที่ผู้ร้องขอการติดต่อ ส่ง ไปยัง Server เพื่อร้องขอการติดต่อ เมื่อ Server ได้รับ แล้วก็ไปค้นหา IP Address ของผู้ที่ถูกร้องขอใน Location Server (อยู่ใน Server ตัวเดียวกันแต่ทำงานคนละหน้าที) ซึ่งจะเก็บหมายเลข IP Address ของ SIP User Agent ที่ลงทะเบียนไว้
- 302 Moved Temporarily เมื่อค้นหา IP Address ได้แล้วก็จะทำการส่งต่อสัญญาณนี้ไปเพื่อบอกให้ผู้เรียกใช้ IP Address นี้ติดต่อไปยังปลายทาง
- ACK เป็นสัญญาณที่ส่งกลับ ไปบอก Server ว่า ได้รับ IP Address ของปลายทาง เรียบร้อยแล้ว
- INVITE เป็นสัญญาณที่ผู้เรียกส่ง ไปยังปลายทางตาม IP Address ที่ได้จาก Server
- 200 OK เป็นสัญญาณที่ปลายทาง ส่ง ไปบอกผู้เรียก ว่าตอนนี้ผู้ใช้ตกลงรับการติดต่อสื่อสาร
- ACK เป็นสัญญาณที่ผู้เรียกส่งกลับ ไปบอกปลายทาง ว่า ได้รับสัญญาณ 200 OK เรียบร้อยแล้ว
- Media Session จะเป็นขั้นตอนการส่งสัญญาณเสียงผ่านทางช่องทาง media ที่ได้ตกลงผ่านทาง

โปรโตคอล SDP โดยจะมีขั้นตอนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

- SESSION ESTABLISHED เป็นขั้นตอนการสร้างช่องสัญญาณเพื่อเตรียมพร้อมที่จะส่งข้อมูล

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

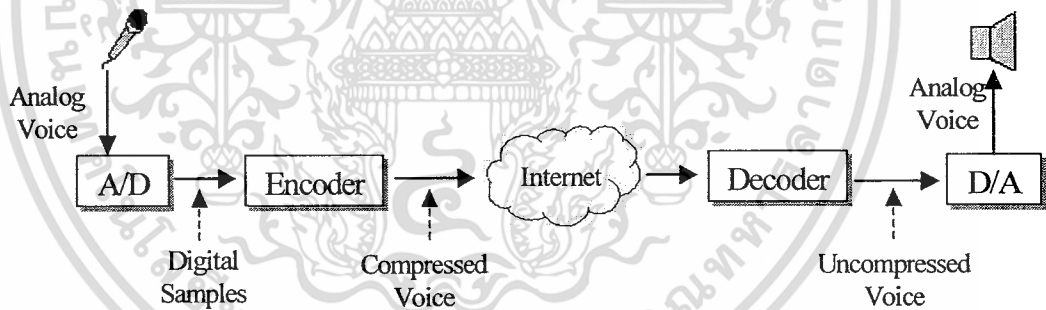
มัลติมีเดียระหว่างกัน

- TRANSFER MEDIA STREAM เป็นขั้นตอนการส่งข้อมูลมัลติมีเดียระหว่างผู้เรียกและผู้ถูกเรียก
- BYE เป็นสัญญาณที่ส่งไปบอก ว่าต้องการยกเลิกการติดต่อ
- 200 OK เป็นสัญญาณที่ตอบตกลงยกเลิกการติดต่อ พร้อมกับปิดช่องทางการส่งข้อมูล

## 2.7 ขั้นตอนการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต

หลักการพื้นฐานของการสื่อสารเสียงที่จะกล่าวถึงในหัวข้อนี้หมายถึงการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต ซึ่งจำเป็นจะต้องมีการแปลงสัญญาณเสียงจากอนาล็อก (Analog) ให้เป็นดิจิทัล (Digital) รวมทั้งต้องมีการลดขนาดของข้อมูลเสียงก่อนจะบรรจุลงในแพ็กเก็ตและส่งให้ผู้รับ และนอกจากนี้ผู้รับจะต้องเข้าใจข้อมูลที่ได้รับจากผู้ส่ง มิฉะนั้นผู้รับก็จะไม่สามารถนำข้อมูลไปประมวลผลได้ถูกต้อง ดังนั้นทั้งผู้รับและผู้ส่งจึงต้องใช้มาตรฐานในการสื่อสารเดียวกัน

ในการส่งเสียงพูดบนเครือข่ายอินเทอร์เน็ต ผู้ส่งจะต้องเปลี่ยนสัญญาณเสียงจากไมโครโฟนที่อยู่ในรูปแบบของสัญญาณอนาล็อกให้เป็นดิจิทัลเสียก่อน โดยใช้ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล (Analog to Digital Converter, A/D) ดังแสดงในรูปที่ 2.31 โดยปกติแล้วเสียงพูดของมนุษย์จะมีความถี่ไม่เกิน 4 กิโลเฮิร์ตซ์ (kHz) ดังนั้นอัตราการซักร้อยอย่าง (Sampling Rate) ตามทฤษฎีของในควิซต์ก็จะมีค่าเท่ากับ 8 กิโลเฮิร์ตซ์ (2 เท่าของความถี่สูงสุด) ซึ่งในระบบโทรศัพท์สาธารณะ (Public Switched Telephone Network, PSTN) ก็ใช้อัตราการซักร้อยอย่างเท่ากับค่านี้เช่นกัน



รูปที่ 2.31 การส่งเสียงผ่านเครือข่ายอินเทอร์เน็ต

### 2.7.1 การบีบอัดเสียง (Audio Compression)

ในการส่งเสียงพูดผ่านเครือข่ายอินเทอร์เน็ตนั้น เพียงแต่การแปลงจากสัญญาณเสียงแบบอนาล็อกให้กลายเป็นข้อมูลเสียงแบบดิจิทัลนั้นยังไม่เพียงพอ สังเกตว่าในการแปลงสัญญาณเสียงจากอนาล็อกเป็นดิจิทัลซึ่งใช้อัตราการซักร้อยอย่าง 8000 เฮิร์ตซ์ (ใน 1 วินาทีมีการซักร้อยอย่าง 8000 ครั้ง) ถ้าหากใช้ข้อมูลดิจิทัล 8 บิตต่อค่าตัวอย่าง แบบด्वิตซ์ที่ใช้ในการส่งข้อมูลก็จะเป็น 64 กิโลบิตต่อวินาที (Kbps) และถ้าใช้ 16 บิตต่อค่าตัวอย่าง แบบด्वิตซ์ก็เพิ่มเป็น 128 กิโลบิตต่อวินาที ซึ่งถือว่าเป็นค่าที่สูงทีเดียว ยิ่งไปกว่านั้นถ้าหากมีผู้ที่ต้องการสื่อสารเสียงพร้อมกันหลายคนบนเครือข่ายที่มีแบนด์วิธอย่างจำกัดก็จะเป็นไปไม่ได้ ผลที่ตามมาคือผู้รับก็จะได้ยินเสียงไม่ชัดเจน เสียงไม่ต่อเนื่อง หรืออาจจะฟังไม่รู้เรื่องเลยก็ได้ ซึ่งวิธีแก้ไขก็คือต้องมีการลดขนาดของข้อมูลเสียงก่อนที่จะส่งข้อมูลผ่านเครือข่าย เทคนิคนี้เรียกว่า การบีบอัดเสียง

ในการบีบอัดเสียงนั้น วิธีการลดขนาดของข้อมูลเสียงจะต้องสอดคล้องกับวิธีการเข้ารหัส มิฉะนั้นจะไม่สามารถนำเสียงไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาเล่นออกทางลำโพงได้ถูกต้อง สหภาพโทรคมนาคมนานาชาติ (International Telecommunication Union, ITU) ได้มีการกำหนดมาตรฐานสำหรับการบีบอัดเสียง (Audio Codec) เอาไว้หลายมาตรฐาน ตารางที่ 2.2 เป็นการเปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดเสียงแต่ละประเภท โดยในคอลัมน์ของ Quality MOS (Mean Opinion Score) เป็นค่าที่แสดงคุณภาพของเสียง ค่าสูงสุดมีค่าเท่ากับ 5 จะเห็นว่า G.711 นั้นมีคุณภาพสูงที่สุดก็จริง แต่ก็มีอัตราบิต (Bit Rate) สูงถึง 64 kbps ทำให้ไม่สามารถนำมาใช้งานได้จริงถ้าเครือข่ายมีแบนด์วิดท์ต่ำกว่านี้ เช่น ในการเชื่อมต่ออินเทอร์เน็ตผ่านทางเครือข่ายโทรศัพท์ซึ่งมีความเร็วอยู่ที่ 28.8 ถึง 56 กิโลบิตต่อวินาที เท่านั้น ส่วน G.729 และ G.723.1 ก็มีคุณภาพใกล้เคียงกับ G.711 แต่มีอัตราบิตน้อยกว่า มาตรฐานทั้งสองจึงน่าจะ เป็นทางเลือกที่ดีมาตรฐานอื่น

ตารางที่ 2.2 เปรียบเทียบคุณสมบัติของมาตรฐานการบีบอัดข้อมูลเสียง

Codec	Bit Rate (kbps)	Complexity compared with G.726	Quality MOS
G.711	64	very low	4.0
G.723.1	5.3*	8	3.9
G.723.1	6.3*	8	3.9
G.726	32	1	3.85
G.728	16	15	3.61
G.729	8	10	3.9
G.729A	8	6	3.7

\*G.723.1 มี 2 อัลกอริทึม ซึ่งมีอัตราบิตต่างกันคือ 5.3 และ 6.3 กิโลบิตต่อวินาที

### 2.7.2 ปัจจัยที่มีผลกระทบต่อคุณภาพของเสียง

เนื่องจากเครือข่ายอินเทอร์เน็ตไม่ได้มีการรับประกันคุณภาพการบริการ ดังนั้นการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ตจึงมีโอกาสที่คุณภาพของเสียงที่ผู้รับได้รับจะต่ำกว่าคุณภาพของเสียงต้นฉบับได้ ซึ่งปัจจัยที่มีผลกระทบต่อคุณภาพของเสียงมีดังนี้

#### 2.7.2.1 เวลาหน่วง (Delay)

เวลาหน่วง คือ เวลาในการเดินทางของเสียงจากไมโครโฟนของผู้ส่งไปถึงลำโพงของผู้รับในระบบโทรศัพท์ PSTN จะไม่เกิดปัญหานี้เนื่องจากการโทรแต่ละครั้ง จะมีการจองทรัพยากรสำหรับแต่ละคู่สายเอาไว้และค่าเวลาหน่วงจะน้อยมาก ประมาณ 50 – 70 มิลลิวินาที ในขณะที่เครือข่ายอินเทอร์เน็ตไม่สามารถที่จะรับประกันได้ว่าค่าเวลาหน่วงเป็นเท่าใด โดยปกติคนเราจะรู้สึกถึงผลกระทบของเวลาหน่วงเมื่อค่าเวลาหน่วงสูงกว่า 250 มิลลิวินาที

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.7.2.2 จิตเตอร์ (Jitter)

จิตเตอร์เกิดจากการที่เวลาหน่วงของแต่ละแพ็กเก็ตไม่เท่ากัน ทั้งนี้เนื่องจากการเดินทางของแพ็กเก็ตบนเครือข่ายอินเทอร์เน็ตนั้น แต่ละแพ็กเก็ตอาจใช้เส้นทางที่ต่างกัน ได้แม้จะมีปลายทางเดียวกัน รวมถึงสภาพความคับคั่งของเครือข่ายที่เปลี่ยนแปลงได้ตลอดเวลา ก็สามารถทำให้เกิดจิตเตอร์ได้เช่นกัน ปัญหาจึงทำให้สัญญาณเสียงที่ได้รับมีอาการกระตุกและฟังได้ไม่ชัดเจน ดังนั้นจึงต้องมีการใช้บัฟเฟอร์เพื่อรวบรวมข้อมูลก่อนเพื่อให้ข้อมูลมีความต่อเนื่อง แล้วจึงส่งไปประมวลผลและเล่นออกทางลำโพงต่อไป การใช้บัฟเฟอร์ขนาดใหญ่จะสามารถกำจัดจิตเตอร์ได้ดีขึ้นแต่ก็มีผลทำให้มีเวลาหน่วงมากขึ้น แต่ถ้าใช้บัฟเฟอร์ขนาดเล็กเกินไปจะทำให้กำจัดจิตเตอร์ได้ไม่ดีนัก ดังนั้นจึงจำเป็นต้องหาวิธีที่ใช้ในการปรับขนาดของบัฟเฟอร์ ให้เหมาะสม

### 2.7.2.3 การสูญหายของแพ็กเก็ต (Packet Loss)

การที่แพ็กเก็ตสูญหายนั้นส่วนใหญ่มักมีสาเหตุมาจากความคับคั่งของเครือข่ายเพราะในขณะที่เราเตอร์ทำการประมวลผลแพ็กเก็ตหนึ่งอยู่ ถ้ามีอีกแพ็กเก็ตเข้ามาจะถูกเก็บไว้ในบัฟเฟอร์ของคิว และถ้าบัฟเฟอร์นั้นเต็มจะทำให้แพ็กเก็ตที่เข้ามาที่หลังถูกตัดทิ้ง ถึงแม้ว่าในการสื่อสารเสียงจะยอมให้มีการสูญหายของแพ็กเก็ต ได้บ้าง แต่การที่มีแพ็กเก็ตสูญหายมากเกินไปก็อาจส่งผลกระทบต่อคุณภาพเสียงได้ โดยการสูญหายของแพ็กเก็ตจะมีผลกระทบอย่างชัดเจนเมื่อมีอัตราการสูญหายมากกว่า 5%

### 2.7.2.4 แบนด์วิดท์ที่มีจำกัด (Limited bandwidth)

แบนด์วิดท์เป็นส่วนที่สำคัญมากในการสื่อสารเสียง ซึ่งปริมาณการใช้แบนด์วิดท์นั้นก็ขึ้นอยู่กับอัลกอริทึมในการบีบอัดข้อมูลเสียง ถ้าแบนด์วิดท์ที่ใช้ในขณะนั้นไม่เพียงพอจะทำให้เกิดปัญหาต่างๆ ตามมาไม่ว่าจะเป็นเวลาหน่วง จิตเตอร์ และสูญหายของแพ็กเก็ต

## 2.7.3 โพรโตคอลที่ใช้ในการสื่อสารเสียงบนเครือข่ายอินเทอร์เน็ต

โพรโตคอลในระดับชั้นทรานสปอร์ต (Transport Layer) ของเครือข่ายอินเทอร์เน็ตที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบันมีอยู่สองโพรโตคอลคือ TCP ซึ่งเป็นโพรโตคอลที่มีการรับประกันว่าสามารถส่งข้อมูลไปถึงปลายทางได้อย่างถูกต้องและครบถ้วน อีกโพรโตคอลหนึ่งก็คือ User Datagram Protocol (UDP) ซึ่งเป็นโพรโตคอลที่ไม่มีการรับประกันความน่าเชื่อถือของข้อมูล จะเห็นว่า TCP เป็นโพรโตคอลที่มีประสิทธิภาพมากกว่าและกลไกการทำงานซับซ้อนมากกว่า UDP แต่ TCP นั้นไม่เหมาะสำหรับการสื่อสารเสียง ทั้งนี้เนื่องจากความล่าช้าของกลไกการทำงานของ TCP นั้นเอง ทั้งนี้การสื่อสารเสียงนั้นต้องการความเร็วโดยไม่จำเป็นต้องส่งแพ็กเก็ตเสียงให้ไปถึงผู้รับอย่างครบถ้วน เพราะถึงแม้ว่าจะมีข้อมูลบางส่วนสูญหายไปบ้าง ก็ไม่ได้ส่งผลกระทบต่อคุณภาพของเสียงมากนัก ดังนั้น UDP จึงเป็นทางเลือกที่ดีกว่า แต่การที่จะใช้ UDP อย่างเดียวนั้นยังไม่เพียงพอ จึงได้มีการพัฒนาโพรโตคอลตัวหนึ่งทำงานอยู่บนเหนือโพรโตคอล UDP โพรโตคอลตัวนี้มีชื่อว่า Real-time Transport Protocol (RTP) ซึ่งองค์กรที่เสนอโพรโตคอลตัวนี้ขึ้นมาคือ Internet Engineering Task Force (IETF) โพรโตคอล RTP ได้รับการพัฒนาขึ้นมาเพื่อใช้ในการสื่อสารแบบเวลาจริง เช่น การสื่อสารด้วย

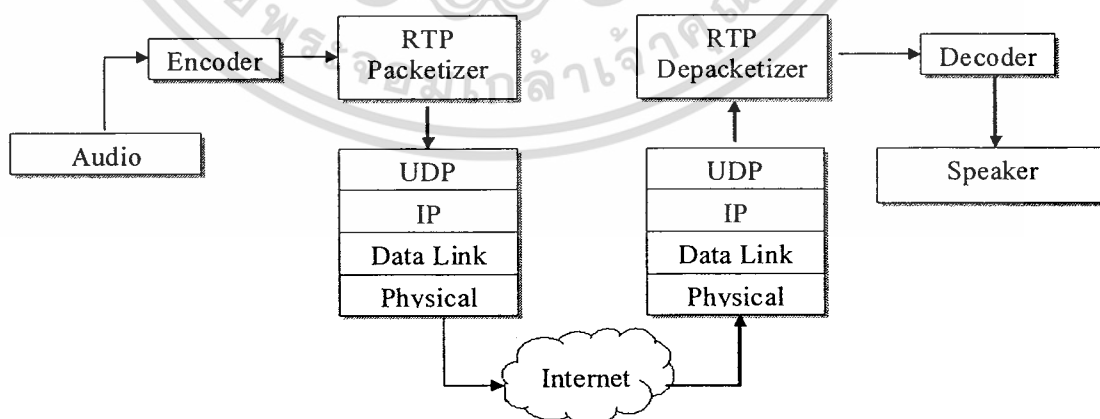
## เสียงหรือวีดิทัศน์

## 2.7.3.1 โพรโทคอล RTP (Real-time Transport Protocol)

RTP เป็นโพรโทคอลที่ได้รับการพัฒนาโดย IETF และมีข้อกำหนดอยู่ใน RFC 3550 RTP ถูกออกแบบมาเพื่อใช้เป็นโพรโทคอลในการส่งข้อมูลของการสื่อสารที่ทำงานในลักษณะเวลาจริง หน้าที่สำคัญของ RTP คือ ทำให้ทางผู้รับสามารถนำข้อมูลที่ได้จากแพ็กเก็ตที่ส่งไปแสดงผลได้ถูกต้อง กลไกหลักที่ใช้ใน RTP ก็คือ การประทับเวลา (Time Stamping) และการกำหนดหมายเลขลำดับ (Sequence Numbering) ผู้ส่งจะกำหนดค่า Timestamp ซึ่งเป็นฟิลด์หนึ่งในเฮดเดอร์ของ RTP ตามเวลาที่ข้อมูลในไบต์แรกของแพ็กเก็ตถูกซัดตัวอย่าง ทางด้านของผู้รับเมื่อได้รับแพ็กเก็ตจะใช้ค่า Timestamp เพื่อกำหนดจังหวะที่ถูกต้องในการแสดงผล และยังอาจจะใช้ในการประสานเวลาระหว่างสัญญาณเสียงและ

วีดิทัศน์ ส่วนกลไกการกำหนดหมายเลขลำดับนั้นใช้ฟิลด์หนึ่งในเฮดเดอร์ RTP ที่ชื่อว่า Sequence Number โดยค่าของหมายเลขนี้จะเพิ่มขึ้นตามจำนวนแพ็กเก็ตที่ส่ง ซึ่งผู้รับจะใช้ข้อมูลนี้ในการเรียงลำดับและตรวจสอบการสูญหายของแพ็กเก็ต นอกจากนี้ RTP ยังให้ข้อมูลเกี่ยวกับวิธีการบีบอัดข้อมูลเสียงหรือวีดิทัศน์ เพื่อให้ผู้รับสามารถแสดงผลข้อมูลได้ถูกต้อง และนอกจากข้อกำหนดเกี่ยวกับการใช้โพรโทคอลในการขนส่งข้อมูลของสื่อแล้ว ใน RFC 3550 ยังมีข้อกำหนดของโพรโทคอลหนึ่งคือ RTP Control Protocol (RTCP) ซึ่งเป็นโพรโทคอลที่ถูกนำมาใช้งานร่วมกับ RTP โดยโพรโทคอล RTCP จะถูกนำไปใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ

รูปที่ 2.32 แสดงให้เห็นถึงบทบาทของ RTP ในการสื่อสารเสียงและวีดิทัศน์ จะเห็นว่าเมื่อเสียงหรือวีดิทัศน์ ได้ผ่านขั้นตอนการเข้ารหัสด้วยวิธีการบีบอัดข้อมูลแล้ว ข้อมูลที่ได้ก็จะมีขนาดลดลง หลังจากนั้นข้อมูลส่วนนี้ก็จะถูกนำมาบรรจุลงในแพ็กเก็ต RTP ซึ่งแต่ละแพ็กเก็ตก็จะมีกระบวนการระบุด้วยว่าใช้วิธีการบีบอัดข้อมูลแบบใด เป็นแพ็กเก็ตในลำดับที่เท่าไร รวมไปถึงข้อมูลอื่นๆ ดังนั้นเมื่อข้อมูลนี้เดินทางผ่านเครือข่ายอินเทอร์เน็ต ไปถึงฝั่งของผู้รับ ผู้รับก็สามารถใช้ข้อมูลนี้ในการแสดงผลได้ถูกต้อง

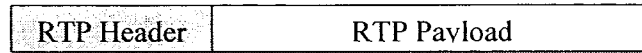


รูปที่ 2.32 บทบาทของ RTP ในการสื่อสารเสียงและวีดิทัศน์บนเครือข่ายอินเทอร์เน็ต

แพ็กเก็ต RTP แต่ละแพ็กเก็ตนั้นจะมีโครงสร้างดังที่แสดงในรูปที่ 2.33 โดยส่วนแรกของ

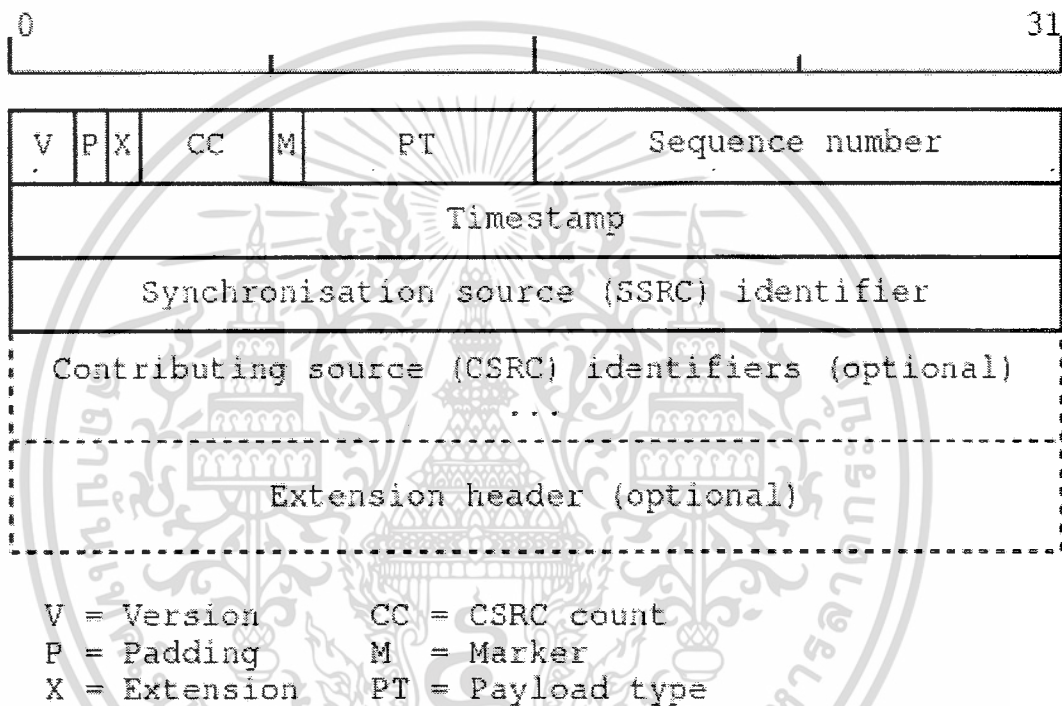
เอกสารนี้เป็นแพ็กเก็ตคือ ส่วนเฮดเดอร์ RTP (RTP Header) เป็นส่วนที่ระบุถึงคุณสมบัติของแต่ละแพ็กเก็ต โดยที่ข้อมูลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียงหรือวิดิทัศน์ที่ผ่านการเข้ารหัสแล้วจะถูกนำไปวางในตำแหน่งของเพย์โหลด (Payload)



รูปที่ 2.33 โครงสร้างแพ็กเก็ตของ RTP

โดยทั่วไปเฮดเดอร์คือ โครงสร้างข้อมูลของแต่ละแพ็กเก็ตที่ถูกรวมเข้ากับข้อมูลที่จะส่ง ซึ่งในแต่ละโปรโตคอลจะมีโครงสร้างของเฮดเดอร์ที่แตกต่างกัน เพื่อกำหนดรูปแบบข้อมูลของโปรโตคอลตัวนั้นๆ สำหรับเฮดเดอร์ของ RTP นั้นมีโครงสร้างดังรูปที่ 2.34



รูปที่ 2.34 โครงสร้างเฮดเดอร์ของ RTP

จากโครงสร้างเฮดเดอร์ของ RTP ในรูปที่ 2.34 จะเห็นว่า RTP จะมีเฮดเดอร์อย่างน้อย 12 ไบต์ โดยมีรายละเอียดของฟิลด์ต่างๆ ดังนี้

- **Version (V)** กำหนดเวอร์ชันของโปรโตคอล RTP ที่ใช้ซึ่งปัจจุบันเป็นเวอร์ชัน 2
- **Padding (P)** เป็นฟิลด์ที่บอกว่าแพ็กเก็ตนั้นได้ถูกเติมด้วยข้อมูลที่เพิ่มให้พอดีกับขนาดของแพ็กเก็ตหรือไม่ และยังใช้ในกระบวนการการเข้ารหัสข้อมูล (Encryption) อีกด้วย
- **Extension (X)** เป็นฟิลด์ที่บอกว่าเฮดเดอร์ RTP มีการขยาย (Extension) หรือไม่
- **CSRC Count (CC)** บอกจำนวนของ Contribution Source Identifier ในแพ็กเก็ต โดยมี CSRC ได้ตั้งแต่ 0 - 15
- **Marker (M)** ใช้สำหรับบางแอปพลิเคชันที่ต้องการ Mark ตำแหน่งบน data stream
- **Payload Type (PT)** บอกชนิดของข้อมูลภายในเพย์โหลด ซึ่งก็คือชนิดของการบีบอัดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่สามารถนำออกเผยแพร่ภายนอกได้โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ อย่างไรก็ตามหากท่านมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายบริการลูกค้าของเราได้ที่ 1676-1676 หรือ 1676-1676-1676

แต่ละแบบ

- **Sequence Number** บอกลำดับของแพ็กเก็ต และเพื่อให้ผู้รับตรวจสอบการสูญหายของแพ็กเก็ต และสามารถใช้ในการเรียงลำดับแพ็กเก็ตใหม่ได้ โดยที่ sequence number แรกของเซสชันจะทำการสุ่มขึ้นมา

- **Timestamp** เป็นค่าที่บอกเวลาที่ใช้ในการซัดตัวอย่างของข้อมูลในแพ็กเก็ต ซึ่งนำไปใช้ในกระบวนการคำนวณการกำจัดจitter และค่า Round Trip Time Delay โดยที่ค่าเริ่มต้นของ time stamp จะทำการสุ่มขึ้นมา

- **SSRC Identifier** เป็นตัวเลขประจำ Session นั้นคือแพ็กเก็ตที่มาจากผู้ส่งเดียวกัน และใช้ค่า SSRC เท่ากัน แสดงว่าเป็น Session เดียวกัน โดยที่ค่าเริ่มต้นของ SSRC Identifier จะทำการสุ่มขึ้นมา

- **CSRC Identifier** ฟิวด์นี้จะถูกใช้เมื่อมีการผสมสื่อ (Media Mixing) ซึ่งเป็นกระบวนการที่ใช้เมื่อมีการประชุมแบบหลายจุด (Multipoint Conference)

### 2.7.3.2 โพรโทคอล RTCP (RTP Control Protocol)

RTCP เป็นโพรโทคอลที่ถูกกำหนดอยู่ใน RFC 3550 เช่นเดียวกับ RTP โพรโทคอลตัวนี้ใช้ในการรายงานเกี่ยวกับสถิติและคุณภาพการบริการ RTCP ใช้ช่องสื่อสารแยกกันกับ RTP โดย RTCP นั้นใช้พอร์ตถัดจาก RTP ไปหนึ่งพอร์ต ช่องสื่อสารของ RTCP นั้นใช้สำหรับการรายงานเกี่ยวกับคุณภาพการบริการ เช่น จำนวนแพ็กเก็ตที่ได้รับ จำนวนแพ็กเก็ตที่สูญหาย จิตเตอร์ เป็นต้น แพ็กเก็ต RTCP นั้นแบ่งออกเป็น 5 ชนิด ได้แก่

- **SR (Sender Report)** สำหรับบอกสถิติเกี่ยวกับการส่งข้อมูลของผู้ส่ง

- **RR (Receiver Report)** สำหรับบอกสถิติเกี่ยวกับการรับข้อมูลของผู้รับ

- **SDES (Source description items)** เป็นรายละเอียดต่างๆ ของผู้ส่ง

- **BYE** สำหรับแจ้งถึงการจบการทำงาน

- **APP (Application Specific Functions)** สำหรับการกำหนดการทำงานบางอย่างในส่วนของโปรแกรมประยุกต์ ซึ่งไม่ได้ระบุเอาไว้เป็นมาตรฐาน

แพ็กเก็ต RTCP ที่น่าสนใจก็คือ RR เพราะว่าผู้รับจะรวบรวมสถิติต่างๆ ของข้อมูลเสียงหรือวิดีโอที่ตนที่ได้รับจากแพ็กเก็ต RTP แล้วบรรจุลงในแพ็กเก็ต RR นี้และส่งให้กับผู้ส่ง ผู้ส่งสามารถใช้ข้อมูลในแพ็กเก็ตชนิดนี้ในการประเมินสถานะของเครือข่าย รวมถึงคุณภาพของสื่อที่ผู้รับได้รับ เพื่อใช้ในการออกแบบวิธีการควบคุมคุณภาพแบบปรับตัวของทั้งการสื่อสารเสียงและวิดีโอ เพื่อให้การสื่อสารเป็นไปอย่างราบรื่น

## 2.8 PJSIP API

ในการพัฒนาโปรแกรมสำหรับ IP Telephony ในปัจจุบันมีเครื่องมือ (tool) ช่วยในการเขียน

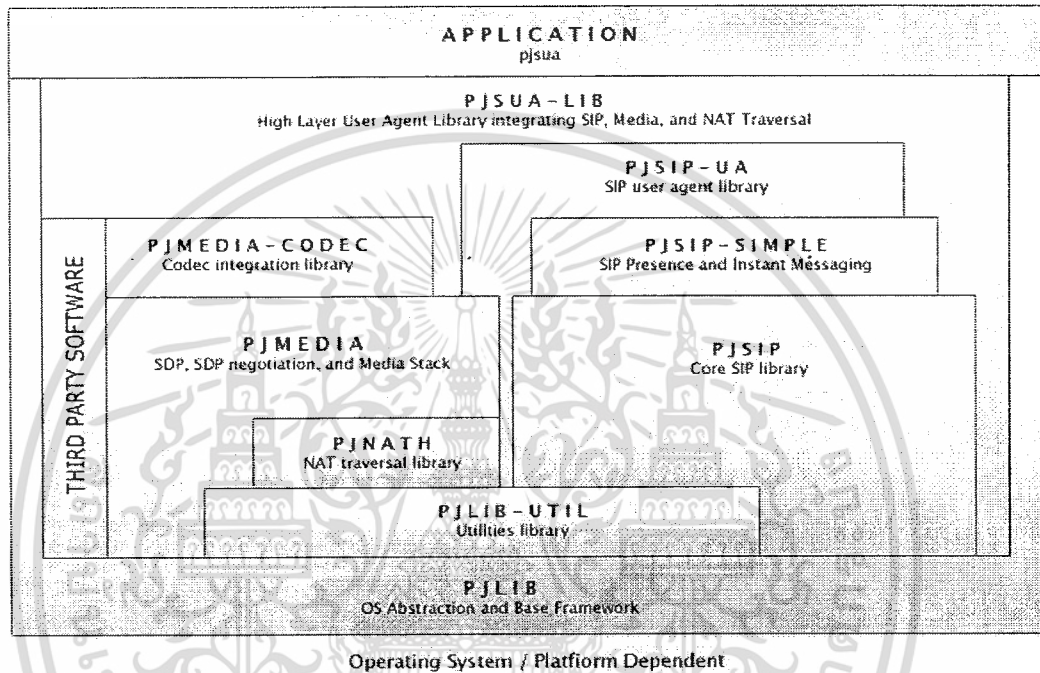
โปรแกรมมากมาย โดยในโครงการนี้จะใช้ PJSIP API ซึ่งมีจุดเด่นคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ส่วนตัวเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

- เป็น Open Source ลิขสิทธิ์แบบ GPL

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีขนาดเล็กเหมาะสำหรับระบบสมองกลฝังตัว ( embedded system)
- สามารถรันได้ใน CPU หลากๆตระกูลเช่น ARM , x86 , MIPS ,PowerPC , SPARC เป็นต้น
- สามารถใช้งานได้ในหลายๆ Platform เช่น Window, Linux, uC-Linux, Symbian, PDA, Window Mobile เป็นต้น
- สามารถเลือกพัฒนาได้หลายภาษา เช่น C , C++ , Python เป็นต้น



รูปที่ 2.35 สถาปัตยกรรมของ PJSIP

สถาปัตยกรรมของ PJSIP จะประกอบไปด้วยส่วนต่างๆ ดังนี้

- **PJLIB** เป็น library หลัก ซึ่งจะถูกเรียกใช้โดยทุกๆ library
- **PJLIB-UTIL** เป็น library เสริม ซึ่งประกอบไปด้วยส่วนของ Algorithm การเข้ารหัสต่างๆ การใช้งาน STUN , XML ซึ่งจะถูกเรียกใช้งานโดย PJMEDIA และ PJSIP
- **PJNATH** เป็น library ซึ่งช่วยเหลือในการใช้งานบน Network ประเภท NAT ได้แก่ STUN TURN , ICE
- **PJMEDIA** เป็น library ที่จัดการในส่วนของ media ไม่ว่าจะเป็น Codec ที่ใช้ในการเข้ารหัสเสียง โดย Codec ที่รองรับ ได้แก่ G.711 (PCMU, PCMA), G.722, GSM, iLBC, IPP, L16 (PCM 16 bits), Speex (narrow/wide/ultrawide-band) เป็นต้น ส่วนที่ติดต่อกับ sound driver รวมถึง Media Stack ต่างๆ ไม่ว่าจะเป็น RTP, RTCP, SDP
- **PJSIP** เป็น library หลักในส่วนของ SIP
- **PJMEDIA-CODEC** เป็นส่วนที่ใช้ในการเข้ารหัสเสียง Codec ต่างๆ
- **PJSIP-SIMPLE** เป็น library สำหรับ event framework หลัก ของ SIP

- **PJSIP-UA** เป็น library ในส่วนของ User Agent โดยจะทำหน้าที่ Invite Session, Registration, Call Transfer เป็นต้น
- **PJSUA-LIB** เป็น library Hi-level SIP UA API ซึ่งรวมความสามารถของ SIP Stack และ Media Stack เพื่อให้ง่ายต่อการใช้งาน API

## 2.9 โปรแกรม Asterisk

โปรแกรม Asterisk เป็นซอฟต์แวร์ประเภท Open source ซึ่งทำงานได้บนหลายๆ ระบบปฏิบัติการ เช่น Linux, OpenBSD, FreeBSD, Sun Solaris เป็นต้น คุณสมบัติของ Asterisk สามารถทำหน้าที่เป็น VoIP Server โดยได้มีการจัดเตรียมฟังก์ชันการใช้งานของตู้สาขาโทรศัพท์ PABX (Private Automatic Branch Exchange) คุณภาพสูงไว้ในตัว ทำให้ผู้ใช้สามารถสร้างระบบโทรศัพท์ภายในโดยผ่านเครือข่าย LAN ภายในองค์กรได้ โดยสามารถรองรับโปรโตคอลที่เกี่ยวข้องกับระบบ VoIP ได้หลายตัว เช่น SIP, H.323, IAX, MGCP, SCCP (Cisco Skinny)

Asterisk มีการเผยแพร่แบบ Open Source ภายใต้ GNU (General Public License) นั้นหมายความว่าเราสามารถดาวน์โหลดโปรแกรมมาใช้ได้ฟรีตามข้อกำหนด

Asterisk ถูกพัฒนาและสร้างโดย Mr. Mark Spencer แห่งบริษัท Digium Inc. เมื่อปีค.ศ.1999 และได้มีการเผยแพร่โปรแกรมไปทั่วโลกในกลุ่ม Open Source เพื่อทดสอบและแก้ไขปัญหาของโปรแกรม (Bug) อย่างต่อเนื่อง จากนั้นเมื่อประมาณปี ค.ศ.2001 Mr. Mark Spencer ได้ร่วมมือกับโครงการ Zapata Project และ Mr. Jim Dixon พัฒนาอุปกรณ์ที่สามารถรองรับการทำงานร่วมกับโปรแกรม Asterisk ซึ่งสามารถใช้กับอุปกรณ์ของบริษัท Digium, Sungoma และบริษัทอื่นๆ ที่ขายผลิตภัณฑ์รองรับกับ Asterisk โดยที่คุณสมบัติหลักของอุปกรณ์ส่วนใหญ่จะใช้ความสามารถของหน่วยประมวลผลกลาง (CPU) และ หน่วยความจำ (RAM) ของเครื่อง Server เป็นหลัก ไม่ว่าจะเป็นการทำงานในส่วนของ Media Streaming, Echo Cancellation และ Transcoding และในปัจจุบันทางบริษัทได้เล็งเห็นถึงความสำคัญในส่วนนี้ จึงได้มีการพัฒนาการ์ดประมวลผลต่างๆ เพื่อลดการทำงานของหน่วยประมวลผลกลาง และ หน่วยความจำของเครื่อง Server หลายชนิดเพื่อให้ Server สามารถใช้หน่วยความจำหรือ หน่วยประมวลผลกลางได้ดีขึ้น

### 2.9.1 สถาปัตยกรรมของ Asterisk

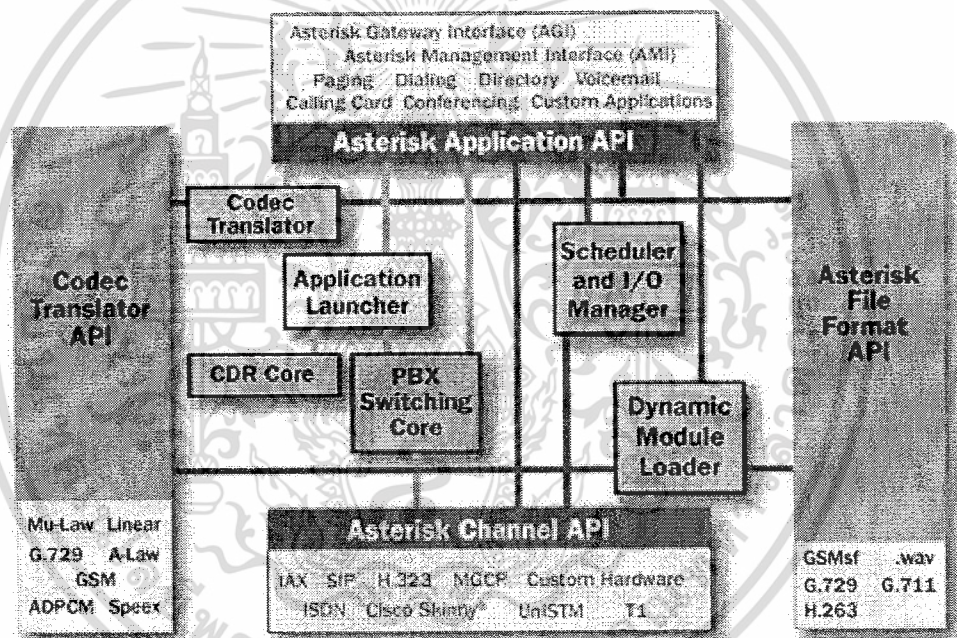
Asterisk ได้มีการออกแบบระบบให้มีความยืดหยุ่นสูงโดยมีการระบุส่วนประกอบของ API (Application Programming Interface) ซึ่งทำให้มีประโยชน์มาก เมื่อมีผู้พัฒนา API ก็ยังสามารถนำมาต่อยอดทำงานร่วมกับระบบโทรศัพท์ Asterisk ได้ทันที และจะมีระบบ Central PABX อยู่เป็นโครงสร้างภายใน ดังรูปที่ 2.36

### 2.9.2 ส่วนประกอบภายนอกของ Asterisk

ส่วนประกอบภายนอกของของสถาปัตยกรรม Asterisk จะเป็น API จำนวน 4 ส่วนด้วยกัน ซึ่งจะมีหน้าที่ในการไหลการทำงานของ API นั้นๆ ซึ่งทำให้ Asterisk ไม่จำเป็นต้องคอยจัดการเองทั้งหมด

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Channel API** จะทำหน้าที่จัดการกับประเภทการเชื่อมต่อที่เข้ามาไม่ว่าจะเป็นประเภทของ VoIP ประเภทต่างๆ เช่น ISDN, IAX, SIP, H.323, MGCP, Cisco Skinny และรวมถึงสัญญาณ Signaling ต่างๆ
- **Application API** จะทำหน้าที่อนุญาตให้งานหลายๆ งานสามารถทำหน้าที่ได้หลายๆ หน้าที่ เช่น Conferencing, Directory Listing, Voice Mail ซึ่งงานบางงานจำเป็นต้องดำเนินการทันที หรืออาจจะดำเนินการในอนาคตก็เป็นได้
- **Code Translator API** ทำหน้าที่โหลดตัวเข้ารหัสและถอดรหัส (Codec) ของไฟล์เสียงรูปแบบต่างๆ ไม่ว่าจะเป็น GSM A-Law  $\mu$ -Law G.729 รวมถึง MP3 เป็นต้น
- **File Format API** ทำหน้าที่ในการอ่านและบันทึกไฟล์หลากหลายรูปแบบ เช่น ไฟล์เสียง .gem .wav .mp3



รูปที่ 2.36 สถาปัตยกรรมของ Asterisk

### 2.9.3 ส่วนประกอบภายในของ Asterisk

- **PBX Switching** นับเป็นส่วนที่สำคัญมากสำหรับ Asterisk จะทำหน้าที่การเชื่อมต่อระหว่างผู้ใช้งานหลายๆคน และการทำงานอัตโนมัติ รวมถึงการจัดการและดูแลเกี่ยวกับซอฟต์แวร์และฮาร์ดแวร์ของระบบอีกด้วย
- **Application Launcher** เป็นตัวประกาศว่าบริการใดจะเริ่มทำงาน เช่น Voice Mail , File Playback เป็นต้น
- **Code Translator** จะมีการใช้ Codec โมดูลเพื่อการเข้ารหัสและถอดรหัสไฟล์เสียงที่ถูกบีบอัดเพื่อใช้ในการส่งสัญญาณเสียง โดยมีการเลือกมาตรฐานของ Codec ให้มีความเหมาะสม โดยจะคำนึงถึงคุณภาพเสียงและการใช้งานของสัญญาณในการส่งผ่านข้อมูลด้วย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต การคัดลอกหรือการนำข้อมูลไปใช้โดยไม่ได้รับอนุญาต อาจก่อให้เกิดความเสียหายทางกฎหมายได้

- **Scheduler and I/O Management** ทำหน้าที่ในการจัดสรรและดูแลตารางงานต่างๆ ซึ่งอยู่ในระบบล่าง เพื่อให้งานเหล่านั้นสามารถทำงานได้อย่างมีประสิทธิภาพภายใต้เงื่อนไขที่กำหนดทั้งหมด
- **Dynamic Module Loader** ทำหน้าที่ในการจัดสรร โมดูลต่างๆที่มีความจำเป็นต่อการทำงานของระบบ Asterisk
- **CDR หรือ (Call Detail Record)** ทำหน้าที่ในการบันทึกข้อมูลการใช้งานโทรศัพท์ของระบบทั้งหมด อาทิเช่น หมายเลข โทรศัพท์ต้นทาง ปลายทาง วันที่ และ เวลาในการสนทนา เป็นต้น

## 2.10 ภาษา PHP

ภาษาพีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะ Open source ภาษา PHP ใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยพื้นฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษา Perl ซึ่ง ภาษา PHP นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีความตอบโต้ได้อย่างรวดเร็ว

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากคาด้านเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับ โปรแกรมในลักษณะ CGI

คำสั่งของพีเอชพี สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น Notepad หรือ vi ซึ่งทำให้การทำงานของพีเอชพี สามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผล Web Server เช่น Apache, Microsoft Internet Information Services (IIS), OmniHTTPd และอื่นๆ อีกมากมาย สำหรับส่วนหลักของ PHP ยังมี Module ในการรองรับ CGI มาตรฐาน ซึ่ง PHP สามารถทำงานเป็นตัวประมวลผล CGI ด้วย และด้วย PHP คุณมีอิสรภาพในการเลือก ระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ นอกจากนี้คุณยังสามารถใช้สร้างโปรแกรมโครงสร้าง สร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่าความสามารถของคำสั่ง OOP มาตรฐานในปัจจุบันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ OOP เท่านั้น

พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ Oracle dBase PostgreSQL IBM DB2 MySQL Informix ODBC โครงสร้างของฐานข้อมูลแบบ DBX ซึ่ง

เอกสารนี้ให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้ และ PHP ยังรองรับ ODBC (Open Database Connection) ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

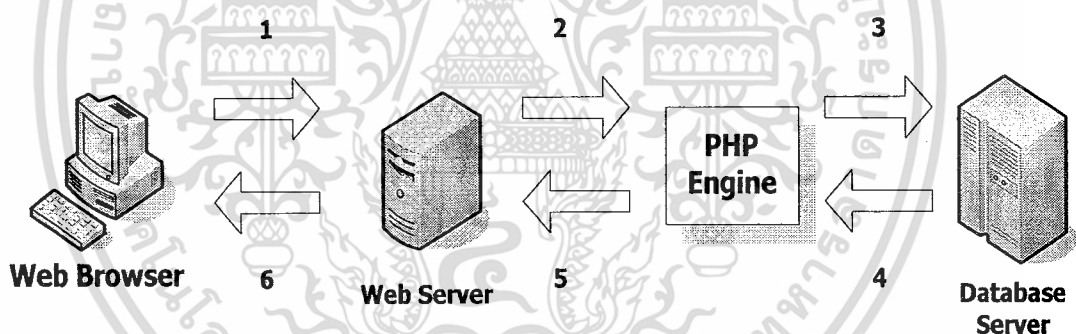
Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย

## 2.11 ระบบจัดการฐานข้อมูล MySQL

MySQL เป็นระบบฐานข้อมูลแบบ Open source ที่ได้รับความนิยมในการใช้งานสูงสุดโปรแกรมหนึ่ง มีความสามารถในการจัดการกับฐานข้อมูลด้วยภาษา SQL (Structures Query Language) อย่างมีประสิทธิภาพ มีความรวดเร็วในการทำงาน รองรับการทำงานจากผู้ใช้หลายๆ คนและหลายๆ งานได้ในขณะเดียวกัน

MySQL ถูกพัฒนาขึ้นโดยบริษัท MySQL AB ประเทศสวีเดน โดยมีลิขสิทธิ์การใช้งาน 2 แบบ คือ ผู้ดูแลระบบสามารถใช้งานซอฟต์แวร์ MySQL ได้โดยไม่มีค่าใช้จ่ายใดๆ ภายใต้ลิขสิทธิ์ของ GPL (General Public License) หรืออาจเลือกใช้แบบที่มีลิขสิทธิ์ทางการค้าของ MySQL AB ซึ่งเป็นผู้ผลิตและพัฒนาซอฟต์แวร์โดยตรงก็ได้ หากไม่ต้องการเกี่ยวข้องกับข้อตกลงเรื่อง GPL

สำหรับการใช้งาน MySQL ร่วมกับ PHP นั้น เมื่อ Web Browser ทำการร้องขอมาที่ Web Server แล้ว PHP จะทำหน้าที่เป็นตัวกลางในการดึงข้อมูลจากฐานข้อมูล หรือนำข้อมูลเข้าไปเก็บยังฐานข้อมูล ซึ่งมีหลักการทำงานดังรูป 2.37



รูปที่ 2.37 หลักการทำงานของระบบฐานข้อมูลร่วมกับ PHP

หลักการทำงานของการติดต่อกับฐานข้อมูลสามารถอธิบายได้ดังนี้

1. Web Browser ทำการร้องขอ Web Page ด้วยโปรโตคอล HTTP ไปยัง Web Server
2. เมื่อ Web Server ได้รับการร้องขอจะทำการเรียกไฟล์ที่ถูกร้องขอแล้วส่งต่อไปยัง PHP Engine เพื่อทำการประมวลผล
3. ในกรณีที่ Script มีคำสั่งให้ทำการติดต่อกับฐานข้อมูลแล้วมีการทำ Query เพื่ออ่านหรือประมวลผลฐานข้อมูล PHP Engine ก็จะทำการติดต่อและส่ง Query ไปยัง Database Server
4. Database Server จะส่งผลลัพธ์ของการ Query กลับไปให้ PHP Engine
5. หลังจาก PHP Engine นำข้อมูลที่ได้รับจาก Database Server มาประมวลผลแล้วจะทำการสร้างผลลัพธ์ในรูปแบบของ HTML และส่งให้กับ Web Server

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาต  
 6. Web Server จะส่งผลลัพธ์ในรูปแบบ HTML กลับไปยัง Web Browser เพื่อทำการแสดงผล  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.12 ระบบสมองกลฝังตัว (Embedded System)

ระบบสมองกลฝังตัวเป็นระบบคอมพิวเตอร์ขนาดเล็กที่ฝังอยู่ในอุปกรณ์ต่างๆ เพื่อให้สามารถใช้งานได้อย่างมีประสิทธิภาพมากขึ้น ในปัจจุบันระบบสมองกลฝังตัวเป็นระบบที่ถูกใช้งานอย่างกว้างขวาง และเข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้น ตั้งแต่การนำพิกเซลข้อมูล มือถือ ไปจนถึงส่วนควบคุมขนาดเล็กในโรงงานอุตสาหกรรมแล้วเป็นผลประจักษ์มาจากระบบสมองกลฝังตัวทั้งสิ้น โดยระบบสมองกลฝังตัวนี้เกิดจากการนำเอาเทคโนโลยีด้านต่างๆ มาประกอบกันเพื่อทำงานเฉพาะทางอย่างใดอย่างหนึ่ง ซึ่งเทคโนโลยีหลักๆ ที่ใช้นั้น ได้แก่ เทคโนโลยีคอมพิวเตอร์ เทคโนโลยีซอฟต์แวร์ เทคโนโลยีสื่อสาร เทคโนโลยีเครือข่าย (Network) เทคโนโลยีการวัดและการควบคุม เป็นต้น โดยข้อดีของระบบสมองกลฝังตัวมีดังนี้

- มีแอปพลิเคชันใช้งานเฉพาะอย่างซึ่งแตกต่างจากระบบไมโครคอมพิวเตอร์ที่ใช้งานได้หลากหลายแอปพลิเคชัน
- ระบบสมองกลฝังตัวถูกออกแบบให้มีขนาดของทรัพยากรเพียงพอต่อการใช้งานเฉพาะอย่าง ทำให้ระบบสมองกลฝังตัวมีขนาดเล็ก
- เนื่องจากระบบสมองกลฝังตัวมีขนาดเล็กทำให้มีการใช้กำลังไฟฟ้าน้อย
- ทนทานต่อสภาวะแวดล้อมที่ไม่ปกติเช่น อุณหภูมิสูง การสั่นสะเทือนที่สูง และบริเวณที่มีสนามแม่เหล็กบริเวณสูง

### 2.12.1 ระบบลินุกซ์แบบฝังตัว (Embedded Linux)

ระบบลินุกซ์แบบฝังตัว คือ การที่เรานำระบบปฏิบัติการลินุกซ์มาติดตั้งลงบนอุปกรณ์แบบสมองกลฝังตัว ซึ่งการนำระบบปฏิบัติการลินุกซ์มาใช้ร่วมกับอุปกรณ์แบบสมองกลฝังตัวนั้นมีข้อดี คือ

1. สามารถลดระยะเวลาในการพัฒนาโดยเราสามารถใช้เวลาในการติดต่อกับอุปกรณ์และระบบเครือข่ายของลินุกซ์มาใช้ได้เลยโดยไม่ต้องพัฒนาขึ้นมาเอง
2. การพัฒนาโปรแกรมข้ามแพลตฟอร์มเราสามารถนำโปรแกรมที่พัฒนาขึ้นบนแพลตฟอร์มหนึ่งไปใช้บนอีกแพลตฟอร์มหนึ่งที่เคอร์เนลสนับสนุนได้โดยการใช้ คอมไพเลอร์ข้ามเครื่อง (Cross Compiler)

โดยลินุกซ์เคอร์เนลที่จะนำมาใช้กับอุปกรณ์สมองกลฝังตัวนั้นจะมีขนาดเล็กกว่าลินุกซ์เคอร์เนลที่ใช้บนเครื่องคอมพิวเตอร์ส่วนบุคคลเนื่องจากอุปกรณ์แบบสมองกลฝังตัวนั้นจะมีข้อจำกัดทางด้านฮาร์ดแวร์ที่มีขนาดจำกัด ซึ่งลินุกซ์เคอร์เนลสำหรับอุปกรณ์สมองกลฝังตัวนั้นก็จะใช้ตัวเดียวกับที่ใช้บนเครื่องคอมพิวเตอร์ส่วนบุคคลแต่นำมาคอมไพล์ (Compile) ใหม่ให้เหมาะสมกับหน่วยประมวลผลที่ใช้ในระบบสมองกลฝังตัวโดยจะมีการตัดส่วนที่ไม่จำเป็นออกเพื่อให้ลินุกซ์เคอร์เนลที่ได้นั้นมีขนาดเล็กลง

## 2.13 ตัวประมวลผล ARM และบอร์ดพัฒนาระบบสมองกลฝังตัว

ในโครงการนี้จะใช้บอร์ดพัฒนารุ่น JampII mini ของบริษัทไชน่า เทคโนโลยี โดยใช้หน่วยประมวลผลสถาปัตยกรรม ARM920T ของบริษัท CIRRUS LOGIC รุ่น EP9302 โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยประมวลผล ARM920T ซึ่งเป็นหน่วยประมวลผลหลักของบอร์ดพัฒนา โดยมีสถาปัตยกรรมแบบ RISC ผลิตโดยบริษัท CIRRUS LOGIC ทำงานที่ความถี่สูงสุด 200 MHz

- มี Instruction Cache ขนาด 16 KB และ Data Cache ขนาด 16 KB

- ความเร็วบัสสูงสุด 100 MHz

- SDRAM ขนาด 32 MB

- Flash Memory ขนาด 8 MB

- พอร์ต USB 2.0 Full Speed Host (OHCI) จำนวน 2 พอร์ต

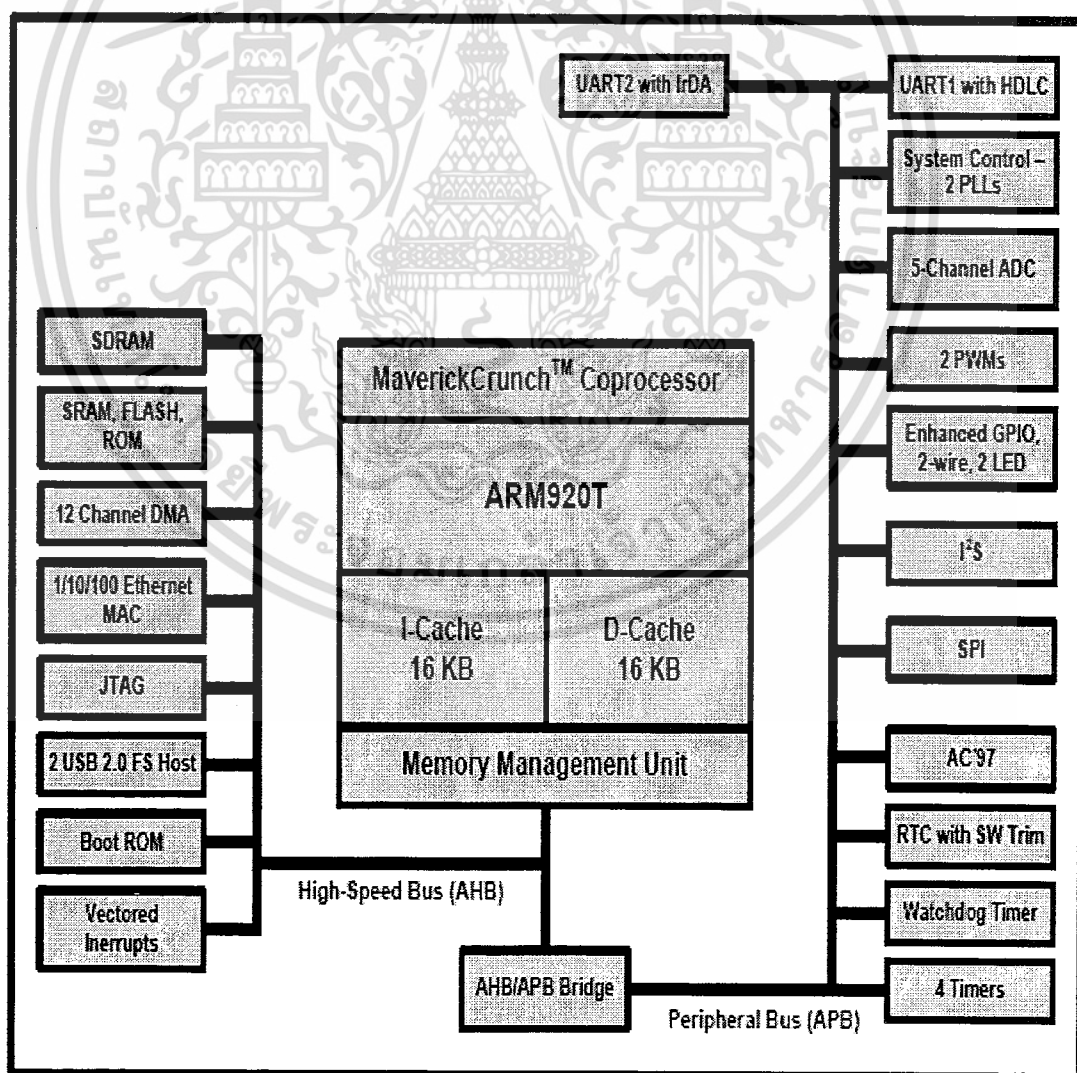
- 20 general purpose I/O

- Ethernet 10/100 Mbps RJ-45 Connector

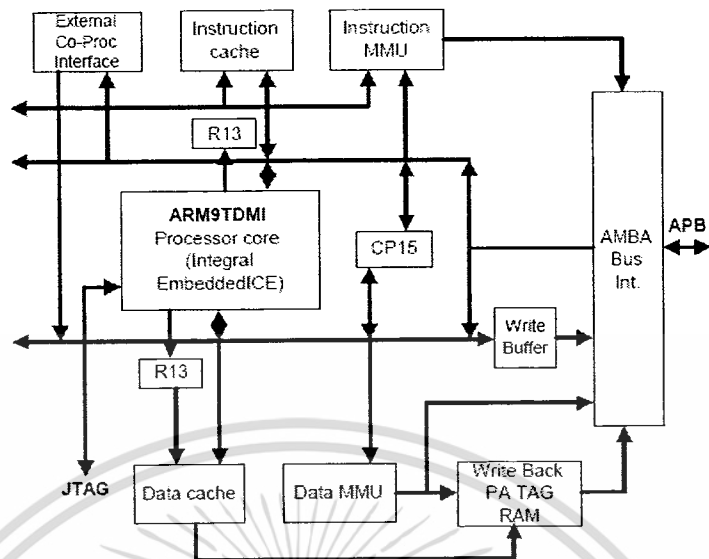
- พอร์ต RS232 (Console Port) และพอร์ต RS232/RS485

- Audio Input/Output Interface (AC'97)

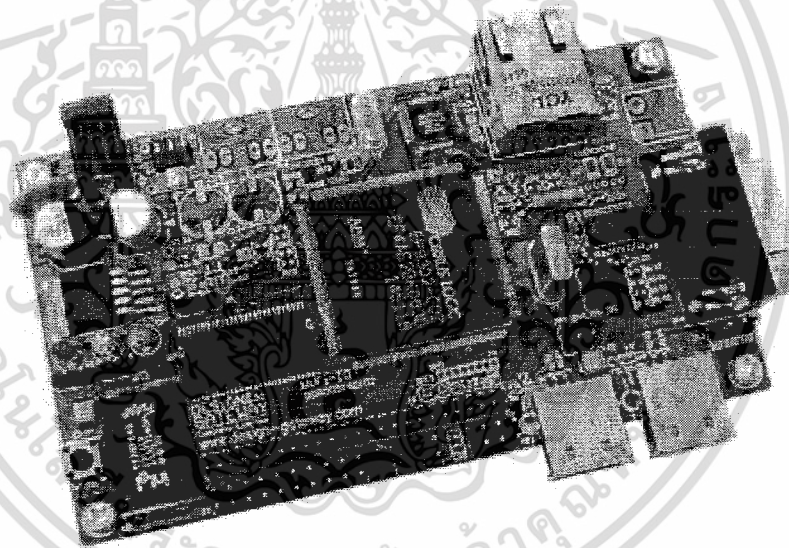
- รองรับระบบปฏิบัติการ Linux kernel 2.6x



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ผู้สอนเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่ไปใช้ประโยชน์ด้านการค้า  
รูปที่ 2.38 บล็อกไดอะแกรมของหน่วยประมวลผล ARM920T  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 บล็อกไดอะแกรมแสดงการทำงานของหน่วยประมวลผล ARM920T



รูปที่ 2.40 บอร์ดพัฒนารุ่น JampII mini

## 2.14 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

### 2.14.1 คุณสมบัติของไมโครคอนโทรลเลอร์เบอร์ P89V51RD2

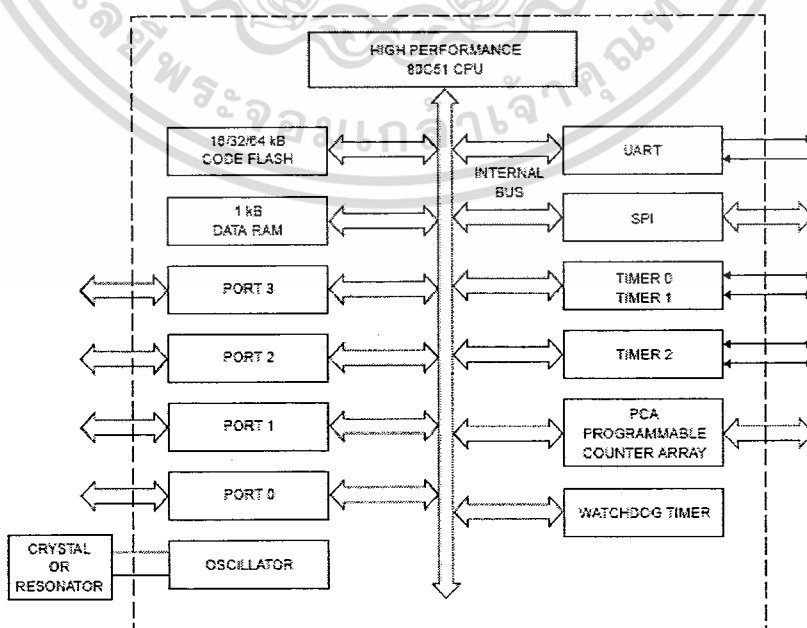
เป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช (Flash Memory) ของบริษัท Phillip Semiconductor ซึ่งปัจจุบันได้เปลี่ยนชื่อเป็น NXP ซึ่งมีคุณสมบัติที่สำคัญดังนี้

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต ที่มีคุณภาพสูงและใช้พลังงานต่ำ
- มีหน่วยความจำภายในแบบแฟลช ขนาด 64 กิโลไบต์ที่โปรแกรมภายในวงจร สามารถเขียนและลบได้ถึงหนึ่งหมื่นครั้ง

- โปรแกรมข้อมูลลงในหน่วยความจำโปรแกรมแบบ ISP (In-System programming) ซึ่งสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
โปรแกรมผ่านพอร์ตอนุกรม RS232 ได้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีขาสัญญาณสำหรับต่อกับอินพุต/เอาต์พุต แบบสองทิศทาง
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- มีแรงดันในการทำงานอยู่ระหว่าง 2.7 -6 โวลต์
- ความถี่สัญญาณนาฬิกาสูงสุด 40 MHz ในกรณีที่ทำงานด้วยสัญญาณนาฬิกา 12 ลูก ต่อ 1 แมกซีนไซเคิล และ 20 MHz ในกรณีที่ทำงานด้วยสัญญาณนาฬิกา 6 ลูกต่อ 1 แมกซีนไซเคิล โดยผู้ผลิตจากโรงงานได้กำหนดให้ทำงานเบื้องต้นในโหมดสัญญาณนาฬิกา 12 ลูกต่อแมกซีนไซเคิล โดยสามารถเปลี่ยนเป็น 6 สัญญาณนาฬิกาต่อแมกซีนไซเคิลได้ด้วยการโปรแกรมแบบ ISP แต่เมื่อเปลี่ยนแล้วหากต้องการเปลี่ยนกลับไปเป็น 12 ลูกต่อแมกซีนไซเคิลต้องใช้การโปรแกรมแบบขนาน
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุตจำนวน 32 ขา
- มีวงจรตั้งเวลาและนับสัญญาณเวลาขนาด 16 บิต 3 ชุด (timer/counter 0, 1 และ 2)
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 8 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรถ่ายสัญญาณนาฬิกาอยู่ในชิป
- มีพอร์ตสื่อสารแบบอนุกรม 1 ช่อง (UART)
- มีรีจิสเตอร์ตัวชี้ตำแหน่งข้อมูลหรือ DPTR 2 ตัว
- มีระบบประหยัดพลังงาน (Low Power Idle And Power Down Mode)
- มีวอตช์ด็อกไทมเมอร์
- มีโมดูลวงจรนับ โปรแกรมได้ (PCA: Programmable Counter Array) ซึ่งบรรจุวงจรจับสัญญาณ (capture), วงจรเปรียบเทียบสัญญาณ (compare), วงจรมอดูเลชันทางความกว้างพัลส์ (PWM) และวอตช์ด็อกไทมเมอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
รูปที่ 2.41 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ P89V51RD2  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

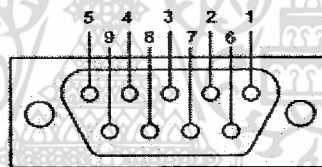
## 2.15 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง ในการนำเอามาตรฐาน RS-232 ไปประยุกต์ใช้งาน จะถูกใช้งานในการส่งข้อมูลระหว่างคอมพิวเตอร์โดยเชื่อมต่อกับโมเด็ม

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เราเห็นได้ชัดคือคอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ทีโมเด็มจะเป็นแบบ DCE

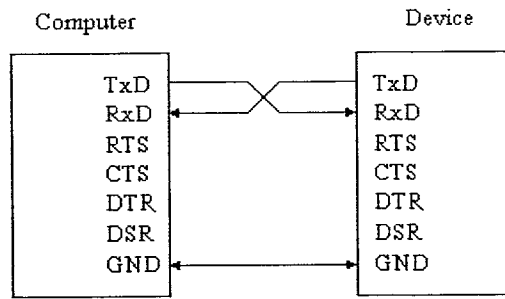
### 2.15.1 รายละเอียดของขาคอนเน็กเตอร์ แบบ DB-9



รูปที่ 2.42 รูปแสดงคอนเน็กเตอร์แบบ DB-9

ตารางที่ 2.3 แสดงการจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมแบบ DB-9

ตำแหน่งขา DB-9	สัญญาณ
1	Data Carrier Detect : DCD
2	Received Data : RxD
3	Transmitted Data : TxD
4	Data Terminal Ready : DTR
5	Signal Ground : GND
6	Data Set Ready : DSR
7	Request To Send : RTS
8	Clear To Send : CTS
9	Ring Indicator : RI

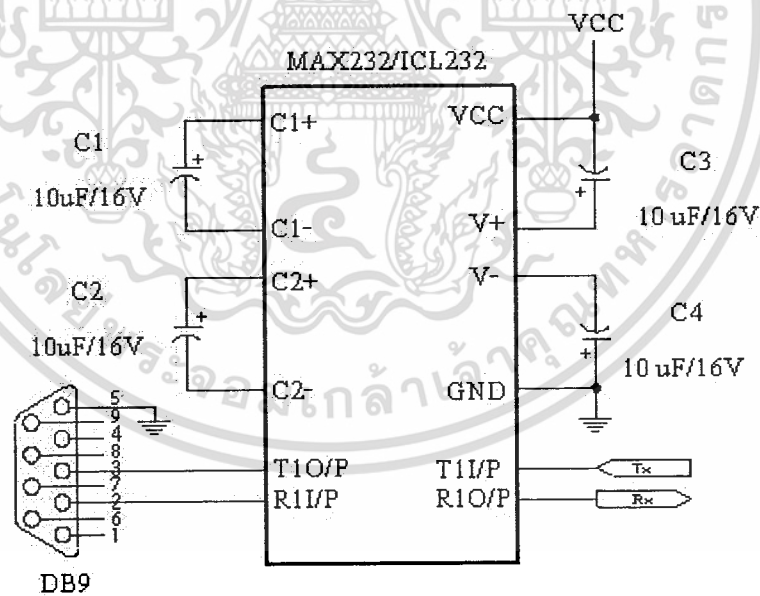


รูปที่ 2.43 แสดงการต่ออุปกรณ์ภายนอกกับคอมพิวเตอร์โดยใช้สัญญาณเพียง 3 เส้น

2.15.2 การแปลงระดับแรงดัน

ในการสื่อสารแบบ RS-232 กำหนดให้แทนระดับแรงดันระหว่าง +3 โวลต์ ถึง +12 โวลต์ มีค่าลอจิกเป็น “1” และระดับแรงดันระหว่าง -3 โวลต์ ถึง -12 โวลต์ มีค่าลอจิกเป็น “0”

ไมโครคอนโทรลเลอร์ส่วนใหญ่จะพิจารณากระดับของแรงดัน 0 V มีค่าลอจิกเป็น “0” และแรงดัน 5 V มีค่าเป็นลอจิก “1” จึงจำเป็นต้องใช้วงจรแปลงระดับแรงดันสำหรับการสื่อสาร RS-232 กับไมโครคอนโทรลเลอร์ นั้นจะใช้ไอซีเบอร์ MAX232 หรือ ICL232



รูปที่ 2.44 แสดงวงจรการแปลงระดับแรงดันของการสื่อสารแบบอนุกรม

การแปลงข้อมูลแบบอนุกรมไปเป็นข้อมูลแบบขนานนั้นจะถูกเคลื่อนเข้าไปใน Shift Register โดยใช้สัญญาณนาฬิกาเป็นตัวควบคุม และหลังจากที่มีการเคลื่อนข้อมูลทุกบิตเข้าไปใน Shift Register ได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้ จะถูกนำออกมาแบบขนานเพื่อนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า อุปกรณ์ที่ทำหน้าที่แปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนานและแปลงข้อมูลแบบขนานเป็นไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลแบบอนุกรม เรียกว่า UART (Universal Asynchronous Receiver-Transmitter) ซึ่งนอกจากจะมีหน้าที่ในการแปลงข้อมูลแล้ว UART ยังมีหน่วยควบคุมและหน่วยตรวจสอบการทำงานอีกด้วย

ในการส่งข้อมูลขนาด 8 บิต แบบอนุกรมนี้จะต้องมีบิตเริ่มต้น (Start Bit) และบิตหยุด (Stop Bit) เพิ่มขึ้นมาซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริงๆ นั้นมีขนาด 10 บิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

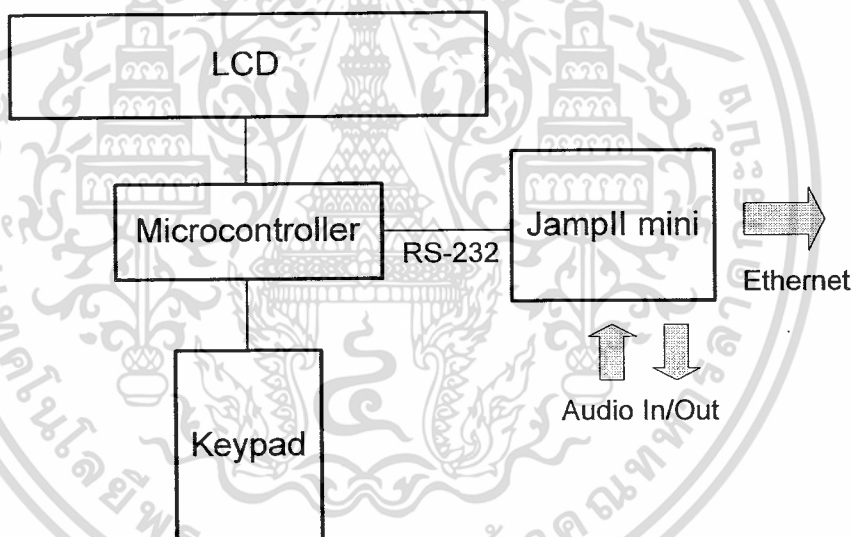
#### การออกแบบและการสร้าง

ในการออกแบบโครงงานนี้จะแบ่งออกเป็น 2 ส่วนหลักๆ คือ

1. ส่วนของโทรศัพท์ที่ใช้ในการสื่อสาร
2. ส่วนของ Server

#### 3.1 การออกแบบในส่วนของโทรศัพท์ที่ใช้ในการสื่อสาร

ในการออกแบบโทรศัพท์ที่ใช้ในการสื่อสารนั้นจะใช้บอร์ดพัฒนา JampII mini เป็นตัวเชื่อมต่อกับระบบเครือข่ายผ่านทางพอร์ต Ethernet (RJ-45) ส่วนการถอดรหัสเสียงและการแสดงผลทาง LCD นั้นจะใช้ไมโครคอนโทรลเลอร์ MCS51 เป็นตัวควบคุมการทำงาน โดยจะสื่อสารผ่านทางพอร์ต RS232 เพื่อรับ-ส่งข้อมูลกับบอร์ด JampII mini



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของโทรศัพท์

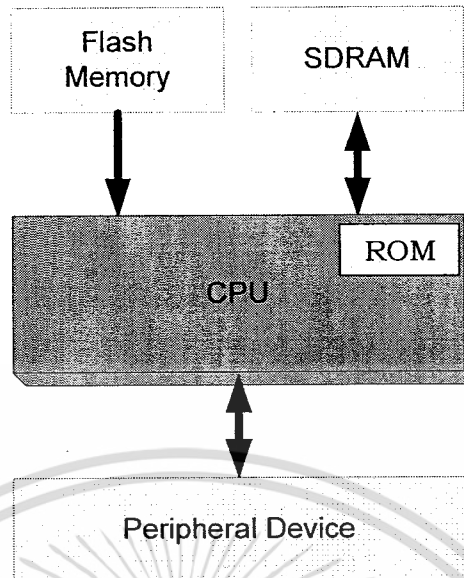
##### 3.1.1 การออกแบบโปรแกรมในส่วนของบอร์ดพัฒนา JampII mini

ในการออกแบบโปรแกรมในส่วนนี้จะเป็นส่วนที่ใช้ในการติดต่อสื่อสารบนเครือข่ายอินเทอร์เน็ตโดยใช้โปรโตคอล SIP ก่อนที่จะไปเขียนโปรแกรมในส่วนของโปรโตคอล SIP เรามาดูการทำงานของบอร์ดพัฒนา JampII mini กันก่อน

##### 3.1.1.1 การทำงานของบอร์ดพัฒนา JampII mini

ในการทำงานของบอร์ดพัฒนา JampII mini จะอยู่ภายใต้ระบบปฏิบัติการลินุกซ์ ซึ่งจะถูกรวมอยู่ใน Flash Memory โดยจะประกอบไปด้วยส่วนของ Kernel และ Root File System ซึ่งจะถูกรวมและโหลดลง SDRAM โดย Boot Loader หลังจากเสร็จกระบวนการนี้แล้วก็จะเข้าสู่การบูตลินุกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ตามปกติ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การทำงานของบอร์ด JampII mini

หลังจากนั้นตัวลินุกซ์ก็จะทำการโหลดไดรเวอร์ของอุปกรณ์ต่างๆ เมื่อเสร็จแล้วก็จะเข้าสู่หน้าจอคอนโซลดังรูปที่ 3.3

```

JampII mini - HyperTerminal
File Edit View Call Transfer Help
[Icons]
scsi0 : SCSI emulation for USB Mass Storage devices
  Vendor: Kingston Model: DataTraveler 2.0 Rev: 1.00
  Type: Direct-Access ANSI SCSI revision: 02
usbcore: registered new driver usb-storage
USB Mass Storage support registered.
SCSI device sda: 1965056 512-byte hdwr sectors (1006 MB)
sda: Write Protect is off
sda: assuming drive cache: write through
sda:<7>usb-storage: queucommand called sda1
Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
get_rtc_time: do command error:-19

Welcome to JAMP-II mini, Design-Gateway Co.,Ltd.

Please press Enter to activate this console.

BusyBox v1.1.3 (2007.05.02-10:58+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

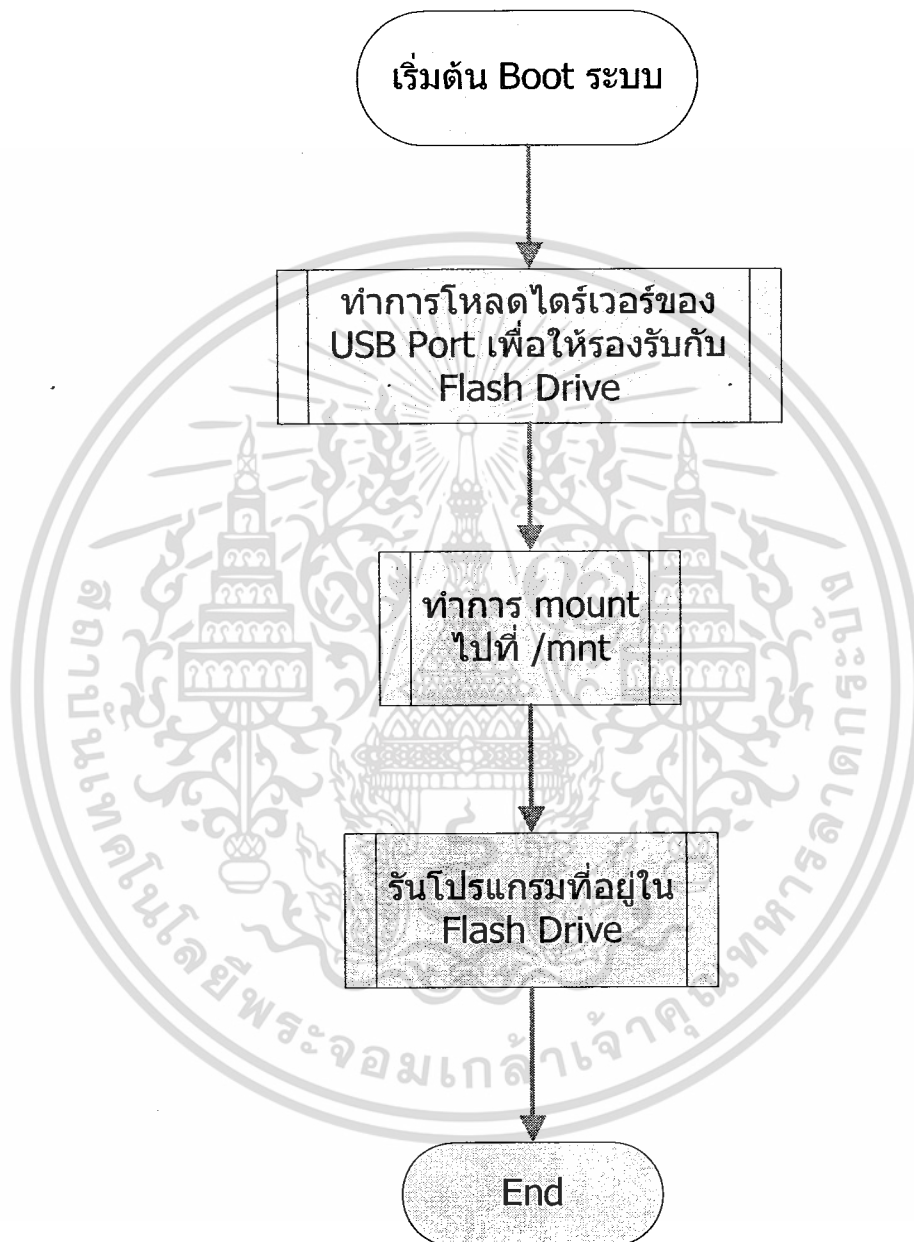
~ #
Connected 0:04:12 | Auto detect | 57600 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo
  
```

รูปที่ 3.3 หน้าจอคอนโซล

### 3.1.1.2 การออกแบบโปรแกรมในส่วนของการติดต่อสื่อสารโดยใช้โปรโตคอล SIP

ในการออกแบบโปรแกรมในส่วนนี้จะใช้ไลบรารี PJSIP ในการสร้างสัญญาณ SIP เพื่อใช้ในการติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตนั้น โดยในขั้นเริ่มต้นเมื่อเปิดระบบขึ้นมาจำเป็นต้องโหลด USB ไม่ว่าจะเป็นใครๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Driver เพื่อให้รองรับกับ Flash Drive ซึ่งเราจะใช้เก็บโปรแกรมที่คอมไพล์ได้ เนื่องจากโปรแกรมนั้นไม่สามารถโหลดรวมไปกับ Root File System ได้เนื่องจาก Flash Memory ของ JampII mini นั้นมีขนาดจำกัด จึงจำเป็นต้องมีการต่อ Flash Drive เพิ่มที่พอร์ต USB จากนั้นเราจะรันไฟล์ที่อยู่ใน Flash Drive แทน



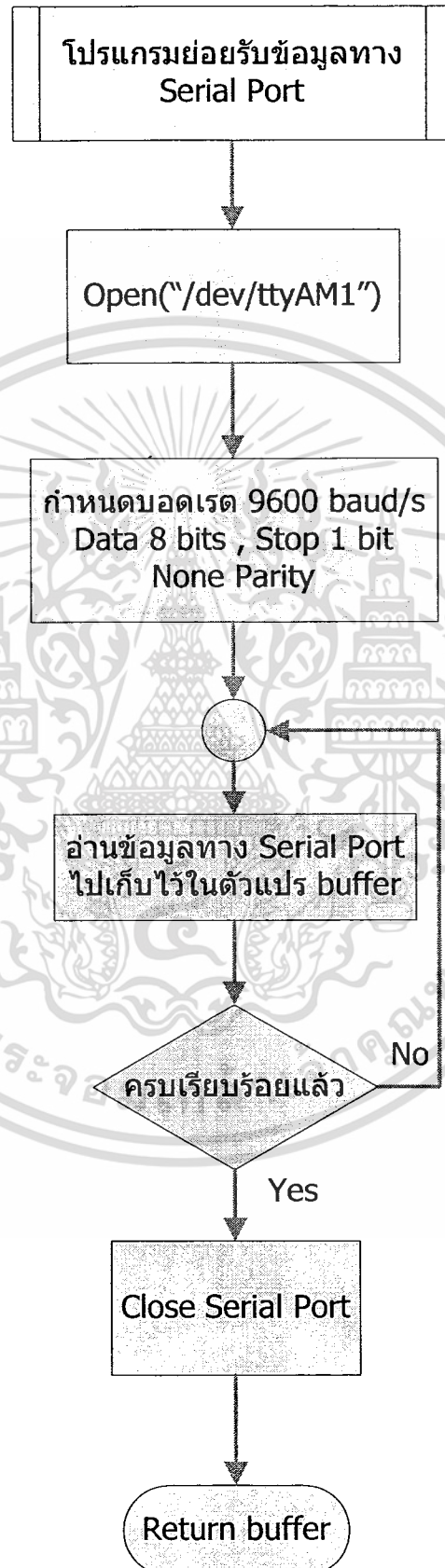
รูปที่ 3.4 โฟลวชาร์ตเริ่มต้นการทำงานของระบบ

โดยก่อนที่จะเริ่ม init ค่าให้กับไลบรารีนั้น จำเป็นต้องมีการตั้งค่า IP Address, Subnet Mask, Default Gateway ให้กับอินเตอร์เฟซของพอร์ต Ethernet ก่อนโดยจะรับค่ามาจากไมโครคอนโทรลเลอร์ผ่านทาง RS232 โดยที่ไมโครคอนโทรลเลอร์จะรับค่ามาจากคีย์แพดอีกทีหนึ่ง

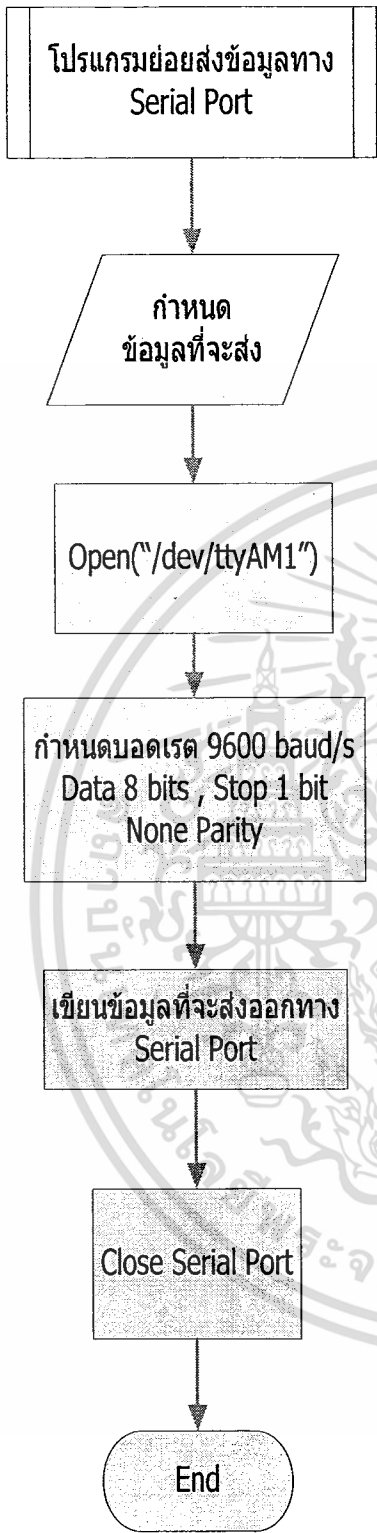
ในการติดต่อผ่านทาง RS232 จะใช้หลักการติดต่ออุปกรณ์เหมือนบนลินุกซ์ทุกๆ ไป โดยที่ Device ต่างๆจะถูกมองเป็นลักษณะของไฟล์ การที่จะเปิด Serial Port จึงสามารถใช้คำสั่ง `open ()` ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในหน่วยงานนี้ ไม่สามารถนำเอกสารนี้ไปเผยแพร่ในที่สาธารณะได้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง Serial Port ของบอร์ดพัฒนา JampII mini นั้นจะอยู่ที่ /dev/ttyAM1 โดยจะมีการทำงานดังรูปที่ 3.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในที่ควรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.5 ฟลิวชาร์ตโปรแกรมย่อยรับข้อมูลทาง Serial Port  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

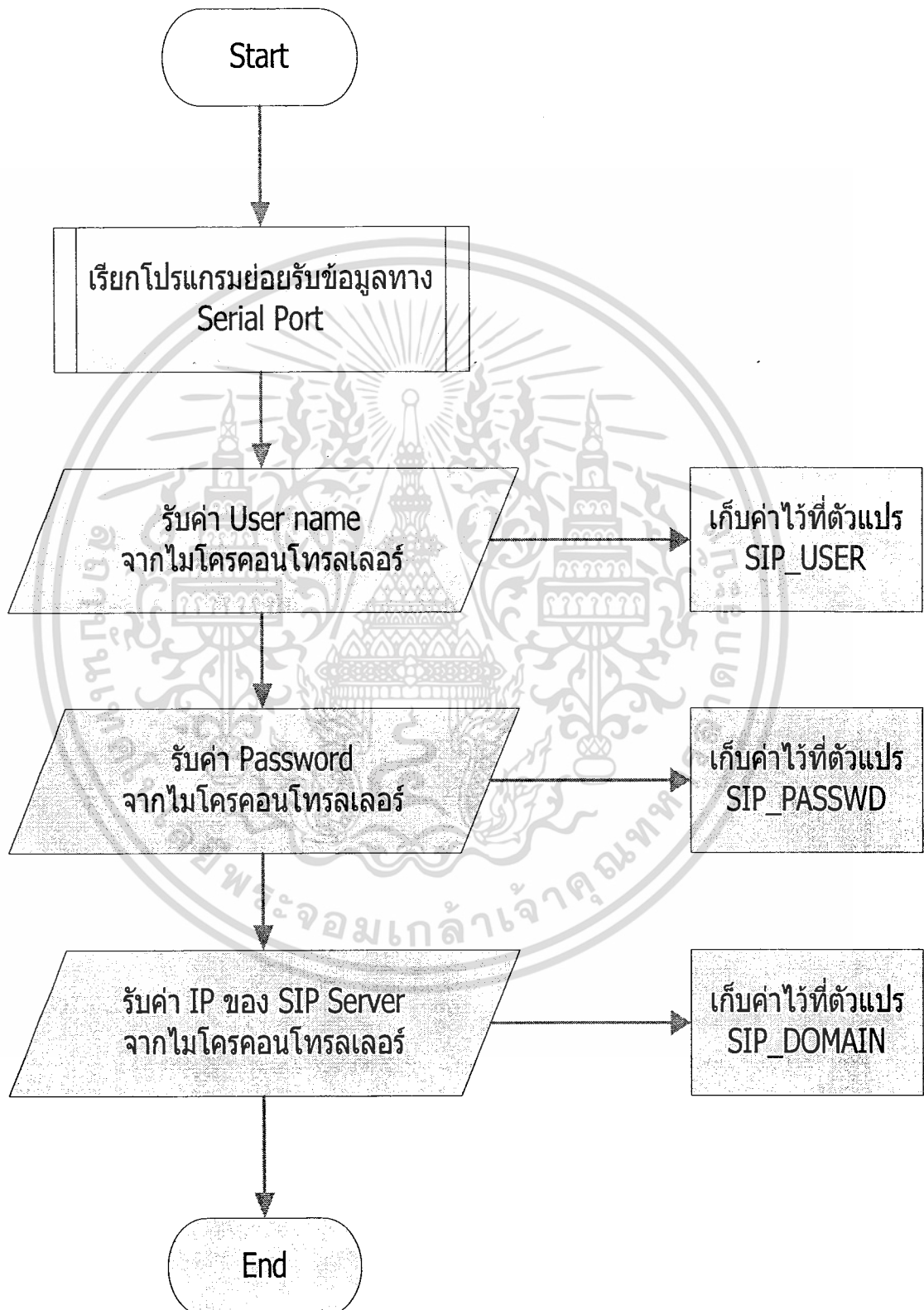


รูปที่ 3.6 โฟลวชาร์ตโปรแกรมย่อยส่งข้อมูล  
ออกทาง Serial Port

รูปที่ 3.7 โฟลวชาร์ตการตั้งค่า IP Address,  
Subnet Mask และ Default Gateway ให้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
อินเทอร์เน็ตเฟซของพอร์ต Ethernet  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นจะเป็นการรับค่า user name และ password เพื่อใช้สำหรับการลงทะเบียนที่ SIP Server ซึ่งจะได้รับมาจากการลงทะเบียนผ่านทางเว็บเพจ โดยจะกล่าวในหัวข้อถัดไป



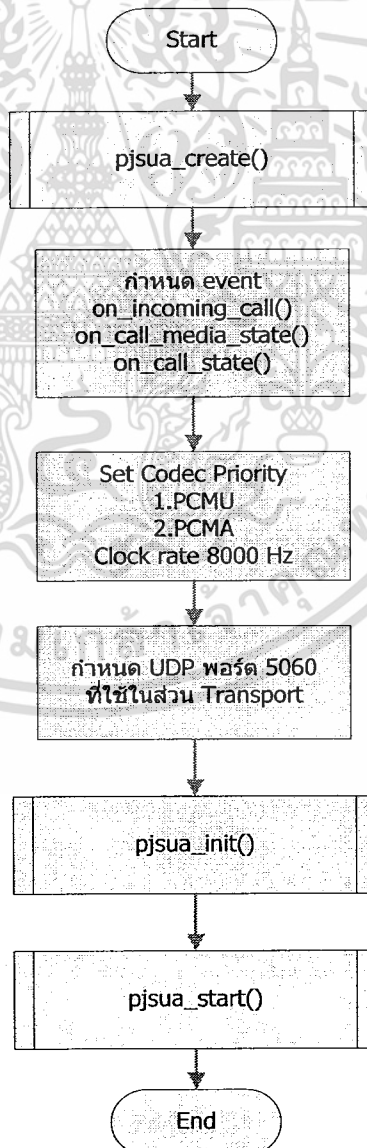
รูปที่ 3.8 โฟลวชาร์ตการรับค่า User name, Password และ IP Address ของ SIP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า เพื่อใช้ในการลงทะเบียนที่ SIP Server ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากรับค่า User name, Password และ IP ของ SIP Server เรียบร้อยแล้ว จากนั้นจะทำการ create pjsua เพื่อเป็นการเริ่มประกาศการใช้งานไลบรารี PJSIP ในส่วนของ PJSUA หลังจากนั้นจะทำการ init ค่าให้กับ pjsua\_config โดยปกติแล้วจะใช้ค่า default ซึ่งจะสามารถเซตผ่านทางฟังก์ชัน pjsua\_config\_default() โดยเราจะเพิ่ม event ที่เกิดขึ้นในขณะที่ใช้งานโทรศัพท์ทั่วไป ได้แก่

- on\_incoming\_call จะถูกเรียกเมื่อได้รับ SIP Message INVITE
- on\_call\_media\_state จะถูกเรียกเมื่อ media\_state มีการเปลี่ยนแปลง
- on\_call\_state จะถูกเรียกเมื่อ call\_state มีการเปลี่ยนแปลงเกิดขึ้น

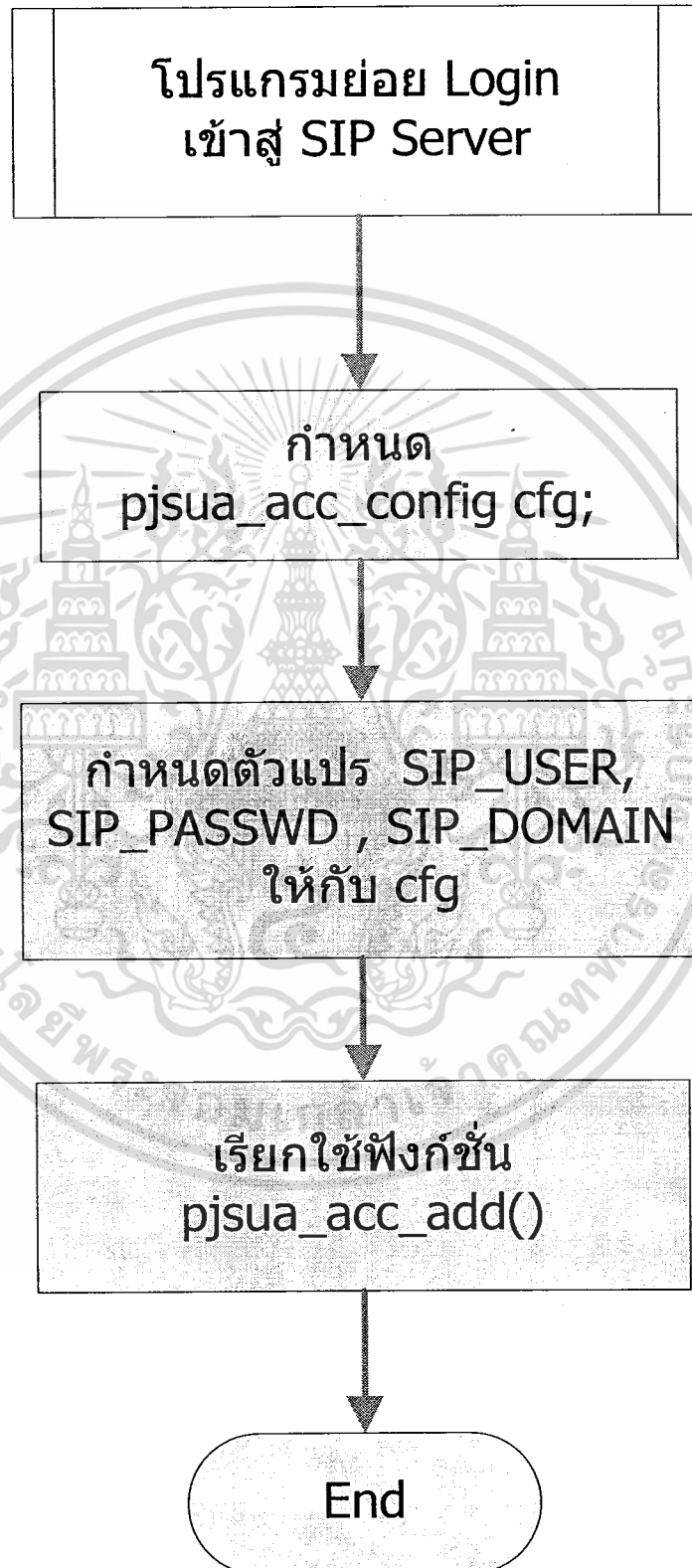
หลังจากนั้นจะทำการ init ค่าของ pjsua\_media\_config ซึ่งจะเป็นค่าที่เกี่ยวกับ media เช่น ความถี่ในการ sampling สัญญาณเสียงจากไมโครโฟน, codec ที่ใช้งาน เป็นต้น หลังจากนั้นจะทำการกำหนดพอร์ตที่ใช้ในการเชื่อมต่อ ซึ่งโดยปกติ SIP Protocol จะใช้ UDP พอร์ต 5060 ในการรับ-ส่งสัญญาณ SIP หลังจากกำหนดค่าเริ่มต้นให้ไลบรารีเสร็จเรียบร้อยแล้วก็จะทำการ start ไลบรารีให้เริ่มทำงานโดยผ่านทางฟังก์ชัน pjsua\_start()



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่ควรนำข้อมูลไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 3.9 การ init ค่าเริ่มต้นให้กับไลบรารี PJSUA  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



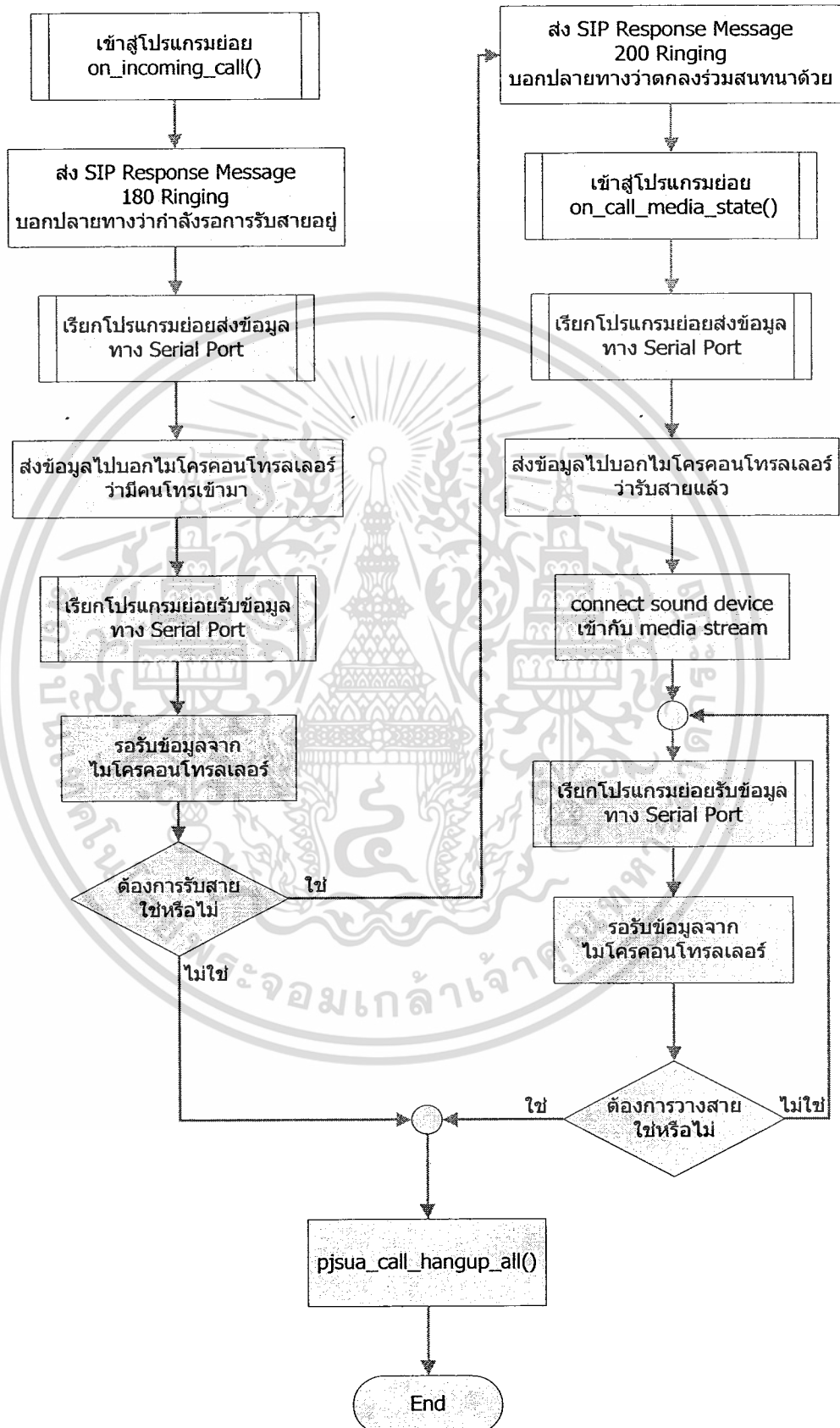
จากขั้นตอนดังรูปที่ 3.10 เราสามารถทำผ่านทางฟังก์ชัน `pjsua_acc_add()` โดยจะมีขั้นตอนการทำงานดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.11 โปรแกรมย่อย Login เข้าสู่ SIP Server  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

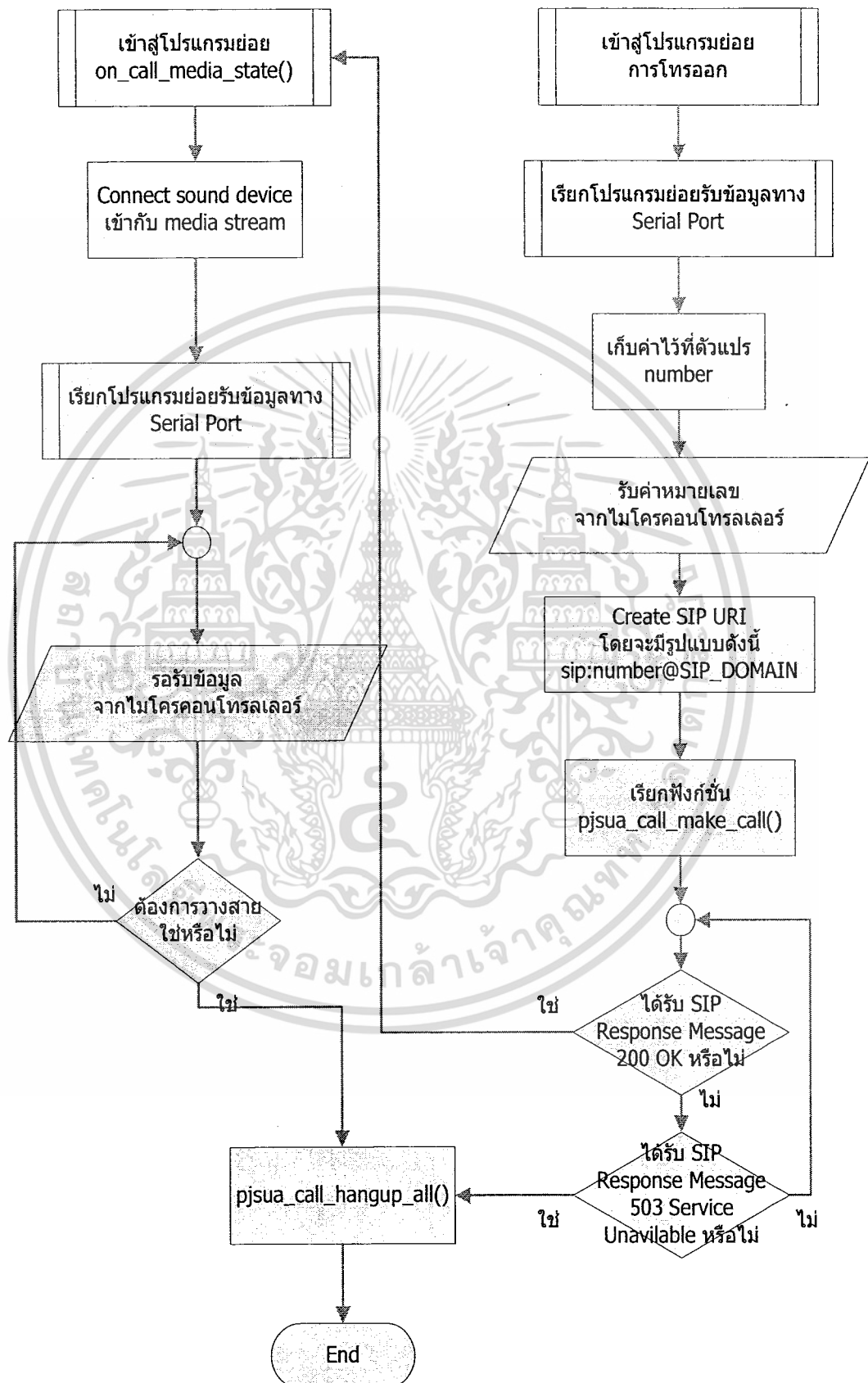
หลังจากนั้นโปรแกรมจะทำการตรวจจับเหตุการณ์ต่างๆ ดังนี้

- เมื่อมีคนโทรเข้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.12 ฟลิวชาร์ตแสดงเหตุการณ์เมื่อมีคนโทรเข้ามา  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

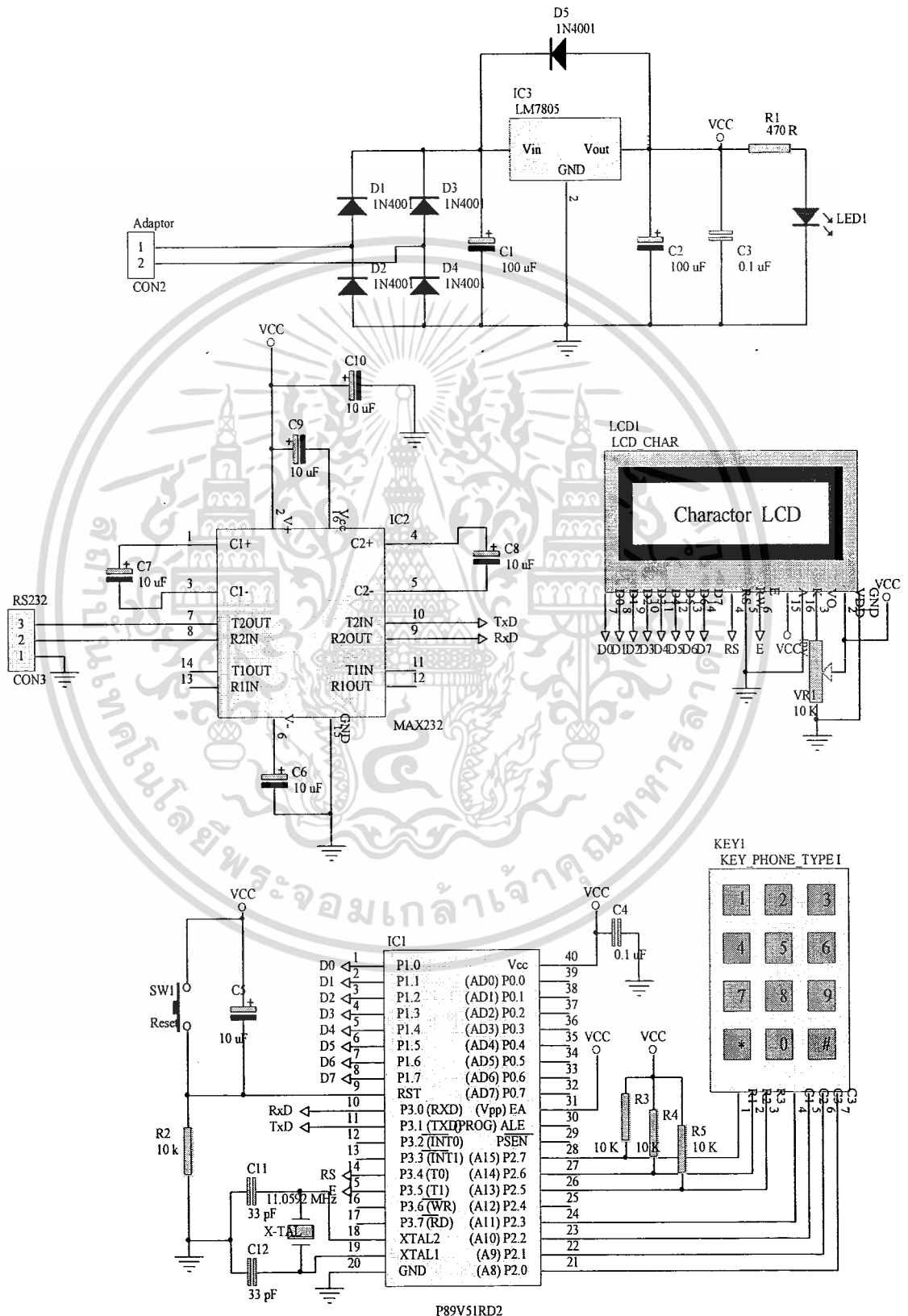
- เมื่อต้องการโทรออก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 3.13 โฟลวชาร์ตแสดงเหตุการณ์เมื่อต้องการโทรออก  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 การออกแบบโปรแกรมในส่วนของไมโครคอนโทรลเลอร์

ในส่วนนี้จะทำหน้าที่รับข้อมูลจากผู้ใช้ผ่านทางคีย์แพด และแสดงผลผ่านทางจอ LCD โดยจะติดต่อกับบอร์ดพัฒนา JampII mini ผ่านทาง Serial Port



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.14 วงจรในส่วนของไมโครคอนโทรลเลอร์ ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

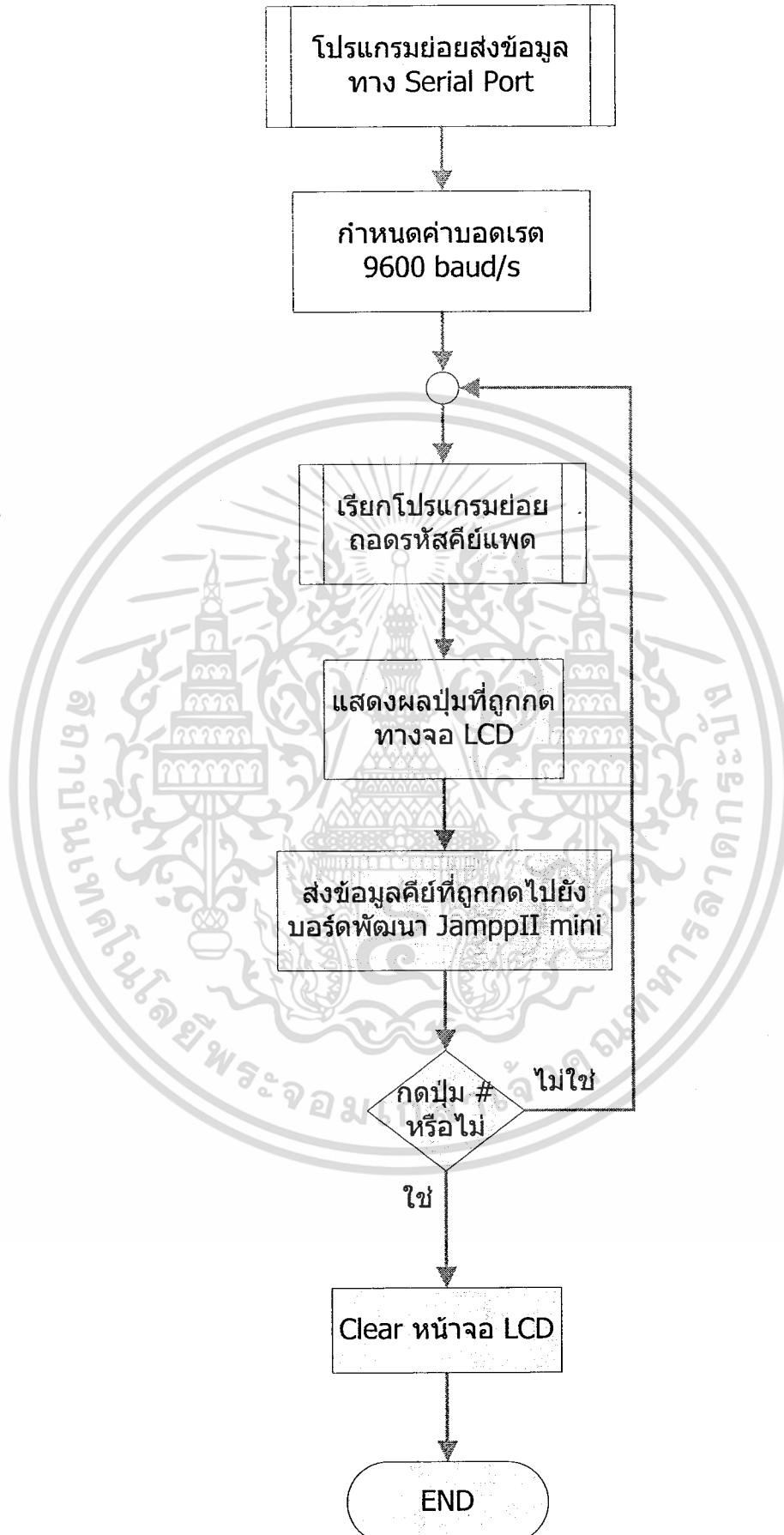


รูปที่ 3.15 โปรแกรมย่อยถอดรหัสคีย์แพด

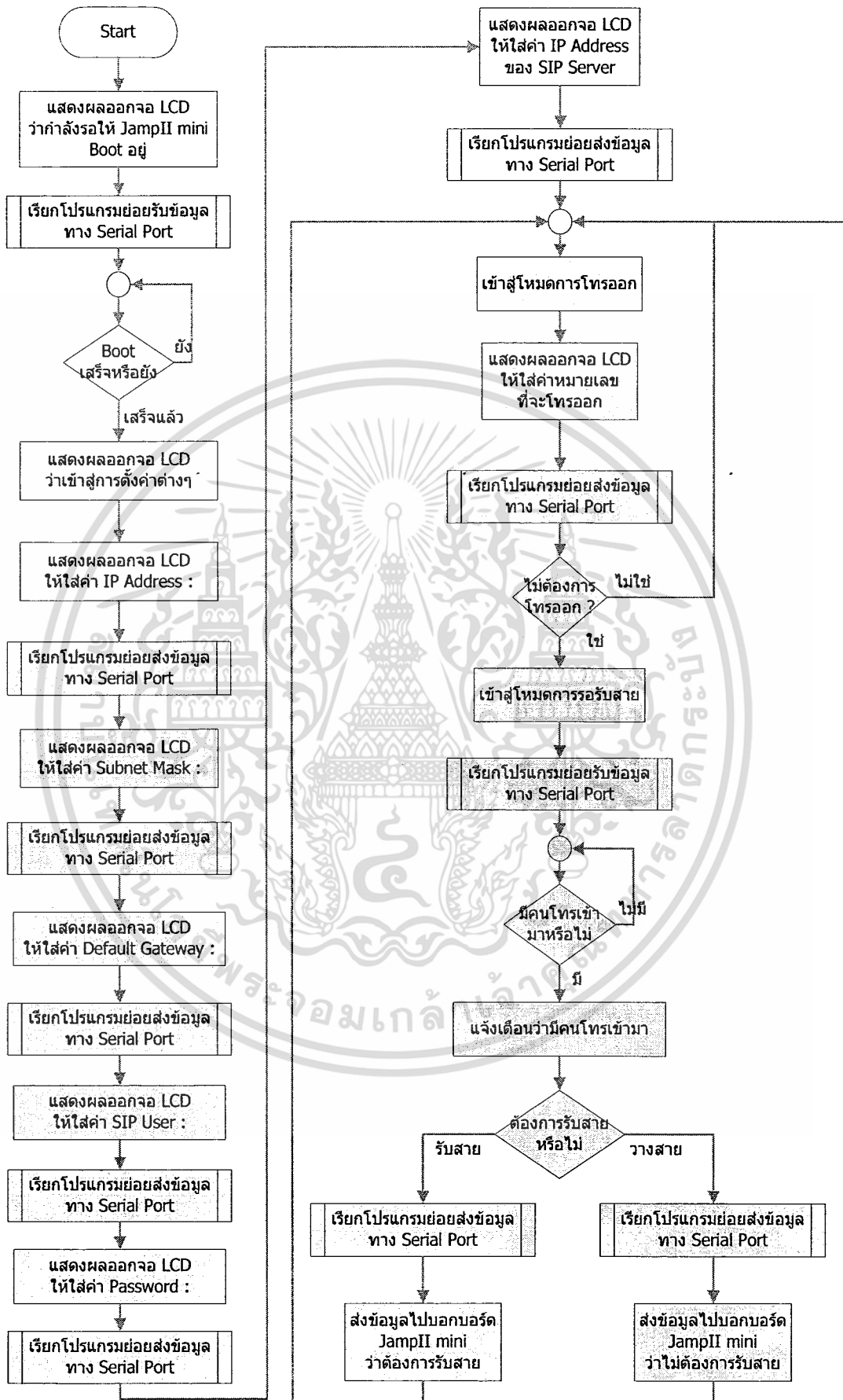


รูปที่ 3.16 โปรแกรมย่อยรับข้อมูลจากบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขหรือใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.17 โปรแกรมย่อยส่งข้อมูลไปยังบอร์ด JamppII mini วัตถุประสงค์ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเฉพาะของส่วนไมโครคอนโทรลเลอร์ รูปที่ 3.18 โฟลวชาร์ตการทำงานของส่วนไมโครคอนโทรลเลอร์ที่ใช้ระบบไอซีของ JampII mini ในการตั้งค่า SIP Server

### 3.2 การออกแบบในส่วนของ Server

การออกแบบในส่วนของ Server จะแบ่งออกเป็น 3 ส่วนหลักๆ คือ

1. การติดตั้ง SIP Server
2. การจัดทำ Web Page ที่ใช้ในการลงทะเบียน
3. การจัดทำระบบฐานข้อมูลที่เก็บข้อมูลผู้ลงทะเบียนและข้อมูลการใช้งานใช้โทรศัพท์

โดยผู้ใช้งานจะลงทะเบียนผ่าน Web Page หลังจากนั้น Web Page จะทำการส่งข้อมูลไปให้กับ SIP Server และข้อมูลของผู้ใช้งานจะถูกเก็บไว้ในระบบฐานข้อมูล โดยสามารถสมัครใช้งานได้ที่เว็บไซต์ <http://161.246.18.134> หรือทาง <http://www.voip-project.com> หลังจากนั้นผู้ใช้งานจะได้รับหมายเลข ID และ Password เพื่อใช้ในการล็อกอินสู่ระบบ ซึ่งสามารถใช้ได้กับเครื่องโทรศัพท์ที่เป็น IP หรือใช้โปรแกรม Soft Phone บนเครื่องคอมพิวเตอร์ก็ได้

#### 3.2.1 การติดตั้ง SIP Server

การติดตั้ง SIP Server จะใช้โปรแกรม Asterisk เพื่อทำเป็น SIP Server โดยที่เราสามารถดาวน์โหลดโปรแกรมได้จากเว็บไซต์ <http://www.asterisk.org/> ซึ่งตัวโปรแกรม Asterisk สามารถทำงานได้หลายๆ ระบบปฏิบัติการเช่น Linux, OpenBSD, FreeBSD, Sun Solaris เป็นต้น ซึ่งในโครงการนี้จะใช้ระบบปฏิบัติการ Linux RedHat ในการติดตั้งเนื่องจากมีแผ่นอยู่แล้ว

##### 3.2.1.1 การติดตั้งโปรแกรม Asterisk

หลังจากทำการติดตั้งระบบปฏิบัติการ Linux เสร็จเรียบร้อยแล้ว จากนั้นจะทำการติดตั้งโปรแกรม Asterisk โดยเมื่อดาวน์โหลดมาเสร็จแล้ว ไฟล์ติดตั้งจะอยู่ในรูปแบบของไฟล์ asterisk-version.tar.gz ซึ่งเป็นไฟล์ที่ถูกบีบอัดอยู่จึงต้องทำการแตกไฟล์ข้อมูลออกมาก่อนโดยใช้คำสั่ง

```
# tar xvfz asterisk-version.tar.gz
```

ซึ่ง version คือ version ของ Asterisk ที่ใช้ในการติดตั้ง ซึ่งในขณะที่ติดตั้งผู้จัดทำได้ใช้ version 1.4.20.1 หลังจากนั้น โดยให้เข้าไปที่ Directory ที่ได้ทำการแตกไฟล์ออกมา แล้วทำการ Compile และติดตั้งโปรแกรม โดยใช้คำสั่ง

```
# make clean
    (เป็นการลบข้อมูลที่มีการ compile มาก่อนหน้านี้)
# ./configure
    (เป็นการโหลดตัว compiler ที่ใช้ในการ compile)
# make
    (ทำการ compile ตัว source code โปรแกรม)
# make install
    (ทำการติดตั้งโปรแกรมที่ได้จากการ compileแล้ว)
# make samples
    (เป็นการติดตั้งไฟล์ที่ใช้ในการตั้งค่าของระบบ)
# make config
    (เป็นการกำหนดให้ Asterisk รันแบบ service ซึ่งสามารถเรียกใช้งานได้โดยใช้คำสั่ง service asterisk start)
```

หลังจากติดตั้งโปรแกรม Asterisk แล้วเราสามารถเรียกใช้งานได้หลายคำสั่ง เช่น

```
# service asterisk start
# /etc/init.d/asterisk start
# asterisk -vvvvvc
```

หลังจาก Asterisk ทำงานแล้ว เราสามารถเข้าสู่หน้า Console ของ Asterisk ได้โดยใช้คำสั่ง

```
# asterisk -r
```

ซึ่งจะปรากฏหน้าจอของ Asterisk CLI (Command Line Interface) ดังรูปที่ 3.19

```
[root@voipproject ~]# asterisk -r
Asterisk 1.4.20.1, Copyright (C) 1999 - 2009 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details
s.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.4.20.1 currently running on voipproject (pid = 8378)
Verbosity is at least 3
voipproject*CLI> █
```

รูปที่ 3.19 หน้าจอของ Asterisk CLI

ในการสั่งงานในโหมด CLI จะต้องใช้คำสั่งของระบบ Asterisk เท่านั้น โดยสามารถพิมพ์ help เพื่อดูคำสั่งการใช้งานต่างๆได้

หลังจากติดตั้งโปรแกรมเสร็จเรียบร้อยแล้วนั้น ไฟล์ที่ใช้ในการปรับแต่งค่าของระบบต่างๆ จะถูกเก็บไว้ที่ /etc/asterisk ดังรูปที่ 3.20

```
[root@voipproject /]# cd /etc/asterisk/
[root@voipproject asterisk]# ls
ads_i.conf          dundi.conf        logger.conf        res_smp.conf
adtranvoivr.conf  enum.conf         manager.conf       rpt.conf
agents.conf        extconfig.conf   meetme.conf        rtp.conf
alarmreceiver.conf extensions.ael     mgcp.conf          say.conf
alsa.conf          extensions.conf  miadn.conf         sip.conf
amd.conf           extensions.conf~ modules.conf        sip.conf~
asterisk.ads_i    features.conf    musiconhold.conf  sip_notify.conf
asterisk.conf     festival.conf    muted.conf         skinny.conf
cdr.conf          followme.conf    ooh323.conf       sla.conf
cdr_custom.conf   func_odb.conf    osp.conf           smdi.conf
cdr_manager.conf  gtalk.conf       oss.conf           telcordia-1.ads_i
cdr_mysql.conf    h323.conf        phone.conf         udptl.conf
cdr_odb.conf      http.conf        privacy.conf       users.conf
cdr_pgsql.conf    iax.conf         queues.conf        voicemail.conf
cdr_tds.conf      iaxprov.conf     res_mysql.conf     vpb.conf
codecs.conf       indications.conf res_odb.conf       zapata.conf
dnsmgr.conf       jabber.conf      res_pgsql.conf
[root@voipproject asterisk]# █
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.20 ไฟล์ที่ใช้ในการปรับแต่งค่าของระบบ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในการสร้าง SIP Server นั้นจะการปรับแต่งค่าพารามิเตอร์ของระบบ นั้นจะใช้ไฟล์ sip.conf และ extensions.conf ซึ่งอยู่ในรูปแบบของ Text File โดยที่แต่ละไฟล์จะมีหน้าที่ดังนี้

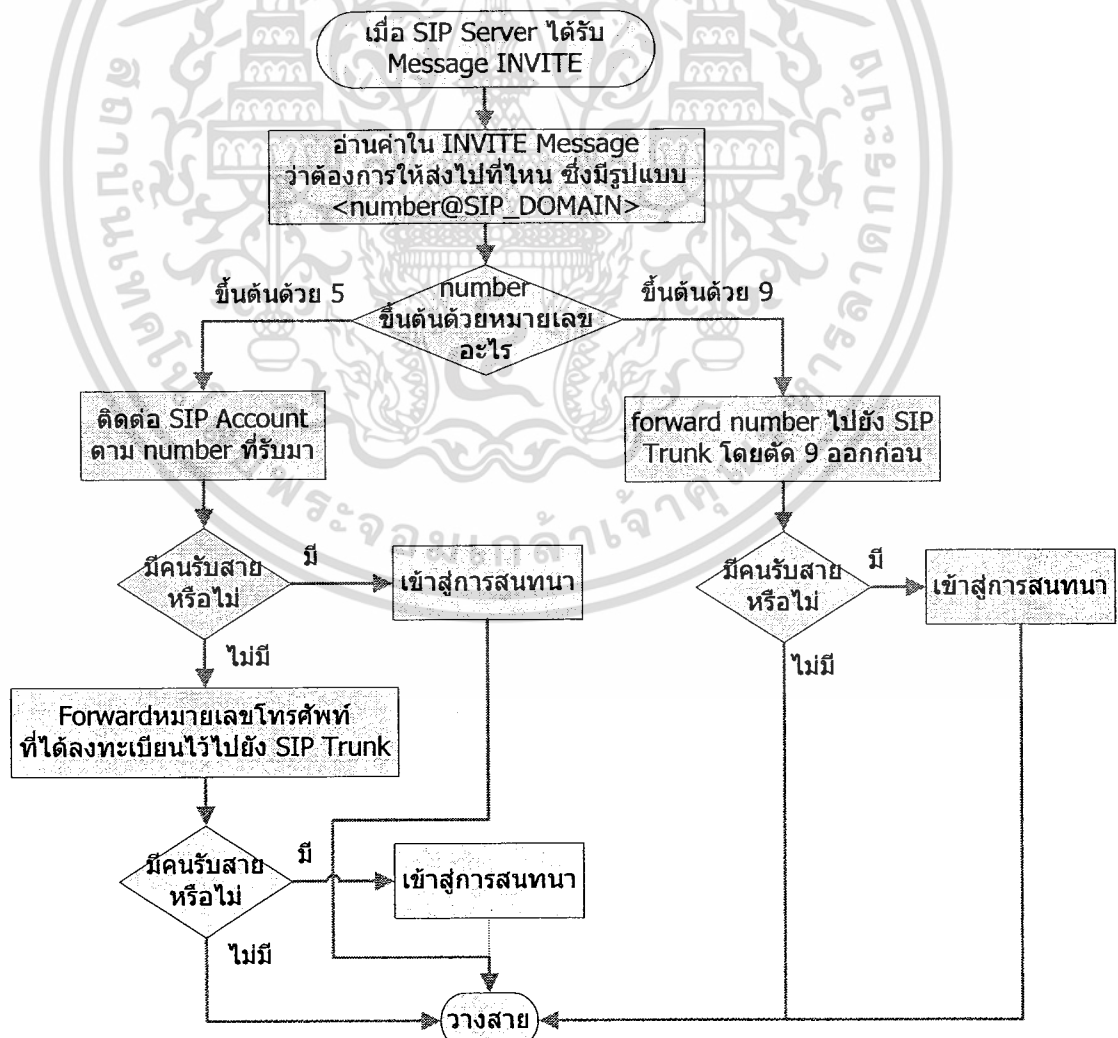
- sip.conf เป็นไฟล์ที่ใช้ปรับแต่งค่าพารามิเตอร์ที่ใช้กับโปรโตคอล SIP เช่น ใช้ในการกำหนด username , password ที่ใช้ในการ Login เข้าสู่ระบบ

- extensions.conf เป็นไฟล์ที่ใช้ในการควบคุมหมายเลขของโทรศัพท์ เช่น เมื่อโทรศัพท์กดปุ่ม 300 ให้ไปเรียกไปที่ SIP Account 300 เป็นต้น

โดยปกติไฟล์คอนฟิกต่างๆ สามารถใช้ Text Editor บน Linux ไม่ว่าจะเป็น vi, pico, nano เปิดไฟล์ขึ้นมาแล้วทำการแก้ไขได้ ซึ่งหลังจากเราทำการแก้ไขไฟล์คอนฟิกของระบบแล้ว จำเป็นต้องสั่งให้ Asterisk โหลดค่าไฟล์คอนฟิกเข้าไปใหม่ ซึ่งจะใช้คำสั่ง

```
# asterisk -rx "sip reload"
# asterisk -rx "extensions reload"
```

### 3.2.1.2 Dial Plan ของระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในรูปแบบที่ 3.2.1 Dial Plan ของระบบเท่านั้นไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dial Plan นับเป็นหัวใจของระบบโทรศัพท์ Asterisk หรือจะเรียกว่าเป็นแผนการโทรศัพท์ก็ได้ เนื่องจาก Dial Plan จะเป็นตัวกำหนดว่าเมื่อมีการเรียกสายเข้ามาที่ Asterisk แล้วจะให้ทำอะไรบ้าง เช่น หมุนโทรศัพท์ไปยังหมายเลขใด หรือจะให้ทำการบันทึกเสียง หรือจะให้พูดข้อความใดๆ ออกมาก็ได้ โดยในโปรแกรม Asterisk จะมีการเขียน Dial Plan ไว้ในไฟล์ extensions.conf โดยในโครงงานนี้จะกำหนดให้ใช้หมายเลขตั้งแต่ 5000, 5001, 5002, ... เป็นหมายเลขภายใน ส่วนการโทรออกไปหาโทรศัพท์มือถือ หรือ โทรศัพท์บ้านนั้น จะต้องกด 9 นำหน้าหมายเลขที่ต้องการจะโทร เช่น ต้องการโทรหาเบอร์โทรศัพท์มือถือหมายเลข 0814445555 ก็ต้องกด 90814445555 โดยการโทรออกไปหมายเลขภายนอกเราจะใช้ SIP Trunk ของ TOTnetcall โดย SIP Server ของเราจะต้องลงทะเบียนกับ SIP Server ของ TOT netcall ไว้ก่อน ซึ่งถ้าติดต่อหมายเลขภายในไม่ได้ก็จะโทรออกไปที่หมายเลขที่ผู้ใช้ได้ลงทะเบียนไว้ผ่านทาง SIP trunk ซึ่งมีหลักการทำงานรูปที่ 3.21

### 3.2.2 การจัดทำ Web Page ที่ใช้ในการลงทะเบียน

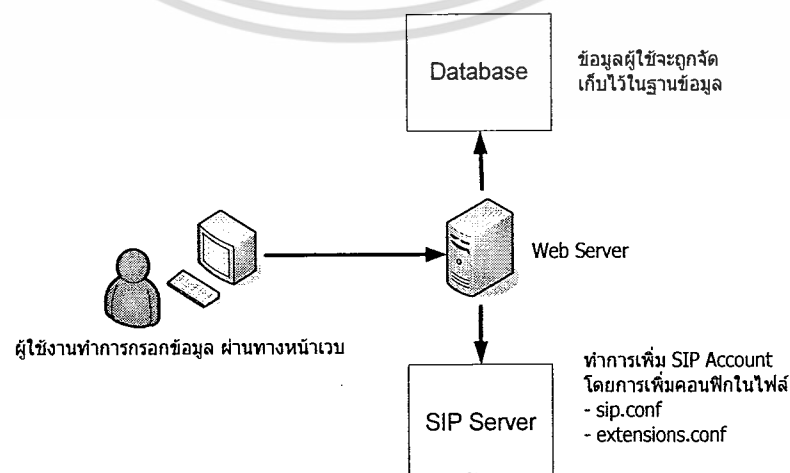
#### 3.2.2.1 การจัดเตรียม Web Server

ในการจัดทำ Web Page ที่ใช้ในการลงทะเบียนนั้น จะใช้ภาษา PHP ในการเขียน ซึ่งจะถูกประมวลผลโดย Web Server โดยในโครงงานนี้จะใช้โปรแกรม Apache ซึ่งทำหน้าที่เป็น Web Server โดยติดตั้งมาพร้อมกับระบบปฏิบัติการ Linux RedHat อยู่แล้ว ซึ่งเราสามารถเรียกใช้งานโดยใช้คำสั่ง

```
# service httpd start หรือคำสั่ง
# /etc/init.d/httpd start
```

โดยที่ Directory ที่ใช้ในการเก็บ Web Page จะอยู่ที่ /var/www/html/ เราสามารถนำ Web Page ที่เขียนเสร็จแล้วมาเก็บไว้ในโพลเดอร์นี้ ซึ่งจะสามารถเรียกดูผ่านทาง IP Address ของเครื่อง Server ซึ่งตั้งไว้ที่ IP 161.246.18.134 หรือเข้าผ่านทาง Domain Name ทาง <http://www.voip-project.com> ซึ่งจะทำการ Link มาที่เครื่อง Server โดยอัตโนมัติ

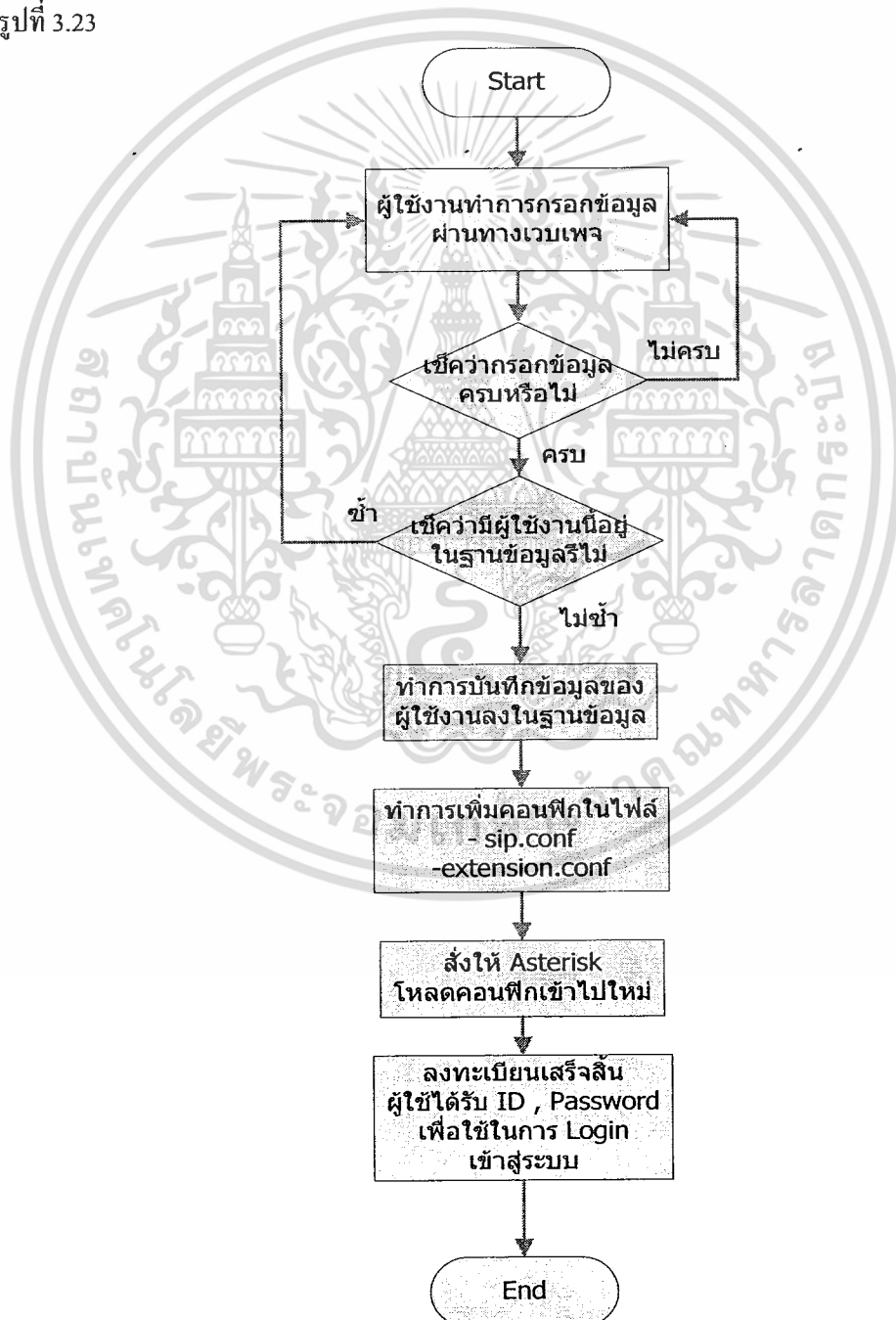
#### 3.2.2.2 การจัดทำ Web Page ลงทะเบียน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
รูปที่ 3.22 หลักการทำงานของ Web Page ที่ใช้ลงทะเบียน

ในส่วนของการจัดทำ Web Page นั้นจะใช้ภาษา PHP เขียนขึ้นมา เพื่อทำการติดต่อกับระบบฐานข้อมูลและ ใช้ในการเพิ่มคอนฟิกไฟล์ของโปรแกรม Asterisk ซึ่งมีหลักการทำงานดังรูปที่ 3.21

เมื่อผู้ใช้งานทำการลงทะเบียนผ่าน Web Page เสร็จแล้ว ข้อมูลของผู้ใช้งานจะถูกส่งไปเก็บยังฐานข้อมูล โดยในฐานข้อมูลก็จะมีหมายเลข ID ของข้อมูลในแต่ละ record ที่ส่งเข้าไปเก็บซึ่งในโครงการนี้เราจะใช้หมายเลข ID มาเป็นหมายเลขโทรศัพท์ หลังจากได้หมายเลข ID จากฐานข้อมูลแล้ว ก็จะทำการเพิ่มหมายเลขใน SIP Server ซึ่งต้องไปเพิ่มที่ไฟล์คอนฟิก sip.conf และ extensions.conf ซึ่งมีลักษณะเป็น Text File หลังจากการแก้ไขไฟล์เสร็จเรียบร้อยแล้ว จะต้องมีการสั่งให้โปรแกรม Asterisk รีโหลดไฟล์คอนฟิกเข้าไปใหม่ หมายเลขที่ลงทะเบียนถึงจะใช้งานได้ ซึ่งสามารถสรุปขั้นตอนทั้งหมดเป็นโฟลวชาร์ตดังรูปที่ 3.23



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ที่ดูแลการเรียนการสอนเท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากระบบปฏิบัติการ Linux นั้นได้มีระบบรักษาความปลอดภัยโดยที่คำสั่งบางคำสั่งจะสามารถใช้งานได้เฉพาะ root เท่านั้น (root คือ user ที่มีสิทธิ์สูงสุดในระบบ) ซึ่งในการทำงานของ Web Page นั้น จะประมวลผลโดย Web Sever ซึ่งในที่นี้คือโปรแกรม Apache โดยเมื่อ Apache รันคำสั่งใดๆ ก็ตามก็จะถือว่าเป็น user หนึ่ง โดยใช้ชื่อว่า "apache" ซึ่งเราต้องกำหนดให้ user "apache" ทำการใช้คำสั่งสั่งงานให้โปรแกรม Asterisk โหลดคอนฟิกเข้าไปให้ได้ โดยใช้คำสั่ง

```
# visudo
```

แล้วทำการเพิ่มบรรทัดนี้ลงไป

```
apache ALL=NOPASSWD : ALL
```

เป็นการอนุญาตให้ apache สามารถรันคำสั่งต่างๆ ได้ ซึ่งในภาษา PHP จะมีฟังก์ชันที่ใช้ในการสั่งงาน shell อยู่คือ exec() โดยที่เราสามารถสั่งงานให้โปรแกรม Asterisk โหลดคอนฟิกเข้าไปใหม่จะใช้คำสั่งดังนี้

```
exec ("sudo asterisk -rx \"sip reload\"");
exec ("sudo asterisk -rx \"extensions reload\"");
```

### 3.2.3 การจัดทำระบบฐานข้อมูลที่ใช้เก็บข้อมูลผู้ลงทะเบียนและข้อมูลการใช้งานโทรศัพท์

#### 3.2.3.1 การจัดเตรียมโปรแกรมฐานข้อมูล

ในโครงการนี้จะใช้โปรแกรม MySQL ในการจัดทำฐานข้อมูลซึ่งติดตั้งมากับระบบปฏิบัติการ Linux อยู่แล้วเช่นเคย ซึ่งเราสามารถเรียกใช้งานได้หลายคำสั่ง เช่น

```
# service mysqld start
# /etc/init.d/mysqld start
```

โดยเราสามารถเข้าสู่หน้าจอ Command ได้จากคำสั่ง

```
# mysql -u root -p
```

ระบบจะทำการถาม password จากนั้นจะเข้าสู่หน้าจอ Command ดังรูปที่ 3.24 โดยเราสามารถใส่ภาษา SQL (Structured Query Language) ในการจัดการกับฐานข้อมูลได้

```
[root@voipproject ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 4.1.12

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> █
```

รูปที่ 3.24 หน้าจอ Command ของ MySQL

#### 3.2.3.2 การจัดทำระบบฐานข้อมูลของผู้ใช้งาน

เอกสารนี้เป็นเอกสารในการสร้างตารางเก็บข้อมูลผู้ใช้งานประกอบด้วย ไฟล์ ข้อมูลดังนี้ ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ID	ใช้กำหนดหมายเลขโทรศัพท์โดยจะให้เรียงตามลำดับลงไปเรื่อยๆ
USERNAME	ใช้เก็บ username ของผู้ใช้งาน
PASSWORD	ใช้เก็บ password ของผู้ใช้งาน
E-MAIL	ใช้เก็บ E-mail ของผู้ใช้งาน
NAME	ใช้เก็บชื่อของผู้ใช้งาน
LASTNAME	ใช้เก็บนามสกุลของผู้ใช้งาน
SEX	ใช้เก็บเพศของผู้ใช้งาน
TEL	ใช้เก็บเบอร์โทรศัพท์ของผู้ใช้งาน
NICKNAME	ใช้เก็บชื่อเล่นของผู้ใช้งาน

```
mysql> desc member;
```

Field	Type	Null	Key	Default	Extra
ID	int(10) unsigned		PRI	NULL	auto_increment
USERNAME	varchar(100)	YES		NULL	
PASSWORD	varchar(100)	YES		NULL	
EMAIL	varchar(100)	YES		NULL	
NAME	varchar(100)	YES		NULL	
LASTNAME	varchar(100)	YES		NULL	
SEX	varchar(50)	YES		NULL	
TEL	varchar(50)	YES		NULL	
NICKNAME	varchar(100)	YES		NULL	

9 rows in set (0.00 sec)

```
mysql>
```

รูปที่ 3.25 โครงสร้างของตาราง member

### 3.2.3.3 การจัดทำระบบฐานข้อมูลการใช้งานโทรศัพท์

ในปกติแล้ว Asterisk จะทำการเก็บข้อมูลการใช้โทรศัพท์อยู่ในรูปของ Log File ซึ่งอยู่ที่ /var/log/asterisk/cdr-csv/Master.csv ซึ่งเราสามารถใช้โปรแกรม Excel บน Window เปิดดูข้อมูลการใช้งาน

โทรศัพท์ได้ ดังรูปที่ 3.26

โดยภายในไฟล์ Master.csv จะประกอบไปด้วย field ข้อมูลดังนี้

calldate	วันที่และเวลาที่มีการเรียกสายไปยังหมายเลขปลายทาง
clid	เป็นหมายเลขโทรศัพท์โชว์เบอร์ที่จะถูกแสดงเมื่อเรามีการเรียกไปยังปลายทาง
src	หมายเลขโทรศัพท์ต้นทาง
dst	หมายเลขโทรศัพท์ปลายทางที่ถูกเรียก
dcontext	เป็นชื่อของ Context ที่เริ่มต้นทำงานโดยค่ามาตรฐาน จะเป็น default
channel	เป็นช่องสัญญาณเสียง (Voice Channel) ที่ถูกกำหนดไว้เป็นเส้นทางในการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูอาจารย์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ การดัดแปลงหรือการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การนำเอกสารนี้ไปใช้ในเชิงพาณิชย์โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

สัญญาณเสียงไปยังผู้รับ

dstchannel	เป็นช่องสัญญาณเสียง (Voice Channel) ปลายทางที่ถูกเรียก
lastapp	เป็นส่วนของรายละเอียดโปรแกรม (Application) ใช้ในการติดต่อการเรียกสาย
lastdata	เป็นข้อมูลสุดท้ายที่มีการเรียกใช้งาน
duration	เป็นช่วงเวลาระหว่างเมื่อมีผู้เรียกสายไปหาปลายทางจนถึงสิ้นสุดการติดต่อกัน ซึ่งจะมีหน่วยเป็นวินาที
billsec	เป็นช่วงเวลาตั้งแต่มีการรับสายจากปลายทาง เมื่อมีการติดต่อได้ มีหน่วยเป็นวินาที
disposition	สถานการณ์ใช้งานครั้งสุดท้ายของการใช้โทรศัพท์ เช่น ANSWERED, NO ANSWER, BUSY, FAILED เป็นต้น
amaflags	เป็นคุณสมบัติเกี่ยวกับเรื่อง Automatic Machine Accounting Flags ซึ่งจะมีการใช้งานเฉพาะบางโปรแกรมที่การทำงานเกี่ยวกับเรื่องของ Billing
accountcode	เป็นค่าของ Caller ID ที่ได้กำหนดไว้ให้แต่ละเลขหมาย ซึ่งจะถูกกำหนดไว้ในไฟล์ sip.conf
userfield	เป็น field ที่ผู้ใช้งานระบุค่าเองได้

โดยที่เราสามารถให้ Asterisk บันทึกข้อมูลการใช้โทรศัพท์ลงในฐานข้อมูล MySQL ได้ซึ่งจะทำให้เราสามารถวิเคราะห์ข้อมูลหรือจัดการกับข้อมูลการใช้งานโทรศัพท์ได้สะดวกขึ้น โดยเราจะต้องสร้างฐานข้อมูลใน MySQL เตรียมพร้อมไว้ก่อนและกำหนดคสิทธิการใช้งานฐานข้อมูล ในโครงการนี้จะใช้ชื่อ Database ว่า Asterisk และ ตารางชื่อ cdr โดยมีรายละเอียด field ข้อมูลดังรูปที่ 3.27

	B	C	D	E	F	G	H	I	J
1	5002	5001	from-internal	"PhonerLite" <5002>	SIP/5002-08aeaea0	SIP/5001-08af4350	Dial	SIP/5001	11/6/2008 12:29
2	5001	5002	from-internal	"AKE" <5001>	SIP/5001-08aeaea0	SIP/5002-08af4350	Dial	SIP/5002	11/6/2008 12:29
3	5001	5002	from-internal	"AKE" <5001>	SIP/5001-08aed808	SIP/5002-08aeaea0	Dial	SIP/5002	11/6/2008 12:31
4	5002	5001	from-internal	"PhonerLite" <5002>	SIP/5002-08aed808	SIP/5001-08aeaea0	Dial	SIP/5001	11/6/2008 12:31
5	5001	5002	from-internal	"AKE" <5001>	SIP/5001-08a98b70	SIP/5002-08a9dbf0	Dial	SIP/5002	11/6/2008 12:33
6	5002	5001	from-internal	"PhonerLite" <5002>	SIP/5002-08a9dbd8	SIP/5001-08a98b70	Dial	SIP/5001	11/6/2008 12:33
7	5001	5002	from-internal	"AKE" <5001>	SIP/5001-08850e80	SIP/5002-08856178	Dial	SIP/5002	12/6/2008 11:22
8	5002	5001	from-internal	"PhonerLite" <5002>	SIP/5002-08856160	SIP/5001-08851270	Dial	SIP/5001	12/6/2008 11:22
9	5002	5001	from-internal	"WaY" <5002>	SIP/5002-08754458	SIP/5001-0875af78	Dial	SIP/5001	17/6/2008 8:10
10	5001	5002	from-internal	"AKE" <5001>	SIP/5001-08754458	SIP/5002-08755af0	Dial	SIP/5002	17/6/2008 8:10
11	5002	5001	from-internal	"WaY" <5002>	SIP/5002-08755af0	SIP/5001-0875af38	Dial	SIP/5001	17/6/2008 8:17
12	5003	5001	from-internal	"WaY" <5003>	SIP/5003-08755af0	SIP/5001-0875af38	Dial	SIP/5001	17/6/2008 8:28
13	5002	5003	from-internal	"Project Room (WaY)" <5002>	SIP/5002-08755af0	SIP/5003-0875af38	Dial	SIP/5003	17/6/2008 8:30
14	5003	5002	from-internal	"WaY" <5003>	SIP/5003-08754458	SIP/5002-08755af0	Dial	SIP/5002	17/6/2008 8:31
15	5003	5002	from-internal	"WaY" <5003>	SIP/5003-08754458	SIP/5002-08755af0	Dial	SIP/5002	17/6/2008 8:32
16	5002	5003	from-internal	"Project Room (WaY)" <5002>	SIP/5002-088c95d8	SIP/5003-088cc710	Dial	SIP/5003	17/6/2008 8:35
17	5001	5002	from-internal	"AKE" <5001>	SIP/5001-088caff0	SIP/5002-088cc580	Dial	SIP/5002	17/6/2008 8:35
18	5001	5001	from-internal	"AKE" <5001>	SIP/5001-088caff0	SIP/5001-088cc580	Dial	SIP/5001	17/6/2008 8:36
19	5001	5001	from-internal	"AKE" <5001>	SIP/5001-088cc580	SIP/5001-088d14d0	Dial	SIP/5001	17/6/2008 8:40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะภายในเท่านั้น การนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย  
 รูปที่ 3.26 ไฟล์ Master.csv เมื่อใช้โปรแกรม Excel เปิด  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mysql> desc cdr;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default          | Extra |
+-----+-----+-----+-----+-----+-----+
| calldate   | datetime      |      | MUL | 0000-00-00 00:00:00 |      |
| clid       | varchar(80)   |      |      |                    |      |
| src        | varchar(80)   |      |      |                    |      |
| dst        | varchar(80)   |      | MUL |                    |      |
| dcontext   | varchar(80)   |      |      |                    |      |
| channel    | varchar(80)   |      |      |                    |      |
| dstchannel | varchar(80)   |      |      |                    |      |
| lastapp    | varchar(80)   |      |      |                    |      |
| lastdata   | varchar(80)   |      |      |                    |      |
| duration   | int(11)       |      |      | 0                  |      |
| billsec    | int(11)       |      |      | 0                  |      |
| disposition | varchar(45)   |      |      |                    |      |
| amaflags   | int(11)       |      |      | 0                  |      |
| accountcode | varchar(20)   |      | MUL |                    |      |
| userfield  | varchar(255)  |      |      |                    |      |
| uniqueid   | varchar(32)   |      |      |                    |      |
+-----+-----+-----+-----+-----+-----+
16 rows in set (0.00 sec)

mysql>
```

รูปที่ 3.27 โครงสร้างของตารางของ cdr

หลังจากสร้างตารางเสร็จเรียบร้อยแล้ว ให้ทำการเปิดไฟล์ `cdr_mysql.conf` เพื่อทำการแก้ไขพารามิเตอร์ต่างๆ ให้เข้ากับระบบดังนี้

```
[global]
hostname = localhost
dbname = asterisk
table = cdr
user = asterisk
password = *****
port = 3306
sock = /var/lib/mysql/mysql.sock
userfield = 1
```

โดยที่เราสามารถตรวจสอบว่า Databases ที่สร้างขึ้นมานั้นทำงานถูกต้องหรือไม่โดยเข้าไปที่ CLI ของ Asterisk แล้ว ใช้คำสั่ง `cdr mysql status` ดังรูปที่ 3.28

```
[root@voipproject asterisk]# asterisk -r
Asterisk 1.4.20.1, Copyright (C) 1999 - 2008 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for detail
3.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.

=====
Connected to Asterisk 1.4.20.1 currently running on voipproject (pid = 8378)
Verbosity is at least 3
voipproject*CLI> cdr mysql status
Connected to asterisk@localhost, port 3306 using table cdr for 1 days, 21 hours,
26 minutes, 32 seconds.
Wrote 0 records since last restart.
voipproject*CLI>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.28 สถานะของฐานข้อมูลเมื่อเชื่อมต่อกับ Asterisk  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mysql> select calldate,src,dst,billsec from cdr;
```

calldate	src	dst	billsec
2008-08-24 00:20:34	5019	9025172715	5
2008-08-29 22:25:09	5023	90814010453	6
2008-08-29 23:36:31	5024	90814010453	0
2008-08-29 23:36:47	5024	90814014053	0
2008-08-29 23:39:13	5024	5023	0
2008-08-29 23:39:33	5023	5024	6
2008-08-29 23:39:47	5024	90814010453	12
2008-08-29 23:42:34	5024	5023	4
2008-08-29 23:42:56	5024	5023	6
2008-08-29 23:43:39	5024	90832029986	12
2008-08-29 23:44:08	5024	90814010453	14
2008-08-30 01:50:11	5024	90832029926	16
2008-08-31 01:04:24	5025	9025172715	11
2008-08-31 01:04:42	5025	90814010453	7
2008-09-02 01:36:16	5025	90859377902	166
2008-10-12 18:17:36	5037	90814010453	32
2008-10-12 18:18:17	5037	9025170715	9
2008-10-12 18:19:01	5037	90814010453	22
2008-10-12 18:19:35	5037	9084010453	13
2008-10-12 18:19:53	5037	90814010453	32
2008-10-13 18:43:03	5023	90814010453	30
2008-10-14 21:06:45	5039	90838882972	11

รูปที่ 3.29 ข้อมูลการใช้งานโทรศัพท์เมื่อจัดเก็บอยู่ในฐานข้อมูล

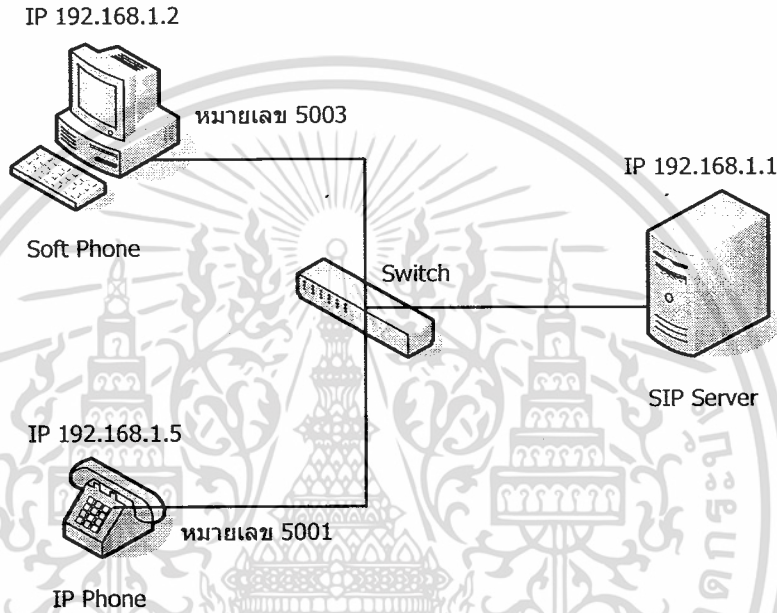
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 4

### ผลการทดลอง และวิเคราะห์ผลการทดลอง

#### 4.1 ผลการทดลองในส่วนของโทรศัพท์

การทดลองในส่วนนี้เราจะใช้โปรแกรม VMware จำลองคอมพิวเตอร์ขึ้นมาเพื่อใช้ทำเป็น SIP Server เพื่อให้ง่ายต่อการดักจับแพ็กเก็ต โดยจะมี Topology ในการทดสอบดังนี้



รูปที่ 4.1 Topology ที่ใช้ในการทดลอง

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.5	192.168.1.1	UDP	source port: sip destination port: sip
2	4.842415	0.0.0.0	255.255.255.255	DHCP	DHCP:discover - transaction ID: Def15225f
3	7.802716	0.0.0.0	255.255.255.255	DHCP	DHCP:discover - transaction ID: Def15225f
4	8.000015	0.0.0.0	255.255.255.255	DHCP	DHCP:discover - transaction ID: Def15225f
5	14.996733	192.168.1.5	192.168.1.1	UDP	source port: sip destination port: sip
6	28.993499	192.168.1.1	192.168.1.255	NBNS	name query NB MYGROUP-ID
7	34.829611	192.168.1.1	192.168.1.255	NBNS	name query NB MYGROUP-ID
8	36.830629	192.168.1.1	192.168.1.255	NBNS	name query NB MYGROUP-ID
9	38.830786	192.168.1.1	192.168.1.255	NBNS	name query NB MYGROUP-ID
10	38.830896	192.168.1.1	192.168.1.255	NBNS	name query NB MYGROUP-ID
11	44.990211	192.168.1.5	192.168.1.1	UDP	source port: sip destination port: sip
12	51.829158	192.168.1.1	192.168.1.255	BROWSER	browser election request
13	51.873132	192.168.1.1	192.168.1.255	BROWSER	browser election request
14	55.836603	192.168.1.1	192.168.1.255	BROWSER	browser election request
15	57.836573	192.168.1.1	192.168.1.255	BROWSER	browser election request
16	57.836797	192.168.1.1	192.168.1.255	NBNS	registration NB <01><02>_#SERVSE<>2<><01>
17	59.872705	192.168.1.1	192.168.1.255	NBNS	registration NB <01><02>_#SERVSE<>2<><01>
18	59.872703	192.168.1.1	192.168.1.255	NBNS	registration NB <01><02>_#SERVSE<>2<><01>
19	59.898698	192.168.1.1	192.168.1.255	UDP	source port: sip destination port: sip
20	61.872419	192.168.1.1	192.168.1.255	NBNS	registration NB <01><02>_#SERVSE<>2<><01>
21	61.872611	192.168.1.1	192.168.1.255	NBNS	registration NB MYGROUP-ID
22	61.985044	192.168.1.1	192.168.1.255	NBNS	registration NB MYGROUP-ID
23	61.985206	192.168.1.1	192.168.1.255	NBNS	registration NB MYGROUP-ID
24	64.981822	00:bad:0:bad:00:00	vmware:3a:37:51	ARP	who has 192.168.1.1? Tell 192.168.1.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.2 โปรแกรม Wireshark ที่ใช้ในการดักจับแพ็กเก็ต ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนในการดักจับแพ็กเก็ตนั้นจะใช้โปรแกรม Wireshark ในการดักจับแพ็กเก็ตที่วิ่งไป-มา ระหว่างอินเตอร์เฟซของพอร์ต Ethernet

#### 4.1.1 ผลการทดลองในส่วนของการ Register กับ SIP Server

ในขั้นตอนนี้จะให้โทรศัพท์ Register เข้ากับ SIP Server ซึ่งจะกำหนด Account ให้เป็นหมายเลข 5001

No.	Time	Source	Destination	Protocol	Info
311	1677.168365	00:ba:d0:0b:ad:00	Vmware_5a:37:51	ARP	Who has 192.168.1.1? Tell 192.168.1.5
312	1677.168474	vmware_5a:37:51	00:ba:d0:0b:ad:00	ARP	192.168.1.1 is at 00:0c:29:5a:37:51
313	1705.433040	192.168.1.5	192.168.1.1	SIP	Request: REGISTER sip:192.168.1.1
314	1705.433645	192.168.1.1	192.168.1.5	SIP	Status: 100 Trying (1 bindings)
315	1705.433804	192.168.1.1	192.168.1.5	SIP	Status: 401 Unauthorized (0 bindings)
316	1705.708066	192.168.1.5	192.168.1.1	SIP	Request: REGISTER sip:192.168.1.1
317	1705.708621	192.168.1.1	192.168.1.5	SIP	Status: 100 Trying (1 bindings)
318	1705.714051	192.168.1.1	192.168.1.5	SIP	Status: 200 OK (1 bindings)

รูปที่ 4.3 แพ็กเก็ตในขณะทำการ Register

จากรูป 4.3 เมื่อเรากำหนด IP Address ให้กับตัวโทรศัพท์เรียบร้อยแล้วหลังจากนั้นจะทำการกำหนด username , password ที่ใช้ในการล็อกอินเข้าสู่ SIP Server ซึ่งในการทดลองนี้ SIP Server จะใช้ IP 192.168.1.1 โดยก่อนที่จะส่ง SIP Request Message ไปยัง SIP Server นั้นจะสังเกตว่าจะมีการใช้โปรโตคอล ARP สอบถามหมายเลข MAC Address ของ SIP Server ก่อนเพื่อใช้ในการส่งข้อมูลไปยังปลายทางในชั้น Datalink Layer หลังจากนั้น SIP Server จะตอบหมายเลข MAC Address ตัวเองกลับไปให้ตัวโทรศัพท์ จากนั้นจะเริ่มทำการส่ง SIP Request Message Register ไปยัง SIP Server โดยผ่านทาง UDP Port 5060 จากนั้น SIP Server จะตอบ SIP Response Message 100 Trying กลับมา ว่ากำลังดำเนินการอยู่ หลังจากนั้นจะส่ง SIP Response Message 401 Unauthorized กลับมา ว่า Account นี้ยังไม่มีที่ยืนยันตัวตนให้ส่ง username , password เพื่อใช้ในการยืนยันตัวตน โดยหลังจากนั้นโทรศัพท์จะส่ง SIP Request Message Register กลับไปอีกครั้งหนึ่งในคราวนี้จะเพิ่มฟิลด์ Authorization เข้าไปใน SIP Header ด้วย ซึ่งจะบรรจุ username , password เข้าไปด้วย หลังจากนั้น SIP Server ก็จะส่ง SIP Response Message 100 Trying บอกว่ากำลังทำการตรวจสอบว่า username , password ถูกต้องหรือไม่ ถ้าถูกต้องก็จะส่ง SIP Response Message 200 OK กลับไปบอกตัวโทรศัพท์ว่าการ Register เสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น หากท่านใดต้องการนำเอกสารนี้ไปใช้โดยไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งท่านมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	Time	Source	Destination	Protocol	Info
311	1677.168365	00:ba:d0:0b:ad:00	Vmware_5a:37:51	ARP	Who has 192.168.1.1? Tell 192.168.1.5
312	1677.168474	Vmware_5a:37:51	00:ba:d0:0b:ad:00	ARP	192.168.1.1 is at 00:0c:29:5a:37:51
313	1705.433040	192.168.1.5	192.168.1.1	SIP	Request: REGISTER sip:192.168.1.1
314	1705.433645	192.168.1.1	192.168.1.5	SIP	Status: 100 Trying (1 bindings)
315	1705.433804	192.168.1.1	192.168.1.5	SIP	Status: 401 Unauthorized (0 bindings)
316	1705.708621	192.168.1.1	192.168.1.5	SIP	Status: 100 Trying (1 bindings)
318	1705.714051	192.168.1.1	192.168.1.5	SIP	Status: 200 OK (1 bindings)

▶ Frame 316 (585 bytes on wire, 585 bytes captured)  
 ▶ Ethernet II, Src: 00:ba:d0:0b:ad:00 (00:ba:d0:0b:ad:00), Dst: Vmware\_5a:37:51 (00:0c:29:5a:37:51)  
 ▶ Internet Protocol, Src: 192.168.1.5 (192.168.1.5), Dst: 192.168.1.1 (192.168.1.1)  
 ▶ User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)  
   Source port: sip (5060)  
   Destination port: sip (5060)  
   Length: 551  
   Checksum: 0x19dc [correct]  
 ▶ Session Initiation Protocol  
   Request-Line: REGISTER sip:192.168.1.1 SIP/2.0  
   Message Header  
     Via: SIP/2.0/UDP 192.168.1.5:5060;rport;branch=z9hG4bKpj75AHpljGV7PNhro7s9lBy5ky2cyQXcs1  
     Max-Forwards: 70  
     From: <sip:5001@192.168.1.1>;tag=P51NsZxGfpqs05zFT.5n7lE-STm1NIE  
     To: <sip:5001@192.168.1.1>  
     Call-ID: RorL9uDANfDnrhkjvYor3J06MBzNEp47  
     CSeq: 60768 REGISTER  
     Contact: "(null)" <sip:5001@192.168.1.5:5060>  
     Expires: 600  
     Authorization: Digest username="5001", realm="asterisk", nonce="7bb9349e", uri="sip:192.168.1.1",  
     Content-Length: 0

รูปที่ 4.4 รายละเอียดของ Header SIP Request Message REGISTER

จะสังเกตว่าในฟิลด์ Authorization ค่า nonce จะเป็น password ที่ถูกเข้ารหัสไว้เพื่อเป็นการรักษาความปลอดภัยในกรณีที่มีการดักจับแพ็กเก็ตข้อมูล โดยจะใช้อัลกอริทึม MD5 ในการเข้ารหัส password

#### 4.1.2 ผลการทดลองในส่วนของการโทรออก

ในการทดลองส่วนนี้จะใช้โทรศัพท์โทรหา SIP Client ที่เป็น Software บนเครื่องคอมพิวเตอร์ โดยในการทดลองนี้จะใช้โปรแกรม X-lite ทำหน้าที่เป็น SIP Client หมายเลข 5003 ซึ่งจะต้องเข้าไป Register ที่ SIP Server ก่อน เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดสอบนี้จะให้ตัวเครื่องโทรศัพท์ซึ่งมีหมายเลข 5001 โทรหาหมายเลข 5003 โดยจะได้ผลการทดลองดังรูปที่ 4.6



รูปที่ 4.5 โปรแกรม X-lite ที่ใช้ในการทดสอบ

NetworkMinion WinPCAP Networking Adapter Driver (Microsoft's Packet Scheduler) - Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
5	3.887474	192.168.1.5	192.168.1.1	SIP/SDP	Request: INVITE sip:5003@192.168.1.1, with session description
6	3.888167	192.168.1.1	192.168.1.5	SIP	Status: 407 Proxy Authentication Required
7	4.025142	192.168.1.5	192.168.1.1	SIP	Request: ACK sip:5003@192.168.1.1
8	4.294397	192.168.1.5	192.168.1.1	SIP/SDP	Request: INVITE sip:5003@192.168.1.1, with session description
9	4.295161	192.168.1.1	192.168.1.5	SIP	Status: 100 Trying
10	4.296349	192.168.1.1	192.168.1.2	SIP/SDP	Request: INVITE sip:5003@192.168.1.2:38480;rinstance=a69ddf986ef
11	4.400235	192.168.1.2	192.168.1.1	SIP	Status: 180 Ringing
12	4.401055	192.168.1.1	192.168.1.5	SIP	Status: 180 Ringing

รูปที่ 4.6 แพ็กเก็ตในขณะที่มีหมายเลข 5001 โทรหาหมายเลข 5003

จากรูปที่ 4.6 เมื่อเราป้อนหมายเลข 5003 ให้กับตัวโทรศัพท์เรียบร้อยแล้วหลังจากนั้นให้กดปุ่ม # เพื่อให้ไมโครคอนโทรลเลอร์ส่งหมายเลขไปให้โปรแกรมที่อยู่บนบอร์ด JampII mini หลังจากโปรแกรมได้รับหมายเลขเรียบร้อยแล้วจะเรียกโปรแกรมย่อยที่ทำการส่ง INVITE Message ออกไป โดยจะส่งไปยัง SIP Server ซึ่งใน Message นี้จะส่ง SDP ไปด้วยโดยเพื่อระบุพอร์ตที่ใช้ในการ transport ของโปรโตคอล RTP และ RTCP และระบุ codec ที่สามารถใช้งานได้ หลังจากนั้น SIP Server จะส่ง SIP Response Message 407 Proxy Authentication Required เพื่อบอกให้เครื่องโทรศัพท์นั้นส่งข้อมูลที่ใช้ในการยืนยัน

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยทางบริษัท Register เมื่อโทรศัพท์ที่ได้รับ Message ก็จะมี username, password เหมือนครวที่ Register เมื่อโทรศัพท์ได้รับ Message ก็จะมี username, password เหมือนครวที่ Register เมื่อโทรศัพท์ได้รับ Message ก็จะมี username, password เหมือนครวที่ Register

กลับไปบอกว่าได้รับแล้ว หลังจากนั้นก็จะส่ง INVITE Message กลับมาใหม่ โดยจะเพิ่มฟิลด์ Proxy – Authorization มาด้วย หลังจาก SIP Server ได้รับ Message INVITE แล้วก็จะทำการตรวจสอบว่าหมายเลข 5001 ที่เคยลงทะเบียนให้กับ SIP Server หรือไม่ เพื่อเป็นการป้องกันการปลอมแปลงเข้ามาใช้ระบบ หลังจากนั้นจะส่ง SIP Response Message 100 Trying กลับไปบอกตัวเครื่องโทรศัพท์ที่กำลังทำการติดต่อ หมายเลขปลายทางให้อยู่ในขณะเดียวกัน SIP Server ก็จะส่ง Message INVITE ไปยังหมายเลข 5003 หลังจากหมายเลข 5003 ได้รับ Message INVITE แล้วก็จะส่ง Response Message 180 Ringing กลับไปบอก SIP Server ว่ากำลังรอการรับสายอยู่ ซึ่ง SIP Server ก็จะส่งไปบอกตัวโทรศัพท์ว่าหมายเลข 5003 กำลังรอการรับสายอยู่

No.	Time	Source	Destination	Protocol	Info
5	3.887474	192.168.1.5	192.168.1.1	SIP/SDP	Request: INVITE sip:5003@192.168.1.1, with session description
6	3.888167	192.168.1.1	192.168.1.5	SIP	Status: 407 Proxy Authentication Required
7	4.025142	192.168.1.5	192.168.1.1	SIP	Request: ACK sip:5003@192.168.1.1
8	4.295181	192.168.1.5	192.168.1.1	SIP/SDP	Request: INVITE sip:5003@192.168.1.1, with session description
9	4.295181	192.168.1.1	192.168.1.5	SIP	Status: 100 Trying
10	4.296349	192.168.1.1	192.168.1.2	SIP/SDP	Request: INVITE sip:5003@192.168.1.2:38480;rinstance=a69ddf986et
11	4.400235	192.168.1.2	192.168.1.1	SIP	Status: 180 Ringing
12	4.401055	192.168.1.1	192.168.1.5	SIP	Status: 180 Ringing

▷ Frame 8 (1042 bytes on wire, 1042 bytes captured)  
 ▷ Ethernet II, Src: 00:ba:d0:0b:ad:00 (00:ba:d0:0b:ad:00), Dst: vmware\_5a:37:51 (00:0c:29:5a:37:51)  
 ▷ Internet Protocol, Src: 192.168.1.5 (192.168.1.5), Dst: 192.168.1.1 (192.168.1.1)  
 ▲ User Datagram Protocol, Src Port: sip (5060), Dst Port: sip (5060)  
   Source port: sip (5060)  
   Destination port: sip (5060)  
   Length: 1008  
   Checksum: 0x0797 [correct]  
 ▲ Session Initiation Protocol  
   Request-Line: INVITE sip:5003@192.168.1.1 SIP/2.0  
   Message Header  
     Via: SIP/2.0/UDP 192.168.1.5:5060;rport;branch=z9hg4bKpJjHR1Kkt8qIJCwIvAXpFe7sCep3j3swo98  
       Max-Forwards: 70  
     From: sip:5001@192.168.1.1;tag=3JeqBfsrjIHgERV1M92h6eqcRnHCKQYGd  
     To: sip:5003@192.168.1.1  
     Contact: "(null)" <sip:5001@192.168.1.5:5060>  
     Call-ID: NrGHai.teszi-55hatIikueWFPetF0pf  
     CSeq: 15501 INVITE  
     Allow: PRACK, INVITE, ACK, BYE, CANCEL, UPDATE, SUBSCRIBE, NOTIFY, REFER, MESSAGE, OPTIONS  
     Supported: replaces, 100rel, norefersub  
   Proxy-Authorization: Digest username="5001", realm="asterisk", nonce="0cc589bd", uri="sip:5003@192.168.1.1"  
   Content-Type: application/sdp  
   Content-Length: 296  
 ▲ Message Body  
   Session Description Protocol  
   Message Body  
     Session Description Protocol  
       Session Description Protocol version (v): 0  
       Owner/Creator, Session Id (o): - 2208997103 2208997103 IN IP4 192.168.1.5  
       Session Name (s): pjmedia  
       Connection Information (c): IN IP4 192.168.1.5  
       Time Description, active time (t): 0 0  
       Session Attribute (a): X-nat:0  
       Media Description, name and address (m): audio 4002 RTP/AVP 0 8 9 101  
       Media Attribute (a): rtcp:4003 IN IP4 192.168.1.5  
       Media Attribute (a): rtpmap:0 PCMU/8000  
       Media Attribute (a): rtpmap:8 PCMA/8000  
       Media Attribute (a): rtpmap:9 G722/8000  
       Media Attribute (a): sendrecv  
       Media Attribute (a): rtpmap:101 telephone-event/8000  
       Media Attribute (a): fmp:101 0-15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต  
รูปที่ 4.7 รายละเอียดของ Header SIP Request Message INVITE โยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 ใน SDP Message จะบอกถึงหมายเลขพอร์ตของ RTP ซึ่งจะใช้พอร์ต 4002 ส่วน RTCP จะใช้พอร์ต 4003 ในการสื่อสาร และ จะบอก codec ที่เครื่องโทรศัพท์สามารถรองรับได้ โดยจะรองรับ PCMU , PCMA , G722 Sampling Rate 8000 Hz หลังจากนั้นเมื่อทางหมายเลข 5003 รับสายก็จะส่ง Message 200 OK ไปบอกกับทาง SIP Server ว่าพร้อมที่จะสนทนาแล้ว ตัว SIP Server ก็จะส่งไปบอกกับตัวโทรศัพท์ว่าปลายทางพร้อมที่จะสนทนาด้วยแล้ว หลังจากนั้นจะเป็นหน้าที่ของโปรโตคอล RTP โดยจะนำส่งแพ็กเก็ตเสียงที่ได้ทำการเข้ารหัสเสียงไว้แล้วส่งไปยังปลายทาง โดยขั้นตอนนี้จะสื่อสารกันโดยตรง ไม่ผ่านทาง SIP Server แล้ว

No.	Time	Source	Destination	Protocol	Info
211	22.375119	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4874, Time=115460
212	22.394800	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4875, Time=115620
213	22.425073	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4876, Time=115780
214	22.444651	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4877, Time=115940
215	22.465032	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4878, Time=116100
216	22.469334	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10029, Time=13600
217	22.469802	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10030, Time=13760
218	22.471377	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10031, Time=13920
219	22.471862	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10032, Time=14080
220	22.472728	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10033, Time=14240
221	22.473181	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10034, Time=14400
222	22.474679	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4879, Time=116260
223	22.495193	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4880, Time=116420
224	22.514725	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4881, Time=116580
225	22.535296	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4882, Time=116740
226	22.554867	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4883, Time=116900
227	22.574293	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4884, Time=117060
228	22.594961	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x4855080F, Seq=4885, Time=117220
229	22.597344	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10035, Time=14560
230	22.597817	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10036, Time=14720
231	22.598717	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10037, Time=14880
232	22.599169	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10038, Time=15040
233	22.600696	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10039, Time=15200
234	22.602088	192.168.1.5	192.168.1.2	RTP	PT=ITU-T G.711 PCMU, SSRC=0x57D1B709, Seq=10040, Time=15360

รูปที่ 4.8 แพ็กเก็ตในระหว่างทำการสนทนา

จากรูปที่ 4.8 จะสังเกตได้ว่า ค่า SSRC จะมีค่าเท่ากันเมื่อมาจากผู้ส่งเดียวกัน และค่า Time หรือค่า Time Stamp นั้นจะมีค่าเพิ่มขึ้นไปเรื่อยๆ เช่นเดียวกับ Seq หรือ Sequence Number ทั้งนี้เพื่อจะใช้ในการตรวจสอบการสูญหายของแพ็กเก็ตและเพื่อให้ทางฝั่งรับสามารถนำข้อมูลไปแสดงผลได้อย่างถูกต้อง

#### 4.1.3 ผลการทดลองในส่วนของการรับสาย

ในการทดลองส่วนนี้จะเหมือนกับในหัวข้อ 4.1.2 แต่จะเปลี่ยนจาก หมายเลข 5003 โทรหมายเลข 5001 แทน และการรับสายจากตัวโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโรงเรียนอาชีวศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NVIDIA GeForce 1060 Networking Adapter Driver (Microsoft's Packet Scheduler) Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
6	18.622370	192.168.1.1	192.168.1.2	SIP	Status: 407 Proxy Authentication Required
7	18.646195	192.168.1.2	192.168.1.1	SIP	Request: ACK sip:5001@192.168.1.1
8	18.648188	192.168.1.2	192.168.1.1	SIP/SDP	Request: INVITE sip:5001@192.168.1.1, with session description
9	18.668068	192.168.1.1	192.168.1.2	SIP	Status: 100 Trying
10	18.691597	192.168.1.1	192.168.1.5	SIP/SDP	Request: INVITE sip:5001@192.168.1.5:5060, with session description
11	18.835453	192.168.1.5	192.168.1.1	SIP	Status: 100 Trying
12	18.971561	192.168.1.5	192.168.1.1	SIP	Status: 180 Ringing
13	18.976694	192.168.1.1	192.168.1.2	SIP	Status: 180 Ringing

### รูปที่ 4.9 แพ็กเก็ตในขณะที่ยหมายเลข 5003 โทรเข้ามา

จากรูปที่ 4.9 จะสังเกตเห็นว่าลำดับสัญญาณ SIP จะคล้ายกับการโทรออกแต่จะมีการส่ง SIP Response 100 Trying กลับไปบอก SIP Server แล้วจึงค่อยส่ง SIP Response Message 180 Ringing ไป ซึ่งจะแตกต่างกับโปรแกรม X-lite โดยเมื่อได้รับ SIP Request INVITE แล้วจะส่ง SIP Response Message 180 Ringing ออกไปทันที โดยที่รายละเอียดของ Header จะแสดงดังรูปที่ 4.10

No.	Time	Source	Destination	Protocol	Info
5	18.622370	192.168.1.2	192.168.1.1	SIP/SDP	Request: INVITE sip:5001@192.168.1.1, with session description
6	18.622370	192.168.1.1	192.168.1.2	SIP	Status: 407 Proxy Authentication Required
7	18.646195	192.168.1.2	192.168.1.1	SIP	Request: ACK sip:5001@192.168.1.1
8	18.648188	192.168.1.2	192.168.1.1	SIP/SDP	Request: INVITE sip:5001@192.168.1.1, with session description
9	18.668068	192.168.1.1	192.168.1.2	SIP	Status: 100 Trying
10	18.691597	192.168.1.1	192.168.1.5	SIP/SDP	Request: INVITE sip:5001@192.168.1.5:5060, with session description
11	18.835453	192.168.1.5	192.168.1.1	SIP	Status: 100 Trying
12	18.971561	192.168.1.5	192.168.1.1	SIP	Status: 180 Ringing
13	18.976694	192.168.1.1	192.168.1.2	SIP	Status: 180 Ringing

Frame 5 (1215 bytes on wire (1215 bytes captured))

Ethernet II, Src: Asiarock\_b1:e1:db (00:13:8f:b1:e1:db), Dst: Vmware\_5a:37:51 (00:0c:29:5a:37:51)

Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.1 (192.168.1.1)

User Datagram Protocol, Src Port: 38480 (38480), Dst Port: sip (5060)

Source port: 38480 (38480)  
Destination port: sip (5060)  
Length: 1181  
Checksum: 0x561b [correct]

Session Initiation Protocol

Request-Line: INVITE sip:5001@192.168.1.1 SIP/2.0

Message Header

Via: SIP/2.0/UDP 192.168.1.2:38480;branch=z9hG4bK-d87543-4928e45e534ce850-1--d87543-;rport  
Max-Forwards: 70  
Contact: <sip:5003@192.168.1.2:38480>  
To: "5001" <sip:5001@192.168.1.1>  
From: "5003" <sip:5003@192.168.1.1>;tag=9b6b6871  
Call-ID: NGE30WNfNTljMGUyZGEyNTU2YWIzZjM3MmFmZiZiNDI.  
CSeq: 1 INVITE  
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO  
Content-Type: application/sdp  
User-Agent: X-Lite release 1011s stamp 41150  
Content-Length: 650

Message Body

Session Description Protocol

Session Description Protocol Version (V): 0  
Owner/Creator, Session Id (o): - 5 2 IN IP4 192.168.1.2  
Session Name (s): CounterPath X-Lite 3.0  
Connection Information (c): IN IP4 192.168.1.2  
Time Description, active time (t): 0 0  
Media Description, name and address (m): audio 30366 RTP/AVP 107 119 100 106 0 97 105 98 8 101  
Media Attribute (a): alt:1 5 : ypyrggts ZA5eiSa0 169.254.248.252 30366  
Media Attribute (a): alt:2 4 : m7Uz61Qs CpoQSIW 192.168.1.2 30366  
Media Attribute (a): alt:3 3 : HvZnhFdp SketYHhH 124.120.51.95 30366  
Media Attribute (a): alt:4 2 : COpUtn+ sJDeL4FK 192.168.111.1 30366  
Media Attribute (a): alt:5 1 : yTVKsQSL Y+uxi9nk 192.168.244.1 30366  
Media Attribute (a): fmtp:101 0-15  
Media Attribute (a): rtpmap:107 BV32/16000  
Media Attribute (a): rtpmap:119 BV32-FEC/16000  
Media Attribute (a): rtpmap:100 SPEEX/16000  
Media Attribute (a): rtpmap:106 SPEEX-FEC/16000  
Media Attribute (a): rtpmap:97 SPEEX/8000  
Media Attribute (a): rtpmap:105 SPEEX-FEC/8000  
Media Attribute (a): rtpmap:98 1LBC/8000  
Media Attribute (a): rtpmap:101 telephone-event/8000  
Media Attribute (a): sendrecv

เอกสารนี้เป็นเอกสารรูปที่ 4.10 รายละเอียด Header SIP Request Message INVITE ที่ส่งไปยัง SIP Server ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.10 จะสังเกตในส่วนของ SDP Message ตัวโปรแกรม X-lite นั้นสามารถรองรับ codec ได้หลายตัว โดยจะเรียง priority ที่ฟิลด์ Media Description, name address (m) โดยจะเรียงจาก codec หมายเลข 107 119 100 106 0 97 105 98 8 101 ซึ่งหมายเลขพวกนี้จะอ้างอิง codec ที่รองรับกับโปรโตคอล RTP ส่วนในการ transport นั้นโปรแกรม X-lite จะใช้พอร์ต 30366 ในการส่งซึ่งจะใช้การสุ่มขึ้นมา

ในขณะที่ตัวเครื่องโทรศัพท์นั้นได้รับ SIP INVITE Message มาจาก SIP Server เข้ามาก็จะตอบ SIP Response Message 100 Trying กลับไปบอก SIP Server ว่าได้รับ Message แล้วกำลังทำงานอยู่ หลังจากนั้นก็จะส่ง SIP Response Message 180 Ringing กลับไปบอก SIP Server ว่ากำลังรอการรับสาย หลังจากนั้นเมื่อมีคนรับสายก็จะส่ง SIP Response Message 200 OK กลับไปบอกว่าพร้อมที่จะทำการสนทนา จากนั้นจะเป็นหน้าที่ของโปรโตคอล RTP จะนำส่งแพ็คเกจเสียงที่ได้ทำการเข้ารหัสเสียงโดย codec ที่มีการตกลงกันไว้แล้วส่งไปยังปลายทาง ผ่านทางพอร์ตที่ได้ตกลงกันไว้ในโปรโตคอล SDP โดยขั้นตอนนี้จะสื่อสารกันโดยตรง ไม่ผ่านทาง SIP Server

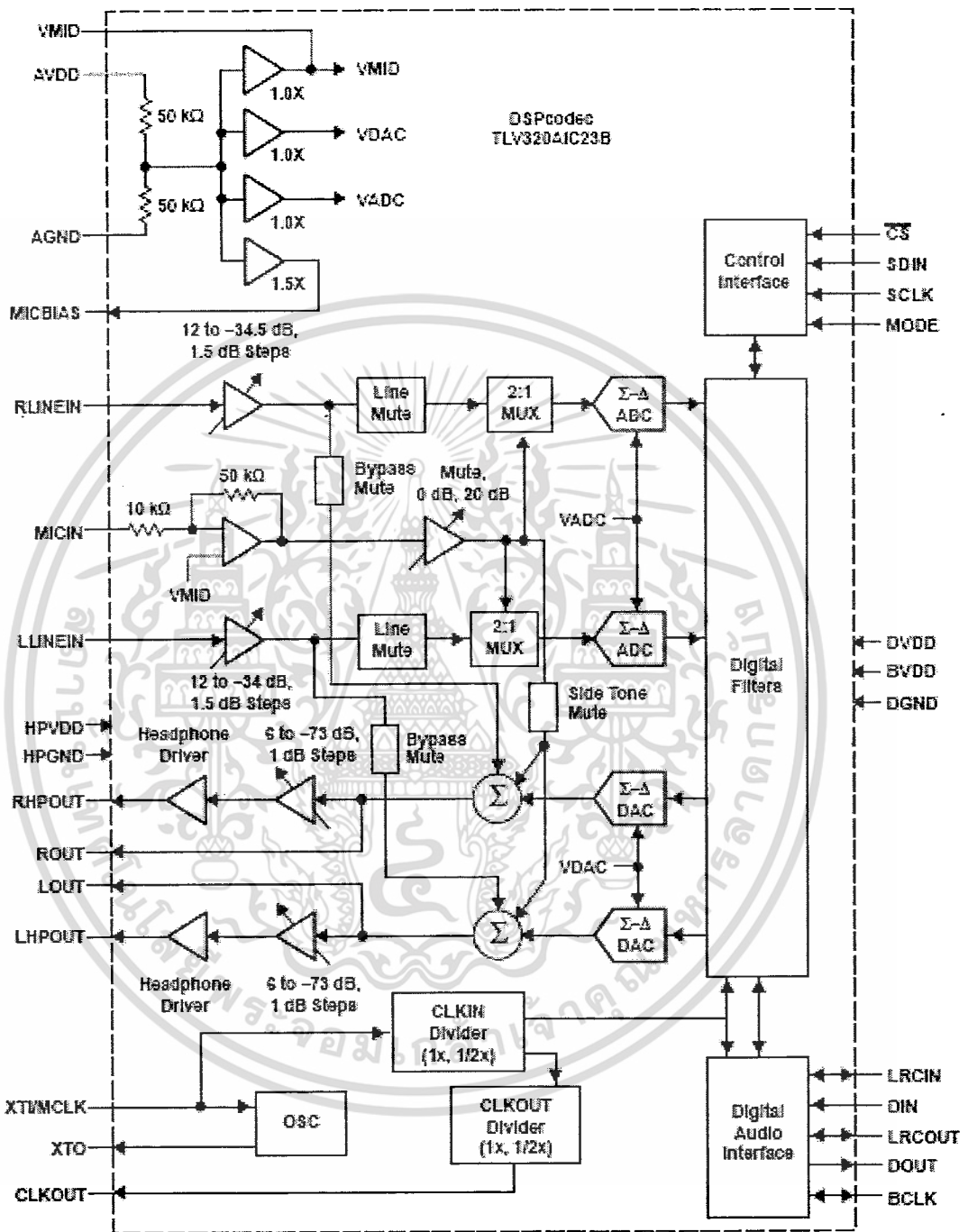
No.	Time	Source	Destination	Protocol	Info
251	137.309557	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8095, Time=2534120
252	137.339233	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8096, Time=2534280
253	137.349437	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8097, Time=2534440
254	137.378961	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8098, Time=2534600
255	137.389513	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8099, Time=2534760
256	137.419299	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8100, Time=2534920
257	137.439199	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8101, Time=2535080
258	137.470306	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8102, Time=2535240
259	137.479486	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8103, Time=2535400
260	137.510238	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8104, Time=2535560
261	137.519468	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8105, Time=2535720
262	137.529227	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8106, Time=2535880
263	137.559470	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8107, Time=2536040
264	137.579163	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8108, Time=2536200
265	137.599502	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8109, Time=2536360
266	137.609291	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8110, Time=2536520
267	137.639720	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8111, Time=2536680
268	137.649415	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8112, Time=2536840
269	137.680045	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8113, Time=2537000
270	137.689411	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8114, Time=2537160
271	137.720246	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8115, Time=2537320
272	137.729453	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8116, Time=2537480
273	137.759812	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8117, Time=2537640
274	137.769431	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8118, Time=2537800
275	137.799136	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8119, Time=2537960
276	137.809526	192.168.1.2	192.168.1.5	RTP	PT=ITU-T G.711 PCMU, SSRC=0x1A9764EB, Seq=8120, Time=2538120

รูปที่ 4.11 แพ็กเก็ตในระหว่างการสนทนา

#### 4.1.4 ผลการทดลองในส่วนของ Audio

ในการรับ-ส่งสัญญาณเสียงผ่านบอร์ด JampII mini นั้นจะผ่านทางไอซี TLV320AIC23B ของบริษัท Texas Instrument ซึ่งจะทำหน้าที่เป็น ADC และ DAC ในตัว รวมทั้งมี Headphone Amplifier ในตัวซึ่งสามารถต่อหูฟังขนาด 32 โอห์มได้โดยเราสามารถควบคุม Volume Gain ผ่านทาง Driver ซึ่งในเอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

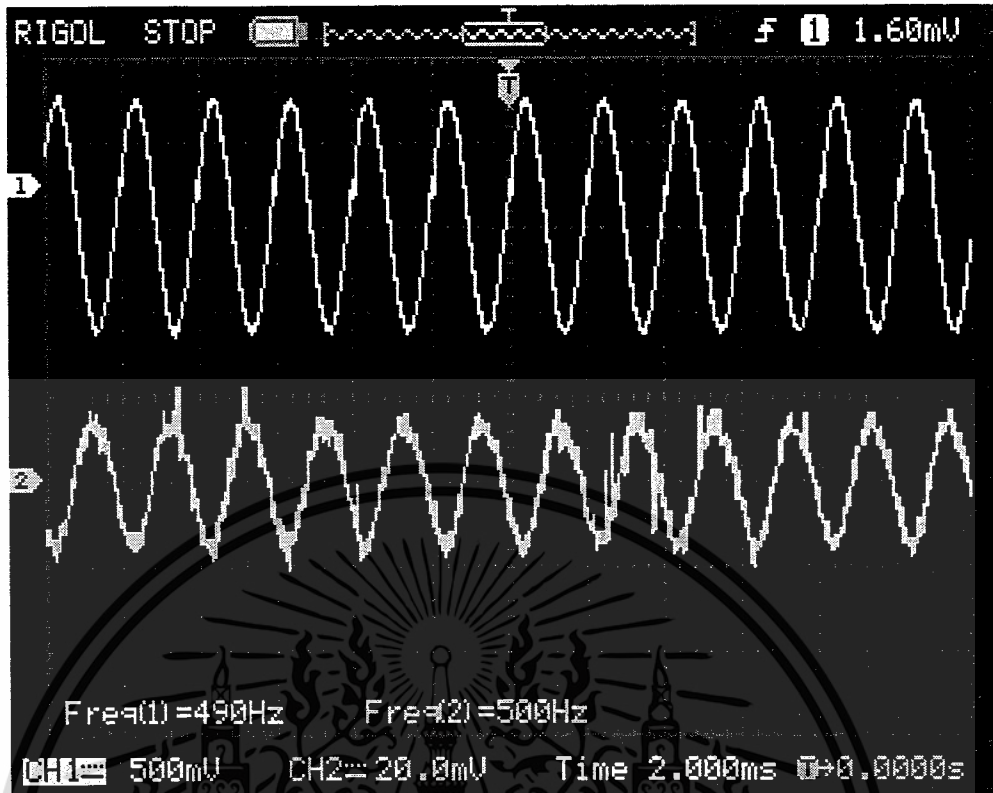
บอร์ดJumpII mini จะใช้ ALSA Driver เป็นตัวควบคุมการทำงานของเสียง



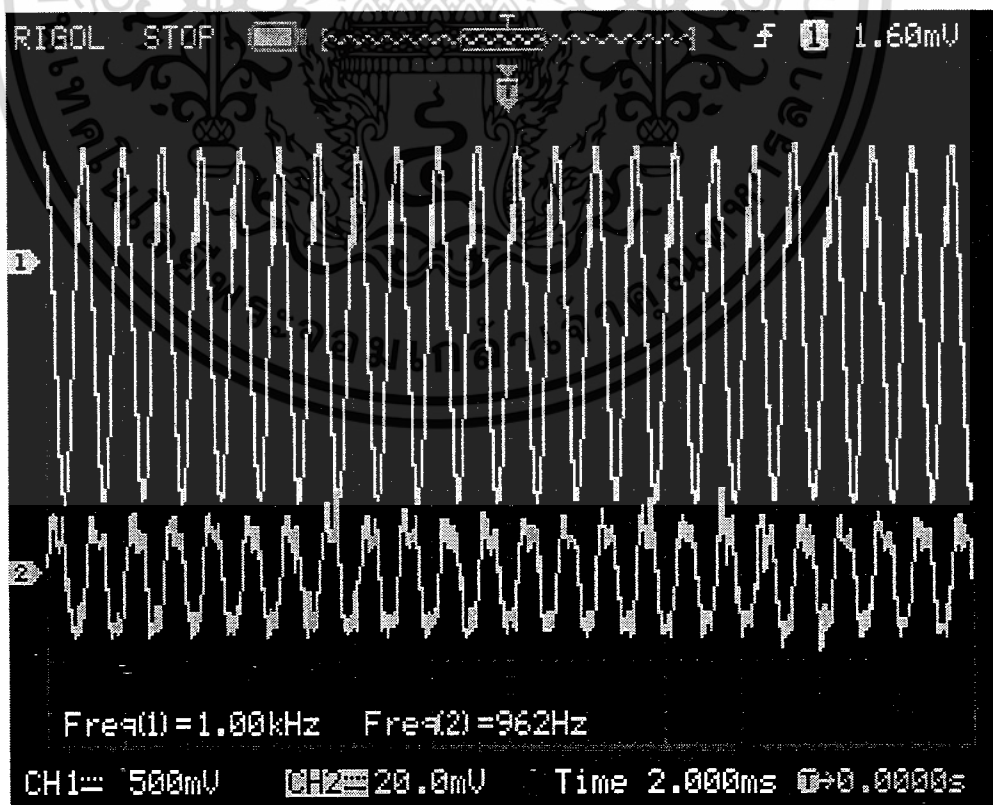
NOTE: MCLK, SCLK, and SCLK are all asynchronous to each other.

รูปที่ 4.12 บล็อกไดอะแกรมการทำงานของ TLV320AIC23B

ในการทดสอบการทำงานของ TLV320AIC23B นั้นเราจะใช้เสียงความถี่ต่างๆ ป้อนให้กับไมโครโฟน เอกสารนี้เ็นในขณะที่ทำการสนทนา แล้วทำการวัดผลลัพธ์ที่ Speaker นั้นๆ ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



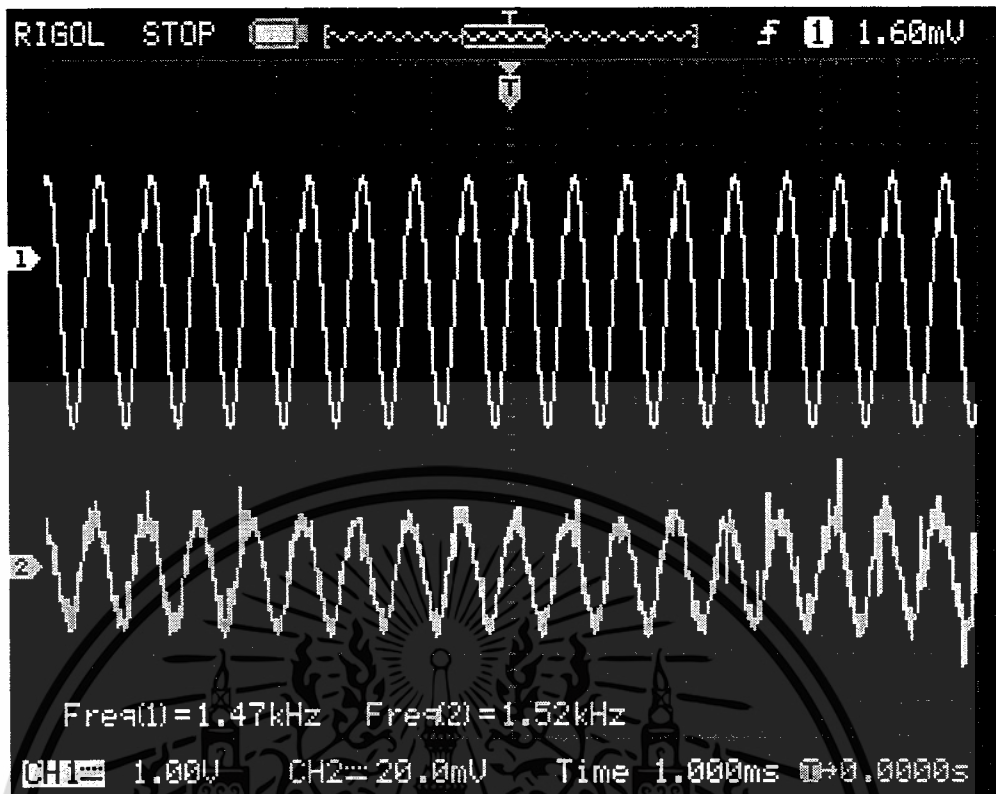
รูปที่ 4.13 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 500 Hz  
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง



รูปที่ 4.14 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 1000 Hz

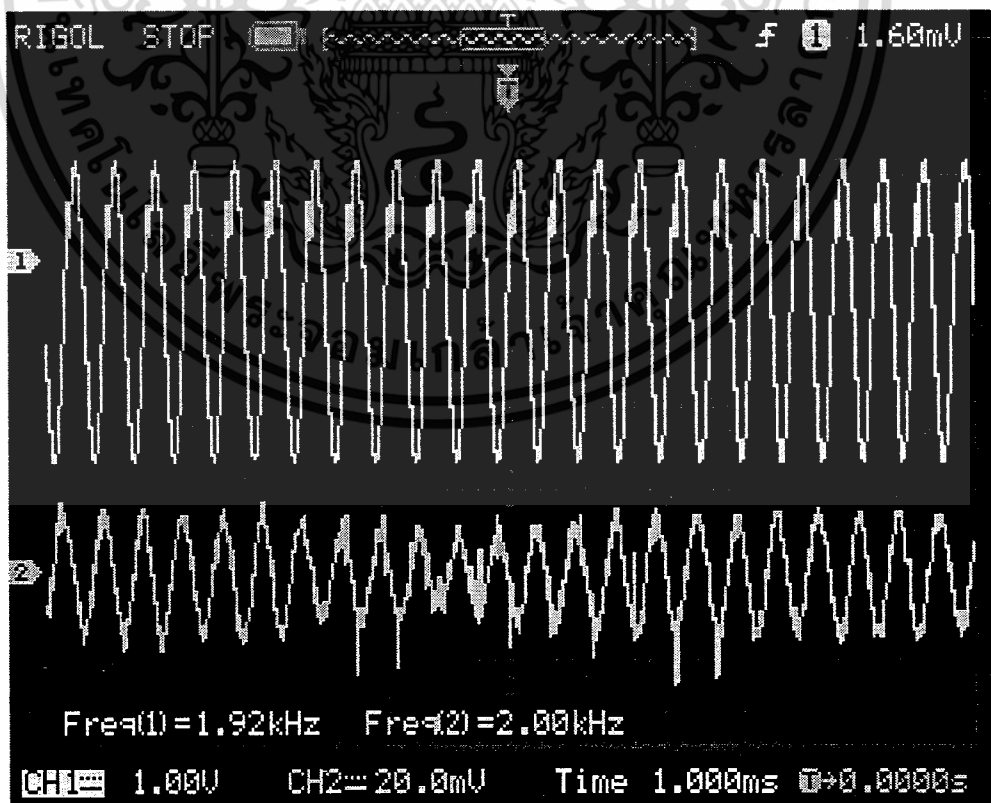
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปเผยแพร่โดยไม่ได้รับอนุญาต  
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 1500 Hz

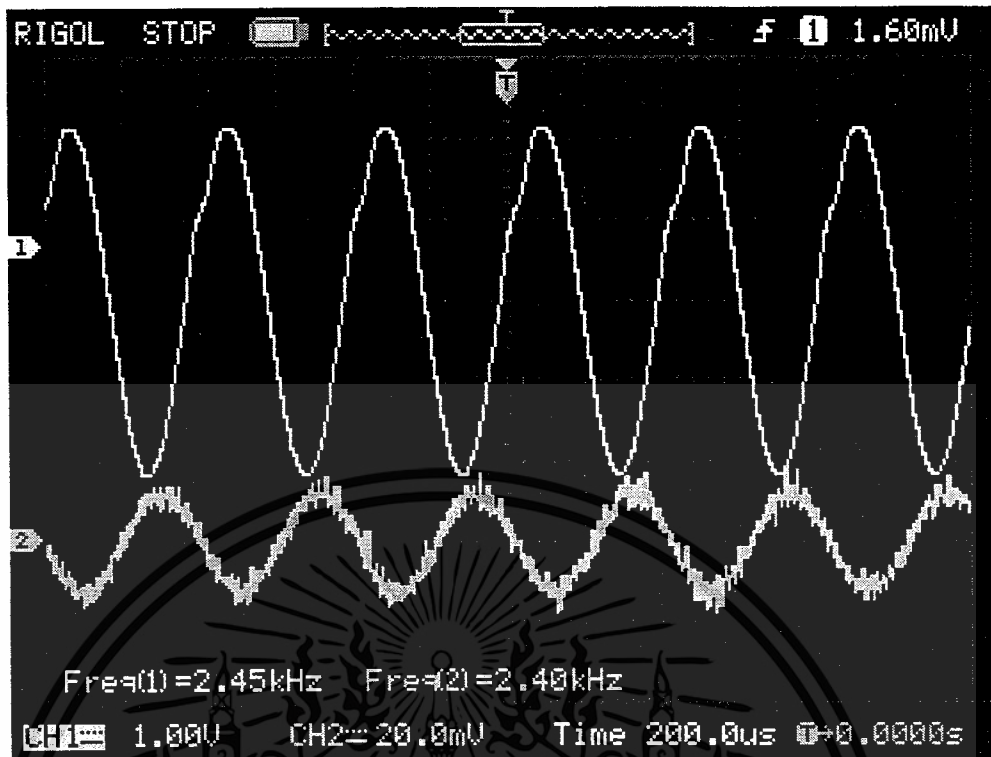
CH1: สัญญาณเสียงจากต้นทาง, CH: 2 สัญญาณเสียงจากปลายทาง



รูปที่ 4.16 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 2000 Hz

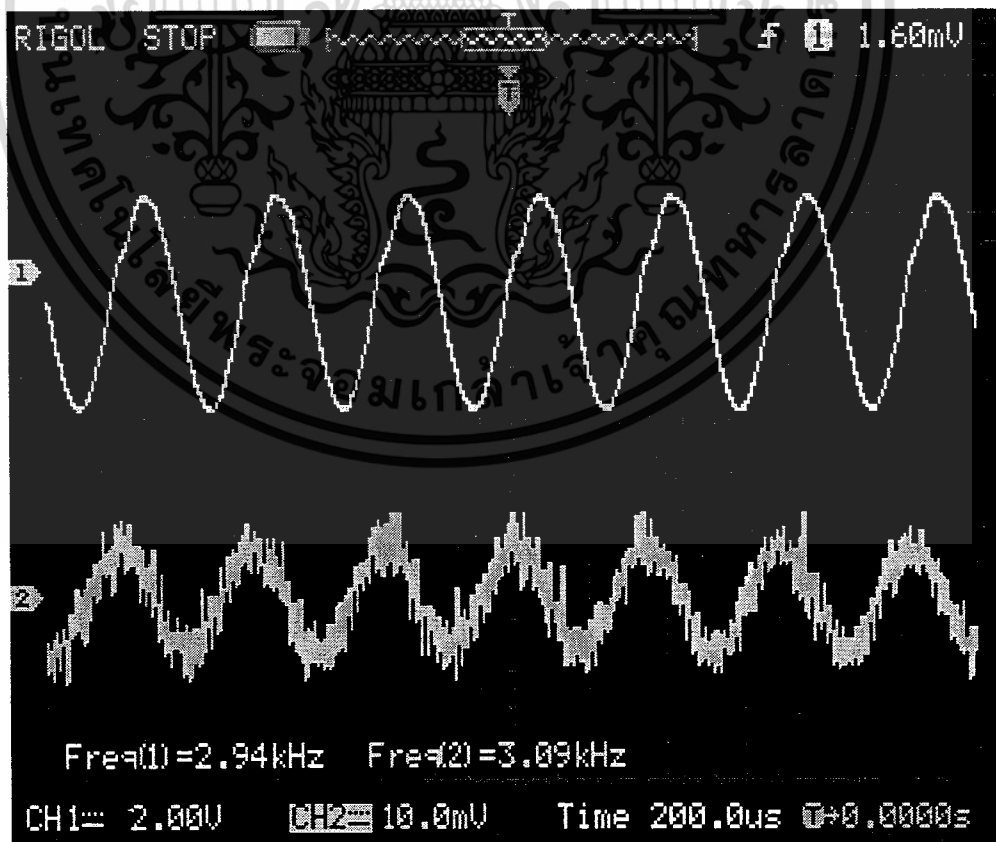
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเฉพาะที่คณะวิศวกรรมศาสตร์เท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 2500 Hz

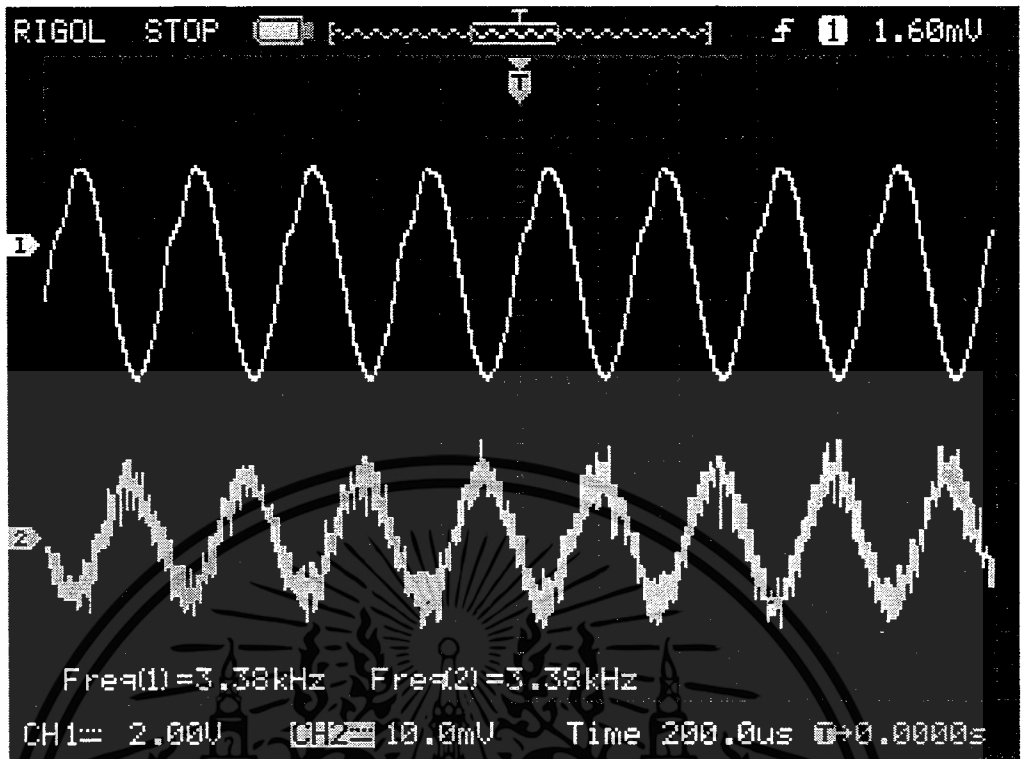
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง



รูปที่ 4.18 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 3000 Hz

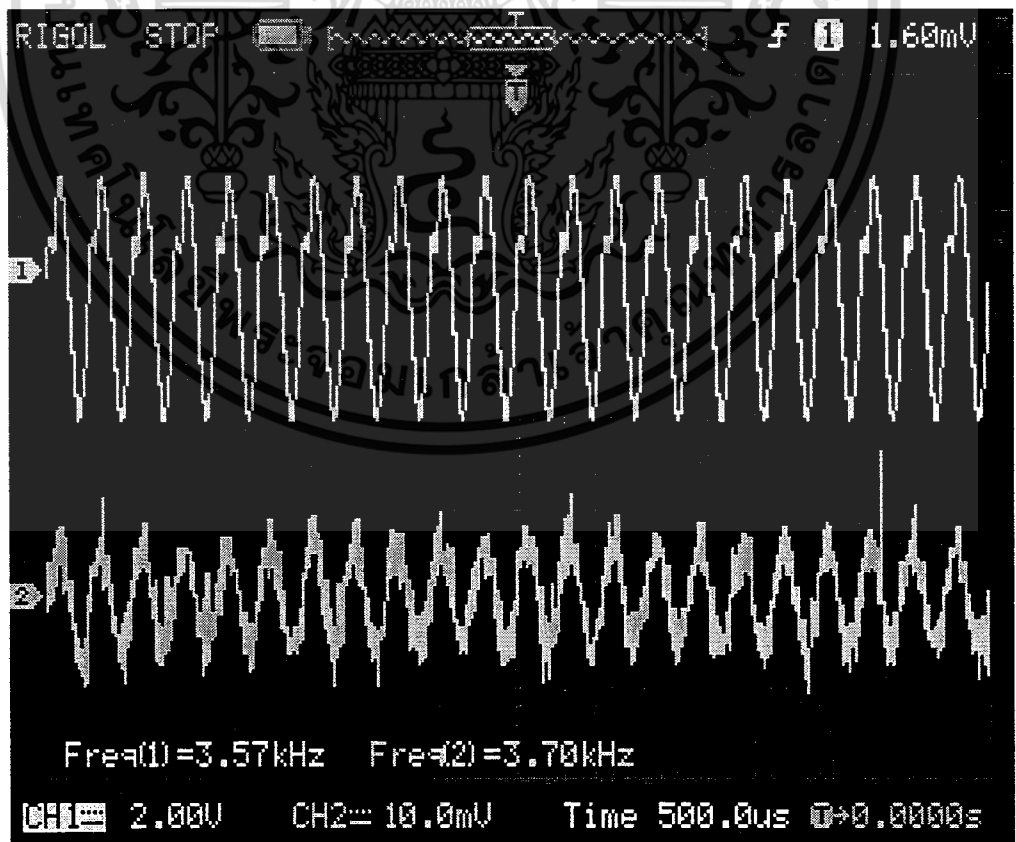
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้ประโยชน์ภายในหน่วยงานราชการ การนำออกเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 3500 Hz

CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

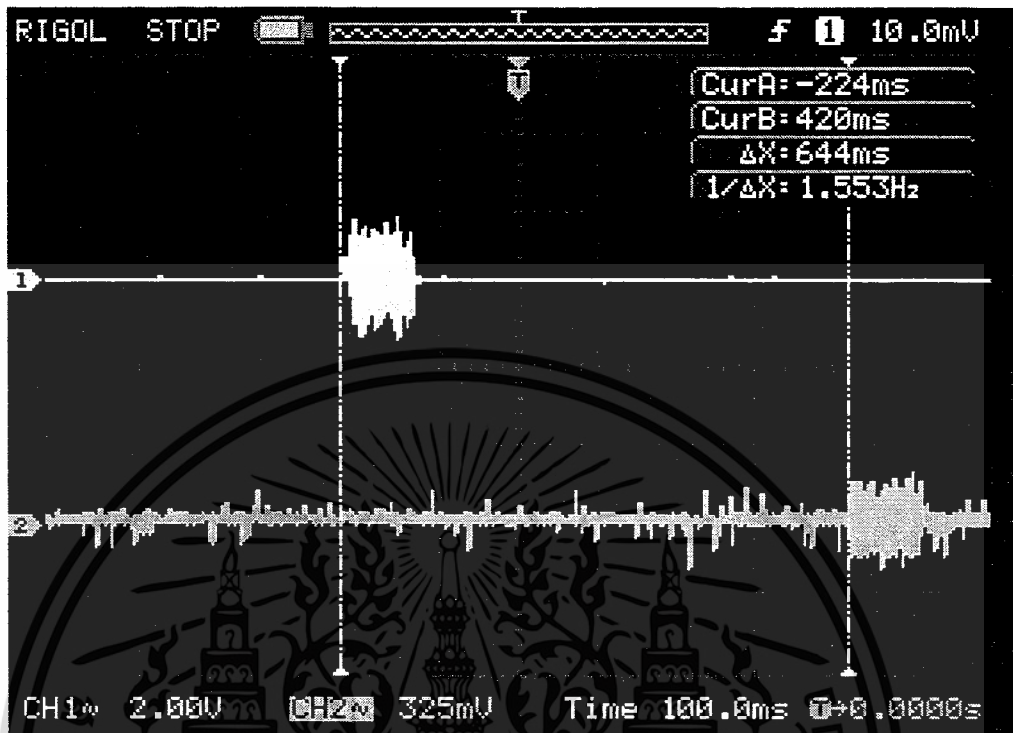


รูปที่ 4.20 เปรียบเทียบสัญญาณต้นทางกับปลายทางที่ความถี่ 4000 Hz

CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

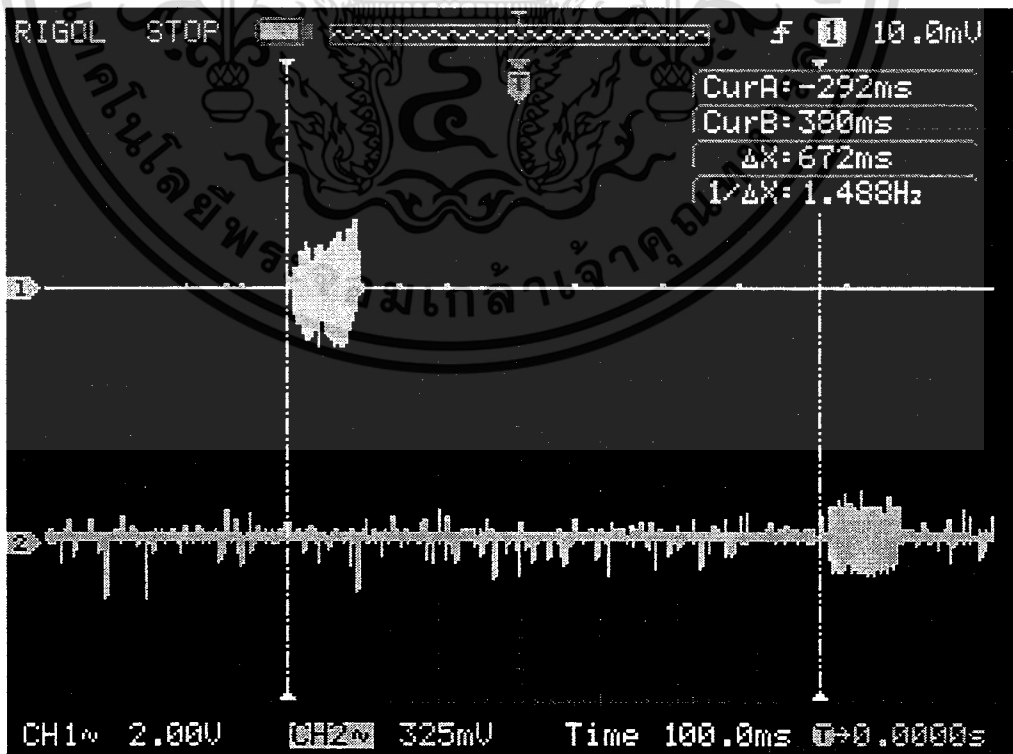
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.5 ผลการทดสอบความล่าช้า (Delay) ของระบบ



รูปที่ 4.21 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 1

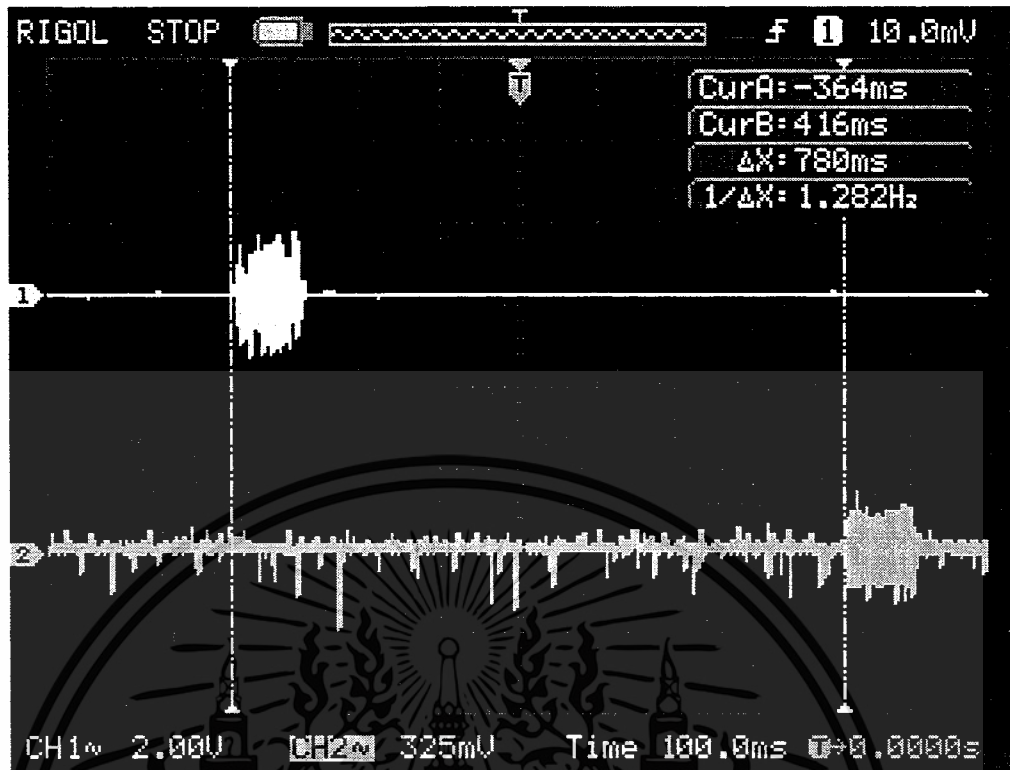
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง



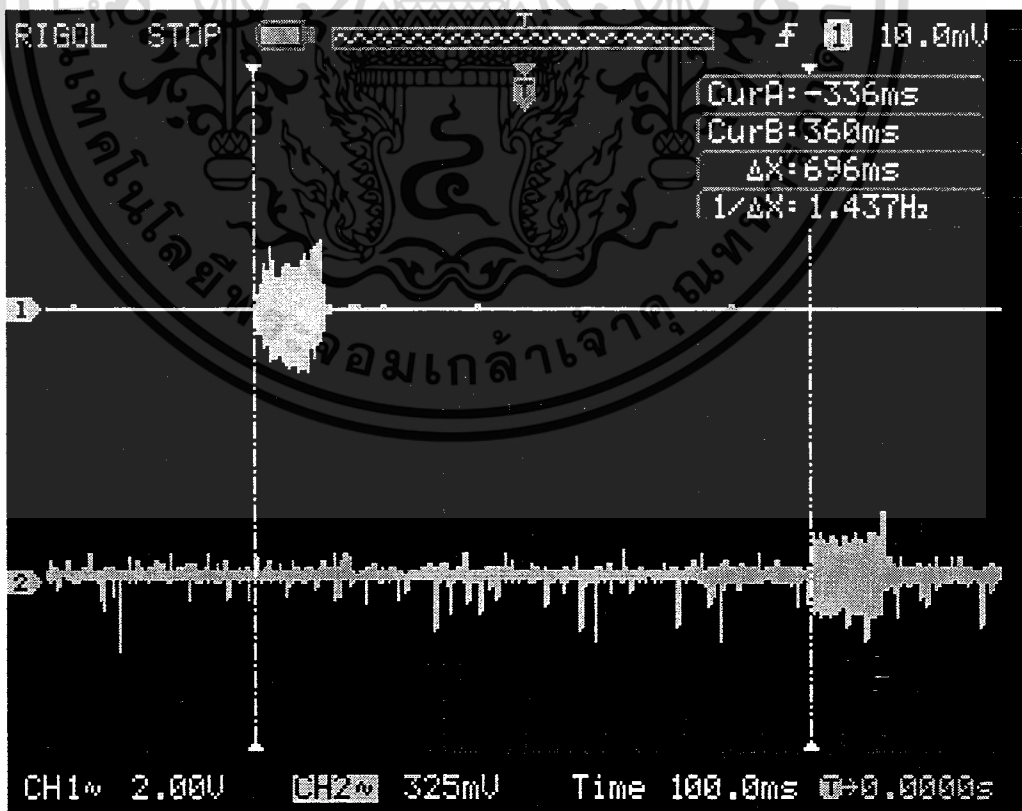
รูปที่ 4.22 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 2

CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

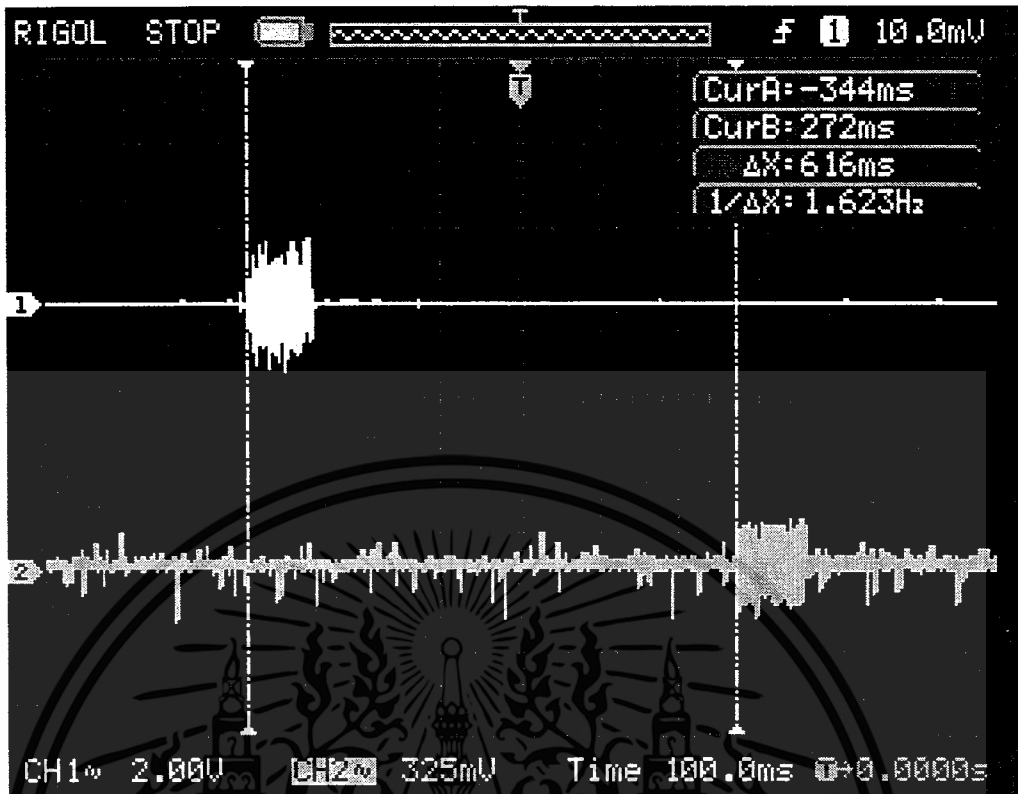


รูปที่ 4.23 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 3  
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

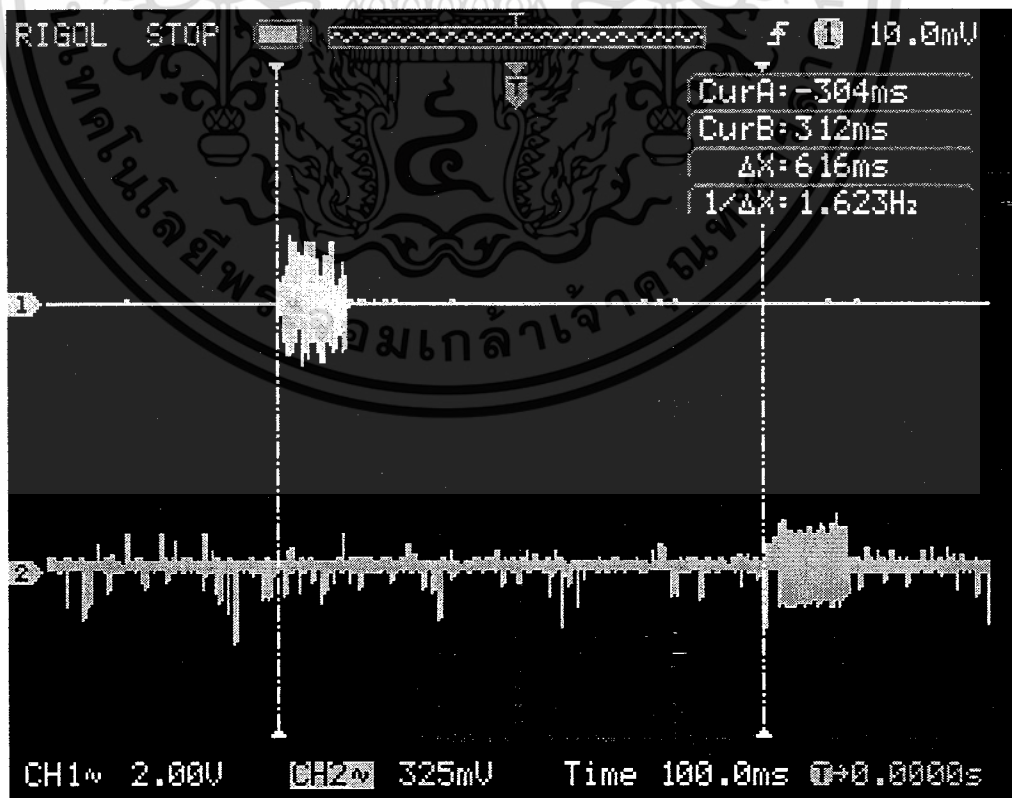


รูปที่ 4.24 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้ CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทางขอชมเกล้าเจ้าคุณหญิง  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

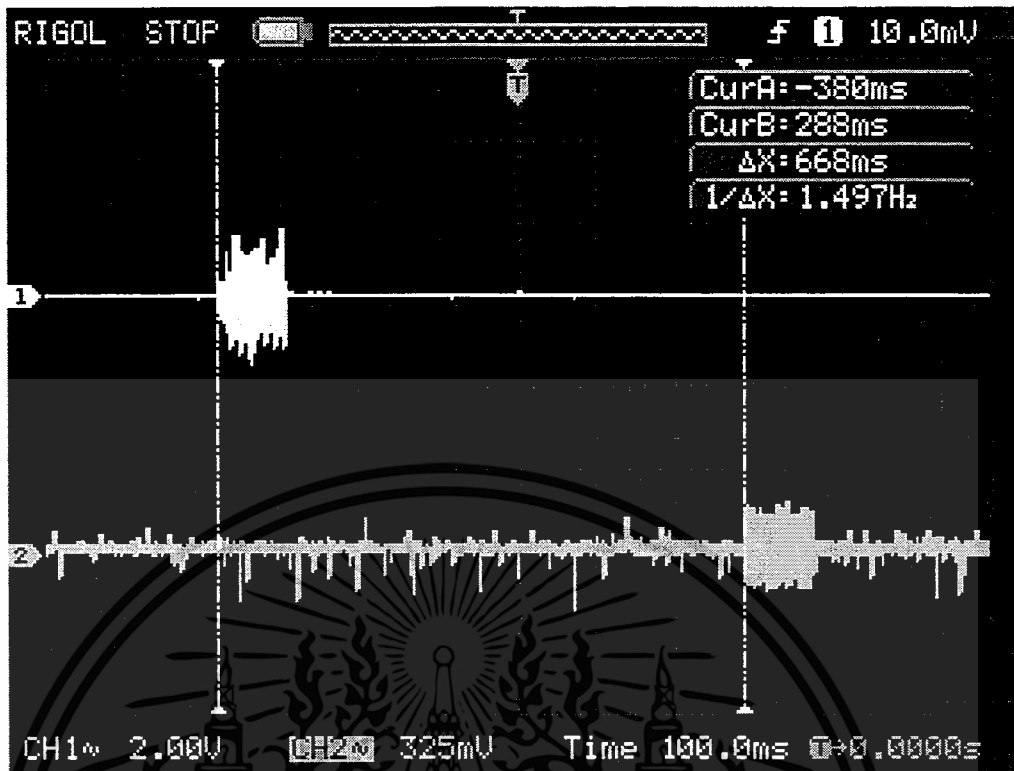


รูปที่ 4.25 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 5  
 CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง



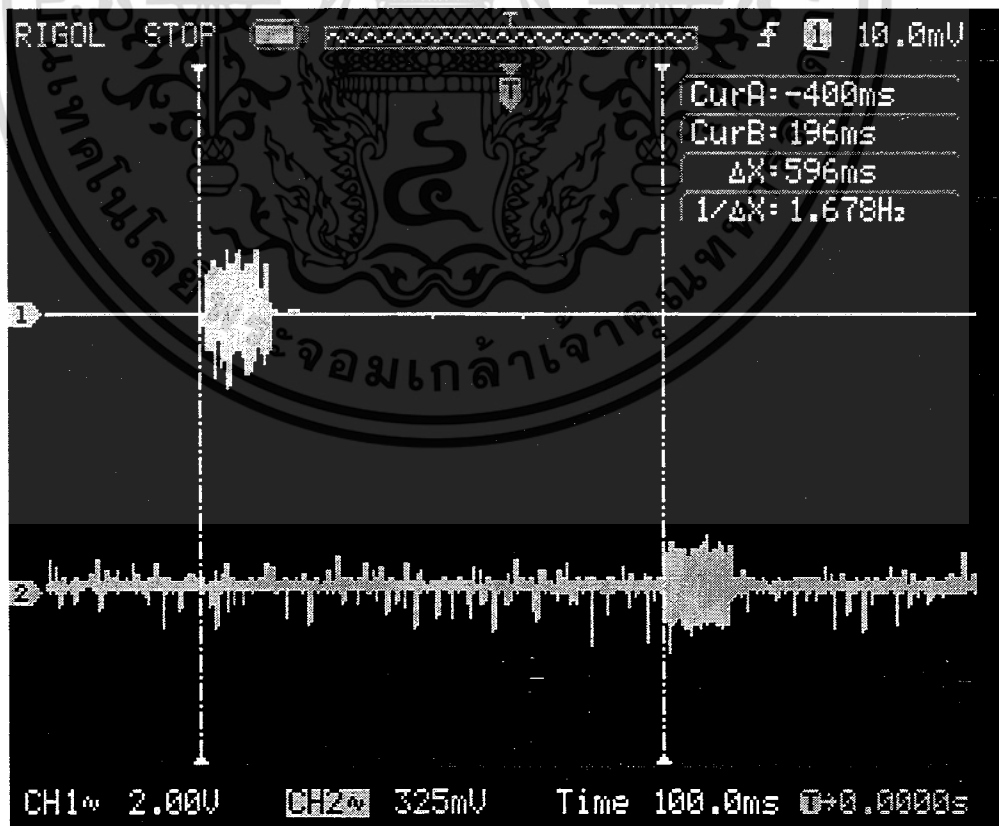
รูปที่ 4.26 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
 CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 7

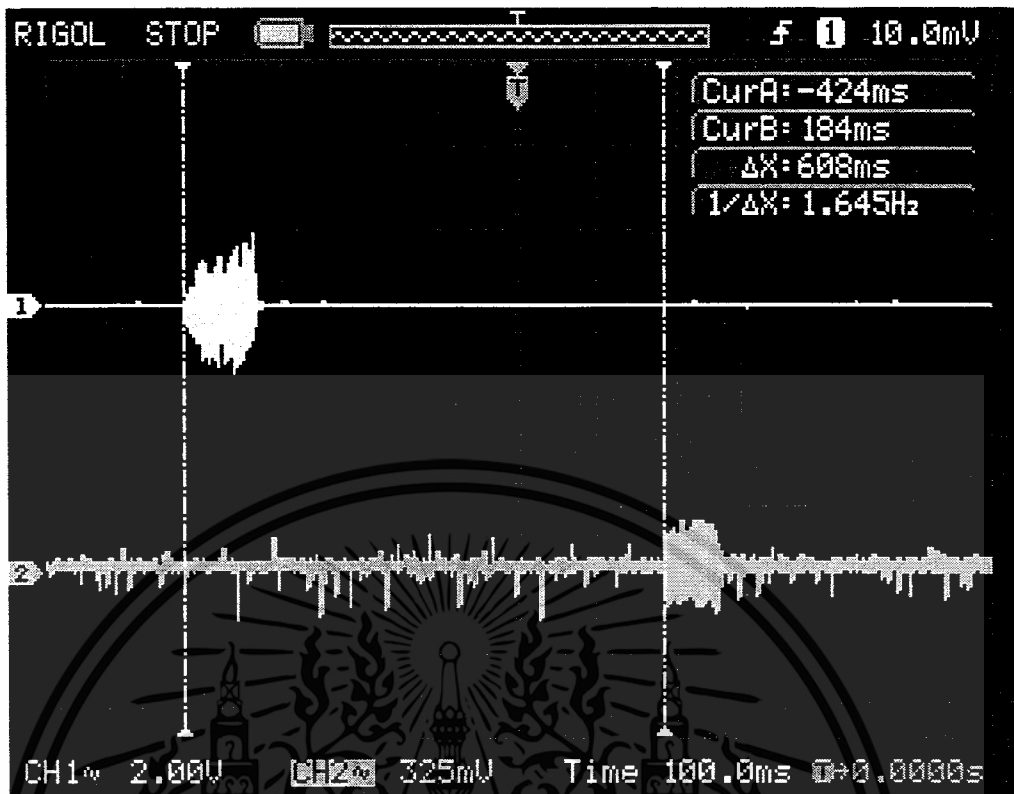
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง



รูปที่ 4.28 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 8

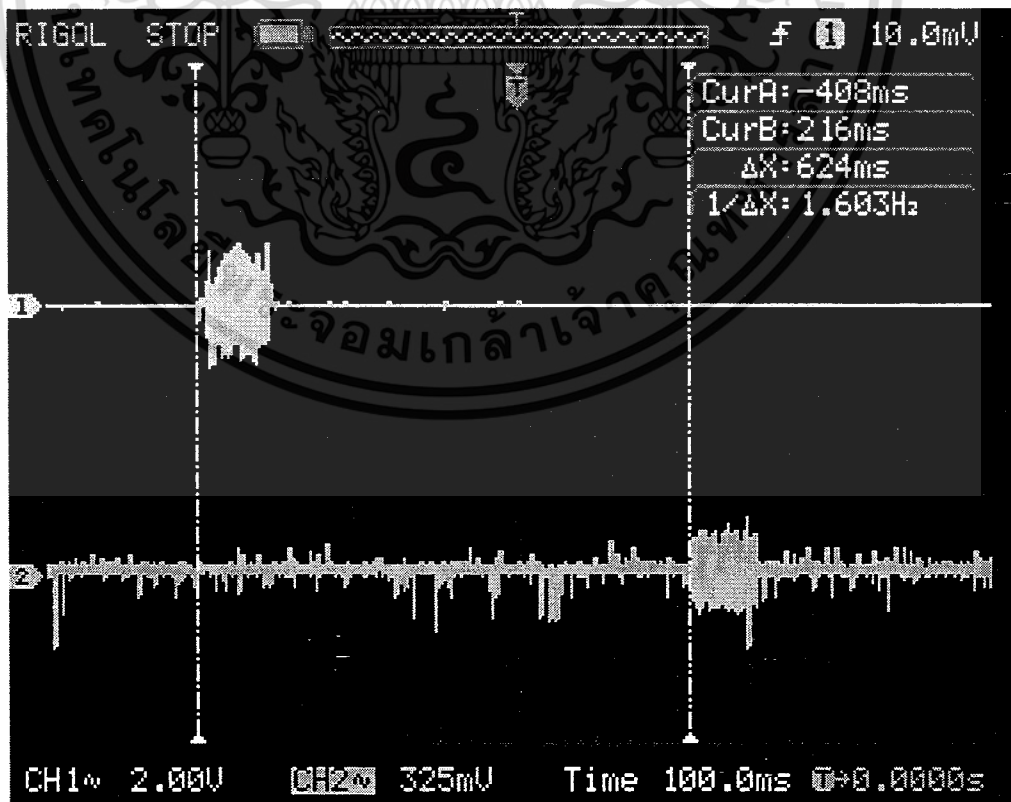
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 9

CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง



รูปที่ 4.30 เปรียบเทียบสัญญาณต้นทางกับปลายทางครั้งที่ 10

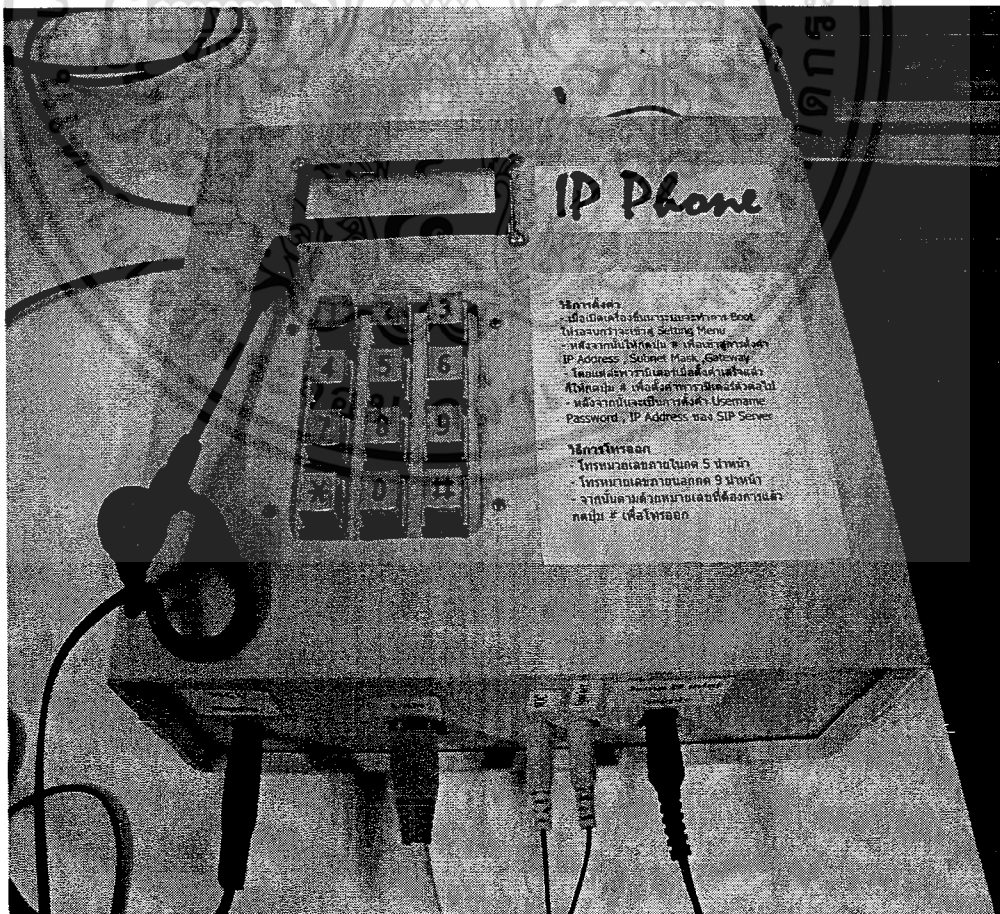
CH1: สัญญาณเสียงจากต้นทาง, CH2: สัญญาณเสียงจากปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดสอบ Delay

ครั้งที่	Delay (ms)
1	644
2	672
3	780
4	696
5	616
6	616
7	668
8	596
9	608
10	624
เฉลี่ย	652

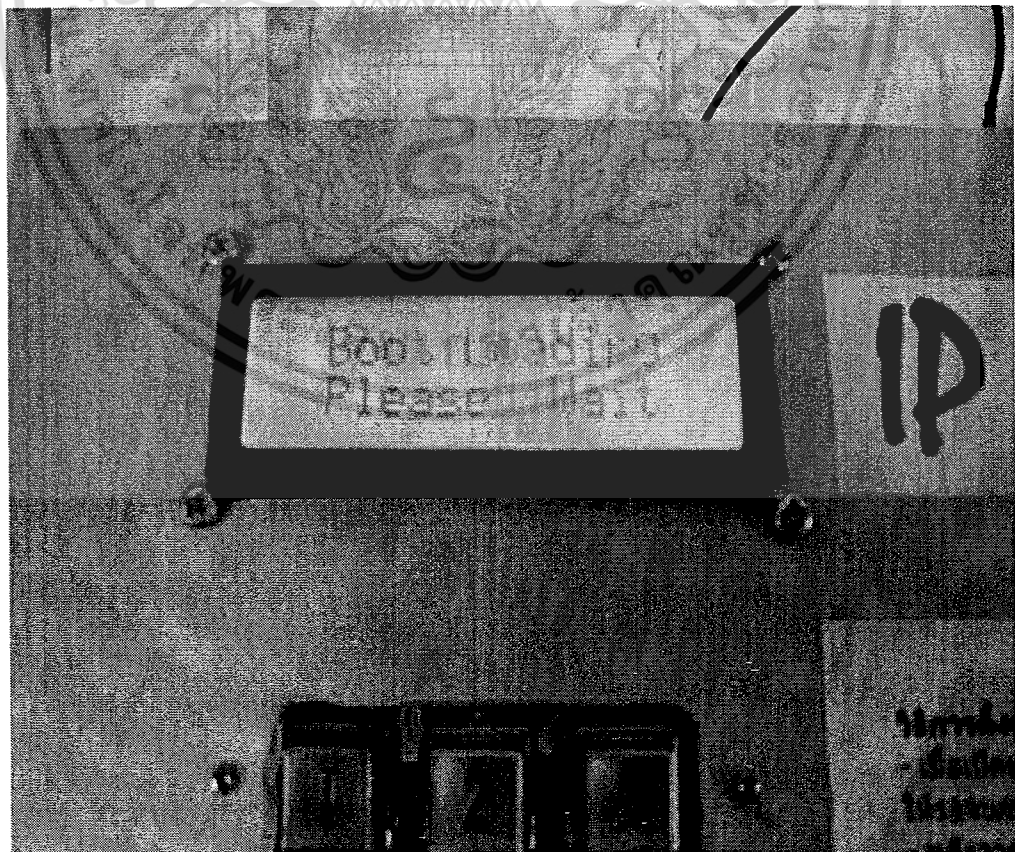
#### 4.1.6 การใช้งานของโทรศัพท์



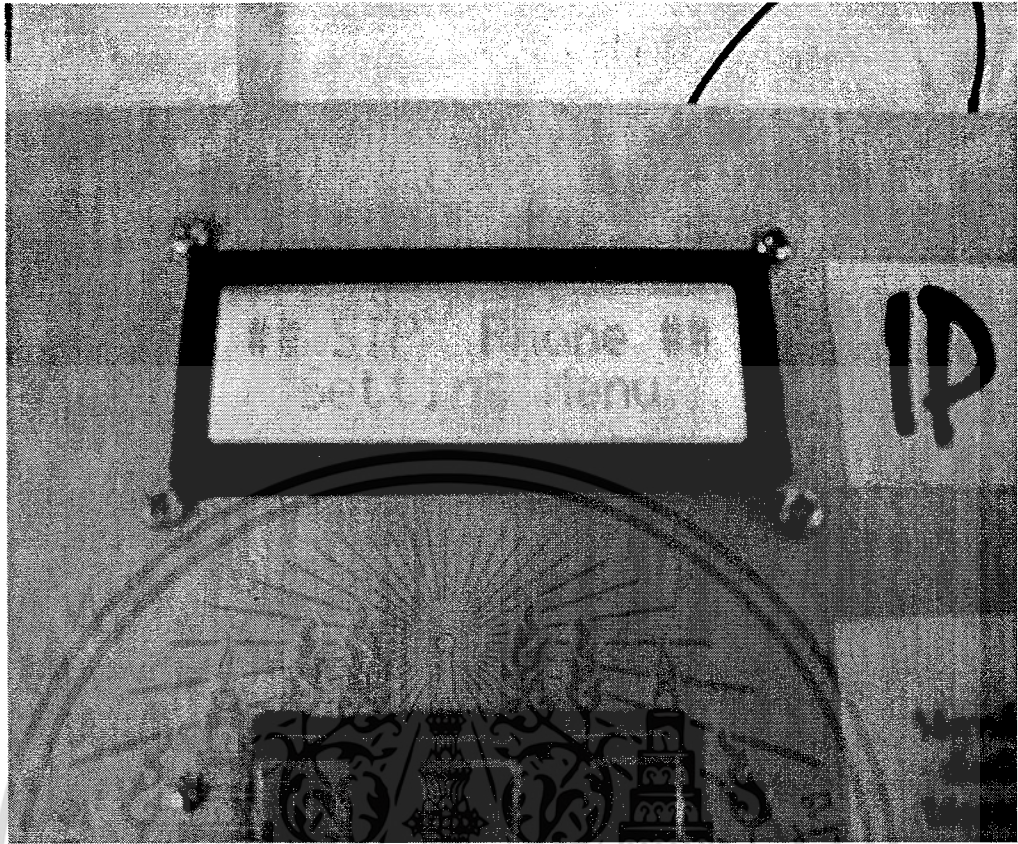
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 4.31 โทรศัพท์ที่ใช้ในการสื่อสาร ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



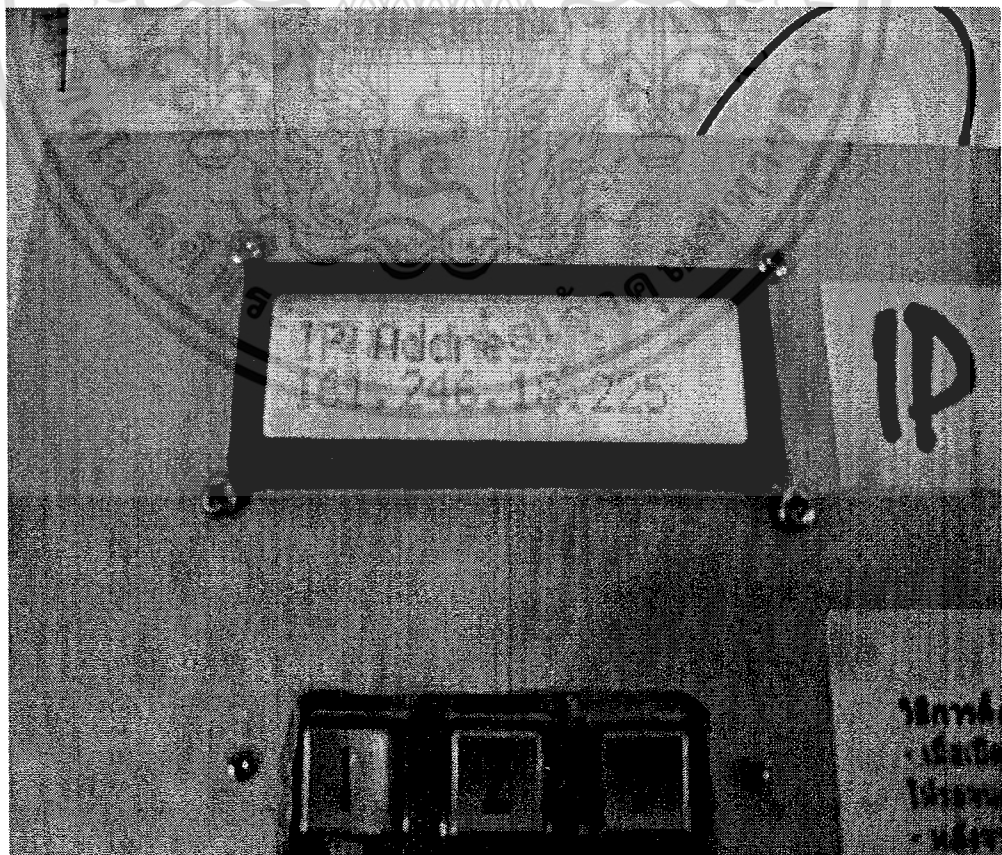
รูปที่ 4.32 พอร์ตที่ใช้ในการเชื่อมต่อต่างๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาดูงานนี้ไปเพื่อประโยชน์ด้านการค้า  
รูปที่ 4.33 สถานะจอ LCD เมื่อรอทำการบูตเข้าสู่ระบบ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 สถานะจอ LCD เมื่อป้อนเข้าสู่ระบบเรียบร้อยแล้ว



รูปที่ 4.35 ทำการตั้งค่า IP Address

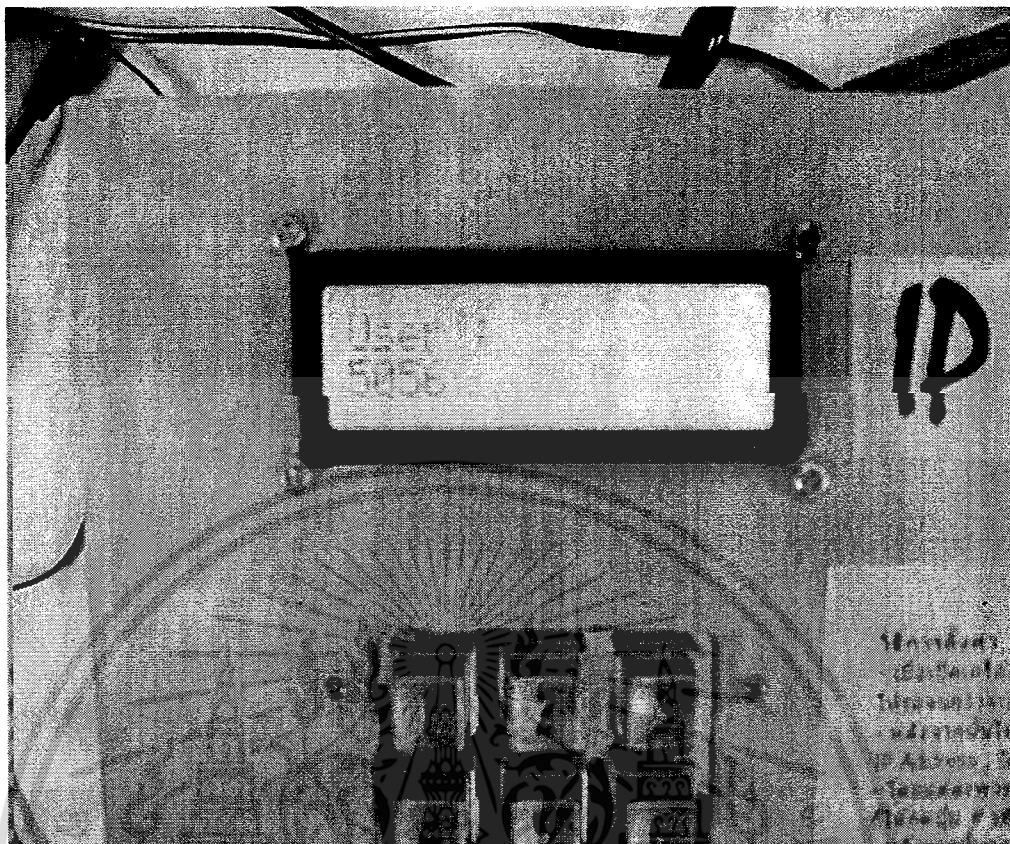
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไปขอจดหมายไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



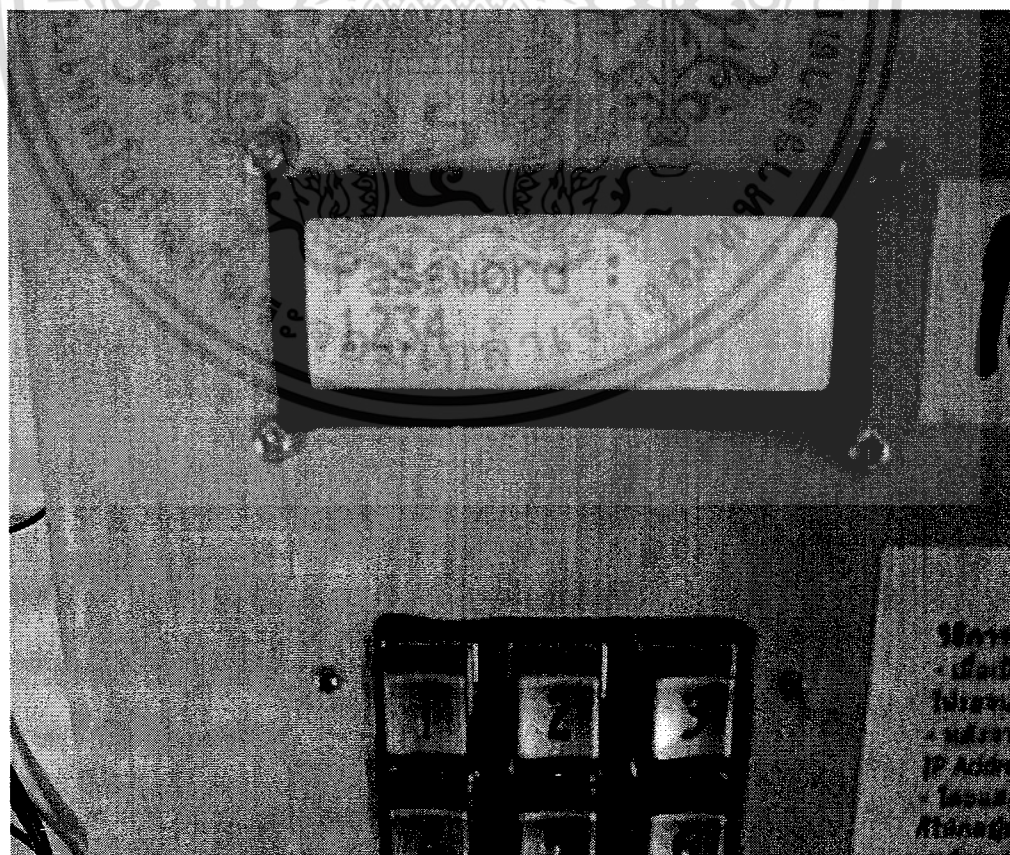
รูปที่ 4.36 ทำการตั้งค่า Subnet Mask



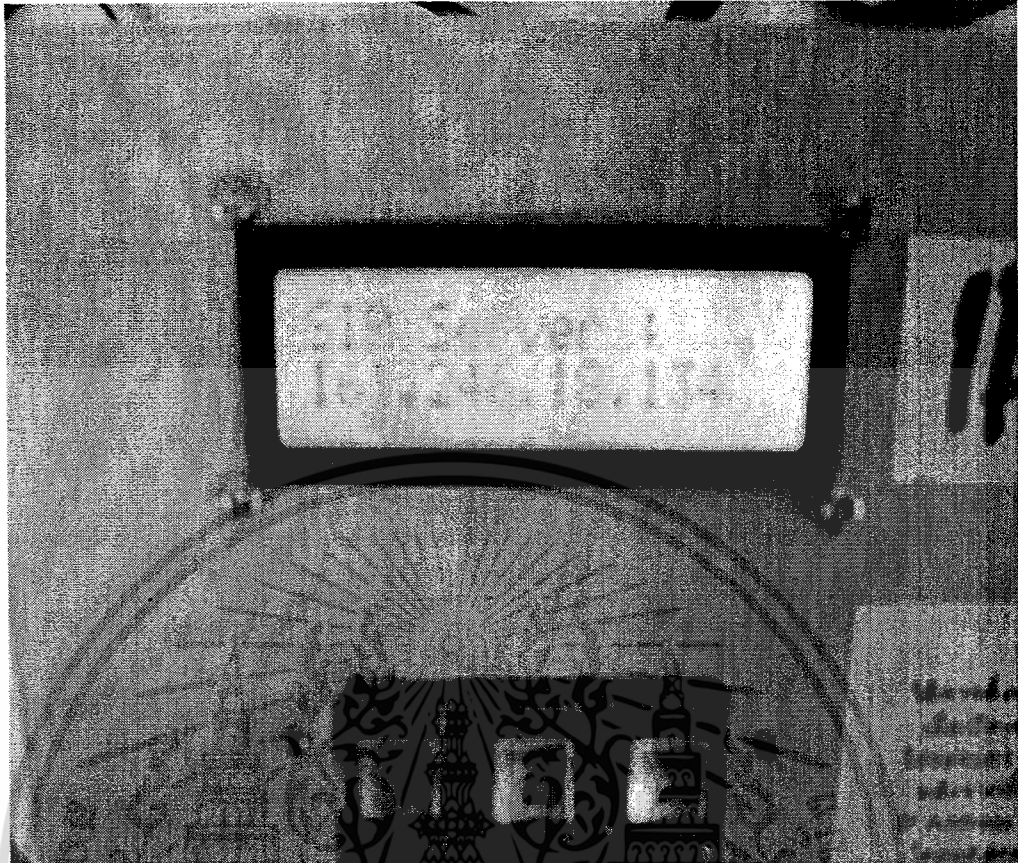
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 4.37 ทำการตั้งค่า Default Gateway หน้าที่ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.38 ใส่ค่า User ID ที่ได้รับจากการลงทะเบียนผ่านทางหน้าเว็บเพจ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.39 ใส่ค่า Password ที่ได้ทำการตั้งค่าเอาไว้ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



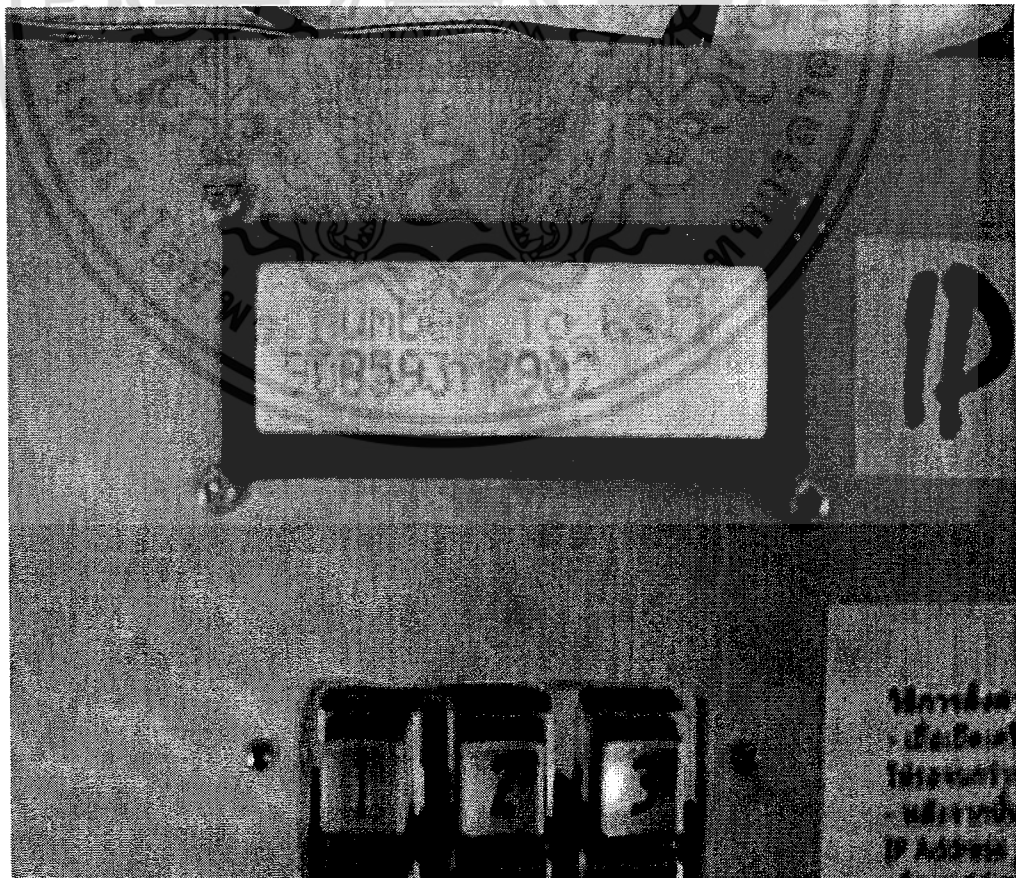
รูปที่ 4.40 ใส่ค่า IP Address ของ SIP Server โดยในปัจจุบันจะตั้งอยู่ที่ 161.246.18.134



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม วิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรเผยแพร่ไปโดยไม่ได้รับอนุญาต  
รูปที่ 4.41 เข้าสู่โหมดกรอกรับหมายเลขเพื่อโทรออก  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.42 การโทรไปหาหมายเลขภายใน



รูปที่ 4.43 การโทรไปหาหมายเลขภายนอก

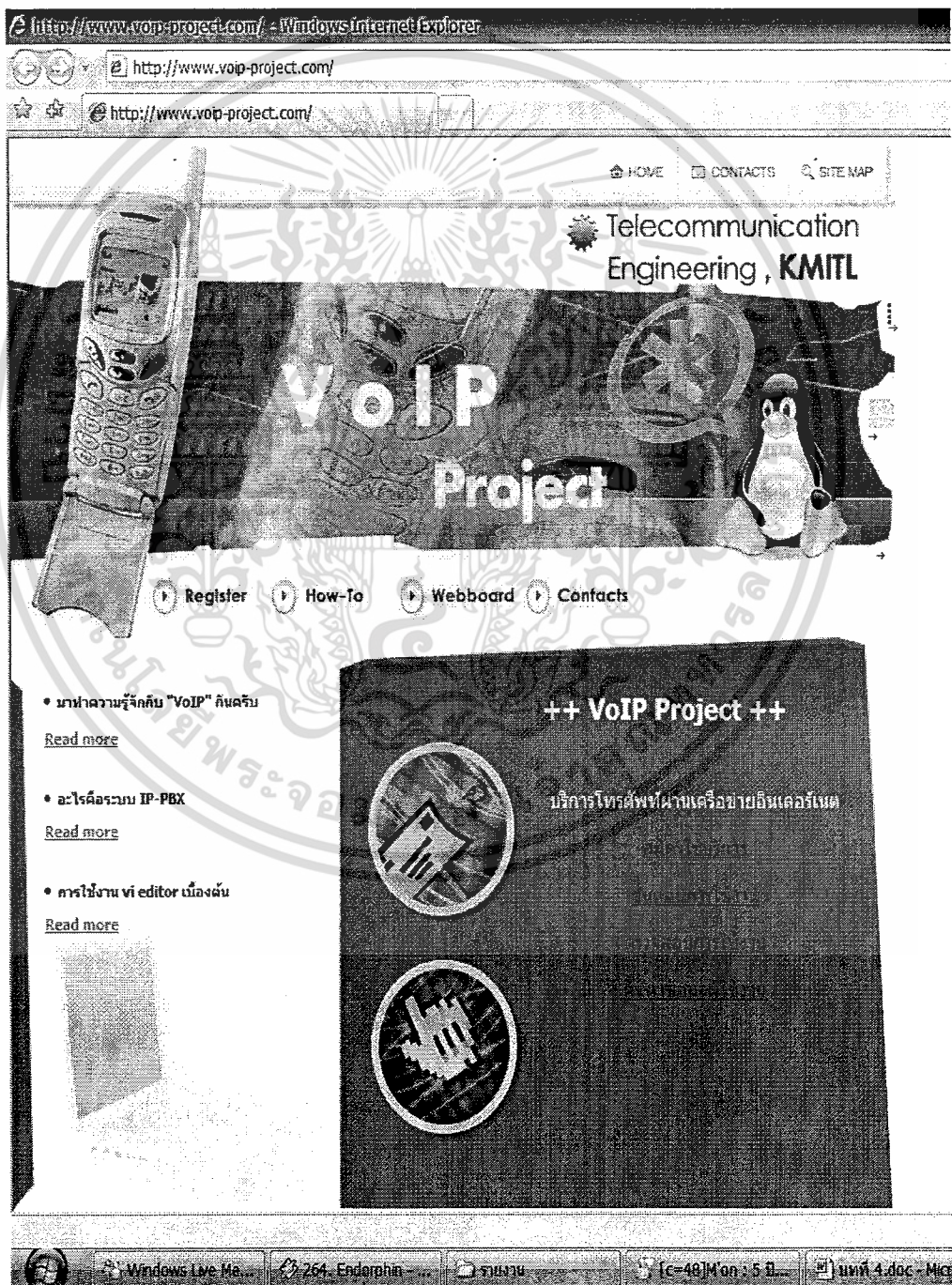
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในขององค์กร ไม่ควรเปิดเผยข้อมูลไปยังบุคคลภายนอกไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ผลการทดลองในส่วนของ Server

ในส่วนนี้จะประกอบไปด้วยส่วนหลักๆ คือ

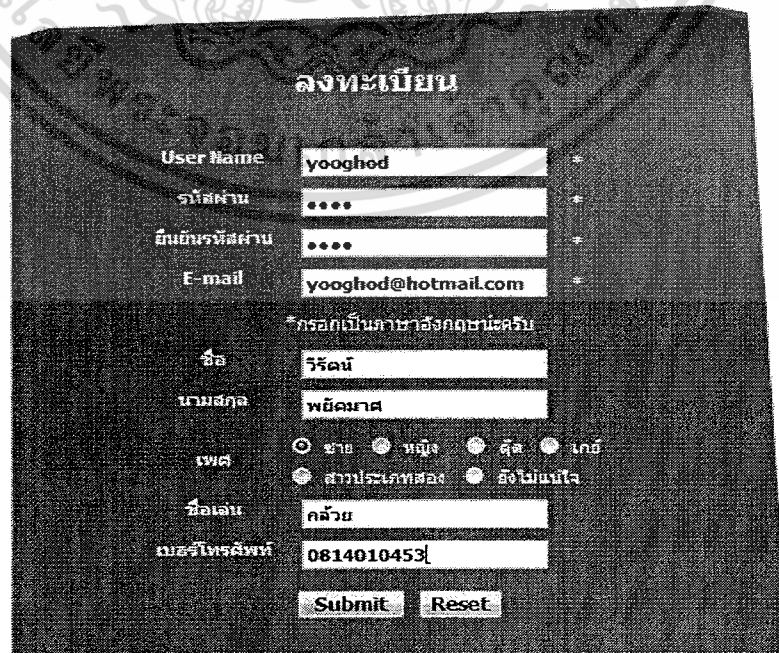
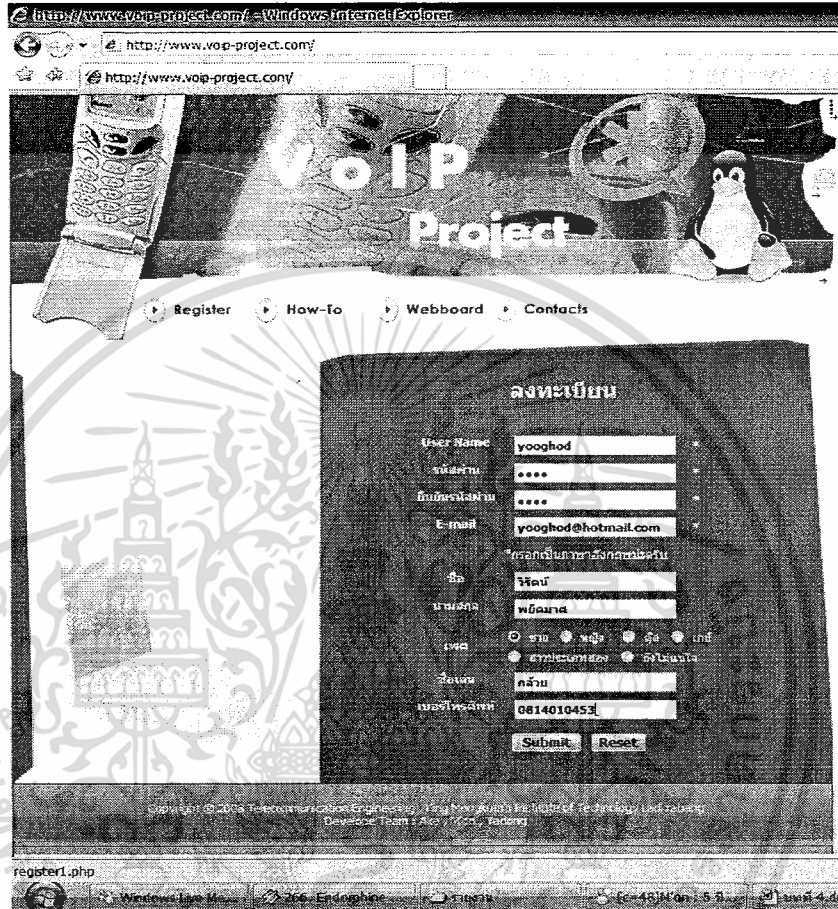
1. เว็บเพจที่ใช้ในการลงทะเบียน
2. ฐานข้อมูลที่เก็บข้อมูลผู้ใช้งาน
3. ฐานข้อมูลที่เก็บการใช้งานโทรศัพท์

โดยที่เว็บเพจที่ใช้ในการลงทะเบียนสมัครใช้งาน SIP Server นั้นจะสามารถเข้าได้ทาง IP 161.246.18.134 หรือ ทาง domain name <http://www.voip-project.com> ดังรูปที่ 4.44



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 4.44 เว็บเพจหน้าหลัก ญาติหน้าไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคลิกเข้าไปที่ Link “สมัครใช้บริการ” จะเข้าสู่หน้าที่ให้กรอกข้อมูล โดยจะนำข้อมูลนี้ไปเก็บในฐานข้อมูลของผู้ใช้งาน ส่วนเบอร์โทรศัพท์ที่มีอยู่นั้นจะใช้ในการติดต่อในยามที่ไม่สามารถติดต่อไปยังหมายเลขโทรศัพท์ที่เป็น IP ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 4.45 เว็บไซต์ที่ใช้ในการลงทะเบียนหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากกด Submit แล้ว ระบบจะทำการจัดเก็บข้อมูลผู้ใช้งานในฐานข้อมูลและเพิ่ม Account เข้าไปที่ SIP Server เพื่อใช้ในการ Login เข้ามาใช้งาน โดยเมื่อเสร็จสิ้นขั้นตอนนี้จะได้รับ ID และ Password ที่ใช้ในการ REGISTER ไปยัง SIP Server

```
mysql> select ID,PASSWORD,EMAIL,TEL from member where ID=5056;
+-----+-----+-----+-----+
| ID | PASSWORD | EMAIL | TEL |
+-----+-----+-----+-----+
| 5056 | 1234 | yooghod@hotmail.com | 0814010453 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Connected to 161.246.18.134

รูปที่ 4.46 ข้อมูลของผู้ใช้งานที่ถูกเก็บลงไปฐานข้อมูล

โดยเมื่อเราเข้าไปที่หน้าจอคอนโซลของโปรแกรม Asterisk แล้วใช้คำสั่ง “sip show peer” จะพบว่า user 5056 ปรากฏขึ้นมาดังรูป

```
[root@voipproject ~]# asterisk -r
Asterisk 1.4.20.1, Copyright (C) 1999 - 2008 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.

=====
Connected to Asterisk 1.4.20.1 currently running on voipproject (pid = 22224)
Verbosity is at least 3
voipproject*CLI> sip show peers
Name/username      Host                Dyn Nat ACL Port      Status
-----
5055                (Unspecified)      D          0         Unmonitored
5054/5054          161.246.18.133    D          8716     Unmonitored
5053/5053          (Unspecified)      D          0         Unmonitored
```

รูปที่ 4.47 หน้าจอคอนโซลของ Asterisk เมื่อใช้คำสั่ง “sip show peers”

ลงทะเบียนเสร็จแล้วครับ

ID	5056
Password	1234

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานภายในอาคารสื่อสารโทรคมนาคม โดยไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต หากมีการนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต จะถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่เราสามารถดูรายละเอียดของหมายเลขที่ต้องการได้โดยเข้าไปที่ Link “ค้นหาข้อมูลผู้ใช้งาน” ในหน้าแรก จากนั้นป้อนหมายเลขที่เราต้องการจะค้นหาลงไป

ค้นหาข้อมูลผู้ใช้งาน

กรอกIDที่ต้องการค้นหา 5056

Submit

รูปที่ 4.49 ค้นหาข้อมูลหมายเลขที่ต้องการ

ตารางแสดงข้อมูลของIDที่ต้องการค้นหา

ID	5056
USERNAME	yooghod
NAME	วีรัตน์
NICKNAME	กล้วย
TELEPHONE	0814010453
EMAIL	yooghod@hotmail.com

รูปที่ 4.50 ข้อมูลผู้ใช้งานหมายเลขที่ค้นหา

จากรูปที่ 4.49 หลังจากกด Submit แล้ว ระบบจะทำการค้นหาข้อมูลของผู้ใช้งานหมายเลขที่กรอกลงไป ในฐานข้อมูลจากนั้นก็ทำการ query มาแสดงบนหน้าเว็บเพจ ตามรูปที่ 4.50

เราสามารถดูข้อมูลการใช้งานการโทรออกของหมายเลขที่ต้องการได้โดยเข้าไปที่ Link “ตรวจสอบการใช้งาน” จากนั้นก็เลือกวันที่ เดือน ปี และหมายเลขที่เราต้องการจะดูลงไป

ข้อมูลการใช้งานโทรศัพท์

วันที่ วันเสาร์ ▼ มกราคม ▼ 2551 ▼

ID 5056

Submit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในวงจำกัดเท่านั้น ไม่ควรเผยแพร่ไปวงกว้าง ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากกด Submit แล้ว ระบบจะทำการค้นหาข้อมูลการโทรศัพท์ของหมายเลขที่กรอกกลงไปในฐานข้อมูลจากนั้นก็ทำการ query มาแสดงบนหน้าเว็บเพจ โดยจะแสดงเฉพาะฟิลด์ calldate, src, DST และ billsec ตามรูปที่ 4.52

**ตารางแสดงข้อมูลการใช้งาน**

วันที่และเวลาที่โทร	ต้นทาง	ปลายทาง	ระยะเวลา (วินาที)
2009-01-29 05:59:51	5056	90814010453	34
2009-01-29 06:02:26	5056	5050	0
2009-01-29 06:05:33	5056	9025172715	89
2009-01-29 06:14:06	5056	5045	28
2009-01-29 06:15:31	5056	5045	60

รูปที่ 4.52 ข้อมูลการใช้งานโทรศัพท์ของหมายเลขที่ค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลและวิจารณ์การทดลอง

#### 5.1 สรุปผลการทดลอง

- จากการทดลองฟังเสียงในส่วนของโทรศัพท์นั้นยังค่อนข้างล่าช้า เนื่องจากหน่วยประมวลผลในส่วนของเสียงมีความเร็วไม่มากนัก
- การจัดการในเรื่องการรับสายยังทำได้ไม่ดีเท่าที่ควร
- โปรแกรมมีการแฉกบ้างในบางครั้ง
- การตอบสนองของการโทรออก มีช้าเป็นบางครั้ง

#### 5.2 ปัญหาและอุปสรรคในการทำงาน

- ผู้จัดทำขาดความเชี่ยวชาญในระบบปฏิบัติการ Linux ทำให้การดำเนินการช่วงๆ แรก เป็นไปอย่างยากลำบาก
- บอร์ดพัฒนาทางด้าน Embedded Linux นั้นมีราคาค่อนข้างแพง ทำให้หมดงบประมาณไปค่อนข้างเยอะ
- การศึกษาไลบรารี PJSIP ในช่วงแรกๆ นั้นเป็นไปอย่างยากลำบาก เนื่องจากผู้จัดทำไม่เคยเขียนโปรแกรมที่ต้องใช้ API มาก่อน

#### 5.3 แนวทางในการพัฒนาต่อ

##### 5.3.1 ส่วนของโทรศัพท์

- เพิ่มปุ่มกดให้มากกว่านี้เพื่อรองรับฟังก์ชันการทำงานเพิ่มต่างๆ
- พัฒนาให้สามารถเก็บเบอร์ผู้ใช้งานได้
- ทดสอบ Codec ชนิดอื่นๆ ที่สามารถนำมาใช้งานร่วมกับบอร์ด JampII mini ได้
- เพิ่มส่วนแสดงผล LCD ให้มีขนาดใหญ่กว่านี้
- เพิ่มส่วนของ Hook Switch และ หูโทรศัพท์เพื่อความ สะดวกต่อการใช้งาน

##### 5.3.2 ส่วนของ Server

- พัฒนาให้ผู้ใช้สามารถแก้ไขข้อมูลส่วนตัวได้
- ศึกษา Application ของ Asterisk เพิ่มเติม เช่น ระบบฝากข้อความเสียง, ระบบเสียงรอสาย, การส่งแฟกซ์ผ่านทาง IP เป็นต้น
- ติดตั้งการ์ดสายนอก เพื่อให้หมายเลขโทรศัพท์ภายนอกสามารถโทรหาหมายเลขภายในได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] Jim Van Meggelen , Leif Madsen , Jared Smith , “Asterisk The Future of Telephony”, O’REILLY , 2007
- [2] Mark Spencer , Mack Allison , Christopher Rhodes , “The Asterisk Handbook version 2” , Digium,Inc. , 2003
- [3] David Gomilion , Barrie Dempster , “Building Telephony Systems with Asterisk” , PACK Publishing , 2005
- [4] Jonny Martin , “Asterisk Tutorial” , Citylink
- [5] Ben Jackson aka Black Ratchet , Champ Clark III aka Da Beave “Asterisk Hacking” , SYNGRESS , 2007
- [6] DOUGLAS E. COMER , “Internetworking With TCP/IP” , ,Prentice Hall of India 2008
- [7] DANEIL COLLINS , “CARRIER GRADE VOICE OVER IP” , McGraw-Hill Networking 2004
- [8] Colin Perkins , “RTP: Audio and Video for the Internet” , Addison Wesley 2003
- [9] Alberto EscuderoPascual , Louise Berthilson , “VoIP 4D Primer” 2007
- [10] Alan B.Johnson ”SIP Understanding Session Initiation Protocol” , Artech House , 2004
- [11] Henry Sinnreich , Alan B.Johnson “Internet Communication Using SIP” , Wiley Publishing
- [12] Serial Programming HOWTO ,Gary Frerking , <http://www.faq.org/docs/Linux-HOWTO>
- [13] PJSIP Developer’s Guide , Benny Prijono , <http://www.pjsip.org>
- [14] กิตติพงษ์ สุวรรณราช , “ออกแบบและติดตั้งระบบโทรศัพท์ IP-PBX ด้วย Asterisk” , บริษัท ออฟเซ็ท เพรส จำกัด , 2551
- [15] ศูนย์วิจัยเทคโนโลยีเครือข่าย ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ , “ASTERISK WORKSHOP”
- [16] ศุภกฤษ นาคป้อมฉิน , “การพัฒนาระบบใบแจ้งค่าใช้จ่ายบริการระบบโทรศัพท์ผ่านเครือข่าย” , สาขาวิทยาการคอมพิวเตอร์ ภาควิชาคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร , 2549
- [17] จตุชัย แพงจันทร์ , อนุโชต วุฒิพรพงษ์ , “เจาะระบบ Network 2<sup>nd</sup>” , บริษัท ไอดีซีอินโฟ คิสทริบิวเตอร์ เซ็นเตอร์ จำกัด , 2551
- [18] เอกสิทธิ์ วิริยจารี , “เรียนรู้ระบบเนตเวิร์คจากอุปกรณ์ของ Cisco” , บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2551
- [19] อาณัติ รัตนธิรกุล , “ก้าวสู่ผู้ดูแลเครือข่ายคอมพิวเตอร์ในองค์กร (ภาคปฏิบัติ)” , บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2550
- [20] สุวัฒน์ ปุณณชัยยะ , ตัน ตันท์สุทธีวงศ์ , สุพจน์ ปุณณชัยยะ , “เปิดโลก TCP/IP และ โปรโตคอลของอินเทอร์เน็ต (Second Edition) , กรุงเทพฯ ; โปรวิชั่น , 2545

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานโดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [21] ก่อกิจ วีระอาชากุล , “ติดตั้ง และปรับแต่งเซิร์ฟเวอร์ Linux สำหรับ Admin Linux โดยเฉพาะ” , บริษัท ไอดีซีอินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด , 2545
- [22] กิตติภูมิ วรจักร , “PHP เปลี่ยนวิธีการสร้างโฮมเพจอย่างมืออาชีพ” , บริษัท วิตดี กรุ๊ป จำกัด , 2543
- [23] สมศักดิ์ โชคชัยชุกติกุล , “อินไซด์ PHP5” , บริษัท โปรวิชั่น จำกัด , 2547
- [24] สาธิต ชัยวิวัฒน์ตระกูล , “เก่ง PHP5 ให้ครบสูตร” , บริษัท วิตดี กรุ๊ป จำกัด , 2550
- [25] นิรุช อำนวยศิลป์ , “PHP How-To and Web-Based Application Techniques เล่ม 1-3” , บริษัท ด้านสุทธาการพิมพ์ จำกัด
- [26] ดวงพร เกียงคำ , วงศ์ประชา จันทร์สมวงศ์ , “อินไซด์ Dreamweaver 8” , บริษัท โปรวิชั่น จำกัด , 2549
- [27] นิรุช อำนวยศิลป์ , “เขียนโปรแกรมภาษาซี ฉบับสมบูรณ์” , บริษัท ด้านสุทธาการพิมพ์ จำกัด
- [28] อรพิน ประวัตินริสุทธิ์ , “คู่มือเรียนภาษาซี” , บริษัท โปรวิชั่น จำกัด , 2547
- [29] อุดม รานอก , “ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51” , บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด , 2548
- [30] ขจร อนุดิษฐ์ , “การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษา C” , สำนักพิมพ์ Core Function , 2550
- [31] รศ. สมยศ จุณณะปิยะ , “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์” , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2546
- [32] <http://www.voip-info.org>
- [33] <http://forum.asteriskthailand.com/>
- [34] <http://gotoknow.org/post/tag/voip>
- [35] <http://www.asteriskdiy.com>
- [36] <http://www.ipbxsystem.com/>
- [37] <http://www.asteriskclub.com>
- [38] <http://www.asterisk.org/>
- [39] <http://forums.digium.com/>
- [40] <http://www.pjsip.org/>
- [41] <http://www.voipthai.net> , “Easy AsteriskNow”
- [42] <http://www.ckmit.com/summer/aek/VoIP/VoIP1.htm>
- [43] <http://www.thaiembedded.com/blog/>
- [44] <http://old.iptel.org/sip/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้