

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบโทรศัพท์บนโครงข่ายIP

IP PHONE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์บนโครงข่าย IP

IP PHONE

โดย

นายณภัศ รุ่งเรืองบางชัน 47010358

นางสาวณภัศร ลาดศิริวงศ์ 47010359

อาจารย์ที่ปรึกษา

รศ.ดร. ปราโมทย์ วาดเขียน

ศ.ดร.วิวัฒน์ กิรานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2550

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบโทรศัพท์บนโครงข่าย IP

IP PHONE

ผู้จัดทำ

1. นายณภัต รุ่งเรืองบางชัน 47010358

2. นางสาวณภัทร ลากศิริวงศ์ 47010359

..... ปาโมทย์ อาจารย์ที่ปรึกษา
(รศ.ดร. ปาโมทย์ วาดเขียน)

..... อาจารย์ที่ปรึกษา
(ศ.ดร.วิวัฒน์ กิรานนท์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบโทรศัพท์บนโครงข่าย IP

IP PHONE

โดย นาย นภัสร รุ่งเรืองบางชัน 47010358

นางสาว นภัสร ลากศิริวงศ์ 47010359

อาจารย์ที่ปรึกษา รศ.ดร.ปราโมทย์ วาดเขียน

ศ.ดร.วิวัฒน์ ภิรานนท์

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้ทำการศึกษาและพัฒนาาระบบโทรศัพท์บนโครงข่าย IP (IP Phone) ซึ่งเป็นเทคโนโลยี VoIP ที่แปลงเสียงพูดให้อยู่ในรูปแบบที่สามารถส่งผ่านโครงข่าย IP โดยในโครงข่ายนี้ได้ใช้อินเทอร์เน็ตโมดูลในการเชื่อมต่อกับโครงข่าย IP บนระบบสมองกลฝังตัว (Embedded System) ซึ่งเป็นระบบที่ทำงานเปรียบเสมือนมีระบบคอมพิวเตอร์ภายใน คอยควบคุมการทำงานเฉพาะทางตามความต้องการ ระบบโทรศัพท์ผ่านโครงข่าย IP ที่นำเสนอดังกล่าว สามารถส่งผ่านข้อมูลได้อย่างรวดเร็ว ทั้งยังประหยัดค่าใช้จ่ายในการวางโครงข่ายโทรศัพท์และยังช่วยลดค่าใช้จ่ายในการสนทนาภายในองค์กรได้อีกด้วย

Abstract

Today's communication system is rapidly developing. People are currently designing telecommunication system through IP network which is being used throughout and has the ability to transfer information quickly. It keeps the expenses low in the process of constructing the telecommunication network and also decreases the cost of communication in the company also. This thesis is to study and develop a IP Phone based on the VoIP technology, therefore is converted voices into appropriate formats that can be transferred through the IP network. In addition, the Ethernet module is employed and connected with the IP network on the embedded system, which is a system that virtually has a computer system to control the particular work.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่1 บทนำ	1
บทที่2 ทฤษฎีและหลักการ	3
2.1 โพรโตคอล TCP/IP	3
2.1.1 จุดประสงค์ของการสื่อสารตามมาตรฐาน ของโปรโตคอล TCP/IP	3
2.1.2 โครงสร้างของโปรโตคอล TCP/IP	3
2.1.3 Encapsulation และ Decapsulation	4
2.1.4 Application Layer	5
2.1.5 Transport Layer	6
2.1.5.1 โพรโตคอล TCP (Transmission Control Protocol)	7
2.1.5.2 โพรโตคอล UDP (User Datagram Protocol)	8
2.1.6 Internet work Layer	11
2.1.6.1 โพรโตคอล IP (Internet Protocol)	11
2.1.6.2 โพรโตคอล ARP	17
2.1.7 Network Interface Layer	18
2.1.8 MAC Address	19
2.1.9 ระบบเครือข่าย Ethernet	19
2.1.10 ส่วนประกอบหลักที่สำคัญของเครือข่าย Ethernet	20
2.1.11 เฟรมบนระบบ Ethernet	20
2.2 Voice over IP (VoIP)	27
2.2.1 รูปแบบการใช้งานของ VoIP	28
2.2.2 กระบวนการทำงานของ VoIP	29
2.2.3 การประยุกต์ใช้ VoIP ภายในองค์กร	30
2.3 สถาปัตยกรรม ซิปียู ARM 7	30
2.3.1 ARM7 TDMI	31
2.3.2 ไมโครคอนโทรลเลอร์ ARM7 Philips LPC2148	32
2.3.3 บล็อกไดอะแกรมของ LPC2148	34
2.3.4 การจัดการหน่วยความจำของ LPC2148	35
2.3.5 การจัดขาของ LPC2148	37
2.4 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	38

สารบัญ(ต่อ)

เรื่อง	หน้า
2.4.1 คุณสมบัติของMCS-51	38
2.4.2 การจัดตำแหน่งขาของ MCS-51 และหน้าที่การทำงาน	38
2.5 ระบบเรียลไทม์ซิสเต็ม (real time system)	41
2.5.1 Task และ Multitasking	41
2.5.1.1 Task	41
2.5.1.2 Multitasking	42
2.5.2 Kernel	42
2.5.3 Scheduler	42
2.5.4 Task Priority	43
2.5.5 Deadlock หรือ Deadly Embrace	43
2.6 ทฤษฎีการชั่งตัวอย่าง (Sampling Theorem)	43
2.7 การควอนไทซ์สัญญาณ	44
บทที่ 3 การคำนวณและการสร้าง	46
3.1 Block Diagram	46
3.2 วงจรแปลงแรงดันไฟจากไฟ +15 V เป็น +5V	46
3.3 วงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V	47
3.4 วงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V	47
3.5 วงจร Microphone Pre-Amplifier	48
3.5.1 วงจร Pre-microphone และ Instrument Amplifier	48
3.5.2 วงจร Non-Inverting Summing Amplifier	49
3.5.3 วงจร Voltage follower (Buffer)	49
3.5.4 วงจร Second order Low Pass Filter	50
3.6 วงจร Power Amplifier with gain=20	51
3.7 วงจร Ethernet module	52
3.8 วงจรไมโครคอนโทรลเลอร์ ARM7 ควบคุม Ethernet module	53
3.9 วงจรไมโครคอนโทรลเลอร์ MCS51	54
3.10 วงจรคีย์แพด (Keypad)	54
3.11 วงจรLCD	55
3.12 แผนผังแสดงการทำงานของโปรแกรม	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.12.1 โปรแกรม Main	55
3.12.2 โปรแกรม Task0: Signaling (Priority=15)	56
3.12.3 โปรแกรม Task1: ADC (Priority=10)	57
3.12.4 โปรแกรม Task2: Ethernet Send (Priority=5)	57
3.12.5 โปรแกรม Task3: Ethernet Receive (Priority=5)	58
3.12.6 โปรแกรม Task4: DAC (Priority=7)	58
3.12.7 โปรแกรม Task5: Ending (Hang up, Priority=20)	59
3.12.8 โปรแกรม Task6: Ending (Receive BYE, Priority=20)	60
3.12.9 โปรแกรม Task7: Sound Signaling (Priority=5)	61
บทที่ 4 การทดลองและผลการทดลอง	62
4.1 วงจรแปลงแรงดันไฟจาก +15V เป็น +5V	62
4.2 วงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V	63
4.3 วงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V	64
4.4 วงจร Microphone Pre-Amplifier	65
4.4.1 วงจร Pre-microphone และ Instrument Amplifier	66
4.4.2 วงจร Voltage follower I (Buffer I)	67
4.4.3 วงจร Non-inverting Summing Amplifier	68
4.4.4 วงจร Voltage follower II (Buffer II)	69
4.4.5 วงจร Second order Low Pass Filter	70
4.4.6 วงจร Voltage follower III (Buffer III)	74
4.5 วงจร Power Amplifier with gain=20	76
4.6 การส่งสัญญาณเสียงผ่านเครือข่าย Ethernet	76
4.7 รูปแบบของสัญญาณเสียงแสดงสถานะ	81
4.7.1 สัญญาณเสียง Ring Tone	81
4.7.2 สัญญาณเสียง Ring Back Tone	82
4.7.3 สัญญาณเสียง Busy Tone	83
4.8 การดักจับข้อมูล (Sniffer) โดยโปรแกรม Wire Shark	84
4.8.1 การดักจับข้อมูลที่ส่งออกโดยการ Sampling สัญญาณ Sine โดยโปรแกรม Wire shark	88

สารบัญ (ต่อ)

เรื่อง	หน้า
4.9 การแสดงผลของหน้าจอ LCD	92
บทที่ 5 บทวิจารณ์และบทสรุป	95
เอกสารอ้างอิง	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 กลไกของโพรโตคอลมาตรฐาน OSI model	4
รูปที่ 2.2 ขั้นตอนการ Encapsulation และ Decapsulation	4
รูปที่ 2.3 Data Structures	5
รูปที่ 2.4 การใช้งาน port ของแต่ละโพรโตคอล	6
รูปที่ 2.5 การส่งข้อมูลจาก Application ไปยัง Transport Layer	7
รูปที่ 2.6 โครงสร้างของโพรโตคอล TCP	8
รูปที่ 2.7 UDP Message encapsulation	8
รูปที่ 2.8 โครงสร้างของ UDP Header	9
รูปที่ 2.9 Pseudo Header	11
รูปที่ 2.10 IP Datagram	12
รูปที่ 2.11 โครงสร้าง IP Header	12
รูปที่ 2.12 การ Bind IP address	17
รูปที่ 2.13 โครงสร้างของโพรโตคอล TCP/IP	18
รูปที่ 2.14 ลักษณะโครงสร้างของเฟรมข้อมูล	21
รูปที่ 2.15 ลักษณะของ Ethernet II Frame	21
รูปที่ 2.16 ลักษณะส่วนการทำงานภายในของ Preamble	22
รูปที่ 2.17 การสื่อสารด้วยเครือข่าย VoIP	27
รูปที่ 2.18 การใช้งาน VoIP แบบ PC-to-PC	28
รูปที่ 2.19 การใช้งาน VoIP แบบ PC-to-Phone	28
รูปที่ 2.20 การใช้งาน VoIP แบบ Phone-to-Phone	29
รูปที่ 2.21 กระบวนการทำงานของ VoIP	29
รูปที่ 2.22 การใช้เครือข่าย IP ร่วมกับ VoIP เพื่อลดต้นทุนวงจรเชื่อมต่อระหว่างองค์กร	30
รูปที่ 2.23 แกนกลางของซีพียู ARM7	31
รูปที่ 2.24 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ LPC2148	34
รูปที่ 2.25 การจัดการหน่วยความจำของ LPC2148	36
รูปที่ 2.26 การจัดการของ LPC2148	37
รูปที่ 2.27 การจัดการมาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x	38
รูปที่ 2.28 มัลติเทิลทาสก์ (Multiple Task)	41
รูปที่ 2.29 สถานะของทาสก์	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 2.30 ค่าตัวอย่างสัญญาณที่ซีกค่าออกมาได้ จะถูกประมาณด้วยค่าระดับคิสกรีตที่ใกล้ที่สุด	44
รูปที่ 2.31 ตัวอย่างการแบ่งระดับการควอนไทซ์สัญญาณ	45
รูปที่ 3.1 Block Diagram	46
รูปที่ 3.2 วงจรแปลงแรงดันไฟจาก +15V เป็น +5V	46
รูปที่ 3.3 วงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V	47
รูปที่ 3.4 วงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V	47
รูปที่ 3.5 วงจรPre-microphone และ Instrument Amplifier	48
รูปที่ 3.6 วงจร Non-Inverting Summing Amplifier	49
รูปที่ 3.7 วงจร Voltage follower (Buffer)	49
รูปที่ 3.8 วงจร Second order Low Pass Filter	50
รูปที่ 3.9 วงจร Microphone Pre-Amplifier	50
รูปที่ 3.10 วงจร Amplifier with gain =20	51
รูปที่ 3.11 วงจร Ethernet Module	52
รูปที่ 3.12 วงจร ไมโครคอนโทรลเลอร์ ARM7 ควบคุม Ethernet module	53
รูปที่ 3.13 วงจร ไมโครคอนโทรลเลอร์ MCS51	54
รูปที่ 3.14 วงจรคีย์แพด (Keypad)	54
รูปที่ 3.15 วงจร LCD	55
รูปที่ 3.16 แผนผังแสดงการทำงานของโปรแกรม Main	55
รูปที่ 3.17 แผนผังแสดงการทำงานของโปรแกรม Task0: Signaling (Priority=15)	56
รูปที่ 3.18 แผนผังแสดงการทำงานของโปรแกรม Task1: ADC (Priority=10)	57
รูปที่ 3.19 แผนผังแสดงการทำงานของโปรแกรม Task2: Ethernet Send (Priority=5)	57
รูปที่ 3.20 แผนผังแสดงการทำงานของโปรแกรม Task3: Ethernet Receive (Priority=5)	58
รูปที่ 3.21 แผนผังแสดงการทำงานของโปรแกรม Task4: DAC (Priority=7)	58
รูปที่ 3.22 แผนผังแสดงการทำงานของโปรแกรม Task5: Ending (Hang up, Priority=20)	59
รูปที่ 3.23 แผนผังแสดงการทำงานของโปรแกรม Task6: Ending (Receive BYE, Priority=20)	60
รูปที่ 3.24 แผนผังแสดงการทำงานของโปรแกรม Task7: Sound Signaling (Priority=5)	61
รูปที่ 4.1 ผลการทดลองวงจรแปลงแรงดันไฟจาก +15V เป็น +5V	62
รูปที่ 4.2 ผลการทดลองวงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V	63
รูปที่ 4.3 ผลการทดลองวงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V	64

สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 4.4 วงจรประกอบของวงจร Microphone Pre-Amplifier	65
รูปที่ 4.5 ผลการทดลองวงจร Pre-microphone และ Instrument Amplifier โดยป้อนสัญญาณเสียง	66
รูปที่ 4.6 ผลการทดลองวงจร Pre-microphone และ Instrument Amplifier โดยป้อนสัญญาณเสียง	66
รูปที่ 4.7 ผลการทดลองวงจร Pre-microphone และ Instrument Amplifier โดยป้อนสัญญาณ sine wave	67
รูปที่ 4.8 ผลการทดลองวงจร Voltage follower I (Buffer I) โดยป้อนสัญญาณ sine wave	67
รูปที่ 4.9 ผลการทดลองวงจร Voltage follower I (Buffer I) โดยป้อนสัญญาณเสียง	68
รูปที่ 4.10 ผลการทดลองของวงจร Non-inverting Summing Amplifier โดยป้อนสัญญาณ sine wave	68
รูปที่ 4.11 ผลการทดลองของวงจร Non-inverting Summing Amplifier	69
รูปที่ 4.12 ผลการทดลองของวงจร Voltage follower II (Buffer II) โดยป้อนสัญญาณ sine wave	69
รูปที่ 4.13 ผลการทดลองของวงจร Voltage follower II (Buffer II) โดยป้อนสัญญาณเสียง	70
รูปที่ 4.14 กราฟลอการิทึมของวงจร Second order Low Pass Filter	71
รูปที่ 4.15 ผลการทดลองของวงจร Second order Low Pass Filter โดยป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz	72
รูปที่ 4.16 ผลการทดลองของวงจร Second order Low Pass Filter โดยป้อนสัญญาณ sine wave ที่ความถี่ 4 kHz	73
รูปที่ 4.17 ผลการทดลองของวงจร Second order Low Pass Filter โดยป้อนสัญญาณเสียง	73
รูปที่ 4.18 ผลการทดลองของวงจร Voltage follower III (Buffer III) โดยป้อนสัญญาณ sine wave	74
รูปที่ 4.19 ผลการทดลองของวงจร Voltage follower III (Buffer III) โดยป้อนสัญญาณเสียง	74
รูปที่ 4.20 ผลการทดลองของวงจร Microphone Pre-Amplifier เมื่อป้อนสัญญาณเสียง	75
รูปที่ 4.21 ผลการทดลองของวงจร Power Amplifier with gain=20 เมื่อป้อนสัญญาณเสียง	76
รูปที่ 4.22 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz	77
รูปที่ 4.23 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz	77
รูปที่ 4.24 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 2 kHz	78
รูปที่ 4.25 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 2 kHz	78

สารบัญรูปภาพ(ต่อ)

	หน้า
รูปที่ 4.26 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 4 kHz	79
รูปที่ 4.27 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 4 kHz	79
รูปที่ 4.28 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณเสียง	80
รูปที่ 4.29 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณเสียง	80
รูปที่ 4.30 รูปแบบสัญญาณเสียง Ring Tone ที่สร้างจาก โปรแกรม MATLAB	81
รูปที่ 4.31 รูปแบบสัญญาณเสียง Ring Tone ที่วัดได้	82
รูปที่ 4.32 รูปแบบสัญญาณเสียง Ring Back Tone ที่สร้างจาก โปรแกรม MATLAB	82
รูปที่ 4.33 รูปแบบสัญญาณเสียง Ring Back Tone ที่วัดได้	83
รูปที่ 4.34 รูปแบบสัญญาณเสียง Busy Tone ที่สร้างจาก โปรแกรม MATLAB	83
รูปที่ 4.35 รูปแบบสัญญาณเสียง Busy Tone ที่วัดได้	84
รูปที่ 4.36 การดักจับแพคเกจจาก Ethernet module	84
รูปที่ 4.37 รายละเอียดของเฟรมข้อมูลที่รับเข้า	85
รูปที่ 4.38 รายละเอียดของการ Encapsulation ของ Layer 2	85
รูปที่ 4.39 รายละเอียดของการ Encapsulation ของ Layer 3	86
รูปที่ 4.40 รายละเอียดของการ Encapsulation ของ Layer 4	86
รูปที่ 4.41 เลขเฟรมข้อมูลที่เวลา 1 วินาที โดยมีค่า 455 เฟรม/วินาที	87
รูปที่ 4.42 ผลการดักจับเฟรมข้อมูล (ก)	88
รูปที่ 4.43 ผลการดักจับเฟรมข้อมูล (ข)	88
รูปที่ 4.44 ผลการดักจับเฟรมข้อมูล (ค)	89
รูปที่ 4.45 ผลการดักจับเฟรมข้อมูล (ง)	89
รูปที่ 4.46 กราฟแสดงข้อมูลของสัญญาณ sine ความถี่ 500 Hz ที่ดักจับได้	91
รูปที่ 4.47 ผลหน้าจอ LCD ในโหมดปกติ	92
รูปที่ 4.48 ผลหน้าจอ LCD ในโหมดทำการติดต่อโดยกำหนดหมายเลขปลายทาง	92
รูปที่ 4.49 ผลหน้าจอ LCD ในโหมดทำการติดต่อปลายทางไม่สำเร็จ	92
รูปที่ 4.50 ผลหน้าจอ LCD ในโหมดทำการติดต่อของทางฝั่งส่ง	93
รูปที่ 4.51 ผลหน้าจอ LCD ในโหมดได้รับการติดต่อของทางฝั่งรับ	93
รูปที่ 4.52 ชิ้นงาน IP Phone	94

สารบัญตาราง

หน้า

ตารางที่ 2.1 รายละเอียด UDP Header	10
ตารางที่ 2.2 UDP Port Numbers	10
ตารางที่ 2.3 ค่าบิตต่างๆใน IP Header	15
ตารางที่ 2.4 ค่าต่างๆของ IP Protocol Field	16
ตารางที่ 2.5 ตัวอย่างของ Source Address ที่แสดงรหัสแอดเดรสของผู้ผลิต	24
ตารางที่ 2.6 ตัวอย่างของรหัสที่ใช้แสดงแทนโปรโตคอลที่ใช้ในช่อง Type 18 รายการ	25
ตารางที่ 2.7 หน้าที่ของขาพอร์ต 3	39
ตารางที่ 4.1 ผลการทดลองการวัดขนาดของสัญญาณ sine wave ที่ความถี่ต่างๆ	70
ตารางที่ 4.2 การแปลงค่าข้อมูลที่จับได้จากฐานสิบหกเป็นฐานสิบ	90



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ปัจจุบันนี้การสื่อสารมีส่วนสำคัญในชีวิตประจำวันของมนุษย์เรามาก ซึ่งจะพบว่าในชีวิตประจำวันของเรา วันหนึ่งๆ ไม่สามารถหลีกเลี่ยงการติดต่อสื่อสารได้เลย และปัจจุบันโทรศัพท์มือถือถือว่าเป็นระบบสื่อสารขั้นพื้นฐานที่สำคัญที่สุดในชีวิตประจำวัน ไม่ว่าจะตามบ้านเรือน บริษัท สำนักงานทั่วไป ระบบการสื่อสารที่สำคัญอีกระบบหนึ่งที่มีความจำเป็นต่อบริษัทสำนักงานหรือแม้แต่ตามที่พักอาศัยที่มีเครื่องคอมพิวเตอร์หลายเครื่องนั้น ก็คือระบบ LAN (Local Area Network)

Ethernet เป็นเทคโนโลยีสำหรับเครือข่ายระบบแบบแลน (LAN) ที่ได้รับความนิยมสูงสุดในปัจจุบันโดยเป็นมาตรฐานการส่งข้อมูลที่อนุญาตให้เครื่องคอมพิวเตอร์ใช้ช่องสัญญาณร่วมกัน นั่นคือเมื่อตรวจสอบแล้วว่าในขณะนั้นไม่มีเครื่องอื่น ๆ กำลังส่งข้อมูล แต่ละอย่างเครื่องจะทำการส่งข้อมูลออกมา โดยเครื่องใดที่ส่งข้อมูลออกจะมีหน้าที่เฝ้าดูว่ามีเครื่องอื่นทำการส่งข้อมูลออกไปพร้อมกันด้วยหรือไม่ เพราะถ้าเกิดการส่งพร้อมกันแล้ว จะก่อให้เกิดการชนกันของข้อมูล แต่ถ้าตรวจจับได้ว่ามีการชนกันขึ้นก็จะหยุดส่งแล้วรอคอยเป็นระยะเวลาสั้น ๆ ก่อนจะทำการส่งข้อมูลออกไปอีกครั้งหนึ่ง เวลาที่ใช้ในการรอคอยนั้นเป็นค่าที่สุ่มขึ้นมา ซึ่งมีความสั้นยาวต่างกันไป ซึ่งเมื่อคุณมีคอมพิวเตอร์หลายเครื่องในเครือข่ายเดียวกันนั้นเหมือนกัน ระบบเครือข่าย Ethernet จะสามารถช่วยเพิ่มประสิทธิภาพการทำงานของคอมพิวเตอร์เหล่านั้นได้ โดยสามารถถ่ายโอนข้อมูลระหว่างเครื่องคอมพิวเตอร์ และเครื่อง Server ได้รวดเร็วขึ้นมาก อีกทั้งยังสามารถส่งพิมพ์งาน ผ่านพรินเตอร์ หรือใช้โปรแกรมต่าง ๆ รวมทั้งระบบ ต่อเชื่อมอินเทอร์เน็ต ร่วมกัน ระหว่าง คอมพิวเตอร์ทุกเครื่อง ในเครือข่ายนั้นด้วย

จากอัตราการเจริญเติบโตของเครือข่ายระบบการสื่อสารในปัจจุบันนั้นพบว่า จำนวนของอุปกรณ์และเครื่องใช้ต่างๆ ที่ต่อพ่วงเข้ากับระบบเครือข่ายนั้นมีการเจริญเติบโตอย่างรวดเร็ว ทำให้ในอุปกรณ์การสื่อสารและอุปกรณ์ควบคุมต่าง ๆ นั้น ได้มีการใช้ระบบควบคุมแบบอัตโนมัติมากขึ้น ซึ่งอุปกรณ์เหล่านี้มักอยู่ในรูปของระบบควบคุมแบบฝังตัว (Embedded Systems)

Embedded Systems เป็นระบบอิเล็กทรอนิกส์ที่ใช้สำหรับงานควบคุมรวมถึงการแสดงผลการทำงานต่าง ๆ โดยที่ระบบเหล่านี้ถูกใช้เป็นส่วนหนึ่งของระบบและอุปกรณ์ควบคุม เครื่องมือ เครื่องจักรต่าง ๆ ระบบแบบฝังตัว (Embedded System) นี้แม้ไม่ใช่เครื่องคอมพิวเตอร์ แต่ก็มียระบบคอมพิวเตอร์อยู่ภายใน อาจจะเป็นเพียงไมโครโพรเซสเซอร์ (Microprocessor) หรือชิป (chip) ธรรมดาหรือโพรเซสเซอร์ (Processor) ที่ประกอบด้วย ชิป (Chip) ที่มีวงจรซับซ้อน โดยจะมีหลักการทำงาน คือ มีสัญญาณข้อมูลเข้า (Input) จากอุปกรณ์ เซ็นเซอร์ (Sensor) เข้าสู่ระบบ และมีสัญญาณผลลัพธ์ (Output) ของระบบไปควบคุมบังคับสวิตซ์เครื่องควบคุมต่าง ๆ

เนื่องจากระบบควบคุมแบบฝังตัว เป็นระบบที่เหมาะสมในการควบคุมงานเฉพาะทาง แต่ระบบเหล่านี้มักจะเป็นระบบปิด ทำให้การควบคุมและใช้งานต้องกระทำผ่านเครื่องมือ หรืออุปกรณ์เฉพาะของผู้ออกแบบหรือผู้ผลิตแต่ละรายเท่านั้น แต่อย่างไรก็ตาม การควบคุมจำเป็นจะต้องมีซอฟต์แวร์พิเศษ ซึ่งมักยึดติดกับระบบนั้นหรือระบบปฏิบัติการที่ใช้ควบคุม

สำหรับปริญญาโทฉบับนี้ได้้นำการสื่อสารทางโทรศัพท์และรูปแบบการสื่อสารแบบ Ethernet มาประยุกต์ร่วมกับระบบควบคุมแบบฝัง (Embedded Systems) ซึ่งจะศึกษาและพัฒนาระบบโทรศัพท์บนโครงข่าย IP ซึ่งเป็นเทคโนโลยี VOIP ที่แปลงเสียงพูดให้อยู่ในรูปแบบที่สามารถส่งผ่านโครงข่าย IP ซึ่งระบบโทรศัพท์ผ่านโครงข่าย IP นี้สามารถส่งผ่านข้อมูลได้อย่างรวดเร็ว ทั้งยังประหยัดค่าใช้จ่ายในการวางโครงข่ายโทรศัพท์และยังช่วยลดค่าใช้จ่ายในการสนทนาภายในองค์กรได้อีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการ

2.1 โพรโทคอล TCP/IP

2.1.1 จุดประสงค์ของการสื่อสารตามมาตรฐาน ของโปรโตคอล TCP/IP

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

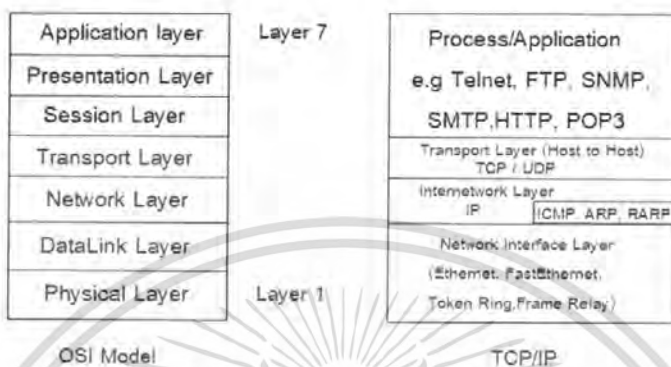
2.1.2 โครงสร้างของโปรโตคอล TCP/IP

โปรโตคอล TCP/IP มีการจัดกลไกการทำงานเป็นชั้นหรือ Layer เรียงต่อกันซึ่งในแต่ละ Layer ของโปรโตคอล TCP/IP จะประกอบด้วย

- Application Layer
เป็นเลขอร์ที่แอปพลิเคชันเรียกโปรโตคอลระดับล่างๆลงไปเพื่อให้บริการต่างๆเช่น FTP,SMTP, Telnet , HTTP , POP
- Transport Layer
รับผิดชอบการรับส่งข้อมูลระหว่างปลายด้านส่งและด้านรับข้อมูล และส่งข้อมูลขึ้นไปให้ Application Layer นำไปใช้งาน
- Internetwork Layer
ทำหน้าที่รับข้อมูลจากชั้นTransport Layerและค้นหาและเลือกเส้นทางระหว่างผู้รับและผู้ส่ง
- Network Access Layer
เลขอร์นี้มีหน้าที่ควบคุมการรับส่งข้อมูลในระดับฮาร์ดแวร์ของเครือข่าย รับผิดชอบการรับส่งข้อมูลในระดับกายภาพ จนถึงการแปลความจากสัญญาณไฟฟ้าเป็นข้อมูลทางคอมพิวเตอร์

โดยในแต่ละ Layer จะมีการทำงานเทียบได้กับ OSI model มาตรฐาน แต่บาง Layer ของโปรโตคอล TCP/IP จะทำงานเทียบกับ OSI หลาย Layer ปนกัน ซึ่งสามารถแสดงได้ดังที่รูปที่ 2.1 ซึ่งเราจะเห็นว่าบางกลไกของโปรโตคอล TCP/IP เทียบได้กับมาตรฐาน OSI model สองชั้น หรือบางกลไกก็จะทำงานคาบเกี่ยวกันระหว่าง

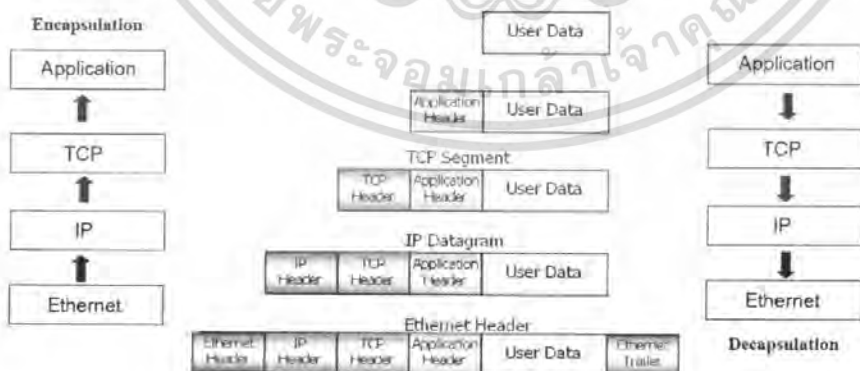
บางชั้นของ OSI model ตัวอย่างเช่น กลไกการทำงานของโพรโทคอล TCP/IP ในส่วน Network Interface Layer เมื่อเทียบกับมาตรฐาน OSI model จะเทียบได้กับ Data Link Layer และ Physical Layer 2 ชั้นรวมกัน เป็นต้น



รูปที่ 2.1 กลไกของโพรโทคอลมาตรฐาน OSI-model

2.1.3 Encapsulation และ Decapsulation

เมื่อทำการส่งข้อมูล ลักษณะการทำงานของโมเดลนี้คือ ข้อมูลจะถูกส่งลงมาจากเลเยอร์ชั้นข้างบนลงมายังเลเยอร์ชั้นข้างล่างสุด เมื่อข้อมูลถูกส่งผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่รับมา กับส่วนควบคุมซึ่งอยู่ส่วนหัวของข้อมูลเรียกว่า Header ซึ่งในแต่ละชั้นนั้น Header จะมีรูปแบบที่สำคัญเป็นของตัวเอง การเพิ่ม Header เข้าไปกับข้อมูลนั้นเราเรียกว่า Encapsulation และเมื่อผู้รับได้รับข้อมูลก็จะเกิดกระบวนการทำงานย้อนกลับคือ ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Decapsulation ดังขั้นตอนแสดงในรูปที่ 2.2

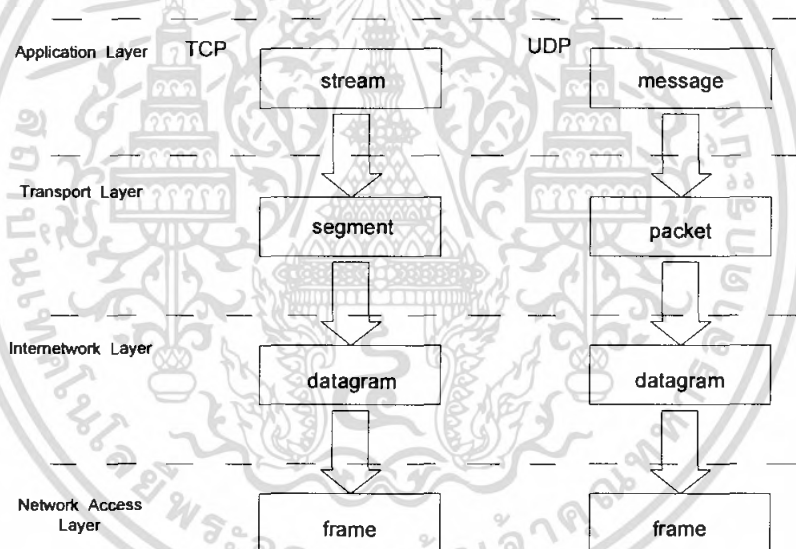


รูปที่ 2.2 ขั้นตอนการ Encapsulation และ Decapsulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

ข้อมูลที่จะผ่านการ Encapsulate ในแต่ละระดับมีชื่อเรียกแตกต่างกัน ข้อมูลที่มาจาก User หรือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า User Data และเมื่อ Application ได้รับข้อมูลจาก user ก็จะนำมาประกอบกับ ส่วนหัวของ Application เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมาพร้อมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยัง โปรโตคอล IP เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมาพร้อมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Network access Layer ในระดับ Network access Layer จะนำ IP Datagram มา เพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่าน สายสัญญาณที่เชื่อมโยงอยู่ต่อไป

จะเห็นว่าหน่วยของข้อมูลที่จะทำการส่งนั้นจะมีชื่อเรียกต่างกันเมื่อมันเดินทางผ่านแต่ละ Layer และ ในการส่งใน TCP/IP นั้นมีอยู่ 2 โปรโตคอลย่อยคือ TCP กับ UDP ชื่อของข้อมูลที่จะผ่านโปรโตคอลทั้งสองนั้น แตกต่างกันในชั้นบนๆ เท่านั้นและชื่อสำคัญๆ ที่เราจะต้องพบและใช้ต่อไปบ่อยๆ ได้แก่ datagram และ frame ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 Data Structures

2.1.4 Application Layer

ในชั้นนี้จะรองรับการทำงานของ Application ต่างๆ โดยโปรโตคอลหลักๆ ที่ทำงานใน Application Layer ซึ่งผู้ใช้มักจะคุ้นเคยกันดีได้แก่

- FTP (File Transfer Protocol) ซึ่งเป็นโปรโตคอลสำหรับการจัดการเพิ่มข้อมูลโดยใช้ในการรับ-ส่ง เพิ่มข้อมูลระหว่างเครื่องลูกข่ายและเครื่อง Server
- Telnet เป็นโปรโตคอลสร้างจอเทอร์มินัลเสมือน สำหรับการควบคุมเครื่องระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- HTTP(Hyper Text Transfer Protocol) เป็นโปรโตคอลใช้ในการติดต่อรับส่งข้อมูลชนิดไฮเปอร์ เท็กซ์ (Hypertext) ระหว่างเครื่องลูกข่ายกับ WWW Server (World Wide Web) โดยที่เอกสารนี้จะอยู่ในรูปแบบที่เขียนในภาษา HTML (Hyper Text Markup Language) เอกสารแต่ละชิ้น จะสามารถเชื่อมโยงไปยังเอกสารชิ้นอื่นได้

- SMTP (SimpleMail Transfer protocol) เป็นโปรโตคอลสำหรับการให้บริการด้านจดหมายอิเล็กทรอนิกส์

นอกจากนี้ยังมีโปรโตคอลอื่นที่อยู่เบื้องหลัง ซึ่งทำงานโดยที่ผู้ใช้ไม่ได้มีการใช้งานโดยตรง เช่น

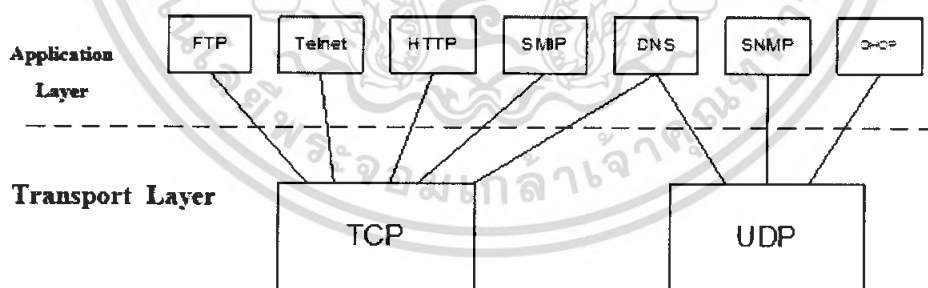
- โปรโตคอล DNS (Domain Name System) ที่ทำหน้าที่แปลงข้อมูลชื่อ domain name หรือชื่อเว็บไซต์ทั้งหลายให้เป็นหมายเลข IP address

- โปรโตคอล DHCP (Dynamic Host Configuration Protocol) ทำหน้าที่แจกจ่ายข้อมูลพารามิเตอร์ของเครื่องข่ายให้กับเครื่องลูกข่ายที่เชื่อมต่ออยู่

ซึ่งในการทำงานของชั้น Application Layer นี้รองรับให้โปรโตคอลอื่นทำงานได้หลายโปรเซสและหลายโปรโตคอลได้พร้อมกันนั้น ทำให้ผู้ใช้สามารถเปิดโปรแกรมใช้งานได้หลายๆ โปรแกรมพร้อมกันได้

2.1.5 Transport Layer

การทำงานของ Transport Layer นี้จะมีการสร้าง connection หรือการเชื่อมต่อกันระหว่างApplicationกับ Transport Layer โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลที่เรียกว่า port หรือ socket (คำว่า port ในที่นี้ไม่ได้หมายถึง port ทางฮาร์ดแวร์) และในแต่ละแอปพลิเคชันก็จะสร้างการเชื่อมต่อผ่าน port ได้พร้อมกันหลายApplication ซึ่งการใช้งาน port ของแต่ละ Application ที่อยู่ในชั้น Application Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้ และแต่ละโปรโตคอลจะมีการใช้งาน port หมายเลขต่าง ๆ ไม่ซ้ำกัน ดังรูปที่ 2.4



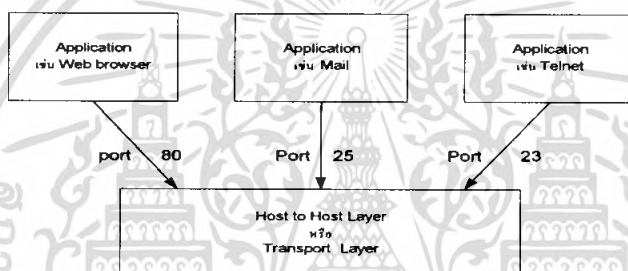
รูปที่ 2.4 การใช้งาน port ของแต่ละโปรโตคอล

ตัวโปรโตคอล TCP และโปรโตคอล UDP จะมีแอปพลิเคชันเฉพาะเพื่อเรียกใช้งานแยกกันคือ แอปพลิเคชันที่ใช้โปรโตคอล FTP, Internet, HTTP และ SMTP จะมีการส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล TCP ส่วนแอป

พลิเคชันที่ใช้โปรโตคอล SNMP และ DHCP จะส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล UDP และสำหรับโปรโตคอล DNS นั้น จะสามารถเรียกใช้งานได้ทั้ง TCP และ UDP

การที่มีการกำหนด port ต่างๆสำหรับการส่งผ่านข้อมูลจาก Application Layer มายังชั้น Transport Layer นั้น ทำให้การรับส่งข้อมูลในแต่ละโปรโตคอลทำได้ถูกต้อง ถึงแม้ว่าในเครื่องเซิร์ฟเวอร์ที่ให้บริการจะมีการทำงานอยู่หลายโปรเซสที่แตกต่างกัน หรือมีผู้ใช้บริการเข้ามาใช้งานพร้อมกันจำนวนมาก และหลาย Application ในเวลาเดียวกัน

ในชั้น Transport Layer ของ TCP/IP มีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกันคือ TCP (Transmission Control Protocol) และ UDP (User Datagram Protocol) ในการส่งผ่านข้อมูลลงไปชั้นถัด ๆ ไป เราจะเห็นว่าโปรโตคอล TCP และ UDP จะถูกผนึกเข้าไปในโปรโตคอล IP อีกทีหนึ่งและส่งต่อไปยังเครือข่ายอินเทอร์เน็ตต่อไป



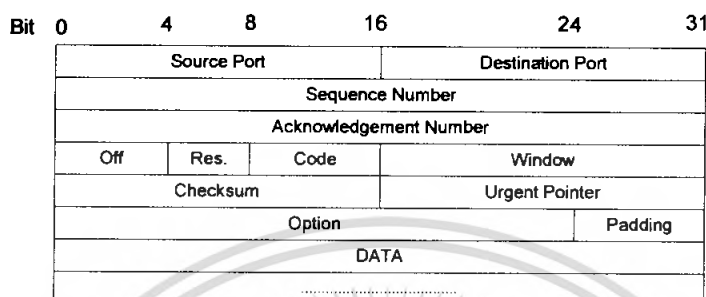
รูปที่ 2.5 การส่งข้อมูลจาก Application ไปยัง Transport Layer

2.1.5.1 โปรโตคอล TCP (Transmission Control Protocol)

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า คือมีความสัมพันธ์ต่อเนื่องกัน มีกลไกในการตรวจสอบทั้งด้านส่ง และด้านรับ เพื่อให้แน่ใจว่าสามารถสื่อสารกันได้จริง จึงจะมีการส่งรับข้อมูลเกิดขึ้น (connection – oriented) ตลอดจนการยกเลิกการติดต่อก็มีกลไกสำหรับแจ้งให้อีกฝั่งทราบ

ข้อมูลที่จะส่งผ่าน TCP จะถูกนำมาแตกย่อยออกเป็นส่วนย่อยๆก่อน แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล โดยข้อมูลแต่ละชุดที่แบ่งออกและ ทำการส่งโดย TCP แต่ละครั้งจะเรียกว่า TCP เซกเมนต์ ในระหว่างการรับส่งข้อมูลนี้ โปรโตคอล TCP จะเพิ่มขบวนการสอบทานข้อมูล เพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgment) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น โปรโตคอล TCP มีกลไกควบคุมการไหลของข้อมูล (Flow Control) โดยการควบคุมนี้จะต้องอาศัยลำดับของการรับส่งที่ถูกต้อง และสัมพันธ์กันทั้ง 2 ฝั่งสำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูล datagram

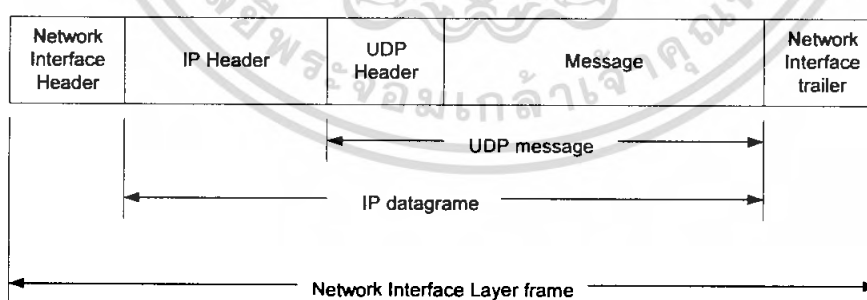
ดังนั้น Application หรือ process ใดที่อาศัยการส่งผ่านข้อมูลด้วยโปรโตคอล TCP จะมีความน่าเชื่อถือของการส่งผ่านข้อมูลมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบในการทำงานเช่นหน่วยความจำและขนาดของช่องสัญญาณ (bandwidth) ที่มากกว่าเช่นกัน



รูปที่ 2.6 โครงสร้างของโปรโตคอล TCP

2.1.5.2 โปรโตคอล UDP (User Datagram Protocol)

UDP เป็นโปรโตคอลที่ถูกออกแบบมาให้ทำหน้าที่รับส่งข้อมูลโดยมีขั้นตอนการทำงานไม่ซับซ้อนและทำงานได้รวดเร็ว โดยรูปแบบการส่งข้อมูลของ UDP ในการส่งข้อมูลแต่ละครั้งจะไม่มีกรส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล การติดต่อของโปรโตคอล UDP นั้นไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่างด้านส่งและด้านรับ ด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้ง และไม่มีกรส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้ Application หรือ Process ใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง



รูปที่ 2.7 UDP Message encapsulation

UDP Header

16-bit Source port	16-bit destination port
length	checksum
data	

รูปที่ 2.8 โครงสร้างของ UDP Header

- **Source Port** มี 16 บิตใช้ระบุเป็น Source Application Layer Protocol ทำการส่ง UDP Message โดย Source Port เป็น port ที่ใช้ในการเลือก เมื่อใดที่ไม่ได้ใช้มัน มันจะตั้งค่าเป็น 0x00-00 IP multicast traffic เปรียบเสมือน videocasts ใช้ส่ง UDP สามารถใช้ค่า 0x00-00 เพราะจะไม่ตอบรับ video traffic เป็นเพียงการสมมุติ Application Layer ใช้ Source Port ในการนำ UDP Message เข้ามา Destination Port สำหรับการตอบรับ
- **Destination Port** มี 16 บิตใช้ระบุเป็น Destination Application Layer Protocol การรวมของ Destination IP Address ของ IP Header และ Destination Port ของ UDP Header จะไม่เหมือนใครสำหรับกระบวนการที่จะส่งข้อมูล
- **Length** มี 16 บิตที่ใช้ในการแสดงความยาวใน UDP Message มีความยาวน้อยที่สุด 8 ไบต์ (ขนาดของ UDP Header) และมากที่สุด 65,515 ไบต์ (ค่าสูงสุด IP Datagram 65,535 ไบต์ น้อยกว่าค่าน้อยที่สุด IP Header 20 ไบต์) ความยาวมากที่สุดที่แท้จริงถูกจำกัดโดย MTU ซึ่งจะทำการเชื่อมโยงโดย UDP Message เป็นตัวส่ง ความยาว UDP สามารถคำนวณได้จากความยาวทั้งหมดและความยาวของ IP Header field ใน IP Header
- **Checksum** มี 16 บิตโดยจะทำการตรวจระดับของบิตอย่างสมบูรณ์สำหรับ UDP Message โดยที่ UDP Checksum คำนวณโดยใช้วิธีเดียวกับกับ IP Header Checksum

ตารางที่ 2.1 รายละเอียด UDP Header

ตำแหน่ง	ชื่อ	อธิบาย
บิต 0-15	Source port number	หมายเลขพอร์ตต้นทางที่ส่งค่าไค้แกรมนี้ มีความยาว 16 บิต
บิต 16-31	destination port number	หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับค่าไค้แกรม มีความยาว 16 บิต เช่นกัน
บิต 32-47	UDP length	ความยาวของค่าไค้แกรม ทั้งส่วน Header และ data นั้นหมายความว่าค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
บิต 48-63	Checksum	เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

UDP Port

UDP Port จะแสดงที่ตั้งหรือแถวของ message ที่ชัดเจนสำหรับการส่ง message ถึง Application Layer protocol โดยใช้ UDP services รวมถึงในแต่ละตัวของ UDP message เป็น Source Port และ Destination Port ซึ่ง Internet Assigned Number Authority (IANA) จะเป็นตัวกำหนดหมายเลข Port

ตารางที่ 2.2 UDP Port Numbers

Port Numbers	Application Layer Protocol
53	Domain Name System (DNS)
67	BOOTP client (Dynamic Host Configuration Protocol [DHCP])
68	BOOTP server (DHCP)
69	Trivial File Transfer Protocol (TFTP)
137	NetBIOS Name Service
138	NetBIOS Datagram Service
161	Simple Network Management Protocol (SNMP)
520	Routing Information Protocol (RIP)
445	Direct hosting of server Message Block (SMB) datagram over TCP/IP
1812 , 1813	Remote Authentication Dial-In User Service (RADIUS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UDP Checksum

Checksum เป็นเลข 16 บิตถูกคำนวณด้วยวิธี one's complement โดยนำ Pseudo Header และข้อมูลทั้งหมดใน UDP Datagram มาคำนวณ

Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ประกอบด้วยฟิลด์ source IP address , destination IP address , zero , protocol , UDP length ดังแสดงในรูปที่ 2.9

16-bit Source IP address		
16-bit Destination IP address		
zero	8-bit protocol (17 for UDP)	16-bit length

รูปที่ 2.9 Pseudo Header

หากค่า Checksum ที่คำนวณออกมาเป็น 0 ค่า checksum จะถูกเซตเป็น 1 ทั้งหมดแทน (มีค่าเท่ากับในระบบ 1's complement) ทั้งนี้เพราะในบาง Application ที่ไม่ต้องการตรวจสอบค่า checksum ในระดับ UDP จะเซตค่านี้เป็น 0 (disable checksum)

2.1.6 Internetwork Layer

ในระดับล่างต่อมาในชั้น Internetwork Layer มีหน้าที่ส่งผ่านข้อมูลในระหว่างเครือข่าย โดยมีโพรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ บนอินเทอร์เน็ต คือ โพรโตคอล IP (Internet Protocol) นอกจากนี้ในชั้น Internetwork Layer ยังมีโพรโตคอลทำงานอยู่ด้วยอีก 2 ชนิดคือ โพรโตคอล Internet Control Message Protocol (ICMP) ซึ่งเป็นโพรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของ datagram ให้แก่ IP และโพรโตคอล Address Resolution Protocol (ARP) ซึ่งทำหน้าที่เปลี่ยนระหว่าง IP Address ให้เป็น MAC Address แต่ที่ใช้ในการทำโครงการนี้คือ โพรโตคอล IP

2.1.6.1 โพรโตคอล IP (Internet Protocol)

โพรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจก Transport Layer เพื่อส่งข้ามไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง แม้ว่าเครือข่ายนั้นอาจจะอยู่คนละเครือข่ายกันก็ตาม โดยโพรโตคอล IP นั้นมีการทำงานแบบ packet switching คือมีการจัดการเกี่ยวกับ Address และข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของ packet และมีการส่งข้อมูลผ่านสวิทช์ (switch) ไปยังปลายทาง โดยข้อมูลจะเดินทางไปยังเครือข่ายต่าง ๆ ผ่านสวิทช์นี้ไปเรื่อยๆ จนกว่าจะถึงปลายทาง ตัววงจรผ่านหรือ switch นี้อาจเป็น gateway หรือ router ในระบบเครือข่ายก็ได้ ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุด และ

สามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และในข้อมูลของโพรโทคอล IP จะมีข้อมูลของหมายเลข IP ที่จะส่งข้อมูลไปและเมื่อถึงเครือข่ายปลายทางแล้ว จะมีกลไกแปลงหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์ประจำเครื่อง (MAC Address) ที่ถูกต้องอีกทีหนึ่งด้วย โพรโทคอล ARP

IP Datagram

IP Datagram ประกอบด้วย IP Header และ IP Payload



รูปที่ 2.10 IP Datagram

- **IP Header** เป็นขนาดที่เปลี่ยนแปลงได้ระหว่าง 20 และ 60 ไบต์ ในการเพิ่มขึ้น 4 ไบต์ มันจะจัดเตรียมการสนับสนุน routing , การแสดงตัว payload , การชี้ให้เห็นถึงขนาด IP Header และ Datagram , การสนับสนุน fragmentation โดยมีโครงสร้างดังรูปที่ 2.11

4-bit Version	Header Length	8-bit Type of service	16-bit Total Length in Byte	
16-bit Identification		3-bit Flag	13-bit Fragment Checksum	
8-bit Time to Live(TTL)	8-bit Protocol		16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
32-bit Option and Padding				
Data				

รูปที่ 2.11 โครงสร้าง IP Header

Version

Version มีขนาด 4 บิต ใช้ในการแสดงว่า IP Header มีความยาวกี่ไบต์ โดยนำค่า IP Header Length คูณ 4 จะได้เท่ากับความยาวของ IP Header เช่นถ้า Offset เป็น 5 จะได้ความยาว IP Header เท่ากับ 20 ไบต์

Header Length

Header Length มีขนาด 4 บิต ใช้ในการแสดง IP Header โดยหมายเลขที่มากที่สุดที่ใช้แสดงให้เห็นว่ามี 4 บิต คือ 15 ดังนั้น Header Length ไม่มีทางที่จะเป็น Byte Counter ได้

Type of Service

Type of Service (TOS) มีขนาด 8 บิต ใช้ในการแสดงคุณสมบัติของ Service ที่ทำการส่ง Datagram โดยใช้ Internetwork Routers โดย TOS ประกอบด้วย Sub – Field และ Flags ที่ต้องการแสดงถึง precedence , delay , throughput , reliability , และ cost characteristics TOS จะทำการตั้งค่าโดย Sending Host และไม่สามารถแก้ไขได้โดย Routers

Total Length

Total Length มีขนาด 2 ไบต์ ใช้ในการแสดงขนาดของ IP Datagram (IP Header และ IP Payload) สำหรับขนาดสูงสุดของ Total Length จะเหมือนกับ IP MTU สำหรับ Network Interface Layer ระหว่าง Header Length กับ Total Length สามารถคำนวณหาค่า IP Payload ได้จาก

$$\text{IP Payload Length (ไบต์)} = \text{Total Length (ไบต์)} - 4 * \text{Header Length (32 bit words)}$$

Identification

Identification มีขนาด 2 ไบต์ ใช้ในการแสดงลักษณะเฉพาะในการส่ง IP Packet ระหว่าง Source กับ Destination Node โดย Sending Host จะเป็นตัวตั้งค่า Identification และจะทำการต่อเข้ากับ IP Datagram

Flags

Flags มีขนาด 3 บิต ประกอบด้วย 2 Flags สำหรับ Fragmentation โดย Flags ตัวแรกใช้แสดง IP Datagram ว่ามีลักษณะที่เหมาะสมสำหรับ Fragmentation หรือไม่ และ Flags อีกตัวหนึ่งใช้แสดงว่ามี Fragment ไปตาม Fragment IP Datagram หรือไม่

Fragment Offset

Fragment Offset มีขนาด 13 บิต ใช้ในการแสดง Offset ที่เป็น Fragment เริ่มต้นที่เกี่ยวข้องกับ IP Payload เดิม

Time To Live

Time To Live มีขนาด 1 ไบต์ ใช้ในการแสดงการเชื่อมต่อของ IP Datagram โดยสามารถที่จะเคลื่อนย้ายก่อนที่ IP Router จะทิ้งมันไป

Protocol

Protocol มีขนาด 1 ไบต์ ใช้ในการแสดง Protocol ใน Layer ที่สูงกว่าซึ่งอยู่ใน IP Payload ค่าที่ใช้โดยปกติใน IP Protocol เป็น 1 สำหรับ ICMP , 6 สำหรับ TCP และ 17 (0x11) สำหรับ UDP

Header Checksum

Header Checksum มีขนาด 2 ไบต์ จะแสดงการตรวจสอบ bit-level ที่สมบูรณ์ครบถ้วนบน IP Header เท่านั้น ไม่รวม IP Payload โดย Sending Host จะแสดง Checksum เริ่มต้นในการส่ง

Source Address

Source Address มีขนาด 4 ไบต์ ประกอบด้วย IP Address ของ Source Host โดย Network Address Translator (NAT) เป็นการเคลื่อนย้าย IP Datagram โดย NAT ใช้เคลื่อนย้ายระหว่าง Public กับ Private Address เมื่อทำการเชื่อมต่อกับอินเทอร์เน็ต

Destination Address

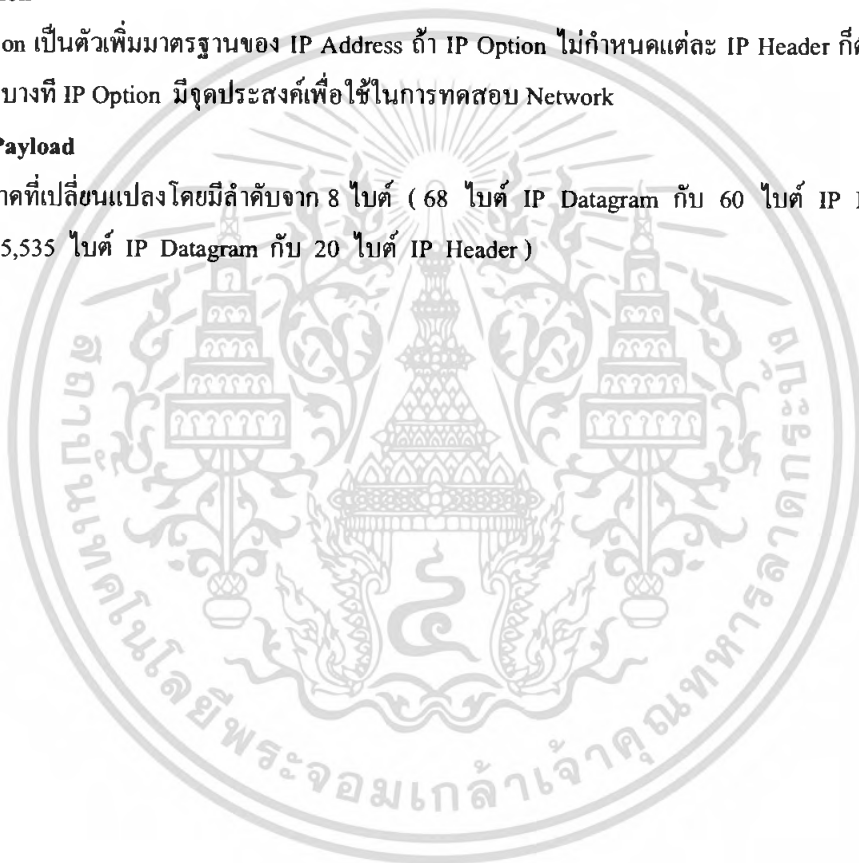
Destination Address ขนาด 4 ไบต์ ประกอบด้วย IP Address ของ Destination Host โดย IP Datagram จะเคลื่อนย้ายโดย NAT ฐานข้อมูลบน IP Source Routing สามารถค้นพบใน IP Option

IP Option

IP Option เป็นตัวเพิ่มมาตรฐานของ IP Address ถ้า IP Option ไม่กำหนดแต่ละ IP Header ก็ต้องกำหนด IP Option มาให้ บางที IP Option มีจุดประสงค์เพื่อใช้ในการทดสอบ Network

- IP Payload

เป็นขนาดที่เปลี่ยนแปลงโดยมีลำดับจาก 8 ไบต์ (68 ไบต์ IP Datagram กับ 60 ไบต์ IP Header) ถึง 65,515 ไบต์ (65,535 ไบต์ IP Datagram กับ 20 ไบต์ IP Header)



ตารางที่ 2.3 ค่าบิตต่างๆ ใน IP Header

ตำแหน่ง	ชื่อ	อธิบาย
0-3	Version	ขนาด 4 บิตเป็นเวอร์ชันของ IP ปัจจุบันค่านี้นักกำหนดให้เป็น 4
4-7	Lenght	มีขนาด 4 บิตเป็นค่าความยาวของ Header นี้ โดยปกติจะเป็น 5 หมายความว่า 5×32 บิต = 20 ไบต์
8-15	Type of Service	ใช้ในการแสดงคุณสมบัติของ Service ที่ทำการส่ง Datagram
16-31	Total length	เป็นฟิลด์ที่บอกจำนวนไบต์ทั้งหมดของ IP Datagram ด้วยขนาด 16 บิตทำให้ Datagram มีขนาดสูงสุดไม่เกิน 65535 ไบต์ และมีขนาดเล็กสุดไม่ต่ำกว่า 512
32-47	Identification	ใช้ในกรณีที่มีการแบ่งค่าไดาแกรมออกเป็นแฟรกเมนต์ เมื่อนำกลับมารวมกันใหม่จะได้รู้ว่ามาจากค่าไดาแกรมเดียวกัน
48-50	Flag	ใช้ในกรณีที่มีการแบ่งข้อมูลออกเป็นแฟรกเมนต์ มีความหมายดังนี้ บิต 0 (reserved) เป็น 0 เสมอ บิต 1 (DF) 0=May Fragment, 1=Don't Fragment บิต 2 (MF) 0=Last Fragment, 1=More Fragments
51-63	fragment offset	เป็นส่วนระบุข้อมูลที่ใช้แสรกวมข้อมูล เพื่อให้ข้อมูลที่ถูกแยกออกเป็นแฟรกเมนต์กลับมา รวมกันได้อย่างถูกต้องตามลำดับ
64-71	Time to Live (TTL)	เป็นจำนวนครั้งสูงสุดที่ค่าไดาแกรมนี้จะถูกส่งผ่านเครือข่ายไปยังปลายทางได้ เพื่อป้องกันไม่ให้ค่าไดาแกรมถูกเรตไปเรื่อยๆ อย่างไม่มีสิ้นสุด ปกติค่านี้จะเริ่มต้นที่ 32 และจะถูกลดค่าลงทีละ 1 เมื่อมีการเรต จนค่านี้มีค่าเป็น 0 ก็จะไม่ถูกเรตอีกต่อไป
72-79	Protocol	เป็นข้อมูลที่ระบุโปรโตคอลที่ส่งค่าไดาแกรมนี้มา ตัวอย่างโปรโตคอลที่ใช้บ่อยๆ ได้แก่ ICMP มีค่าในฟิลด์โปรโตคอล=1 TCP มีค่าในฟิลด์โปรโตคอล=6 UDP มีค่าในฟิลด์โปรโตคอล=17
80-95	Header Checksum	เป็นส่วนตรวจสอบความถูกต้องของข้อมูลใน Header โดยไม่เกี่ยวข้องกับส่วนข้อมูลที่อยู่ใน payload ค่านี้จะถูกคำนวณใหม่ทุกครั้งที่มีการเปลี่ยนแปลงข้อมูลใน Header (เช่น TTL ที่มีการเปลี่ยนแปลงทุกครั้งที่ IP datagram ถูกส่งผ่านเราเตอร์)
86-127	Source IP Address	คือ IP Address ของผู้ส่งค่าไดาแกรม
128-163	Destination IP Address	คือ IP Address ของผู้รับค่าไดาแกรม
ไม่แน่นอน	Option	มีขนาดข้อมูลไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน
ขึ้นอยู่กับ Option	Padding	มีข้อมูลว่างเปล่า ใช้เป็นส่วนเติมเต็มของฟิลด์ Option ให้ครบ 32 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Protocol

Protocol Field มีขนาดยาว 1 ไบต์ และใช้เป็นตัวบ่งบอกในการบรรจุข้อมูลไปยัง Layer ที่สูงขึ้น กับใน IP Payload และยังเป็นตัวบ่งบอกลูกข่ายโปรโตคอลที่ชัดเจน โดยปกติค่าของ IP Protocol Field จะเป็น 1 สำหรับ ICMP, 6 สำหรับ TCP และ 17 (0x11) สำหรับ UDP Protocol field จะแสดงตัวได้อย่างมากมาย ดังนั้น payload สามารถที่จะผ่านไปยัง Layer ที่สูงขึ้นไปได้ถูกต้อง โดยจะได้รับที่ destination

ตารางที่ 2.4 ค่าต่างๆของ IP Protocol Field

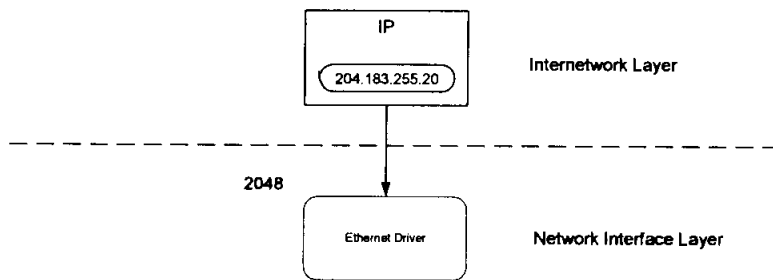
Value	Protocol
0	Reserved
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
4	IP in IP encapsulation
6	Transmission Control Protocol (TCP)
8	Exterior Gateway Protocol (EGP)
17	User Datagram Protocol (UDP)
46	Resource Reservation Protocol (RSVP)
47	Generic Routing Protocol (GRE)
50	IP Security Encapsulating Security Payload (ESP)
51	IP Security Authentication Header (AH)
89	Open Shortest Path First (OSPF)

การกำหนด IP address ให้กับอุปกรณ์

ต้องกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเข้ากับเครือข่ายทุกจุด จุดเชื่อมต่อหรือ Interface อาจจะมีหมายถึง Network Interface card (LAN card) ที่ติดตั้งในเซิร์ฟเวอร์หรือ WAN port, Ethernet port ที่ Router ใช้เชื่อมต่อเข้ากับเครือข่ายเป็นต้นการกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อนี้ทำให้เราเข้าใจได้ว่าในบางอุปกรณ์ที่มีจุดเชื่อมต่อเข้ากับเครือข่ายมากกว่าหนึ่งจุด ต้องกำหนดหมายเลข IP address ให้ครบ

การ Bind IP address

เมื่อได้กำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเช่น LAN card แล้วที่เครื่องเซิร์ฟเวอร์จะต้องมีการ bind หรือผนวกค่า IP address ดังกล่าวเข้ากับ Ethernet driver เพื่ออ้างอิงหมายเลข IP กับฮาร์ดแวร์ให้ทำหน้าที่ติดต่อส่งข้อมูลในระดับ Network Interface ได้ต่อไป ดังตัวอย่างในรูปที่ 2.12



รูปที่ 2.12 การ Bind IP address

จากรูปจะแสดงค่า bind IP address 204.183.255.20 เข้ากับ Ethernet driver โปรโตคอล IP จะใช้ค่า IP address นี้ในการติดต่อกันและผ่านฮาร์ดแวร์ที่ถูก bind ไว้อีกต่อหนึ่ง ค่าหมายเลขฮาร์ดแวร์ก็ได้แก่ MAC address ที่มีประจำอยู่บน Lan card ซึ่งจะไม่ได้ใช้งานอ้างอิงโดยตรงแต่จะผ่านหมายเลข IP address แทน

2.1.6.2 โปรโตคอล ARP

โปรโตคอล ARP (Address Resolution Protocol) ถูกเรียกใช้งานโดยโปรโตคอล IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น เว็บเซิร์ฟเวอร์เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต และในการเชื่อมต่อนี้ต้องอาศัย Network Interface Card (NIC) หรือ LAN card ติดตั้งอยู่ที่ LAN card นี้เองจะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานในโปรโตคอล TCP/IP ก็จะต้องมีการกำหนดหมายเลข IP address ประจำตัวเพื่อใช้อ้างอิงกัน และโปรโตคอล ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงไว้ในระดับการทำงานที่ Internetwork Layer นี้ ซึ่งกลไกการแปลงนี้เรียกว่า address resolution

โปรโตคอล ARP ย้อนกลับ RARP (Reverse Address Resolution Protocol)

วิธีการ ARP ช่วยแก้ปัญหาในการค้นหาที่อยู่ ของข้อมูลที่ใช้การกำหนดที่อยู่ฮาร์ดแวร์ แบบ IP แต่ถ้าทราบที่อยู่แบบ ฮาร์ดแวร์ แล้วต้องการแปลงที่อยู่เป็น IP จะทำอย่างไร ปัญหานี้มักเกิดขึ้นกับเครื่องคอมพิวเตอร์ ที่เริ่มทำงานด้วยการอ่านข้อมูลทั้งหมดจากเครื่อง Host เครื่องประเภทนี้จะทราบเพียงที่อยู่ ของตนเองจากอุปกรณ์สื่อสารเครือข่ายเท่านั้น

การค้นหาคำตอบสามารถทำได้โดยวิธีควบคุมการสื่อสารแบบ ARP ย้อนกลับ หรือ RARP (Reverse Address Resolution Protocol) วิธีการนี้คอมพิวเตอร์ที่เพิ่งจะเริ่มทำงาน (หรือเครื่องใดก็ได้แล้วแต่) จะส่งคำถามออกไปในทำนอง “ที่อยู่ขนาด 48 Bits แบบ ฮาร์ดแวร์ ของฉันคือ 14.04.05.18.01.25 มีใครทราบที่อยู่ IP ของฉันบ้าง” เครื่องที่ให้บริการ RARP จะตรวจดูข้อมูลในตารางข้อมูลของตนเองแล้วจึงส่งหมายเลข IP กลับไปให้ วิธีการนี้ช่วยให้เกิดความอ่อนตัวและเพิ่มประสิทธิภาพในการใช้หมายเลข IP เนื่องจากผู้ใช้ไม่มีหมายเลข IP เป็นของ

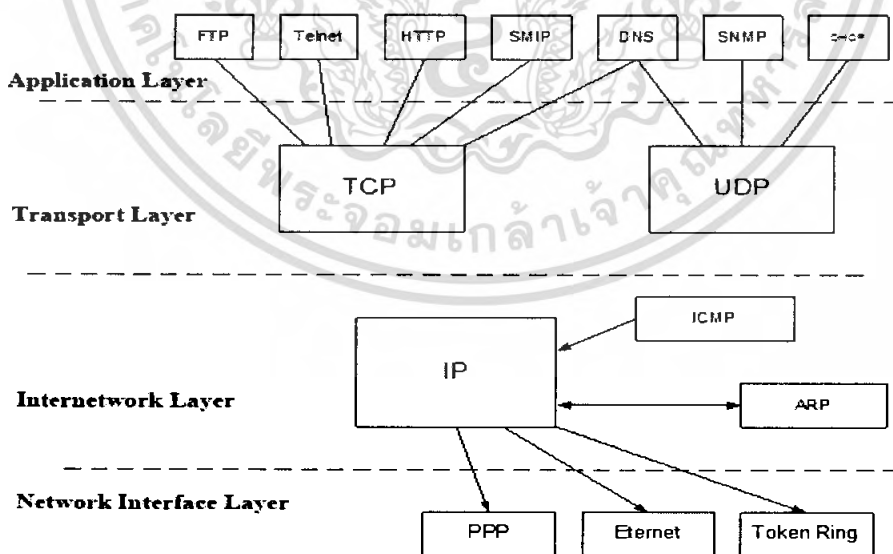
ตนเอง ผู้ควบคุมระบบสามารถกำหนดหมายเลข IP ใดๆที่ไม่มีผู้ใช้งานในขณะนั้นให้ใช้ได้ หมายเลข IP ในที่นี้จึงเป็นเสมือนสมบัติส่วนกลางที่ทุกคนใช้ร่วมกัน

ข้อด้อยของวิธี RARP คือการที่ผู้ใช้จะส่งคำถามโดยใช้หมายเลข 1 จำนวน 48 ตัวเป็นที่อยู่ของผู้ให้บริการ หมายเลขนี้เป็นหมายเลขพิเศษที่ Router จะไม่ยอมส่ง Packet ผ่านไปยังเครือข่ายอื่นเลย ฉะนั้นผู้ให้บริการ RARP จะต้องมียุ่ประจำทุกเครือข่าย อย่างไรก็ตาม Protocol แบบ BOOTP ได้รับการพัฒนาขึ้นมาเพื่อแก้ปัญหานี้โดยการใช้ Packet UDP แทน Packet ชนิดนี้สามารถส่งไปได้ทั่วทุกเครือข่ายและยังให้ข้อมูลอื่นเพิ่มเติม เช่น หมายเลข IP ของผู้ให้บริการเพิ่มข้อมูล หมายเลข IP ของ Router อัตโนมัติ และตารางข้อมูลเครือข่ายย่อเป็นต้น

2.1.7 Network Interface Layer

เนื่องจากในด้านกายภาพของเครือข่ายนั้น มีหลายวิธีการและหลายรูปแบบในการเชื่อมต่อระบบให้เป็นเครือข่าย แต่อย่างไรก็ตามในเครือข่ายอินเทอร์เนตนี้ ข้อมูลหรือ IP datagram จะถูกถ่ายทอดและส่งผ่านไปยังปลายทางโดยไม่คำนึงถึงรูปแบบการเชื่อมต่อทางกายภาพ ไม่ว่าจะเป็นการใช้เครือข่ายใยแก้วนำแสงหรือเครือข่ายสาย Unshielded Twist Pair (UTP) เชื่อมต่อเป็นแบบเครือข่าย Ethernet ธรรมดาหรือเครือข่าย Token Ring, ATM, ISDN ฯลฯ ก็ตาม

การทำงานระดับล่างสุดต่อจาก Internetwork Layer จะเป็นการแปลงข้อมูล IP datagram ให้อยู่ในรูปแบบที่เหมาะสม และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่ายต่อไป ซึ่งในชั้น Network Interface Layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นการรวม 2 Layer เข้าด้วยกันคือ Data link layer และ Physical Layer กล่าวโดยสรุปคือ การทำงานในชั้นต่างๆ ตามโครงสร้างของโพรโทคอล TCP/IP จะมีลักษณะดังรูปที่ 2.13



รูปที่ 2.13 โครงสร้างของโพรโทคอล TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 MAC Address

MAC Address เป็นตัวบ่งชี้ลักษณะเฉพาะของแต่ละ machine ดังนั้นจึงต้องเป็นค่าที่ไม่ซ้ำกัน (unique) MAC Address เป็นเลข 48 bits โดยแบ่งออกเป็น 2 ส่วน โดย 24 bits แรกเป็นค่าที่แสดงถึงบริษัทที่ผลิตการ์ดนั้น ๆ ส่วน 24bits หลังเป็น serial number ที่ทางบริษัทกำหนดให้ซึ่งแต่ละตัวต้องไม่ซ้ำกัน เราเรียกเลข 24 bit นี้ว่า OUI (Organizationally Unique Identifier) ซึ่ง OUI จะใช้เพียง 22 bits เท่านั้น ส่วนอีก 2 bits ที่เหลือจะถูกใช้เพื่อวัตถุประสงค์อื่น โดย bit หนึ่งจะใช้เพื่อแสดงว่า address นั้นเป็น broadcast/multicast address ส่วนอีก bit หนึ่งนั้นไว้แสดงว่า adapter นั้นถูกกำหนด locally administered address ซึ่ง admin ของระบบจะทำการกำหนด MAC Address เพื่อความเหมาะสมของนโยบายระบบ เช่น

MAC Address = 03 00 00 00 00 00 01 ซึ่งจะเห็นว่า byte แรก = 03 = 00000011 นั่นคือ ทั้ง 2 bits ถูก set (reset = 0) ซึ่งเอาไว้กรณี multicast ให้ทุกเครื่องที่ run บนโปรโตคอล NetBEUI

2.1.9 ระบบเครือข่าย Ethernet

ระบบเครือข่าย Ethernet เป็นระบบเครือข่ายท้องถิ่นหรือ LAN (Local Area Network) ประกอบด้วยส่วนที่เป็นฮาร์ดแวร์และซอฟต์แวร์ที่ทำงานร่วมกันเพื่อการส่งถ่ายข้อมูลในระบบ Digital ระหว่างคอมพิวเตอร์ ระบบที่ใช้เทอร์เน็ตนั้นเหมาะกับการรับส่ง/ข้อมูลในอัตราความเร็วสูงเป็นช่วง ๆ เป็นครั้งคราว การรับ/ส่งข้อมูลในเครือข่ายแบบ Ethernet แต่ละครั้งเป็นไปอย่างไม่มีวินัย นั่นคือเมื่อตรวจสอบแล้วว่าในขณะนั้นไม่มีเครื่องอื่น ๆ กำลังส่งข้อมูล แต่ละเครื่องจะแย่งกันส่งข้อมูลออกมา โดยเครื่องใดที่ส่งข้อมูลออกมาก็จะมีหน้าที่เฝ้าดูว่ามีเครื่องอื่นทำการส่งข้อมูลออกไปพร้อมกันด้วยหรือไม่ เพราะถ้าเกิดการส่งพร้อมกันแล้วจะก่อให้เกิดการชนกันของข้อมูล แต่ถ้าตรวจจับได้ว่าการชนกันขึ้นก็จะหยุดส่งแล้วรอคอยเป็นระยะเวลาสั้น ๆ ก่อนจะทำการส่งข้อมูลออกไปอีกครั้งหนึ่ง เวลาที่ใช้ในการรอคอยนั้นเป็นค่าที่สุ่มขึ้นมา ซึ่งมีความสั้นยาวต่างกันไป เทคนิคหลายอย่างเช่นที่นำมาใช้ในการรอคอยเพื่อหลีกเลี่ยงการชนกันซ้ำสอง หนึ่งในนั้นคือ ค่าวนการเพิ่มระยะเวลารอคอยแบบ Exponential ซึ่งมีชื่อเรียกว่า Carrier Sense Multiple Access with Collision Detection (CSMA/CD) ระบบเครือข่าย Ethernet มีลักษณะพิเศษดังนี้

- เป็นระบบเครือข่ายที่มีความเร็วในการส่งข้อมูลในรูปแบบดิจิทัลที่มีความเร็วตั้งแต่ 10 Mbps จนถึง 1,000 Mbps (1 Gbps)
- เป็นเครือข่ายที่มีขนาด Diameter ตั้งแต่ 205 เมตรจนถึง 4,000 เมตร
- ใช้โปรโตคอลการทำงานที่เรียกว่า CSMA/CD (Carrier Sense Multiple Access with Collision Detect) ซึ่งเป็นมาตรฐานของ IEEE802.3 นอกจากนี้ก็ยังมีมาตรฐาน IEEE802.3u สำหรับ 100 Mbps Fast Ethernet และ IEEE2.3 สำหรับ Gigabit Ethernet รวมทั้ง IEEE802.3ab สำหรับ Gigabit Ethernet ที่ใช้สายทองแดง
- หนึ่งเครือข่าย Ethernet สามารถมีอุปกรณ์เชื่อมต่อ เช่น คอมพิวเตอร์ลูกข่าย อุปกรณ์ Repeater เป็นต้น ได้มากมายถึง 1,024 รายการหรือเรียกว่า Node

- เป็นเครือข่ายที่สามารถใช้สายสัญญาณได้หลายแบบ เช่น สาย Coaxial ทั้งแบบหนาและแบบบาง สาย Twisted Pair ทั้งแบบ Shield และ Unshield รวมทั้งสาย Optical Fiber แบบและขนาดต่างๆ นอกจากนี้ยังสามารถใช้สื่อที่ไร้รับส่งข้อมูลแบบไร้สาย เช่น คลื่นวิทยุที่มีความถี่ Spread Spectrum รวมทั้งไมโครเวฟ (Microwave) ที่ใช้ความถี่ในช่วง 14 GHz และอินฟราเรด (infrared) เป็นต้น
- เป็นระบบเครือข่ายที่มีการเชื่อมต่อในรูปแบบ Bus และ Star Topology
- อุปกรณ์ที่ใช้มีราคาประหยัด
- มีความน่าเชื่อถือสูง โดยเฉพาะหากใช้สื่อที่เป็นสาย Optical Fiber
- มีเครื่องมือในรูปแบบของซอฟต์แวร์ที่ใช้บริหารจัดการเครือข่ายมากมายที่ทำงานภายใต้ SNMP (Simple Network Management Protocol)

2.1.10 ส่วนประกอบหลักที่สำคัญของเครือข่าย Ethernet

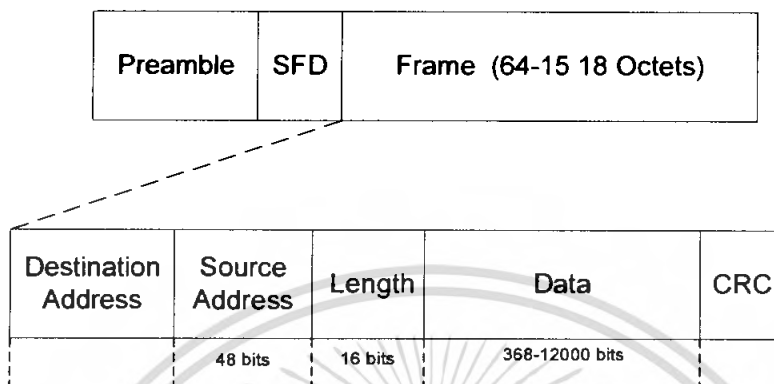
ระบบเครือข่าย Ethernet มีส่วนประกอบหลักซึ่งเมื่อทำงานด้วยกันแล้วก็จะจะเป็นเครือข่ายที่มีประสิทธิภาพการทำงานสูงดังนี้

1. ตัวเฟรมเป็นชุดรูปแบบของบิตข้อมูลข่าวสารที่ใช้ส่งผ่านมายนระบบ หากไม่มีเฟรมเราจะไม่สามารถสื่อสารข้อมูลบนเครือข่ายได้โดยเด็ดขาด การรับส่งข้อมูลข่าวสารบนเครือข่าย Ethernet จะต้องเป็นไปในรูปแบบเฟรมมาตรฐาน 2 แบบ และเป็นแบบใดแบบหนึ่งเท่านั้น (การ์ด LAN เป็นผู้สร้างเฟรมนี้ขึ้นมา)
2. ชุดโปรโตคอลที่ใช้ในการควบคุมการแอกเซสเข้าไปที่เครือข่าย (Media Access Control Protocol) ซึ่งประกอบด้วยชุดของกฎกติกาที่อยู่ใน Ethernet Interface (เช่น การ์ด LAN เป็นต้น) ซึ่งเป็นกฎมาตรฐานที่จะยอมให้คอมพิวเตอร์ต่างๆสามารถเข้ามาที่เครือข่าย และแบ่งใช้ทรัพยากรต่างๆ บนเครือข่ายได้อย่างมีประสิทธิภาพ
3. อุปกรณ์ที่ใช้รับส่งสัญญาณบนเครือข่าย (Signaling Components) ประกอบด้วยชุดของอุปกรณ์ที่ใช้เชื่อมต่อและส่งสัญญาณเพื่อการรับส่งข้อมูลภายในเครือข่าย
4. สื่อที่ใช้ในการรับส่งสัญญาณข้อมูลบนเครือข่าย (Physical Medium) ประกอบด้วยสายสัญญาณรวมทั้งอุปกรณ์ทางฮาร์ดแวร์อื่นๆ ที่จะช่วยในการนำพาข้อมูลข่าวสารต่างๆในรูปแบบ Digital วิ่งไปมาบนเครือข่าย

2.1.11 เฟรมบนระบบ Ethernet

หัวใจสำคัญของระบบ Ethernet ได้แก่ เฟรมข้อมูลทางข่าวสารและอุปกรณ์ทางฮาร์ดแวร์ที่ใช้เชื่อมต่อสื่อสารบนเครือข่าย ซึ่งได้แก่ การ์ด Ethernet LAN สายสัญญาณและอุปกรณ์เสริมอื่นๆ ที่จะช่วยนำพาข้อมูลในรูปแบบของบิตทางดิจิทัลที่เรียกว่าเฟรม วิ่งไปมาระหว่างคอมพิวเตอร์บนเครือข่าย

เฟรมข้อมูลสำหรับระบบ Ethernet ประกอบขึ้นด้วยกลุ่มของบิตที่เป็นข้อมูลและข่าวสารสำคัญ แบ่งออกเป็นขนาดสัดส่วนที่แน่นอนที่เรียกว่าช่อง Field ดังรูปที่ 2.14



รูปที่ 2.14 ลักษณะ โครงสร้างของเฟรมข้อมูล

จากรูปที่ 2.14 แสดงให้เห็นรูปแบบของเฟรมข้อมูลที่ใช้บน Ethernet ได้แก่ Ethernet Frame ตามมาตรฐาน IEEE802.3 ส่วนรูปที่ 2.15 เป็น Ethernet II Frame ซึ่งทั้งสองเฟรมจะมีความแตกต่างกันเล็กน้อย ทำให้เครือข่ายที่ใช้เฟรมแตกต่างกันนี้อาจไม่สามารถเข้ากันได้ หมายความว่าระบบเครือข่าย Ethernet ของท่านจะต้องเลือกใช้อุปกรณ์เครือข่ายที่คอยสนับสนุนเฟรมอย่างใดอย่างหนึ่งเท่านั้น แต่ก็เป็นเรื่องที่ดีที่ผู้ผลิตอุปกรณ์ทางฮาร์ดแวร์ที่ใช้เชื่อมต่อเครือข่าย Ethernet มีการใช้มาตรฐาน IEEE802.3 เป็นหลัก อย่างไรก็ตามก็ยังมีผู้ผลิตอุปกรณ์สนับสนุนเฟรมทั้งสองแบบในตัวเองกัน

Preamble	Destination MAC Address (6 Byte)	Source MAC Address (6 Byte)	Type (2 Byte)	Data Field (1500 Byte Max)	Frame Check Sequence (4 Byte)
----------	----------------------------------	-----------------------------	---------------	----------------------------	-------------------------------

รูปที่ 2.15 ลักษณะของ Ethernet II Frame

เฟรมมาตรฐานของ IEEE802.3

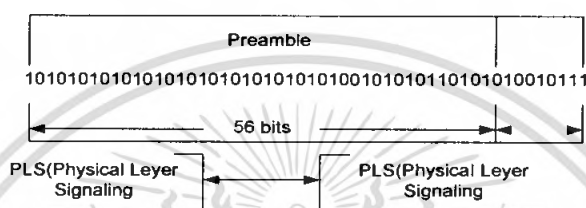
ช่อง Preamble

ช่อง Preamble ประกอบด้วยบิตข่าวสารที่เป็นเลข 1 และ 0 สลับกัน และสิ้นสุดที่ 11 ซึ่งเป็นบิตที่ 63 และ 64 เป็นบิตข่าวสารที่ยังไม่ใช่ข้อมูลจริงของผู้ส่ง Preamble ประกอบด้วยข่าวสารที่มีขนาด 7 หรือ 8 ไบต์ จุดประสงค์ของข่าวสารนี้เพื่อใช้สร้างจังหวะการรับข้อมูลให้แก่ผู้รับโดยที่ส่วนนี้จะไปถึงตัวผู้รับก่อน ทำให้

เครื่องคอมพิวเตอร์ของผู้รับสามารถปรับจังหวะความเร็วให้เข้ากับผู้ส่งได้ (Synchronize) สำหรับเฟรมแบบ Ethernet II จะมีขนาด 8 ไบต์ และถ้าเป็นมาตรฐาน IEEE802.3 แล้ว ช่องนี้จะถูกแบ่งออกเป็น 2 ส่วน ได้แก่

1. Preamble
2. Start of Frame Delimeter

จุดประสงค์อีกประการหนึ่งของ Preamble ได้แก่การทำให้แน่ใจว่าระยะเวลาความห่างระหว่างเฟรมที่ 1 กับเฟรมถัดมาจะมีความห่างอยู่ที่ 9.6 ไมโครวินาทีโดยมาตรฐาน ทั้งนี้ก็เพื่อการปฏิบัติการตรวจสอบความผิดพลาด และกู้สถานการณ์เมื่อเกิดปัญหา



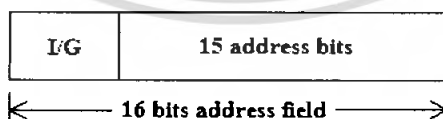
รูปที่ 2.16 ลักษณะส่วนการทำงานภายในของ Preamble

ช่อง Preamble รวมทั้ง Start of Frame Delimeter นี้จะถูกถอดออกไปเป็นอันดับแรกหลังจากคอมพิวเตอร์ของผู้รับได้เฟรมข้อมูลนี้เป็นที่เรียบร้อยแล้ว โดยการ์ด LAN ของผู้รับจะทำการตรวจสอบว่าจำนวนบิตในช่อง Preamble มีครบ 64 บิตหรือไม่ หากไม่ครบก็แสดงว่าเป็นเฟรมที่ไม่สมบูรณ์ โดยอาจเป็นเฟรมที่หลงเหลือมาจากการชนกับเครือข่ายก็ได้ ลักษณะเช่นนี้ การ์ด LAN ของผู้รับจะปฏิเสธที่จะรับและโยนทิ้ง (Frame Drop) ในทันที

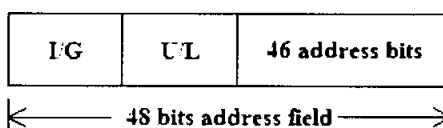
ช่อง Destination Address

ในช่อง Destination Address ประกอบด้วยข้อมูลข่าวสารเกี่ยวกับแอดเดรสหรือที่อยู่ของผู้รับปลายทาง ช่องนี้ถูกแบ่งออกเป็นช่องย่อยๆที่เรียกว่า Sub-fields ซึ่งเก็บข้อมูลข่าวสารที่ใช้ดูแลการทำงานของเครือข่าย ทั้งนี้ขึ้นอยู่กับแอดเดรสที่ปรากฏอยู่ในช่องนี้ไม่ว่าช่องแอดเดรสจะมีแอดเดรสที่ถูกเจาะจงเป็นรายบุคคลหรือเป็นกลุ่มของผู้รับก็ตาม

ก. ช่องข่าวสารขนาด 2 ไบต์



ข. ช่องข่าวสารขนาด 6 ไบต์



หมายเหตุ

หากค่าบิตในช่องย่อย I/G มีค่าเป็น “0” ก็หมายถึงแอดเดรสที่ระบุตัวคอมพิวเตอร์ผู้รับอย่างเฉพาะเจาะจง แต่ถ้ามีค่าเป็น “1” ก็หมายถึงคอมพิวเตอร์ที่ระบุแอดเดรสเป็นกลุ่ม ไม่เจาะจงเครื่องใดเครื่องหนึ่ง

หากค่าบิตในช่อง U/L มีค่าเป็น “0” ก็หมายความว่าแอดเดรสนี้ถูกกำหนดมาตรฐานโดย IEEE และถ้ามีค่าเป็น “1” ก็จะหมายถึงเป็นแอดเดรสมาตรฐานเฉพาะองค์กรที่ระบุมาตรฐานในซอฟต์แวร์ซึ่งแอดเดรสมาตรฐานที่เราใช้กันอยู่ทุกวันนี้ถูกควบคุมโดย IEEE

จากกรอบแผนผังในข้อ ก. และข้อ ข. เป็นการแสดงรูปภาพในช่องย่อยๆ ของ Destination Address ซึ่งมีทั้งแบบขนาด 2 ไบต์สำหรับเฟรมแบบ IEEE802.3 เท่านั้น ส่วนแบบ 6 ไบต์สำหรับเฟรมแบบ Ethernet และ IEEE802.3 ผู้ใช้งานสามารถเลือกใช้ช่องย่อยใน Destination Address ทั้งแบบ 2 ไบต์และ 6 ไบต์ได้อย่างใดอย่างหนึ่ง และเนื่องจากอุปกรณ์ที่ใช้เป็นแบบ IEEE802.3 ดังนั้นทุกสถานีเครือข่ายทุกเครื่องจะต้องใช้โครงสร้างการอ้างอิงแอดเดรสแบบเดียวกันเสมอ เช่น หากเลือกใช้แบบ 2 ไบต์ก็หมายความว่าคอมพิวเตอร์ทุกเครื่องที่เชื่อมต่อเข้ากับเครือข่ายจะต้องใช้แบบ 2 ไบต์ทั้งหมด

ทุกวันนี้เครือข่ายเกือบทั้งหมดใช้มาตรฐานแบบ 6 ไบต์เพื่อการอ้างอิงแอดเดรสถึงกัน แต่ก็ยังร่วมช่องขนาด 2 ไบต์นี้เข้าไปเป็นทางเลือกอีกด้วย

ช่อง I/G

มีการจัดตั้งค่าบิตขึ้นในช่องนี้โดยการ์ด LAN ซึ่งหากค่านี้ถูกตั้งค่าไว้ที่ “0” ก็แสดงว่าตัวแอดเดรสที่ระบุอยู่ในช่อง Destination Address นั้นเป็นแอดเดรสที่ระบุตัวคอมพิวเตอร์ผู้รับแบบเฉพาะเจาะจง แต่ถ้าถูกตั้งค่าเป็น “1” ก็แสดงว่าแอดเดรสในช่อง Destination Address นี้เป็นแอดเดรสที่ใช้ติดต่อผู้รับที่เป็นกลุ่มคอมพิวเตอร์ทั้งหลาย เราเรียก Group Address ตัวอย่างของ Group Address ได้แก่ “FFFFFFFF” ซึ่งถือว่าเป็น Broadcasting Address หรือแอดเดรสที่ไม่เจาะจงผู้รับ โดยผู้รับเป็นกลุ่มหรือทั้งหมดก็สามารถรับข้อมูลข่าวสารนี้ได้

ช่องย่อย U/L

ช่องย่อย U/L มีไว้สำหรับช่องขนาด 6 ไบต์เท่านั้น ค่าที่ถูกตั้งไว้ในช่องย่อยนี้เป็นการบ่งบอกให้ทราบว่าแอดเดรสที่ปรากฏอยู่ในช่อง Destination Address นี้เป็นแอดเดรสที่ถูกกำหนดมาตรฐานโดย IEEE หรือองค์กรอย่างเฉพาะเจาะจง

ช่อง Source Address

สำหรับช่อง Source Address นี้มีไว้เพื่อแสดงตัวสถานีเครือข่ายต้นทางที่เป็นต้นทางส่งข้อมูลข่าวสารเข้ามาและเช่นเดียวกับช่อง Destination Address กล่าวคือ ช่อง Source Address สามารถมีช่องย่อยได้ทั้งแบบ 2 ไบต์หรือ 6 ไบต์อย่างใดอย่างหนึ่ง

ช่องย่อย Source Address แบบ 2 ไบต์ใช้กับมาตรฐาน IEEE802.3 และต้องใช้แอดเดรสปลายทางขนาด 2 ไบต์เท่านั้น รวมทั้งทุกสถานีถูกข่ายจะต้องใช้ช่องแอดเดรสขนาด 2 ไบต์ส่วนช่องย่อยขนาด 6 ไบต์ สามารถใช้ได้ทั้งมาตรฐาน Ethernet ทั่วไปและ IEEE802.3 และเมื่อมีการเลือกใช้ช่องย่อย 6 ไบต์ก็จะมีการกำหนดให้ 3 ไบต์แรก

เป็นแอดเดรสที่ IEEE กำหนดให้ผู้ผลิตต่างๆซึ่งแอดเดรสนี้จะถูกฝังตัวอยู่ในไมโครชิป บนการ์ด LAN ส่วนที่เหลืออีก 3 ไบต์ก็จะเป็นแอดเดรสที่ผู้ผลิตการ์ด LAN แต่ละแห่งนำไปกำหนดกันเองต่อไป

ตารางที่ 2.5 ตัวอย่างของ Source Address ที่แสดงรหัสแอดเดรสของผู้ผลิต

ผู้ผลิตการ์ด LAN	รหัสผู้ผลิตขนาด 3 ไบต์
Cisco	00-00-0C
Cabletron	00-00-1D
Intel	00-AA-00
3 Com	02-60-8C
Hewlett Packard	08-00-09
Sun	08-00-20
DEC	08-00-2B
Shiva	00-80-D3
Xerox	00-00-AA
IBM	08-00-5A

ช่องแสดง Type

ช่องแสดง Type มีขนาด 2 ไบต์ ใช้กับ Ethernet Frame เท่านั้น โดยช่องนี้ใช้เพื่อแสดงว่าโปรโตคอลการทำงานของเฟรมนี้เป็นแบบใด จุดประสงค์คือเพื่อต้องการให้ทราบว่าข้อมูลที่อยู่ในเฟรมนี้ จะทำงานภายใต้โปรโตคอลใด ซึ่งผู้รับจะได้เตรียมการแปลความหมายที่อยู่ในช่องข้อมูล (Data Field) ได้ถูกต้อง

ภายใต้ระบบเครือข่าย Ethernet เราสามารถใช้โปรโตคอลได้หลายตัวพร้อมกันบนเครือข่าย LAN และบริษัท XEROX ทำหน้าที่เป็นผู้ให้บริการ กำหนดทึ่กัระยะของแอดเดรสที่เป็นลิขสิทธิ์ให้แก่ผู้ผลิตการ์ด LAN ต่างๆรวมทั้งการกำหนดค่าที่ใช้แสดงแทนโปรโตคอลที่ใช้ในช่อง Type แห่งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 ตัวอย่างของรหัสที่ใช้แสดงแทนโปรโตคอลที่ใช้ในช่อง Type 18 รายการ

โปรโตคอลที่ใช้	ค่าที่เป็นรหัสแบบเลขฐาน 16
IP	0800
X.75 Internet	0801
X.25 Level 3	0805
Address Resolution Protocol (ARP)	0806
Banyan Systems	0BAD
BBN Simnet	5208
DEC MOP Dump/Load	6001
DEC MOP Remote Console	6002
DEC DECNET Phase IV Route	6003
DEC LAT	6004
DEC Diagnostic Protocol	6005
DEC LANBridge	8038
DEC Ethernet Encryption	803D
Apple Talk	809B
IBM SNA Service on Ethernet	80D5
Apple Talk ARP	80F3
NetWare IPX/SPX	8137
SNMP	814C

ช่อง Length

ช่องนี้มีขนาดความยาวเพียง 2 ไบต์ใช้ได้กับเฟรมมาตรฐาน IEEE802.3 เท่านั้นเป็นช่องที่ใช้แสดงขนาดจำนวนของไบต์ที่มีปรากฏอยู่ในช่อง Data

ภายใต้มาตรฐาน Ethernet และ IEEE802.3 ขนาดของเฟรมจะมีขนาดเล็กที่สุดไม่ต่ำกว่า 64 ไบต์ นับตั้งแต่จุดแรกจุดคือ Preamble จนถึงช่องสุดท้าย ได้แก่ FCS และการกำหนดให้มีขนาดเล็กที่สุดไม่น้อยกว่า 64 ไบต์นี้จุดประสงค์ก็เพื่อให้แน่ใจว่าช่วงระยะเวลาส่งข้อมูลมีมากพอที่จะทำให้การ์ด LAN สามารถตรวจพบการเกิด การชนกันของข้อมูลบนสายสัญญาณที่มีขนาดความยาวที่สุดของเครือข่าย หากขนาดเฟรมเล็กกว่า 64 ไบต์ก็อาจเกิดปัญหาการชนกันของข้อมูล ซึ่งจะนำไปสู่ปัญหาบนเครือข่ายได้

บนพื้นฐานของเฟรมขนาดเล็กที่สุดคือ 64 ไบต์ และมีการใช้ช่องแอดเดรสขนาด 2 ไบต์หมายความว่าช่องสำหรับ Data จะต้องมีความยาวเล็กที่สุดไม่ต่ำกว่า 46 ไบต์ (เมื่อหักขนาดและจำนวนช่องต่างๆออกไปหมดแล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใดที่ข้อมูลมีขนาดเล็กกว่า 46 ไบต์ของ Data ที่อยู่เฟรมจะถูกใส่ค่าเพิ่มให้ได้อย่างน้อยเป็น 46 ไบต์

ช่อง Data (Data Field)

ดังที่ได้กล่าวมาแล้วว่าช่องของ Data อย่างน้อยต้องมีขนาดไม่เล็กกว่า 46 ไบต์ เพื่อให้แน่ใจว่าเฟรมมีขนาดไม่ต่ำกว่า 64 ไบต์ซึ่งหมายความว่า การแพร่ข้อมูลขนาดหนึ่งไม่ว่า 1 หรือ 10 ไบต์ก็ตามต้องมาจาก 46 ไบต์นี้ แต่ถ้าข้อมูลในช่องนี้เล็กกว่า 46 ไบต์แน่นอนว่าต้องมีการเพิ่มไบต์ลงไปอีกเพื่อให้ได้ขนาด 46 ไบต์อยู่ดี ขนาดของข้อมูลที่อยู่ใน Data จะต้องมีความยาวสูงสุดไม่เกิน 1,500 ไบต์ช่องตรวจสอบความผิดพลาดของข้อมูลในเฟรม (Frame Check Sequence)

ช่อง Frame Check Sequence

ใช้ได้เฉพาะมาตรฐานทั้ง Ethernet และ IEEE802.3 เป็นช่องที่ประกอบด้วยข้อมูลที่ใช้เป็นกลไกในการตรวจสอบความผิดพลาดของข้อมูลภายในเฟรม

หลักการการทำงานมีอยู่ว่าก่อนที่เครื่องผู้ส่งจะส่งข้อมูลออกไปที่เครือข่าย การ์ด LAN จะคำนวณค่าต่างๆ ในช่องต่างๆ ซึ่งครอบคลุมตั้งแต่ช่อง Address ต่างๆ ของ type และช่อง Length รวมทั้งช่อง Data การคำนวณค่าแบบนี้เรียกว่า Cyclic Redundancy Check (CRC) ซึ่งหลังจากที่ได้คำนวณค่าเสร็จสิ้นแล้ว ผลลัพธ์ที่คำนวณได้มีขนาด 4 ไบต์จะถูกนำไปใส่ไว้ในช่อง Frame Check Sequence แห่งนี้

เมื่อเฟรมถูกส่งมาถึงผู้รับแล้ว ตัวการ์ด LAN ของผู้รับจะทำการตรวจสอบค่าที่อยู่ในช่อง Preamble เพื่อดูว่ามีความถูกต้องหรือไม่ เพื่อให้แน่ใจว่าเฟรมที่ทำการตรวจสอบอยู่นี้ไม่ได้เป็นเฟรมที่หลุดรอดจากการชนกันของสัญญาณในเครือข่าย และหาก Preamble ไม่มีปัญหาเรื่องความถูกต้องก็จะมีค่าที่อยู่ในช่อง Frame Check Sequence ต่อไป หากมีความผิดพลาดก็คือ ค่าที่ได้ไม่ต้องกับค่าที่ได้จากการที่คำนวณได้จากต้นทาง แสดงว่าเป็นเฟรมที่มีปัญหา ซึ่งก็มักเป็นปัญหาจากสายสัญญาณ รวมทั้งสัญญาณรบกวน ซึ่งในที่สุดการ์ด LAN ก็จะมีปฏิกิริยาที่จะรับเฟรมที่เข้ามานี้ในที่สุด

ข้อกำหนดเกี่ยวกับขนาดของ Data Frame

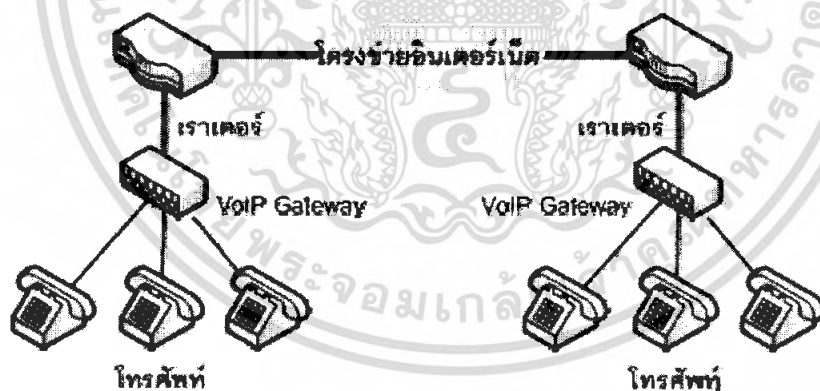
ขนาดของ Data Frame มีมาตรฐานดังต่อไปนี้

- ขนาดเล็กที่สุด ต้องไม่น้อยกว่า 64 ไบต์ โดยมี 12 ไบต์สำหรับแอดเดรส 2 ไบต์สำหรับช่อง Length 46 ไบต์สำหรับเก็บข้อมูล และ 4 ไบต์สำหรับตรวจสอบความผิดพลาดข้อมูล หรือ Frame Check Sequence
- ขนาดใหญ่ที่สุด ต้องไม่เกิน 1,518 ไบต์ โดยแบ่งออกเป็น 12 ไบต์สำหรับแอดเดรส 2 ไบต์สำหรับ Length 1,500 ไบต์สำหรับข้อมูล และ 4 ไบต์สำหรับช่องตรวจสอบความผิดพลาดข้อมูล
- เฟรมที่มีขนาดเล็กที่สุด 64 ไบต์นี้เทียบกับ 512 บิต (Bit/Byte) โดยที่ระบบ Ethernet มีค่า Bit Time อยู่ที่ 0.1 ไมโครวินาที (Bit Time เป็นช่วงเวลาที่ใช้ในการส่งข้อมูลขนาด 1 บิต) ดังนั้นการส่งข้อมูลขนาดเล็กที่สุดคือ 64 ไบต์จะต้องใช้เวลาอยู่ที่ 51.2 ไมโครวินาที

2.2 Voice over IP (VoIP)

เทคโนโลยีที่ใช้ในการรับส่งเสียงพูดของมนุษย์โดยผ่านเครือข่ายสื่อสารประเภทโทรศัพท์ ไม่ว่าจะ เป็นโทรศัพท์พื้นฐาน หรือโทรศัพท์เคลื่อนที่ รวมถึงในอุปกรณ์ชุมสายโทรศัพท์ขนาดเล็กภายในสำนักงาน นั้นล้วนอยู่บนพื้นฐานของการเปลี่ยนสัญญาณเสียงพูดของมนุษย์ที่เป็นแบบ อนาลอก ให้กลายเป็นสัญญาณ ดิจิตอล โดยใช้ สุ่มสัญญาณ (Sampling) ด้วยความถี่ 8 kHz โดยเสียงพูดที่ถูกสุ่มสัญญาณดังกล่าวจะถูกนำไปแปลงระดับความแรงของสัญญาณโดยแทนที่ด้วยข้อมูล 8บิต ต่อหนึ่งการสุ่ม เรียกกันว่าเป็นกระบวนการ Pulse Code Modulation (PCM) จนเมื่อโลกอินเทอร์เน็ตเริ่มเติบโตขึ้น และอินเทอร์เน็ตเป็นเครือข่ายคอมพิวเตอร์ที่เชื่อมต่อกันและเข้าถึงผู้ใช้งานได้มากและหลากหลายที่สุด ซึ่งทำให้เกิดแนวคิดในการพัฒนารูปแบบในการรับส่งข้อมูลบนอินเทอร์เน็ต โดยมุ่งหวังให้สามารถรับส่งเสียงพูดผ่านเครือข่ายอินเทอร์เน็ต โดยการรับส่งข้อมูลบนเครือข่ายคอมพิวเตอร์เป็นแบบ Packet ไม่ว่าจะข้อมูลที่ต้องการรับส่งจะเป็นภาพ ข้อความหรือมัลติมีเดียใดๆก็ตาม จะต้องได้รับการแบ่งและจัดให้อยู่ในรูปแบบของ Packet เสียก่อน

การส่งเสียงบนเครือข่ายไอพี (VoIP) เป็นระบบที่แปลงสัญญาณเสียงในรูปของสัญญาณไฟฟ้ามาเปลี่ยนเป็นสัญญาณดิจิตอล โดยหนึ่งช่องเสียงเมื่อแปลงเป็นข้อมูลจะมีขนาด 64 kbps การนำข้อมูลเสียงขนาด 64 Kbps นี้ต้องนำมาบีบอัด โดยทั่วไปจะเหลือประมาณ 8-10 Kbps ต่อช่องสัญญาณเสียงแล้วจึง บรรจุลงใน IP Packet แล้วส่งผ่านระบบไป และ VoIP ยังเป็นการยังเป็นการส่งข้อมูลเสียงแบบ 2 ทางบนระบบเครือข่ายแบบ packet-switched IP network. ซึ่งข้อมูลนี้จะถูกส่งผ่านเครือข่ายเพื่อสื่อสารระหว่าง VoIP ด้วยกัน



รูปที่ 2.17 การสื่อสารด้วยเครือข่าย VoIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 รูปแบบการใช้งานของ VoIP

สามารถแบ่งได้ 3 วิธีด้วยกันคือ

1. จากเครื่องคอมพิวเตอร์ไปสู่เครื่องคอมพิวเตอร์ (PC-to-PC)

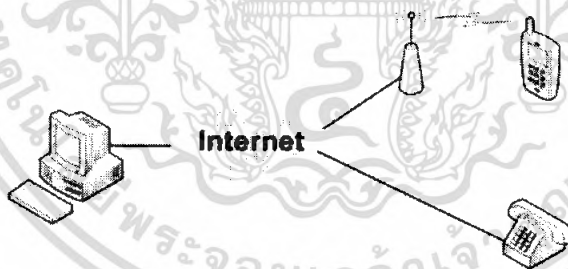
คอมพิวเตอร์ มีการติดตั้ง Sound card และไมโครโฟน ที่เชื่อมต่ออยู่กับเครือข่าย IP การประยุกต์ใช้ PC และ IP-enabled telephones สามารถสื่อสารกันได้แบบจุดต่อจุด หรือ แบบจุดต่อหลายจุด โดยอาศัย software ทางด้าน IP telephony



รูปที่ 2.18 การใช้งาน VoIP แบบ PC-to-PC

2. จากเครื่องคอมพิวเตอร์สู่เครื่องโทรศัพท์ (PC-to-Phone)

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

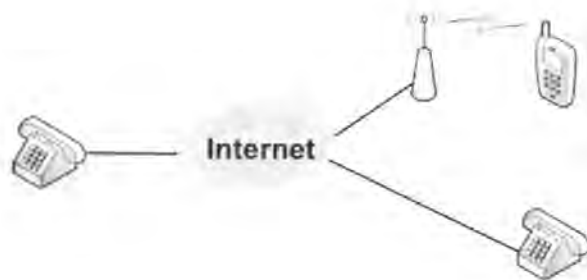


รูปที่ 2.19 การใช้งาน VoIP แบบ PC-to-Phone

3. จากเครื่องโทรศัพท์สู่เครื่องโทรศัพท์ (Phone-to-Phone)

เป็นการใช้โทรศัพท์ธรรมดา ติดต่อกับโทรศัพท์ธรรมดา แต่ในกรณีนี้จริงๆแล้วประกอบด้วยขั้นตอนการส่งเสียงบนเครือข่าย Packet ประเภทต่างๆซึ่งการใช้โทรศัพท์ร่วมกับเครือข่ายข้อมูลจำเป็นต้องใช้ VoIP gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้



รูปที่ 2.20 การใช้งาน VoIP แบบ Phone-to-Phone

2.2.2 กระบวนการทำงานของ VoIP

เมื่อมีการรับสัญญาณเสียงเข้ามาในตอนแรกนั้น จะต้องมีการแปลงสัญญาณเสียงที่เป็นรูปแบบของสัญญาณอนาล็อก ให้เป็นรูปแบบดิจิทัล (Analog to Digital : A/D) เสียก่อน แล้วจึงทำการแบ่งและจัดรูปของสัญญาณดิจิทัล นั้นขึ้นมาใหม่ให้อยู่ในรูปของเฟรม หลังจากนั้นจะทำการแปลงเฟรมให้อยู่ในรูป packet โดยมีการเพิ่ม Header เข้าไปใน packet และหลังจากที่ข้อมูลอยู่ในรูปแบบ packet เรียบร้อยแล้ว ก็จะถูกนำมาวิเคราะห์และใส่ค่า IP Address ปลายทางต่อนั้นจึงทำการส่งข้อมูล และเมื่อทางปลายทางได้รับข้อมูลแล้วนั้นก็ทำการนำ packet นั้นมาแยก Header ออกเพื่อให้เหลือเพียง Voice Frame เท่านั้น หลังจากนั้นจึงทำการแปลงสัญญาณดิจิทัลนี้กลับมาเป็นสัญญาณอนาล็อก (Digital to Analog : D/A) เพื่อให้ได้ยินอีกครั้ง

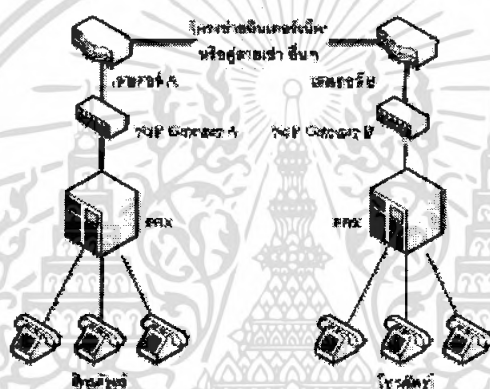


รูปที่ 2.21 กระบวนการทำงานของ VoIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การประยุกต์ใช้ VoIP ภายในองค์กร

องค์กรใหญ่ๆส่วนมากจะมีการเชื่อมต่อเครือข่าย WAN หรือ LAN ของตนกับสำนักงานหรือสาขาต่างๆ เข้าด้วยกันเพื่อรองรับเครือข่ายคอมพิวเตอร์อยู่แล้ว ซึ่งเทคโนโลยี VoIP นั้นสามารถใช้อุปกรณ์ Router/Gateway VoIP ในการแปลงสัญญาณโทรศัพท์แล้วใช้ประโยชน์จากเครือข่าย WAN หรือ LAN นั้นได้ ซึ่งแนวทางนี้สามารถช่วยลดค่าใช้จ่ายในการโทรศัพท์หากันระหว่างสำนักงานได้ แต่ยังไม่สามารถรองรับการโทรออกไปยังหมายเลขภายนอกองค์กร หรือรับสายเรียกเข้าจากภายนอกได้ ซึ่งในกรณีนี้จำเป็นต้องเชื่อมต่ออุปกรณ์ PABX เข้ากับเครือข่ายโทรศัพท์พื้นฐานอยู่ตามปกติ ซึ่งอีกวิธีคือเปลี่ยนโทรศัพท์ทั้งหมดให้เป็นโทรศัพท์ IP ซึ่งจะต้องทำพร้อมๆกับการติดตั้งอุปกรณ์คอมพิวเตอร์เซิร์ฟเวอร์ที่ทำหน้าที่บริหารจัดการบริการด้านโทรศัพท์ (Call Manager) ซึ่งในกรณีนี้ไม่จำเป็นต้องมีตู้สาขา PABX



รูปที่ 2.22 การใช้เครือข่าย IP ร่วมกับ VoIP เพื่อลดต้นทุนวงจรเชื่อมต่อระหว่างองค์กร

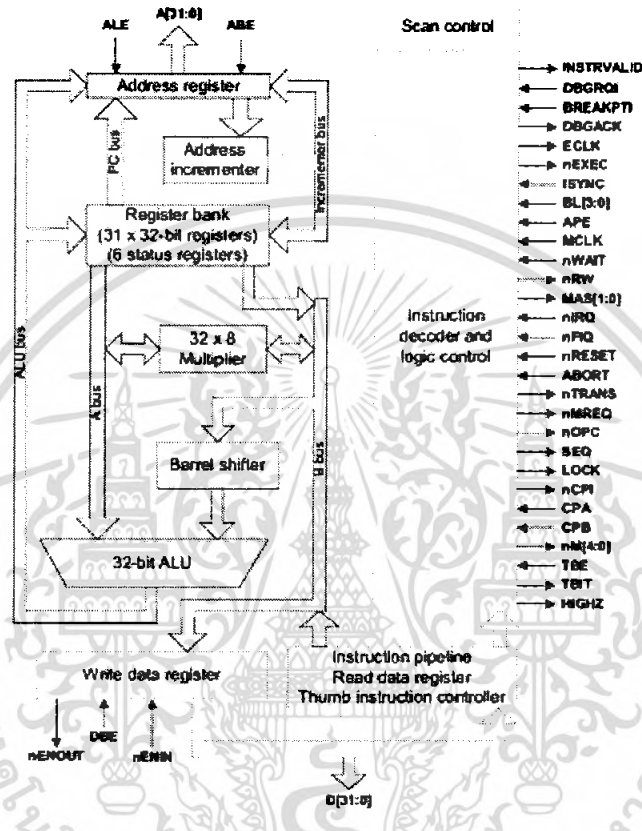
2.3 สถาปัตยกรรม ซีพียู ARM 7

สถาปัตยกรรมของ ARM7 เป็นซีพียูแบบ RISC ขนาด 32 บิต ภายในมีบัสขนาด 32 บิตตัวเดียวที่ใช้สำหรับรับส่งข้อมูล และคำสั่ง ชุดคำสั่งจะมีขนาด 32 บิตคองที่ ในขณะที่ข้อมูลสามารถเลือกได้ว่าจะมีขนาด 8, 16 หรือ 32 บิต โดยแสดงแกนกลาง (core) ของซีพียู ARM7 ได้ดังรูปที่ 2.23

โครงสร้างของ ARM7 จะเป็นแบบเรียบง่าย มีชุดคำสั่งไม่มากนัก ประหยัดพื้นที่ สารกึ่งตัวนำที่ใช้สร้างประหยัดพลังงาน

สถาปัตยกรรมของ ARM7 จะเป็นแบบ load-and-store ในการประมวลผลข้อมูลใดๆต้องกระทำผ่านทางรีจิสเตอร์เริ่มต้นด้วยการ โหลดค่าจากหน่วยความจำเก็บในรีจิสเตอร์นำค่ามาประมวลผล เสร็จแล้วจะเขียนค่าเก็บในหน่วยความจำดั้งเดิม

รีจิสเตอร์ของ ARM7 ที่ใช้ได้สำหรับผู้ใช้มีทั้งหมด 16 ตัวคือ R0-R15 โดยทุกตัวมีขนาด 32 บิต โดย R0-R12 เป็นรีจิสเตอร์ทั่วไปที่ไม่ได้กำหนดหน้าที่การทำงานพิเศษ ส่วน R12 ทำหน้าที่เป็น stack pointer (SP) R14 ทำหน้าที่เป็น link register (LR) และ R15 ทำหน้าที่เป็น Program Counter (PC)



รูปที่ 2.23 แกนกลางของซีพียู ARM7

2.3.1 ARM7 TDMI

เป็นไมโครคอนโทรลเลอร์ตระกูล ARM7 ที่ภายในมีแกนกลางเป็นซีพียู ARM7 ที่เพิ่มความสามารถอีก 4 ประเภท ที่นำตัวอักษรมาเขียนเป็นชื่อย่อได้แก่

- T: สนับสนุนคำสั่ง 16 บิตที่มีชื่อว่า Thumb instruction set
- D: สนับสนุนการดีบัก (debug)
- M: สนับสนุนการคูณแบบยาว (long multiplies)
- I: มีโมดูล Embedded ICE เพื่อสนับสนุนการดีบักภายในซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thumb mode (T)

ชุดคำสั่งของ ARM มีขนาด 32 บิตในซีพียู ARM7TDMI จะสนับสนุนชุดคำสั่งประเภทที่สอง ที่บีบอัดคำสั่งให้มีขนาด 16 บิต เรียกว่า Thumb instruction set เมื่อทำงานในโหมดนี้จะทำงานกับหน่วยความจำขนาด 16 บิตได้เร็วขึ้น และบีบอัดโปรแกรมให้มีขนาดเล็กลงทำให้สามารถนำ ARM7TDMI ไปใช้งานกับสมองกลฝังตัวได้ดี อย่างไรก็ตาม Thumb mode มีข้อจำกัดคือ เมื่อใช้กับงานประเภทเดียวกันจะใช้จำนวนคำสั่งมากกว่าโหมด 32 บิต ทำให้ทำงานช้ากว่า ดังนั้นสำหรับงานความเร็วสูงยังคงต้องใช้โหมด 32 บิตปกติ

ประการที่สองใน Thumb instruction set ไม่มีคำสั่งที่จำเป็นสำหรับจัดการกับเอ็กเซพชัน (exception handling)

Long multiplies (M)

ในชุดคำสั่งของ ARM7TDMI มีการเพิ่มคำสั่งพิเศษอีก 4 คำสั่งที่สามารถคูณเลขขนาด 32 x 32 บิตได้ผลลัพธ์เป็น 64 บิต และการคูณสะสมค่า (multiplication accumulation: MAC) โดยสามารถคูณข้อมูลขนาด 32 x 32 บิตจำนวนหลายชุดได้ผลลัพธ์เป็น 64 บิต ทำให้สามารถทำการคำนวณคณิตศาสตร์ที่ซับซ้อนได้ โดยไม่จำเป็นต้องใช้ชิปประมวลผลสัญญาณดิจิทัล (Digital Signal Processor: DSP) ช่วย

Debugging (D)

ภายในมีส่วนขยายของฮาร์ดแวร์เพื่อรองรับการดีบั๊กโปรแกรมได้ในขณะที่ทำงาน ซึ่งทำงานกับพอร์ต JTAG และ TAG controller

Embedded ICE (I)

ส่วนของ Embedded ICE ช่วยเพิ่มฟังก์ชันการทำการการดีบั๊กโปรแกรม และภายในโมดูลนี้มีเบรกพอยต์และวอล์ทพอยต์รีจิสเตอร์ ทำให้สามารถพักการทำงานของโปรแกรมเพื่อดีบั๊กการทำงาน เราสามารถควบคุมรีจิสเตอร์เหล่านี้ผ่านทาง JTAG test port และ ซอร์ฟแวร์ดีบั๊กกึ่งทูลล์ที่ทำงานบนเครื่องคอมพิวเตอร์ เมื่อพบเบรกพอยต์หรือวอล์ทพอยต์ ตัวซีพียูจะหยุดการทำงาน และเข้าสู่สถานะดีบั๊ก เมื่ออยู่ในสถานะดีบั๊กจะสามารถดูค่าของรีจิสเตอร์หรือค่าของหน่วยความจำทั้งแบบ Flash/EEPROM, SRAM และค่าของรีจิสเตอร์ที่จัดเทียบตำแหน่งกับหน่วยความจำ (Memory Mapped Registers)

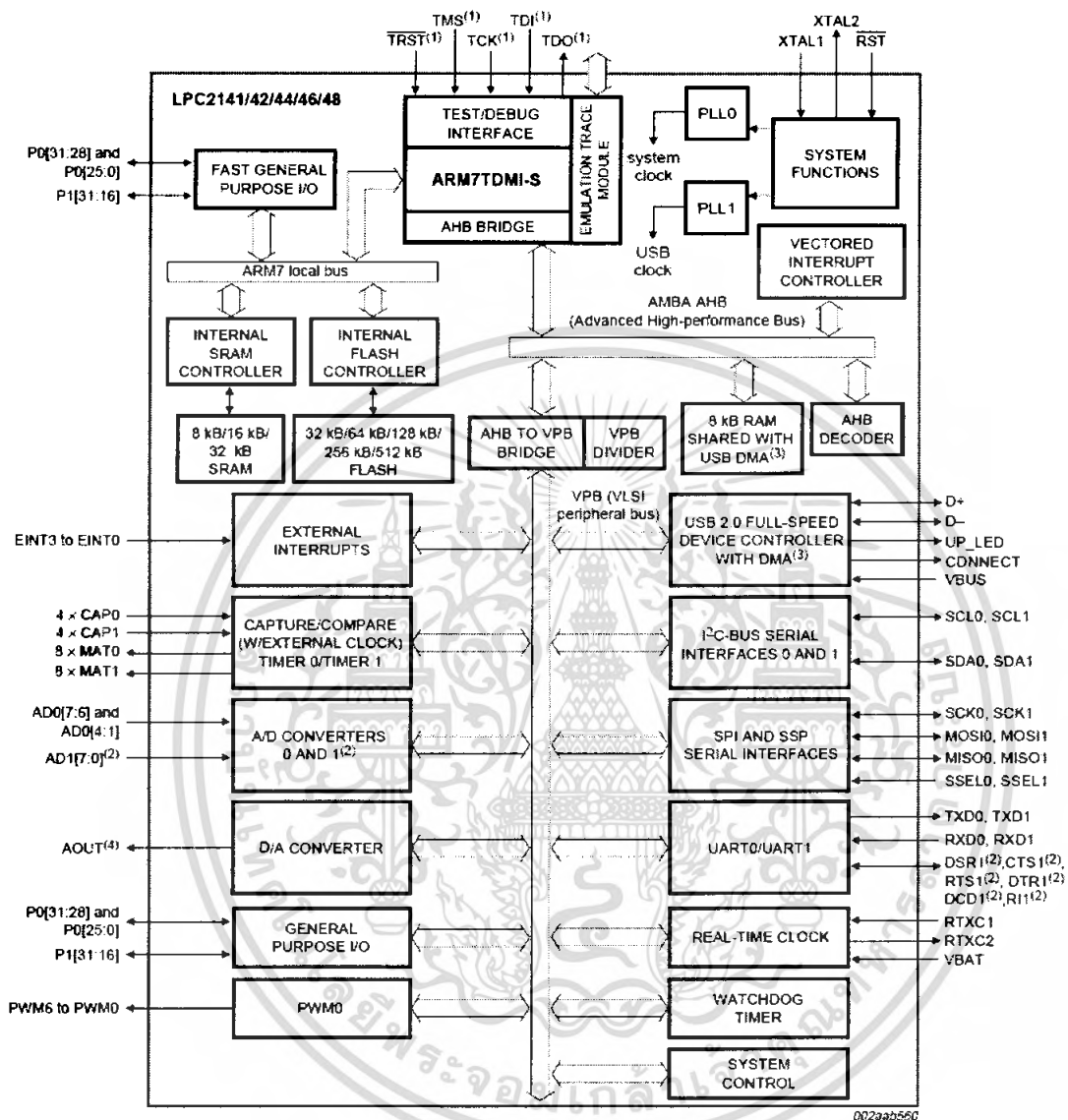
2.3.2 ไมโครคอนโทรลเลอร์ ARM7 Philips LPC2148

บริษัท Philips ได้นำซีพียูตระกูล ARM7 มาพัฒนาเพิ่มหน่วยความจำภายใน และเพิ่มโมดูลอุปกรณ์ต่างๆ จำนวนมากที่จำเป็นต่อการใช้งาน ทำให้กลายเป็นไมโครคอนโทรลเลอร์ที่สามารถทำงานได้ด้วยตัวเอง โดยไม่ต้องใช้อุปกรณ์อื่นๆเพิ่มโดยตั้งชื่อเป็นตระกูล LPC ซึ่งสรุปความสามารถของไมโครคอนโทรลเลอร์ ARM7 Philips LPC2148 ได้ดังนี้

- ไมโครคอนโทรลเลอร์ ขนาด 16/32 บิต ARM7TDMI-S ในตัวถัง LQFP 64 ขา
- หน่วยความจำ Static RAM ขนาด 8 kB-40 kB โดย LPC2148 มีขนาด 40 kB

- หน่วยความจำ Flash Program Memory ขนาด 32 kB – 512 kB อยู่ภายในชิป ที่สามารถ ลบ/เขียนซ้ำได้ถึง 10,000 ครั้ง ถ้าเป็น LPC2148 จะมีขนาด 512 kB
- โปรแกรมชิปได้ทันทีผ่าน In-System Programming (ISP) และ In-Application Programming (IAP) โดยใช้ซอฟต์แวร์ boot-loader ที่อยู่ภายในชิป
- ตัวควบคุมอุปกรณ์ USB 2.0 Full-speed โดยมี RAM สำหรับ endpoint ขนาด 2 kB ถ้าเป็น LPC2148 จะมี RAM 8 kB สำหรับการติดต่อ DMA
- วงจรแปลงอนาลอกเป็นดิจิทัลความละเอียด 10 บิต จำนวน 2 ชุด ที่รับอินพุตได้ถึง 14 อินพุต โดยมีเวลาในการแปลงค่าต่ำถึง 2.44 us
- วงจรแปลงดิจิทัลเป็นอนาลอกความละเอียด 10 บิต 1 ตัว
- วงจรไทมเมอร์ขนาด 32 บิต 2 ชุด (มี 4 capture และ 4 compare channel)
- PWM (Pulse width modulation) 6 เคาต์พุต
- โมดูลนาฬิกาเวลาจริง (Real Time Clock) ที่สามารถต่อกับคริสตอลความถี่ 32 kHz และแบตเตอรี่ภายนอกได้ และวอชด็อกซ์ (Watchdog)
- วงจรสื่อสารอนุกรม UART (16C550) จำนวน 2 ชุด
- วงจรสื่อสารอนุกรม I²C ความเร็วสูง (400 kbits/s)
- วงจรสื่อสารอนุกรม SPI และ SSP
- มีวงจร Phase lock loop ภายในเพื่อคุมค่าให้สัญญาณนาฬิกาภายในทำงานที่ความถี่สูงสุดถึง 60 MHz
- Vectored Interrupt Controller ที่สามารถกำหนดลำดับความสำคัญ และกำหนดแอดเดรสของเวกเตอร์ได้
- ใช้กับแหล่งจ่ายไฟชุดเดียวขนาด 3.0 ถึง 3.6 (3.3±10%)
- มี I/O pin อนุกรมประสงค์ที่สามารถใช้กับระดับแรงดัน 5 V ได้สูงสุด 45 ขา โดยสามารถจัดเป็นขาอินพุต/รับต์จากภายนอกได้สูงสุด 21 ขา
- มีโหมดประหยัดพลังงาน 2 โหมด ได้แก่ Idle และ Power-down

2.3.3 บล็อกไดอะแกรมของ LPC2148



- (1) Pins shared with GPIO.
- (2) LPC2144/46/48 only.
- (3) USB DMA controller with 8 kB of RAM accessible as general purpose RAM and/or DMA is available in LPC2146/48 only.
- (4) LPC2142/44/46/48 only.

รูปที่ 2.24 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ LPC2148

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.24 ส่วนบนเป็นไมโครคอนโทรลเลอร์ ARM7TDMI-S ซึ่งเป็นหัวใจหลักทางด้านซ้ายมือที่เป็น ส่วนของ ARM 7 Local Bus ที่ใช้ในการติดต่อกับหน่วยความจำแบบ Flash ที่ใช้เก็บโปรแกรม และหน่วยความจำ SRAM ที่ใช้ในการติดต่อกับหน่วยความจำภายนอกมีการติดต่อผ่านบัส AMBA AHB (Advance High-performance Bus) ซึ่งใน LPC2148 ไม่สามารถติดต่อกับหน่วยความจำภายนอกได้

ในการติดต่อกับอุปกรณ์รอบข้างเช่น GPIO, I²C, SPI, UART ฯลฯ จะติดต่อผ่านทางบัส VPB (VLSI peripheral BUS) ซึ่ง VPB บัสนี้ต่อกับ AHB บัสผ่าน AHB to VPB Bridge โดยสามารถปรับลดค่าความถี่ของ VPB บัสให้ทำงานช้ากว่าความถี่ขีพยูได้เพื่อให้ทำงานร่วมกับอุปกรณ์เสริมต่างๆที่มีความเร็วต่ำกว่าได้

2.3.4 การจัดการหน่วยความจำของ LPC2148

เนื่อง ARM7 เป็นซีพียูขนาด 32 บิต ที่มีขาแอดเดรสต่อกับหน่วยความจำ จำนวน 32 เส้น ทำให้สามารถ อ้างอิงหน่วยความจำได้ถึง 4 GB ($2^{32} = 4 \text{ GB}$) ตัวแกนหลักของ ARM7TDMI จะมีสถาปัตยกรรมแบบ Von Neumann ที่ใช้บัสขนาด 32 บิตชุดเดียวกันสำหรับตัวคำสั่งของโปรแกรม และข้อมูล โดยมีคำสั่ง load, store และ swap เท่านั้นที่ใช้ในการเรียกข้อมูลที่เก็บในหน่วยความจำ การติดต่อกับพอร์ตอินพุตหรือเอาต์พุต ก็จะใช้คำสั่ง เดียวกันกับการใช้คำสั่งจัดการเกี่ยวกับหน่วยความจำในไมโครคอนโทรลเลอร์ LPC2148 ได้จัดสรร หน่วยความจำ โดยรวมดังรูปที่ 2.25

เมื่อซีพียูถูกรีเซ็ตจะเริ่มค้นหาที่หน่วยความจำ 0x000 000

จากหน่วยความจำทั้งหมด 4 GB ได้แบ่งออกเป็น 4 ส่วนโดยใน 1GBแรก (แอดเดรส 0x000 0000 – 0x3FFF FFFF) จัดเป็นส่วนของหน่วยความจำสำหรับเก็บโปรแกรม ซึ่งกรณีของ LPC2148 ก็คือหน่วยความจำ Flash Memory ขนาด 512 kB ซึ่งมีแอดเดรสจาก 0x0000 0000 – 0x0007 FFFF

หน่วยความจำในช่วง 1 GB ถึง 2 GB (แอดเดรส 0x4000 0000 – 0x7FFF FFFF) จัดเป็นส่วนของ หน่วยความจำ RAM ใน LPC2148 จะเป็น SRAM ขนาด 40 kB โดยแบ่งเป็นสองส่วนคือ 32 kB แรกอยู่ที่แอดเดรส 0x4000 0000 – 0x4000 7FFF และอีก 8 kB เป็นหน่วยความจำใช้สำหรับ USB สามารถนำหน่วยความจำ RAM ส่วนนี้มาใช้เป็น RAM สำหรับใช้งานทั่วไปได้ หน่วยความจำส่วนที่ใกล้กับ 2 GB จะเป็นส่วนของ Boot Block ขนาด 12 kB ซึ่งเป็นโปรแกรมที่ทำงานเพื่อเขียนโปรแกรมลงในหน่วยความจำ Flash ของ LPC2148

4.0 GB	AHB PERIPHERALS	0xFFFF FFFF
3.75 GB	VPB PERIPHERALS	0xF000 0000
3.5 GB	RESERVED ADDRESS SPACE	0xE000 0000
3.0 GB	RESERVED ADDRESS SPACE	0xC000 0000
2.0 GB	BOOT BLOCK (12 KB REMAPPED FROM ON-CHIP FLASH MEMORY)	0x8000 0000 0x7FFF FFFF
	RESERVED ADDRESS SPACE	0x7FFF D000 0x7FFF CFFF
	8 KB ON-CHIP USB DMA RAM (LPC2146/2148)	0x7FD0 2000 0x7FD0 1FFF
	RESERVED ADDRESS SPACE	0x7FD0 0000 0x7FCF FFFF
	32 KB ON-CHIP STATIC RAM (LPC2146/2148)	0x4000 8000 0x4000 7FFF
	16 KB ON-CHIP STATIC RAM (LPC2142/2144)	0x4000 4000 0x4000 3FFF
	8 KB ON-CHIP STATIC RAM (LPC2141)	0x4000 2000 0x4000 1FFF
1.0 GB	RESERVED ADDRESS SPACE	0x4000 0000 0x3FFF FFFF
	TOTAL OF 512 KB ON-CHIP NON-VOLATILE MEMORY (LPC2148)	0x0008 0000 0x0007 FFFF
	TOTAL OF 256 KB ON-CHIP NON-VOLATILE MEMORY (LPC2146)	0x0004 0000 0x0003 FFFF
	TOTAL OF 128 KB ON-CHIP NON-VOLATILE MEMORY (LPC2144)	0x0002 0000 0x0001 FFFF
	TOTAL OF 64 KB ON-CHIP NON-VOLATILE MEMORY (LPC2142)	0x0001 0000 0x0000 FFFF
	TOTAL OF 32 KB ON-CHIP NON-VOLATILE MEMORY (LPC2141)	0x0000 8000 0x0000 7FFF
0.0 GB		0x0000 0000

002aab553

รูปที่ 2.25 การจัดการหน่วยความจำของ LPC2148

หน่วยความจำช่วง 2 GB ถึง 3.5 GB (แอดเดรส 0x8000 0000 – 0xDFFF FFF) สงวนไว้สำหรับต่อกับหน่วยความจำภายนอกซึ่งกรณีของ LPC2148 ไม่ได้ใช้งาน

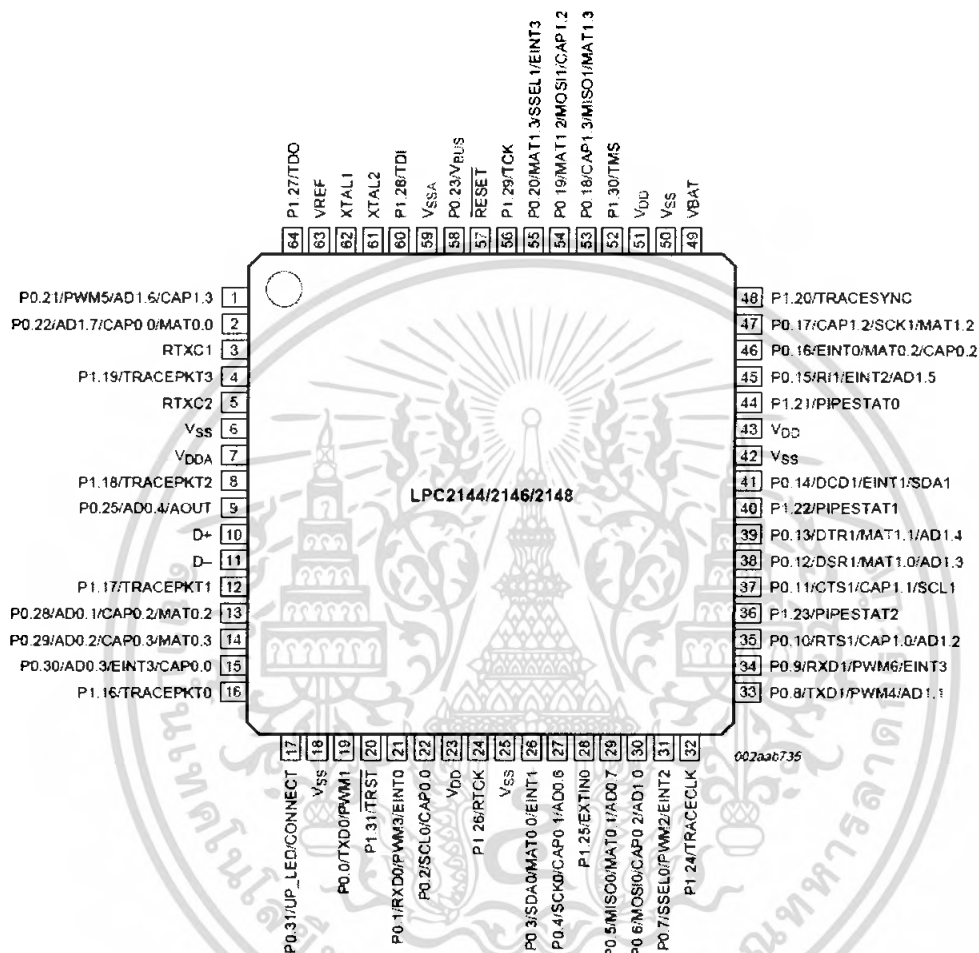
หน่วยความจำในช่วง 3.5 GB ถึง 4 GB จะเป็นพื้นที่สำหรับติดต่อกับอุปกรณ์อื่นๆที่อยู่ในชิป โดยแบ่งเป็นอุปกรณ์ที่ต่อกับ VPB บัสจะติดต่อกับหน่วยความจำช่วง 0xE000 0000 ถึง 0xEFFF FFFF

ถ้าเป็นอุปกรณ์ที่ติดต่อผ่านทาง AHB จะมีช่วงแอดเดรส 0xF000 0000 ถึง 0xFFF FFFF ซึ่งในกรณีของ LPC2148 ไม่สามารถติดต่อกับหน่วยความจำภายนอก จึงไม่ได้ใช้หน่วยความจำส่วนนี้

อุปกรณ์เสริมต่างๆเช่น UART, RTC ฯลฯ ที่ติดต่อกับซีพียูผ่าน VPB bus มีชื่อเรียกว่า VPB peripheral#0 – VPB peripheral#127 โดย VPB peripheral แต่ละตัวจะมีช่วงของหน่วยความจำอุปกรณ์ละ 16 KB ทำให้คิดเป็นหน่วยความจำรวม 2 MB ในการติดต่อกับอุปกรณ์เสริมเหล่านี้ ใช้คำสั่งเดียวกันกับที่ใช้ในการติดต่อกับหน่วยความจำ

2.3.5 การจัดขาของ LPC2148

ไมโครคอนโทรลเลอร์ LPC2148 อยู่ในตัวถังแบบ plastic low profile quad flat package (LQFP64) ที่มีขาต่อจำนวน 64 ขาโดยมีการจัดขาแสดงในรูปที่ 2.26



รูปที่ 2.26 การจัดขาของ LPC2148

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

MCS-51 เป็นไมโครคอนโทรลเลอร์ที่ออกแบบมาเพื่อสนองความต้องการของผู้ใช้แบบสำเร็จในไอซีตัวเดียว คือมีสายสัญญาณอินพุต/เอาต์พุตภายในตัว พอร์ตของอินพุตและเอาต์พุตบัพเฟอร์ที่ใช้เชื่อมต่อกับวงจรภายนอก (interface) และสายสัญญาณควบคุมอื่นๆ ที่ใช้สำหรับแยกสายสัญญาณข้อมูลกับสายสัญญาณกำหนดตำแหน่งหน่วยความจำ และยังมีชุดคำสั่งพิเศษเพื่อจัดการข้อมูลเพิ่มขึ้นอีก

2.4.1 คุณสมบัติของMCS-51

- ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
- มีพอร์ตI/O ขนาด 8บิต 4 พอร์ต
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 Kbytes
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 Kbytes
- หน่วยความจำสำหรับเก็บข้อมูลและโปรแกรมแยกจากกันอย่างละ 64 Kbytes
- มีไทม์เมอร์และ เคนต์เตอร์ขนาด 16บิต 2ชุด ทำงานได้ 4 โหมด
- รับอินเตอร์รัปต์ได้ 6 แหล่ง 5 เวกเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปแบบได้ 4 โหมด มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต
- มีวงจรรอสซิงเกิลเตอร์และวงจรรนาฬิกาบนชิพ
-

2.4.2 การจัดตำแหน่งขาของ MCS-51 และหน้าที่การทำงาน

P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/PP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

รูปที่ 2.27 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 0

มีทั้งหมด 8 บิตคือ P0.0-P0.7 ใช้งานได้ 2หน้าที่ คือเป็นทางผ่านของแอดเดรสบัสและคาส์บัส เมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือ I/O

Port 1

มีขา 8 ขา คือ P1.0-P1.7 แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อ

Port 2

มีทั้งหมด 8 บิต คือขา P2.0- P2.7 พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A8-A15) และเป็น I/O port ใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุตพอร์ตต้องส่งลอจิก “1” มาที่พอร์ตนี้

Port 3

มีขา 8 ขาคือ P3.0-P3.7 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อกับ นอกจากนี้ขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นตอนต่อไปนี้

ตารางที่ 2.7 หน้าที่ของขาพอร์ต 3

P3.0/RxD	เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)
P3.1/TxD	เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)
P3.2/INT0	ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0
P3.3/INT1	ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1
P3.4/T0	ขารับสัญญาณพัลส์อินพุตเข้าไปยังวงจร Counter 0
P3.5/T1	ขารับสัญญาณพัลส์อินพุตเข้าไปยังวงจร Counter 1
P3.6/WR	ขาสัญญาณใช้ในการ เขียนข้อมูล ลงหน่วยความจำข้อมูลภายนอก
P3.7/RD	ขาสัญญาณใช้ในการ อ่านข้อมูล ลงหน่วยความจำข้อมูลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PSEN (Program Store Enable)

ขานี้ทำงานที่สภาวะลอจิกเป็น “0” โดยจะอ่านค่าจากหน่วยความจำภายนอกที่เป็นข้อมูล แต่ถ้าเป็นการอ่านข้อมูลจากหน่วยความจำภายใน ขานี้จะไม่ได้ใช้งานและมีค่าลอจิกเป็น “1”

ALE (Address Latch Enable)

เป็นขานี้ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้น ขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการ โปรแกรมสำหรับ โปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

EA (External Access)

ทำหน้าที่เลือกการทำงานของหน่วยความจำโดยจะเลือกอ่านโปรแกรมจากภายในชิพเมื่อลอจิกเป็น “1” และจะเลือกอ่านโปรแกรมจากภายนอกชิพเมื่อลอจิกเป็น “0”

RST (Reset)

จะรีเซ็ตระบบได้ก็ต่อเมื่อส่งค่าลอจิก “1” เข้าที่ขานี้อย่างน้อย 2 Machine Cycles

ขาสัญญาณนาฬิกา

ทำหน้าที่เป็นตัวกำหนดสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์ใช้เป็นฐานเวลาในการทำงาน โดยใช้แผ่นผลึก Crystal ร่วมกับตัวเก็บประจุ

2.5 ระบบเรียลไทม์ซิสเต็ม (real time system)

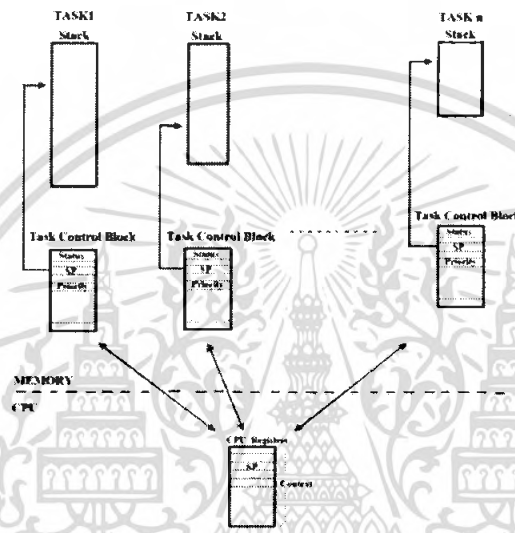
ระบบเรียลไทม์ซิสเต็ม (real time system) หรือระบบทันเวลา เป็นระบบการทำงานที่ไม่สามารถคาดการณ์ได้ว่าช่วงเวลาของอินพุตที่เข้ามาจะเกิดขึ้นเมื่อไหร่ และต้องทำงานให้ได้ผลลัพธ์ออกไปให้ทันเวลา ซึ่งเรียลไทม์ซิสเต็มสามารถแบ่งออกได้เป็น 2 ประเภท คือ ซอฟต์แวร์เรียลไทม์ซิสเต็ม (Soft real time system) และ ฮาร์ดเรียลไทม์ซิสเต็ม (Hard real time system) ซอฟต์แวร์เรียลไทม์ซิสเต็ม นั้นเป็นเรียลไทม์ซิสเต็มที่อาจจะมีความล่าช้าได้บ้าง ซึ่งจะไม่ส่งผลกระทบต่อระบบเช่นระบบการแพทย์ การบิน การจราจร ซึ่งอาจจะล่าช้าได้หลายวินาที โดยไม่เกิดความเสียหายอะไร ส่วนฮาร์ดเรียลไทม์ซิสเต็มนั้นระบบจะต้องทำงานให้ถูกต้องและต้องไม่เกินตามเวลาที่กำหนดไว้ ซึ่งความล่าช้าที่เกิดขึ้นในระบบนั้นเป็นสิ่งที่ยอมรับไม่ได้และอาจจะส่งผลกระทบให้เกิดความเสียหายต่อระบบได้

IP phone นั้นจัดเป็นระบบแบบฮาร์ดเรียลไทม์ซิสเต็ม เนื่องจากต้องส่งเสียงให้ทันตามเวลาที่กำหนด ถ้าไม่ทันจะทำให้ระบบทำงานต่อไม่ได้ หรือทำให้ข้อมูลข่าวสารนั้นสูญหายไป ซึ่งในสถานการณ์จริงแล้วการที่เราได้ยินเสียงไม่ครบทุกคำพูดนั้น ผู้พูดและผู้ฟังอาจจะยังสื่อสารเข้าใจกันได้อยู่ แต่ถ้าการสื่อสารนั้นเป็นข่าวสารที่สำคัญการที่ผู้พูดและผู้ฟังได้ยินเสียงไม่ครบถ้วนนั้นอาจทำให้เกิดความเสียหายอย่างมากได้

2.5.1 Task และ Multitasking

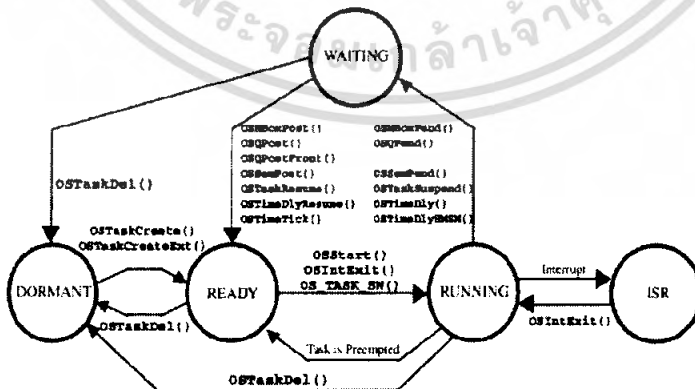
2.5.1.1 Task

ทาสก์ (Task) หรือเรียกว่าเธรด (Thread) คือ ตัวโปรแกรมที่คิดว่ามีซีพียูเป็นของตัวเองทั้งหมด การออกแบบการทำงานของเรียลไทม์แอปพลิเคชัน จะเริ่มจากการแยกหน้าที่ให้แก่ทาสก์รับผิดชอบสำหรับการใช้แก้ปัญหาต่างๆ โดยในแต่ละทาสก์จะกำหนดค่าความสำคัญ (priority), กำหนดคอนเท็กซ์ (context) ภายในรีจิสเตอร์ (register) ของซีพียู และกำหนดพื้นที่สแตก (stack) ของแต่ละทาสก์



รูปที่ 2.28 มัลติเทิลทาสก์ (Multiple Task)

ในแต่ละทาสก์มักจะอยู่ในสถานะซึ่งจะเป็น 1 ใน 5 สถานะ ได้แก่ DORMANT, READY, RUNNING, WAITING (สำหรับสถานการณ์ต่างๆ) และ ISR



รูปที่ 2.29 สถานะของทาสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถานะ DORMANT จะทำหน้าที่สร้างทาสก์ขึ้นมา (แต่ยังไม่พร้อมที่จะทำงาน) และทำลายทาสก์
- ทาสก์จะเป็นสถานะ READY เมื่อทาสก์ที่สร้างมาสามารถพร้อมจะทำงาน แต่ ความสำคัญมีค่าต่ำกว่าทาสก์ที่กำลังทำงานอยู่ในสถานะ RUNNING
- สถานะ RUNNING เป็นสถานะที่ทาสก์กำลังทำงานโดยเข้าใช้งานโปรเซสเซอร์
- ทาสก์จะเป็นสถานะ WAITING หรือ BLOCKING เมื่อทาสก์ต้องการให้อีเวนต์เกิดขึ้น เช่น การรอที่จะให้ I/O ทำงานเสร็จ, รอจะใช้งานแชร์รีซอสหรือการรอให้เกิดพัลส์ (pulse) เป็นต้น
- ทาสก์จะเป็นสถานะ ISR เมื่อมีอินเทอร์รัพต์เกิดขึ้น และซีพียูก็จะไปทำงานในส่วนของอินเทอร์รัพต์

2.5.1.2 Multitasking

มัลติทาสกิง (Multitasking) คือกระบวนการจัดลำดับการทำงาน (scheduling) และสวิตชิง (switching) ของ CPU (Central Processing Unit) ระหว่างหลาย ๆ ทาสก์โดยซีพียูตัวเดียวจะพิจารณาหลาย ๆ ลำดับของทาสก์ซึ่งมัลติทาสกิงจะคล้ายกับฟอร์กราวด์/แบ็กกราวด์ด้วย โดยจะเป็นในรูปแบบหลาย ๆ แบ็กกราวด์ การทำมัลติทาสกิงเป็นการใช้ประโยชน์ของซีพียูให้คุ้มค่าที่สุด และยังเป็นการจัดเตรียมองค์ประกอบให้กับแอปพลิเคชันอีกด้วย ลักษณะสำคัญอีกประการหนึ่งของมัลติทาสกิงคือสามารถให้โปรแกรมเมอร์ ผู้พัฒนาแอปพลิเคชันสามารถจัดการความซับซ้อนที่มีในตัวเรียล ไทม์แอปพลิเคชันได้โดยไม่ต้องคำนึงถึงแอปพลิเคชันตัวอื่น ทำให้แอปพลิเคชันโปรแกรมง่ายต่อการออกแบบและดูแล

2.5.2 Kernel

เคอร์เนล (kernel) เป็นส่วนหนึ่งของระบบมัลติทาสกิง ซึ่งจะรับผิดชอบโดยทำหน้าที่ในการจัดการทาสก์ เช่น การจัดการเวลาของซีพียูและใช้ในการติดต่อระหว่างทาสก์ การให้บริการทั่วไปของเคอร์เนล คือ การทำคอนเท็กซ์สวิตชิง ซึ่งการใช้งานเรียล ไทม์เคอร์เนลทั่วไปจะใช้ในการจัดการทาสก์ โดยการออกแบบตัวระบบให้แอปพลิเคชันสามารถแบ่งออกได้หลาย ๆ ทาสก์ โดยเคอร์เนลจะมีการเพิ่มโอเวอร์เฮดเข้าไปในระบบเพราะต้องการใช้รอม (ROM) หรือพื้นที่ที่โค้ด (code space) และแรม (RAM) ในส่วนของโครงสร้างข้อมูลของเคอร์เนลเพิ่ม ซึ่งต้องระวังเพราะธรรมดาแล้ว แต่ละทาสก์จะต้องการพื้นที่สแตกเป็นของตัวเอง ซึ่งมีแนวโน้มว่าจะกินพื้นที่ในแรมสูงพอสมควร

2.5.3 Scheduler

การจัดลำดับงาน (Scheduler) หรือเรียกว่าดิสแพทเชอร์ (dispatcher) เป็นส่วนหนึ่งในหน้าที่ของเคอร์เนลที่จะช่วยตัดสินใจว่าจะให้ทาสก์ตัวไหนทำงานต่อไป ซึ่งเรียลไทม์เคอร์เนลจะคิดจากค่าความสำคัญ (priority) เป็นหลัก โดยจะให้ทาสก์ที่มีความสำคัญสูงสุดและพร้อมที่จะทำงานได้ทำงานก่อน แต่อย่างไรก็ตามการตัดสินใจว่าจะ

ให้ทาสก์ตัวไหนทำงาน จะขึ้นกับชนิดของเคอร์เนลด้วย โดยเคอร์เนลที่อาศัยค่าความสำคัญเป็นหลัก ประกอบด้วย 2 ชนิด นอนพรีเอมทิฟเคอร์เนล (non-preemptive kernel) และพรีเอมทิฟเคอร์เนล (preemptive kernel)

2.5.4 Task Priority

ความสำคัญ(priority) จะกำหนดให้ในแต่ละทาสก์โดยทาสก์ที่มีความสำคัญจะถูกกำหนดให้มีความสำคัญสูง ซึ่งแบ่งได้เป็นความสำคัญแบบสแตติก (static priority) และไดนามิก (dynamic priority) ซึ่งทาสก์จะเป็นแบบสแตติกเมื่อแต่ละทาสก์ไม่มีการเปลี่ยนความสำคัญระหว่างที่แอฟพลิเคชันกำลังทำงาน โดยแต่ละทาสก์จะถูกกำหนดความสำคัญมาตั้งแต่แรกแล้ว ส่วนความสำคัญแบบไดนามิก ทาสก์สามารถเปลี่ยนความสำคัญได้ระหว่างที่แอฟพลิเคชันกำลังทำงาน แต่อาจจะเกิดปัญหาคือไพโรอริตีอินเวอร์ชัน (priority inversion) ได้ ซึ่งในเรียลไทม์เคอร์เนลจะมีการป้องกันการเกิดไพโรอริตีอินเวอร์ชัน

2.5.5 Deadlock หรือ Deadly Embrace

เดดล็อก (Deadlock) หรือ deadly embrace คือกรณีที่ทาสก์ 2 ตัวต่างรอการใช้งานทรัพยากรที่ใช้งานโดยอีกตัว เช่น ทาสก์ T1 ใช้งานทรัพยากร R1 และทาสก์ T2 ใช้งานทาสก์ T2 ถ้า T1 ต้องการใช้งาน R2 และ T2 ต้องการใช้ R1 จะเกิดเดดล็อกขึ้น ซึ่งการแก้ปัญหาทำได้โดย

- ก่อนทำงานต้องได้รับทรัพยากรก่อน
- ให้การใช้งานทรัพยากรตามลำดับ
- ในการเลิกใช้งานทรัพยากรต้องเลิกใช้งานแบบย้อนลำดับ

เคอร์เนลทั่วไปจะอนุญาตให้กำหนดไทม์เอาต์ (Timeout) ได้เมื่อใช้งานเซมาฟอร์ซึ่งจะช่วยในการป้องกันการเกิด เดดล็อก ถ้าทาสก์ที่ต้องการใช้งานเซมาฟอร์ไม่ได้รับในเวลาที่ต้องการ ทาสก์นั้นจะกลับไปทำงานต่อ และโค้ดแสดงความผิดพลาด (error code) จะถูกส่งไปเพื่อไปแจ้งว่าเกิดไทม์เอาต์ขึ้น ซึ่งโค้ดแสดงความผิดพลาดครั้งนี้จะช่วยป้องกันการที่ทาสก์คิดว่าได้รับทรัพยากรนั้น โดยเดดล็อกมักจะเกิดกับระบบมัลติทาสก์ที่มีขนาดใหญ่มาก พบน้อยในระบบสมองกลฝังตัว (Embedded system)

2.6 ทฤษฎีการซักรตัวอย่าง (Sampling Theorem)

สัญญาณอนาลอกเป็นสัญญาณที่เกิดอย่างต่อเนื่องโดยข้อมูลจะมีความซ้ำซ้อนเกินความจำเป็นรวมอยู่ด้วยมากในระหว่างที่ทำการส่งสัญญาณ ดังนั้นถ้าทำการซักรค่าตัวอย่าง(Sampling) ของสัญญาณนั้นที่ระยะเวลาต่างๆ ที่อยู่ห่างกันอย่างเหมาะสมแล้ว ก็จะสามารถเก็บข้อมูลของสัญญาณได้อย่างสมบูรณ์ ซึ่งการเลือกซักรค่าตัวอย่างสัญญาณที่เวลาที่เหมาะสม จะทำให้สามารถลดความซ้ำซ้อนหรือความฟุ่มเฟือยของการใช้ค่าสัญญาณลงได้ และเนื่องจากตัวอย่างสัญญาณที่ซักรออกมา นั้น เกิดที่เวลาห่างกันเป็นช่วงๆ ไม่ติดต่อกันเราจึงเรียกสัญญาณนี้ว่าสัญญาณดิสครีต (Discrete signal) โดยทฤษฎีการซักรตัวอย่างนี้มีใจความดังนี้

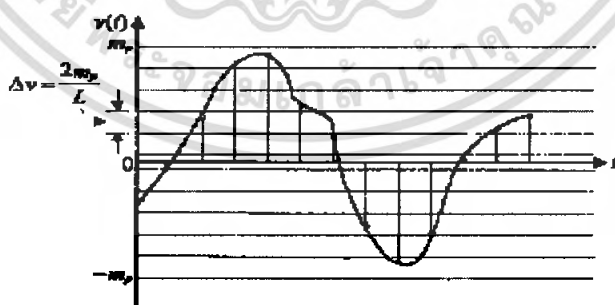
ถ้าสัญญาณใดๆที่มีฟังก์ชันมีความหนาแน่นสเปกตรัมอยู่ในช่วงความถี่ที่มีแบนด์วิดท์ไม่เกิน B เฮิรตซ์ แล้วการซัดค่าตัวอย่างของสัญญาณนั้น ที่ระยะเวลาที่ห่างกันอย่างสม่ำเสมอเป็นคาบไม่เกิน $\frac{1}{2B}$ วินาที จะทำให้สามารถกำหนดค่าสัญญาณออกนอกเดิม กลับคืนมาได้จากตัวอย่างสัญญาณเหล่านั้น

โดยค่าคาบเวลา $T = \frac{1}{2B}$ นี้มีชื่อเรียกว่า ช่วงเวลาการซัดตัวอย่างไนควิสต์ หรือ ช่วงเวลาไนควิสต์ (Nyquist interval) และค่าความถี่ $f = 2B$ นี้มีชื่อเรียกว่า ความถี่การซัดตัวอย่างไนควิสต์ (Nyquist sampling frequency) หรือ ความถี่ไนควิสต์

2.7 การควอนไทซ์สัญญาณ

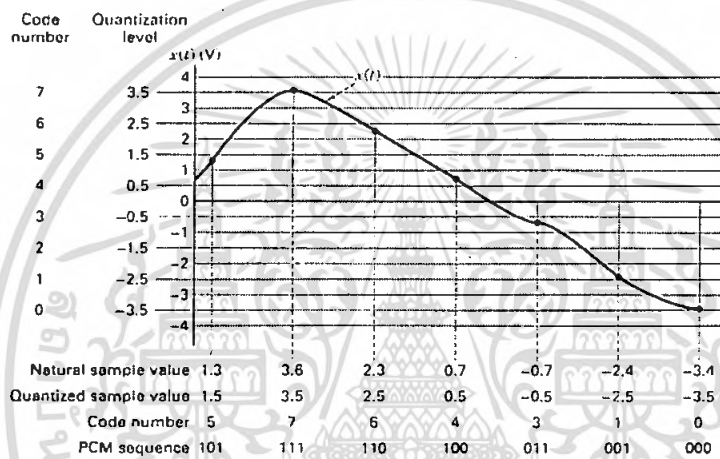
การทำควอนไทซ์คือ กระบวนการที่ทำการปรับค่าของสัญญาณที่เกิดขึ้นอย่างต่อเนื่อง ไปเป็นค่าที่มีระดับเป็นดิสคริตที่มีจำนวนจำกัด โดยค่าตัวอย่างสัญญาณอนาลอกที่ซัดค่าได้มานั้น จะมีระดับของขนาดที่มีค่าเปลี่ยนแปลงเป็นอนาลอกด้วย กล่าวคือจะมีค่าที่เป็นไปได้เป็นจำนวนนับไม่ถ้วน ดังนั้นถ้าจะกำหนดรหัสให้กับค่าตัวอย่างเหล่านี้ทั้งหมด ก็จะต้องใช้รหัสจำนวนมหาศาล ซึ่งเป็นไปไม่ได้ในทางปฏิบัติ ดังนั้นจึงต้องมีการลดจำนวนรหัสที่ใช้ลงให้เหมาะสมกับสภาวะที่จะใช้งานจริงได้ เมื่อความจำเป็นมีดังนี้ จึงจำเป็นที่จะต้องจัดแบ่งค่าขนาดของตัวอย่างสัญญาณออกเป็นกลุ่มเท่าจำนวนรหัสที่ใช้ โดยการกำหนดให้ค่าตัวอย่างสัญญาณที่มีขนาดอยู่ในกลุ่มเดียวกันให้มีรหัสเดียวกัน ก็จะทำให้เกิดการจัดแบ่งระดับเป็นระดับสัญญาณดิสคริตที่มีจำนวนจำกัดขึ้น

ถ้าสัญญาณ $m(t)$ มีค่าของขนาดอยู่ระหว่าง $(-m_p, m_p)$ หากเราต้องการแบ่งกลุ่มขนาดของค่าตัวอย่างสัญญาณออกเป็น L กลุ่ม เราอาจทำได้โดยการแบ่งช่วงขนาดของสัญญาณออกเป็น L ช่วงเท่ากัน ซึ่งจะได้ความกว้างของแต่ละช่วงเท่ากับ $\frac{2m_p}{L}$ ดังแสดงในรูปที่ 2.30 ค่าตัวอย่างสัญญาณที่มีระดับความแรงที่อยู่ในช่วงเดียวกัน ก็จะถือว่าอยู่ในกลุ่มเดียวกัน



รูปที่ 2.30 ค่าตัวอย่างสัญญาณที่ซัดค่าออกมาได้ จะถูกประมาณด้วยค่าระดับดิสคริตที่ใกล้ที่สุด

ถ้าเราใช้ระดับของขนาดที่กึ่งกลางของแต่ละช่วงเป็นระดับการทำควอนไทซ์ ค่าตัวอย่างของสัญญาณที่สุ่มได้จะถูกปรับให้มีค่าเท่ากับระดับกึ่งกลางช่วงที่มันบรรจุอยู่ ค่าความแตกต่างของการปรับค่าตัวอย่างสัญญาณที่ได้กับระดับการทำควอนไทซ์แต่ละระดับนี้ จะมีค่าอยู่ในช่วง $\left(-\frac{m_p}{L}, \frac{m_p}{L}\right)$ ค่าความแตกต่างคือ ค่าความเพี้ยน หรือค่าความคลาดเคลื่อนของการทำควอนไทซ์ ค่าตัวอย่างสัญญาณที่ผ่านการควอนไทซ์แล้วจะถูกเข้ารหัส และส่งไปยังเครื่องรับในรูปกลุ่มของบิตส์ เครื่องรับจะมีวงจรถอดรหัสเพื่อนำค่าตัวอย่างสัญญาณกลับคืน ในรูปสัญญาณดิจิทัล และวงจรกรองความถี่ต่ำผ่านจะทำการกรองตรวจจับเอาสัญญาณอนาล็อก $\hat{m}(t)$ ที่ประมาณเท่ากับสัญญาณข่าวสาร $m(t)$ กลับคืนมา

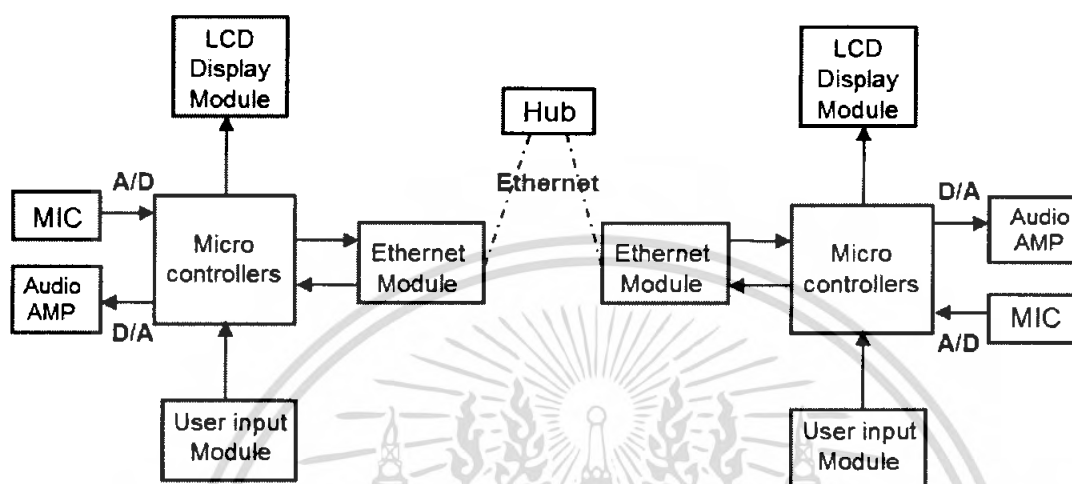


รูปที่ 2.31 ตัวอย่างการแบ่งระดับการควอนไทซ์สัญญาณ

บทที่ 3

การคำนวณและการสร้าง

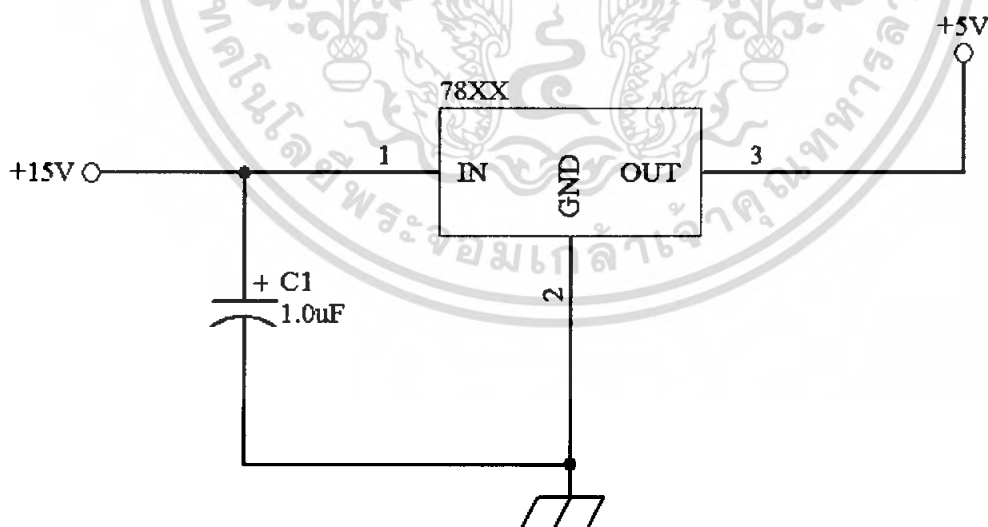
3.1 Block Diagram



รูปที่ 3.1 Block Diagram

3.2 วงจรแปลงแรงดันไฟจาก +15 V เป็น +5V

วงจรแปลงแรงดันนี้ ใช้เรกูเลเตอร์เบอร์ 7805 ซึ่งจะทำการแปลงแรงดัน ไฟจาก +15V เป็น +5V

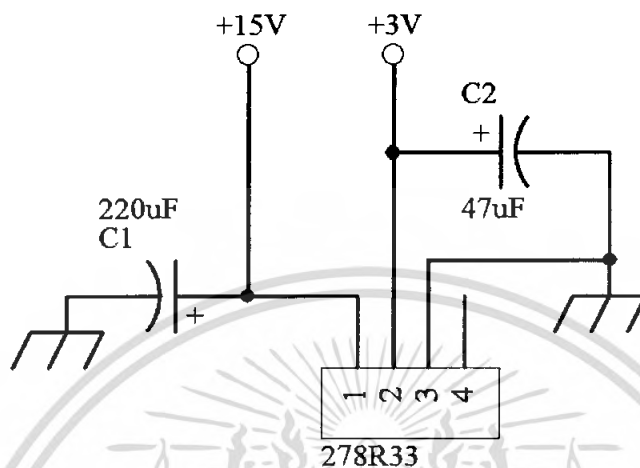


รูปที่ 3.2 วงจรแปลงแรงดันไฟจาก +15V เป็น +5V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V

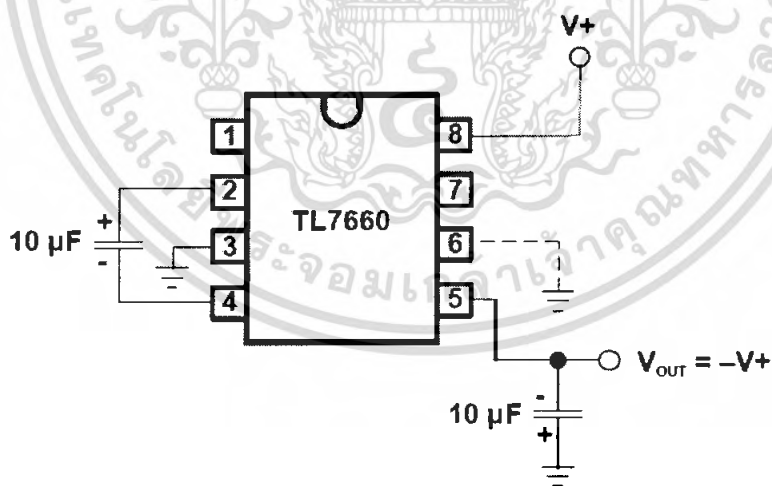
วงจรแปลงแรงดันนี้ ใช้เรกูเลเตอร์เบอร์ 278R33 ซึ่งจะทำการแปลงแรงดันไฟจาก +15 เป็น +3.3V



รูปที่ 3.3 วงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V

3.4 วงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V

วงจรแปลงแรงดันนี้ใช้ไอซี TL7660 ในการช่วยแปลงแรงดันไฟจาก +3.3V เป็น -3.3V



รูปที่ 3.4 วงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วงจร Microphone Pre-Amplifier

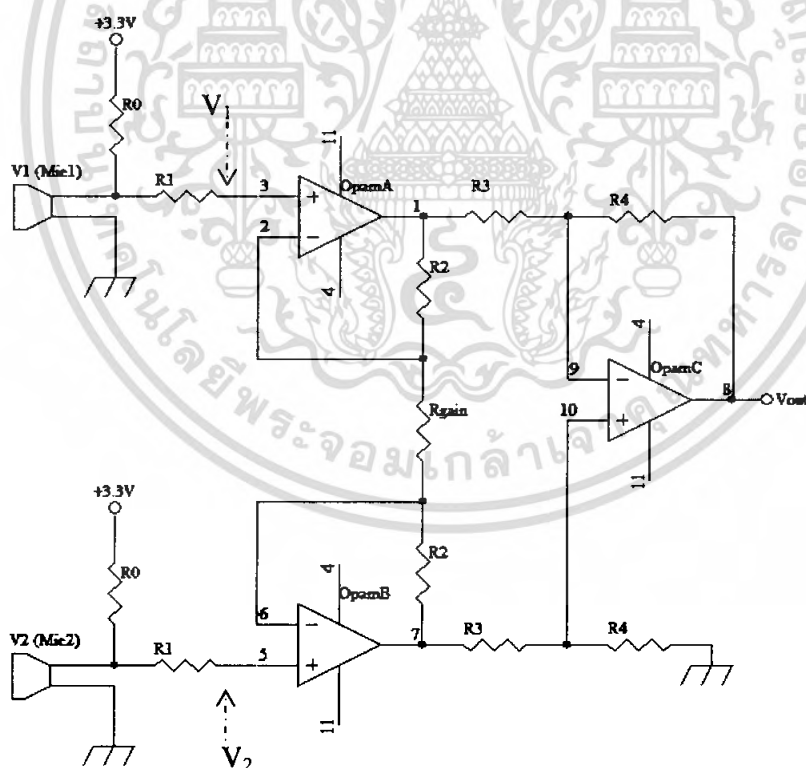
วงจรไมโครโฟนจะอยู่ในส่วนของภาคส่งทำหน้าที่เปลี่ยนสัญญาณเสียงให้เป็นสัญญาณไฟฟ้าและขยายกำลังไฟฟ้านั้นเข้าสู่ ไมโครคอนโทรลเลอร์ ARM7 ซึ่งในส่วนของวงจรไมโครโฟนนี้ จะแบ่งเป็น 4 ส่วนคือ

- Pre-microphone และ Instrument Amplifier
- Non-Inverting Summing Amplifier
- Voltage follower (Buffer)
- Second order Low Pass Filter

3.5.1 วงจร Pre-microphone และ Instrument Amplifier

ในส่วน Pre-microphone ทำหน้าที่เปลี่ยนสัญญาณเสียงเป็นสัญญาณไฟฟ้า และส่วนของวงจร Instrument Amplifier นั้น จะประกอบด้วยวงจรขยายสัญญาณผลต่าง (Differential Amplifier) และวงจรขยายสัญญาณแบบไม่กลับเฟส (Non-inverting Amplifier) โดยอัตราขยายของวงจร Instrument Amplifier คำนวณได้ตามสมการ

$$\frac{V_{out}}{V_2 - V_1} = \left(1 + \frac{2R_2}{R_{gain}}\right) \frac{R_4}{R_3}$$



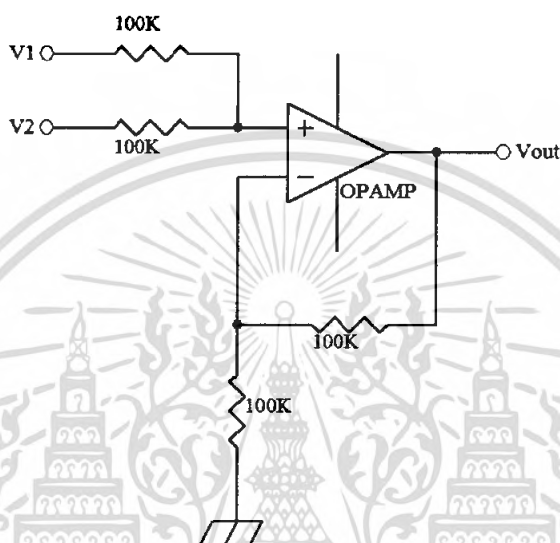
รูปที่ 3.5 วงจร Pre-microphone และ Instrument Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 วงจร Non-Inverting Summing Amplifier

วงจรขยายรวมสัญญาณแบบไม่กลับเฟส หรือ Non-Inverting Summing Amplifier นั้นทำหน้าที่รวมสัญญาณจากหลายอินพุตเข้าด้วยกัน ซึ่งสัญญาณเอาต์พุตที่ได้มันจะมีค่าเท่ากับผลรวมของอินพุตทั้งหมด โดยสามารถคำนวณได้ดังสมการ

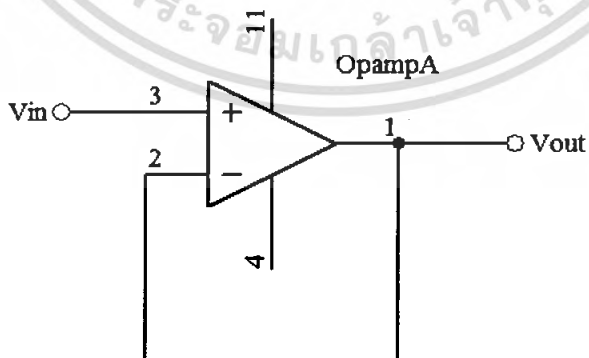
$$V_{out} = V_1 + V_2$$



รูปที่ 3.6 วงจร Non-Inverting Summing Amplifier

3.5.3 วงจร Voltage follower (Buffer)

วงจร Voltage follower หรือวงจร Buffer นั้นมีค่าเหมือนกับวงจรขยายแบบไม่กลับเฟส (Non-inverting Amplifier) ที่มีกำลังขยายเป็น 1 ซึ่งจะช่วยรักษาระดับแรงดันของสัญญาณไฟฟ้าได้



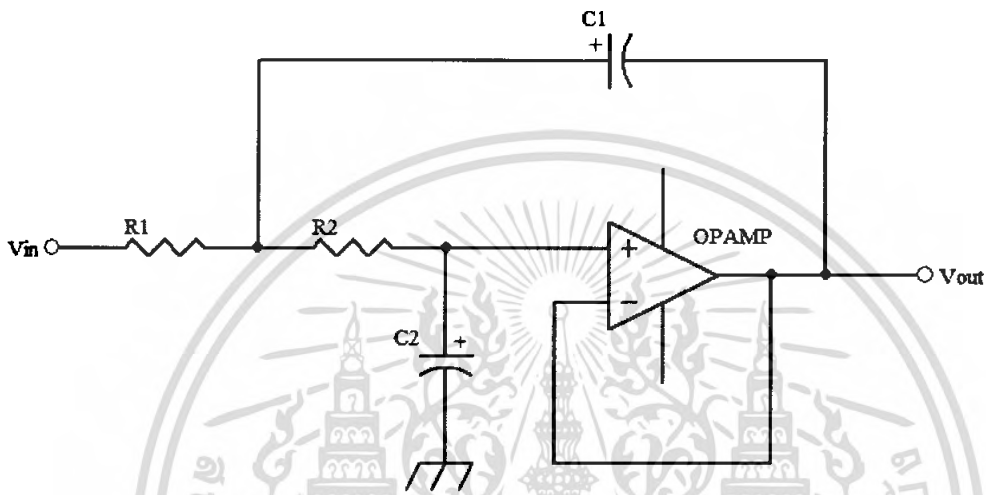
รูปที่ 3.7 วงจร Voltage follower (Buffer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 วงจร Second order Low Pass Filter

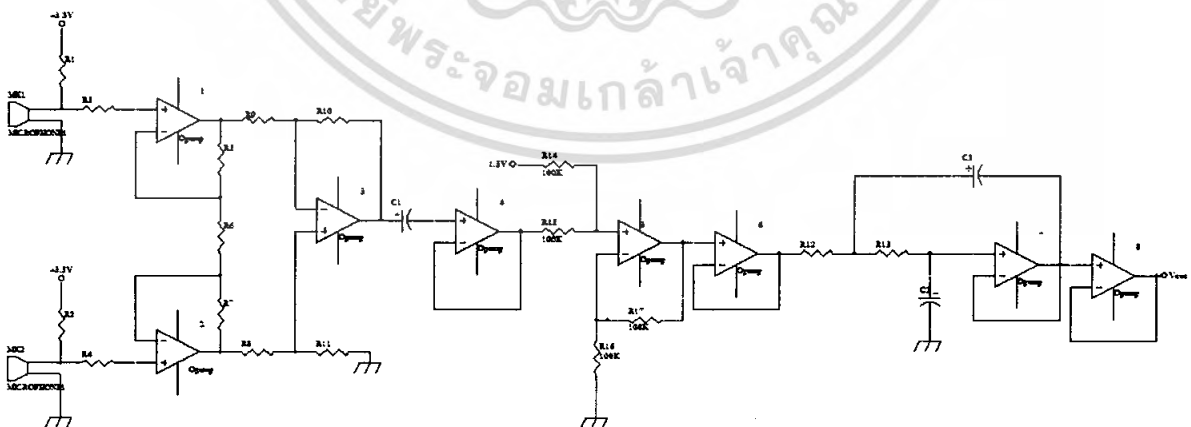
วงจร Second order Low Pass Filter จะช่วยในการกรองสัญญาณรบกวนทิ้งไป โดยความถี่ cutoff ของ Second order Low Pass Filter นั้นเราใช้ค่าประมาณ 4 KHz โดยคำนวณได้จากสมการ

$$f = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}}$$



รูปที่ 3.8 วงจร Second order Low Pass Filter

เมื่อนำวงจร Pre-microphone และ Instrument Amplifier รูปที่ 3.5, วงจร Non-Inverting Summing Amplifier รูปที่ 3.6, วงจร Voltage follower (Buffer) รูปที่ 3.7, วงจร Second order Low Pass Filter รูปที่ 3.8 มาต่อรวมกันดังรูปที่ 3.9 จึงได้เป็นวงจร Microphone Pre-Amplifier

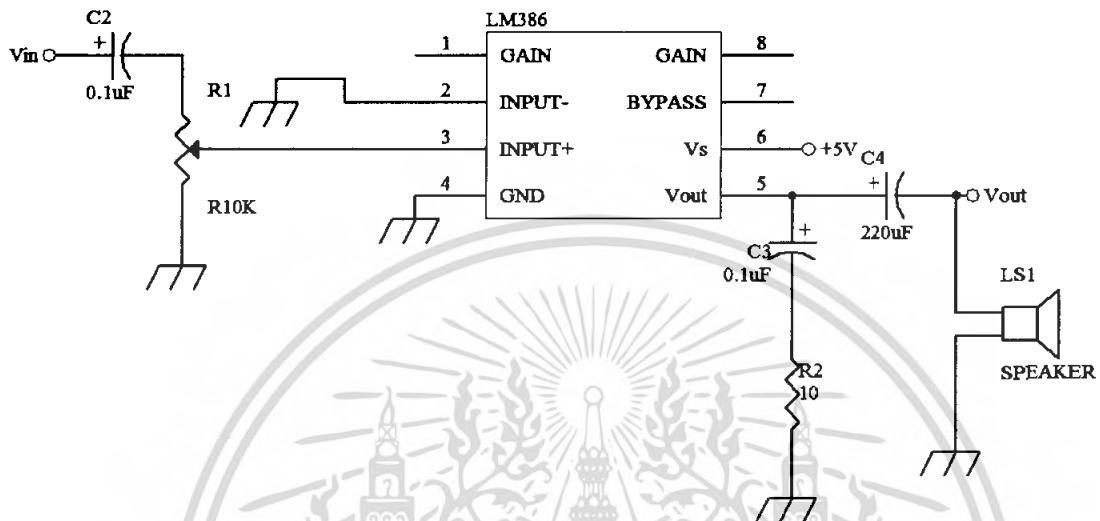


รูปที่ 3.9 วงจร Microphone Pre-Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วงจร Power Amplifier with gain=20

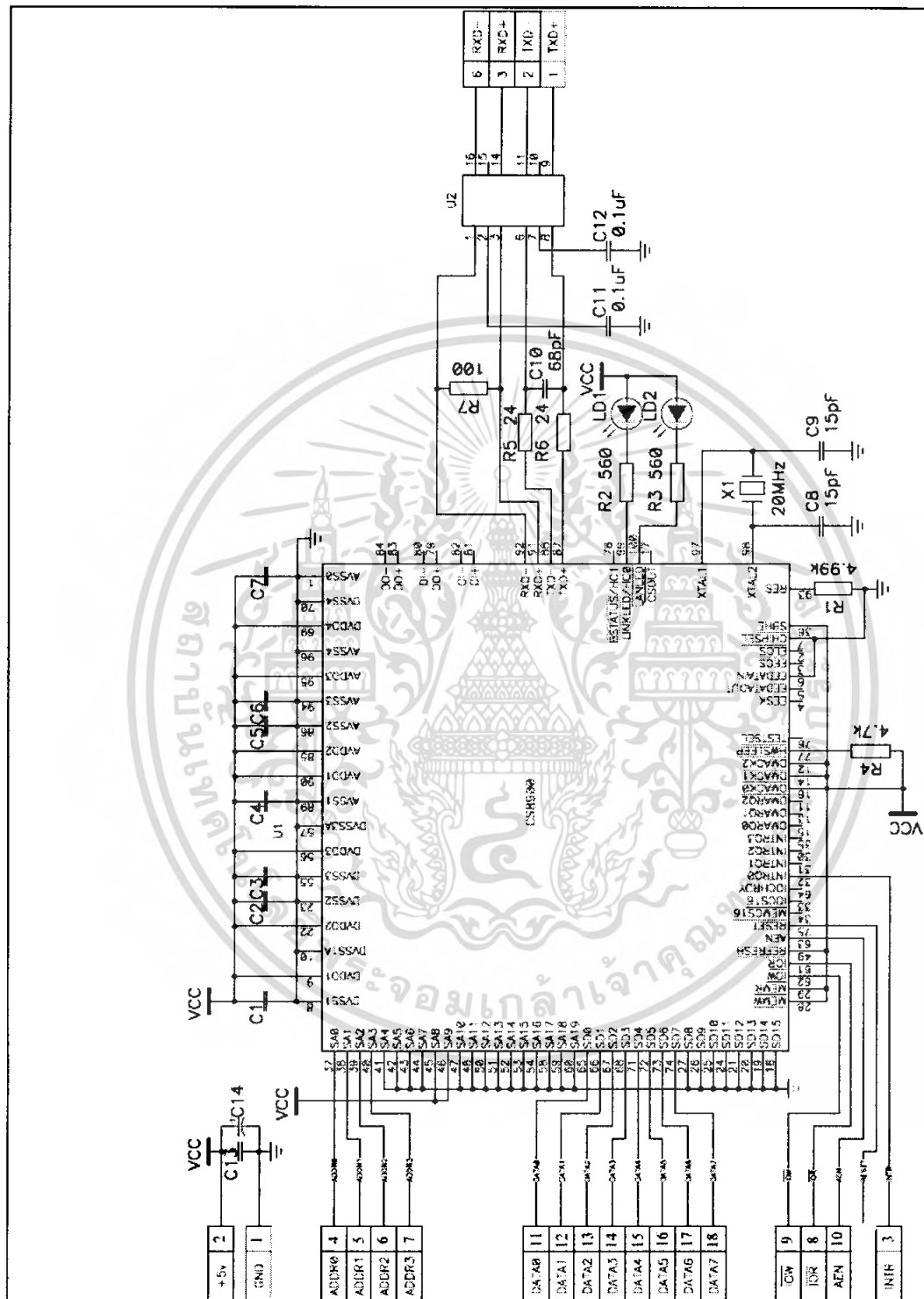
วงจร Power Amplifier with gain = 20 จะอยู่ในส่วนของภาครับ ทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าจากไมโครคอนโทรลเลอร์ ARM7 ให้เป็นสัญญาณเสียงโดยผ่านลำโพง



รูปที่ 3.10 วงจร Amplifier with gain =20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 วงจร Ethernet module

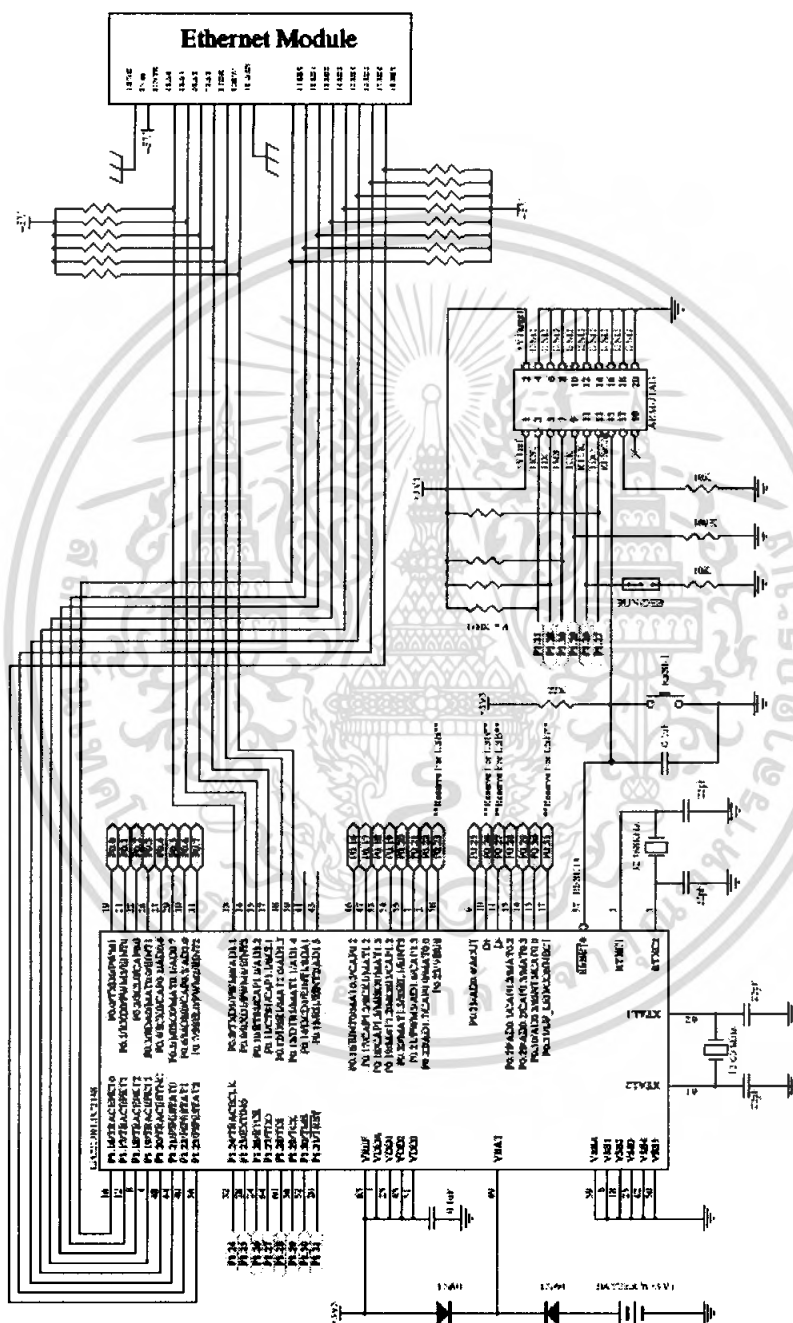


รูปที่ 3.11 วงจร Ethernet Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 วงจรไมโครคอนโทรลเลอร์ ARM7 ควบคุม Ethernet module

วงจรไมโครคอนโทรลเลอร์ ARM7 นี้จะทำหน้าที่ควบคุม Ethernet module โดยรูปวงจรมีแสดงได้ดังรูปที่ 3.12

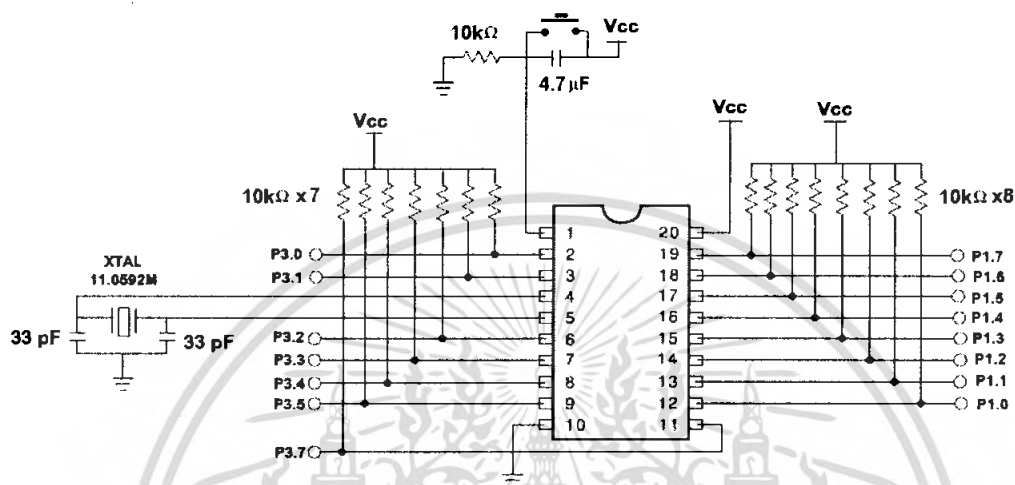


รูปที่ 3.12 วงจรไมโครคอนโทรลเลอร์ ARM7 ควบคุม Ethernet module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

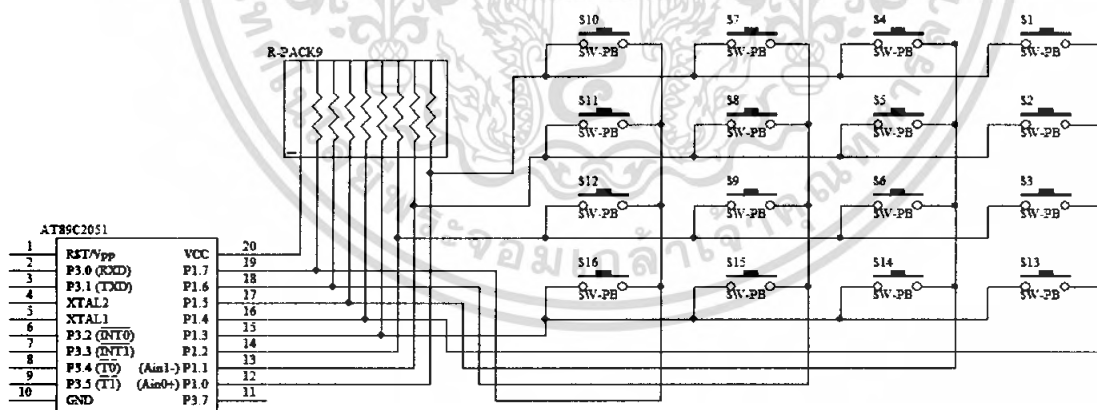
3.9 วงจรไมโครคอนโทรลเลอร์ MCS51

วงจรไมโครคอนโทรลเลอร์ MCS 51 นั้นจะทำหน้าที่รับค่าจากวงจรรคีย์แพด เพื่อส่งต่อค่านั้นไปให้กับ วงจรไมโครคอนโทรลเลอร์ ARM7 เพื่อประมวลผลต่อไป



รูปที่ 3.13 วงจรไมโครคอนโทรลเลอร์ MCS51

3.10 วงจรรคีย์แพด (Keypad)

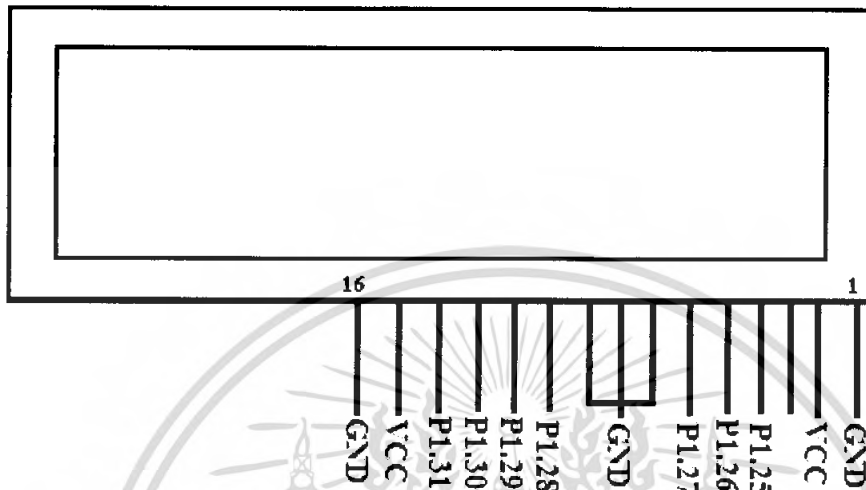


รูปที่ 3.14 วงจรรคีย์แพด (Keypad)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 วงจร LCD

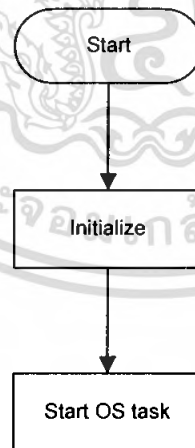
วงจร LCD ทำหน้าที่แสดงค่าโดยรับคำสั่งจากไมโครคอนโทรลเลอร์ ARM7 โดยทำการต่อขาต่างๆของ LCD กับขาพอร์ตต่างๆของ ARM7 ดังรูปที่ 3.15



รูปที่ 3.15 วงจร LCD

3.12 แผนผังแสดงการทำงานของโปรแกรม

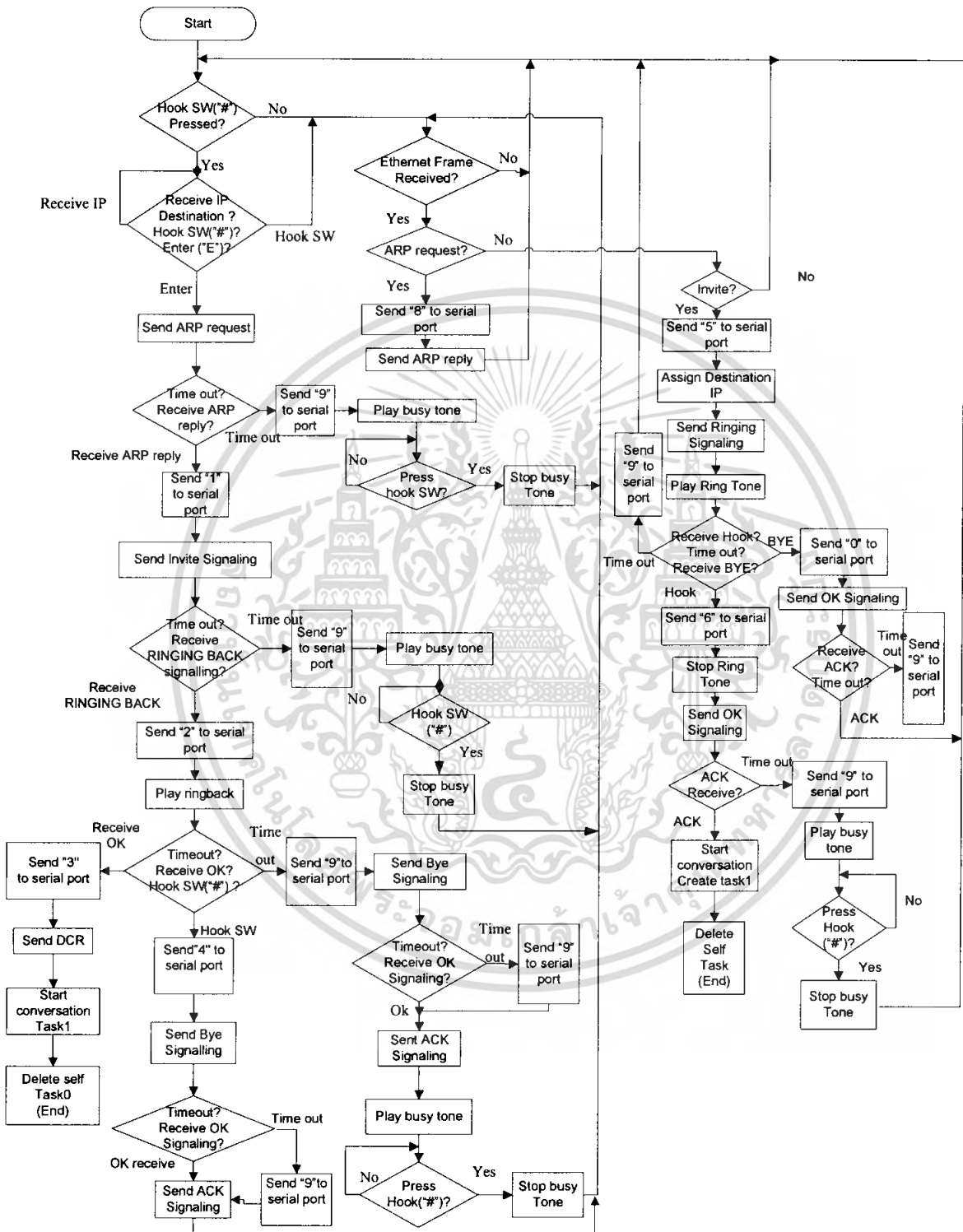
3.12.1 โปรแกรม Main



รูปที่ 3.16 แผนผังแสดงการทำงานของโปรแกรม Main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

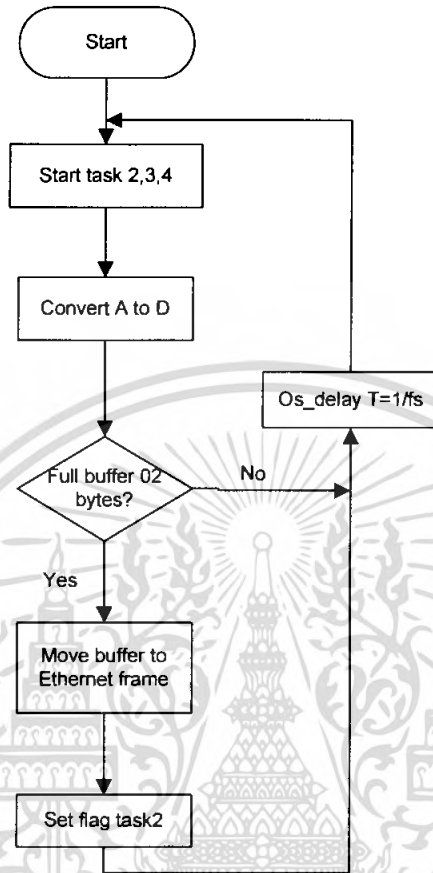
3.12.2 โปรแกรม Task0: Signaling (Priority=15)



รูปที่ 3.17 แผนผังแสดงการทำงานของโปรแกรม Task0: Signaling (Priority=15)

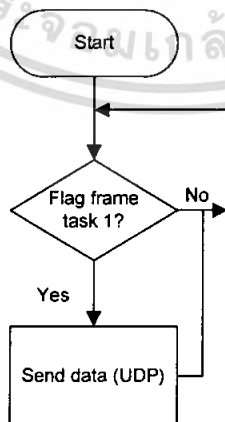
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12.3 โปรแกรม Task1: ADC (Priority=10)



รูปที่ 3.18 แผนผังแสดงการทำงานของโปรแกรม Task1: ADC (Priority=10)

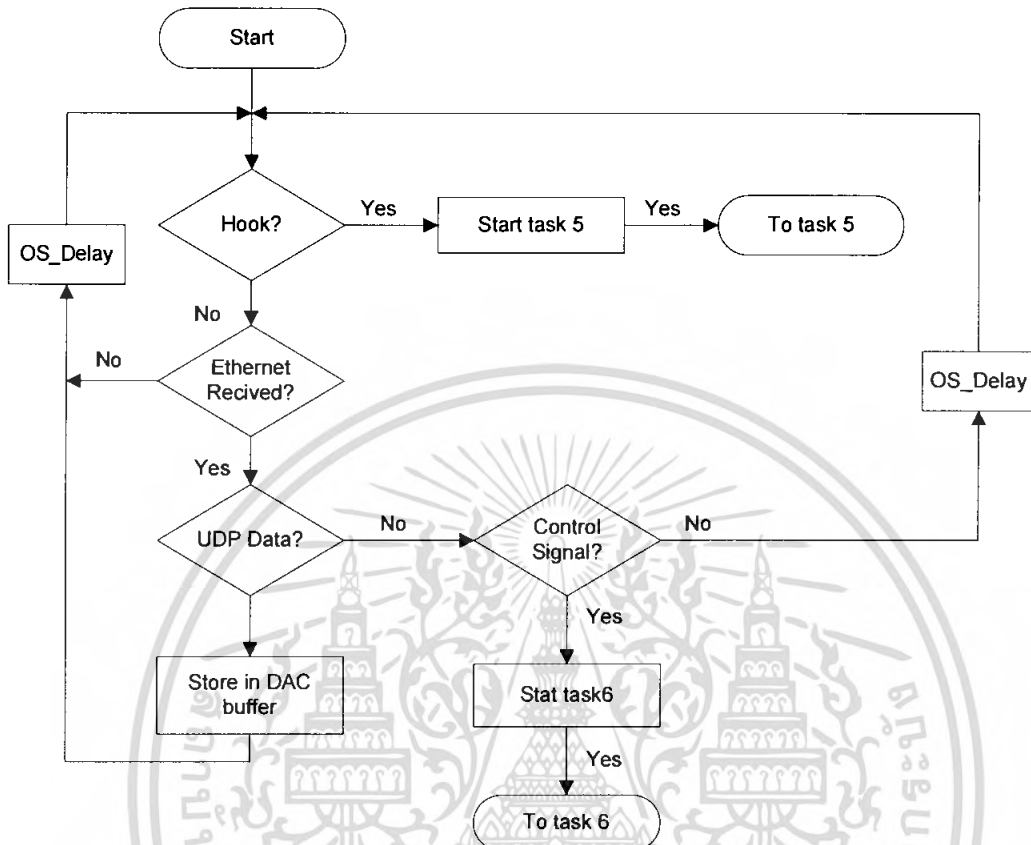
3.12.4 โปรแกรม Task2: Ethernet Send (Priority=5)



รูปที่ 3.19 แผนผังแสดงการทำงานของโปรแกรม Task2: Ethernet Send (Priority=5)

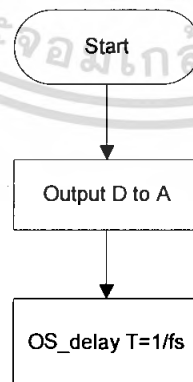
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12.5 โปรแกรม Task3: Ethernet Receive (Priority=5)



รูปที่ 3.20 แผนผังแสดงการทำงานของโปรแกรม Task3: Ethernet Receive (Priority=5)

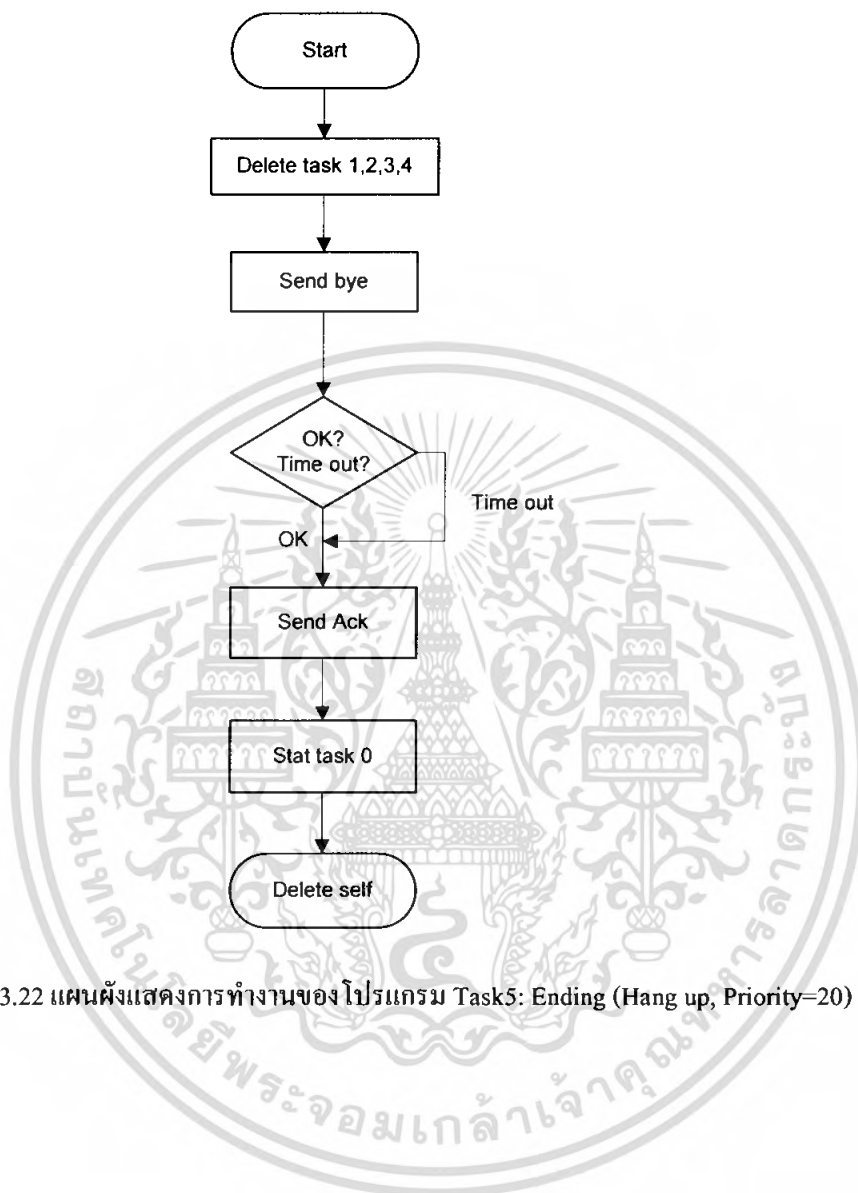
3.12.6 โปรแกรม Task4: DAC (Priority=7)



รูปที่ 3.21 แผนผังแสดงการทำงานของโปรแกรม Task4: DAC (Priority=7)

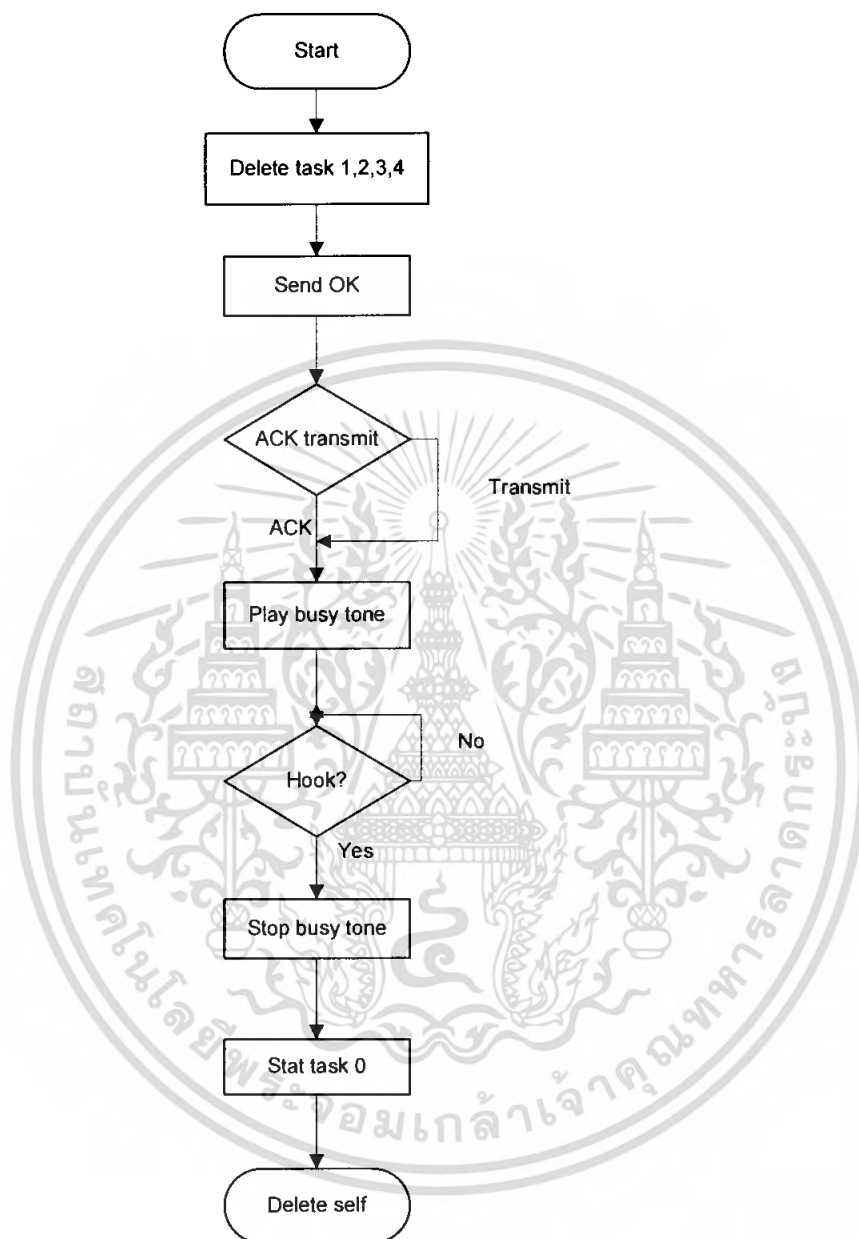
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12.7 โปรแกรม Task5: Ending (Hang up, Priority=20)



รูปที่ 3.22 แผนผังแสดงการทำงานของ โปรแกรม Task5: Ending (Hang up, Priority=20)

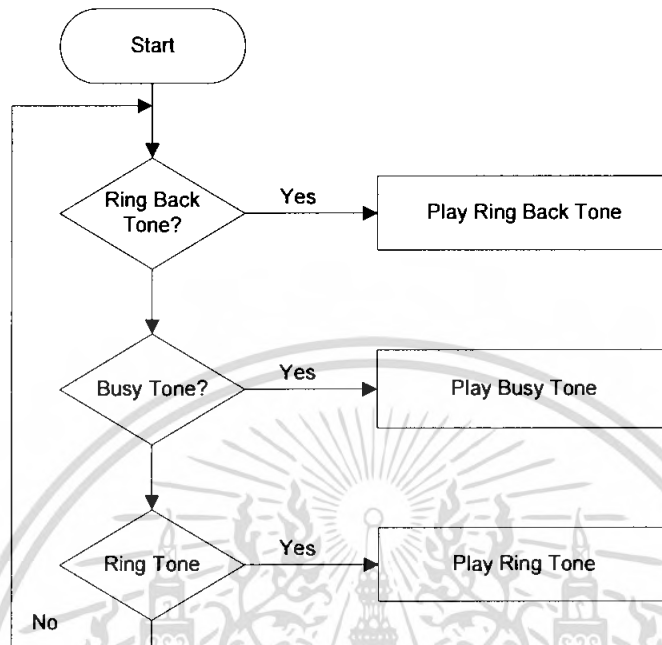
3.12.8 โปรแกรม Task6: Ending (Receive BYE, Priority=20)



รูปที่ 3.23 แผนผังแสดงการทำงานของโปรแกรม Task6: Ending (Receive BYE, Priority=20)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12.9 โปรแกรม Task7: Sound Signaling (Priority=5)



รูปที่ 3.24 แผนผังแสดงการทำงานของโปรแกรม Task7: Sound Signaling (Priority=5)

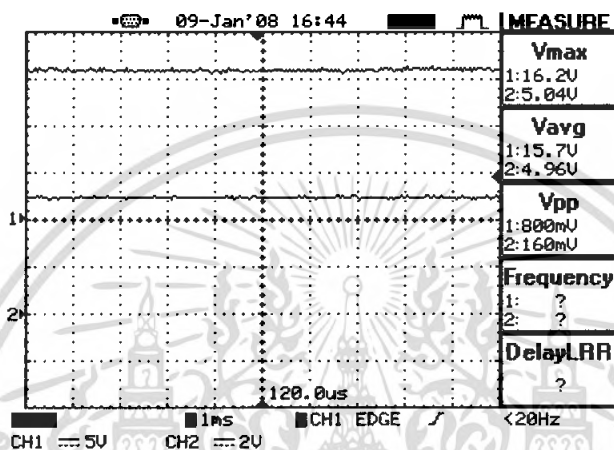
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 วงจรแปลงแรงดันไฟจาก +15V เป็น +5V

ทำการป้อนแรงดันอินพุต +15V ให้กับวงจรแปลงแรงดันดังแสดงในรูปที่ 3.2 ซึ่งจะได้แรงดันเอาต์พุต +5V ดังผลในรูปที่ 4.1



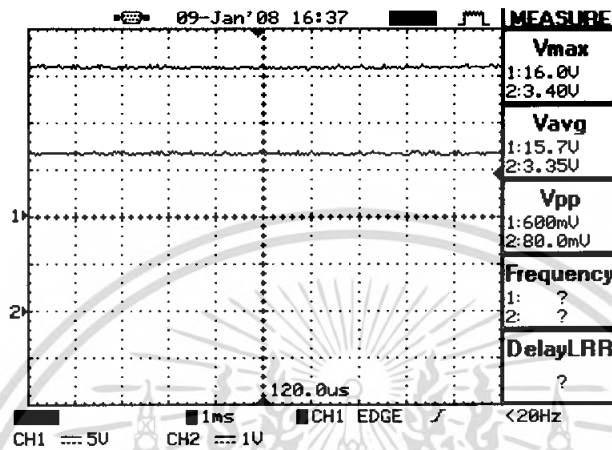
รูปที่ 4.1 ผลการทดลองวงจรแปลงแรงดันไฟจาก +15V เป็น +5V

Ch1: สัญญาณอินพุต

Ch2: สัญญาณเอาต์พุต

4.2 วงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V

ทำการป้อนแรงดันอินพุต +15V ให้กับวงจรแปลงแรงดันดังแสดงในรูปที่ 3.3 ซึ่งจะได้แรงดันเอาต์พุต +3.3V ดังผลในรูปที่ 4.2



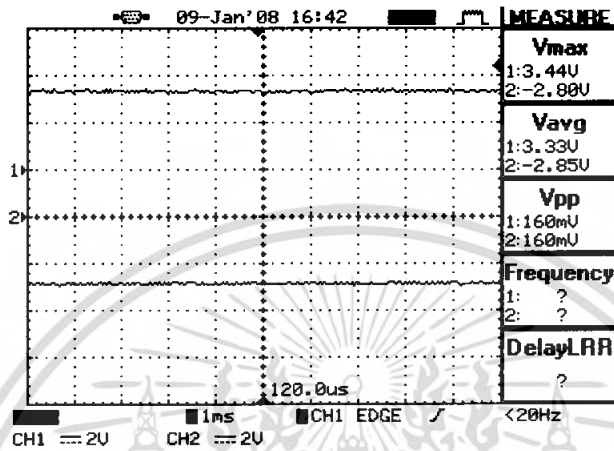
รูปที่ 4.2 ผลการทดลองวงจรแปลงแรงดันไฟจาก +15V เป็น +3.3V

Ch1: สัญญาณอินพุต

Ch2: สัญญาณเอาต์พุต

4.3 วงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V

ทำการป้อนแรงดันอินพุต +3.3V ให้กับวงจรแปลงแรงดันดังแสดงในรูปที่ 3.4 ซึ่งจะได้แรงดันเอาต์พุต -3.3V ดังผลในรูปที่ 4.3

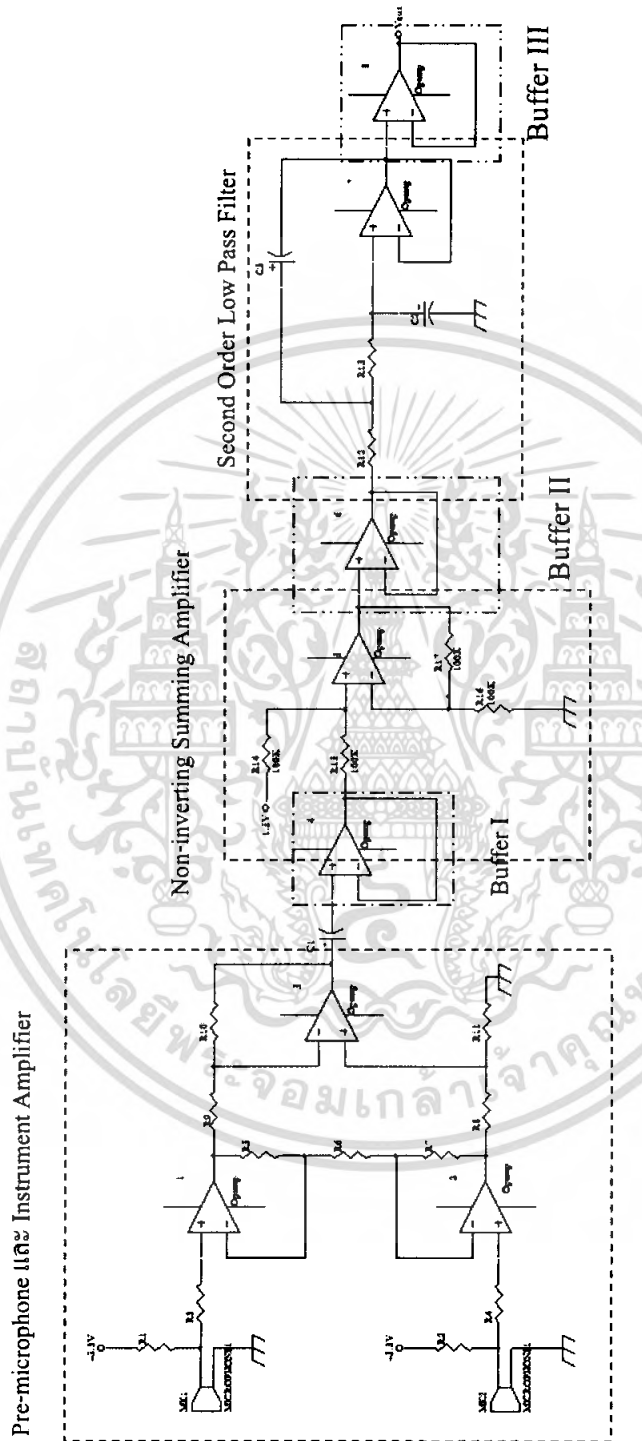


รูปที่ 4.3 ผลการทดลองวงจรแปลงแรงดันไฟจาก +3.3V เป็น -3.3V

Ch1: สัญญาณอินพุต

Ch2: สัญญาณเอาต์พุต

4.4 วงจร Microphone Pre-Amplifier

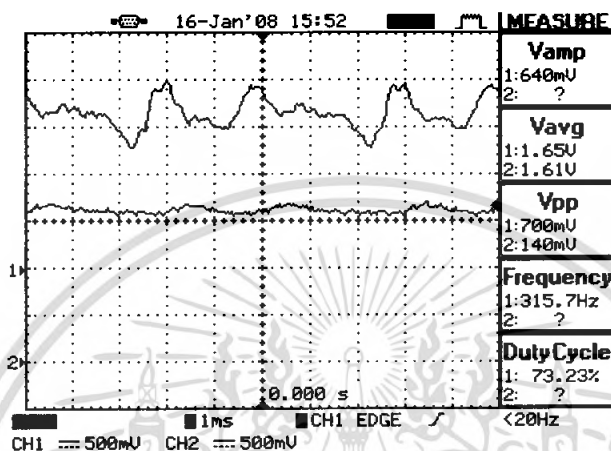


รูปที่ 4.4 วงจรประกอบของวงจร Microphone Pre-Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1 วงจร Pre-microphone และ Instrument Amplifier

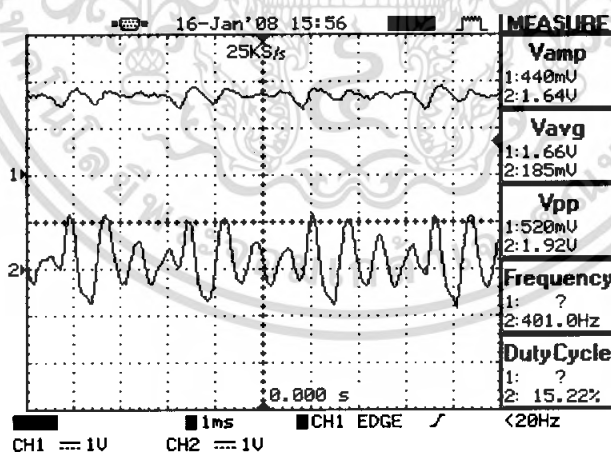
ทำการป้อนสัญญาณอินพุตให้กับ Mic1 และ Mic2 แล้วทำการวัดเอาต์พุตของวงจร Pre-microphone และ Instrument Amplifier ดังแสดงในรูปที่ 4.4 ซึ่งจะได้ผลดังรูปที่ 4.5, 4.6 และรูปที่ 4.7 โดยค่าสัญญาณเอาต์พุตของวงจรนั้นจะมีค่าดังสมการ $V_{out} = 3(V_2 - V_1)$



รูปที่ 4.5 ผลการทดลองวงจร Pre-microphone และ Instrument Amplifier โดยป้อนสัญญาณเสียง

Ch1: สัญญาณอินพุตที่ Mic2

Ch2: สัญญาณอินพุตที่ Mic1

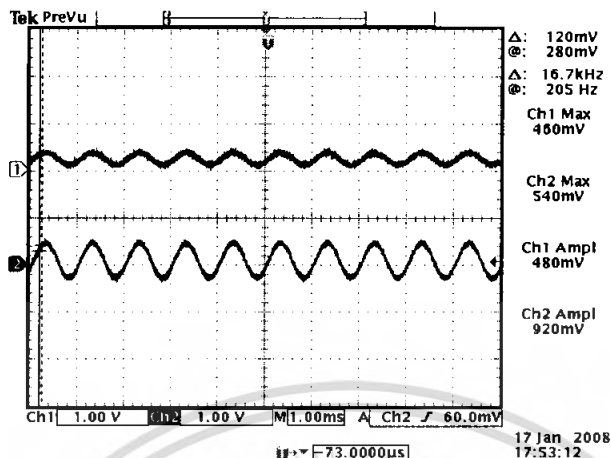


รูปที่ 4.6 ผลการทดลองวงจร Pre-microphone และ Instrument Amplifier โดยป้อนสัญญาณเสียง

Ch1: สัญญาณอินพุตที่ Mic2

Ch2: สัญญาณเอาต์พุตของวงจร Instrument Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



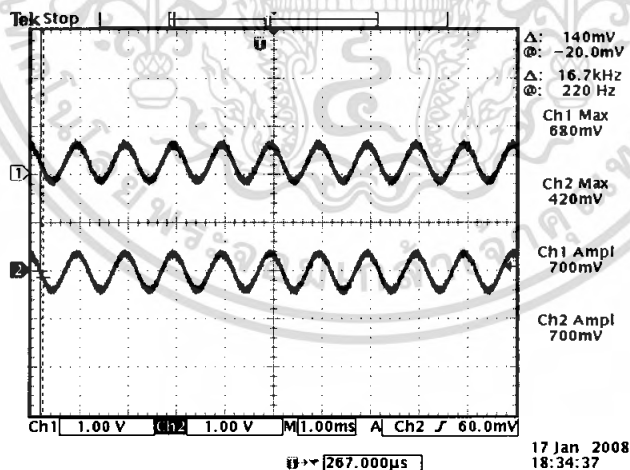
รูปที่ 4.7 ผลการทดลองวงจร Pre-microphone และ Instrument Amplifier โดยป้อนสัญญาณ sine wave

Ch1: สัญญาณอินพุตที่ Mic2

Ch2: สัญญาณเอาต์พุตของวงจร Instrument Amplifier

4.4.2 วงจร Voltage follower I (Buffer I)

ทำการวัดสัญญาณอินพุตของวงจร Buffer I (สัญญาณเอาต์พุตของวงจร Pre-microphone และ Instrument Amplifier) และสัญญาณเอาต์พุตของวงจร Buffer I ดังรูปที่ 4.4 ซึ่งจะได้ผลดังรูปที่ 4.8 และรูปที่ 4.9

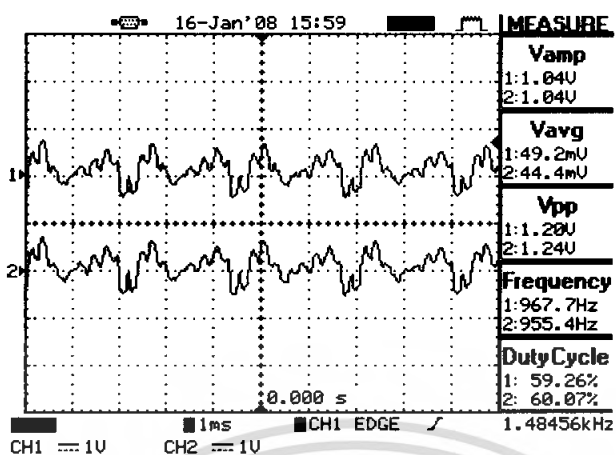


รูปที่ 4.8 ผลการทดลองวงจร Voltage follower I (Buffer I) โดยป้อนสัญญาณ sine wave

Ch1: สัญญาณอินพุตของวงจร Buffer I

Ch2: สัญญาณเอาต์พุตของวงจร Buffer I

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



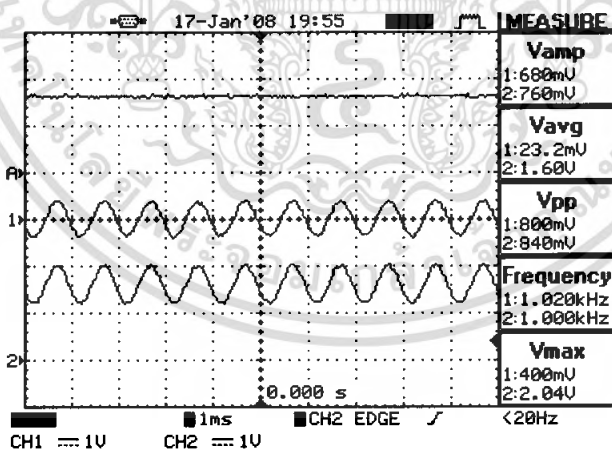
รูปที่ 4.9 ผลการทดลองวงจร Voltage follower I (Buffer I) โดยป้อนสัญญาณเสียง

Ch1: สัญญาณอินพุตของวงจร Buffer I

Ch2: สัญญาณเอาต์พุตของวงจร Buffer I

4.4.3 วงจร Non-inverting Summing Amplifier

ทำการวัดสัญญาณอินพุตและสัญญาณเอาต์พุตของวงจร Non-inverting Summing Amplifier ดังรูปที่ 4.4 โดยสัญญาณอินพุตนั้นประกอบด้วยสัญญาณแรงดันไฟกระแสตรง (DC) และสัญญาณเอาต์พุตของวงจร Buffer I ซึ่งทำให้ได้ผลดังแสดงในรูปที่ 4.10 และ รูปที่ 4.11



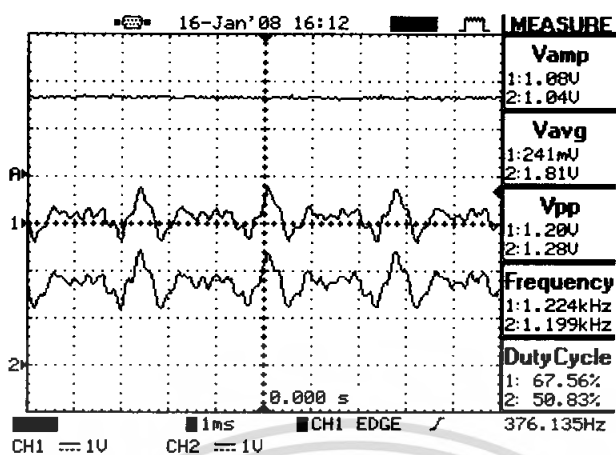
รูปที่ 4.10 ผลการทดลองของวงจร Non-inverting Summing Amplifier โดยป้อนสัญญาณ sine wave

Ref A: อินพุต 1 (แรงดันไฟกระแสตรง (DC))

Ch1: สัญญาณเอาต์พุตของวงจร Buffer I

Ch2: สัญญาณเอาต์พุตของวงจร Non-inverting Summing Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ผลการทดลองของวงจร Non-inverting Summing Amplifier

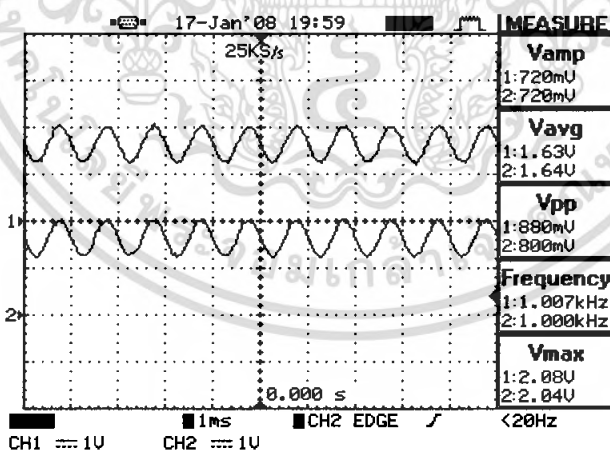
Ref A: อินพุต 1 (แรงดันไฟกระแสตรง (DC))

Ch1: สัญญาณเอาต์พุตของวงจร Buffer I

Ch2: สัญญาณเอาต์พุตของวงจร Non-inverting Summing Amplifier

4.4.4 วงจร Voltage follower II (Buffer II)

ทำการวัดสัญญาณอินพุต (สัญญาณเอาต์พุตของวงจร Non-inverting Summing Amplifier) และสัญญาณเอาต์พุตของวงจร Buffer II ดังรูปที่ 4.4 ซึ่งจะได้ผลดังรูปที่ 4.12 และรูปที่ 4.13

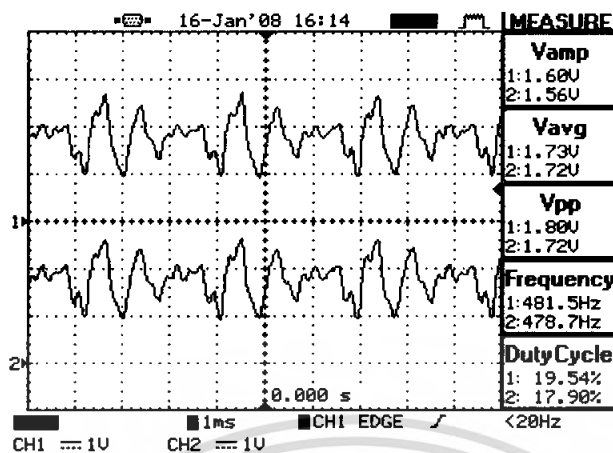


รูปที่ 4.12 ผลการทดลองของวงจร Voltage follower II (Buffer II) โดยป้อนสัญญาณ sine wave

Ch1 : สัญญาณอินพุตของวงจร Buffer II

Ch2 : สัญญาณเอาต์พุตของวงจร Buffer II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ผลการทดลองของวงจร Voltage follower II (Buffer II) โดยป้อนสัญญาณเสียง

Ch1 : สัญญาณอินพุตของวงจร Buffer II

Ch2 : สัญญาณเอาต์พุตของวงจร Buffer II

4.4.5 วงจร Second order Low Pass Filter

ทำการวัดสัญญาณอินพุตของวงจร Second order Low Pass Filter (สัญญาณเอาต์พุตของวงจร Buffer II) และสัญญาณเอาต์พุตของวงจร Second order Low Pass Filter ดังรูปที่ 4.4 โดยทำการป้อนสัญญาณ sine wave ขนาด 1V แล้วทำการเปลี่ยนค่าความถี่ของสัญญาณ sine wave ตั้งแต่ความถี่ 0.25 kHz – 24 kHz ซึ่งได้ผลการทดลองดังตารางที่ 4.1

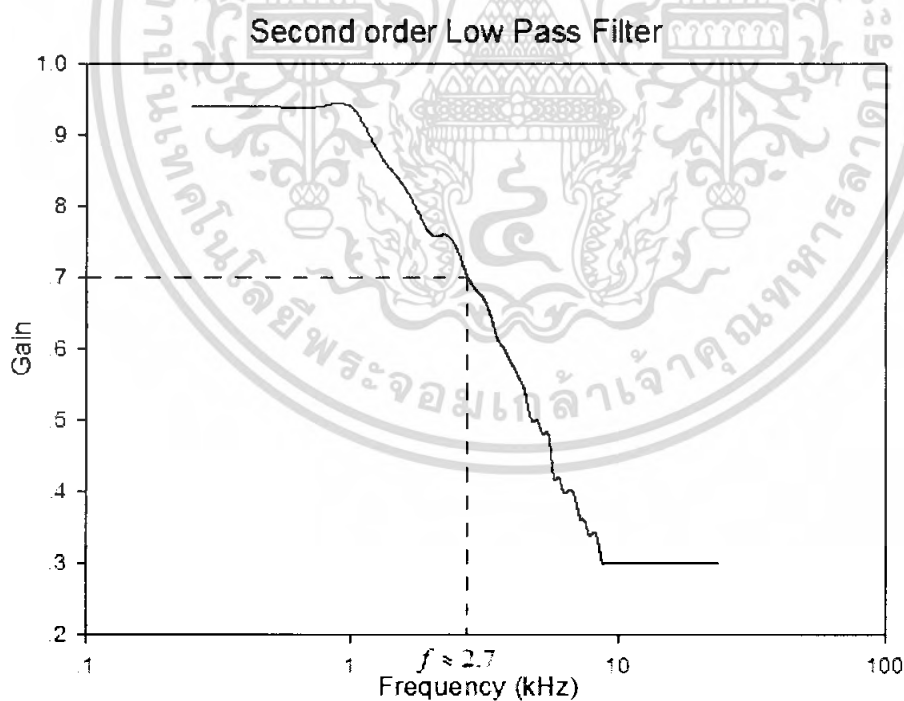
ตารางที่ 4.1 ผลการทดลองการวัดขนาดของสัญญาณ sine wave ที่ความถี่ต่างๆ

ความถี่ (kHz)	ขนาด (mV)	ความถี่ (kHz)	ขนาด (mV)	ความถี่ (kHz)	ขนาด (mV)
0.25	940	4.25	560	8.25	340
0.5	940	4.5	540	8.5	320
0.75	940	4.75	500	8.75	320
1	940	5	500	9	300
1.25	880	5.25	480	9.25	300
1.5	840	5.5	480	9.5	300
1.75	800	5.75	420	9.75	300
2	760	6	420	10	300
2.25	760	6.25	400	10.25	300
2.5	740	6.5	400	10.5	300
2.75	700	6.75	400	10.75	300
3	680	7	380	11	300
3.25	660	7.25	360	11.25	300
3.5	620	7.5	360	11.5	300
3.75	600	7.75	340	11.75	300
4	580	8	340	12	300

ตารางที่ 4.1(ต่อ) ผลการทดลองการวัดขนาดของสัญญาณ sine wave ที่ความถี่ต่าง

ความถี่ (kHz)	ขนาด (mV)	ความถี่ (kHz)	ขนาด (mV)	ความถี่ (kHz)	ขนาด (mV)
12.25	300	16.25	300	20.25	300
12.5	300	16.5	300	20.5	300
12.75	300	16.75	300	20.75	300
13	300	17	300	21	300
13.25	300	17.25	300	21.25	300
13.5	300	17.5	300	21.5	300
13.75	300	17.75	300	21.75	300
14	300	18	300	22	300
14.25	300	18.25	300	22.25	300
14.5	300	18.5	300	22.5	300
14.75	300	18.75	300	22.75	300
15	300	19	300	23	300
15.25	300	19.25	300	23.25	300
15.5	300	19.5	300	23.5	300
15.75	300	19.75	300	23.75	300
16	300	20	300	24	300

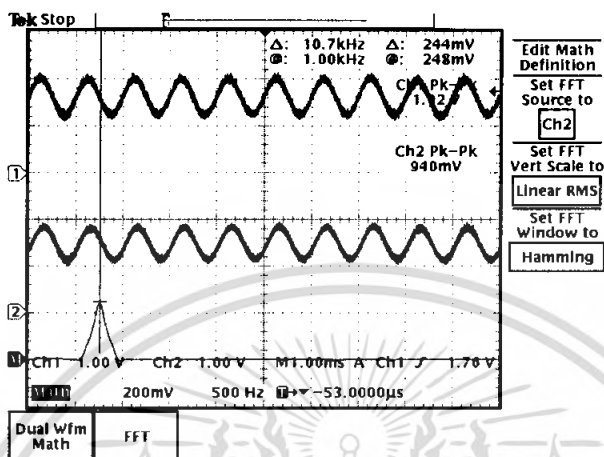
จากตารางที่ 4.1 เมื่อนำมาเขียนเป็นกราฟลอการิทึมจะได้ผลแสดงดังรูปที่ 4.14



รูปที่ 4.14 กราฟลอการิทึมของวงจร Second order Low Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างผลการทดลองของการป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz และที่ความถี่ 4 kHz จากตารางที่ 4.1 แสดงดังรูปที่ 4.15 และรูปที่ 4.16



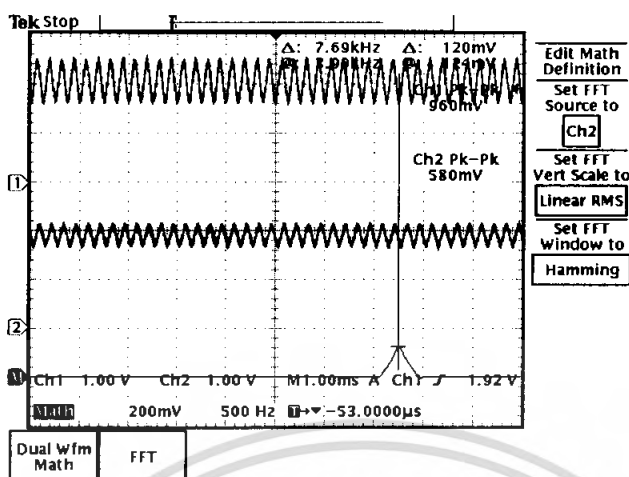
รูปที่ 4.15 ผลการทดลองของวงจร Second order Low Pass Filter โดยป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz

Ch1 : สัญญาณอินพุตของวงจร Second order Low Pass Filter

(สัญญาณเอาต์พุตของวงจร Buffer II)

Ch2 : สัญญาณเอาต์พุตของวงจร Second order Low Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

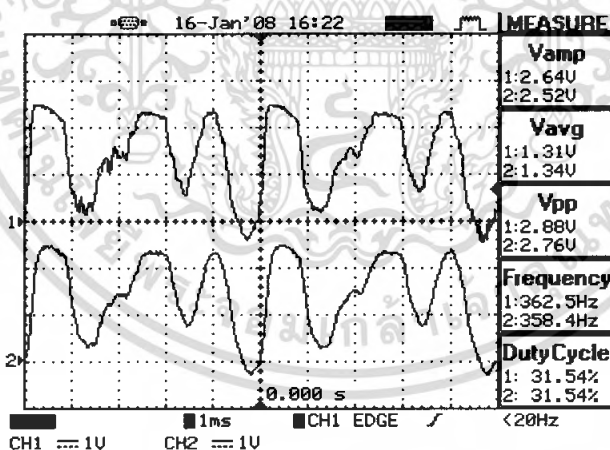


รูปที่ 4.16 ผลการทดลองของวงจร Second order Low Pass Filter โดยป้อนสัญญาณ sine wave ที่ความถี่ 4 kHz

Ch1 : สัญญาณอินพุตของวงจร Second order Low Pass Filter
(สัญญาณเอาต์พุตของวงจร Buffer II)

Ch2 : สัญญาณเอาต์พุตของวงจร Second order Low Pass Filter

ตัวอย่างผลการทดลองของวงจร Second order Low Pass Filter เมื่อป้อนสัญญาณเสียงแสดงดังรูปที่ 4.17



รูปที่ 4.17 ผลการทดลองของวงจร Second order Low Pass Filter โดยป้อนสัญญาณเสียง

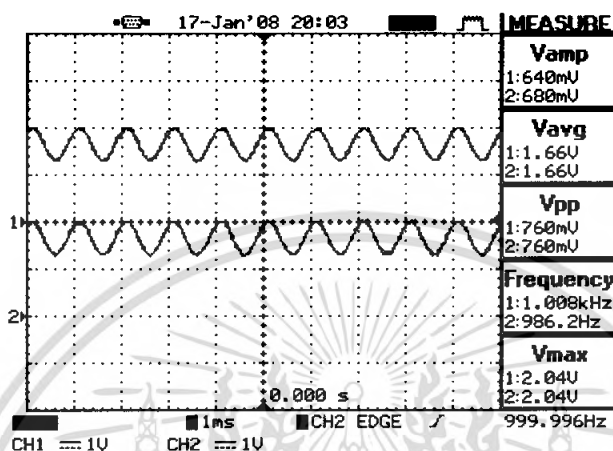
Ch1 : สัญญาณอินพุตของวงจร Second order Low Pass Filter
(สัญญาณเอาต์พุตของวงจร Buffer II)

Ch2 : สัญญาณเอาต์พุตของวงจร Second order Low Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.6 วงจร Voltage follower III (Buffer III)

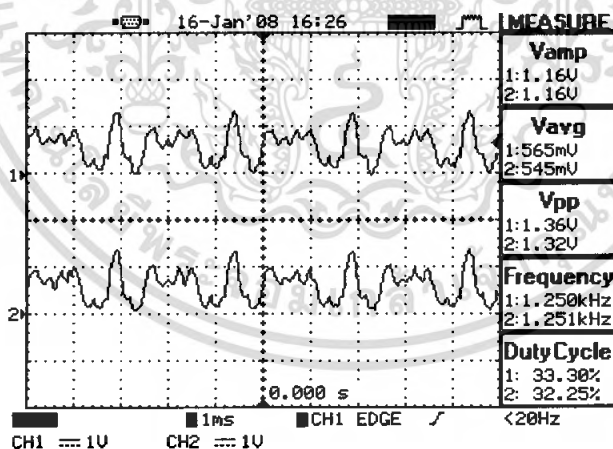
ทำการวัดสัญญาณอินพุต (สัญญาณเอาต์พุตของวงจร Second order Low Pass Filter) และสัญญาณเอาต์พุตของวงจร Buffer III ดังรูปที่ 4.4 ซึ่งจะได้ผลดังรูปที่ 4.18 และรูปที่ 4.19



รูปที่ 4.18 ผลการทดลองของวงจร Voltage follower III (Buffer III) โดยป้อนสัญญาณ sine wave

Ch1 : สัญญาณอินพุตของวงจร Buffer III

Ch2 : สัญญาณเอาต์พุตของวงจร Buffer III



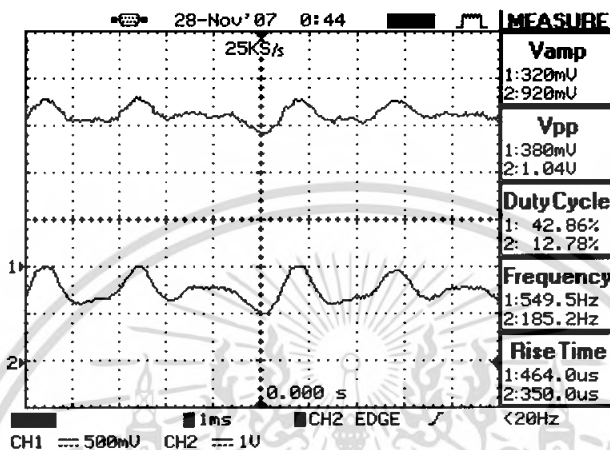
รูปที่ 4.19 ผลการทดลองของวงจร Voltage follower III (Buffer III) โดยป้อนสัญญาณเสียง

Ch1 : สัญญาณอินพุตของวงจร Buffer III

Ch2 : สัญญาณเอาต์พุตของวงจร Buffer III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปวงจรประกอบของวงจร Microphone Pre-Amplifier ดังแสดงในรูปที่ 4.4 เมื่อเราทำการวัดสัญญาณอินพุตแรกสุดจาก Mic2 ของวงจร Pre-microphone และ Instrument Amplifier แล้วทำการวัดเอาต์พุตสุดท้ายที่วงจร Buffer III จะทำให้ได้อินพุตและเอาต์พุตของวงจร Microphone Pre-Amplifier ดังแสดงในรูปที่ 4.20



รูปที่ 4.20 ผลการทดลองของวงจร Microphone Pre-Amplifier เมื่อป้อนสัญญาณเสียง

Ch1: สัญญาณอินพุต Mic2 ของวงจร Pre-microphone

และ Instrument Amplifier

Ch2: สัญญาณเอาต์พุตของวงจร Buffer III

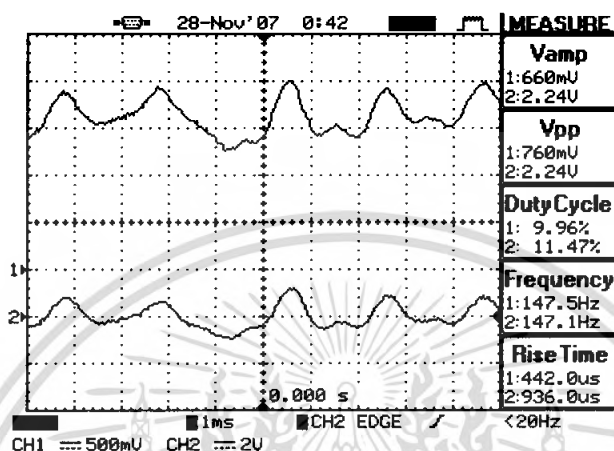
จากการทดลองวงจร Microphone Pre-Amplifier นั้นสามารถรับค่าแรงดันอินพุตต่ำสุดได้ที่ค่าแรงดันประมาณ 8 mV ดังนั้นจึงคำนวณค่า Sensitivity ได้ดังนี้

$$\begin{aligned} \text{Sensitivity} &= 10 \log_{10} 8 \times 10^3 \\ &= -20 \text{ dB} \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 วงจร Power Amplifier with gain=20

ทำการป้อนสัญญาณอินพุตให้กับวงจร Power Amplifier with gain=20 ดังแสดงในรูปที่ 3.10 แล้วทำการวัดสัญญาณเอาต์พุต ซึ่งจะได้ผลดังรูปที่ 4.21



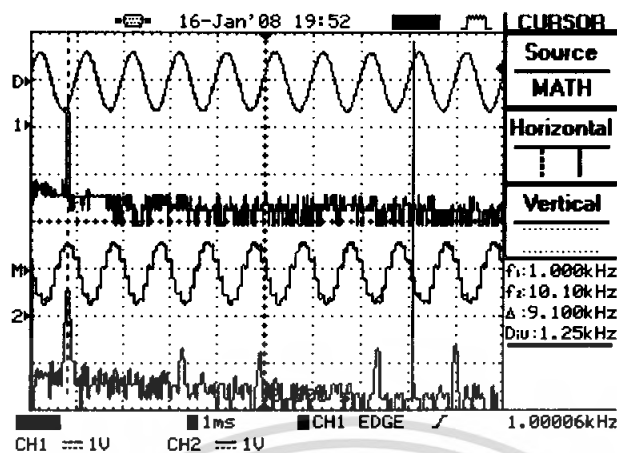
รูปที่ 4.21 ผลการทดลองของวงจร Power Amplifier with gain=20 เมื่อป้อนสัญญาณเสียง

Ch1 : สัญญาณอินพุตของวงจร Power Amplifier with gain=20

Ch2: สัญญาณเอาต์พุตของวงจร Power Amplifier with gain=20

4.6 การส่งสัญญาณเสียงผ่านเครือข่าย Ethernet

ในการส่งสัญญาณผ่านเครือข่าย Ethernet นั้น จะต้องมีการแปลงสัญญาณเสียงให้เป็นสัญญาณไฟฟ้าและทำการขยายสัญญาณเสียงนั้น โดยผ่านวงจร Microphone Pre-Amplifier เสียก่อน จากนั้นจึงเริ่มกระบวนการส่งสัญญาณผ่านเครือข่าย Ethernet โดยเริ่มจากการส่งสัญญาณไฟฟ้านี้เข้าสู่ไมโครคอนโทรลเลอร์ ARM7 เพื่อทำการประมวลผลและแปลงสัญญาณไฟฟ้าที่อยู่ในรูปสัญญาณอนาลอกนั้น ให้เป็นสัญญาณไฟฟ้าในรูปแบบดิจิทัลแล้ว จึงทำการส่งผ่านสัญญาณนั้นไปยัง Ethernet Module เพื่อเข้าสู่เครือข่าย Ethernet ไปยังปลายทาง ซึ่ง Ethernet module ทางปลายทางหรือทางด้านรับนั้น จะทำการส่งผ่านสัญญาณไปยังไมโครคอนโทรลเลอร์ ARM7 เพื่อประมวลผลและทำการแปลงสัญญาณไฟฟ้าในรูปแบบดิจิทัลนั้นกลับคืนสู่รูปแบบอนาลอก แล้วทำการส่งต่อไปยังวงจร Power Amplifier with gain=20 เพื่อทำการขยายสัญญาณและแปลงสัญญาณไฟฟ้าให้อยู่ในรูปสัญญาณเสียงดั้งเดิม โดยผลการทดลองของการส่งสัญญาณเสียงผ่านเครือข่าย Ethernet ซึ่งมีการกำหนดความถี่ของการซำคค่าสัญญาณหรือความถี่ Sampling ที่ 10 kHz นั้นแสดงดังรูปที่ 4.22 – 4.29



รูปที่ 4.22 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz

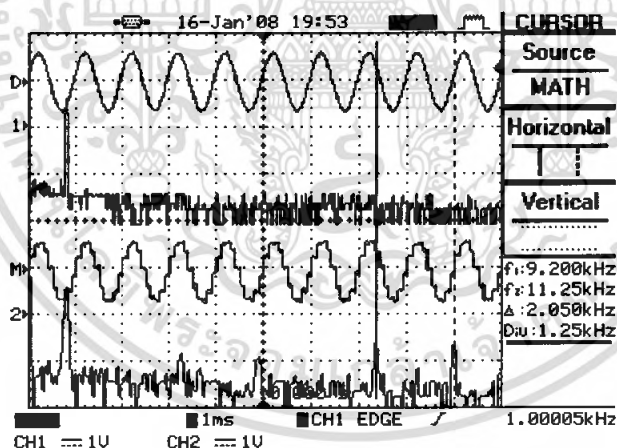
Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ref A: สเปกตรัมของสัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ref M: สเปกตรัมของสัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(เคอร์เซอร์แสดงความถี่ sampling ที่แกนความถี่ 10 kHz)



รูปที่ 4.23 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 1 kHz

Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

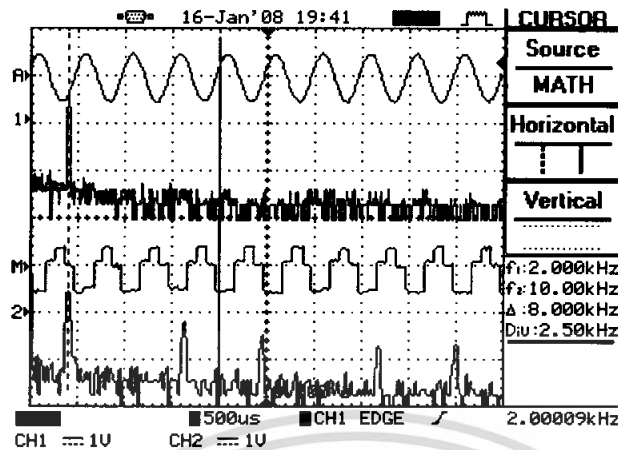
Ref D: สเปกตรัมของสัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ref M: สเปกตรัมของสัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(เคอร์เซอร์แสดงแท่งสเปกตรัมของสัญญาณที่ความถี่ 9 kHz และ 11 kHz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 2 kHz

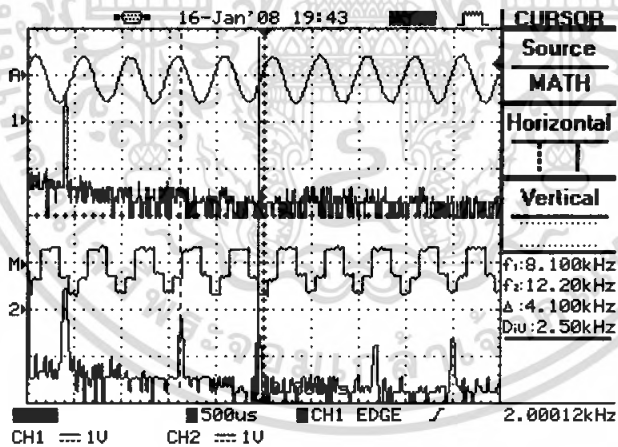
Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ref A: สเปกตรัมของสัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ref M: สเปกตรัมของสัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(เคอร์เซอร์แสดงความถี่ sampling ที่แกนความถี่ 10 kHz)



รูปที่ 4.25 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 2 kHz

Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

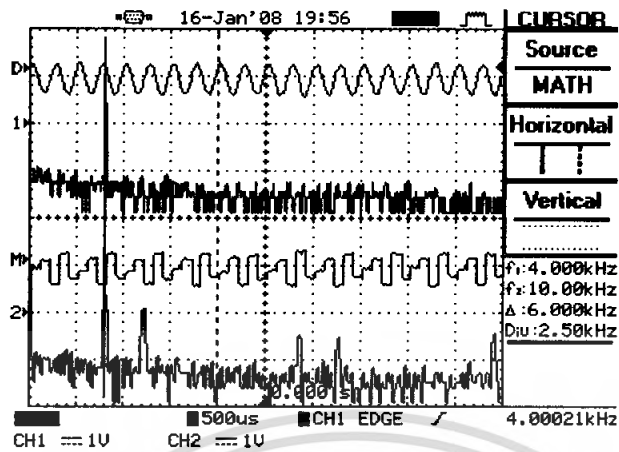
Ref A: สเปกตรัมของสัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ref M: สเปกตรัมของสัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(เคอร์เซอร์แสดงแท่งสเปกตรัมของสัญญาณที่ความถี่ 8 kHz และ 12 kHz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 4 kHz

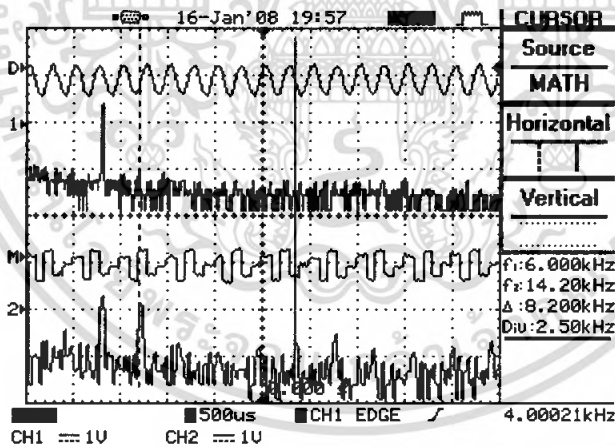
Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ref D: สเปกตรัมของสัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ref M: สเปกตรัมของสัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(เคอร์เซอร์แสดงความถี่ sampling ที่แกนความถี่ 10 kHz)



รูปที่ 4.27 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณ sine wave ที่ความถี่ 4 kHz

Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

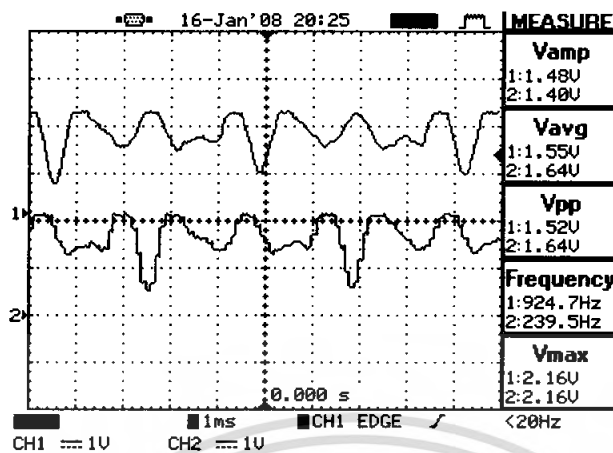
Ref D: สเปกตรัมของสัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ref M: สเปกตรัมของสัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(เคอร์เซอร์แสดงแท่งสเปกตรัมของสัญญาณที่ความถี่ 6 kHz และ 14 kHz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

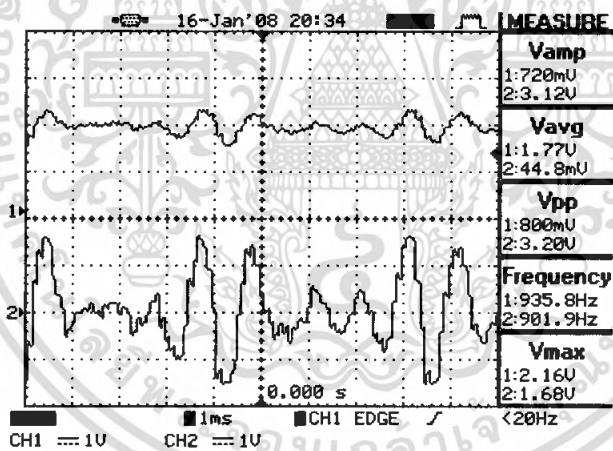


รูปที่ 4.28 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณเสียง

Ch1: สัญญาณอินพุตก่อนส่งเข้าเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

(สังเกตได้ว่าสัญญาณเอาต์พุตนั้น จะเกิดการดีเลย์)



รูปที่ 4.29 ผลการทดลองของการส่งสัญญาณผ่านเครือข่าย Ethernet โดยป้อนสัญญาณเสียง

Ch1: สัญญาณเอาต์พุตหลังจากส่งผ่านเครือข่าย Ethernet

Ch2: สัญญาณเอาต์พุตหลังจากผ่านวงจร Power Amplifier with gain=20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 รูปแบบของสัญญาณเสียงแสดงสถานะ

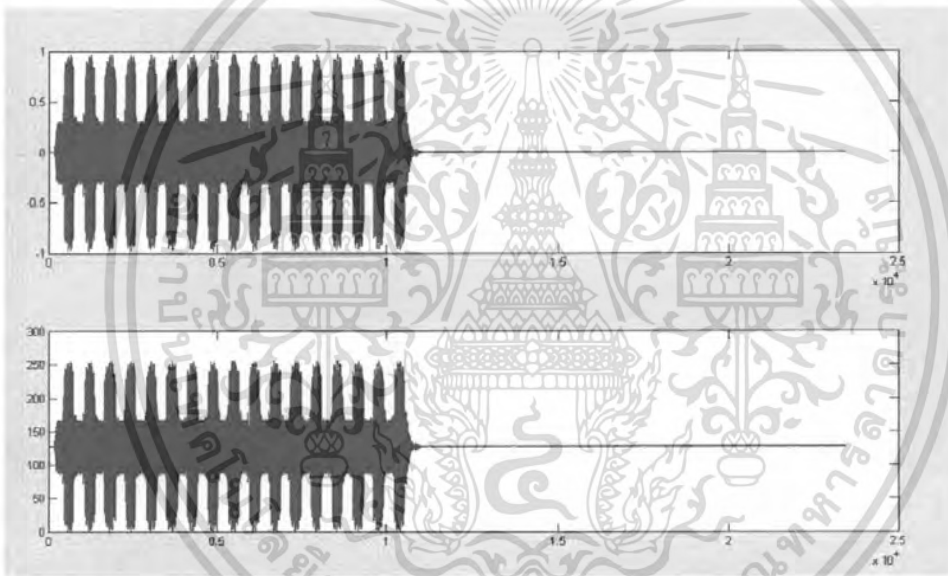
สัญญาณเสียงที่ใช้แสดงถึงสถานะนั้นมีจำนวน 3 รูปแบบคือ

- Ring Tone
- Ring Back Tone
- Busy Tone

4.7.1 สัญญาณเสียง Ring Tone

ใช้แสดงสถานะรอสายกรณีมีผู้ติดต่อหรือโทรเข้า โดยแสดงที่ฝั่งรับ ซึ่งแสดงผลดังรูปที่ 4.30 และรูปที่

4.31

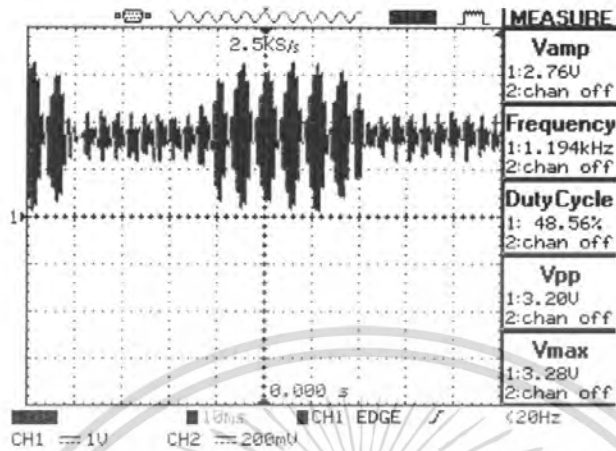


รูปที่ 4.30 รูปแบบสัญญาณเสียง Ring Tone ที่สร้างจากโปรแกรม MATLAB

ด้านบน : แสดงสัญญาณเสียงที่สร้าง

ด้านล่าง : แสดงสัญญาณเสียงที่ยกระดับ

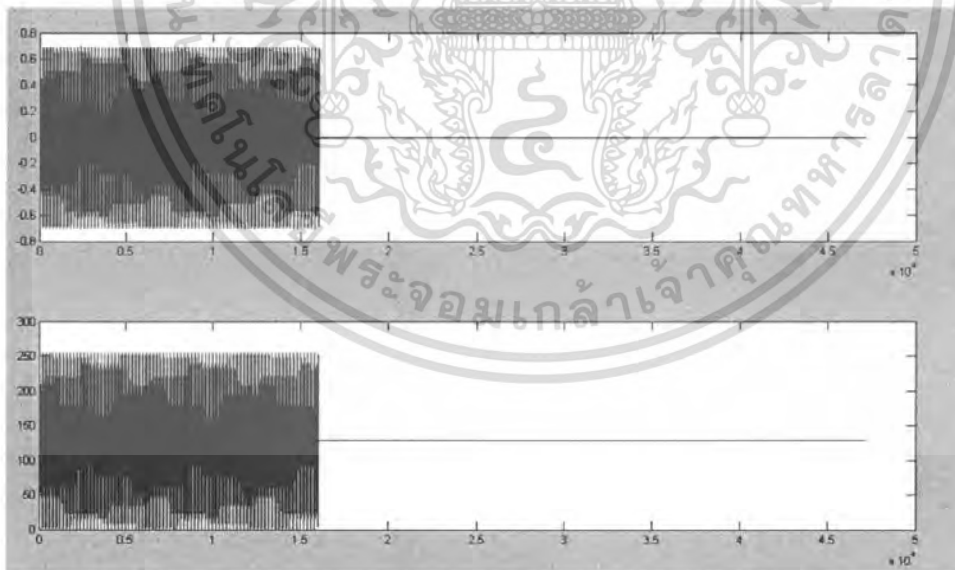
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 รูปแบบสัญญาณเสียง Ring Tone ที่วัดได้

4.7.2 สัญญาณเสียง Ring Back Tone

ใช้แสดงสถานะรอสายกรณีทำการติดต่อหรือโทรออก โดยแสดงที่ฝั่งส่ง ซึ่งแสดงผลดังรูปที่ 4.32 และรูปที่ 4.33

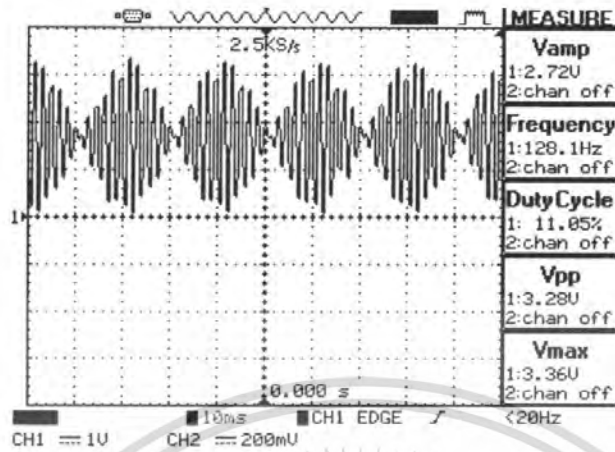


รูปที่ 4.32 รูปแบบสัญญาณเสียง Ring Back Tone ที่สร้างจากโปรแกรม MATLAB

ด้านบน : แสดงสัญญาณเสียงที่สร้าง

ด้านล่าง : แสดงสัญญาณเสียงที่ยกระดับ

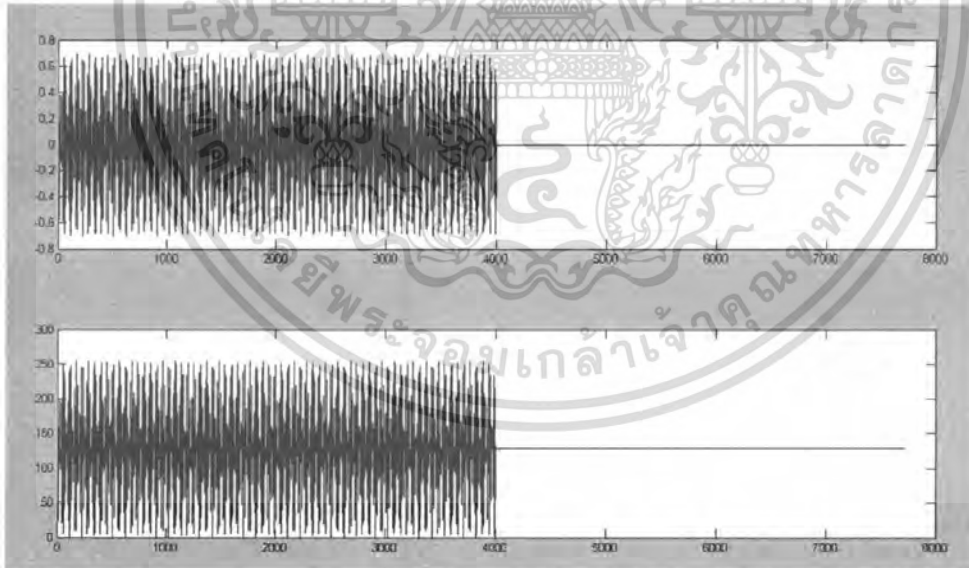
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 รูปแบบสัญญาณเสียง Ring Back Tone ที่วัดได้

4.7.3 สัญญาณเสียง Busy Tone

ใช้แสดงสถานะกรณียกเลิกการติดต่อหรือทำการติดต่อไม่สำเร็จ ซึ่งอาจมีเหตุมาจากสายไม่ว่างหรือเกิดความขัดข้องของระบบ ซึ่งแสดงผลดังรูปที่ 4.34 และรูปที่ 4.35

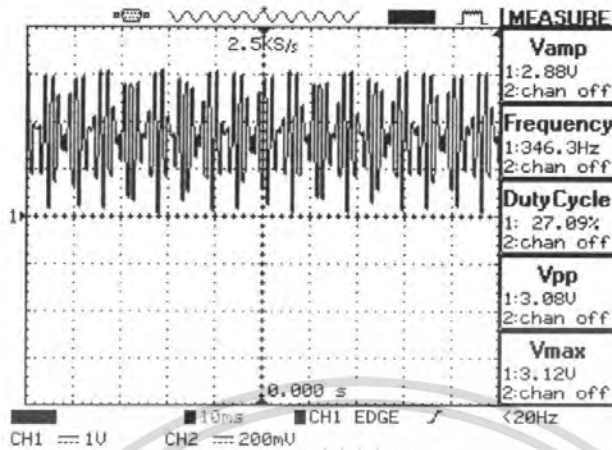


รูปที่ 4.34 รูปแบบสัญญาณเสียง Busy Tone ที่สร้างจาก โปรแกรม MATLAB

ด้านบน : แสดงสัญญาณเสียงที่สร้าง

ด้านล่าง : แสดงสัญญาณเสียงที่ยกระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 รูปแบบสัญญาณเสียง Busy Tone ที่วัดได้

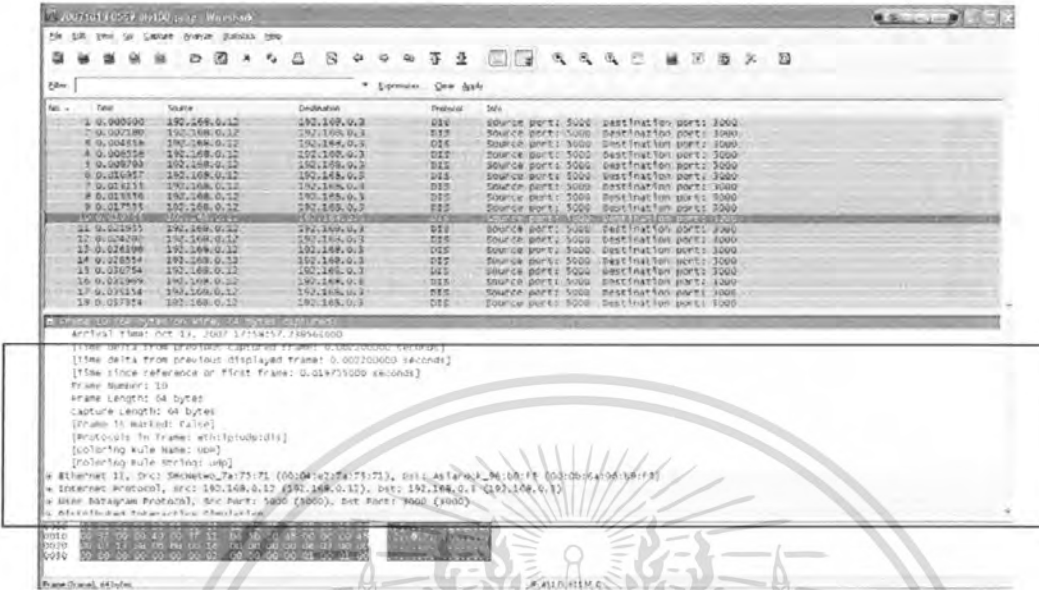
4.8 การดักจับข้อมูล (Sniffer) โดยโปรแกรม Wire Shark

ทำการดักจับข้อมูลที่จุดเชื่อมต่อเครือข่าย Ethernet

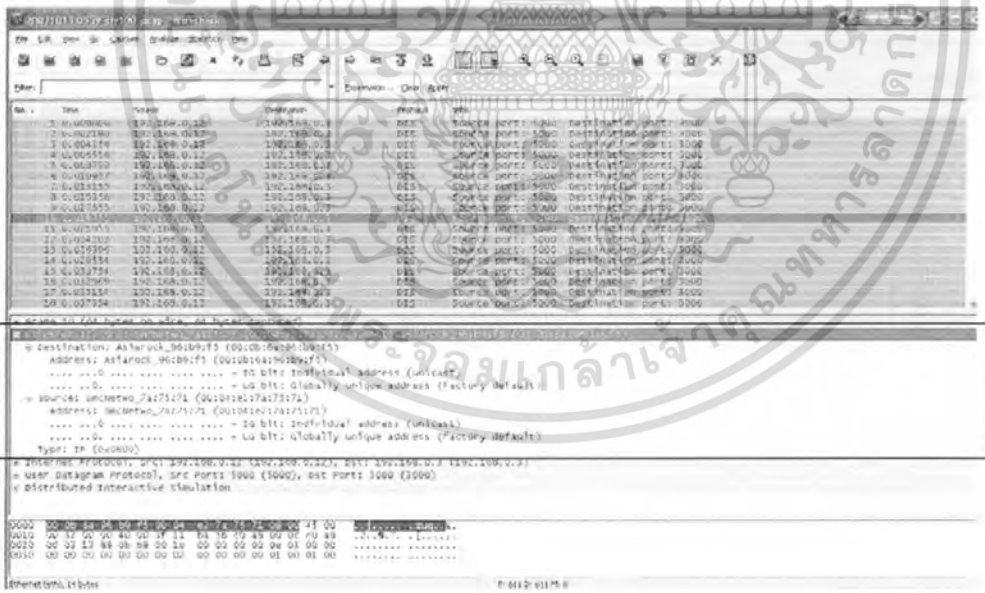


รูปที่ 4.36 การดักจับแพคเกจจาก Ethernet module
แบ่งส่วนแสดงผลออกเป็น 5 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

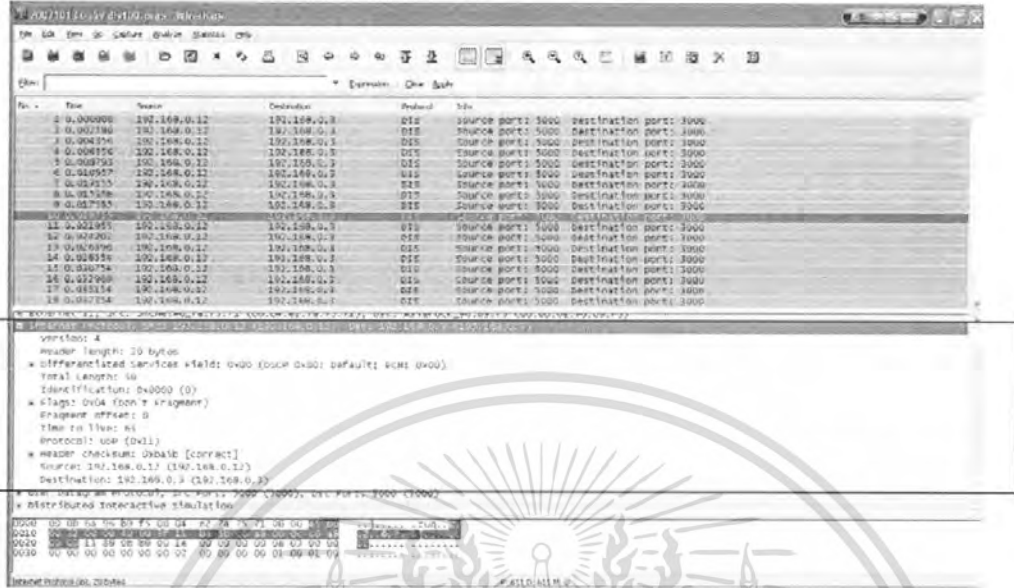


รูปที่ 4.37 รายละเอียดของเฟรมข้อมูลที่รับเข้า แสดง เวลา ขนาดของเฟรม และ โปรโตคอลของแพคเกจ

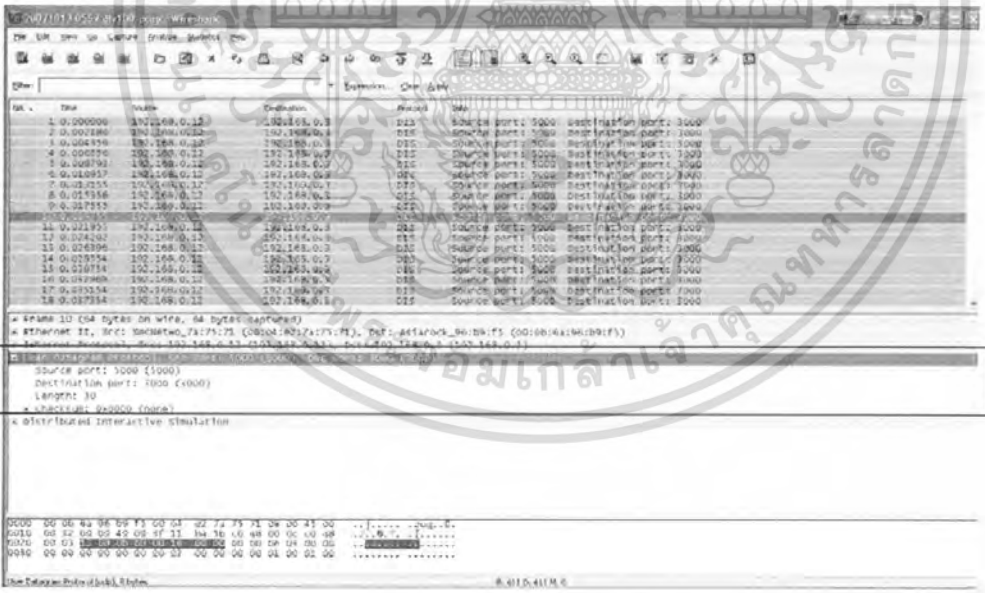


รูปที่ 4.38 รายละเอียดของการ Encapsulation ของ Layer 2 แสดงค่า Source และ Destination address (MAC ADDRESS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

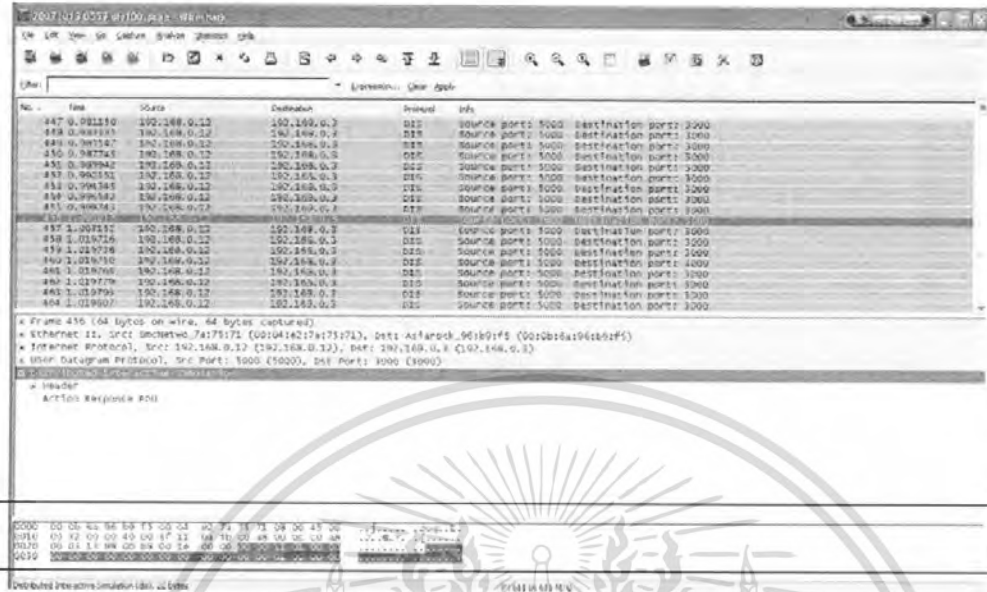


รูปที่ 4.39 รายละเอียดของการ Encapsulation ของ Layer 3 แสดง Source และ Destination address (IP ADDRESS)



รูปที่ 4.40 รายละเอียดของการ Encapsulation ของ Layer 4 แสดงรูปแบบ Segment UDP และ Port ที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 เลขเฟรมข้อมูลที่เวลา 1 วินาที โดยมีค่า 455 เฟรม/วินาที

จากการทดสอบดักจับข้อมูล ใน 1 เฟรมข้อมูล มี Data 22 Bytes จึงสามารถคำนวณ

$$\text{อัตราเร็วของการส่งข้อมูล} = 22 \times 455$$

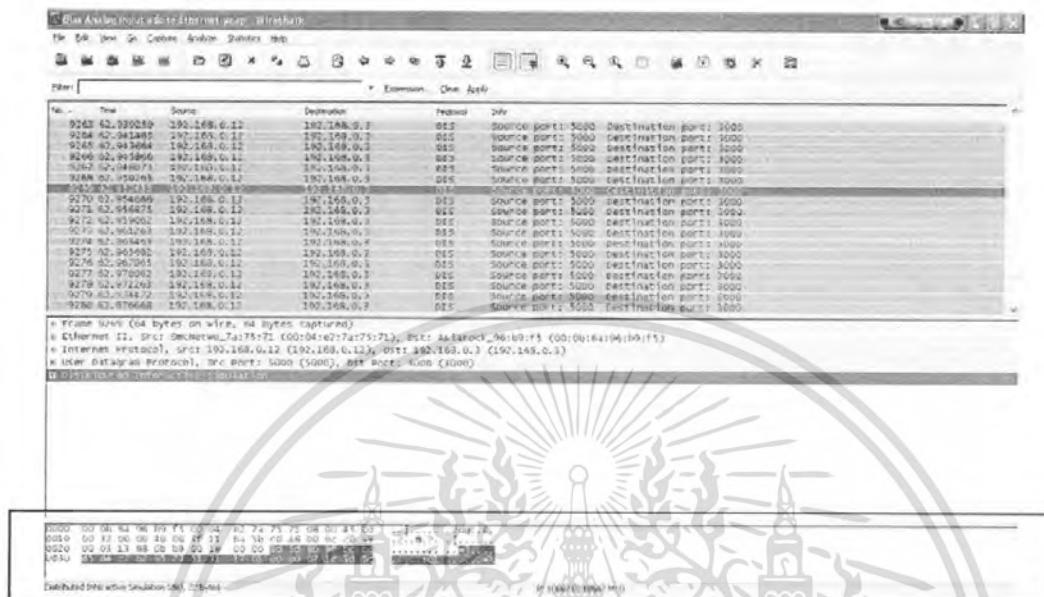
$$= 80,080 \text{ bps}$$

$$\text{ความถี่การ Sampling} = 22 \times 455$$

$$= 10,010 \text{ Hz}$$

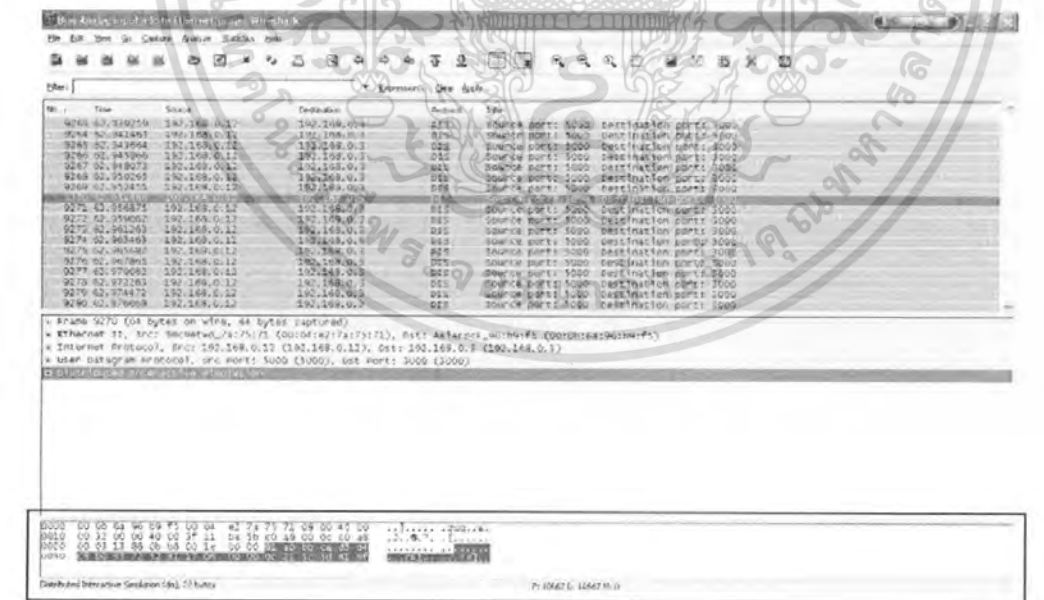
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8.1 การดักจับข้อมูลที่ส่งออกโดยการ Sampling สัญญาณ Sine โดยโปรแกรม Wire shark



รูปที่ 4.42 ผลการดักจับเฟรมข้อมูล (ก)

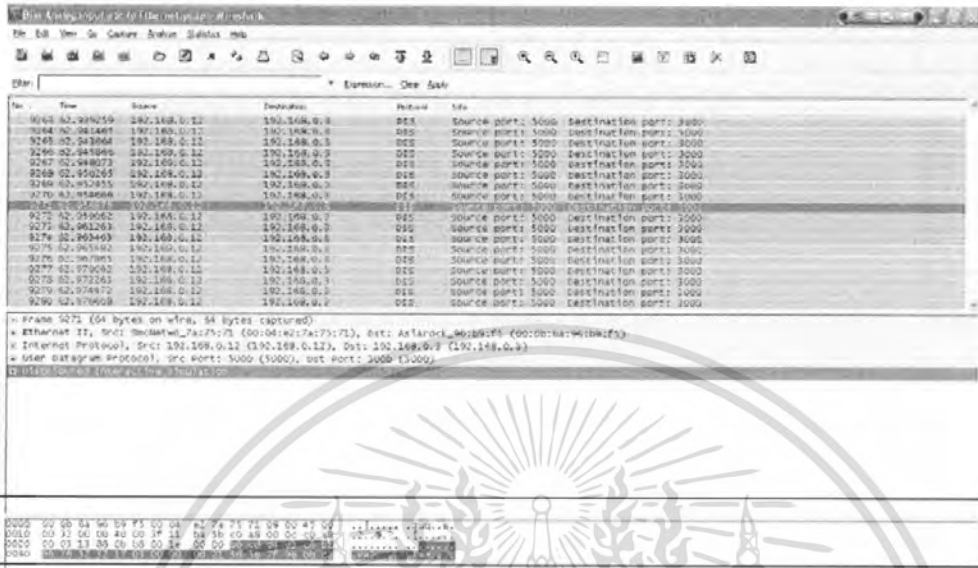
ข้อมูลเฟรม (ก): 3D 5D 80 9F BC CC D5 D4 C7 B2 95 73 51 31 17 01 00 00 07 1F 3D 5C



รูปที่ 4.43 ผลการดักจับเฟรมข้อมูล (ข)

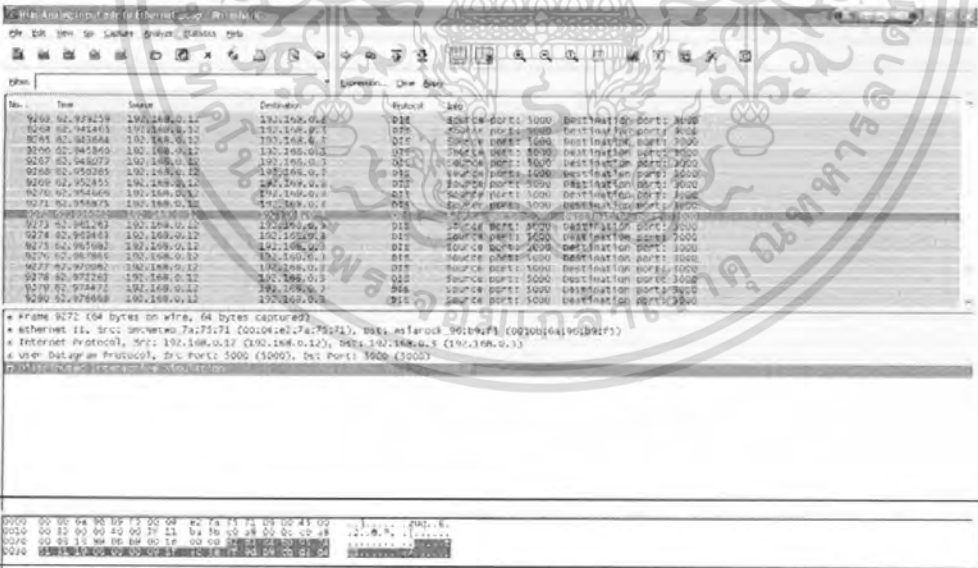
ข้อมูลเฟรม (ข): 81 A0 BB CE D5 D4 C8 B0 93 72 52 31 17 06 00 00 0C 21 3C 5D 81 9F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.44 ผลการดักจับเฟรมข้อมูล (ค)

ข้อมูลเฟรม (ค): B6 CD D4 D3 C6 B1 96 76 52 32 17 03 00 00 0D 21 3D 5E 7F 9E BB CA



รูปที่ 4.45 ผลการดักจับเฟรมข้อมูล (ง)

ข้อมูลเฟรม (ง): D2 D3 C4 B0 95 74 51 31 19 06 00 00 09 1F 3C 5E 7F 9D B9 CB D3 D4

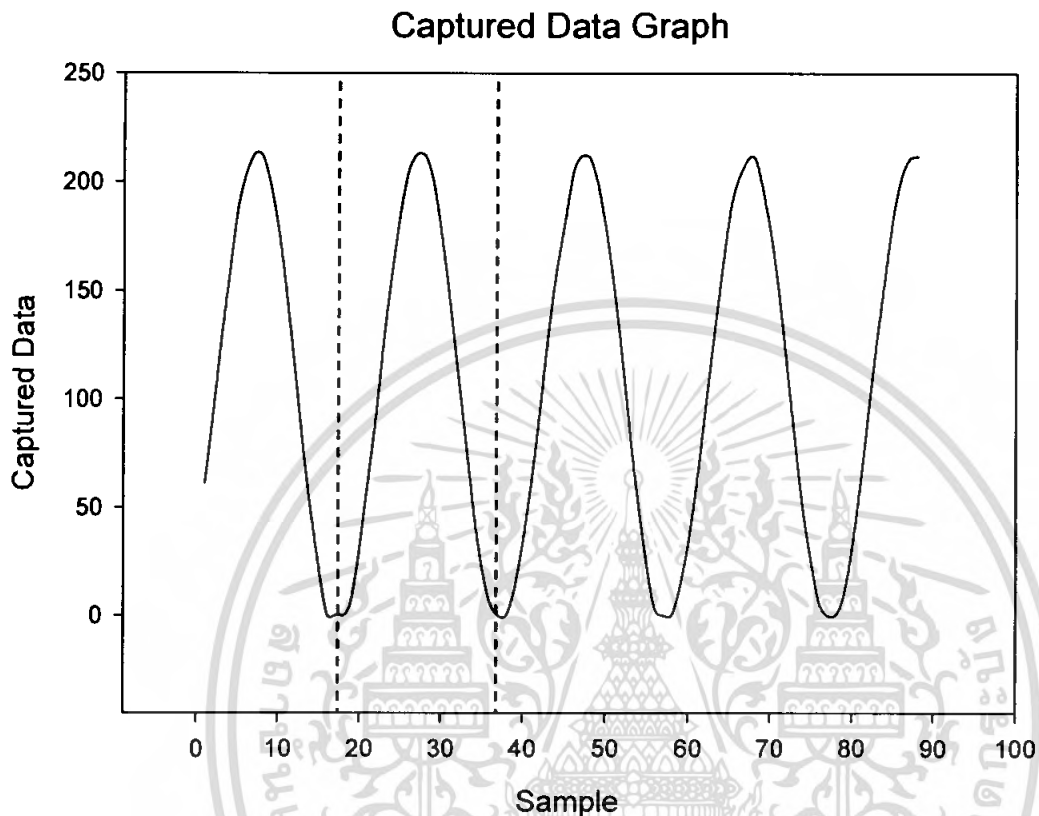
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การแปลงค่าข้อมูลที่จับได้จากฐานสิบหกเป็นฐานสิบ
จากเฟรมข้อมูล (ก), (ข), (ค) และ (ง) ในรูปที่ 4.42 – 4.45

ข้อมูลเฟรม (ก)		ข้อมูลเฟรม (ข)		ข้อมูลเฟรม (ค)		ข้อมูลเฟรม (ง)	
ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ
3D	61	81	129	B6	182	D2	210
5D	93	A0	160	CD	205	D3	211
80	128	BB	187	D4	212	C4	196
9F	159	CE	206	D3	211	B0	176
BC	188	D5	213	C6	198	95	149
CC	204	D4	212	B1	177	74	116
D5	213	C8	200	96	150	51	81
D4	212	B0	176	76	118	31	49
C7	199	93	147	52	82	19	25
B2	178	72	114	32	50	06	6
95	149	52	82	17	23	00	0
73	115	31	49	03	3	00	0
51	81	17	23	00	0	09	9
31	49	06	6	00	0	1F	31
17	23	00	0	0D	13	3C	60
01	1	00	0	21	33	5E	94
00	0	0C	12	3D	61	7F	127
00	0	21	33	5E	94	9D	157
07	7	3C	60	7F	127	B9	185
1F	31	5D	93	9E	158	CB	203
3D	61	81	129	BB	187	D3	211
5C	92	9F	159	CA	202	D4	212

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.2 เมื่อนำมาเขียนในรูปกราฟจึงได้ผลแสดงดังรูปที่ 4.46



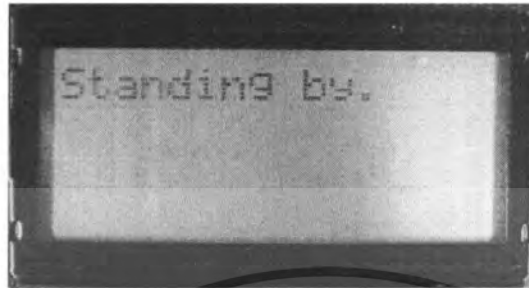
รูปที่ 4.46 กราฟแสดงข้อมูลของสัญญาณ sine ความถี่ 500 Hz ที่ดักจับได้
 $1Sample = 0.1ms, \Delta T = 20Sample, f = 500Hz$

จากการทดลอง อัตราการสูญหายของข้อมูล (Packet Loss) ที่วัดได้มีค่า 0 เปอร์เซ็นต์โดยในการทดลองนี้ ได้ทำการเชื่อมต่อ IP Phone ทั้งสองเครื่องโดยใช้สวิสต์เพียงตัวเดียว ทั้งนี้อัตราการสูญหายของข้อมูลอาจเปลี่ยนแปลงได้ โดยขึ้นกับสภาวะแวดล้อมซึ่งอาจเกิดจาก

- ความคับคั่งของข้อมูลเนื่องจากการใช้งานมาก
- การใช้อุปกรณ์ต่อพ่วงที่ไม่สมดุลกัน
- การชำรุดเสียหายของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การแสดงผลของหน้าจอ LCD



รูปที่ 4.47 ผลหน้าจอ LCD ในโหมดปกติ



รูปที่ 4.48 ผลหน้าจอ LCD ในโหมดทำการติดต่อโดยกำหนดหมายเลขปลายทาง



รูปที่ 4.49 ผลหน้าจอ LCD ในโหมดทำการติดต่อปลายทางไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

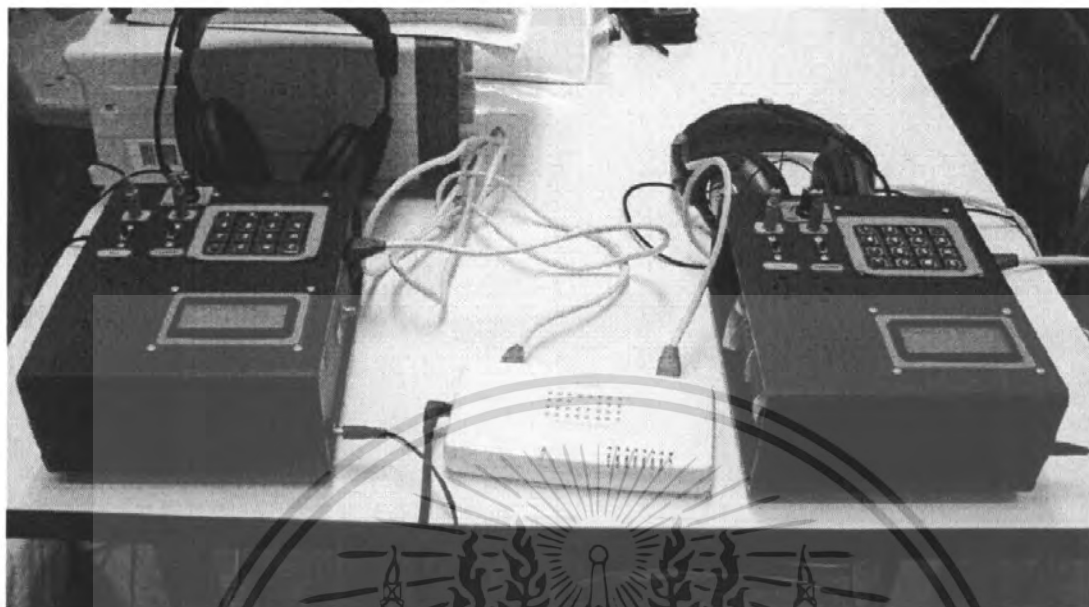


รูปที่ 4.50 ผลหน้าจอ LCD ในโหมดทำการติดต่อของทางฝั่งส่ง
โดยทำการติดต่อไปยังปลายทางหมายเลข IP 192.168.100.102



รูปที่ 4.51 ผลหน้าจอ LCD ในโหมดได้รับการติดต่อของทางฝั่งรับ
โดยได้รับการติดต่อจากต้นทางหมายเลข IP 192.168.100.101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.52 ชิ้นงาน IP Phone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และบทสรุป

โครงการนี้ได้ทำการออกแบบระบบโทรศัพท์บนระบบเครือข่ายไอพี ซึ่งเป็นเทคโนโลยี Voice over IP โดยทำการออกแบบอุปกรณ์ที่ใช้ส่งข้อมูลสัญญาณเสียงเข้าไปบนระบบโครงข่ายไอพี โดยใช้อุปกรณ์ความเร็วสูง ซึ่งมีการประมวลผลแบบ 32 บิตและมีการเขียนซอฟต์แวร์แบบ Multi tasking เพื่อให้ระบบทำงานในลักษณะของ Real-time OS เพื่อเพิ่มประสิทธิภาพของการประมวลผลและส่งข้อมูลเสียงให้มีประสิทธิภาพทันต่อเวลาที่ต้องการ หรือเป็นระบบเรียลไทม์ตามความต้องการ จากการทดสอบประสิทธิภาพรวมไปถึง ปรับปรุงการทำงานของวงจร และ โปรแกรมสำหรับการติดต่อสื่อสารระหว่างเครื่องโทรศัพท์บนโครงข่ายไอพี ได้ผลเป็นที่น่าพอใจเป็นไปตามความต้องการ

ปัญหาและแนวทางแก้ไข

เนื่องจากระบบที่ออกแบบต้องการความเร็วในการประมวลผลสูง ในครั้งแรกได้มีการเลือกใช้ CPU เบอร์ ADUC812 และ ADUC842 ซึ่งเป็นคอนโทรลเลอร์แบบ 8 บิตในตระกูล MCS-51 แต่เนื่องจากมีความต้องการของระบบสูงทั้งในด้านความเร็วและหน่วยความจำ จึงมีการปรับปรุงระบบโดยการเปลี่ยนมาใช้ CPU แบบ 32 บิตในตระกูล ARM7 เบอร์ LPC2148 ซึ่งมีความเร็วและหน่วยความจำสูงกว่า เพื่อให้สามารถรองรับต่อการพัฒนาระบบได้ดียิ่งขึ้น

ปัญหาอีกประการหนึ่งคือ ในการออกแบบวงจรไมโครโฟนครั้งแรก สัญญาณเสียงที่ได้มีสัญญาณรบกวนสูงมาก จึงได้ทำการออกแบบวงจรใหม่โดยมีการรับสัญญาณเสียงจากภายนอกเพิ่มเข้ามาอีกเส้นทางหนึ่ง เพื่อใช้ในการตัดสัญญาณเสียงรบกวนจากภายนอก ซึ่งทำให้ประสิทธิภาพของสัญญาณเสียงที่ได้มีคุณภาพดีขึ้น

เอกสารอ้างอิง

- [1] โอภาส ศิริครรชิตถาวร, เรียนรู้และพัฒนาไมโครคอนโทรลเลอร์ ARM7LPC2148 ด้วยภาษาซี, วชิรินทร์สาส์น, 2549.
- [2] วิรินทร์ เมฆประดิษฐสิน, กัมภีร์ระบบเครือข่าย, ซีเอ็ด, 2547.
- [3] Martin R. Arick, Ph.D, The TCP/IP Companion A Guide For The Common User, QED Publishing Group, 1993.
- [4] Eric Harper, Inside TCP/IP, New Riders Publishing, 1994.
- [5] วิวัฒน์กิรานนท์, วิศวกรรมการสื่อสาร, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2540.
- [6] www.wikipedia.com



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้