

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เว็บเซิร์ฟเวอร์แบบฝังตัว



เลขหมู่.....
เลขทะเบียน..... **62305**
วัน,เดือน,ปี..... - 8 ส.ค. 2549

b. 14620109
i.....

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา พ.ศ. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Embedded Web Server



A Special Project Submitted in Partial Fulfillment of the Requirement for the Degree of

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพิเศษเรื่อง **Embedded Web Server**

นักศึกษา นายวรวิทย์ กองสุข
นายวรวุฒิ วิศวกรรม
ภาควิชา ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สาขาวิชา ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา รศ.วิชิต สิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ประธานกรรมการ ดร.ปิติพร दनอมงาม	
กรรมการ ผศ.สาหร่าย เล็กชะอุ่ม	
กรรมการ อ.ภูมินทร์ จินดาจิราวัฒน์	
กรรมการที่ปรึกษา รศ.วิชิต สิริโชติ	

(รองศาสตราจารย์วิชาญ เตชิตธีระ)
หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพิเศษเรื่อง	Embedded Web Server		
นักศึกษา	นายวรวิทย์	กongsu	
	นายวรวุฒิ	วิศวกรรม	
ภาควิชา	ฟิสิกส์ประยุกต์	คณะ	วิทยาศาสตร์
สาขาวิชา	ฟิสิกส์ประยุกต์		
ปีการศึกษา	2548		
อาจารย์ที่ปรึกษา	รศ.วิชุด ศิริโชติ		

บทคัดย่อ

โครงการพิเศษนี้ได้ออกแบบระบบเซิร์ฟเวอร์แบบฝังตัว แผงวงจรเซิร์ฟเวอร์นี้สร้างด้วยไมโครโพรเซสเซอร์ขนาด 16 บิต ของ Intel และ อีเทอร์เน็ตชิพเบอร์ 8019 มีความเร็วในการรับ/ส่งข้อมูล 10 Mbit/s ของ Realtek ได้ใช้ Pacific C compiler พัฒนา Internet Protocol stack โปรโตคอลที่ประสบความสำเร็จในการทดสอบแล้วได้แก่ ARP ICMP UDP และ SNMP เซิร์ฟเวอร์นี้เชื่อมต่อกับ หัววัด ความชื้นและอุณหภูมิ เครื่องไคลแอนคร์ยะไกลซึ่งรันโปรแกรม SNMP viewer สามารถแสดงค่าความชื้นและอุณหภูมิได้อย่างสะดวก นอกจากนี้ยังได้ศึกษา โปรโตคอล TCP/IP และ HTTP อีกด้วย

Special Project Title	Embedded Web Server	
Name	Mr.Worawit	Kongsuk
	Mr.Worravut	Vissavakitcharoen
Department	Applied Physics	Faculty of Science
Program	Applied Physics	
Academic	2005	
Special Project Advisor	Assoc.Prof.Wichit Sirichote	

Abstract

An Embedded Web Server has been developed. The single board server is built with the 80C188, Intel 16-bit embedded microprocessor and the 8019, Realtek 10Mbit/s Ethernet chip. The Internet Protocol stack was developed with Pacific C compiler. The successfully tested protocols are ARP, ICMP, UDP and SNMP. The server provides Humidity/Temperature sensor interface. Client running SNMP viewer software can show the reading of humidity and temperature easily. The TCP/IP and HTTP have been also studied.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการพิเศษนี้สำเร็จลุล่วงไปได้ด้วยดี ต้องขอขอบคุณ รองศาสตราจารย์วิชิต ศิริโชติ อาจารย์ที่ปรึกษาโครงการพิเศษนี้ โดยให้ทั้งความรู้ คำปรึกษาและข้อเสนอแนะในการแก้ปัญหาต่าง ๆ อย่างสม่ำเสมอ อีกทั้งบิดา มารดา ครอบครัว เพื่อน ๆ พี่ ๆ น้อง ๆ ที่มีความห่วงใยและให้การช่วยเหลือตลอดมา นอกจากนี้ยังมีท่านอาจารย์อีกหลาย ๆ ท่าน ซึ่งไม่สามารถกล่าวในที่นี้ได้ทั้งหมด ที่ให้ความรู้ คำปรึกษา ตั้งแต่พื้นฐาน จนถึงขั้นที่สามารถนำไปประยุกต์ใช้งานได้ ทำให้การทำโครงการพิเศษนี้มีความรู้สะสมเรื่อยมาจนถึงปัจจุบัน ทำให้มีโอกาสสร้างโครงการพิเศษนี้ขึ้นมา

นายวรวิทย์ กองสุข
นายวรวุฒิ วิศวกิจเจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญ (ต่อ)	จ
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงานพิเศษ	1
1.2 วัตถุประสงค์	3
1.3 ขอบเขตในการศึกษา	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	4
1.5 ระยะเวลาในการดำเนินงาน	5
บทที่ 2 ทฤษฎีและหลักการ	6
2.1 โพรโทคอลเออาร์พี (ARP)	6
2.1.1 หลักการทำงานของ Address Resolution Protocol (ARP)	6
2.1.2 เออาร์พีเคทาแกรม (ARP Data gram)	8
2.2 โพรโทคอลไอซีเอ็มพี (ICMP)	9
2.2.1 ไอซีเอ็มพีชนิด 0 และ 8	11
2.2.2 Ping	11
2.2.2.1 ตัวอย่างการใช้ ping	12
2.3 โพรโทคอล ยูดีพี User Datagram Protocol (UDP)	14
2.3.1 ยูดีพีเคทาแกรม	15
2.3.2 การคำนวณค่าผลรวมตรวจสอบ	15
2.4 โพรโทคอลทีซีพี	17
2.4.1 บริการรับประกันความเชื่อถือของทีซีพี	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.2 ทีซีพีเฮคเตอร์	18
2.4.3 กลไกการทำงานของทีซีพี	20
2.4.3.1 การสถาปนาการเชื่อมโยง	20
2.4.3.2 การถ่ายโอนข้อมูล	21
2.4.3.3 การยกเลิกการเชื่อมต่อ	22
2.5 เอชทีทีพี (HTTP)	22
2.5.1 เอชทีทีพี 1.1	23
2.6 ไมโครโปรเซสเซอร์	23
2.6.1 ไมโครโปรเซสเซอร์ 8088/8086	24
2.7 ไมโครโปรเซสเซอร์ 80C188EB	25
2.7.1 คุณสมบัติของไมโครโปรเซสเซอร์ 80C188EB	25
2.7.2 โครงสร้างภายในของไมโครโปรเซสเซอร์ 80C188EB	26
2.7.3 รีจิสเตอร์และเซกเมนต์	28
2.8 RTL8019AS Network Interface Controller (NIC)	31
2.8.1 คุณสมบัติของ RTL8019AS	31
บทที่ 3 วิธีดำเนินการวิจัย	32
3.1 การออกแบบฮาร์ดแวร์	32
3.1.1 ส่วนประกอบทางฮาร์ดแวร์ของเว็บเซิร์ฟเวอร์แบบฝังตัว	32
3.1.2 การเชื่อมต่อไอซีตรวจจับอุณหภูมิ SHT15 เข้ากับ 80C188EB	34
บทที่ 4 ผลการทดลองและอภิปรายผล	36
4.1 ผลการทดลอง	36
4.1.1 ผลการทดลองโดยใช้โปรแกรม Ping ของวินโดวส์	36
4.1.2 ผลการทดลองโดยใช้โปรแกรม SNMPView 2.5	38
บทที่ 5 สรุปและข้อเสนอแนะ	41
5.1 สรุป	41
5.2 ข้อเสนอแนะ	41
เอกสารอ้างอิง	43
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 1.1 เปรียบเทียบมาตรฐาน ISO/OSI กับมาตรฐาน TCP/IP	3
ตารางที่ 2.1 ประเภทความผิดพลาดของไอซีเอ็มพี	9
ตารางที่ 3.1 แสดงตัวอย่างของค่าอุณหภูมิต่อหน่วยที่อ่านได้จาก DS18S20	40



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 แสดงการทำงานระหว่างเครื่องแม่ (Web Server) กับเครื่องลูก (Client)	1
รูปที่ 1.2 แสดงการทำงานของระบบเว็บเซิร์ฟเวอร์แบบฝังตัว	2
รูปที่ 2.1 การทำงานของเออาร์พี	7
รูปที่ 2.2 เออาร์พีเคทาแกรม	9
รูปที่ 2.3 ข่าวดสารของไอซีเอ็มพีบรรจุอยู่ในไอพี	9
รูปที่ 2.4 เคทาแกรมไอซีเอ็มพีชนิด 0 และ 8	12
รูปที่ 2.5 ยูดีพีเคทาแกรม	15
รูปที่ 2.6 ยูดีพีและเฮดเดอร์	16
รูปที่ 2.7 ทีซีพีเฮดเดอร์	18
รูปที่ 2.8 แสดงภาพไมโครโปรเซสเซอร์ 8088 (a) และ 8086 (b) ตามลำดับ	24
รูปที่ 2.9 แสดงขาต่างๆ ของไมโครโปรเซสเซอร์ 80C188EB	25
รูปที่ 2.10 โครงสร้างภายในไมโครโปรเซสเซอร์ 80C188EB	26
รูปที่ 2.11 โครงสร้างรีจิสเตอร์แบบ 16 บิต และแบบ 8 บิต	28
รูปที่ 2.12 อินเด็กซ์รีจิสเตอร์ และ พ้อยเตอร์ รีจิสเตอร์ แบบ 16 บิต และแบบ 8 บิต	29
รูปที่ 2.13 เซกเมนต์รีจิสเตอร์	30
รูปที่ 2.14 รายละเอียดขาต่างๆ ของ RTL8019AS	31
รูปที่ 3.1 แสดงส่วนประกอบทางฮาร์ดแวร์ของเว็บเซิร์ฟเวอร์แบบฝังตัว	33
รูปที่ 3.2 แสดงการจัดการหน่วยความจำรวมและแรม พร้อมตำแหน่งของ RTL8019AS ใน Memory Map	33
รูปที่ 4.1 แสดงคำสั่งที่ใช้ในการ ping	36
รูปที่ 4.2 แสดงผลการ ping จากวินโดวส์	37
รูปที่ 4.3 แสดงผลการอ่านค่าข้อมูลจากการเรียกดู จากโปรแกรม SNMPView ผ่านระบบเน็ตเวิร์ค	38
รูปที่ 4.4 แสดงข้อมูลทั้งหมดที่โปรแกรม SNMPView ทำการเก็บไว้ได้	39
รูปที่ 4.5 แสดงข้อมูลผ่านบราวส์เซอร์	

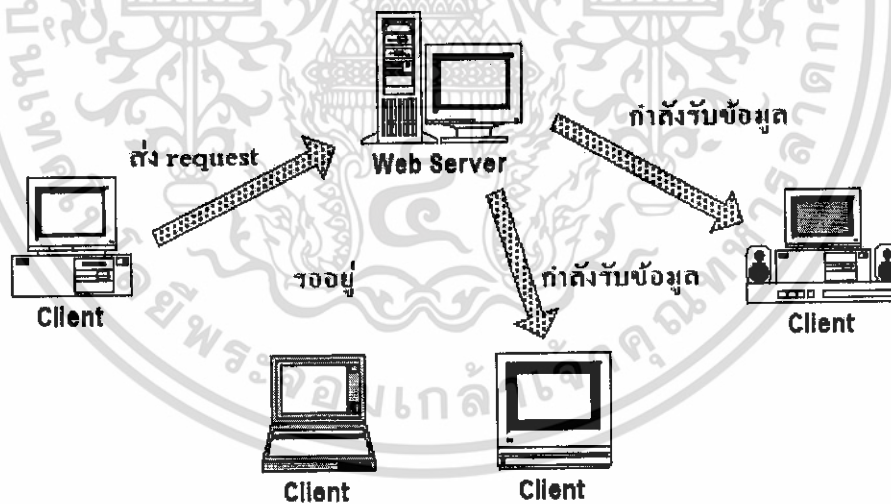
บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการพิเศษ

ในปัจจุบันระบบเครือข่ายอินเทอร์เน็ตมีการเจริญเติบโตและมีความสำคัญในด้านต่างๆ มากขึ้น เช่น ด้านการสื่อสารข้อมูล การแลกเปลี่ยนข้อมูล โดยการแสดงผลต่างๆ จะกระทำผ่านเว็บเบราว์เซอร์ โดยข้อมูลบนเว็บเบราว์เซอร์เหล่านั้นจะถูกเก็บเอาไว้บนเครื่องเว็บเซิร์ฟเวอร์หรือเครื่องแม่อีกทีหนึ่ง

เครื่องที่นำมาใช้เป็นเว็บเซิร์ฟเวอร์ในปัจจุบันนั้นมีหลายชนิด เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray ซึ่งแต่ละชนิดก็มีข้อดีและข้อเสียแตกต่างกันออกไป โดยลักษณะการทำงานของเครื่องเซิร์ฟเวอร์จะเป็นการรับการร้องขอและการส่งข้อมูลกลับให้กับเครื่องลูก (Client) นั้นเอง ดังรูปที่ 1.1

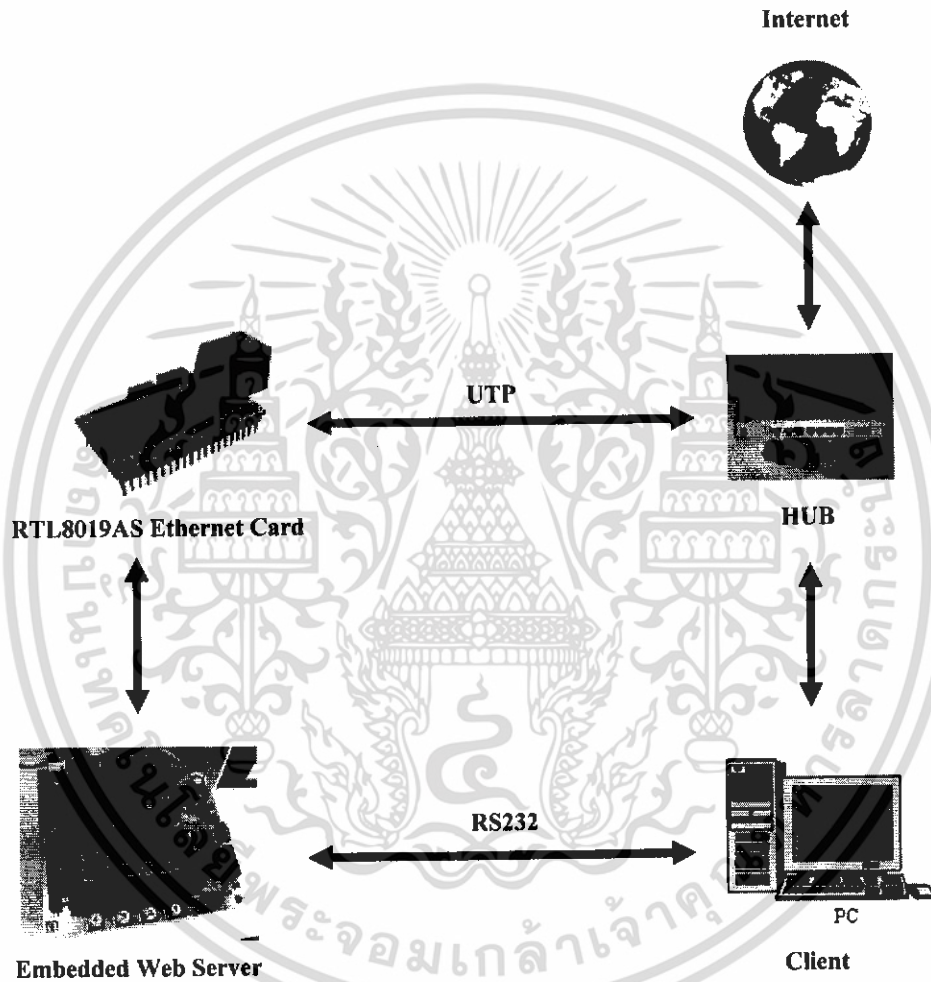


รูปที่ 1.1 แสดงการทำงานระหว่างเครื่องแม่ (Web Server) กับเครื่องลูก (Client)

จากความสำคัญของเครื่องเว็บเซิร์ฟเวอร์จึงทำให้เกิดโครงการนี้ขึ้นมา โดยจะทำการลดขนาดของเครื่องเว็บเซิร์ฟเวอร์ให้เล็กลงมา โดยจะนำระบบเว็บเซิร์ฟเวอร์แบบฝังตัวมาใช้เป็นเครื่องเซิร์ฟเวอร์แทน ซึ่งจะใช้ไมโครโปรเซสเซอร์ขนาด 16 บิต 80C188EB ต่อร่วมกับการ์ดอีเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTL8019AS มาตรฐาน IEEE802.3 อีเทอร์เน็ต 10BaseT โดยการเขียนโปรแกรมควบคุมการ์ด อีเทอร์เน็ตนี้จะสนับสนุนมาตรฐาน NE2000 โดยจะใช้ภาษาซีในการพัฒนาโปรแกรมและใช้ Pacific-C Complier ช่วยในการแปลภาษาออกมาเป็นโปรแกรมเพื่อนำระบบเว็บเซิร์ฟเวอร์แบบฝังตัวนี้ติดต่อกับเน็ตเวิร์คผ่านทางสายแลน ซึ่งจะทำได้ผ่านระบบอินเทอร์เน็ตได้ทั่วโลกนั่นเอง ดังรูปที่ 1.2



รูปที่ 1.2 แสดงการทำงานของระบบเว็บเซิร์ฟเวอร์แบบฝังตัว

การที่เลือกใช้ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวนี้ก็เนื่องจากเป็นระบบที่เล็กและทำงานเฉพาะทางได้ดี แทนที่จะใช้คอมพิวเตอร์มาทำงานอย่างหนึ่งอย่างใด เช่น การวัด การควบคุม หรือการเก็บข้อมูล ก็จะเป็นการสิ้นเปลือง ถ้ามาใช้ระบบแบบฝังตัวก็จะมีต้นทุนที่ถูกกว่านั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1. เพื่อเป็นการเรียนรู้การทำงานของระบบเครือข่ายอินเทอร์เน็ต
2. เพื่อเป็นการเรียนรู้กลไกการทำงานของไมโครโปรเซสเซอร์ 16 บิต 80C188EB
3. เพื่อศึกษาการสร้างเซิร์ฟเวอร์ โดยใช้การ์ดแลนทำการเชื่อมต่อกับไมโครโปรเซสเซอร์
4. เพื่อศึกษาการเขียนโปรแกรมภาษาซีผ่าน Pacific-C Compiler
5. เพื่อศึกษาการเขียนโปรแกรมภาษาซีติดต่อบริเวณระหว่างการ์ดแลนกับไมโครโปรเซสเซอร์
6. เพื่อศึกษาการเขียนโปรแกรมภาษาซีให้ระบบเว็บเซิร์ฟเวอร์แบบฝังตัวติดต่อกับระบบเครือข่ายอินเทอร์เน็ต

1.3 ขอบเขตในการศึกษา

การทำงานของระบบเครือข่ายอินเทอร์เน็ตนั้นจะอาศัยการทำงานของมาตรฐานการสื่อสารแบบ TCP/IP ดังตารางที่ 1.1

Layer	ISO/OSI	TCP/IP (Internet)
7	Application	Telnet, FTP, SMTP, HTTP, DNS, BOOTP, DHCP, SNMP
6	Representation	
5	Session	
4	Transport	TCP, UDP
3	Network	IP, ICMP, IGMP
2	Data Link	Device Driver and Interface
1	Physical	Media

ตารางที่ 1.1 เปรียบเทียบมาตรฐาน ISO/OSI กับมาตรฐาน TCP/IP

โดยขอบเขตของโครงการนี้ก็คือจะทำการพัฒนาระบบเว็บเซิร์ฟเวอร์แบบฝังตัว โดยในที่นี้จะอาศัยมาตรฐาน IEEE 802.3 อีเทอร์เน็ต 10BaseT ในการเชื่อมต่อเข้าสู่ระบบอินเทอร์เน็ต และจะศึกษาโปรโตคอลการส่งข้อมูลตั้งแต่ระดับชั้นที่ 1 คือชั้น Physical หรือระบบฮาร์ดแวร์ของเซิร์ฟเวอร์จนถึงระดับชั้นที่ 7 คือชั้น Application นั่นก็คือส่วนของการทำ HTTP เพื่อการติดต่อกับระบบอินเทอร์เน็ตเพื่อการแสดงผลผ่านเว็บเพจต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้การทำงานของระบบเน็ตเวิร์คและสามารถประยุกต์ใช้งานได้
2. มีความเข้าใจการออกแบบระบบเว็บเซิร์ฟเวอร์แบบฝังตัว
3. สามารถพัฒนา TCP/IP ที่มีขนาดเล็กและเพียงพอต่อการใช้งานพื้นฐาน
4. เพื่อเป็นแนวทางในการพัฒนาการประมวลผลบนระบบเว็บเซิร์ฟเวอร์แบบฝังตัวต่อไป
5. ได้รู้จักการทำงานอย่างเป็นระบบทำให้สามารถทำงานร่วมกับผู้อื่นได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ระยะเวลาในการดำเนินงาน

	2548										2549			
	ม.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	ม.ค.	ก.พ.	มี.ค.	
ขั้นตอนและวิธีการดำเนินงาน														
1. ประกอบฮาร์ดแวร์	■													
2. ศึกษาระบบเน็ตเวิร์ก	■													
3. พัฒนาซอฟต์แวร์เน็ตเวิร์ก		■	■	■	■	■	■	■	■	■	■	■	■	■
4. ทดสอบระบบเน็ตเวิร์ก								■	■	■	■	■	■	■
5. พัฒนาโปรแกรมเพิ่มเติม											■	■	■	■
6. ตรวจสอบครั้งสุดท้าย												■	■	■
7. จัดทำเอกสารประกอบ	■	■	■	■	■	■	■	■	■	■	■	■	■	■

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 โพรโทคอลเออาร์พี (ARP)

Address Resolution Protocol หรือ ARP เป็นโปรโตคอลที่ออกแบบมาเพื่อใช้ในเครือข่ายที่สนับสนุน Broadcast หน้าที่ของ ARP ในเครือข่ายคือ ช่วยแปลง IP Address ไปสู่ Hardware Address เพื่อให้สถานีในเครือข่ายสามารถนำ Hardware Address ไปสร้างเฟรมระดับ Data link ได้ และเป็นเกณฑ์วิธีแยกที่อยู่และมาตรฐานในการหาหมายเลขตำแหน่ง IP บนระบบเครือข่ายแบบต่างๆ ARP จะจับคู่ระหว่างหมายเลข IP และ Physical Address ของมาตรฐานในระบบเครือข่ายต่างๆ

การสื่อสารระดับล่างระหว่างสถานีในเครือข่ายที่ใช้ฮาร์ดแวร์ซึ่งกำหนดอยู่ในการ์ดอินเตอร์เฟซ ดีไวซ์ใดเวอร์ในสถานีในเครือข่ายจะใช้ฮาร์ดแวร์แอดเดรสของสถานีปลายทางในเครือข่ายเพื่อนำมาใช้สร้างเฟรม Data link

การทำ Hardware Address อาศัยการค้นจากรายชื่อซึ่งเก็บคู่ของ IP Address และ Hardware Address การสร้างรายชื่อและการค้นหามีรูปแบบที่กำหนดด้วยโปรโตคอล ARP รายชื่อที่เก็บคู่แอดเดรส เรียกว่า ARP Table

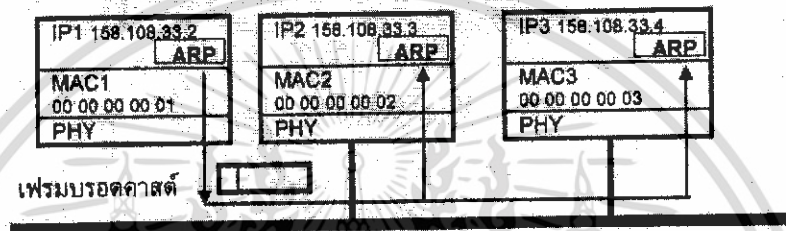
หากวิเคราะห์จากลักษณะ โครงสร้างของเฟรมบนมาตรฐานอีเทอร์เน็ต และ IP แล้ว จะพบว่าขนาดของหมายเลขตำแหน่งหรือ Address ซึ่งใช้บ่งบอกถึงที่อยู่บนระบบเครือข่ายของทั้งสองมาตรฐานนี้ มีความยาวแตกต่างกันโดย MAC Address ของอีเทอร์เน็ต นั้นมีขนาด 48 บิต ในขณะที่หมายเลขตำแหน่ง IP มีเพียง 32 บิต จากลักษณะดังกล่าวจึงเป็นที่มาของโปรโตคอล ARP เพื่อใช้ในการแปลง และค้นหา MAC Address ของอีเทอร์เน็ต หรือของ Data Link Layer บนมาตรฐานเครือข่ายแบบอื่น ของหมายเลข IP ที่ระบบต้องติดต่อสื่อสารด้วย

2.1.1 หลักการทำงานของ Address Resolution Protocol (ARP)

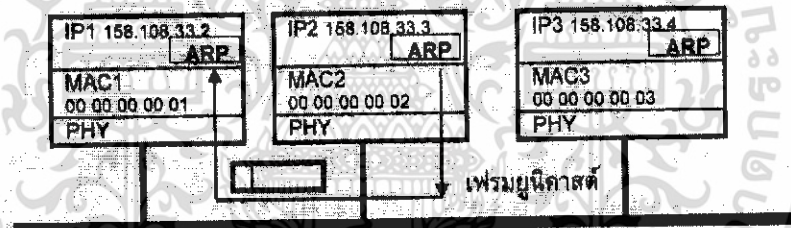
การทำงานของ ARP ทำงานโดยสถานีต้นทางที่ต้องการส่งแพ็กเก็ตไปยังสถานีปลายทาง จะเปิดตารางหาค่า Hardware Address ที่ตรงกับ IP Address จาก ARP Cache หากไม่พบค่าใน Cache สถานีจะสร้างเฟรม ARP และ Broadcast เดทากรมร้องขอ (ARP Request) โดยใส่ IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address ของสถานีที่ต้องการถามหา Hardware Address ทุกสถานีในเครือข่ายเมื่อได้รับเฟรม Broadcast แล้วจะทำการตรวจสอบ IP Address ประจำตัวกับ IP Address ที่ร้องขอ หากว่าตรงกัน กับสถานีนั้นก็จะส่ง เดตาแกรมตอบรับ (ARP Reply) โดยส่ง Hardware Address กลับไปยังสถานีที่ ร้องขอ สถานีต้นทางที่ได้รับคำตอบก็จะนำ Hardware Address ไปใช้งานและเก็บเข้าตาราง ใน ขณะเดียวกันสถานีที่ตอบรับก็จะนำค่า IP และ Hardware Address ของสถานีที่ร้องขอเก็บเข้าตาราง เช่นกันเพื่อใช้งานในโอกาสต่อไป



(ก) IP1 broadcast อีอาร์พีเพื่อสอบถามฮาร์ดแวร์แอดเดรสของ IP2



(ข) IP2 แจกฮาร์ดแวร์แอดเดรสกลับไปยัง IP1

รูปที่ 2.1 การทำงานของเออาร์พี

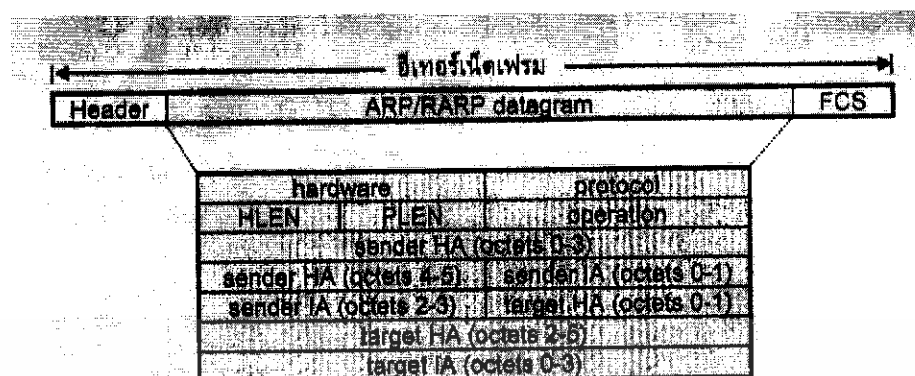
คู่แอดเดรสในตาราง ARP จะคงค่าอยู่เพียงระยะหนึ่งและจะถูกกำจัดออกเมื่อถึงเวลาที่ กำหนด เวลาที่กำหนดอายุค่าในตาราง ARP (ARP Cache timeout) อาจแตกต่างกันไปตามแต่ ระบบปฏิบัติการ โดยปกติแล้วหากไม่มีการใช้ค่าใดค่าหนึ่งประมาณ 15 นาที ค่านั้นก็จะถูกลบทิ้ง ออกไป วิธีนี้ช่วยให้ค่าในตาราง ARP เปลี่ยนแบบไดนามิกตามสภาพแวดล้อม

2.1.2 เออาร์พีเดทาแกรม (ARP Data gram)

เนื่องจาก ARP ทำงานในระดับ Data link เดทาแกรมของ ARP ก็จะบรรจุอยู่ในเฟรม Data link อย่างเช่น อีเทอร์เน็ต โดยมีค่าประจำโปรโตคอลเท่ากับ 0x0806 แต่ละฟิลด์มีความหมายดังนี้

1. Hardware 16 Bits : กำหนดชนิดของ Hardware เครื่องข่ายที่ ARP ทำงานอยู่ ค่าที่ใช้งานมีดังตัวอย่างต่อไปนี้
 - 1 อีเทอร์เน็ต
 - 4 โทเค็นริง
 - 5 เคออส (Chaos)
 - 6 เครื่องข่าย IEEE 802
 - 7 อาร์คเน็ต
 - 12 โลคัลทอสส์
2. Protocol 16 Bits : ชนิดของโปรโตคอลที่ร้องขอใช้ ARP
3. HLEN 8 Bits : ขนาดของ Hardware Address เป็นจำนวนไบต์ ค่าปกติที่ใช้งานคือ 6 ซึ่งเท่ากับขนาด 6 ไบต์ของอีเทอร์เน็ต Hardware Address
4. PLEN 8 Bits : ขนาดของแอดเดรสระดับเน็ตเวิร์คเป็นจำนวนไบต์ ค่าปกติที่ใช้คือ 4 ซึ่งเท่ากับขนาด 4 ไบต์ของ IP Address
5. Operation 16 Bits : กำหนดรูปแบบการใช้เดทาแกรม ค่าในฟิลด์นี้ใช้กำหนดการทำงานของทั้ง ARP และ RARP ซึ่งมี 4 ค่าคือ
 - ARP request (ค่าเท่ากับ 1)
 - ARP reply (ค่าเท่ากับ 2)
 - RARP request (ค่าเท่ากับ 3)
 - RARP reply (ค่าเท่ากับ 4)
6. Address : ฟิลด์แอดเดรสเรียงลำดับจาก Hardware และ Network Address ของสถานีที่ร้องขอ ตามด้วย Hardware Address ของสถานีที่ตอบรับ

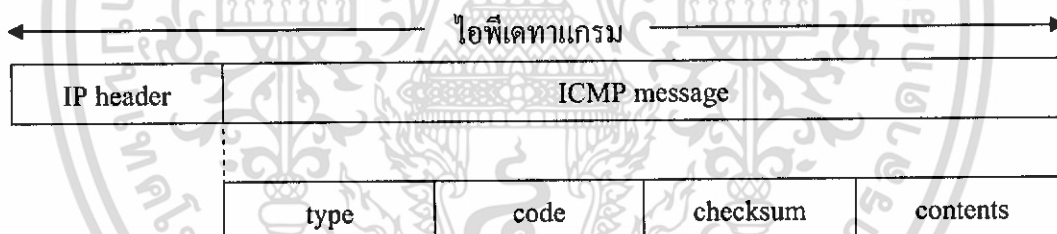
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 เออาร์พีเคทาแกรม

2.2 โพรโทคอลไอซีเอ็มพี (ICMP)

Internet Control Message Protocol หรือ ICMP มีการทำงานในระดับเดียวกันกับ IP โดยทำหน้าที่รายงานข้อผิดพลาดหรือสถานะการทำงานของ IP และแจ้งปัญหาที่เกิดจากการลำเลียง IP เคทาแกรม ภายในแพ็กเก็ต ICMP จะบรรจุข่าวสารบ่งบอกถึงข้อผิดพลาดและปัญหาที่เกิดขึ้น



รูปที่ 2.3 ข่าวสารของไอซีเอ็มพีบรรจุอยู่ในไอพี

แพ็กเก็ต ICMP จะใช้ IP เคทาแกรมสำหรับแจ้งข่าวสาร ข้อมูลของ IP เคทาแกรมก็คือข่าวสารที่บ่งบอกถึงปัญหาข้อผิดพลาด สาเหตุของปัญหาที่ทำให้ไม่สามารถส่งเคทาแกรมไปยังปลายทางมิได้มากมาย เช่น ไม่สามารถติดต่อโฮสต์ปลายทางได้เนื่องจากสารสื่อสารขัดข้อง หรือโฮสต์ทำงานผิดปกติ หรือเราเตอร์ระหว่างทางอาจกำจัดเคทาแกรมทิ้งไปเนื่องจากหน่วยความจำไม่พอเพียงที่จะเก็บเคทาแกรม รวมทั้งปัญหาอื่นอีกหลายสาเหตุ หากโฮสต์หรือเราเตอร์ต้องทิ้งเคทาแกรมเนื่องจากสาเหตุใดสาเหตุหนึ่ง ซอร์ฟแวร์ ICMP ภายในโฮสต์หรือเราเตอร์จะแจ้งปัญหากลับไปยังต้นทางทันทีเพื่อให้ทราบถึงปัญหานอกเหนือไปจากการรายงานความผิดพลาดแล้ว ยังมีการใช้ ICMP เพื่อการตรวจสอบและร้องขอข้อมูลจากสถานีปลายทางที่กำหนดได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ICMP จะจัดรูปแบบการรายงานข้อผิดพลาดโดยแบ่งฟิลด์ของข้อมูลในเคทาแกรมออกเป็น 4 ฟิลด์ โดยแต่ละฟิลด์มีความหมายดังนี้

- Type ขนาด 8 Bits : กำหนดทั้งค่าความผิดพลาดและการรายงานสถานะ การใช้งานในปัจจุบันมีทั้งหมด 15 ประการ
- Code ขนาด 8 Bits : รหัสความผิดพลาดย่อย
- Checksum ขนาด 16 Bits : ค่าผลรวมตรวจสอบแบบ 1's complement สำหรับใช้ตรวจสอบความผิดพลาด โดยคำนวณผลรวมของ Type, Code และ Contents
- Contents ขนาดไม่คงที่ : ฟิลด์นี้ใช้บรรจุข้อมูลข่าวสารเพิ่มเติมเพื่อแจ้งกลับซึ่งจะขึ้นอยู่กับค่า Type และ Code

ถึงแม้ว่า ICMP จะใช้ IP เคทาแกรมซึ่งอาจเกิดปัญหาเคทาแกรมไปไม่ถึงปลายทางได้ เช่นเดียวกับไอพีเคทาแกรมทั่วไป แต่จะไม่มีรายงานความผิดพลาดด้วย ICMP หากมีปัญหาที่เกิดขึ้นกับเคทาแกรมของ ICMP เอง ทั้งนี้เพื่อป้องกันการแจ้งความผิดพลาดแบบวนซ้ำ นอกจากนี้จะไม่ใช้ ICMP เมื่อเกิดความผิดพลาดของเคทาแกรมต่อไปนี้

- ไม่ใช้ ICMP กับ IP เคทาแกรมที่ใช้เพื่อบรอดคาสต์หรือมัลติคาสต์
- ไม่ใช้ ICMP กับ IP เคทาแกรมแฟรกเมนต์อื่นๆยกเว้นแฟรกเมนต์แรก กล่าวคือ ICMP จะตอบสนองกับเคทาแกรมแฟรกเมนต์แรกเท่านั้น
- ไม่ใช้ ICMP กับ IP เคทาแกรมที่แอดเรสต้นทางไม่ได้ใช้กำหนดเลขโฮสต์ เช่น แอดเรสรูปแบบ บรอดคาสต์ หรือมัลติคาสต์

ตารางที่ 2.1 ประเภทความผิดพลาดของไอซีเอ็มพี

type	ความหมาย	การใช้งาน
0	Echo reply	แจ้งการตอบรับ
3	Destination unreachable	ไม่สามารถติดต่อปลายทางได้
1	Source quench	ให้ต้นทางลดระดับภาระงาน
5	Redirect	แจ้งเส้นทางที่เหมาะสมกว่า
8	Echo request	ร้องขอการตอบรับ
9	Router advertisement	เราเตอร์แจ้งประกาศตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type	ความหมาย	การใช้งาน
10	Router solicitation	โฮสต์ค้นหาเราเตอร์
11	Time exceeded for datagram	เดตาแกรมใช้เวลาดำเนินการเกินกำหนด
12	Parameter problem on datagram	มีปัญหาพารามิเตอร์ในเดตาแกรม
13	Time stamp request	ร้องขอเวลาระบบ
14	Time stamp reply	แจ้งเวลาระบบกลับ
15	Information request	ร้องขอข่าวสาร
16	Information reply	แจ้งข่าวสารกลับ
17	Address mask request	ร้องขอแอดเดรสมาสก์
18	Address mask reply	แจ้งแอดเดรสมาสก์กลับ

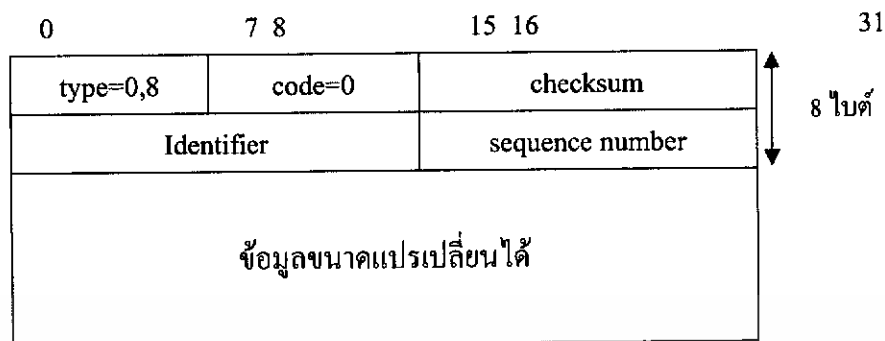
2.2.1 ไอซีเอ็มพีชนิด 0 และ 8

ไอซีเอ็มพีชนิด 0 และ 8 ใช้ตรวจสอบว่าสถานีหรือเราเตอร์ปลายทางยังทำงานอยู่หรือไม่ ตัวอย่างการใช้ประโยชน์จากไอซีเอ็มพีกลุ่มนี้คือ โปรแกรม ping

2.2.2 Ping

Ping จะส่ง ICMP ชนิด 8 เพื่อสอบถาม (request) ไปยังสถานีหรือเราเตอร์ที่ต้องการตรวจสอบและรอการตอบกลับ หากโฮสต์หรือเราเตอร์เปิดให้บริการอยู่ก็จะตอบกลับด้วย ICMP ชนิด 0 (reply) ต่อจากนั้น ping จึงนำผลการตอบกลับมาแสดงบนจอภาพฟอร์มเมตของเดตาแกรมที่ใช้ร้องขอและตอบกลับมีโครงสร้างเช่นเดียวกัน

โปรแกรม ping จะส่งเดตาแกรมต่อเนื่องกันไป การตอบกลับจะต้องตอบอ้างอิงกับทุกๆ เดตาแกรมด้วยค่า identifier และใช้ sequence number บอกลำดับเดตาแกรม



รูปที่ 2.4 เคทาแกรมไอซีเอ็มพีชนิด 0 และ 8

ฟิลด์ข้อมูลถัดมาเป็นข้อมูลตรวจสอบค่าคือเมื่อส่งเคทาแกรมที่บรรจุข้อมูลหนึ่งๆ ออกไปแล้ว ทางฝ่ายรับจะต้องส่งข้อมูลที่ได้รับกลับมาด้วย โดยปกติแล้ว ping จะใช้ 8 ไบต์แรกของข้อมูลเพื่อประทับเวลาสำหรับใช้คำนวณเวลาไปกลับ ขนาดข้อมูลปกติที่ใช้คือ 56 ไบต์และสามารถเลือกค่าอื่นได้

2.2.2.1 ตัวอย่างการใช้ ping

เพื่อแสดงถึงการทำงานของ ICMP โดยใช้ ping ขอพิจารณาตัวอย่างผลลัพธ์จากคำสั่ง ping ดังต่อไปนี้

```
$ ping std.cpc.ku.ac.th
PING std.cpc.ku.ac.th (158.108.2.68) : 56 data bytes
64 bytes from 158.108.2.68: icmp_seq=0 ttl=252 time=1.0 ms
64 bytes from 158.108.2.68: icmp_seq=1 ttl=252 time=1.0 ms
64 bytes from 158.108.2.68: icmp_seq=2 ttl=252 time=1.0 ms
64 bytes from 158.108.2.68: icmp_seq=3 ttl=252 time=1.0 ms
64 bytes from 158.108.2.68: icmp_seq=4 ttl=252 time=1.1 ms
^C
--- std.cpc.ku.ac.th ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.1 ms
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างนี้แสดงการใช้ ping ส่ง ICMP ชนิด 8 ไปยัง std.cpc.ku.ac.th โดยใส่ข้อมูล 56 ไบต์ไว้ในเดตาแกรม โฮสต์ std.cpc.ac.th จะตอบกลับมาด้วย ICMP ชนิด 0 ผลที่ได้มีความหมายดังนี้คือ

- ตัวเลข 64 ไบต์ซึ่งเป็นขนาดรวมของเดตาแกรมที่ได้รับ โดยคำนวณจากข้อมูลขนาด 56 ไบต์บวกกับเฮดเดอร์ของ ICMP อีก 8 ไบต์
- ค่า icmp_seq ลำดับเดตาแกรมตาม โดยนำมาจากฟิลด์ sequence number
- ค่า ttl ใช้คำนวณหาจำนวนเราเตอร์ที่เดตาแกรมเดินทางผ่าน คำสั่ง ping กำหนดค่า ttl เริ่มต้นเท่ากับ 255 และลดค่าลงครั้งละหนึ่งทุกครั้งที่ผ่านมาเราเตอร์ตัวเลข 253 จึงหมายความว่าเดตาแกรมเดินทางผ่านเราเตอร์ 2 ตัว
- เวลาเดินทางไปกลับ (round-trip time : RTT) ซึ่งนับเริ่มต้นจากการส่งแพ็กเก็ตและได้รับแพ็กเก็ตตอบกลับในหน่วยวินาที ping คำนวณเวลาเดินทางไปกลับโดยบรรจุเวลาไว้ในเดตาแกรมที่ส่งออกไป เมื่อได้รับเดตาแกรมตอบซึ่งบรรจุเวลานี้กลับมาด้วยก็นำมาลบกับเวลาปัจจุบัน

ค่า sequence ยังใช้เป็นตัวตรวจสอบด้วยว่าเดตาแกรมใดบ้างที่สูญหาย เพื่อนำมาใช้คำนวณสถิติ และเวลาเดินทางไปกลับโดยเฉลี่ยของเดตาแกรมดังเช่น ลองพิจารณาการทดสอบกับโฮสต์ที่อยู่ห่างไกลมักพบว่ามีแพ็กเก็ตสูญหายดังนี้

```

$ ping www.yahoo.com
PING www.yahoo.com (204.71.200.74) : 56 data bytes
64 bytes from 204.71.200.74: icmp_seq=0 ttl=252 time=2949.9 ms
64 bytes from 204.71.200.74: icmp_seq=1 ttl=252 time=2942.9 ms
64 bytes from 204.71.200.74: icmp_seq=2 ttl=252 time=2790.1 ms
64 bytes from 204.71.200.74: icmp_seq=3 ttl=252 time=2686.7 ms
64 bytes from 204.71.200.74: icmp_seq=4 ttl=252 time=2759.9 ms
64 bytes from 204.71.200.74: icmp_seq=7 ttl=252 time=2694.9 ms
--- www.yahoo.com ping statistics ---
8 packets transmitted, 6 packets received, 27% packet loss
round-trip min/avg/max = 1.0/1.0/1.1 ms
  
```

↑ เดตาแกรมลำดับ 5 และ 6 สูญหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างต่อไปนี้กำหนดให้ ping ส่งเคทาแกรมที่บรรจุข้อมูลขนาด 256 ไบต์จำนวน 4 เคทาแกรมต่อเนื่องออกไป

```
$ ping -s 256 -c 4 std.cpc.ku.ac.th
PING std.cpc.ku.ac.th (158.108.2.68) : 256 data bytes
264 bytes from 158.108.2.68: icmp_seq=0 ttl=252 time=1.0 ms
264 bytes from 158.108.2.68: icmp_seq=1 ttl=252 time=1.0 ms
264 bytes from 158.108.2.68: icmp_seq=2 ttl=252 time=1.0 ms
264 bytes from 158.108.2.68: icmp_seq=3 ttl=252 time=1.1 ms

--- std.cpc.ku.ac.th ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.1 ms
```

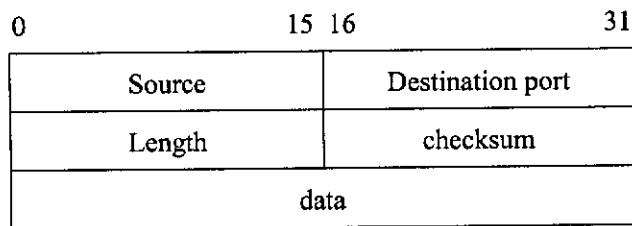
2.3 โพรโทคอล ยูดีพี User Datagram Protocol (UDP)

User Datagram Protocol หรือ UDP เป็นโพรโทคอลระดับทรานสปอร์ตทำหน้าที่นำส่งข้อมูลจากโพรโทคอลประยุกต์ไปยัง IP ข้อมูลรวมกับยูดีพีเฮดเดอร์เรียกว่า ยูดีพีเคทาแกรม หรือ ยูสเซอร์เคทาแกรม

ยูดีพีให้บริการแบบ “connectionless” กล่าวคือไม่สถาปนาการเชื่อมต่อระหว่างสถานีต้นทางและปลายทาง ยูดีพีส่งเคทาแกรมโดยไม่ตรวจสอบว่าสถานีปลายทางพร้อมที่จะติดต่อหรือไม่ การสื่อสารลักษณะนี้อาจเทียบได้กับการส่งจดหมาย ผู้ส่งเพียงแต่มอบหมายให้ไปรษณีย์จัดส่งโดยไม่ต้องทราบว่าผู้รับปลายทางพร้อมรับหรือไม่

หากมีปัญหาเกิดขึ้นกับยูดีพีเคทาแกรม เช่น เคทาแกรมสูญหาย หรือผิดลำดับหรือมีเคทาแกรมซ้ำกัน ยูดีพีจะไม่จัดการกับปัญหาเหล่านี้เนื่องจากไม่มีกลไกที่จะรับประกันความถูกต้องของเคทาแกรม โพรโทคอลประยุกต์ที่ใช้ยูดีพีต้องดำเนินการกับปัญหาเหล่านี้เอง

2.3.1 ยูติพีเคทาแกรม



รูปที่ 2.5 ยูติพีเคทาแกรม

ยูติพีเคทาแกรมประกอบด้วย

- Source port ขนาด 16 Bits : พอร์ตสถานีต้นทาง
- Destination port ขนาด 16 Bits : พอร์ตสถานีปลายทาง
- Length ขนาด 16 Bits : บอกความยาวของเคทาแกรม (ทั้งเฮดเคอร์และข้อมูล) เป็นจำนวนไบต์
- Checksum ขนาด 16 Bits : ผลรวมตรวจสอบ คำนวณจากผลรวมของเฮดเคอร์และข้อมูล

ยูติพีเคทาแกรมมีเฮดเคอร์ที่เรียบง่ายและมีฟิลด์จำนวนน้อย ฟิลด์ที่สำคัญมีเพียงพอร์ตต้นทางและปลายทาง และค่าผลรวมตรวจสอบโดยไม่มีฟิลด์อื่นใดใช้ในการรับประกันความเชื่อถือการส่งข้อมูล

2.3.2 การคำนวณค่าผลรวมตรวจสอบ

ยูติพีที่คำนวณผลรวมตรวจสอบจากเฮดเคอร์และข้อมูล ต่างจากผลรวมตรวจสอบของ IP ที่คำนวณจากเฮดเคอร์โดยไม่รวมข้อมูล ค่าผลรวมตรวจสอบของยูติพีเป็นค่าเพื่อเลือก (optional) ที่ใช้หรือไม่ก็ได้ ในขณะที่ที่ซีพีบังคับว่าต้องคำนวณผลรวมตรวจสอบเสมอ

ผลรวมตรวจสอบของยูติพีคำนวณโดยบวกค่า 16 Bits แบบเลขเติมเต็มหนึ่ง แต่ถ้าความยาวของยูติพีเคทาแกรมเป็นเลขคี่ให้แทรกค่า 0 อีก 1 ไบต์ เพื่อใช้เฉพาะขั้นตอนการคำนวณเท่านั้น (ไบต์ที่เพิ่มมาไม่ถูกส่งไปด้วย)

ยูติพีและที่ซีพีจะนำค่า เฮดเคอร์เทียม (psuedo-header) อีก 12 ไบต์เข้ามารวมคำนวณผลรวมตรวจสอบด้วย เฮดเคอร์เทียมนำมาจากบางฟิลด์ของ IP เคทาแกรม

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.4 โพรโตคอลทีซีพี

ทีซีพีเป็นโปรโตคอลที่ให้บริการชนิดที่จำเป็นต้องมีการเชื่อมต่อ และรับประกันความเชื่อถือในการลำเลียงข้อมูล ทีซีพีรับประกันความเชื่อถือโดยทำหน้าที่ตรวจสอบเซกเมนต์ที่ผิดปกติและจัดส่งเซกเมนต์ซ้ำใหม่ รวมทั้งจัดลำดับให้ถูกต้องก่อนส่งไปยังโปรแกรมประยุกต์ระดับบนเซคเตอร์และข้อมูลของทีซีพีเรียกว่า เซกเมนต์ (segment)

สถานีต้นทางจะต้องสถาปนาการเชื่อมโยงกับสถานีปลายทางก่อนส่งทีซีพีเซกเมนต์ การสถาปนาการเชื่อมโยงใช้ประโยชน์เพื่อให้มั่นใจว่าปลายทางพร้อมจะสื่อสารด้วย และเมื่อเสร็จสิ้นการส่งถ่ายข้อมูลแล้วก็จะปิดการเชื่อมโยง

การสถาปนาการเชื่อมโยงของทีซีพีอาจเปรียบเทียบกับได้กับการการติดต่อทางโทรศัพท์ กล่าวคือเมื่อหมุนหมายเลขปลายทางแล้ว ผู้เรียกต้องรอให้ปลายทางรับสาย เมื่อทักทายและแจ้งให้ทราบว่าใครเป็นผู้เรียกสายแล้วจึงเริ่มการสนทนา กระบวนการสถาปนาของทีซีพีมีขั้นตอนเฉพาะ

ทีซีพีทำงานตามแบบไคลเอ็นต์-เซิร์ฟเวอร์ ไคลเอ็นต์จะเป็นผู้ร้องขอบริการและขอสถาปนาการเชื่อมโยง ส่วนเซิร์ฟเวอร์รับการร้องขอและให้บริการต่อไคลเอ็นต์ การขนถ่ายข้อมูลระหว่างไคลเอ็นต์และเซิร์ฟเวอร์เป็นแบบฟูลดูเพล็กซ์ เสมือนมีท่อลำเลียงสองท่อต่อเชื่อมระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

ทีซีพีลำเลียงข้อมูลในรูปของไบต์ขนาด 8 บิต หรือมักเรียกว่าออกเตต (octet) และเรียกบริการลำเลียงแบบนี้ว่า บริการสายข้อมูลแบบไบต์ (byte stream service) โพรโตคอลประยุกต์ปลายทางไม่จำเป็นต้องรับข้อมูลมาดำเนินการครั้งละไบต์เสมอไป หากแต่ดำเนินการไปตามการทำงานของแต่ละโปรโตคอล

2.4.1 บริการรับประกันความเชื่อถือของทีซีพี

ทีซีพีให้บริการจัดการด้านความเชื่อถือการลำเลียงเซกเมนต์ที่สำคัญคือ การตรวจจับและแก้ไขข้อผิดพลาด, การควบคุมปริมาณการไหลข้อมูล, การจัดลำดับ, และการกำจัดเซกเมนต์ซ้ำ โดยมีรายละเอียดดังนี้

- การตรวจจับและแก้ไขข้อผิดพลาดมุ่งเน้นแก้ปัญหาหาเซกเมนต์ที่ผิดปกติจากปัญหาสายสื่อสารหรือโปรโตคอลระดับล่าง หรือเซกเมนต์เดินทางไม่ไปถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง 62305 อย่างอึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปลายทาง ทีซีพีจัดการกับปัญหานี้โดยใช้ผลรวมตรวจสอบเช่นเดียวกับยูดีพีหากค่าผลรวมตรวจสอบไม่ถูกต้อง ทีซีพีจะส่งเซกเมนต์ซ้ำใหม่

- เมื่อทีซีพีได้รับเซกเมนต์จะตอบกลับไปต้นทาง ทีซีพีใช้การตอบรับเพื่อยืนยันว่าได้รับข้อมูลอย่างถูกต้อง
- ทีซีพีจะตั้งเวลาเมื่อส่งเซกเมนต์เพื่อรอการตอบรับจากปลายทาง หากไม่มีการตอบรับภายในเวลาที่กำหนดทีซีพีจะส่งเซกเมนต์ซ้ำ วิธีนี้เรียกว่า การตอบรับแบบบวกพร้อมกับการส่งซ้ำ (positive acknowledge with retransmission)
- ทีซีพีฝ่ายรับสามารถกำหนดให้ฝ่ายส่งจัดส่งข้อมูลเป็นปริมาณเท่าที่จะรับได้จริงตามขนาดบัฟเฟอร์ที่มีอยู่ การควบคุมปริมาณการไหลข้อมูลเป็นการป้องกันไม่ให้ส่งเซกเมนต์ไปเกินขีดความสามารถของฝ่ายรับ
- หากเซกเมนต์มาไม่เป็นลำดับ ทีซีพีฝ่ายรับต้องจัดลำดับเซกเมนต์ให้ถูกต้อง
- ทีซีพีมีหน้าที่กำจัดเซกเมนต์ซ้ำซ้อน

2.4.2 ทีซีพีเฮดเดอร์

ทีซีพีเฮดเดอร์ประกอบด้วยฟิลด์จำนวนมากทำหน้าที่ให้บริการตามฟังก์ชันที่กล่าวข้างต้น แต่ละฟิลด์มีความหมายดังนี้

0	15	16	31
Source port		Destination port	
Sequence number			
Acknowledgment number			
offset	Reserved	code	Window size
Checksum			Urgent pointer
Option + pad			
data			

รูปที่ 2.7 ทีซีพีเฮดเดอร์

- Source port ขนาด 16 บิต : หมายเลขพอร์ตของสถานีต้นทาง
- Destination port ขนาด 16 บิต : หมายเลขพอร์ตของสถานีปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Sequence number ขนาด 32 บิต : ทีซีพีใช้ เลขลำดับ เป็นตัวนับจำนวนไบต์ที่ส่ง ทุกครั้งที่สถาปนาการเชื่อมโยงทีซีพีจะเลือกเลขลำดับเริ่มต้นสำหรับชี้ตำแหน่ง ข้อมูลไบต์แรกที่จะจัดส่ง หมายเลขเริ่มต้นไม่จำเป็นต้องเริ่มด้วย 1 แต่อาจจะเริ่ม ด้วยค่าใดๆ ก็ได้ ข้อมูลในเซกเมนต์ถัดไปจะมีเลขลำดับที่สัมพันธ์เลขลำดับใน เซกเมนต์ก่อนหน้า
- Acknowledgement number ขนาด 32 บิต : ค่ากำหนด เลขตอบรับ ซึ่งใช้ตอบ กลับไปว่าได้รับข้อมูลแล้ว เลขตอบรับจะมีค่าเท่ากับเลขลำดับประจำเซกเมนต์ บวกด้วยจำนวนไบต์ข้อมูลและบวกด้วยหนึ่ง เช่นเซกเมนต์หนึ่งมีเลขลำดับเท่ากับ 21 และมีข้อมูล 20 ไบต์ เลขตอบรับที่ต้องส่งกลับ ไปจะเท่ากับ $21+20+1=42$ ซึ่ง แจ้งว่าได้รับข้อมูลตั้งต้นถึง ไบต์ลำดับที่ 41 แล้ว และคาดว่าไบต์ถัดไปคือไบต์ที่ 42
- Offset (data offset) ขนาด 4 บิต : บอกถึงตำแหน่งเริ่มต้นของไบต์ข้อมูลหรืออีก นัยหนึ่งใช้บอกขนาดเฮดเดอร์ ตัวเลขนี้มีหน่วยเป็นจำนวนเท่าของ 4 ไบต์ เช่นเดียวกับที่ใช้ในเคทาแกรม เฮดเดอร์ของทีซีพีมีความยาวขึ้นกับฟิลด์ option ตัวเลขในฟิลด์ offset จะเท่ากับ 5 ซึ่งเท่ากับ 20 ไบต์ ($5 \times 4 = 20$) หากไม่ใช่ออฟชัน ใด
- Reserved (RSV) ขนาด 4 บิต : สำรองไว้ใช้ในอนาคต
- Code ประกอบด้วย 6 ฟิลด์ย่อย แต่ละฟิลด์ย่อยมีขนาด 1 บิต ทำหน้าที่เป็นแฟล็ก เြียงลำดับจากซ้ายไปขวา
 - URGent ถ้าบิตนี้เป็น "1" หมายความว่า Urgent pointer บรรจุตำแหน่ง ข้อมูลที่ต้องรีบดำเนินการเร่งด่วนก่อน
 - ACKnowledgement ถ้าบิตนี้เป็น 1 หมายถึงเป็นเซกเมนต์ตอบรับ โดย ตอบอ้างอิงเลขลำดับตามที่กำหนดในฟิลด์ acknowledgement number
 - PuSH ถ้าบิตนี้เป็น "1" หมายความว่าทันทีที่สถานีปลายทางได้รับ เซกเมนต์ต้องรีบส่งข้อมูลไปยัง โปรโตคอลประยุกต์ทันทีโดยไม่ต้องรอ ให้บัฟเฟอร์เต็ม
 - ReSeT ถ้าบิตนี้เป็น "1" หมายถึงให้ยกเลิกการเชื่อมต่อนี้ เนื่องจากอาจมี ความผิดปกติเกิดขึ้นระหว่างคู่สถานีที่กำลังติดต่อกันอยู่ หากจำเป็นต้อง ส่งข้อมูลระหว่างกันอีกก็ต้องเริ่มต้นสถาปนาการเชื่อมต่อใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SYNchronize ถ้าบิตนี้เป็น “1” หมายถึงของเริ่มต้นสถาปนาการเชื่อมต่อ และเมื่อการสถาปนาเสร็จสิ้น บิตนี้จะถูกกำหนดให้เป็น “0” หลังจากนั้น จึงสามารถส่งผ่านข้อมูลระหว่างกันได้
- FINish ถ้าบิตนี้เป็น “1” หมายถึงจบการเชื่อมต่อ
- Window size ขนาด 16 บิต : สถานีปลายทางใช้ฟิลด์นี้แจ้งขนาดบัฟเฟอร์ที่มีอยู่ (เป็นหน่วยไบต์) สถานีที่ติดต่อดำเนินการไม่ส่งข้อมูลเกินค่านี้
- Checksum ขนาด 16 บิต : ผลรวมตรวจสอบความถูกต้องของเซกเมนต์โดย คำนวณทั้งเฮดเดอร์และข้อมูล (ใช้เฮดเดอร์เทียบเช่นเดียวกับยูติพี)
- Urgent pointer ขนาด 16 บิต : พอยเตอร์ชี้ตำแหน่งไบต์ข้อมูลที่ต้องดำเนินการเร่งด่วนที่ต้องการให้โปรแกรมประยุกต์ดำเนินการทันที ค่าที่บรรจุในฟิลด์นี้จะมี ความหมายก็ต่อเมื่อเฟล็ก URG ถูกเซตเป็น “1”
- Options ขนาดแปรเปลี่ยนได้ : ใช้กำหนดงานเพิ่มเติมให้กับทีซีพีซึ่งจะมีหรือไม่ก็ได้หากฟิลด์ offset หากมีค่าเป็น 5 แสดงว่ามีเฮดเดอร์มีขนาด 20 ไบต์ซึ่งหมายถึง ไม่ใช่ออปชัน
- Pad ขนาด 0 ถึง 24 บิต : ใช้เป็นส่วนที่ทำให้ขนาดของออปชันเป็นจำนวนเท่าของ 32 บิต (เพื่อให้เฮดเดอร์ลงตัวที่ค่าจำนวนเท่าของ 32)

2.4.3 กลไกการทำงานของทีซีพี

ทีซีพีมีกลไกการลำเลียงข้อมูลระหว่างต้นทางและปลายทางหลายส่วน ในที่นี้จะกล่าว เฉพาะกลไกที่สำคัญ 3 ประการก่อนได้แก่ การสถาปนาการเชื่อมโยง การถ่ายโอนข้อมูล และการยกเลิกการเชื่อมโยงตามหัวข้อต่อไปนี้

2.4.3.1 การสถาปนาการเชื่อมโยง

ขอยกตัวอย่างการสถาปนาการเชื่อมต่อระหว่างสถานีสองเครื่องคือ cc1 และ cc2 โดยให้ cc1 เป็นฝ่ายขอบริการจาก cc2 ดังนั้นโปรโตคอลประยุกต์ด้าน cc1 จะเป็นไคลเอ็นต์ และโปรโตคอลประยุกต์ด้าน cc2 จะเป็นเซิร์ฟเวอร์

Cc2 จะอยู่ในสภาพที่เรียกว่า การเปิดแบบพาสซีฟ (passive open) คือเซิร์ฟเวอร์ ส่งให้ทีซีพีรอรับการเชื่อมต่อ ส่วน cc1 จะเริ่มการติดต่อเมื่อไคลเอ็นต์สั่งงาน การเปิดการเชื่อมต่อ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโคลเ็นต์เรียกว่า การเปิดแบบแอคทีฟ (active open) กระบวนการนี้ประกอบด้วยการส่ง เซกเมนต์ 3 เซกเมนต์

1. ทีซีพีของ cc1 เลือกลำดับเริ่มต้น (ในที่นี้คือ 500) แล้วส่งเซกเมนต์ที่บรรจุเลขลำดับนี้ ไปยัง cc2 พร้อมทั้งเซตแฟล็ก SYN ให้เป็น “1” ขอให้สังเกตว่าเซกเมนต์นี้ (เซกเมนต์ 1) แฟล็ก ACK จะมีค่าเป็น “0”
2. ทีซีพีของ cc2 ได้รับเซกเมนต์จาก cc1 ก็จะเลือกเลขลำดับเริ่มต้นประจำตัวเช่นเดียวกัน (ในที่นี้คือ 700) แล้วตอบกลับด้วยเซกเมนต์ SYN (เซกเมนต์ 2) พร้อมทั้งเซตแฟล็ก ACK เพื่อแจ้งว่าได้รับเซกเมนต์ 1 โดยใช้เลขลำดับที่ได้รับจากทาง cc1 บวกด้วยหนึ่ง (ในที่นี้คือ 501)
3. ทีซีพีของ cc1 จะส่งเซกเมนต์ตอบรับกลับ ไป (เซกเมนต์ 3) โดยเซตแฟล็ก ACK และ ใช้เลขลำดับที่ได้รับจาก cc2 บวกด้วยหนึ่ง

ต่อจากนั้นทีซีพีของ cc1 จะแจ้งไปยังโปรโตคอลระดับบนว่าเชื่อมต่อแล้ว ส่วนทีซีพีของ cc2 เมื่อได้รับเซกเมนต์ตอบรับ (เซกเมนต์ 3) ก็จะแจ้งขึ้นไปยังโปรโตคอลระดับบนว่าเชื่อมต่อแล้วเช่นกัน เมื่อสิ้นสุดขั้นตอนนี้การสถาปนาทั้งสองด้านก็จะสมบูรณ์และพร้อมจะส่งข้อมูลได้ กระบวนการสถาปนาแบบนี้เรียกว่า “three way handshake” เพราะใช้ 3 เซกเมนต์ได้ตอบระหว่างกัน [Tom175] เซกเมนต์ทั้งสามนี้นิยมเรียกว่า SYN, SYN และ ACK ตามลำดับ

2.4.3.2 การถ่ายโอนข้อมูล

การขนถ่ายข้อมูลเริ่มได้หลังจากการสถาปนาเสร็จสิ้นแล้ว โดยสมมติให้ cc1 ส่งข้อมูลไปยัง cc2 ครั้งละ 100 ไบต์ จำนวน 4 ครั้ง

เซกเมนต์แรกที่ cc1 ส่งไปยัง cc2 บรรจุข้อมูลไบต์ 501 ถึง 600 และใช้เลขตอบรับ 701 เมื่อ cc2 รับข้อมูลเซกเมนต์ 1 แล้ว ก็จะตอบกลับด้วยเซกเมนต์ 2 พร้อมเลขตอบรับกำหนดข้อมูลที่คาดว่าจะได้รับต่อไปคือ 601

Cc1 สามารถส่งเซกเมนต์ไปอย่างต่อเนื่องได้ เช่นในแผนภาพมีการส่ง 3 เซกเมนต์คือไบต์ 601, 701 และ 801 ทาง cc2 สามารถตอบกลับในคราวเดียวกันได้ (เซกเมนต์ 6)

ในกรณีที่มีการส่งข้อมูลทั้งสองทิศทางระหว่าง cc1 และ cc2 หลักการทั่วไปยังคงเป็นเช่นเดียวกัน เพียงจะมีการตอบรับพร้อมกับส่งข้อมูลระหว่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.3 การยกเลิกการเชื่อมต่อ

เนื่องจากการถ่ายโอนในทีซีพีเป็นแบบฟลูคูเพิล็กซ์ การยกเลิกการเชื่อมต่อจึงมีขั้นตอนเกิดขึ้นทั้งสองด้าน ในที่นี้สมมติว่า cc1 ไม่มีข้อมูลส่งอีกต่อไปจึงต้องเป็นฝ่ายขอปิดการเชื่อมต่อ โปรโตคอลประยุกต์ของ cc1 จะแจ้งไปยังทีซีพีว่าส่งข้อมูลหมดแล้ว ถัดจากนั้นจะมีการแลกเปลี่ยนเซกเมนต์จำนวน 4 เซกเมนต์ดังนี้

1. ทีซีพีของ cc1 ส่งเซกเมนต์พร้อมที่มีเลขลำดับและเลขตอบรับตามปกติแต่เซตแฟล็ก FIN เป็น “1”
2. เมื่อทีซีพีของ cc2 ได้รับเซกเมนต์ FIN จะส่งเซกเมนต์ตอบรับ (เซกเมนต์ 2) และแจ้งไปยังโปรแกรประยุกต์ว่า cc1 ขอปิดการเชื่อมต่อ โปรแกรมประยุกต์ของ cc2 จะแจ้งกลับมายังทีซีพีว่าปิดการเชื่อมต่อได้
3. ทีซีพี cc2 ส่งเซกเมนต์ FIN ไปยัง cc1 (เซกเมนต์ 3)
4. เมื่อ cc1 ได้รับเซกเมนต์ FIN จะตอบรับกลับไป (เซกเมนต์ 4) และแจ้งไปยังโปรแกรมประยุกต์ว่าการเชื่อมโยงปิดลงแล้ว

2.5 เอชทีทีพี (HTTP)

เอชทีทีพีเป็น โปรโตคอลกำหนดการส่งข้อมูลระหว่างบราวเซอร์และเว็บเซิร์ฟเวอร์ในระบบเว็บ และยังใช้เป็น โปรโตคอลสำหรับแอปพลิเคชันแบบไคลเอ็นต์-เซิร์ฟเวอร์ใดๆก็ได้ชื่อเอชทีทีพีอาจสื่อความหมายถึง โปรโตคอลสำหรับการขนถ่ายข้อมูลไฮเปอร์เท็กซ์ แต่ความสามารถของเอชทีทีพีครอบคลุมถึงการขนถ่ายข้อมูลเท็กซ์ ไฮเปอร์เท็กซ์ ไบนารี เสียง ภาพ หรือข้อมูลแบบอื่นในอินเทอร์เน็ต โดยมีขีดความสามารถเฉพาะในการทำงานกับลิงค์แบบไฮเปอร์เท็กซ์ภายใต้ข้อกำหนดในรูปของ ภาษาเอชทีเอ็มแอล (HTML : Hypertext Markup Language)

เอชทีทีพีเป็น โปรโตคอลไคลเอ็นต์-เซิร์ฟเวอร์แบบทรานแซกชัน และอาศัยทีซีพีพอร์ต 80 เอชทีทีพีมีลักษณะการทำงานแบบ “ไร้สถานะ” (stateless) แต่ละทรานแซกชันจะเป็นอิสระต่อกัน เมื่อผู้ใช้เรียกบราวเซอร์เพื่ออ่านเว็บเพจ บราวเซอร์จะทำหน้าที่เป็น เอชทีทีพีไคลเอ็นต์ ติดต่อกับเอชทีทีพีเซิร์ฟเวอร์ (หรือเรียกว่า เว็บเซิร์ฟเวอร์)

เอชทีทีพีไคลเอ็นต์จะเปิดการเชื่อมต่อด้วยทีซีพีกับเอชทีทีพีเซิร์ฟเวอร์พร้อมทั้งส่งคำสั่งกำหนดชื่อเว็บเพจ เซิร์ฟเวอร์จะตอบกลับด้วยรหัสแสดงสถานะและเฮดเคอร์แบบไม่รวมทั้งข้อมูลเว็บเพจ เมื่อไคลเอ็นต์ได้รับคำตอบแล้วก็จะปิดการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 เอชทีทีพี 1.1

เอชทีทีพีที่ใช้แพร่หลายในปัจจุบันคือรุ่น 1.1 [RFC 2068] ซึ่งพัฒนาต่อจากรุ่น 1.0 [RFC 1945] เอชทีทีพี 1.1 มีข้อปรับปรุงเพิ่มเติมจาก 1.0 หลายประการเช่น

- ใช้การเชื่อมต่อแบบรักษาสภาพ เอชทีทีพี 1.0 จะสถาปนาและยกเลิกการเชื่อมต่อเป็นรายครั้งต่อการร้องขอครั้งหนึ่งๆ ซึ่งทำให้เกิดการสร้างและยกเลิกการเชื่อมโยงที่ซีพีจีจำนวนมาก เอชทีทีพี 1.1 ใช้วิธีคงสถานะการเชื่อมต่อไว้เพื่อให้ทำงานอย่างมีประสิทธิภาพมากขึ้น
- สนับสนุนการบีบอัดข้อมูลเพื่อนำส่งและการคลายข้อมูลเมื่อได้รับ
- มีกลไกควบคุมการแคชอย่างชัดเจนโดยเพิ่มเฮดเดอร์เพื่อกำหนดว่าไคลเอ็นต์ควรจะแคชข้อมูลที่ได้รับหรือไม่
- สนับสนุนระบบโฮสต์เสมือนโดยให้เว็บเซิร์ฟเวอร์เครื่องเดียวให้บริการในชื่อโดเมนที่ต่างกัน
- สนับสนุนการทำงานแบบหลายภาษา

2.6 ไมโครโปรเซสเซอร์

การทำงานของทั้งหมดของไมโครคอมพิวเตอร์จะถูกควบคุมด้วยไมโครโปรเซสเซอร์ (Microprocessor) โดยทำหน้าที่เป็นหน่วยประมวลผลกลาง (Central Processing Unit : CPU) การทำงานของไมโครโปรเซสเซอร์จะทำงานตามคำสั่งที่เราป้อนให้กับมัน ในปัจจุบันไมโครโปรเซสเซอร์ที่ใช้กันอยู่มีหลายรุ่น ถ้าหากเป็นคอมพิวเตอร์ IBM PC ที่ใช้กันอยู่ทุกวันนี้จะเป็นไมโครโปรเซสเซอร์ของบริษัทอินเทล หรือที่เรียกกันว่าไมโครโปรเซสเซอร์ตระกูลอินเทล ซึ่งมีอยู่หลายรุ่น ในการพัฒนาไมโครโปรเซสเซอร์รุ่นใหม่ๆ จะพัฒนาให้ทำงานได้เร็วขึ้นมีชุดคำสั่งมากขึ้นมีการพัฒนาริเจิสเตอร์ให้มีประสิทธิภาพดีขึ้นและอ้างหน่วยความจำได้มากขึ้นเครื่องคอมพิวเตอร์ IBM PC มีการพัฒนาออกมาหลายรุ่น เครื่องรุ่นใหม่ๆ จะใช้ ไมโครโปรเซสเซอร์ที่ทำงานได้เร็วขึ้น ไมโครโปรเซสเซอร์ที่นิยมใช้กันในคอมพิวเตอร์ IBM PC รุ่นแรกๆ ถึงรุ่น ปัจจุบันมีดังนี้

2.6 ไมโครโปรเซสเซอร์ 8088/8086

คอมพิวเตอร์ IBM PC รุ่นแรกจะเป็นคอมพิวเตอร์แบบ 16 บิต โดยใช้ CPU ที่เป็นไมโครโปรเซสเซอร์เบอร์ 8088 เรียกคอมพิวเตอร์รุ่นนี้ว่ารุ่น XT หรือ PC/XT (XT ย่อมาจาก eXtended) ส่วนประกอบภายในมีรีจิสเตอร์ขนาด 16 บิต จำนวน 14 ตัว โดยการประมวลผลภายในจะกระทำกับข้อมูลขนาด 16 บิต หรือ 2 ไบต์ แต่การติดต่อกับอุปกรณ์ภายนอก CPU จะใช้บัสข้อมูลขนาด 8 บิต CPU รุ่นนี้จะมีสายแอดเดรสทั้งหมด 20 เส้น ทำให้อ้างตำแหน่งได้ 1 MB สำหรับ CPU รุ่น 8086 โครงสร้างภายในจะเหมือนกับรุ่น 8088 แต่จะติดต่อกับอุปกรณ์ภายนอกด้วยบัสข้อมูลแบบ 16 บิต จึงเรียกรุ่น 8086 ว่าเป็น 16 บิตที่แท้จริง

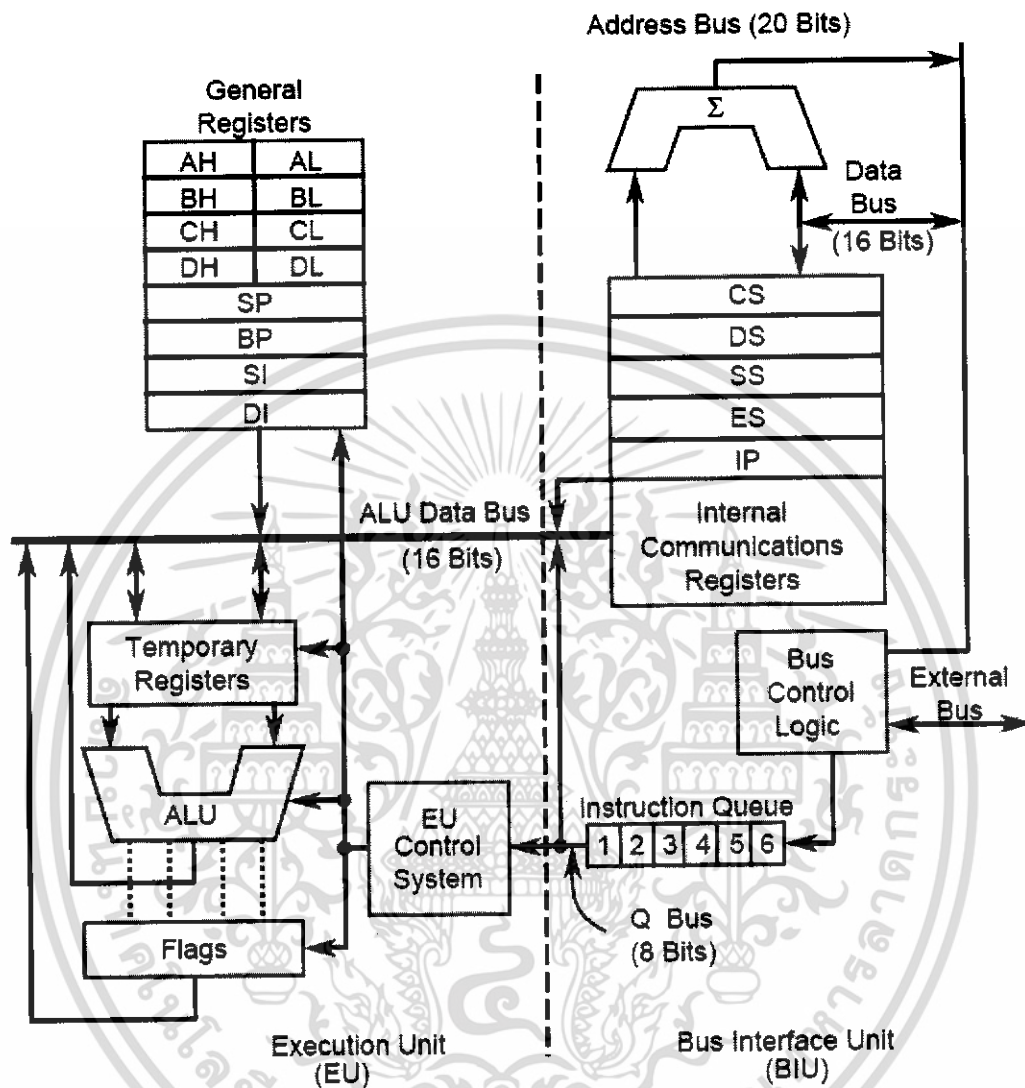


(a)

(b)

รูปที่ 2.8 แสดงภาพไมโครโปรเซสเซอร์ 8088 (a) และ 8086 (b) ตามลำดับ

2.7.2 โครงสร้างภายในของไมโครโปรเซสเซอร์ 80C188EB



รูปที่ 2.10 โครงสร้างภายในไมโครโปรเซสเซอร์ 80C188EB

ในไมโครโปรเซสเซอร์หรือตัวประมวลผลจะแบ่งการทำงานออกเป็นสองส่วนคือ Execute Unit (EU) และ Bus Interface Unit (BIU) ดังรูปที่ 2.14 การทำงานของ EU จะทำหน้าที่ประมวลผล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสคำสั่งและข้อมูลที่ได้รับมาจาก BIU การเปลี่ยนแปลงหรือการกระทำกับข้อมูลใดๆ จะต้องเกิดจากส่วน EU นี้ โครงสร้างภายใน EU จะประกอบด้วยรีจิสเตอร์ที่ใช้งานทั่วไปได้แก่ AX, BX, CX, DX สแต็กพอยน์เตอร์ (SP) เบสพอยน์เตอร์ (BP) ซอร์สอินเด็กซ์ (SI) เดสทินชันอินเด็กซ์ (DI) รีจิสเตอร์แฟลค หน่วยควบคุม รวมทั้งหน่วยประมวลผลทางคณิตศาสตร์และลอจิกที่เรียกว่า ALU ส่วน BIU จะทำหน้าที่เชื่อมต่อบัสในการส่งข้อมูลต่างๆ ควบคุมการเข้าถึงหน่วยความจำและการติดต่อกับอุปกรณ์ภายนอกต่างๆ ของไมโครโปรเซสเซอร์ ประกอบด้วย หน่วยควบคุมบัส (bus control) รีจิสเตอร์เซกเมนต์ (segment register) วงจรลอจิกที่ใช้อ้างตำแหน่งหน่วยความจำ (memory address logic) เป็นต้น

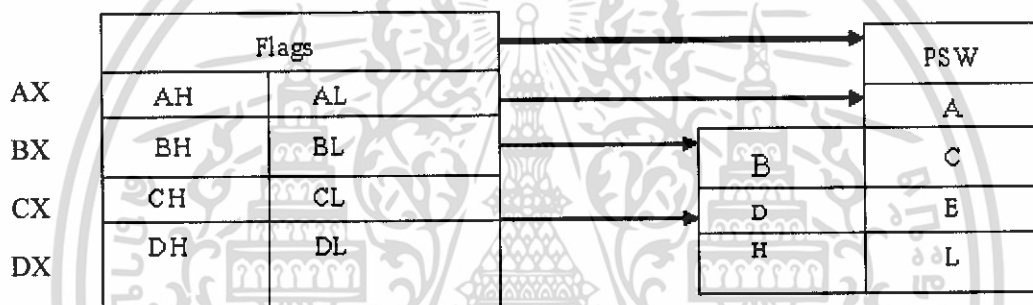
การให้คอมพิวเตอร์ทำงานนั้นก็คือการให้ไมโครโปรเซสเซอร์ทำงาน โดยจะเริ่มจากการอ่านข้อมูลจากหน่วยความจำโปรแกรมเข้ามาในตัวไมโครโปรเซสเซอร์จากนั้นจะถอดรหัสคำสั่งและทำงานตามคำสั่งเมื่อทำงานตามคำสั่งแล้วจะไปอ่านรหัสคำสั่งต่อไปออกมาเราเรียกกระบวนการนี้ว่า fetch and execute cycle และการทำงานคำสั่งหลายๆ คำสั่งต่อกันเราเรียกว่าการทำงานโปรแกรม เมื่อไมโครโปรเซสเซอร์เริ่มทำงานหลังจากการกดปุ่มรีเซ็ตของคอมพิวเตอร์ กระบวนการต่างๆ จะเป็นดังต่อไปนี้

1. ค่าในรีจิสเตอร์ IP (instruction pointer) จะถูกคำนวณเป็นค่าตำแหน่งหน่วยความจำโปรแกรมและจะส่งออกมาทางเอาต์พุตของ BIU เพื่ออ้างไปที่ตำแหน่งหน่วยความจำโปรแกรมที่ไมโครโปรเซสเซอร์ต้องการติดต่อ จากนั้นจะอ่านรหัสโปรแกรมจากหน่วยความจำเข้ามายัง BIU เพื่อให้หน่วย EU ทำการประมวลผล
2. ค่ารีจิสเตอร์ IP จะเพิ่มค่าขึ้นอีกเพื่ออ่านข้อมูลรหัสคำสั่งต่อไป
3. ไมโครโปรเซสเซอร์จะทำงานตามรหัสคำสั่งที่อ่านเข้ามาและอ่านค่าคำสั่งต่อไป

2.7.3 รีจิสเตอร์และเซกเมนต์

รีจิสเตอร์ (Register) เป็นหน่วยความจำขนาดเล็กที่อยู่ใน CPU ที่สามารถทำการคำนวณ เคลื่อนย้ายตำแหน่งต่างๆได้

รีจิสเตอร์ คือ หน่วยความจำชนิดซึ่งแตกต่างจาก ROM (Random Only Memory : เป็นหน่วยความจำที่ อยู่ติดกับแผ่นวงจรหลักของ เครื่องไม่สามารถเพิ่มขยายขนาดได้) และ RAM (Random Access Memory: เป็นหน่วย ความจำที่สามารถเพิ่มขยายขนาดได้) ซึ่งเป็นเพียงที่พัก ข้อมูลหรือคอยเก็บข้อมูลไว้เท่านั้นเอง แต่รีจิสเตอร์นั้น สามารถใช้ในการคำนวณนั้นไปเก็บไว้ เป็นหน่วยความจำที่อยู่ในไมโคร โพรเซสเซอร์ 80C188EB จะมีขนาด 16 บิต ดังนั้น รีจิสเตอร์จะมีขนาด 16 บิตด้วย

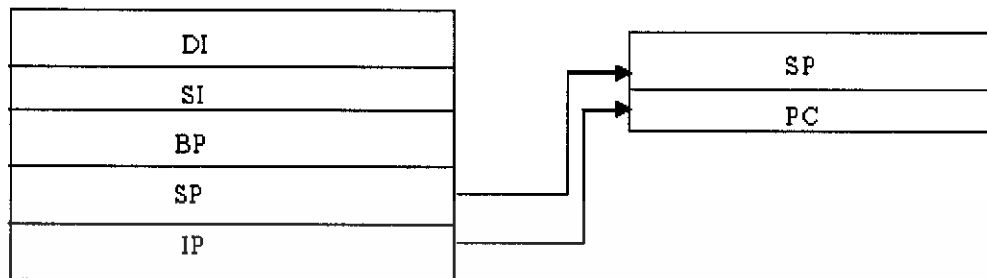


รูปที่ 2.11 โครงสร้างรีจิสเตอร์แบบ 16 บิต และแบบ 8 บิต

แอกคิวมูลเตอร์ (Accumulator) หรือ Primary Accumulator	AX	ทั้งหมดนี้เรียกว่ารี- จิสเตอร์ใช้งานทั่วไป
เบสพอยน์เตอร์ (Base Pointer) หรือ Accumulator and Base Register	BX	
เคานต์รีจิสเตอร์ (Count Register) หรือ Accumulator and Counter	CX	
รีจิสเตอร์ข้อมูล (Data Register) หรือ Accumulator and I/O Address	DX	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจิสเตอร์ตัวชี้และอินเด็กซ์



รูปที่ 2.12 อินเด็กซ์รีจิสเตอร์ และ พ้อยเตอร์ รีจิสเตอร์ แบบ 16 บิต และแบบ 8 บิต

DI (Destination Index): รีจิสเตอร์ปลายทาง

SI (Source Index): รีจิสเตอร์ต้นทาง

BP (Stack Base Pointer): เบสพ้อยเตอร์

SP (Stack Pointer): สแต็กพ้อยเตอร์

IP (Instruction Pointer): อินสตรักชันพ้อยเตอร์

- รีจิสเตอร์คำสั่ง Instruction Pointer (IP) จะทำงานร่วมกับรีจิสเตอร์ CS โดยจะใช้เก็บตำแหน่งคำสั่งถัดไปที่จะให้ไมโครโปรเซสเซอร์ทำงาน

- รีจิสเตอร์แฟลก (Flags Register) จะใช้เก็บสถานะต่างๆ จากการประมวลผลของไมโครโปรเซสเซอร์ เมื่อไมโครโปรเซสเซอร์ทำงานคำสั่งบางคำสั่ง ผลลัพธ์ที่ได้จะมีผลต่อแฟลกเพื่อบอกสถานะของผลลัพธ์ที่ได้ รีจิสเตอร์แฟลกแบ่งออกเป็นสองส่วนคือ แฟลกสถานะ (status flags) และแฟลกควบคุม (control flags)

- เซกเมนต์ (Segment) คือ Block ของหน่วยความจำที่เริ่มต้น ณ ตำแหน่งที่คงที่ สามารถหาได้จากค่าของ เซกเมนต์รีจิสเตอร์ (Segment Register) 80C188EB มี เซกเมนต์รีจิสเตอร์ อยู่ 4 ตัว ซึ่งสามารถชี้ไป ยังตำแหน่งใดๆ ในแอดเดรส 1 MB ก็ได้

CS
DS
SS
ES

รูปที่ 2.13 เซกเมนต์รีจิสเตอร์

CS (Code Segment) : โค้ดเซกเมนต์

DS (Data Segment) : คาดต้าเซกเมนต์

SS (Stack Segment) : สแต็คเซกเมนต์

ES (Extra Segment) : เอ็กตราเซกเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

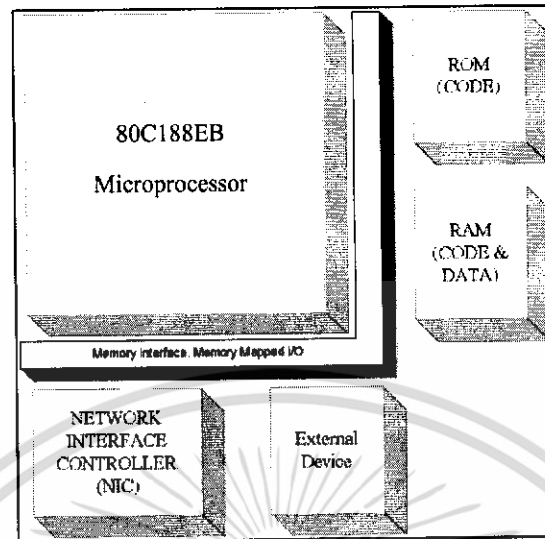
วิธีดำเนินการวิจัย

3.1 การออกแบบฮาร์ดแวร์

ฮาร์ดแวร์ของระบบควบคุมแบบฝังตัวทั่วไปนั้นประกอบด้วย ไมโครโปรเซสเซอร์(CPU) หน่วยความจำ (Memory) และส่วนรับส่งข้อมูลเข้า-ออก (Input/Output) นอกจากนี้ยังประกอบด้วย ช่องสัญญาณสำหรับการสื่อสาร (BUS) เพื่อใช้เชื่อมต่อกับอุปกรณ์ควบคุม หรืออุปกรณ์ที่ใช้ในการวัดค่าต่างๆ เช่น เซ็นเซอร์วัดความชื้น เซ็นเซอร์วัดอุณหภูมิ เป็นต้น ดังนั้นรายละเอียดการออกแบบ ส่วนของฮาร์ดแวร์ในที่นี่จะประกอบไปด้วย ส่วนประกอบหลักของไมโครโปรเซสเซอร์ 80C188EB ของอินเทล การเชื่อมต่อไมโครโปรเซสเซอร์เข้ากับเครือข่ายเน็ตเวิร์คโดยอาศัยอุปกรณ์ควบคุมระบบเครือข่ายหรือ NIC (Network Interface Controller) ของ Realtek เบอร์ RTL8019AS รวมถึงการเชื่อมต่อเข้ากับอุปกรณ์วัดค่าต่างๆ เช่น เซ็นเซอร์วัดอุณหภูมิ DS18S20 ของ Dallas เป็นต้น

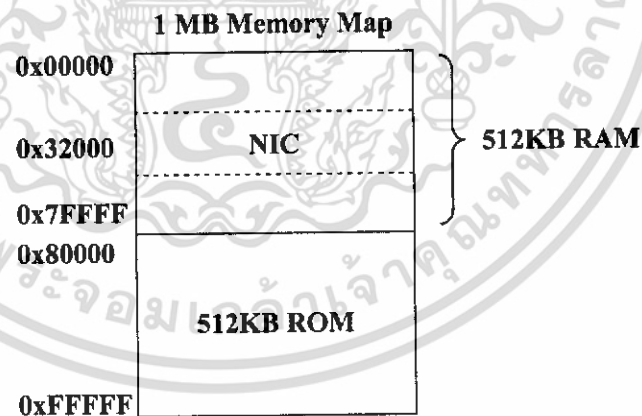
3.1.1 ส่วนประกอบทางฮาร์ดแวร์ของเว็บเซิร์ฟเวอร์แบบฝังตัว

ระบบควบคุมแบบฝังตัวมีลักษณะทั่วไปคล้ายคอมพิวเตอร์ขนาดเล็ก กล่าวคือ ประกอบด้วย ไมโครโปรเซสเซอร์ หน่วยความจำโปรแกรมหรือรอม (ROM: Read Only Memory) หน่วยความจำข้อมูลหรือแรม (RAM: Random Access Memory) และระบบอินพุต/เอาต์พุต เพื่อใช้ในการเชื่อมต่ออุปกรณ์ภายนอกอื่นๆ แต่ลักษณะของเว็บเซิร์ฟเวอร์แบบฝังตัวนั้น จำเป็นต้องมีการติดต่อสื่อสารผ่านระบบเครือข่ายด้วย ในที่นี้คือระบบอีเทอร์เน็ต ซึ่งการเชื่อมต่อกับระบบเครือข่ายอีเทอร์เน็ตนั้นจะอาศัยอุปกรณ์ควบคุมการเชื่อมต่อระบบเครือข่ายหรือ NIC (Network Interface Controller) ดังนั้นส่วนประกอบโดยรวมของเว็บเซิร์ฟเวอร์แบบฝังตัวแสดงดังภาพที่ 3.1



รูปที่ 3.1 แสดงส่วนประกอบทางฮาร์ดแวร์ของเว็บเซิร์ฟเวอร์แบบฝังตัว

จากภาพตำแหน่งของแรมอยู่ที่ตำแหน่ง 0x00000 ถึง 0x7FFFF และตำแหน่งของรอมอยู่ที่ตำแหน่ง 0x80000 ถึง 0xFFFFF ซึ่งทั้งรอมและแรมมีขนาดเท่ากันคือ 512 KB และทำการ Memory Map ตัว RTL8019AS ไว้ที่ตำแหน่ง 0x3200 ดังรูปที่ 3.2



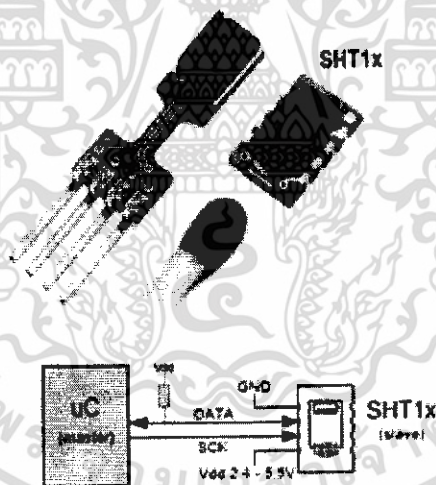
รูปที่ 3.2 แสดงการจัดการหน่วยความจำรอมและแรม พร้อมตำแหน่งของ RTL8019AS ใน Memory Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การเชื่อมต่อไอซีตรวจอุณหภูมิ SHT 15 เข้ากับ 80C188EB

เซนเซอร์รุ่น SHT 1X ของ SENSIRION เป็นเซนเซอร์ขนาดเล็กจิ๋วเมื่อเทียบกับขนาดหัวไม้จิกไฟและเป็นเซนเซอร์ตรวจจับความชื้นสัมพัทธ์แบบดิจิทัลที่วัดค่าอุณหภูมิได้ด้วย โดยมีจุดเด่นต่างๆ ดังนี้

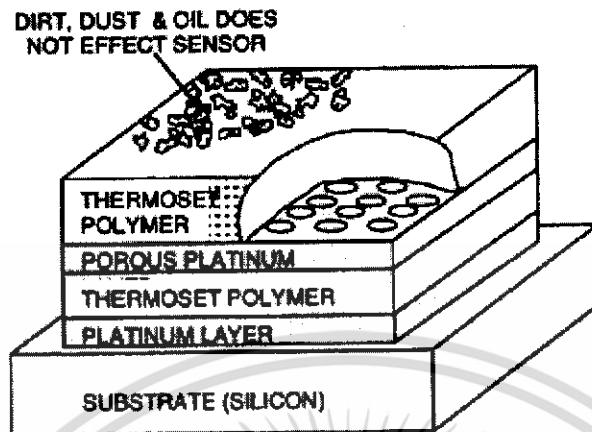
- เอาท์พุทแบบดิจิทัล
- เสถียรภาพสูงมาก
- ไม่ต้องมีวงจรใดมาเชื่อมต่อเพิ่มเติมอีก
- ใช้พลังงานไฟฟ้าต่ำมาก
- มีตัวถังให้เลือกแบบ Surface Mount หรือแบบ 4 ขาสัญญาณ
- ขนาดเล็ก
- มีระบบปิดเครื่องอัตโนมัติ
- สามารถเชื่อมต่อกับไมโครคอนโทรเลอร์ได้



แสดงหน้าตาของเซนเซอร์ SH11

โครงสร้างของเซนเซอร์วัดความชื้นและการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โครงสร้างมี 3 ชั้น วางอยู่บน SUBSTRATE (SILICON) ประกอบไปด้วยชั้นที่เป็น PLATINUM 2 ชั้นประกบชั้น ที่เป็น POLYMER ทำให้มีลักษณะเป็นตัว capacitance ตัวหนึ่ง ซึ่งชั้นที่เป็น POLYMER นี้เป็นวัสดุที่สามารถดูดซับความชื้นและคายความชื้นได้ อย่างที่เราทราบกันดีว่า ในอากาศประกอบไปด้วยน้ำเมื่อปริมาณน้ำในอากาศมากขึ้น วัสดุในชั้นนี้ก็จะดูดซับน้ำในอากาศมาทำให้ค่าความจุไฟฟ้าเปลี่ยนไปเราจึงนำเอาค่าความจุไฟฟ้าที่เปลี่ยนไปนั้นมาคำนวณหาค่าความชื้นและค่าอุณหภูมิได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและอภิปรายผล

4.1 ผลการทดลอง

การทดลองจะทำโดยการเชื่อมต่อ Embedded Web Server เข้ากับระบบเครือข่ายแลน ซึ่งในการทดลองนี้ได้ตั้งหมายเลข IP address ของ Embedded Web Server ของเราไว้ที่หมายเลข 161.246.13.69 และจะทำการทดลองโดยใช้โปรแกรม ping ของวินโดวส์เพื่อทำการทดสอบการเชื่อมต่อระหว่าง Embedded Web Server กับระบบแลนภายใต้โปรโตคอล ICMP หลังจากนั้นจะใช้โปรแกรม SNMPView ช่วยในการทดสอบโปรโตคอล SNMP ที่ใช้ในการดูค่าของอุณหภูมิ, ความชื้น, Dewpoint และ เวลาของระบบ Embedded Web Server ที่เปิดใช้งาน ซึ่งผลการทดลองเป็นดังนี้

4.1.1 ผลการทดลองโดยใช้โปรแกรม Ping ของวินโดวส์



รูปที่ 4.1 แสดงคำสั่งที่ใช้ในการ ping

คำสั่งที่ใช้ในการทดสอบโปรโตคอล ICMP โดยใช้โปรแกรม ping ของวินโดวส์ทำโดยการพิมพ์คำสั่ง ping และตามด้วยหมายเลข IP address ปลายทางหลังจากนั้นจึงกดปุ่ม enter ที่ keyboard

เพื่อแสดงการทำงานของโปรโตคอล ICMP โดยใช้โปรแกรม ping ของวินโดวส์ ผลการทดลองได้ดังนี้

```

C:\WINDOWS\system32\cmd.exe
C:\>ping 161.246.13.69

Pinging 161.246.13.69 with 32 bytes of data:

Reply from 161.246.13.69: bytes=32 time=17ms TTL=32
Reply from 161.246.13.69: bytes=32 time=8ms TTL=32
Reply from 161.246.13.69: bytes=32 time=17ms TTL=32
Reply from 161.246.13.69: bytes=32 time=16ms TTL=32

Ping statistics for 161.246.13.69:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli seconds:
        Minimum = 8ms, Maximum = 17ms, Average = 14ms

C:\>

```

รูปแสดงผลการ ping จากวินโดวส์

ผลการทดลองนี้แสดงการใช้โปรแกรม ping ส่ง ICMP ไปยัง 161.246.13.69 ซึ่งเป็นหมายเลข IP address ประจำของ Embed Web Server ของเรา หลังจากนั้นระบบของเราก็จะตอบกลับมาโดยแสดงรายละเอียดต่างๆ ดังนี้

- Pinging 161.246.13.69 with 32 bytes of data:

หมายถึง โปรแกรม ping ของวินโดวส์ได้ส่งข้อมูลขนาด 32 ไบต์ ไปที่ 161.246.13.69

- Reply from 161.246.13.69:

หมายถึง ระบบของเราได้ทำการตอบกลับจากการร้องขอของเครื่องต้นทาง

- bytes = 32

หมายถึง ขนาดของข้อมูลตอบกลับซึ่งเป็นข้อมูลเดียวกันกับเครื่องต้นทางที่ส่งเข้ามา

- time = xx ms

หมายถึง เวลาที่ใช้ในการตอบกลับไปยังเครื่องต้นทาง xx หมายถึง เวลาที่ใช้ ในหน่วย ms (หนึ่งในร้อยวินาที)

- TTL = 32

ค่า TTL ใช้กำหนดหาจำนวนเราเตอร์ที่ข้อมูลเดินทางผ่าน ค่า ping กำหนด TTL เริ่มต้นเท่ากับ 255 และลดค่าลงครั้งละหนึ่งทุกครั้งที่ผ่านเราเตอร์ ตัวเลข 32 หมายความว่าข้อมูลเดินทางผ่านเราเตอร์เท่ากับ $255-32 = 213$ ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Packets : Sent = 4, Received = 4, Lost = 0 <0% loss>

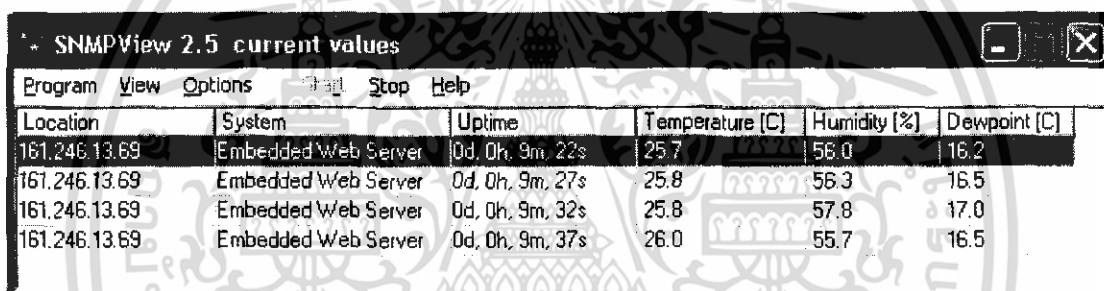
หมายถึง ข้อมูลที่ส่งไป 4 ข้อมูล ได้รับ 4 ข้อมูล สูญหาย 0 ข้อมูล สรุป ข้อมูลที่สูญหายเท่ากับ 0%

- Minimum = 8ms, Maximum = 17ms, Average = 14ms

หมายถึง เวลาที่ใช้ในการตอบกลับน้อยสุดเท่ากับ 8ms เวลาที่ใช้ในการตอบกลับมากที่สุดเท่ากับ 17ms เวลาที่ใช้ในการตอบกลับเฉลี่ยเท่ากับ 14ms

4.1.2 ผลการทดลองโดยใช้โปรแกรม SNMPView 2.5

โปรแกรม SNMPView มีความสามารถในการดูข้อมูลผ่านระบบเน็ตเวิร์คภายใต้โปรโตคอล SNMP ซึ่งเราได้ทำการตั้งค่าเริ่มต้นให้กับโปรแกรมเพื่อใช้ในการดูค่าต่างๆ ดังนี้



Location	System	Uptime	Temperature [C]	Humidity [%]	Dewpoint [C]
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 22s	25.7	56.0	16.2
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 27s	25.8	56.3	16.5
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 32s	25.8	57.8	17.0
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 37s	26.0	55.7	16.5

รูป 4.3 แสดงผลการอ่านค่าข้อมูลจากการเรียกดูจาก โปรแกรม SNMPView ผ่านระบบเน็ตเวิร์ค

- Location

หมายถึง IP address ของปลายทางที่ต้องการซึ่งจากการใช้โปรแกรม SNMPView นี้เราจะเห็นได้ว่าเป็น IP address ของที่เราได้ทำการตั้งค่าไว้เป็น 161.246.13.69

- System

หมายถึง ชื่อของระบบที่ได้ตอบกลับมาซึ่งเป็นชื่อที่เราตั้งไว้ในอุปกรณ์

- Uptime

หมายถึง เวลาของระบบตั้งแต่เริ่มต้นการใช้งานจนถึงเวลาที่แสดงค่า

- Temperature (C)

หมายถึง ค่าของอุณหภูมิที่เราได้ทำการอ่านค่าได้จาก Embed Web Server มีหน่วยเป็น องศาเซลเซียส

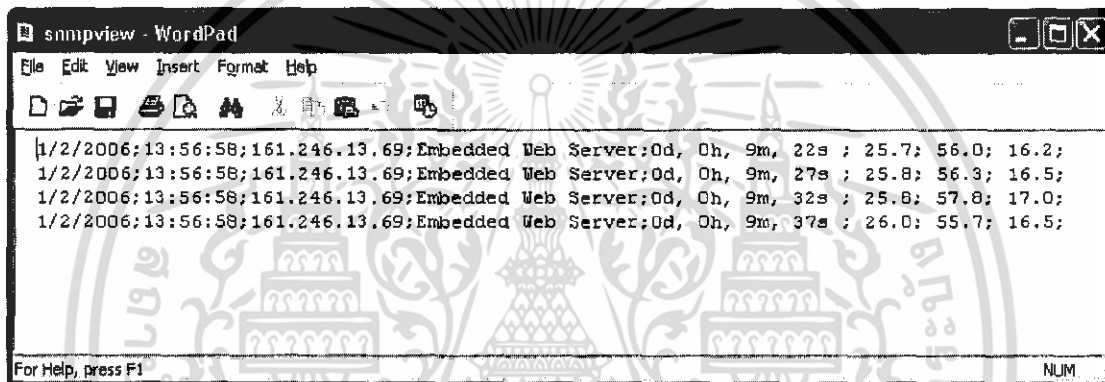
- Humidity (%)

หมายถึง ค่าปริมาณความชื้นที่อุปการณในการตรวจวัดได้ทำการจับได้ เมื่อทำการเปรียบเทียบกับจุด Dewpoint แล้วจะแสดงค่าเป็นเปอร์เซ็นต์

- Dewpoint (C)

หมายถึง ค่าที่ได้จากการคำนวณระหว่างค่าอุณหภูมิกับค่าปริมาณความชื้นที่วัดได้ มีหน่วยเป็นองศาเซลเซียส

โปรแกรม SNMPView สามารถบันทึกผลซึ่งผลนี้เราสามารถนำไปใช้ประโยชน์ได้



รูป 4.4 แสดงข้อมูลทั้งหมดที่โปรแกรม SNMPView ทำการเก็บไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถอีกอย่างหนึ่งของโปรแกรม SNMPView คือ สามารถแสดงผลในรูปแบบของเว็บเบราว์เซอร์ได้

Output from SNMPView

[show Eventlog](#)
[show as jpg](#)

Location	System	Uptime	Temperature [C]	Humidity [%]	Dewpoint [C]
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 22s	25.7	56.0	16.2
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 27s	25.8	56.3	16.5
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 32s	25.8	57.8	17.0
161.246.13.69	Embedded Web Server	0d, 0h, 9m, 37s	26.0	55.7	16.5

generated by [SNMPView 2.5](#) 1.2.2006. 13:56:58

รูป 4.5 แสดงข้อมูลผ่านเบราว์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและข้อเสนอแนะ

การพัฒนาเว็บเซิร์ฟเวอร์แบบฝังตัวนั้นสามารถนำไปประยุกต์ใช้ในงานต่างๆ ได้อย่างมาก เนื่องจากเทคโนโลยีในปัจจุบันทำให้ระบบอินเทอร์เน็ตได้รับความนิยมและแพร่หลายในทุกๆ พื้นที่ ทำให้การเข้าถึงอินเทอร์เน็ตเป็นเรื่องที่ง่ายและสะดวก ดังนั้นการพัฒนาเว็บเซิร์ฟเวอร์แบบฝังตัวนี้จึงมีความเหมาะสมในเรื่องของการใช้งานที่ง่าย สะดวกในการติดตั้งพร้อมทั้งมีราคาที่ถูกสามารถนำไปประยุกต์ใช้งานต่างๆ ได้มากมายในทุกๆ พื้นที่ที่อินเทอร์เน็ตเข้าถึง อย่างไรก็ตามโครงการนี้ยังคงเป็นเพียงต้นแบบในการศึกษา ดังนั้นยังคงมีข้อบกพร่องอยู่หลายประการ และยังคงต้องการการพัฒนาให้มีความสามารถมากขึ้นต่อไป

5.1 สรุป

จากการศึกษาโครงการเว็บเซิร์ฟเวอร์แบบฝังตัวทำให้เราได้ทราบถึงหลักการการทำงานของระบบเครือข่ายอินเทอร์เน็ต สามารถเขียนโปรแกรมควบคุมการทำงานของไมโครโปรเซสเซอร์ขนาด 16 บิต (80C188EB) เพื่อทำการเชื่อมต่อกับอีเทอร์เน็ตชิพ (RTL8019AS) กับไมโครโปรเซสเซอร์นั้นได้ ทำให้เราได้ทราบถึงหลักการการทำงานของโปรโตคอลต่างๆ ในชั้นของ TCP/IP Stack และนำความรู้นั้นไปประยุกต์ใช้ในการเขียนโปรแกรมเพื่อติดต่อกับระบบอินเทอร์เน็ตได้ เช่น การอ่านค่าจากเซ็นเซอร์ผ่านทางระบบอินเทอร์เน็ต ทำให้เราสามารถนำความรู้ที่ได้ไปพัฒนาอุปกรณ์ต่างในระบบเครือข่ายอินเทอร์เน็ตได้ต่อไป

5.2 ข้อเสนอแนะ

มาตรฐานของโปรโตคอล TCP/IP นั้นยังคงมีความสามารถอีกมาก จากโครงการเว็บเซิร์ฟเวอร์แบบฝังตัวนี้เรายังคงใช้ความสามารถเพียงส่วนหนึ่งของมาตรฐานโปรโตคอล TCP/IP เท่านั้น จึงทำให้สามารถพัฒนาให้เว็บเซิร์ฟเวอร์แบบฝังตัวนี้มีความสามารถเพิ่มมากขึ้นได้อีก เช่น การปรับแต่งค่าผ่านระบบเครือข่ายอินเทอร์เน็ต การควบคุมอุปกรณ์ไฟฟ้าผ่านระบบเครือข่ายอินเทอร์เน็ต เป็นต้น

ในการใช้งานเว็บเซิร์ฟเวอร์แบบฝั่งตัวนี้ยังมีข้อจำกัดในเรื่องของความเร็วในการทำงาน ซึ่งยังไม่สามารถทำงานในระบบ real time ได้ดีนัก จึงเหมาะที่จะใช้งานในด้านการตรวจสอบข้อมูลย้อนหลังมากกว่าจะเป็นการอ่านข้อมูลในแบบ real time ข้อจำกัดอีกประการคือโปรโตคอล TCP/IP Stack นั้นมีความต้องการใช้ทรัพยากรอย่างมากทั้งในเรื่องของหน่วยความจำ ความเร็วในการทำงานของซีพียู ดังนั้นควรทำการปรับปรุง โปรแกรมให้ทำงานให้ได้ประสิทธิภาพมากที่สุดในขณะที่ใช้หน่วยความจำน้อยที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

Jeremy Bentham. **TCP/IP Lean: Web Servers for Embedded Systems (2nd Edition)**

สุรศักดิ์ สงวนพงษ์. **สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2543.

ธีรวัฒน์ ประกอบผล. **การโปรแกรมภาษาซีสำหรับงานวิทยาศาสตร์**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2545.

ธีรวัฒน์ ประกอบผล. **ระบบคอมพิวเตอร์และภาษาแอสเซมบลี**. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2543.

รองศาสตราจารย์ มัณฑนา ปราการสมุทร. **การเขียนชุดคำสั่งภาษาซี**. ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2534

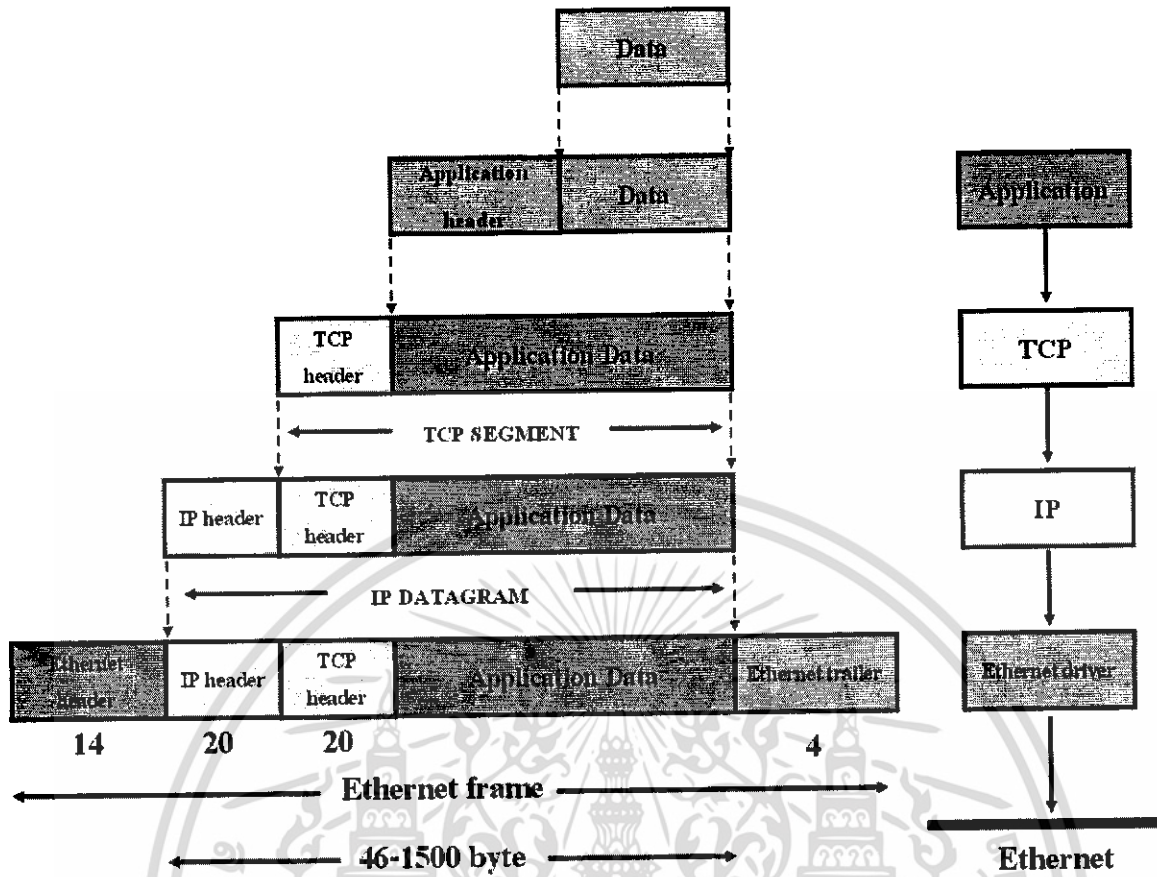


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

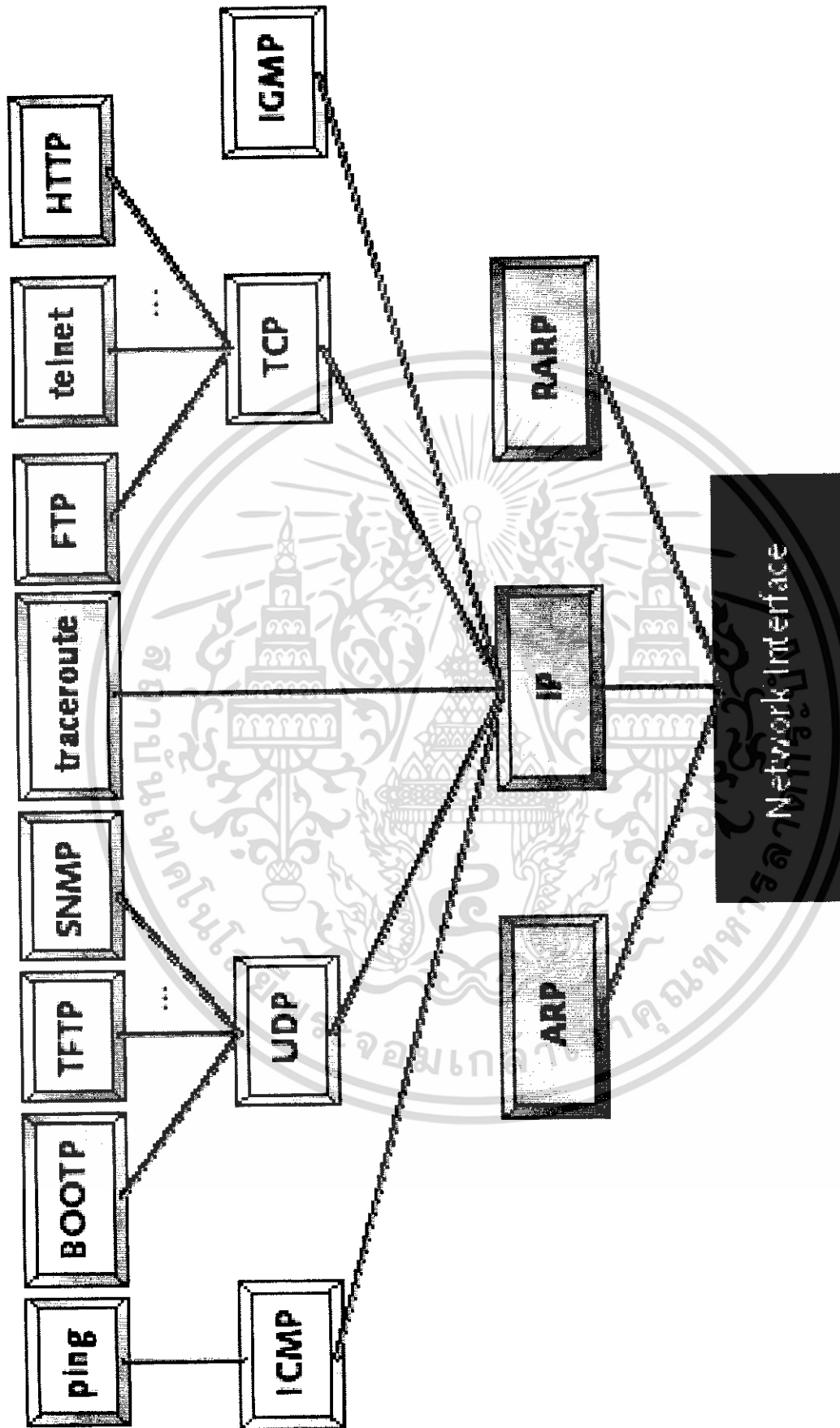


ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้