สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

EMBEDDED IPV6 DEVICE

นางสาวชิตสุธา สุ่มเล็ก

นายฐิติวุฒิ ศิลปพิพัฒน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่.....

เลขทะเบียน.....55085

วัน,เดือน,ปี.....- 8 มิ.ย. 2548

b.....

i.....

ไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

EMBEDDED IPV6 DEVICE

โดย

นางสาวชิตสุธา สุ่มเล็ก

นายฐิติวุฒิ ศิลปพิพัฒน์

อาจารย์ที่ปรึกษา

ผศ.อภิเนตร อูนากูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

EMBEDDED IPV6 DEVICE

คณะผู้จัดทำ นางสาวชิตสุธา สุ่มเล็ก รหัส 43010106

นายฐิติวุฒิ ศิลปพิพัฒน์ รหัส 43010115

..... อาจารย์ที่ปรึกษา

(ผศ.อภิเนตร อูนากูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

นางสาวชิตสุธา สุ่มเล็ก 43010106
นายฐิติวุฒิ ศิลปพิพัฒน์ 43010115
ผศ.อภิเนตร อูนากูล อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

เนื่องจากปัจจุบันเทคโนโลยีระบบฝังตัวมีการเติบโตอย่างรวดเร็ว และความต้องการในการใช้ระบบเครือข่ายนั้นสูงขึ้นเรื่อยๆ จนมีการนำเสนอเทคโนโลยีของไอพีรุ่น 6 เพื่อแก้ปัญหาการขาดแคลนไอพีแอดเดรสในอนาคตและเพื่อเพิ่มประสิทธิภาพของโปรโตคอลไอพี แต่การจะปรับเปลี่ยนไปใช้ไอพีรุ่น 6 นั้นยังต้องอาศัยการศึกษาและพัฒนาทั้งทางด้านฮาร์ดแวร์และซอฟแวร์อีกมาก รวมทั้งต้องอาศัยการลงทุนที่ค่อนข้างสูงด้วยเช่นกัน อีกทั้งการพัฒนาไอพีรุ่น 6 สำหรับอุปกรณ์ขนาดเล็กนั้นมีอยู่น้อยรวมทั้งการแจกจ่ายความรู้ด้วย ทำให้การพัฒนาอุปกรณ์ขนาดเล็กซึ่งมีอยู่เป็นจำนวนมากในปัจจุบันและมีแนวโน้มที่จะเพิ่มขึ้นมากในอนาคตนั้นกระทำได้โดยยาก

ปริญญานิพนธ์ฉบับนี้จึงได้นำเสนอแนวทางในการพัฒนาโปรโตคอลไอพีรุ่น 6 บนระบบฝังตัว โดยอยู่บนพื้นฐานของชุดพัฒนาคอม 86 กับการนำเอาเทคโนโลยีที่มีอยู่มาประยุกต์ใช้ให้เกิดประโยชน์ และการสร้างระบบฝังตัวไอพีรุ่น 6 ที่สามารถนำไปศึกษาและพัฒนาต่อได้ โดยเน้นไปที่อุปกรณ์ขนาดเล็ก รวมทั้งลดขั้นตอนในการพัฒนาและค่าใช้จ่ายในการลงทุนด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EMBEDDED IPV6 DEVICE

Miss Chitsutha Soomlek
Mr. Thitiwut Sillapapipat
Assist. Prof. Apinetr Unakul Advisor

ABSTRACT

Nowadays, embedded system technology is being grown sharply. And, the demand of using the network is increased continuously. Moreover, in order to solve the insufficient IP address problem and improve the performance of the Internet Protocol , the IPv6 protocol is released. However, transforming to IPv6 technology is not easy. We have to study and develop a plethora of new hardware and software. In addition, the cost of investment in the new technology is very high. The development of IPv6 in small devices is usually not distributed openly. So, developing small devices is a difficult thing to do.

This thesis concern with developing the IPv6 protocol on the embedded system, which is based on COM86 development kit and using the ready-support technology . Constructing the embedded IPv6 device, analysis and design of the system are included. This thesis can be used in education and development, in order to reduce the evolutionary steps and the cost of investment, especially in small devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ คำแนะนำและความร่วมมือจากหลายๆฝ่ายด้วยกัน โดยเฉพาะอย่างยิ่งอาจารย์ที่ปรึกษาของข้าพเจ้า ผศ.อภิเนตร อูนากูล ที่ได้ให้โอกาสข้าพเจ้าในการทำปริญญานิพนธ์ฉบับนี้ ซึ่งท่านได้ให้คำปรึกษา เอาใจใส่ ให้คำแนะนำและการช่วยเหลือตลอดมา นอกจากนั้นยังให้ความรู้และแนวคิดในการทำงานรวมถึงแนวทางในการดำเนินชีวิต อันมีค่ายิ่งต่อข้าพเจ้า ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาคอมพิวเตอร์ที่ได้อำนวยความสะดวกต่างๆ ในการทำวิจัย พี่ๆ เพื่อนๆ น้องๆ ในห้องวิจัยทุกๆห้อง ที่เป็นกำลังใจและให้ความช่วยเหลือในทุกๆเรื่อง ขอขอบคุณพี่เต๋า พี่บอมบ์ ที่ถึงแม้จะอยู่ไกลก็คอยเอาใจใส่ให้คำแนะนำอยู่เสมอ

ขอขอบคุณบริษัทเน็ตเกตเจ็ตที่เอื้อเฟื้อบอร์ดคอม 86 ที่ใช้ในการทำวิจัย ขอขอบคุณพี่โอที่ให้คำปรึกษาและเอื้อเฟื้อซอฟแวร์ ขอขอบใจเส่ง ตี๋และโจวที่ช่วยพิมพ์งาน ช่วยหาทรัพยากรต่างๆมาให้และคอยเอาใจใส่ดูแล เบิร์ดที่ช่วยงานกับให้กำลังใจ และเพื่อนๆทุกคน สำหรับน้ำใจอันมีค่า

และสุดท้ายนี้ต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดในชีวิตของที่ทำให้ข้าพเจ้าสามารถมีวันนี้ได้ นั่นคือ บิดา มารดา รวมทั้งญาติพี่น้องอันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดู อบรมสั่งสอนข้าพเจ้ามาเป็นอย่างดี รวมทั้งให้โอกาสทางการศึกษาอย่างเต็มที่ และให้กำลังใจเอาใจใส่เสมอมา ในทุกๆด้าน ข้าพเจ้าขอระลึกในพระคุณอันหาที่สุดมิได้และกราบขอบพระคุณมา ณ ที่นี้ด้วย

นางสาวชิตสุธา สุ่มเล็ก

นายฐิติวุฒิ ศิลปพิพัฒน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	IX
สารบัญภาพ	X
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขอบเขตของงานโครงงาน	2
1.4 เป้าหมายของโครงงาน	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 สถาปัตยกรรมทีซีพี/ไอพี	4
2.1.1 แบบการอ้างอิงทีซีพี/ไอพี	4
2.1.2 โปรโตคอลแสต็ก	5
2.1.3 การส่งถ่ายข้อมูลระหว่างชั้น	6
2.2 ไอพีรุ่น 6 และ ไอซีเอ็มพีรุ่น 6	7
2.3 ชุดพัฒนาคอม 86	11
2.3.1 คุณสมบัติของบอร์ดคอม86	11
2.3.2 สถาปัตยกรรมทางด้านฮาร์ดแวร์	12
2.3.2.1 ไมโครคอนโทรลเลอร์ Am186CC	13
2.3.2.2 หน่วยความจำ (Memory)	14
2.3.2.3 ยูเอสบี (USB)	16
2.3.2.4 พอร์ตอนุกรม	17
2.3.2.5 ระบบรีเซ็ต	17
2.3.2.6 อีเทอร์เน็ท	18
2.3.2.7 ยูอาร์ต UART (Universal Asynchronous Receiver - Transmitter)	18
2.3.2.8 พาวเวอร์ซัพพลาย (Power supply)	18
2.3.2.9 แอลอีดีอินดิเคเตอร์ (LED Indicator)	19
2.3.2.10 เอ็กแพลนชั่นพอร์ต (Expansion Port)	19
2.3.3 สถาปัตยกรรมทางด้านซอฟต์แวร์	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
2.3.4 จุดเด่นของคอม86	22
2.3.4.1 ใช้ไมโครคอนโทรลเลอร์ที่มีความสามารถสูง	22
2.3.4.2 มีเครื่องมือในการพัฒนาโปรแกรมมาก	23
2.3.4.3 ง่ายในการเรียนรู้และพัฒนา	23
2.3.4.4 มีความสามารถในการเพิ่มขยายได้	24
2.3.4.5 แอพพลิเคชันเป้าหมายของชุดพัฒนา (Target application)	24
บทที่ 3 ทฤษฎีและหลักการของเทคนิคที่เกี่ยวข้อง	26
3.1 การเขียนโปรแกรมให้ทำงานเป็นมัลติทาสก์	26
3.2 ไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2	27
3.2.1 จุดเด่นของไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2	27
3.2.2 แนวคิดแบบเรียลไทม์ซิสเต็ม	28
3.2.3 มิวเซอร์ลเอ็กซ์คูลชัน (Mutual Exclusion)	29
3.2.4 เซมาฟอร์ (Semaphores)	29
3.2.5 การสื่อสารระหว่างทาสก์ (Intertask Communication)	30
3.2.6 แมสเซสเมล์บ็อกซ์	30
3.3 ไลท์เวททีซีพี/ไอพีแสต็ก (Light weight protocol stack : lwIP)	31
3.3.1 โพรเซสโมเดล (Process Model)	32
3.3.2 อีมูเลชันเลเยอร์ของระบบปฏิบัติการ (The Operating System Emulation Layer)	32
3.3.3 การจัดการบัฟเฟอร์และหน่วยความจำ (Buffer and Memory management)	33
3.3.3.1 แพ็กเก็ตบัฟเฟอร์ (Packet buffers-pbufs)	33
3.3.3.2 การจัดการกับหน่วยความจำ (Memory management)	35
3.3.4 โปรโตคอลที่สนับสนุน	36
3.3.5 โครงสร้างข้อมูลที่เกี่ยวข้อง	37
3.3.5.1 eth_addr	37
3.3.5.2 eth_hdr	38
3.3.5.3 ip_addr	38
3.3.5.4 ip_hdr	38
3.3.5.5 netif	39
3.3.5.6 pbuf	39
3.3.5.7 tcp_hdr	40
3.3.5.8 tcp_pcb	40
3.3.5.9 tcp_pcb_listen	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
3.3.5.10 tcp_seg	42
3.3.5.11 udp_hdr	42
3.3.5.12 udp_pcb	43
บทที่ 4 การออกแบบและพัฒนา	44
4.1 ขอบเขตของโครงงาน.	44
4.2 โครงสร้างและองค์ประกอบของระบบ	45
4.2.1 โครงสร้างของสถาปัตยกรรมของระบบไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86	46
4.2.2 องค์ประกอบของระบบ	49
4.3 การพัฒนาไอพีรุ่น 6 บนระบบฝังตัว	52
4.3.1 การปรับค่าต่างๆของฮาร์แวร์	52
4.3.2 Internet Protocol, Version 6 (IPv6) Specification [rfc2460]	53
4.3.2.1 Hop-by-Hop Options Header	53
4.3.2.2 Routing Header	53
4.3.2.3 Fragment Header	53
4.3.2.4 Destination Options Header	54
4.3.2.5 ไม่มี Next Header	54
4.3.3 Path MTU Discovery for IP version 6	54
4.3.4 Neighbor Discovery for IP Version 6 (IPv6) [rfc2461]	54
4.3.5 IPv6 Stateless Address Autoconfiguration [rfc2462]	54
4.3.6 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [rfc2463]	55
4.3.7 IP Version 6 Addressing Architecture [rfc2373]	55
4.3.8 การพิจารณาเกี่ยวกับขอบเขต	56
4.3.9 การเลือกแอดเดรสหลักสำหรับไอพีรุ่น 6	56
4.3.10 Multicast Listener Discovery (MLD) for IPv6 [rfc2710]	56
4.3.11 IPsec สำหรับระบบฝังตัว	56
4.3.11.1 IPsec specification สำหรับระบบฝังตัว	56
4.3.11.2 Security Policy Database	56
4.3.11.3 Security Protocol	57
4.3.11.4 การทรานสฟอร์มและอัลกอลิทึม	57
4.3.11.5 SA และการจัดการเกี่ยวกับคีย์	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้าที่

4.4 การพัฒนาโปรแกรมบนชุดพัฒนาคอม86 57

4.4.1 การใช้งานพอร์ตอนุกรมของชุดพัฒนาคอม86 58

4.4.2 การเขียนโปรแกรมเพื่อให้ชุดพัฒนาคอม86 สามารถเชื่อมต่อเข้าสู่เครือข่าย 58

4.5 การพัฒนาโปรแกรมบนไลท์เวททีซีพี/ไอพีแสต็ก 59

4.6 การออกแบบระบบ 60

4.6.1 Use case diagram 60

4.6.2 Component diagram 62

4.6.3 Sequence diagram 63

4.6.3.1 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทาง คอนโซลพอร์ตอนุกรม 63

4.6.3.2 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางอีเทอร์เน็ท 64

4.6.3.3 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 4 65

4.6.3.4 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 6 66

บทที่ 5 ลักษณะการใช้งานระบบ 69

5.1 นักพัฒนา 70

5.2 ผู้เรียกใช้บริการจากระบบฝังตัว 72

5.3 ผู้พัฒนาไอพีรุ่น 6 บนระบบฝังตัว 73

บทที่ 6 การทดสอบและวิเคราะห์ผล 74

6.1 ยูนิตเทสติ้ง (Unit testing) 74

6.1.1 ทดลองส่งไฟล์เข้าไปในบอร์ดคอม 86 และรันโปรแกรม 74

6.1.2 ทดลองรับ-ส่งข้อมูลกับบอร์ดคอม 86 ผ่านทางพอร์ตอนุกรม 75

6.1.3 ทดลองให้บอร์ดคอม 86 ทำงานบนระบบเครือข่าย 75

6.1.4 ทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการวินโดว์ เอ็กซ์พี (Window XP) 75

6.1.5 ทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการลีนุกซ์ 78

6.2 อินทิเกรชันเทสติ้ง (Integration testing) 80

6.2.1 การทดสอบการทำงานแบบมัลติทาสก์กิ้ง 80

6.2.2 การทดสอบการทำงานของไลท์เวททีซีพี/ไอพีแสต็ก 80

6.2.3 การทดสอบการการกำหนดค่าคุณสมบัติผ่านคอนโซลพอร์ตอนุกรม 82

6.2.4 การทดสอบการการกำหนดค่าคุณสมบัติผ่านระบบเครือข่ายอีเทอร์เน็ท 84

6.2.5 การทดสอบ ping6 88

บทที่ 7 บทสรุปและวิจารณ์ 90

7.1 วิเคราะห์ผลจากการออกแบบและสร้างระบบ 90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้าที่
7.2 ปัญหาและอุปสรรค	90
7.3 แนวทางการพัฒนาต่อ	91
ภาคผนวก	92
ภาคผนวก ก. (การติดตั้งชุดพัฒนาคอม 86 และการติดตั้งโปรแกรม)	93
ภาคผนวก ข. (การติดตั้งไอพีรุ่น 6 บนระบบปฏิบัติการลีนุกซ์)	98
ภาคผนวก ค. (การสร้างไฟล์สำหรับคอมไพล์โปรแกรม)	99
บรรณานุกรม	102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ชื่อ	หน้าที่
ตารางที่ 3-1 โปรโตคอลที่สนับสนุน	36
ตารางที่ 4-1 แสดงเกี่ยวกับไอซีแมสแซสรุ่น 6 สำหรับระบบฝังตัว	55
ตารางที่ 4-2 อธิบาย Use case diagram	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

ชื่อ	หน้าที่
รูปที่ 1-1 กราฟแสดงการสำรวจจำนวนโฮสต์ประจำเดือนมกราคมปี ค.ศ.2004 ของ www.isc.org	1
รูปที่ 2-1 แบบอ้างอิงทีซีพี/ไอพี	5
รูปที่ 2-2 โปรโตคอลแสต็กของทีซีพี/ไอพี	5
รูปที่ 2-3 การห่อหุ้มข้อมูลตามลำดับโปรโตคอลแสต็ก	6
รูปที่ 2-4 บอร์ดคอม 86	12
รูปที่ 2-5 ส่วนประกอบต่างๆของคอม 86	12
รูปที่ 2-6 องค์ประกอบของไมโครคอนโทรลเอลร์AM186CC	14
รูปที่ 2-7 การแบ่งการใช้งานหน่วยความจำหลัก	15
รูปที่ 2-8 การจัดวางหน่วยความจำในส่วนของอินพุตท์เอาท์พุต	16
รูปที่ 2-9 คอนเนคเตอร์ของพอร์ตอนุกรม	17
รูปที่ 2-10 การเชื่อมต่อ PIO ของบอร์ดคอม 86	19
รูปที่ 2-11 ไอซ่าคอนเนคเตอร์ (ISA Connector)	21
รูปที่ 2-12 จีซีไอ	22
รูปที่ 2-13 รูปแบบการพัฒนาโปรแกรมโดยใช้คอม 86	23
รูปที่ 2-14 การเชื่อมต่อของบอร์ดขยายกับบอร์ดคอม 86	24
รูปที่ 3-1 มัลติทาสก์กิ้ง	26
รูปที่ 3-2 แมสเซสเมล์บ็อกซ์	31
รูปที่ 3-3 โครงสร้างของไลท์เวททีซีพี/ไอพีแสต็ก	31
รูปที่ 3-4 pbuf_ram	33
รูปที่ 3-5 pbuf_ram เชื่อมต่อเข้ากับ pbuf_rom ที่มีข้อมูลอยู่ที่ external memory	34
รูปที่ 3-6 chained pbuf_pool จาก pbuf_pool	34
รูปที่ 3-7 โครงสร้างของการกำหนดหน่วยความจำ	35
รูปที่ 3-8 โครงสร้างของ eth_hdr	38
รูปที่ 3-9 โครงสร้างของ ip_hdr	38
รูปที่ 3-10 โครงสร้างของ netif	39
รูปที่ 3-11 โครงสร้างของ pbuf	39
รูปที่ 3-12 โครงสร้างของ tcp_pcb	40
รูปที่ 3-13 โครงสร้างของ tcp_pcb_listen	42
รูปที่ 3-14 โครงสร้างของ tcp_seg	42
รูปที่ 3-15 โครงสร้างของ udp_pcb	43
รูปที่ 4-1 ขอบเขตของโครงงาน	44
รูปที่ 4-2 ภาพรวมของระบบและการเชื่อมต่อ	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

ชื่อ	หน้าที่
รูปที่ 4-3 แสดงการออกแบบลักษณะการใช้งานอุปกรณ์	45
รูปที่ 4-4 แสดงโครงสร้างสถาปัตยกรรมของระบบไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86	46
รูปที่ 4-5 แสดงโครงสร้างการทำงานระหว่างเลเยอร์	47
รูปที่ 4-6 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของไอซีเอ็มพี	47
รูปที่ 4-7 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของยูดีพี	48
รูปที่ 4-8 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของทีซีพี	48
รูปที่ 4-9 แสดงการทำงานของดีไวซ์ไดรเวอร์	50
รูปที่ 4-10 ส่วนที่ติดต่อกับผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม	51
รูปที่ 4-11 ส่วนที่ติดต่อกับผู้ใช้ผ่านทางอีเทอร์เน็ท	52
รูปที่ 4-12 ขั้นตอนพื้นฐานในการพัฒนาโปรแกรมทั่วไปบนชุดพัฒนาคอม 86	57
รูปที่ 4-13 รูปแบบการใช้โปรแกรมในการรับส่งไฟล์กับชุดพัฒนาคอม 86	58
รูปที่ 4-14 รีจิสเตอร์ที่เกี่ยวข้องในการเขียนโปรแกรมเพื่อติดต่อผ่านทางพอร์ตอนุกรม	59
รูปที่ 4-15 การสื่อสารกันผ่านเอพีไอ	60
รูปที่ 4-16 Use case diagram ของระบบ	60
รูปที่ 4-17 Component diagram ของระบบ	62
รูปที่ 4-18 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม	63
รูปที่ 4-19 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางอีเทอร์เน็ท	64
รูปที่ 4-20 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 4	65
รูปที่ 4-21 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบลิงค์โลคอลแอดเดรส	66
รูปที่ 4-22 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบ 6to4	67
รูปที่ 4-23 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบ IPv4 compatible	68
รูปที่ 5-1 โครงสร้างของไดเร็คทอรี่ในการเก็บซอร์ดโค้ด	69
รูปที่ 5-2 ตัวอย่างการแก้ไขบางส่วนในไฟล์ lwipopts.h	70
รูปที่ 5-3 ตัวอย่างการกำหนดค่าบางส่วนในไฟล์ os_cfg.h	71
รูปที่ 5-4 ตัวอย่างการกำหนดค่าบางส่วนในไฟล์ makefile	71
รูปที่ 5-5 ตัวอย่างการกำหนดในไฟล์ maketest.bat	72
รูปที่ 5-6 ตัวอย่างส่วนที่ต้องปรับแต่งเมื่อต้องการใช้งานหน้าเว็บเพ็จอื่น-1	72
รูปที่ 5-7 ตัวอย่างส่วนที่ต้องปรับแต่งเมื่อต้องการใช้งานหน้าเว็บเพ็จอื่น-2	72
รูปที่ 6-1 การทดลองส่งไฟล์เข้าไปในอุปกรณ์คอม 86	74
รูปที่ 6-2 การทดลองให้ชุดพัฒนาคอม 86 ทำงานบนเครือข่าย	75
รูปที่ 6-3 การติดตั้งไอพีรุ่น 6 ลงบนระบบปฏิบัติการวินโดว์	76
รูปที่ 6-4 ผลลัพธ์ที่ได้บางส่วนหลังจากการติดตั้งไอพีรุ่น 6 ลงบนระบบปฏิบัติการวินโดว์ -1	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

ชื่อ	หน้าที่
รูปที่ 6-5 ผลลัพธ์ที่ได้บางส่วนหลังจากการติดตั้งไอพีรุ่น 6 ลงบนระบบปฏิบัติการวินโดว์ -2	77
รูปที่ 6-6 ผลลัพธ์ที่ได้บางส่วนที่ได้จากโปรแกรมอีเทอร์เรียลและการ ping6	78
รูปที่ 6-7 ผลลัพธ์ที่ได้จากการใช้ ping6 ในระบบปฏิบัติการลีนุกซ์	79
รูปที่ 6-8 ผลลัพธ์ที่ได้จากการใช้ tracepath6 ไปยัง www.6bone.net ในระบบปฏิบัติการลีนุกซ์	79
รูปที่ 6-9 การทดสอบการทำงานของไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2	80
รูปที่ 6-10 การทดสอบการทำงานของไลท์เวททีซีพี/ไอพีแสต็ก-1	80
รูปที่ 6-11 การทดสอบการทำงานของไลท์เวททีซีพี/ไอพีแสต็ก-2	81
รูปที่ 6-12 การทดสอบการทำงานของไลท์เวททีซีพี/ไอพีแสต็ก-3	81
รูปที่ 6-13 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น 4 ให้กับระบบ	82
รูปที่ 6-14 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น6 แบบลิงค์โลคอลแอดเดรส	82
รูปที่ 6-15 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น6 แบบ Compatibility address (6to4)	83
รูปที่ 6-16 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น6 แบบ Compatibility address (IPv4 compatible)	83
รูปที่ 6-17 มีการรัน test.exe ก่อนที่จะไปกำหนดค่าผ่านหน้าเว็บโดยในที่นี่กำหนดให้รันที่ HTTP://161.246.5.140	84
รูปที่ 6-18 การเข้าถึงระบบด้วยโปรโตคอลเอชทีทีพี	84
รูปที่ 6-19 หน้าที่ใช้สำหรับล็อคอินเข้าสู่ระบบ	85
รูปที่ 6-20 หน้าเว็บที่แสดงออกมาหลังจากล็อกอินเข้ามาได้	85
รูปที่ 6-21 ส่วนที่แสดงผลออกมาขณะล็อกอินเข้ามาในเว็บแล้วถูกแสดงผ่านคอม86โดยจะมีการบอกว่ามีการเปิดหน้าเว็บนี้ขึ้นมาแล้ว (FSopenning menu.html)	86
รูปที่ 6-22 ใช้สำหรับกรอกค่าในการทดสอบการ Ping ในไอพีรุ่นสี่	86
รูปที่ 6-23 ใช้สำหรับกรอกค่าในการทดสอบการ Ping ในไอพีรุ่นหก	87
รูปที่ 6-24 ใช้สำหรับกรอกค่าการกำหนดค่าต่างๆให้กับระบบในแบบไอพีรุ่นสี่	87
รูปที่ 6-25 ใช้สำหรับกรอกค่าการกำหนดค่าต่างๆให้กับระบบในแบบไอพีรุ่นหก	88
รูปที่ 6-26 แสดงสถานะว่าได้รับ IPv6 packet เข้ามา	89
รูปที่ 6-27 แสดงการส่งและรับ ICMPv6 packet จาก Linux กับ COM86	89
รูปที่ ก-1 ภาพการเชื่อมต่อสาย serial console เข้ากับบอร์ดคอม86	93
รูปที่ ก-2 ภาพการเชื่อมต่อสายอีเทอร์เน็ทเข้ากับบอร์ดคอม86	93
รูปที่ ก-3 ใส่ชื่อรูปแบบการเชื่อมต่อ	94
รูปที่ ก-4 เลือกพอร์ตอนุกรมที่มาเชื่อมต่อ	94
รูปที่ ก-5 ภาพการตั้งค่าต่างๆ ของโปรแกรมเทอร์มินัล	95
รูปที่ ก-6 ภาพแสดงการเชื่อมต่ออแดปเตอร์กับบอร์ดคอม86	95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

ชื่อ	หน้าที่
รูปที่ ก-7 ภาพแสดงตัวอย่างการแสดงผลของบอร์ดคอม86เมื่อบอร์ดเริ่มทำงาน	96
รูปที่ ก-8 การ transfer fileกับอุปกรณ์คอม 86	96
รูปที่ ก-9 การส่งไฟล์โดยใช้เครื่องมือของโปรแกรมไฮเปอร์เทอร์มินัล	97
รูปที่ ก-10 การรันโปรแกรม	97

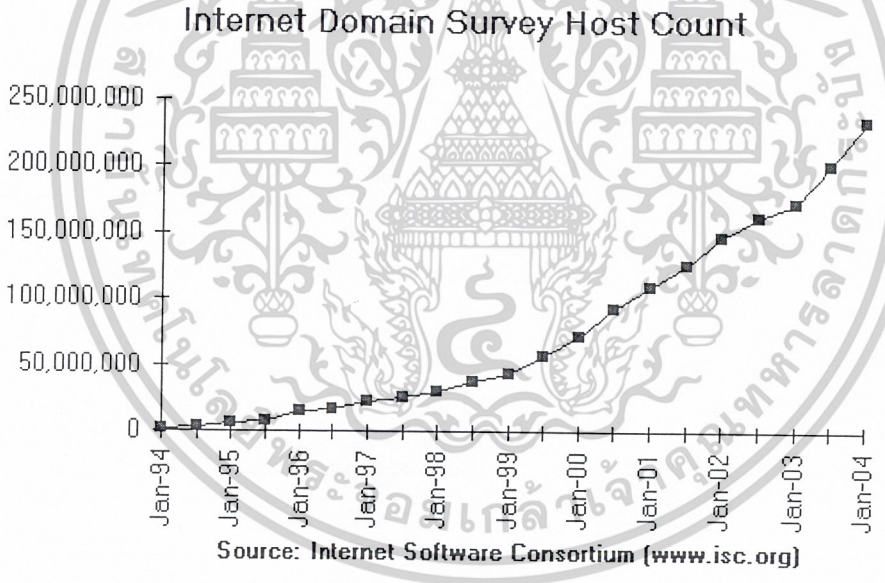
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบันการพัฒนาเทคโนโลยีนั้นมีการเติบโตอย่างรวดเร็ว และมีแนวโน้มที่จะเติบโตมากขึ้นเรื่อยๆในอนาคต เทคโนโลยีระบบฝังตัว (Embedded system) เองก็เป็นเทคโนโลยีหนึ่งที่ได้รับการพัฒนาและเข้ามาเป็นส่วนหนึ่งของชีวิตประจำวันของคนเรามากขึ้น ทั้งยังเข้ามามีบทบาทสำคัญต่อนักพัฒนาทั้งซอฟแวร์ (Software) และนักพัฒนาฮาร์ดแวร์ (Hardware) มากขึ้นเรื่อยๆ เนื่องจากความก้าวหน้าของเทคโนโลยีที่พัฒนาอย่างต่อเนื่องทำให้อุปกรณ์ที่ใช้เทคโนโลยีระบบฝังตัวต่างๆ นั้นได้รับการพัฒนาให้มีความสามารถมากขึ้น สามารถทำงานได้หลากหลายมากขึ้น มีขนาดที่เล็กลง และในขณะเดียวกันนั้นก็ใช้พลังงานจากแหล่งพลังงานน้อยลงด้วย ซึ่งส่งผลให้เทคโนโลยีระบบฝังตัวนั้นมีความสำคัญมากขึ้นในการพัฒนาทั้งทางด้านเทคโนโลยี อุตสาหกรรม การศึกษา และสร้างความสะดวกสบายให้ผู้ใช้มากขึ้นด้วยเช่นกัน

นอกจากนั้นการเชื่อมต่อคอมพิวเตอร์และอุปกรณ์ต่างๆ เป็นเครือข่ายนั้นก็มีการขยายตัวขึ้นอย่างรวดเร็ว ทำให้เกิดความต้องการเพื่อจะใช้บริการเครือข่ายนั้นสูงขึ้นเป็นเงาตามตัวด้วยเช่นกัน ดังรูปที่ 1-1 แสดงให้เห็นถึงความต้องการใช้บริการเครือข่ายที่มากขึ้นเรื่อยๆ ซึ่งสำรวจโดยเว็บไซท์ www.isc.org



รูปที่ 1-1 กราฟแสดงการสำรวจจำนวนโฮสต์ประจำเดือนมกราคมปี ค.ศ.2004 ของ www.isc.org

ผลที่ตามมาก็คือไอพีแอดเดรสที่มีอยู่นั้นจะไม่เพียงพอที่จะรองรับการขยายตัวที่เพิ่มขึ้นอย่างรวดเร็วดังกล่าว ซึ่งการนำเอาเทคโนโลยีไอพีรุ่น 6 (IPv6) มาใช้แทนไอพีรุ่น 4 ที่มีการใช้งานกันอยู่ในปัจจุบันก็เป็นทางเลือกหนึ่งที่ได้นำมาใช้ในการแก้ปัญหานี้ ซึ่งเป็นโปรโตคอล (Protocol) ที่ปรับปรุงมาจากโปรโตคอลไอพีให้มีประสิทธิภาพสูงขึ้นและสอดคล้องกับความต้องการทั้งในปัจจุบันและอนาคต

อย่างไรก็ตามการที่จะนำเอาเทคโนโลยีไอพีรุ่น 6 มาใช้นั้นยังต้องมีการพัฒนาอีกมากและไม่สามารถเข้าไปแทนที่การทำงานแบบเก่าได้ในทันที รวมทั้งการทำงานที่จะยังคงต้องรองรับการทำงานของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮาร์ดแวร์และซอฟแวร์ของไอพีรุ่น 4 ซึ่งมีอยู่มากมายในปัจจุบันนี้ให้ได้ด้วย ทำให้การพัฒนาเทคโนโลยีไอพีรุ่น 6 และการปรับเปลี่ยนเพื่อที่จะใช้งานเป็นการลงทุนที่สูงมาก ซึ่งเป็นอุปสรรคต่อการลงทุนและความต้องการในการใช้งานเทคโนโลยีในการพัฒนาทางด้านอุตสาหกรรมดังกล่าว ทำให้การเติบโตและการพัฒนาเป็นไปได้อย่างยากลำบาก อันส่งผลต่อการพัฒนาเศรษฐกิจของประเทศ รวมทั้งส่งผลให้เกิดความขาดแคลนโอกาสในการศึกษาเทคโนโลยีดังกล่าวแก่นิสิตนักศึกษาและผู้สนใจ ซึ่งไม่มีเงินทุนเพียงพอในการซื้อหาอุปกรณ์เหล่านี้มาศึกษาและทดลองใช้งานได้เอง

จากแนวคิดดังกล่าว ทางคณะผู้จัดทำปริญญานิพนธ์จึงได้พัฒนาอุปกรณ์ที่มีไอพีรุ่น 6 ฝังตัวอยู่ขึ้นมา โดยให้มีความสามารถเพียงพอที่ผู้ที่ต้องการศึกษา และนักพัฒนาที่ต้องการพัฒนาทางด้านเทคโนโลยีไอพีรุ่น 6 สามารถนำไปใช้งานได้โดยง่าย ซึ่งได้เลือกพัฒนาบนชุดพัฒนาคอม 86 เนื่องจากมีราคาถูกและคุณสมบัติมากเพียงพอที่จะพัฒนาให้เป็นอุปกรณ์ต่างๆ ได้หลายรูปแบบ

1.2 วัตถุประสงค์ของโครงงาน

1. สร้างแหล่งสะสมความรู้เกี่ยวกับไอพีรุ่น6
2. เพื่อเพิ่มเติมความสามารถของโอเพ่นซอร์สโปรโตคอลแสต็ก (open source protocol stack)
3. เพื่อพัฒนาเอ็นจิน (Engine) ที่จัดการการทำงานของโปรโตคอลแสต็ก (Protocol stack) ไอพีรุ่น 6
4. เพื่อให้ผู้ที่ต้องการศึกษาและนักพัฒนาสามารถลดขั้นตอนและค่าใช้จ่ายในการพัฒนาลงได้

1.2 ขอบเขตของโครงงาน

งานวิจัยนี้ครอบคลุมถึงการนำเอาทรัพยากรที่มีอยู่แล้วมารวมกันเพื่อให้ง่ายต่อการใช้งาน ให้เกิดประโยชน์ในการใช้งานมากขึ้นและลดขั้นตอนของการพัฒนาระบบต่างๆให้แก่นักพัฒนา โดย

1. การทำงานอยู่ภายใต้ความสามารถของอุปกรณ์คอม 86
2. ใช้ lwIP (Light weight TCP/IP protocol stack) ในการพัฒนา

1.3 เป้าหมายของโครงงาน

สร้างเอ็นจินในการควบคุมการทำงานของโปรโตคอลแสต็กไอพีรุ่น 6

1.4 วิธีการดำเนินงาน

1. วางแผนการดำเนินงานเบื้องต้น

เป็นการกำหนดขั้นตอนในการศึกษาและรายละเอียดของแต่ละขั้นตอนคร่าวๆ เพื่อให้ทราบถึงภาพรวมของการทำงาน และกำหนดระยะเวลาในการดำเนินงานของแต่ละขั้นตอน

2. ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้อง

ในขั้นตอนนี้จะทำการรวบรวมทฤษฎีต่างๆ และข้อมูลที่เกี่ยวข้องกับงานวิจัย ได้แก่ โปรโตคอลต่างๆ ที่เกี่ยวข้อง ชุดพัฒนาคอม 86 การทำงานของโปรโตคอลแสต็ก การทำงานแบบมัลติทาสก์กิ้ง (Multitasking) และเทคนิคต่างๆที่เกี่ยวข้องในการพัฒนา เพื่อนำมาใช้เป็นความรู้พื้นฐานสำหรับการวิเคราะห์ ออกแบบพัฒนาระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วิเคราะห์ข้อมูล

ทำการศึกษาและวิเคราะห์ข้อมูลที่ได้มา รวมทั้งการใช้วิธีการของวิศวกรรมย้อนกลับ(Reverse Engineering) ในการศึกษาโครงสร้าง ทำให้สามารถกำหนดขอบเขตและขั้นตอนของการทำงานได้โดยละเอียด

4. วิเคราะห์และออกแบบระบบ

นำความรู้ที่ได้จากการศึกษามาทั้งหมดนั้นมาทำการวิเคราะห์และออกแบบระบบโดยใช้หลักการของวิศวกรรมซอฟแวร์ (Software Engineering) ให้เหมาะสมกับการทำงานของชุดพัฒนาคอม 86

5. พัฒนาระบบ

นำเอาข้อมูลในส่วนวิเคราะห์และออกแบบระบบมาพัฒนาซอฟแวร์ และองค์ประกอบต่างๆของซอฟแวร์ โดยรวมถึงการพอร์ตระบบปฏิบัติการและโปรโตคอลแสต็กลงบนชุดพัฒนาคอม 86 ให้สามารถทำงานได้ตามที่ต้องการ

6. ทดสอบการทำงาน

ทำการทดสอบการทำงานของระบบว่าสามารถทำงานได้ถูกต้องตามที่กำหนดไว้หรือไม่ รวมทั้งเพื่อหาข้อผิดพลาดและแก้ไขหากพบข้อผิดพลาด

7. ติดตั้งและประเมินผลการใช้งาน

ทำการติดตั้งระบบในสภาพแวดล้อมที่สร้างขึ้นและทำการประเมินผลการใช้งาน เช่น ความสามารถของระบบ ความเหมาะสมของส่วนที่ติดต่อกับผู้ใช้ หาจุดบกพร่องที่ต้องแก้ไข เป็นต้น

8. สรุปผลการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 สถาปัตยกรรมทีซีพี/ไอพี

2.1.1 แบบการอ้างอิงทีซีพี/ไอพี

ระบบการสื่อสารข้อมูลในเครือข่ายคอมพิวเตอร์ประกอบด้วยทั้งฮาร์ดแวร์และซอฟแวร์ที่ซับซ้อน การมองภาพของระบบโดยรวมทั้งหมดเป็นหน่วยใหญ่ย่อมยากต่อการทำความเข้าใจ การใช้แบบอ้างอิงที่แบ่งระบบออกเป็นส่วนย่อยจะช่วยลดความซ้ำซ้อนและสร้างความเข้าใจได้ง่ายกว่า

เครือข่ายคอมพิวเตอร์มีแบบอ้างอิงที่ใช้เป็นมาตรฐานคือ แบบอ้างอิงโอเอสไอ (OSI : Open Systems Interconnection Reference Model) ในขณะที่ทีซีพี/ไอพีเป็นโปรโตคอลที่กำเนิดก่อนโอเอสไอและมีแบบอ้างอิงเฉพาะตามรูปที่ 2-1 แบบอ้างอิงทีซีพี/ไอพีระดับชั้นจากล่างขึ้นบนและลักษณะสมบัติประจำชั้นต่างๆ ดังต่อไปนี้

ฟิสิคัล ชั้นของการกำหนดคุณสมบัติฮาร์ดแวร์ เช่น คุณสมบัติทางกล (หัวต่อ และชนิดสายสื่อสาร) และคุณสมบัติทางไฟฟ้า (ลักษณะสัญญาณ และอัตราเร็ว) กล่าวโดยรวมแล้วระดับชั้นฟิสิคัล กำหนดวิธีการถ่ายโอนข้อมูลในระดับบิต ตัวอย่างของการเชื่อมต่อที่ตรงกับระดับชั้นฟิสิคัลได้แก่ อาร์เอส232 (RS232) และ เอ็กซ์.21 (X.21) เป็นต้น

เดทาลิงค์ ชั้นของซอฟแวร์ (ดีไวซ์ไดรเวอร์) และฮาร์ดแวร์ซึ่งทำงานด้านการเชื่อมโยงเข้ากับสารสื่อสาร ตัวอย่างมาตรฐานในระดับชั้นนี้ได้แก่ อีเทอร์เน็ทและโทเค็นริง เป็นต้น

เน็ตเวิร์ค ชั้นที่ทำหน้าที่เลือกเส้นทางเพื่อส่งข้อมูลระหว่างสถานีต้นทางและสถานีปลายทาง ตัวอย่างโปรโตคอลในระดับชั้นนี้ได้แก่ ไอพี

ทรานสปอร์ต ชั้นที่ทำหน้าที่จัดเตรียมการส่งข้อมูลระหว่างสถานีต้นทางและปลายทาง โดยสถาปนาการเชื่อมต่อและรักษาสภาพการเชื่อมต่อ ตลอดจนยกเลิกการเชื่อมต่อเมื่อสิ้นสุดกระบวนการ และอาจมีหน้าที่เพิ่มเติมในการรับประกันความถูกต้องของข้อมูลที่จัดส่ง ทีซีพี/ไอพีมีโปรโตคอลประจำชั้นนี้จำนวน 2 โปรโตคอลคือ ทีซีพีและยูดีพี

แอพพลิเคชัน ระดับชั้นนี้กำหนดการทำงานของโปรโตคอลประยุกต์ ตัวอย่างโปรโตคอลในระดับชั้นนี้ได้แก่ เอฟทีพี (FTP) เอสเอ็มทีพี (SMTP) หรือเทลเน็ท (TELNET) เป็นต้น

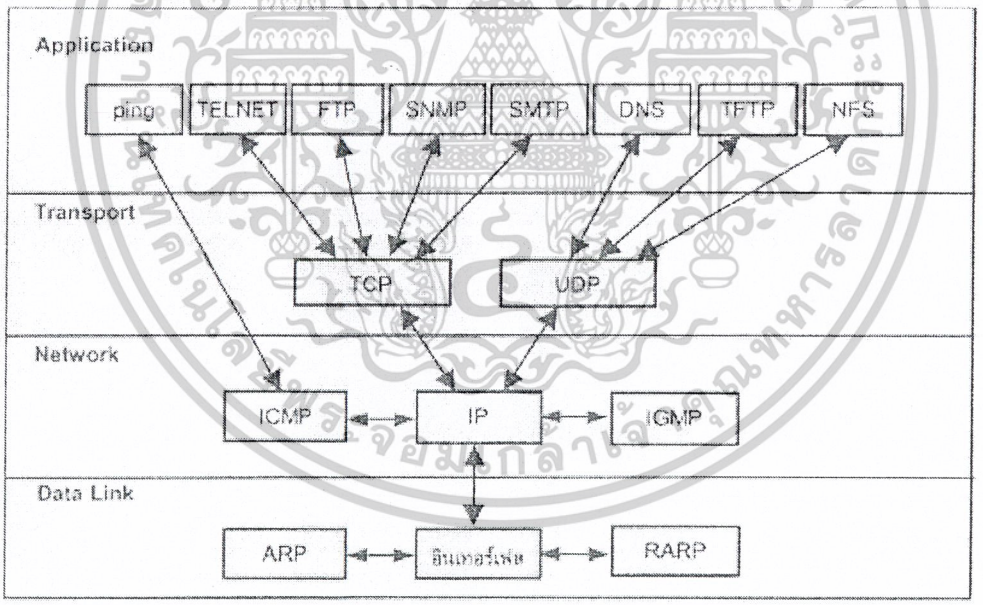
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอพพลิเคชัน	เทลเน็ท, เอฟทีพี, ดีเอ็นเอส ,…
ทรานสปอร์ต	ทีซีพี, ยูดีพี
เน็ตเวิร์ค	ไอพี, ไอซีเอ็มพี
เดทาลิงค์	ไดร์เวอร์และฮาร์ดแวร์เครือข่าย
ฟิสิคัล	อินเทอร์เฟสระดับชั้นกายภาพ

รูปที่ 2-1 แบบอ้างอิงทีซีพี/ไอพี

2.1.2 โปรโตคอลแสต็ก

การทำงานตามโปรแกรมประยุกต์หนึ่งๆไม่ได้ใช้โปรโตคอลพร้อมกันทั้งชุด หากแต่ใช้เพียงโปรโตคอลที่สัมพันธ์กันไปในแต่ละชั้นของแบบอ้างอิง ตัวอย่างเช่น เทลเน็ทจะอาศัยทีซีพีและไอพีตามลำดับ การซ้อนทับของโปรโตคอลจากระดับชั้นบนไปชั้นล่างเรียกว่า **โปรโตคอลแสต็ก** (protocol stack) รูปที่ 2-2 แสดงโปรโตคอลแสต็กของทีซีพี/ไอพี



รูปที่ 2-2 โปรโตคอลแสต็กของทีซีพี/ไอพี

ไอพีซึ่งอยู่ในระดับชั้นเน็ตเวิร์คตามรูป 2-2 นั้นเป็นแกนสำคัญของโปรโตคอลแสต็ก เนื่องจากทั้งทีซีพีและยูดีพีต้องใช้ไอพีเพื่อเลือกเส้นทางส่งแพ็กเก็ต ในระดับชั้นเน็ตเวิร์คยังมีไอซีเอ็มพีสนับสนุนการทำงานของไอพีเพื่อรายงานข้อผิดปกติที่เกิดขึ้นจากการส่งแพ็กเก็ตและมีไอจีเอ็มพีดูแลการจัดกลุ่มโฮสต์ในเครือข่ายมัลติคาสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

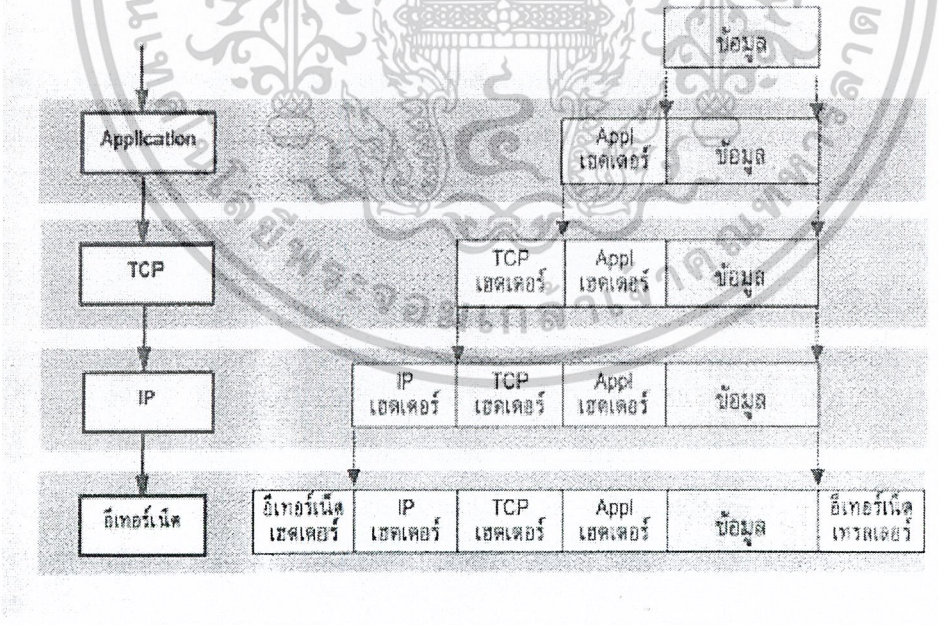
ระดับชั้นทรานสพอร์ตมี 2 โปรโตคอลสำคัญคือทีซีพีและยูดีพี แอพพลิเคชันจะเลือกใช้ทีซีพีหรือยูดีพีตามลักษณะงาน

โปรโตคอลระดับล่างถัดจากไอพีได้แก่ โปรโตคอลระดับเดทาลิงค์ซึ่งกำหนดการทำงานตามเทคโนโลยีเครือข่ายที่ใช้งานเช่น ซีเอสเอ็มเอ/ซีดี (CSMA/CD) ตามมาตรฐานอีเทอร์เน็ท ในระดับชั้นนี้มีโปรโตคอลในชุดของทีซีพี/ไอพีทำหน้าที่สนับสนุนการทำงานอยู่ 2 โปรโตคอลคือเออาร์พี (ARP) และอาร์เออาร์พี (RARP) ทั้งสองโปรโตคอลทำหน้าที่แปลงค่าระหว่างไอพีแอดเดรสกับฮาร์ดแวร์แอดเดรส

2.1.3 การส่งถ่ายข้อมูลระหว่างชั้น

โปรโตคอลในแต่ละชั้นล้วนมีหน้าที่เกี่ยวข้องในการส่งผ่านข้อมูลจากสถานีต้นทางไปยังสถานีปลายทาง ข้อมูลจะถูกส่งผ่านจากโปรโตคอลระดับบนสุดจากสถานีต้นทางไปยังระดับล่างจนกระทั่งข้อมูลถูกแปลงให้อยู่ในรูปของสัญญาณไฟฟ้าแล้วเดินทางผ่านเครือข่ายไปยังสถานีปลายทาง โปรโตคอลระดับล่างสุดที่สถานีปลายทางจะรับสัญญาณและส่งผ่านขึ้นไปยังโปรโตคอลระดับบนต่อไป

เมื่อข้อมูลผ่านแต่ละระดับชั้น โปรโตคอลในชั้นนั้นจะผนวกข่าวสารกำกับการทำงานประจำโปรโตคอลซึ่งเรียกว่า **โปรโตคอลเฮดเดอร์** (protocol header) เข้ากับข้อมูล เฮดเดอร์และตัวข้อมูลจากระดับบนจะถูกส่งผ่านไปยังระดับล่าง โปรโตคอลระดับล่างจะมองเฮดเดอร์และตัวข้อมูลรวมเป็นเสมือนข้อมูลและเพิ่มเฮดเดอร์ประจำชั้นเข้าไป ข้อมูลเดิมจึงมีเฮดเดอร์หุ้มเป็นชั้นๆ กระบวนการนี้เรียกว่า **การเอ็นแคปซูแลต** (encapsulation) ตัวอย่างในรูปที่ 2-3 แสดงการเอ็นแคปซูแลตแพ็กเก็ตทีซีพี/ไอพีในอีเทอร์เน็ท



รูปที่ 2-3 การห่อหุ้มข้อมูลตามลำดับโปรโตคอลแสต็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสถานีปลายทางได้รับแพ็กเก็ตก็จะดำเนินการส่งไปตามลำดับชั้น โปรโตคอลประจำชั้นจะถอดเฮดเดอร์ออกและส่งส่วนที่เหลือไปยังชั้นถัดไป เฮดเดอร์จะถูกถอดออกเหลือเฉพาะข้อมูลเมื่อถึงชั้นบนสุด กระบวนการที่เรียกว่า **การดีแคปซูลเลต** (decapsulation)

2.2 ไอพีรุ่น 6 และไอซีเอ็มพีรุ่น 6

ความต้องการไอพีใหม่เนื่องมาจากปัญหาการขยายตัวของอินเทอร์เน็ทซึ่งเป็นเหตุผลเดียวกับเรื่องไซเดอร์ (CIDR:Classless Inter-Domain Routing) ไซเดอร์ช่วยให้การใช้ไอพีแอดเดรสมีประสิทธิภาพเพิ่มขึ้น และยังช่วยรักษาขนาดตารางเลือกเส้นทางไม่ให้ขยายเกินไปกว่าที่เราเตอร์ (Router) จะรองรับได้ แต่ไซเดอร์เป็นเพียงการบรรเทาปัญหาเฉพาะหน้าและไม่ใช่หนทางแก้ปัญหาในระยะยาว

การใช้ไอพีแอดเดรสให้ได้ประสิทธิภาพร้อยเปอร์เซ็นต์เต็มตามจำนวนแอดเดรสทั้งหมดที่มีอยู่ประมาณสี่พันล้านแอดเดรสเป็นสิ่งที่ทำไม่ได้ในทางปฏิบัติ ปัญหาขาดแคลนไอพีแอดเดรสย่อมเกิดก่อนที่จะใช้แอดเดรสได้ทั้งหมด หรือถึงแม้จะมีวิธีใช้แอดเดรสได้อย่างเต็มประสิทธิภาพก็ตาม แอดเดรสไอพีรุ่นสี่ดูเหมือนจะไม่เพียงพอต่อการใช้งานในรูปแบบใหม่ๆ เช่น คอมพิวเตอร์ในรถยนต์ หรือเซ็ตท็อปบ็อกซ์สำหรับเว็บทีวี ความต้องการใช้แอดเดรสจำนวนมากเป็นสิ่งที่หลีกเลี่ยงไม่ได้ในอนาคต

ไออีทีเอฟเริ่มมองปัญหาขาดแคลนแอดเดรสและหาแนวทางแก้ไขมาตั้งแต่ปี 2524 ตลอดระยะเวลาที่ผ่านมามีผู้เสนอวิธีการหลากหลายรูปแบบ ไออีทีเอฟได้คัดเลือกข้อเสนอและกำหนดให้ไอพีในยุคถัดไปคือ **ไอพีรุ่นหก** (IP version 6 หรือ IPv6)

เนื่องจากแอดเดรสบรรจุอยู่ในเฮดเดอร์ของทุกแพ็กเก็ต การขยายขนาดแอดเดรสย่อมต้องเปลี่ยนแปลงขนาดของเฮดเดอร์ และหมายถึงการปรับเปลี่ยนโปรโตคอลใหม่ ผลที่ตามมาอีกคือซอฟต์แวร์ในทุกๆโฮสต์และเราเตอร์ในอินเทอร์เน็ทต้องเปลี่ยนแปลงด้วยเช่นกัน การเปลี่ยนแปลงทั่วทั้งอินเตอร์เน็ทเป็นเรื่องใหญ่เกินกว่าที่จะทำได้ในระยะเวลาสั้น และยิ่งทำได้ยากหากไอพีรูปแบบใหม่มีความแตกต่างไปอย่างสิ้นเชิงจากรูปแบบเดิม

ในความเป็นจริงแล้วการปรับปรุงไอพีอาจใช้วิธีขยายขนาดแอดเดรสเพียงอย่างเดียวโดยไม่เปลี่ยนแปลงส่วนอื่นใดก็ได้ เพราะไอพีเดิมถือได้ว่าผ่านการออกแบบและปรับมาตามลำดับ แต่จากประสบการณ์การใช้ไอพีรุ่นสี่ตลอดระยะเวลาเกือบยี่สิบปีที่ผ่านมา ช่วยชี้ให้เห็นถึงจุดที่มีอยู่ในไอพีเดิมรวมทั้งความต้องการใช้งานแบบใหม่ที่จำเป็นให้มีประสิทธิภาพสูงขึ้นและสอดคล้องกับความต้องการทั้งในปัจจุบันและอนาคต ไอพีรุ่นหกมีข้อปรับปรุงเพิ่มมาจากไอพีรุ่นสี่ดังต่อไปนี้

- ขยายขนาดแอดเดรสขึ้นเป็น 128 บิต
- เพิ่มขีดความสามารถการเลือกเส้นทาง และสนับสนุนโมไบล์โฮสต์ (Mobile hosts)
- สนับสนุนการทำงานแบบเวลาจริง (real-time services)
- มีระบบติดตั้งแอดเดรสอัตโนมัติ (autoconfiguration)
- เพิ่มระบบรักษาความปลอดภัย

ส่วนไอซีเอ็มพีรุ่น 6 หรืออินเทอร์เน็ทคอนโทรลแมสเซสโปรโตคอลสำหรับอินเทอร์เน็ทโปรโตคอลรุ่น 6 (ICMPv6 : Internet Message Protocol for the Internet Protocol version 6) นั้นถูกใช้งานโดยโหนดไอพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่น 6 (IPv6 node) เพื่อรายงานข้อผิดพลาดที่พบในกระบวนการที่เกี่ยวกับแพ็กเก็ต และเพื่อสร้างอินเทอร์เน็ทเลเยอร์ฟังก์ชันอื่นๆ เช่น การวินิจฉัย (ICMPv6 "ping") ไอซีเอ็มพีรุ่น 6 นั้นรวมเอามาเป็นส่วนหนึ่งของไอพีรุ่น 6 และต้องถูกอิมพลีเมนท์ (implement) อย่างครบถ้วนโดยทุกโหนดไอพีรุ่น 6

ซึ่งรายละเอียดทั้งหมดของไอพีรุ่น 6 และไอซีเอ็มพีรุ่น 6ผู้อ่านสามารถศึกษาได้จาก

RFC791

Internet Protocol

RFC792

Internet Control Message Protocol (ICMP)

RFC1809

Using the Flow Label Field in IPv6. C. Partridge. June 1995. (Format: TXT=13591 bytes) (Status: INFORMATIONAL)

RFC1881

IPv6 Address Allocation Management. IAB & IESG. December 1995. (Format: TXT=3215 bytes) (Status: INFORMATIONAL)

RFC1884

IP version 6 Addressing Architecture

RFC1887

An Architecture for IPv6 Unicast Address Allocation. Y. Rekhter & T. Li, Editors. December 1995. (Format: TXT=66066 bytes) (Status: INFORMATIONAL)

RFC1924

A Compact Representation of IPv6 Addresses. R. Elz. April 1996. (Format: TXT=10409 bytes) (Status: INFORMATIONAL)

RFC1933

Transition Mechanisms for IPv6 Hosts and Routers. R. Gilligan & E. Nordmark. April 1996. (Format: TXT=47005 bytes) (Status: PROPOSED STANDARD)

RFC2073

An IPv6 Provider-Based Unicast Address Format

RFC2080

RIPng for IPv6. G. Malkin, R. Minnear. January 1997. (Format: TXT=47534 bytes) (Status: PROPOSED STANDARD)

RFC2081

RIPng Protocol Applicability Statement. G. Malkin. January 1997. (Format: TXT=6821 bytes) (Status: INFORMATIONAL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RFC2133

Basic Socket Interface Extensions for IPv6. R. Gilligan, S. Thomson, J. Bound, W. Stevens. April 1997. (Format: TXT=69737 bytes) (Status: INFORMATIONAL)

RFC2147

TCP and UDP over IPv6 Jumbograms. D. Borman. May 1997. (Format: TXT=1883 bytes) (Status: PROPOSED STANDARD)

RFC2185

Routing Aspects of IPv6 Transition. R. Callon, D. Haskin. September 1997. (Format: TXT=31281 bytes) (Status: INFORMATIONAL)

RFC2292

Advanced Sockets API for IPv6. W. Stevens, M. Thomas. February 1998. (Format: TXT=152077 bytes) (Status: INFORMATIONAL)

RFC2374

An IPv6 Aggregatable Global Unicast Address Format. R. Hinden, M. O'Dell, S. Deering. July 1998. (Format: TXT=25068 bytes) (Obsoletes RFC2073) (Status: PROPOSED STANDARD)

RFC2375

IPv6 Multicast Address Assignments. R. Hinden, S. Deering. July 1998. (Format: TXT=14356 bytes) (Status: INFORMATIONAL)

RFC2428

FTP Extensions for IPv6 and NATs. M. Allman, S. Ostermann, C. Metz. September 1998. (Format: TXT=16028 bytes) (Status: PROPOSED STANDARD)

RFC2460

Internet Protocol, Version 6 (IPv6) Specification. S. Deering, R. Hinden. December 1998. (Format: TXT=85490 bytes) (Obsoletes RFC1883) (Status: DRAFT STANDARD) (in French)

RFC2461

Neighbor Discovery for IP Version 6 (IPv6). T. Narten, E. Nordmark, W. Simpson. December 1998. (Format: TXT=222516 bytes) (Obsoletes RFC1970) (Status: DRAFT STANDARD)

RFC2462

IPv6 Stateless Address Autoconfiguration. S. Thomson, T. Narten. December 1998. (Format: TXT=61210 bytes) (Obsoletes RFC1971) (Status: DRAFT STANDARD)

RFC2463

Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. A. Conta, S. Deering. December 1998. (Format: TXT=34190 bytes) (Obsoletes RFC1885) (Status: DRAFT STANDARD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RFC2464

Transmission of IPv6 Packets over Ethernet Networks. M. Crawford. December 1998. (Format: TXT=12725 bytes) (Obsoletes RFC1972) (Status: PROPOSED STANDARD)

RFC2465

Management Information Base for IP Version 6: Textual Conventions and General Group. D. Haskin, S. Onishi. December 1998. (Format: TXT=77339 bytes) (Status: PROPOSED STANDARD)

RFC2466

Management Information Base for IP Version 6: ICMPv6 Group. D. Haskin, S. Onishi. December 1998. (Format: TXT=27547 bytes) (Status: PROPOSED STANDARD)

RFC2467

Transmission of IPv6 Packets over FDDI Networks. M. Crawford. December 1998. (Format: TXT=16028 bytes) (Obsoletes RFC2019) (Status: PROPOSED STANDARD)

RFC2470

Transmission of IPv6 Packets over Token Ring Networks. M. Crawford, T. Narten, S. Thomas. December 1998. (Format: TXT=21677 bytes) (Status: PROPOSED STANDARD)

RFC2471

IPv6 Testing Address Allocation. R. Hinden, R. Fink, J. Postel (deceased). December 1998. (Format: TXT=8031 bytes) (Obsoletes RFC1897) (Status: EXPERIMENTAL)

RFC2472

IP Version 6 over PPP. D. Haskin, E. Allen. December 1998. (Format: TXT=29696 bytes) (Obsoletes RFC2023) (Status: PROPOSED STANDARD)

RFC2473

Generic Packet Tunneling in IPv6 Specification. December 1998. (Format: TXT=77956 bytes) (Status: PROPOSED STANDARD)

RFC2474

Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. K. Nichols, S. Blake, F. Baker, D. Black. December 1998. (Format: TXT=50576 bytes) (Obsoletes RFC1455, RFC1349) (Status: PROPOSED STANDARD)

RFC2491

IPv6 over Non-Broadcast Multiple Access (NBMA) networks. G. Armitage, P. Schulter, M. Jork, G. Harter. January 1999. (Format: TXT=100782 bytes) (Status: PROPOSED STANDARD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RFC2492

IPv6 over ATM Networks. G. Armitage, P. Schulter, M. Jork. January 1999. (Format: TXT=21199 bytes) (Status: PROPOSED STANDARD)

RFC2497

Transmission of IPv6 Packets over ARCnet Networks. I. Souvatzis. January 1999. (Format: TXT=10304 bytes) (See Also: RFC1201) (Status: PROPOSED STANDARD)

และสามารถศึกษาเพิ่มเติมได้จากแหล่งอ้างอิงในส่วนของบรรณานุกรม

2.3 ชุดพัฒนาคอม 86

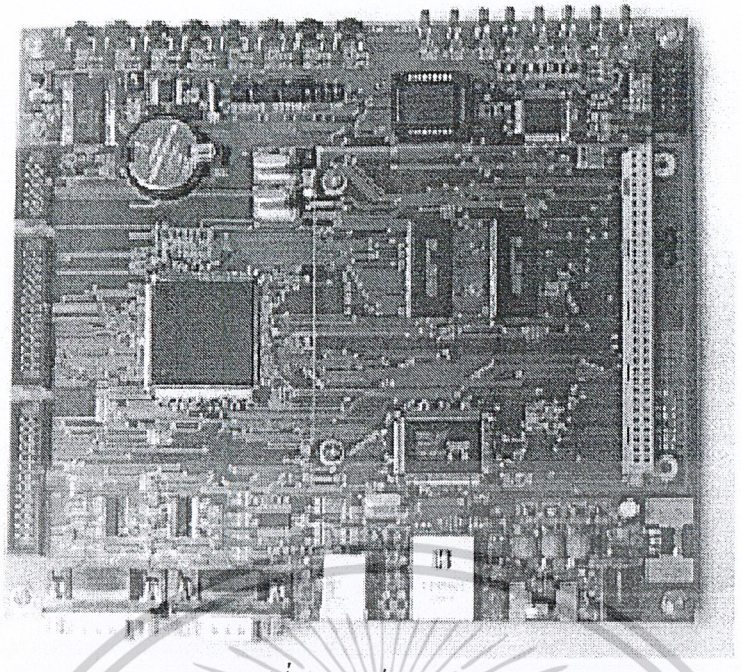
ชุดพัฒนาคอม86 เป็นชุดเครื่องมือสำหรับใช้ในการพัฒนาอุปกรณ์ทางด้านระบบฝังตัวที่อยู่บนพื้นฐาน การพัฒนาโดยการใช้แพลตฟอร์มที่มีไมโครโพรเซสเซอร์ตระกูล x86 ซึ่งเป็นที่คุ้นเคยกับผู้ใช้งานในปัจจุบันเป็นส่วนประกอบหลักในการทำงาน

2.3.1 คุณสมบัติของบอร์ดคอม86

บอร์ดคอม86มีคุณสมบัติดังนี้

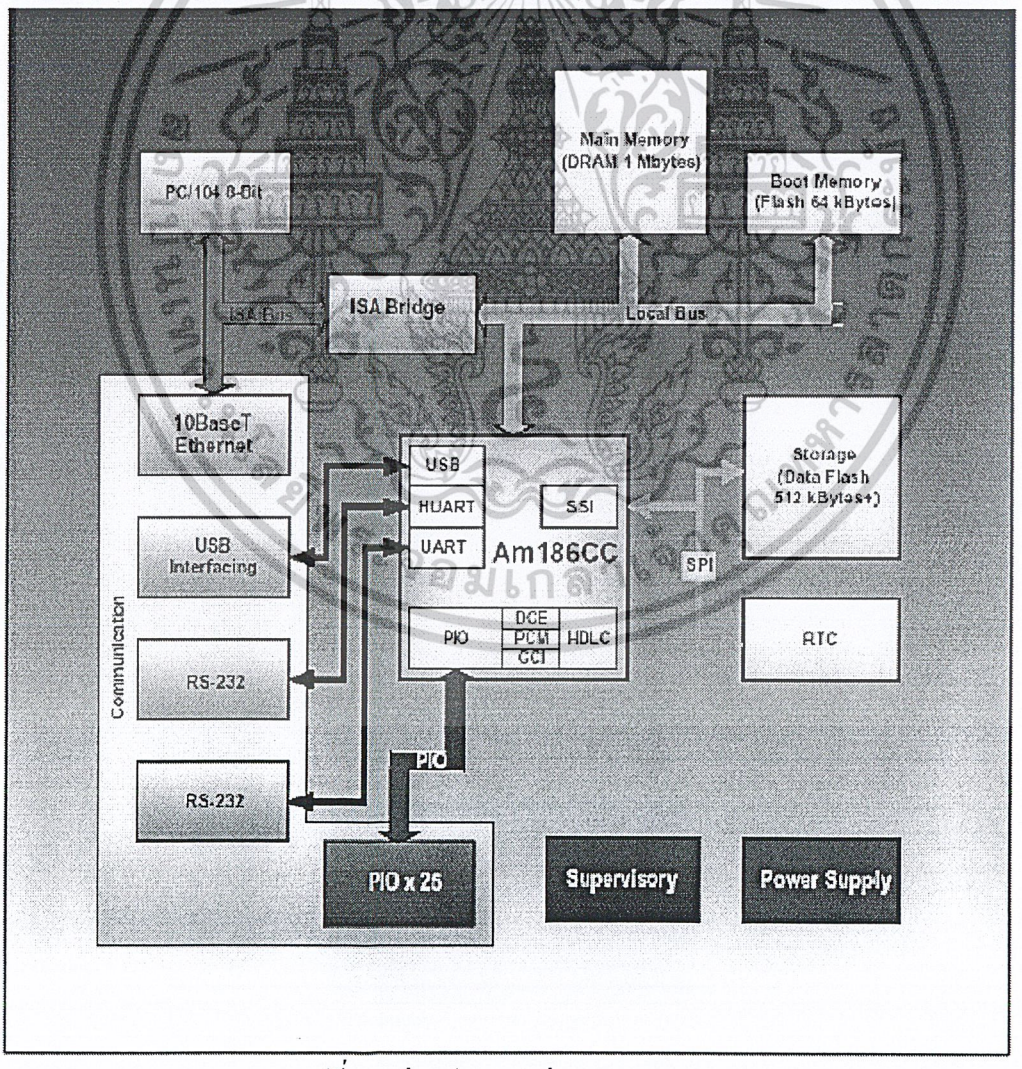
- ซีพียู เอเอ็มหนึ่งแปดหกซีซี (CPU Am186CC)
- ดีแรม 1 เมกกะไบต์ (1 MBytes DRAM)
- แฟลตไบออส 64 กิโลไบต์ (64 kBytes Flash BIOS)
- หน่วยความจำ 512 กิโลไบต์
- ฟูลดูเพล็ก IEEE 802.3 บนบอร์ดอีเทอร์เน็ทคอนโทรลเลอร์ด้วย RJ45 คอนเน็คเตอร์
- ยูเอสบี คอนเน็คเตอร์
- พอร์ตอนุกรมคู่ อาร์เอส232 ด้วย ดีบี-9 คอนเน็คเตอร์
- 4 ช่องสัญญาณเอชดีเอลซี (HDLC : High level Data Link Control)
- สี่ไทม์สล๊อต (TSAs : Time slot Assigner)
- จีซีไอ คอนโทรลเลอร์
- 3 ไทม์เมอร์ที่สามารถโปรแกรมได้
- นาฬิกา เรียลไทม์
- วอชด็อก ไทม์เมอร์ และ รีเซ็ท คอนโทรลเลอร์
- อินเทอร์รัพ คอนโทรลเลอร์
- 25 อินพุต/เอาท์พุต ที่สามารถโปรแกรมได้
- ไอซ่า บัส ขนาด 8 บิต
- เพาเวอร์ซัพพลาย 3.3 โวลต์ และ 5 โวลต์
- เอลอีดี แสดงสถานะของเพาเวอร์ซัพพลาย และ อีเทอร์เน็ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 บอร์ดคอม 86

2.4.2 สถาปัตยกรรมทางด้านฮาร์ดแวร์



รูปที่ 2-5 ส่วนประกอบต่างๆของคอม 86

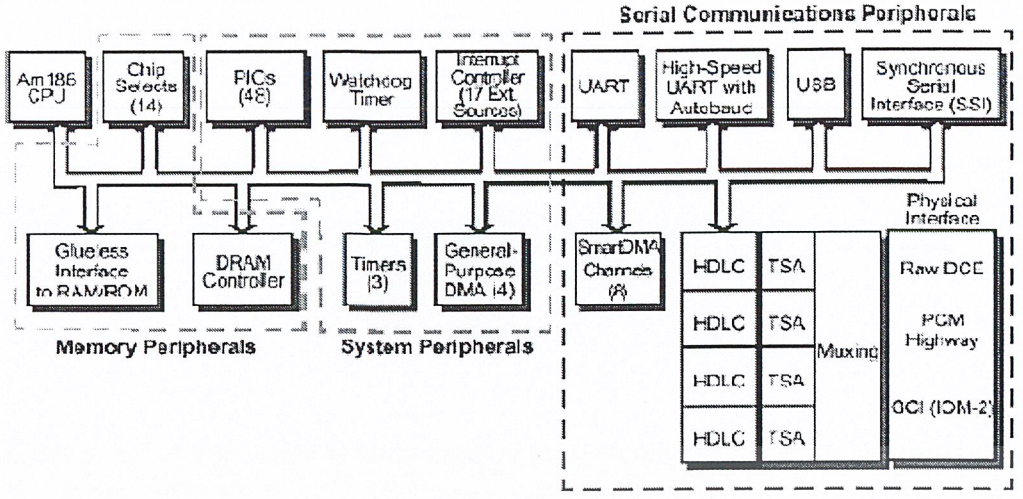
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.1 ไมโครคอนโทรลเลอร์ Am186CC

ไมโครคอนโทรลเลอร์ AM186CC ถือว่าเป็นหัวใจหลักที่ควบคุมการทำงานส่วนต่างๆของบอร์ดคอม 86 ซึ่งไมโครคอนโทรลเลอร์ตัวนี้เป็นไมโครคอนโทรลเลอร์ขนาด 16 บิต ในตระกูล อี86 ของบริษัทเอเอ็ม ดี ชื่อว่า Am186CC ซึ่งเป็นคอมมิวนิเคชันไมโครคอนโทรลเลอร์ (Communication Microcontroller) ที่ใช้แกนหลัก (core) ของ 80186 โดยได้เพิ่มความสามารถทางด้านการสื่อสาร และส่วนประกอบอื่นๆ โดยมีส่วนสำคัญที่ได้เพิ่มเข้าไปดังนี้

- ส่วนเอชดีแอลซี (HDLC) หรือ ไฮเลเวลดาต้าลิ้งค์คอนโทรล (High–level Data Link Control) สำหรับเป็นส่วนจัดการเรื่องการติดต่อสื่อสารแบบเอชดีแอลซี ที่สามารถเชื่อมต่อกับ ดีซีอี, พีซีเอ็ม ไฮเวย์ และ จีซีไอ ผ่านทางพีไอโอโดยมีทีซ่า (TSAs) หรือ ไทม์สลอต แอสซายเนอร์ (Time Slot Assigners) สำหรับการทำงานในแบบมัลติเพลกซ์ (Multiplex)
- ยูเอสบีเพอร์ริพารอลคอนโทรล (USB Peripheral Controller) สำหรับการทำเป็นอุปกรณ์ยูเอสบี
- เอชยูอาร์ต (HUART) หรือ ไฮสปีดยูอาร์ต (High-speed UART) สำหรับใช้งานในการเชื่อมต่อ แบบอนุกรมความเร็วสูง โดยสามารถสื่อสารได้ในอัตราเร็วสูงสุด 460 กิโลบิตต่อวินาที (Kbit /s) สามารถตรวจสอบอัตราเร็วในการเชื่อมต่อได้อัตโนมัติ (Automatic Baud Rate Detection) และมีฮาร์ดแวร์แฮนชาร์คกิ้ง(Handshaking)
- ยูอาร์ต (UART) หรือ ยูนิเวอร์ซัลดซิงโคนัสรีซีพเวอร์/ทรานส์มิทเทอร์ (Universal Asynchronous Receiver/Transmitter) สำหรับการเชื่อมต่อแบบอนุกรมโดยสามารถสื่อสารได้ในอัตราเร็วสูงสุด 115.2 กิโลบิตต่อวินาที และมีฮาร์ดแวร์แฮนชาร์คกิ้ง
- เอสเอสไอ (SSI) หรือ ซิงโคนัสซีเรียลอินเทอร์เฟส (Synchronous Serial Interface) สำหรับการเชื่อมต่อ สื่อสารแบบอนุกรมความเร็วสูง 25 เมกกะบิตต่อวินาที (Mbit/s) ซึ่งสามารถสื่อสารแบบ
- โปรแกรมเมเบิลไทม์เมอร์ (Programmable Timer) ขนาด 16 บิต 3 ตัว
- ฮาร์ดแวร์วอชด็อกไทม์ (Hardware Watchdog Time)
- ดีแรมคอนโทรลเลอร์ (DRAM Controller)
- อินเทอร์รัพคอนโทรลเลอร์ (Interrupt Controller) 36 มาสก์เอเบิลอินเทอร์รัพ
- โปรแกรมเมเบิลอินพุทเอาท์พุทซิกแนล (Programmable I/O Signal) ซึ่งมีจำนวน 48 แชนแนล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 องค์ประกอบของไมโครคอนโทรลเอลร์AM186CC

ไมโครคอนโทรลเลอร์ Am186CC นี้ เป็นไมโครคอนโทรลเลอร์ที่ที่มีสถาปัตยกรรมแบบ x86 มีชุดคำสั่งที่เข้ากันได้กับชุดคำสั่งของ 80286 (Real Mode) สนับสนุนการอ้างอิงหน่วยความจำหลักได้ถึง1 เมกกะไบต์

2.3.2.2 หน่วยความจำ (Memory)

ภายในบอร์ดคอม86 ประกอบด้วยหน่วยความจำหลายส่วนประกอบเข้าด้วยกันซึ่งมีรายละเอียดต่างๆ ดังนี้

1. แรม (RAM)

หน่วยความจำส่วนนี้เป็นส่วนที่ใช้ในการทำงานของโปรแกรมต่างๆ ของบอร์ดคอม86 ซึ่งมีขนาด 1 เมกกะไบต์ อันเนื่องมาจากจากสถาปัตยกรรมของไมโครคอนโทรลเลอร์หน่วยความจำในส่วนนี้จะใช้ เป็นพื้นที่ในการใช้งานสำหรับโปรแกรมต่างๆทั้งระบบปฏิบัติการและโปรแกรมของนักพัฒนา โดยหน่วยความจำนี้เป็นหน่วยความจำชนิดอีดีโอ (EDO) ขนาด 1 เมกกะไบต์โดยมีการควบคุมการทำงานจากส่วนของดีแรมคอนโทรลเลอร์ภายในไมโครคอนโทรลเลอร์

2. แฟลชไบออส

หน่วยความจำส่วนนี้เป็นหน่วยความจำแบบแฟลชขนาด 64 กิโลไบต์สำหรับใช้ในการเก็บโปรแกรมไบออสสำหรับควบคุมการทำงานพื้นฐานของระบบ หน่วยความจำในส่วนนี้จะถูกเรียกใช้งานเป็นอันดับแรกเมื่อระบบเริ่มทำงาน

3. ซีเรียลดาต้าแฟลช (Serial DataFlash)

หน่วยความจำแบบแฟลชตัวนี้จะทำหน้าที่เป็นสตอเรท (Storage) สำหรับใช้ในการเก็บข้อมูลต่างๆ ของระบบที่ไม่ต้องการใช้เกิดการสูญหายในขณะที่บอร์ดคอม86 ขาดกระแสไฟฟ้าหล่อเลี้ยงเช่น ข้อมูลจากการตรวจวัดต่างๆ, โปรแกรมแอพพลิเคชั่นต่างๆ รวมไปถึงระบบปฏิบัติการด้วย ซึ่งบนบอร์ด คอม86 นี้ได้ติดตั้ง ดาต้าแฟลช (IC U12) เบอร์ เอที45ดีบี041ขนาด 512 กิโลไบต์สำหรับใช้งานในส่วนนี้ และได้ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

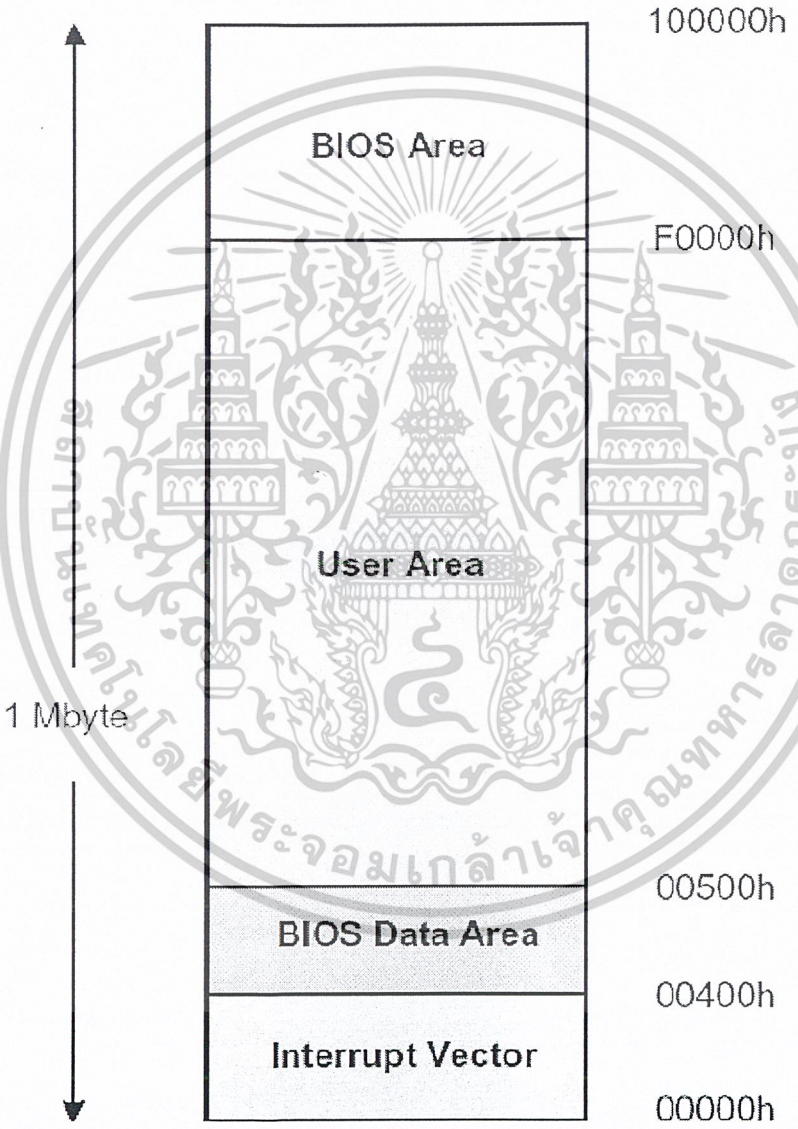
ให้สามารถขยายขนาดได้ตามความต้องการ ซึ่งในการใช้งานนักพัฒนาจะมองหน่วยความจำในส่วนนี้เปรียบเสมือนดิสก์ไดรฟ์อันหนึ่ง (ไดรฟ์ A) ของเครื่องคอมพิวเตอร์

4. แผนผังหน่วยความจำ

จากหน่วยจำทั้งหมดที่กล่าวมาในข้างต้นสามารถแสดงเป็นภาพการจัดวางของหน่วยความจำทั้งหมดได้ ดังนี้คือ

- **หน่วยความจำหลัก**

หน่วยความจำในส่วนนี้มีขนาด 1 เมกกะไบต์ สำหรับเป็นพื้นที่ใช้งานของโปรแกรมต่างๆซึ่งมีการจัดแบ่งดังนี้



รูปที่ 2-7 การแบ่งการใช้งานหน่วยความจำหลัก

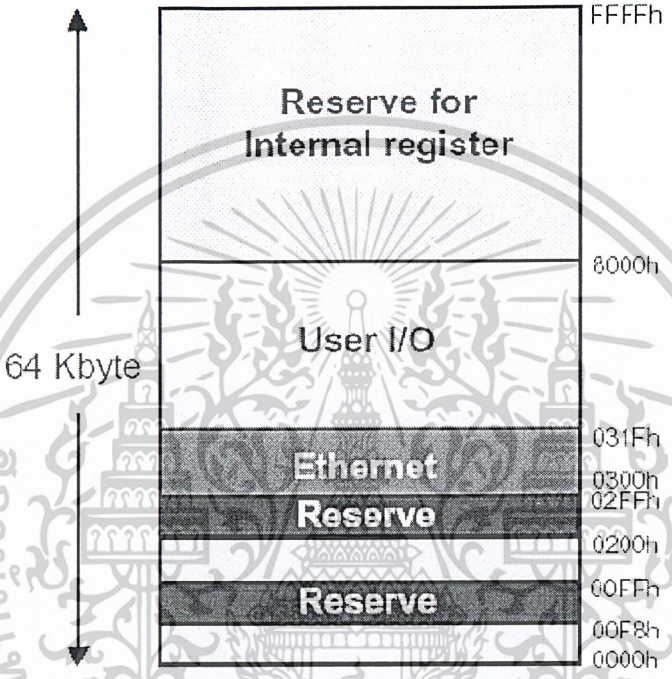
จากภาพหน่วยความจำช่วงแรกคือจากตำแหน่ง 00000h จนถึงตำแหน่ง 003FFh เป็นส่วนของอินเทอร์รัพเวกเตอร์เทเบิล (Interrupt Vector Table) สำหรับรองรับการทำงานของการอินเทอร์รัพ (Interrupt) ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนต่อมาคือช่วงตำแหน่ง 00400h ถึง 004FFh หน่วยความจำ ส่วนนี้ใช้สำหรับเก็บข้อมูลของไบออส (BIOS) ส่วนถัดมาคือช่วงตำแหน่ง 00500h ถึง F0000h ส่วนนี้จะเป็นพื้นที่ สำหรับใช้งาน โดยโปรแกรมทั่วไปทั้งระบบปฏิบัติการและโปรแกรมของนักพัฒนาที่สร้างขึ้นมา ส่วนที่เหลือในช่วงตำแหน่ง F0000h ถึง FFFFFh ส่วนนี้จะเป็นพื้นที่ที่ถูกสงวนไว้สำหรับไบออสของบอร์ดคอม86

- **หน่วยความจำที่เป็นส่วนอินพุท/เอาท์พุท (I/O)**

ส่วนนี้คือพื้นที่สำหรับใช้ในการติดต่อกับอุปกรณ์อินพุท/เอาท์พุทต่างๆ และใช้ในการติดต่อกับรีจิสเตอร์ภายในของไมโครคอนโทรลเลอร์ ซึ่งมีการจัดวางดังภาพ



รูปที่ 2-8 การจัดวางหน่วยความจำในส่วนของอินพุตท์เอาท์พุต

จากภาพจะแบ่งตำแหน่งของหน่วยความจำได้ออกเป็น 2 ส่วนหลักคือส่วนบนตั้งแต่ตำแหน่งที่ 8000h ถึงตำแหน่ง FFFFh ส่วนนี้จะเป็นพื้นที่สำหรับการใช้งานโดยตัวบอร์ดคอม86 และเป็นส่วนที่ใช้ในการติดต่อรีจิสเตอร์ต่างๆของไมโครคอนโทรลเลอร์ซึ่งบริเวณนี้จะถูกสงวนไว้ไม่สามารถนำมาใช้งานได้ ส่วนที่สองคือส่วนของหน่วยความจำส่วนล่างคือ ตั้งแต่ตำแหน่ง 0000h ถึงตำแหน่ง7FFFh บริเวณนี้จะเป็นพื้นที่สำหรับให้นักพัฒนานำส่วนที่ว่างไปใช้งานโดยเว้นส่วนที่สงวนไว้ใช้งานสำหรับตัวไมโครคอนโทรลเลอร์คือ ตำแหน่ง 00F8h ถึงตำแหน่ง 00FFh และส่วนที่สงวนไว้สำหรับการใช้งานของวงจรอีเทอร์เน็ทคือในตำแหน่ง 0300h จนถึง 031Fh บริเวณที่เหลือนอกจากที่กล่าวมานักพัฒนาสามารถนำไปใช้งานได้ตามต้องการ

2.3.2.3 ยูเอสบี (USB)

บอร์ดคอม86 นั้น มีการสื่อสารกับภายนอกได้อีกรูปแบบหนึ่งคือในรูปแบบของยูเอสบี พอร์ต โดยภายในไมโครคอนโทรลเลอร์ AM186CC ของบอร์ดคอม86นั้นมีส่วนของยูเอสบีคอนโทรลเลอร์ (USB Controller) ที่เป็นยูเอสบีดีไวซ์คอนโทรลเลอร์ (USB device controller) ภายในไมโครคอนโทรลเลอร์และยู

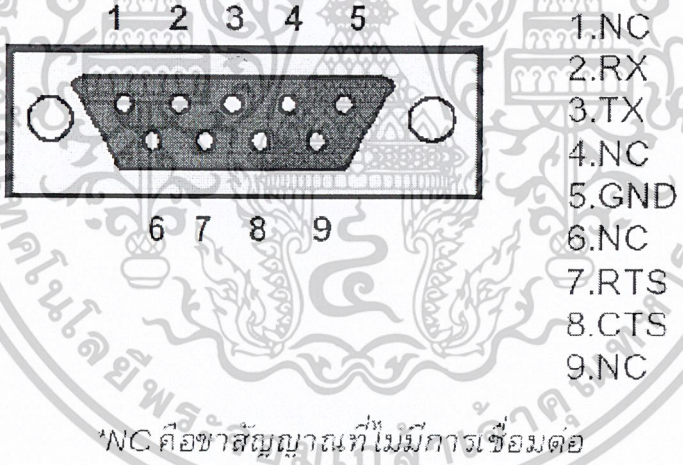
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เอสบีทรานส์ซิพเวอร์ (USB Transceiver) อยู่ภายในบอร์ดซึ่งนักพัฒนาสามารถเขียนโปรแกรมจำลองตัวบอร์ดคอม86 ให้สามารถทำตัวเองเป็นอุปกรณ์ยูเอสบีชนิดหนึ่งได้ทันที โดยบนบอร์ดคอม86 จะมีคอนเนคเตอร์ยูเอสบี ชนิดไทพ์บี (Type B (J7)) ติดตั้งอยู่บนบอร์ดอยู่แล้วซึ่งนักพัฒนาเพียงแต่เขียนโปรแกรมเพิ่มเติมก็จะสามารถนำบอร์ดไปใช้งานเป็นอุปกรณ์ยูเอสบีได้ในทันทีซึ่งบนยูเอสบีคอนโทรลเลอร์ (USBController) บนบอร์ดคอม86 นี้สนับสนุนการทำงานในฟูลโหมดสปีด (Mode Full Speed) คือที่อัตราเร็ว 12 เมกกะบิตต่อวินาที (Mbit/s) ในการรับส่งข้อมูล

2.3.2.4 พอร์ตอนุกรม

บนบอร์ดคอม86 มีพอร์ตการสื่อสารแบบอนุกรมตามมาตรฐาน อาร์เอส-232 สำหรับการนำไปใช้งานจำนวน 2 พอร์ตด้วยกัน ซึ่งอยู่ในรูปแบบของคอนเนคเตอร์แบบ ดีบี9 บนบอร์ด (J8, J9) ซึ่งพอร์ตอนุกรม ทั้ง 2 นี้เชื่อมต่อมาจากวงจรยูอาร์ต ที่อยู่ภายในไมโครคอนโทรลเลอร์ AM186CC และทำการปรับแรงดันจาก 5 โวลท์ให้เข้ากับมาตรฐาน อาร์เอส232 โดยชิพ อาร์เอส232 ไดร์เวอร์ (U9, U10) ซึ่งพอร์ตอนุกรมทั้งสองพอร์ตนี้ พอร์ตคอม1 จะถูกนำไปใช้งานสำหรับเป็นซีเรียลคอนโซล (Serial Console) สำหรับใช้ในการแสดงผลควบคุมการทำงานต่างๆ ของบอร์ดในขณะพัฒนาโปรแกรม ส่วนพอร์ต คอม2 นั้นสามารถนำไปใช้งานได้โดยทั่วไป ซึ่งการเชื่อมต่อสัญญาณต่างๆ ของพอร์ตอนุกรมทั้งสองนี้เป็นไปดังภาพ



รูปที่ 2-9 คอนเนคเตอร์ของพอร์ตอนุกรม

สำหรับในการพัฒนานั้น ภายในชุดพัฒนาคอม86 จะมีสายเชื่อมต่อระหว่างพอร์ตคอม1 ที่ทำหน้าที่เป็นคอนโซล (Console) กับคอมพิวเตอร์ซึ่งจะเป็นสายชนิดนูโมเด็ม (null modem) สำหรับใช้ในการแสดงผลและรับส่งข้อมูลต่างๆ ในระหว่างการพัฒนาโปรแกรม

2.3.2.5 ระบบรีเซ็ต

ภายในบอร์ดคอม86 จะมีวงจรสำหรับสร้างสัญญาณรีเซ็ตให้กับอุปกรณ์ต่างๆ บนบอร์ดโดยใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
55085

ชิพเอลเอ็ม3722 (U15) เป็นตัวควบคุมการทำงานโดยวงจรจะสร้างสัญญาณรีเซ็ตเมื่อมีการกดสวิทช์รีเซ็ต (SW1) บนบอร์ดหรือในกรณีที่แรงดันไฟฟ้าบนบอร์ดในส่วนของแหล่งจ่ายแรงดัน 3.3 โวลต์มีแรงดันต่ำกว่า 3.08 โวลต์

2.3.2.6 อีเทอร์เน็ท

การเชื่อมต่อแบบอีเทอร์เนตกับภายนอกนั้นเป็นรูปแบบหนึ่งที่ชุดพัฒนาคอม86 สนับสนุนการทำงาน ซึ่งภายในบอร์ดคอม86 ได้มีการออกแบบติดตั้งทั้งตัวชิพอีเทอร์เน็ทคอนโทรลเลอร์ (U6) ซึ่งเป็นชิพอีเทอร์เน็ทที่สนับสนุนการเชื่อมต่อที่ความเร็วขนาด 10 เมกกะบิตต่อวินาที (Mbps) พร้อมทั้งติดตั้ง อาร์เจ45 คอนเน็คเตอร์ (J2) บนบอร์ดคอม86 ซึ่งมีการเชื่อมต่อสัญญาณตามมาตรฐานทั่วไปโดยนักพัฒนาสามารถใช้สายแลนชนิดแบบครอส (twisted pair cross cable) สำหรับเชื่อมต่อระหว่าง PC กับบอร์ดคอม86 โดยตรงหรือใช้สายแลนชนิดแบบสายตรง (straight-through twisted pair cable) ในกรณีที่ต่อผ่านฮับ (Hub) หรือ เน็ตเวิร์คสวิทช์ (Network Switch) สำหรับให้นักพัฒนาเชื่อมต่อใช้งานได้ทันที ซึ่งตามมาตรฐาน IEEE 802.3 นั้นการเชื่อมระหว่างฮับกับบอร์ดคอม86 สายที่ใช้ควรมีระยะทางไม่เกิน 100 เมตร

2.3.2.7 ยูอาร์ต UART (Universal Asynchronous Receiver - Transmitter)

เป็น ส่วนประกอบที่ใช้สำหรับการสื่อสารระหว่าง อุปกรณ์อินพุตและเอาท์พุต ที่มีการส่งข้อมูลเป็นแบบ อนุกรมซึ่ง ยูอาร์ตจะควบคุมอัตราการส่งข้อมูลที่ทั้งสองฝั่งของการรับส่ง เพื่อไม่ให้เกิดปัญหาเมื่อ อัตราเร็วในการรับส่งข้อมูลทั้ง 2 ด้านไม่เท่ากัน

2.3.2.8 พาวเวอร์ซัพพลาย (Power supply)

ระบบจ่ายไฟของบอร์ดคอม86 แบ่งได้ออกเป็น 2 ส่วนหลักด้วยกันคือ ในส่วนแรกจะเป็นระบบจ่ายไฟแรงดัน 5 โวลท์ (Volt) ซึ่งจะมีไอซีแอลดีโอ LDO (U14) ทำหน้าที่จ่ายกระแสไฟฟ้าสำหรับเลี้ยงอุปกรณ์บนบอร์ดที่ต้องการแรงดันไฟฟ้าในการทำงานที่ 5 โวลท์ เช่น ดีแรม, แฟลชไบออส, เรียลไทม์คล็อก เป็นต้น และสำหรับอุปกรณ์ที่ต้องการแรงดันไฟฟ้า 3.3 โวลท์ ในการทำงานนั้นจะมีไอซีแอลดีโอ LDO (U16) ทำหน้าที่จ่ายกระแสไฟฟ้าให้ซึ่งอุปกรณ์เหล่านี้ได้แก่ตัวไมโครคอนโทรลเลอร์, แฟลชดิสก์เป็นต้น ซึ่งอินพุทของภาคพาวเวอร์ซัพพลายของบอร์ดคอม86 นั้นจะรับเป็นไฟฟ้ากระแสตรงที่แรงดันประมาณ 8 โวลท์ และทำการลดระดับแรงดันลงเป็น 5 โวลท์และ 3.3 โวลท์ตามลำดับ โดยภายในชุดพัฒนาจะมีอแดปเตอร์ให้ 1 ตัวสำหรับเป็นส่วนจ่ายกระแสไฟฟ้าให้กับวงจรทั้งหมด

โดยทั่วไปส่วนวงจรจ่ายกระแสไฟฟ้าแรงดัน 5 โวลท์และ 3.3 โวลท์จะสามารถจ่ายกระแสไฟฟ้าให้กับวงจรที่ต่อเพิ่มเติมได้อีก 300 มิลลิแอมป์ (mA) หากมีการต่อวงจรส่วนขยายที่ต้องการใช้งานกระแสไฟฟ้ามากกว่านี้ แนะนำให้ควรมีการต่อวงจรพาวเวอร์ซัพพลายจากภายนอกเพิ่มเติมเพื่อป้องการเสียหายของ ส่วนวงจรภาคจ่ายไฟของบอร์ดคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.9 แอลอีดีอินดิเคเตอร์ (LED Indicator)

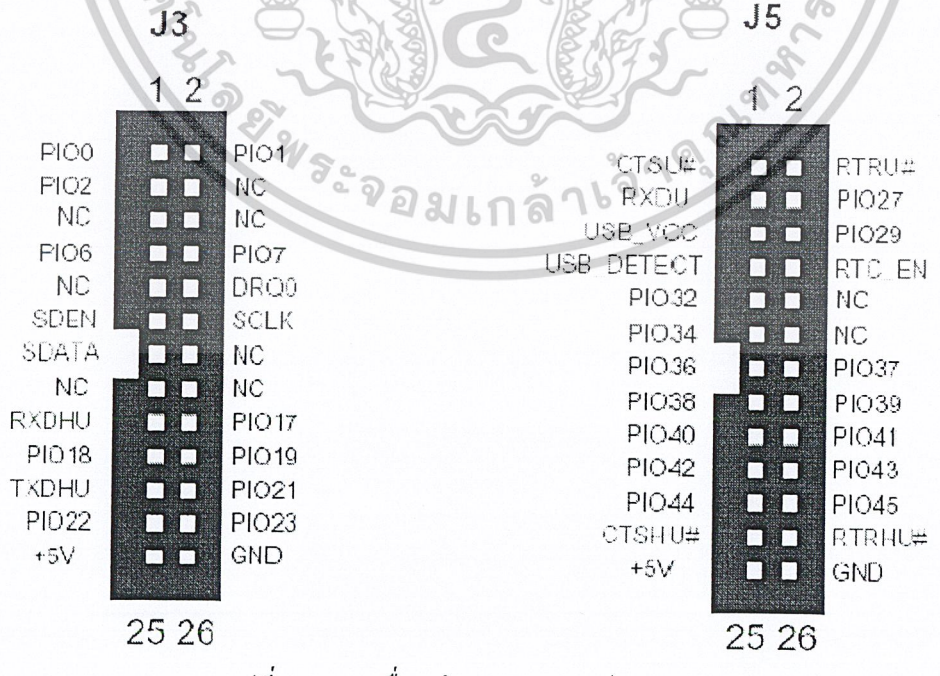
บนบอร์ดคอม86 จะมีแอลอีดี (LED) สำหรับแสดงสถานะต่างๆดังนี้คือ ดวงแรกเป็นแอลอีดีของพาวเวอร์ซัพพลาย (D4) มีสีแดงโดยจะติดสว่าง เมื่อบอร์ดคอม86 ได้รับการจ่ายกระแสไฟฟ้าจากภายนอก และวงจรพาวเวอร์ซัพพลาย 5 โวลต์ และ 3 โวลต์ สามารถทำงานได้อย่างถูกต้อง ดวงที่ 2 คือ LED สีเขียวของส่วนวงจรอีเทอร์เน็ท(D2) ซึ่งดวงนี้จะติดสว่างขึ้นเพื่อแสดงว่าการมีการเชื่อมต่อสายอีเทอร์เน็ทและพร้อมที่จะรับส่ง ข้อมูล ดวงที่ 3 คือ Lแอลอีดีดวงสีเขียวของส่วนวงจรอีเทอร์เน็ทที่เหลืออีกดวงหนึ่ง (D1) ซึ่งดวงนี้จะติดกระพริบเมื่อมีการรับส่งข้อมูลผ่านทางอีเทอร์เน็ทพอร์ต

2.3.2.10 เอ็กแพลนชั่นพอร์ต (Expansion Port)

ในการใช้งานชุดพัฒนาต่างๆสำหรับพัฒนาอุปกรณ์ทางด้านระบบฝังตัวนั้น เป็นธรรมดาที่นักพัฒนาย่อมอาจเกิดความต้องการที่จะออกแบบวงจรส่วนขยายต่ออุปกรณ์ต่างๆ เพิ่มเติมจากชุดพัฒนาพื้นฐานที่มีใช้งานอยู่ ซึ่งสำหรับบอร์ดคอม86 นี้ได้จัดเตรียมออกแบบการเชื่อมต่อใช้งานอุปกรณ์ภายนอกในรูปแบบต่างๆ ดังนี้

1. พีไอโอ (PIO)

ในส่วนของพีไอโอนี้จะเป็นลักษณะการต่อใช้งานเหมือนกับที่ใช้งานภายในไมโครคอนโทรลเลอร์โดยทั่วไปซึ่งสามารถโปรแกรมบังคับควบคุมการใช้งานได้เป็นอิสระจากซึ่งกันและกันโดยบนบอร์ดคอม 86 นี้ มีพีไอโอให้นักพัฒนาได้ใช้งานทั้งสิ้นจำนวน 25 แชนแนล (Channel) จากทั้งหมด 48 แชนแนลที่มีในไมโครคอนโทรลเลอร์ เนื่องจากการที่สัญญาณพีไอโอบางสัญญาณถูกนำไปใช้งานในลักษณะของการมัลติเพล็กกับสัญญาณอื่นของบอร์ดคอม86 ซึ่งพีไอโอที่ใช้งานได้บนบอร์ดคอม86 และการเชื่อมต่อมายังคอนเน็คเตอร์ของพีไอโอต่างๆ เป็นไปดังภาพ



รูปที่ 2-10 การเชื่อมต่อ PIO ของบอร์ดคอม 86

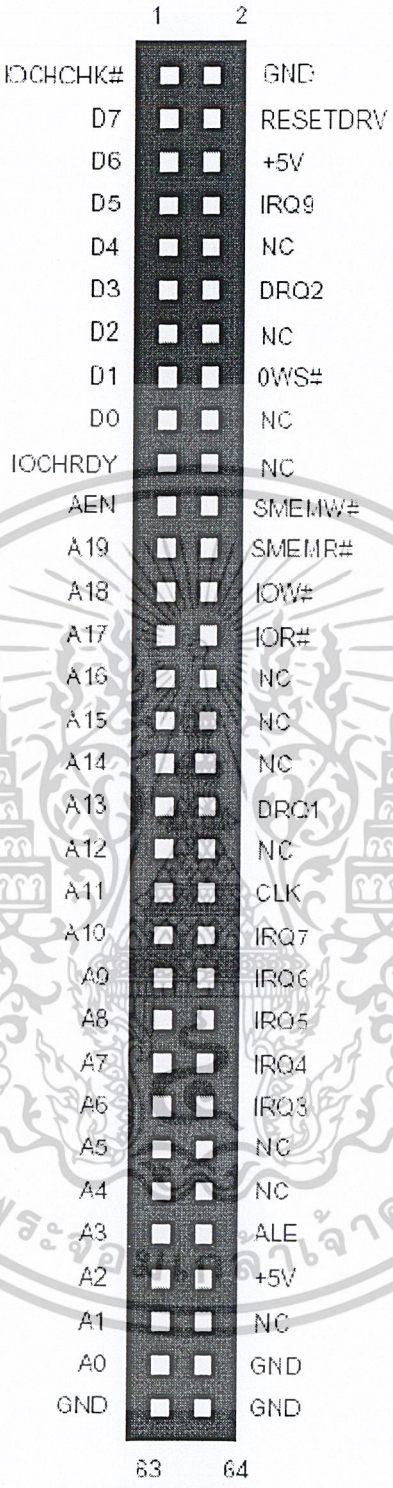
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพในส่วนที่เป็นชื่อของสัญญาณอื่นที่มิใช่พีไอโอหมายเลขต่างๆ นั้นคือ พีไอโอ ที่ถูกใช้งานในรูปแบบของสัญญาณอื่นที่มัลติเพล็กซ์ (Multiplex) อยู่และสามารถนำมาใช้งานเป็นพีไอโอได้ในกรณีที่ไม่ต้องการใช้งาน เช่นขา CTSHU# ถ้าไม่ต้องการใช้งานส่วนของ โฟลว์คอนโทรล (Flow Control) ของพอร์ต คอม1 ก็สามารถนำมาใช้งานเป็น พีไอโอ46 ได้ เอ็นซี คือขาพีไอโอที่ถูกนำไปใช้ในลักษณะของสัญญาณอื่นที่มัลติเพล็กซ์อยู่และไม่สามารถนำมาใช้งานได้ ซึ่งไม่มีการนำมาเชื่อมต่อให้ไว้ที่คอนเน็คเตอร์ส่วนขาที่เหลือจะเป็นพีไอโอที่สามารถใช้งานได้

2. ไอซ่า (ISA)

เป็นรูปแบบของการเชื่อมต่อกับอุปกรณ์แบบขนาน โดยได้ออกแบบให้มีลักษณะการเชื่อมต่อใช้งานคล้ายกับการเชื่อมต่อใช้งานของไอซ่าบัส (ISA Bus) ของคอมพิวเตอร์โดยทั่วไปแตกต่างกันเพียงรูปแบบการจัดวางอุปกรณ์ต่างๆ ที่พยามให้จัดวางในรูปลักษณะของมาตรฐาน พีซี/104 ซึ่งการออกแบบวงจรต่างๆ ที่จะนำมาเชื่อมต่อนั้นสามารถออกแบบในลักษณะเดียวกันกับการออกแบบการ์ดไอซ่าของเครื่องคอมพิวเตอร์ โดยทั่วไปจะมีคอนเนคเตอร์ เจ4 สำหรับใช้ในการเชื่อมต่อกับการ์ดขยายภายนอกที่จะนำมาเชื่อมต่อ ซึ่งการเชื่อมต่อของสัญญาณต่างๆ ของคอนเน็คเตอร์ เจ4 จะเป็นไปดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



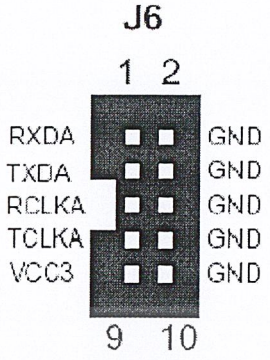
รูปที่ 2-11 ไอซ่าคอนเนคเตอร์ (ISA Connector)

3. จีซีไอ (GCI)

จีซีไอเป็นมาตรฐานการสื่อสารแบบอนุกรมสำหรับใช้ในการเชื่อมต่อวงจรทางด้านการสื่อสารต่างๆ ซึ่งมาตรฐานครอบคลุมถึงการเชื่อมต่อของไลน์การ์ด (Linecard), เอนที1 หรือ เทอร์มินัลอาร์คิเทกเจอร์ (Terminal Architecture) ของไอเอสดีเอน (ISDN) แอพพลิเคชั่นซึ่งบอร์ดคอม86 นี้สนับสนุนรองรับการทำงานในส่วนของมาตรฐานการเชื่อมต่อแบบเทอร์มินัลเวอร์ชั่น (Terminal Version) ของจีซีไอสแตนดาร์ด (GCI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Standard) ซึ่งมีการเชื่อมต่อกับภายนอกผ่านทางพอร์ตจีซีไอ (J6 ของบอร์ดคอม86) โดยมีการเชื่อมต่อของสัญญาณต่างๆ ดังภาพ



รูปที่ 2-12 จีซีไอ

2.3.3 สถาปัตยกรรมทางด้านซอฟต์แวร์

1. BIOS เป็นโปรแกรมที่คอยควบคุมการทำงานของฮาร์ดแวร์ต่างๆบนบอร์ดคอม86 เป็นส่วนที่จัดการการสื่อสารระหว่างฮาร์ดแวร์กับแอพพลิเคชันซอพท์แวร์ และระบบปฏิบัติการ

2. Driver เป็นโปรแกรมขนาดเล็กที่ทำหน้าที่ติดต่อสื่อสารกับฮาร์ดแวร์ อื่นๆที่ไบออสไม่ได้รองรับการทำงานไว้ เช่น ฮาร์ดแวร์ที่สร้างขึ้นมาเพิ่มเติมเพื่อเชื่อมต่อเข้ากับบอร์ดคอม86

3. Operating System ระบบปฏิบัติการเป็นโปรแกรมที่ควบคุมการทำงานของโปรแกรมต่างๆในระบบ ทั้งในส่วนของการจัดการหน่วยความจำ เป็นส่วนสื่อสารระหว่างโปรแกรมแอพพลิเคชันของผู้ใช้และส่วนของฮาร์ดแวร์ระบบต่างๆ

4. Packet Driver เป็นโปรแกรมจัดการเกี่ยวกับแพ็กเก็ตของการส่งข้อมูล สำหรับแอพพลิเคชันทางด้านการสื่อสาร

5. TCP/IP เป็นโปรแกรมจัดการทางด้านโพรโตคอลทีซีพี/ไอพีสำหรับแอพลิเคชันทางด้านการสื่อสาร

6. Network Application แอพพลิเคชั่นทางด้านการสื่อสาร เป็นโปรแกรมแอพพลิเคชันทางด้านการสื่อสารต่างๆ เช่น โปรแกรมเว็บเซิร์ฟเวอร์ในระบบฝังตัว (Embedded web server) เป็นต้น

7. Application เป็นโปรแกรมแอพลิเคชันอื่นๆ ที่ไม่ได้ใช้ความสามารถทางด้านการสื่อสาร เช่นโปรแกรมควบคุมการทำงานของหุ่นยนต์ หรือเครื่องจักร เป็นต้น

2.3.4 จุดเด่นของคอม86

2.3.4.1 ใช้ไมโครคอนโทรลเลอร์ที่มีความสามารถสูง

ภายในชุดพัฒนาคอม86 มีบอร์ดคอม86 ซึ่งได้เลือกใช้ไมโครคอนโทรลเลอร์ Am186CC ของบริษัทเอเอ็มดี (AMD) ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 16 บิตทำงานอยู่ที่ความเร็ว 24 เมกกะเฮิร์ต (MHz) ทำให้มีความสามารถในการประมวลผลต่างๆเพิ่มมากขึ้นเมื่อเทียบกับการพัฒนาโดยใช้ไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่นิยมใช้กันอยู่ในปัจจุบัน นอกจากนี้ภายในตัวไมโครคอนโทรลเลอร์เองยังได้รวมเอาอุปกรณ์พื้นฐานต่างๆที่จำเป็นในการใช้งาน เช่น ดีแรมคอมโทรลเลอร์ (DRAM controller), อินเทอร์รัพคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

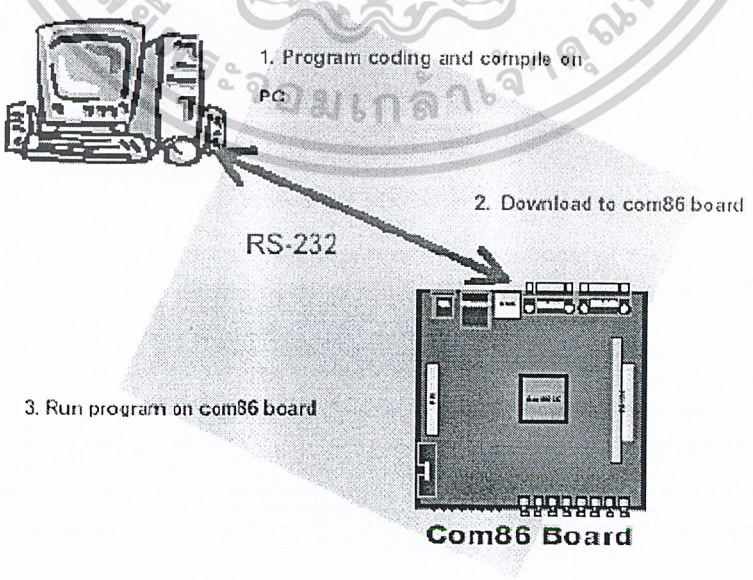
(Interrupt controller), ยูอาร์ต (UART) เข้าไว้ภายในตัวไมโครคอนโทรลเลอร์ซึ่งทำให้ง่ายต่อการออกแบบอุปกรณ์เพื่อทำเป็นสินค้าภายหลังการพัฒนาตัวอย่างเช่น การนำบอร์ดคอม86 ไปพัฒนาเป็นอุปกรณ์อย่างหนึ่งที่ต้องการควบคุมการทำงานผ่านทางยูเอสบีพอร์ต (USB port) จะสามารถทำได้โดยง่ายเนื่องจากไม่ต้องการออกแบบวงจรควบคุมการทำงานในส่วนยูเอสบีนี้เพิ่มอีก เพียงแต่ออกแบบวงจรการทำงานส่วนอื่นที่ต้องการเพิ่มเติมทำให้ทุ่นเวลาในการพัฒนา เป็นต้น นอกจากนี้ภายในไมโครคอนโทรลเลอร์ยังเพิ่มส่วนโมดูล (Module) ทางด้านการสื่อสารต่างๆ เข้าไปภายในตัว ทำให้เป็นไมโครคอนโทรลเลอร์ที่มีความสามารถเด่นทางด้านการสื่อสารเหมาะกับการพัฒนาอุปกรณ์ที่ต้องใช้คุณสมบัติการสื่อสารต่างๆ ซึ่งจะสามารถทำได้ง่ายเมื่อเทียบกับการใช้ไมโครคอนโทรลเลอร์ขนาดเล็กแบบเดิมซึ่งมีข้อจำกัดต่างๆ อยู่มากมาย

2.3.4.2 มีเครื่องมือในการพัฒนาโปรแกรมมาก

เนื่องจากการพัฒนาโปรแกรมโดยใช้ชุดพัฒนาคอม86 นั้นเป็นเสมือนกับการพัฒนาโปรแกรมบนของคอมพิวเตอร์ส่วนบุคคลทั่วไป ดังนั้นจึงสามารถนำโปรแกรมเครื่องมือต่างๆที่ใช้พัฒนาโปรแกรมสำหรับคอมพิวเตอร์ส่วนบุคคลมาใช้งานได้มากมาย และโปรแกรมเครื่องมือเหล่านี้ก็เป็นที่คุ้นเคยของนักพัฒนาจึงทำให้สามารถพัฒนาอุปกรณ์ได้อย่างรวดเร็ว ลดต้นทุนในการพัฒนาต่างๆได้เป็นจำนวนมากนอกจากนี้ภายในชุดพัฒนายังมีชุดซอฟต์แวร์เครื่องมือต่างๆที่จำเป็นสำหรับช่วยในการพัฒนาโปรแกรมด้วยอีกทางหนึ่ง

2.3.4.3 ง่ายในการเรียนรู้และพัฒนา

บอร์ดคอม86 ภายในชุดพัฒนาคอม86 นี้มีสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบ x86 ซึ่งเป็นสถาปัตยกรรมเดียวกับคอมพิวเตอร์ส่วนบุคคลทั่วไป โดยมีชุดคำสั่งที่เข้ากันได้กับชุดคำสั่งของ 80286 (Real mode) ทำให้ง่ายต่อการเรียนรู้ในการพัฒนาโปรแกรมต่างๆ และยังสามารถพัฒนาและทดสอบซอฟต์แวร์ต่างๆได้บนคอมพิวเตอร์ ก่อนที่จะทดสอบกับบอร์ดคอม86 ทำให้สามารถพัฒนาอุปกรณ์ทางด้านระบบฝังตัวต่างๆได้อย่างรวดเร็ว ซึ่งรูปแบบการพัฒนามีลักษณะเป็นไปดังภาพ

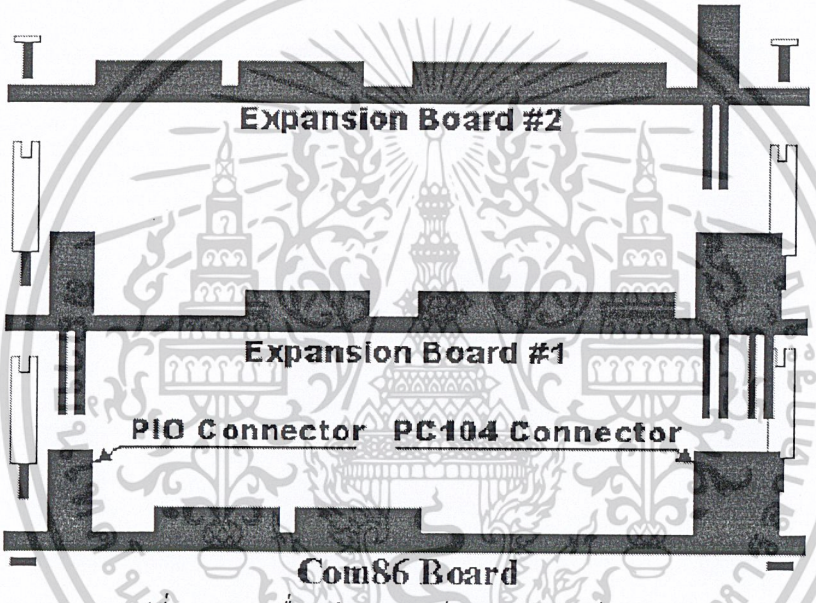


รูปที่ 2-13 รูปแบบการพัฒนาโปรแกรมโดยใช้คอม 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.4 มีความสามารถในการเพิ่มขยายได้

ในส่วนของการเพิ่มขยายของฮาร์ดแวร์ (Hardware) สามารถทำได้โดยผ่านทางคอนเนคเตอร์ (Connector) ของบอร์ดในรูปแบบของมาตรฐาน (พีซีหนึ่งศูนย์สี่) PC/104 ซึ่งจะมีการเชื่อมต่อแบบบัสอินเทอร์เฟส (BUS Interface) แบบไอเอสเอบัส (ISA BUS) ขนาด 8 บิต คล้ายกับไอเอสเอบัส ที่มีอยู่ในคอมพิวเตอร์ส่วนบุคคลทั่วไป ซึ่งผู้พัฒนาสามารถออกแบบฮาร์ดแวร์ให้เป็นลักษณะของบอร์ดขยายและนำมาเชื่อมต่อกับบอร์ดคอม86 และเพียงเขียนซอฟต์แวร์ไดร์เวอร์ (Software Driver) สำหรับควบคุมการทำงานของส่วนเพิ่มเข้ามาใหม่ซึ่งจะทำงานร่วมกับระบบปฏิบัติการในการรองรับการทำงานของแอพพลิเคชันต่างๆที่เพิ่มเข้ามา นอกจากนี้ตัวบอร์ดคอม86 ยังสามารถเชื่อมต่อกับอุปกรณ์ภายนอกในลักษณะของพีไอโอ (PIO) หรือโปรแกรมเมเบิลอินพุท/เอาท์พุท (Programmable Input/Output) เหมือนกันการใช้งานไมโครคอนโทลเลอร์รุ่นอื่นๆได้ด้วยเช่นกัน



รูปที่ 2-14 การเชื่อมต่อของบอร์ดขยายกับบอร์ดคอม 86

2.3.4.5 แอพพลิเคชันเป้าหมายของชุดพัฒนา (Target Application)

จากความสามารถของไมโครคอนโทรลเลอร์ของบอร์ดคอม86 ทำให้การประยุกต์ใช้งานกับแอพลิเคชั่นต่างๆนั้น สามารถทำได้มากมาย โดยตัวอย่างแอพพลิเคชันต่างๆที่เหมาะแก่การพัฒนาโดยบอร์ดคอม 86 มีดังนี้

1. เกี่ยวกับอุตสาหกรรม (Industrial control and Automation)

ตัวอย่างของอุปกรณ์ควบคุมภายในโรงงาน เช่น การควบคุมเครื่องจักรจากระยะไกล การตรวจวัดสภาวะต่างๆ ภายในโรงงานและส่งกลับไปยังศูนย์ควบคุมโดยการใช้บอร์ดคอม86 เป็นอุปกรณ์ที่ช่วยทำหน้าที่ในการสื่อสาร ควบคุมสภาวะแวดล้อม หรือเครื่องจักรต่างๆภายในโรงงานและรายงานกลับไปยังศูนย์กลางตามกำหนดเวลา เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อุปกรณ์ยูเอสบี (USB Peripheral)

อุปกรณ์ยูเอสบีต่างๆ สามารถพัฒนาได้โดยการใช้บอร์ดคอม86 ซึ่งภายในบอร์ดคอม86 นี้มียูเอสบีคอนโทรลเลอร์ภายในตัวไมโครคอนโทรลเลอร์ Am186CC สามารถเชื่อมต่อเข้ากับคอมพิวเตอร์ได้ที่ความเร็ว 12 เมกกะบิตต่อวินาที (Mbps) ซึ่งการพัฒนาเป็นอุปกรณ์ยูเอสบี ต่างๆ โดยบอร์ดคอม86 นี้สามารถทำได้โดยไม่ต้องมีการต่อวงจรทางด้านยูเอสบีใดๆ เพิ่มเติมอีก

3. เว็บเซิร์ฟเวอร์ (Embedded web server)

เอ็มเบดเดดเว็บเซิร์ฟเวอร์ (Embedded web server) สามารถสร้างได้บนบอร์ดคอม86 โดยไม่จำเป็นต้องเชื่อมต่อวงจรอื่นๆเพิ่มเติม ซึ่งบอร์ดคอม86 นี้สามารถสื่อสารกับภายนอกผ่านทางอีเทอร์เน็ทพอร์ตที่อยู่บนบอร์ดที่ความเร็ว 10 เมกกะบิตเปอร์เซ็ก ทำให้สามารถนำไปประยุกต์ใช้งานในกาออกแบบอุปกรณ์ควบคุมต่างๆ ผ่าน เอ็มเบดเดดเว็บเซิร์ฟเวอร์ ที่ทำงานอยู่บนบอร์ดคอม86 ได้โดยง่าย

4. โรโบติก (Robotics)

บอร์ดคอม86 มีไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพสูงในการทำงาน การพัฒนาซอฟต์แวร์สามารถพัฒนาได้ซับซ้อนมากยิ่งขึ้น การเชื่อมต่อขยายทางด้านฮาร์ดแวร์ต่างๆสามารถทำได้โดยง่าย เหมาะสำหรับการพัฒนาหุ่นยนต์ที่มีการทำงานที่ซับซ้อน

5. การควบคุมบ้านหรือสิ่งก่อสร้าง (Home and building control)

อุปกรณ์สำหรับควบคุมระบบต่างๆภายในบ้าน หรืออาคารสามารถนำบอร์ดคอม86 ไปประยุกต์ใช้งานได้เช่นกัน โดยใช้เป็นส่วนในการควบคุมการทำงานของอุปกรณ์ต่างๆ หรือคอยตรวจสอบการทำงานของเซ็นเซอร์ (Senser) ต่างๆ ภายในบ้าน และแจ้งไปยังเจ้าของบ้านเมื่อเกิดเหตุร้ายขึ้น นอกจากนี้ตัวไมโครคอนโทรลเลอร์ยังรองรับการนำไปสร้างเป็นอุปกรณ์ต่างๆ ที่เกี่ยวข้องกับงานทางด้านการสื่อสารต่างๆ เช่น ไอเอสดีเอ็นโมเด็มแอนด์เทอมินัลอแดปเตอร์ (ISDN Modem and Terminal Adaptor), โลเอนด์เราเตอร์ (Low-end Router), ไลน์การ์ดแอพพลิเคชั่น (Line card application), เอ็กซ์ดีเอสแอลแอพพลิเคชั่น (xDSL application), ดิจิตอลคอร์ดโฟน (Digital corded phone) เป็นต้น ซึ่งสามารถนำบอร์ดคอม86 ไปประยุกต์ใช้งานในการออกแบบพัฒนาได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

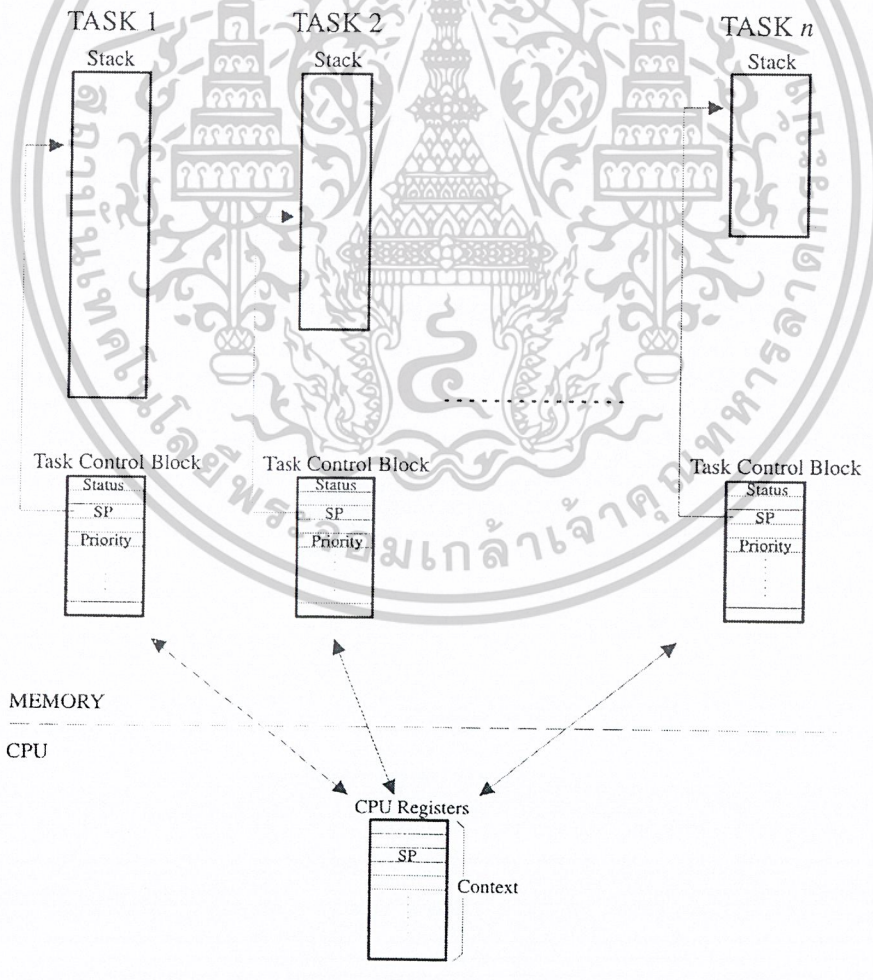
บทที่ 3
ทฤษฎีและหลักการของเทคนิคที่เกี่ยวข้อง

3.1 การเขียนโปรแกรมให้ทำงานเป็นมัลติทาสก์

เป็นโหมดการทำงานที่อนุญาตให้สามารถทำงานพร้อมๆกันได้ หรือ ประมวลผลแบบสอดแทรกกันของสองทาสก์ หรือมากกว่า คล้ายกับมัลติโปรแกรมมิ่งที่มีรูปแบบแตกต่างกัน

ในระบบปฏิบัติการดอสมีข้อจำกัดคือไม่อนุญาตโปรแกรมหลายๆ โปรแกรมทำงานพร้อมๆกันต้องทำงานให้เสร็จทีละงานเป็นลำดับไป ดังนั้นงานในการจัดการให้แต่ละโปรแกรมหรือหลายๆงานทำไปพร้อมกันจึงเป็นของผู้สร้างโปรแกรม

ในปัจจุบันมีไลบรารีที่มาช่วยจัดการให้โปรแกรมหนึ่งๆ สามารถรันหลายๆฟังก์ชันสลับกันไปมาได้หลายตัวด้วยกัน อาทิเช่น ไมโครซีโอเอส (อ่านว่า MicroC/OS) , Multi-C library เป็นต้น ซึ่งในโครงงานนี้ผู้พัฒนาได้เลือกใช้ uC/OS-II เนื่องจากไม่ขึ้นกับหน่วยประมวลผล และสนับสนุนการทำงานแบบเรียลไทม์ (real time)



รูปที่ 3-1 มัลติทาสก์กิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2

ไมโครคอนโทรลเลอร์โอเปอร์เรทติ้งซิสเต็มเวอร์ชัน 2 หรือไมโครซีโอเอส2 (µC/OS-II) ถูกสร้างขึ้นบนพื้นฐานของเรียลไทม์เคอร์เนล (Real-time kernel) ซึ่งนิยมนำไปใช้ในอุปกรณ์หลายๆอย่างด้วยกัน เช่น กล้องถ่ายรูป เน็ตเวิร์คอะเดปเตอร์ อุตสาหกรรม เป็นต้น

3.2.1 จุดเด่นของไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2

1. ซอร์ดโค้ด (Source Code)

มีการจัดระเบียบไว้เป็นอย่างดีและเข้าใจง่าย

2. พอร์ตเทเบิล (Portable)

โดยส่วนใหญ่แล้วไมโครซีโอเอส2 จะถูกเขียนโดย ANSI C ที่มีความสามารถในการพอร์ตเทเบิลสูง ด้วยการเขียนด้วยเอเอสเอ็ม (ASM) ที่ทำให้ไมโครซีโอเอส2 มีขนาดเล็กและสามารถพอร์ตไปยังชิพประมวลผลตัวอื่น ๆ ได้เช่นเดียวกันกับ ไมโครซีโอเอส ไมโครซีโอเอส2สามารถถูกพอร์ตลงไปในชิพประมวลผลตัวที่ใหญ่กว่าได้ ตราบใดที่ชิพประมวลผลนั้นๆ มีแสต็กพอยเตอร์ (stack pointer) และซีพียู (CPU register) ที่สามารถถูกพุช (push) และป๊อป (pop) ลงไปในแสต็กได้

พอร์ตทั้งหมดที่มีอยู่ในไมโครซีโอเอสสามารถถูกแปลงมาลงในไมโครซีโอเอส2 ภายในเวลาประมาณ 1 ชม. เพราะไมโครซีโอเอส2 สามารถเข้ากันได้กับไมโครซีโอเอส ทำให้โปรแกรมประยุกต์ของไมโครซีโอเอสสามารถรันบนไมโครซีโอเอส2 ได้ด้วยการเปลี่ยนแปลงที่เล็กน้อยหรือว่าไม่เปลี่ยนแปลงเลย สามารถตรวจสอบพอร์ตที่มีอยู่ในปัจจุบันของไมโครซีโอเอส2 ได้จาก web site www.uCOS-II.com

3. รอมเมเบิล (ROMable)

ไมโครซีโอเอส2ได้ถูกออกแบบสำหรับโปรแกรมประยุกต์ฝังตัวหมายความว่าถ้าเรามี proper tool chain (เช่น C computer, assembler, และ linker/locator) คุณสามารถฝัง uC/OS-II ลงไปในส่วนของชิ้นงานของคุณได้

4. แสกลเอเบิล (Scalable)

ผู้สร้างได้ออกแบบไมโครซีโอเอส2ในแบบที่เราสามารถใช้เฉพาะบริการที่เราต้องการในงานได้ หมายถึงงานของเราสามารถใช้งานบริการบนไมโครซีโอเอส2บางส่วน จนถึงใช้งานบริการทุกส่วน และยอมให้คุณลดขนาดของหน่วยความจำ (ทั้ง ROM และ RAM) ที่ถูกใช้งานโดยไมโครซีโอเอส2 บนแต่ละงานได้

5. พรีเอ็มทีพ (Preemptive)

ไมโครซีโอเอส2เป็น “fully preemptive real-time kernel” ซึ่งหมายความว่า ไมโครซีโอเอส2จะทำงานงานที่มีความสำคัญสูงที่สุดที่พร้อมทำงานก่อนเสมอ ซึ่งโดยเคอร์เนลตามท้องตลาดส่วนใหญ่แล้วจะเป็นแบบพรีเอ็มทีพและไมโครซีโอเอส2เทียบเคียงได้กับเคอร์เนลเหล่านั้น

6. มัลติทาสก์กิ้ง (Multitasking)

ไมโครซีโอเอส2สามารถจัดการทาสก์ได้ถึง 64 ทาสก์พร้อมกัน อย่างไรก็ตามซอฟแวร์รุ่นปัจจุบันได้จองทาสก์ ไว้ 8 ทาสก์เพื่อให้สำหรับระบบใช้งาน ทำให้เหลือเพียง 56 ทาสก์เท่านั้น แต่ละทาสก์จะมีเลขกำกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ไม่ซ้ำกัน ซึ่งหมายความว่าไมโครซีโอเอส2ไม่สามารถที่จะทำ round-robin scheduling ได้เหตุนี้ทำให้มี 64 ระดับความสำคัญ (priority level)

7. ดีเทอร์มินิสติก (Deterministic)

เวลาในการประมวลผลของไมโครซีโอเอส2สามารถวัดได้ นั่นหมายถึงเราสามารถวัด รู้และคำนวณเวลาที่ไมโครซีโอเอส2ใช้ในการคำนวณฟังชันก์และบริการ แต่อย่างไรก็ตามเวลาในการประมวลผลของไมโครซีโอเอส2เซอร์วิสทั้งหมดไม่ขึ้นอยู่กับจำนวนของทาสก์ในงานของเรา

8. ทาสก์แสต็ก (Task Stacks)

แต่ละทาสก์ต้องการแสต็กของตัวเอง อย่างไรก็ตามไมโครซีโอเอส2ยอมให้แต่ละทาสก์มีขนาดของแสต็กที่แตกต่างกันได้ ซึ่งสามารถช่วยให้เราลดขนาดการใช้งานบนแรมบนงานของเราได้ด้วย stack-checking ของไมโครซีโอเอส2 เราสามารถประมาณค่าของแสต็กที่เราต้องการใช้บนแต่ละทาสก์ของเราได้อย่างเท่าที่เราใช้งานจริง ๆ

9. บริการ (Services)

ไมโครซีโอเอส2 ได้เตรียมบริการต่าง ๆ เช่น mailboxes queues semaphores partition เป็นต้น

10. การจัดการอินเทอร์รัพท์ (Interrupt Management)

อินเทอร์รัพท์สามารถหยุดการทำงานของทาสก์ไว้ชั่วคราวได้ โดยที่ทาสก์ที่มีความสำคัญที่สูงกว่าจะสามารถตื่นขึ้นมาทำงานระหว่างอินเทอร์รัพท์ของทาสก์ที่มีความสำคัญต่ำกว่าได้ โดยที่อินเทอร์รัพท์สามารถถูกใช้ได้ถึง 255 ระดับซ้อนกัน

11. มีความสามารถมากและเชื่อถือได้ (Robust and Reliable) ไมโครซีโอเอส2มีพื้นฐานอยู่บน ไมโครซีโอเอส ซึ่งถูกใช้ในหลายร้อยงานทั่วโลกมาตั้งแต่ปี 1992 ซึ่งไมโครซีโอเอส2ใช้คอร์ (core) เดิมและฟังก์ชันที่ยังคงอยู่เหมือนเดิมเป็นส่วนใหญ่

3.2.2 แนวคิดแบบเรียลไทม์ซิสเต็ม

เรียลไทม์ซิสเต็มได้ถูกกำหนดลักษณะโดยผลที่ตามมาซึ่งถูกบังคับอันเนื่องมาจากทางตรรกะเช่นเดียวกันกับคุณสมบัติความถูกต้องของเวลาของระบบที่ไม่ได้เป็นดังนั้น แบ่งออกเป็น 2 ชนิดคือ ซอฟท์ (SOFT) และฮาร์ด (HARD) สำหรับ SOFT real-time system นั้นทาสก์จะถูกกระทำโดยระบบเร็วที่สุดตราบเท่าที่จะทำได้ แต่ทาสก์อาจจะไม่เสร็จได้ทันตามเวลาที่ได้กำหนดไว้ สำหรับแบบ Hard real-time system จะถูกกำหนดให้งานออกมาถูกต้องและเสร็จตามเวลาที่กำหนด เรียลไทม์ซิสเต็มโดยส่วนใหญ่แล้ว จะประกอบไปด้วยทั้งแบบ SOFT และ HARD real-time application นั้นมีครอบคลุมกว้างมาก แต่เรียลไทม์ซิสเต็มส่วนใหญ่นั้นจะเป็นแบบฝังตัว หมายความว่าคอมพิวเตอร์ที่ถูกสร้างในระบบจะไม่ถูกผู้ใช้เห็น โดยที่เรียลไทม์ซิสเต็มนั้นออกแบบได้ยากกว่าระบบที่ไม่เรียลไทม์ซิสเต็มมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 มิวเซอร์ลเอ็กซ์คูลชัน (Mutual Exclusion)

วิธีการที่ง่ายที่สุดสำหรับทาสก์ที่จะติดต่อสื่อสารกันผ่านข้อมูลที่ใช้ร่วมกัน เมื่อมีการแลกเปลี่ยนข้อมูลระหว่างกันบนข้อมูลที่ใช้ร่วมนั้น เราต้องมั่นใจว่าแต่ละทาสก์จะมีการเข้าถึงข้อมูลร่วมนั้นโดยหลีกเลี่ยงการแย่งชิงข้อมูล โดยวิธีการโดยทั่วไปที่จะเข้าถึงข้อมูลร่วมมีดังนี้

- ไม่ให้ใช้อินเทอร์รัพท์ (disable interrupts)
- ทำเทสท์และเซ็ทโอเปอร์เรชัน (performing test-and-set operations)
- ไม่ใช้มีการใช้การจัดลำดับ (disabling scheduling)
- ใช้เซมาฟอร์ (using semaphores)

3.2.4 เซมาฟอร์ (Semaphores)

เป็นโปรโตคอลสำหรับมัลติทาสก์กิ้งเคอร์เนล (multitasking kernels) โดยส่วนใหญ่ มีจุดประสงค์เพื่อ

- ควบคุมการเข้าถึงทรัพยากรร่วม
- ส่งสัญญาณของเหตุการณ์ที่เกิดขึ้น
- ยอมให้ 2 ทาสก์ทำการซิ้งค์ (sync) งานกันได้

เซมาฟอร์เป็นคุณแจที่จะทำให้โค้ดของเราเรียงลำดับการทำงาน ถ้าเซมาฟอร์กำลังถูกใช้งานอยู่ทาสก์ที่เรียกเข้ามาจะถูกพักไว้ชั่วคราวจนกระทั่งเซมาฟอร์ถูกปล่อยจากเจ้าของปัจจุบัน โดยแบ่งเซมาฟอร์ได้เป็น 2 ชนิด คือ binary semaphore และ counting semaphore ดังเช่นที่ชื่อได้แสดงไว้ binary semaphore สามารถมีค่าได้เพียงแค่ 0 หรือ 1 ส่วน counting semaphore สามารถมีค่าได้ตั้งแต่ 0 ถึง 255 , 65535 , 4294967295 แล้วแต่โครงสร้างของแต่ละเซมาฟอร์ที่เขียนโดยใช้ 8, 16, 32 บิต แต่ขนาดจริงๆ จะขึ้นอยู่กับที่เคอร์เนลที่ใช้ เมื่อมีการนำเซมาฟอร์มาใช้งาน ตัวระบบเองต้องมีการเก็บเส้นทางของทาสก์ที่ทำการรอเพื่อให้เซมาฟอร์ดำเนินการได้

โดยปกติแล้ว มีเพียงแค่ 3 โอเปอร์เรชันที่ถูกเตรียมไว้ใช้ในเซมาฟอร์ INITIALIZE (หรือเรียกว่า CREATE), WAIT (หรือเรียกว่า PEND), SIGNAL (หรือเรียกว่า POST) สำหรับค่าเริ่มต้นของเซมาฟอร์ต้องถูกเตรียมในช่วงที่เซมาฟอร์ถูก initialized และทาสก์ที่รออยู่ของเซมาฟอร์จะต้องถูกทำให้ว่าง

ทาสก์ที่ต้องการเซมาฟอร์จะเรียก WAIT แล้วถ้าเซมาฟอร์นั้นสามารถใช้งานได้ (ค่ามีค่าเกิน 0) ค่าของเซมาฟอร์จะถูกลดค่าลงและทาสก์จะสามารถทำงานได้ต่อไป แต่ถ้าค่าของเซมาฟอร์มีค่าเป็น 0 แล้วทาสก์มีการเรียก WAIT จะถูกทำการถูกนำไปไว้ใน waiting list โดยเคอร์เนล ส่วนใหญ่จะยอมให้กำหนด timeout เพื่อกำหนดเวลาที่รอเซมาฟอร์ โดยจะมีการเรียกทาสก์ขึ้นมารันและแสดงข้อผิดพลาดกลับไปยังผู้ที่เรียกทาสก์นั้น

ทาสก์ที่ปล่อย เซมาฟอร์ออกมาจะไปเรียก SIGNAL ถ้าไม่มีทาสก์อื่นรออยู่ใน waiting list ค่าของ เซมาฟอร์ก็จะเพิ่มขึ้น แต่ถ้ามีทาสก์เหลืออยู่ในลิสท์จะทำการเรียกทาสก์ในลิสท์ขึ้นมารันโดยไม่มีกี่เพิ่มค่าเซมาฟอร์ สำหรับเงื่อนไขในการหยิบทาสก์ขึ้นมาทำงานนั้น ขึ้นอยู่กับเคอร์เนลดังเช่น

- ทาสก์ที่มีระดับความสำคัญสูงสุดที่รออยู่
- ทาสก์แรกที่เข้ามารอในลิสท์ (First In First Out หรือ FIFO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การสื่อสารระหว่างทาสก์ (Intertask Communication)

บางครั้งก็เป็นความจำเป็นของทาสก์หรือไอเอสอาร์ (ISR) ที่จะติดต่อข้อมูลกับทาสก์อื่น ๆ ข้อมูลที่แลกเปลี่ยนเรียกการสื่อสารระหว่างทาสก์ โดยข้อมูลที่ถูกใช้ในการสื่อสารระหว่างทาสก์ทำได้ 2 ทาง คือ ผ่านตัวแปรกลาง หรือ ส่งข้อความ

เมื่อใช้การผ่านตัวแปรกลาง แต่ละทาสก์หรือไอเอสอาร์ต้องแน่ใจว่ามีการเข้าถึงข้อมูลเพียงทีละหนึ่งทาสก์เท่านั้น ถ้าไอเอสอาร์ได้ถูกเรียกแล้วมีทางเดียวที่จะเข้าถึงข้อมูลได้อย่างปลอดภัยคือ การไม่ให้ใช้อินเทอร์รัพท์ (disabled interrupt) แต่ถ้ามี 2 ทาสก์ที่ขอใช้งานตัวแปรร่วมตัวเดียวกันสามารถทำได้โดยไม่ให้ใช้อินเทอร์รัพท์หรือเรียกใช้เซมาฟอร์

ทาสก์สามารถติดต่อกับไอเอสอาร์ได้เพียงเดียวเท่านั้นคือผ่านตัวแปรกลาง ซึ่งโดยที่จริงแล้วทาสก์ไม่จำเป็นต้องกลัวว่าไอเอสอาร์จะไปเปลี่ยนค่าในตัวแปรกลาง จึงไม่มีความจำเป็นที่ไอเอสอาร์จะต้องส่งสัญญาณไปยังทาสก์ โดยใช้เซทาฟอร์หรือโพล์ (poll) เพื่อให้เหมาะกับสถานการณ์เช่นนี้ คุณควรเลือกใช้แมสเซสเมล์บ็อกซ์ (message mailbox)

3.2.6 แมสเซสเมล์บ็อกซ์

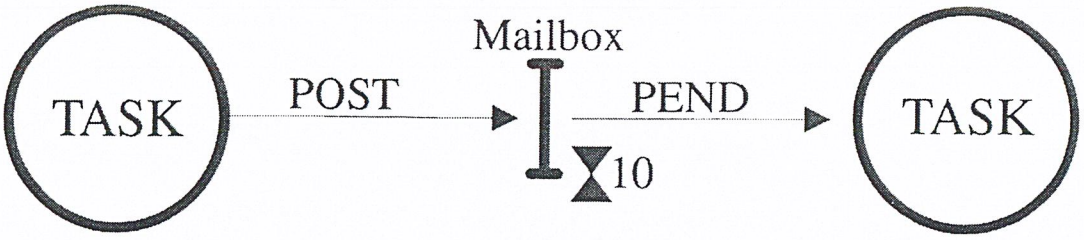
ข้อความสามารถถูกส่งไปยังทาสก์ผ่าน " kernel services " คือแมสเซสเมล์บ็อกซ์หรือเรียกว่า " Message exchange " โดยเก็บผ่านตัวแปรพอยเตอร์ โดยทาสก์หรือไอเอสอาร์สามารถฝากแมสเซสลงในเมล์บ็อกซ์และ 1 หรือ หลายทาสก์สามารถรับแมสเซสผ่านบริการที่เคอร์เนลเตรียมไว้ โดยการส่งและรับข้อความจะเป็นการส่งข้อมูลที่พอยเตอร์นั้น ๆ ชี้อยู่

Waiting list ที่สัมพันธ์แต่ละเมล์บ็อกซ์ ในกรณีที่มี 1 หรือ หลาย ๆทาสก์ต้องการรับข้อความผ่าน เมล์บอกซ์ โดยทาสก์ที่ต้องการข้อความจากเมล์บอกซ์ที่ว่าง จะถูกทำให้พักการทำงานและถูกทำให้รออยู่ใน Waiting list จนกระทั่งมีข้อความเข้ามาในเมล์บ็อกซ์ โดยเคอร์เนลอนุญาตให้รอได้จนถึง timeout ถ้าข้อความยังมาไม่ถึงก่อนเวลาที่กำหนดไว้ทาสก์ที่รอรับข้อความจะถูกทำให้ ready และทำงาน โดยส่ง error code กลับไป เมื่อข้อความได้ถูกนำออกจากเมล์บ็อกซ์ทาสก์ที่มีระดับความสำคัญสูงที่สุดจะทำการรับข้อความ (แบบ priority based) หรือทาสก์แรกที่เข้ามารอเป็นผู้รับข้อความ (แบบ FIFO based)

เคอร์เนลได้เตรียมบริการสำหรับเมล์บ็อกซ์ดังนี้

- กำหนดค่าเริ่มต้นของเนื้อหาในเมล์บ็อกซ์ในตอนแรกเริ่มเมล์บ็อกซ์สามารถกำหนดให้มีหรือไม่มีแมสเซสได้
- ฝากแมสเซสลงไปในเมล์บ็อกซ์ (POST)
- รอรับแมสเซสที่ได้ทำการฝากลงมาในเมล์บ็อกซ์ (PEND)
- รับแมสเซสจากเมล์บ็อกซ์ ถ้ามีข้อความอยู่ แต่จะไม่พักการทำงานของผู้ที่เรียกถ้าเมล์บ็อกซ์ ว่างเปล่า (ACCEPT) ถ้าเมล์บ็อกซ์มีข้อความอยู่ ข้อความจะถูกนำออกจากเมล์บ็อกซ์ และส่งโค้ดกลับจะถูกใช้นำไปใช้ในการเตือนผู้ที่เรียกเกี่ยวกับผลลัพธ์ของการเรียก

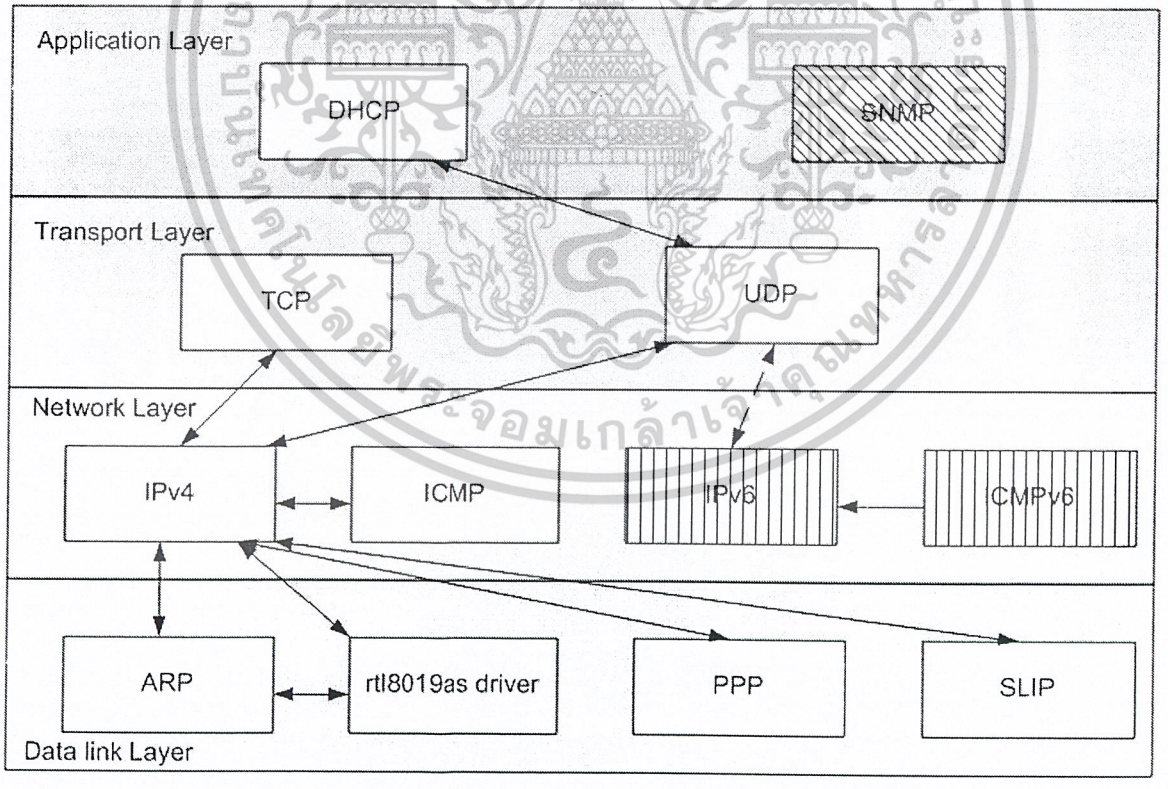
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-2 แมสเซสเมล์บ็อกซ์

3.3 ไลท์เวททีซีพี/ไอพีแสต็ก (Light weight protocol stack : lwIP)

ไลท์เวททีซีพี/ไอพีแสต็กเป็นการอิมพลีเมนท์ทีซีพี/ไอพีแสต็ก ไลท์เวททีซีพี/ไอพีแสต็กนั้นมีเป้าหมายในการลดขนาดการใช้หน่วยความจำลงและขนาดของโค้ด ทำให้ไลท์เวททีซีพี/ไอพีแสต็กนั้นเหมาะกับที่จะใช้ในไคลเอนท์ที่มีขนาดเล็กมีทรัพยากรจำกัดอย่างเช่น ระบบฝังตัว เป็นต้น โดยสามารถที่จะลดโพรเซสและการใช้หน่วยความจำลงได้ ไลท์เวททีซีพี/ไอพีแสต็กใช้วิธีการของ API ซึ่งทำให้ไม่จำเป็นที่ต้องคัดลอกข้อมูล โดยแสดงโครงสร้างของทีซีพี/ไอพีแสต็กได้ดังภาพ



รูปที่ 3-3 โครงสร้างของไลท์เวททีซีพี/ไอพีแสต็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 โพรเซสโมเดล (Process Model)

แบ่งออกเป็น 2 แบบด้วยกัน

1. ทำให้แต่ละโปรโตคอลรันแบบแสตนด์อโลนโพรเซส (stand alone process) หมายความว่าแต่ละโปรโตคอลจะแยกกันทำงานเดี่ยว ๆ ด้วยโมเดลแบบนี้จะเข้มงวดในการบังคับโปรโตคอลเลเยอร์ (Strict Protocol layering) และจุดของการติดต่อสื่อสารระหว่างโปรโตคอลจะเข้มงวดในการกำหนด ซึ่งมีข้อดีคือ โปรโตคอลสามารถถูกเพิ่มเข้าไปในระบบในระหว่างรันไทม์ (run time) ได้ ทำให้การทำความเข้าใจโค้ดและการดีบักเป็นไปได้ง่ายขึ้น แต่ก็มีข้อเสียคือ การที่จะทำให้เป็นโปรโตคอลที่เป็นแบบเข้มงวดในการบังคับโปรโตคอลเลเยอร์นั้นต้องมีการทำคอนเท็กซ์สวิชชิ่ง (context switching) เช่น ถ้ามีทีซีพีเซ็กเมนท์เข้ามาจะมีการทำ 3 คอนเท็กซ์สวิช จากดีไวซ์ไดรเวอร์ไปยังเน็ตเวิร์คอินเทอร์เฟส จากไอพีโพรเซสไปยังทีซีพีโพรเซส แล้วจากทีซีพีโพรเซสไปยังแอพพลิเคชันโพรเซส เป็นต้น

2. ทำให้การสื่อสารระหว่างโปรโตคอลนั้นอยู่ข้างๆกับเคอร์เนลของโอเอส (เคอร์เนลอิมพลีเมนท์เทชัน) โดยแอพพลิเคชันโพรเซสจะสื่อสารกับโพรเซสผ่านทางซิสเต็มคอล (System calls) ใช้เทคนิคของการข้ามชั้นโปรโตคอล (crossing the protocol layering) เพราะโปรโตคอลนั้นไม่ได้ถูกแบ่งออกเป็นชั้นอย่างชัดเจนเหมือนแบบที่ 1

ไลท์เวททีซีพี/ไอพีแสต็กนั้นใช้โพรเซสโมเดลแบบให้โปรโตคอลนั้นอยู่ข้างๆกับชิงเกิ้ลโพรเซส ซึ่งแยกออกจากเคอร์เนลของระบบปฏิบัติการ โดยที่แอพพลิเคชันโปรแกรมสามารถที่จะอยู่ข้างๆ ไลท์เวททีซีพี/ไอพีแสต็กโพรเซสได้หรือเป็นโพรเซสแยกต่างหากก็ได้ การสื่อสารระหว่างทีซีพี/ไอพีแสต็กและแอพพลิเคชันโปรแกรมจะกระทำโดยการใช้ฟังก์ชันในการเรียก สำหรับกรณีที่แอพพลิเคชันโปรแกรมนั้นแบ่งปันโพรเซสกับไลท์เวททีซีพี/ไอพีแสต็ก หรือใช้ abstract API ไลท์เวททีซีพี/ไอพีแสต็กจะใช้โพรเซสในส่วนของผู้ใช้ (user space process) แทนที่จะใช้เคอร์เนลของระบบปฏิบัติการ ซึ่งมีข้อดีก็คือ สามารถพอร์ตข้ามระบบปฏิบัติการที่ต่างกันได้ ถ้าออกแบบให้ใช้งานกับระบบปฏิบัติการขนาดเล็ก มันจะไม่รองรับ swapping out process กับหน่วยความจำเสมือน (virtual memory) ทำให้ไม่เกิดความล่าช้า (delay) จากการที่ต้องรอ I/O (disk) ถ้าแค่บางส่วนของไลท์เวททีซีพี/ไอพีแสต็กโพรเซสนั้นถูก swap หรือ page out ไปยังดิสก์นั้นไม่ใช่ปัญหา แต่ก็มีข้อเสียคือ มีการรอ scheduling quantum ก่อนที่จะมีโอกาสที่จะขอรับบริการได้

3.3.2 อีมูเลชันเลเยอร์ของระบบปฏิบัติการ (The Operating System Emulation Layer)

ในการที่จะทำให้ไลท์เวททีซีพี/ไอพีแสต็กสามารถพอร์ตกับระบบปฏิบัติการต่างๆได้นั้น ระบบปฏิบัติการจะไม่ได้เรียกใช้ฟังก์ชันเฉพาะและคาต้าสตรักเจอร์โดยตรงจากโค้ด และเมื่อจำเป็นที่จะต้องเรียกใช้ก็จะใช้อีมูเลชันเลเยอร์ขึ้นมาแทน ซึ่งเป็นยูนิฟอร์มอินเทอร์เฟส (uniform interface) ไปยังโอเอสเซอร์วิส เช่น ไทม์เมอร์ โพรเซสซิงโครไนซ์เซชัน กระบวนแมสเซสพาสซิ่ง เป็นต้น การที่จะพอร์ตไลท์เวททีซีพี/ไอพีแสต็กกับระบบปฏิบัติการอื่นนั้นก็เพียงแก้ไขในส่วนนี้ให้เหมาะสมกับระบบปฏิบัติการที่จะใช้

อีมูเลชันเลเยอร์ของระบบปฏิบัติการนั้นมีฟังก์ชันเกี่ยวกับไทม์เมอร์ที่ถูกใช้งานโดยทีซีพีเป็นแบบอันช็อตไทม์เมอร์ (one-shot timer) ที่อย่างน้อย 200 มิลลิวินาทีก็จะเรียกรีจิสเตอร์ฟังก์ชันขึ้นมา (เมื่อเกิด time-out)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับกระบวนการโพรเซสซิงโครไนซ์นั้นใช้เซมาฟอร์ ถ้าระบบปฏิบัติการนั้นไม่มีการทำเซมาฟอร์ก็จะจำลองแบบอื่นๆ ขึ้นมาใช้งานได้ เช่น เงื่อนไขของตัวแปร หรือล็อค (Lock)

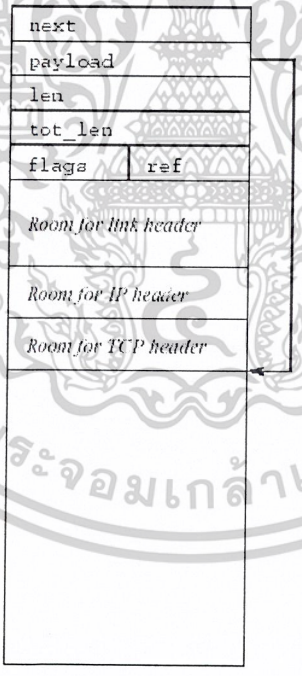
แมสเซสพาสซิ่งนั้นทีการกระทำผ่านการใช้ abstraction called mailboxes โดยที่เมล์บอกซ์นั้นมี 2 โอเปร์เรชันคือ post กับ fetch โดยที่การ post นั้นจะไม่บล็อกโพรเซสและแมสเซสที่ถูก post ในเมล์บอกซีจะถูกจัดคิว โดยอีมูเลชันเลเยอร์ของระบบปฏิบัติการจนกว่าจะมีโพรเซสอื่นมา fetch มันออกไป

3.3.3 การจัดการบัฟเฟอร์และหน่วยความจำ (Buffer and Memory management)

3.3.3.1 แพ็กเก็ตบัฟเฟอร์ (Packet buffers-pbufs)

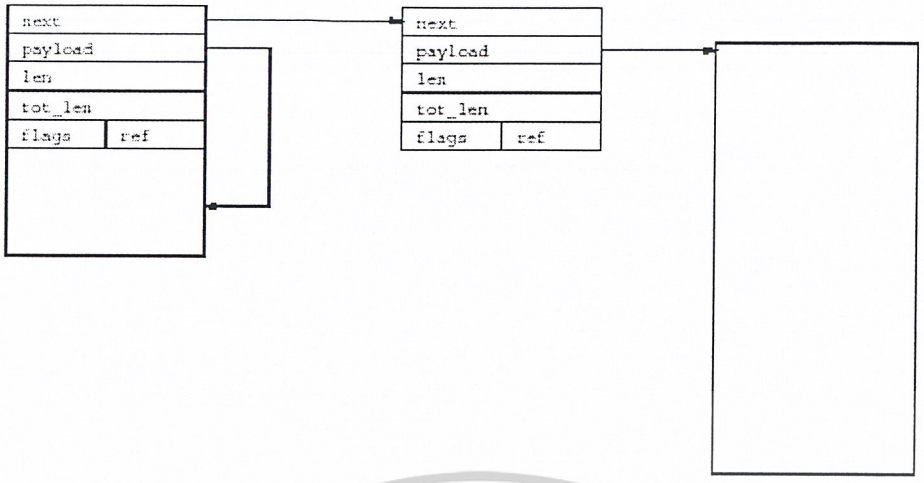
pbuf ใช้เป็นตัวแทนของแพ็กเก็ตภายในไลท์เวททีซีพี/ไอพีแสต็ก และถูกออกแบบมาสำหรับความต้องการพิเศษที่จะใช้แสต็กน้อยๆ ซึ่งคล้ายกันกับ mbuf ใน BSD implementations โดยที่โครงสร้างของ pbuf จะสนับสนุนการกำหนดพื้นที่แบบไดนามิกส์ เพื่อเก็บแพ็กเก็ตคอนเท็นท์และสำหรับแพ็กเก็ตดาต้าที่อยู่ข้าง static memory pbuf สามารถเชื่อมเข้าด้วยกันแบบลิงค์ลิสต์ได้เรียกว่า pbuf chain ดังนั้นบางทีแพ็กเก็ตถูกแบ่งเป็นหลายๆ ส่วน

โดยที่ pbuf นั้นมี 3 แบบคือ pbuf_ram, pbuf_rom, pbuf_pool ดังนี้

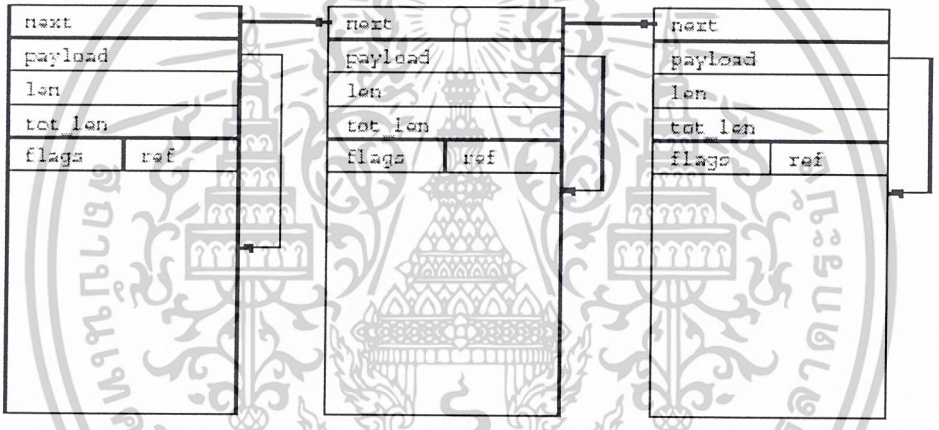


รูปที่ 3-4 pbuf_ram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-5 pbuf_ram เชื่อมต่อเข้ากับpbuf_rom ที่มีข้อมูลอยู่ที่ external memory



รูปที่ 3-6 chained pbuf_pool จาก pbuf_pool

pbuf_pool นั้นใช้โดยเน็ตเวิร์คดีไวซีไดร์เวอร์ โดยที่โอเปอร์เรชันในการกำหนดพื้นที่ของ single pbuf นั้นเร็วและเหมาะกับที่จะใช้กับอินเทอร์รัพท์แฮนด์เดิลเลอร์ ส่วน pbuf_rom นั้นจะใช้เมื่อโปรแกรมประยุกต์ส่งข้อมูลที่อยู่ในหน่วยความจำที่จัดการโดยโปรแกรมประยุกต์ ข้อมูลเหล่านั้นจะไม่ถูกเปลี่ยนแปลงหลังจาก pbuf ได้เก็บไว้ในทีซีพี/ไอพีแสต็กและเมื่อข้อมูลอยู่ในรอม (pbuf_rom) แล้ว

pbuf ของ pbuf_ram นั้นจะถูกใช้เมื่อโปรแกรมประยุกต์ส่งข้อมูลเป็นไดนามิกส์ Pbuf system จะจองหน่วยความจำ ไม่เพียงแต่ข้อมูลของโปรแกรมประยุกต์ แต่จะมี Header จองเตรียมไว้ล่วงหน้า และในแบบที่แย่ที่สุด (worse case) ขนาดของ Header จะถูกติดตั้งในเวลาที่คอมไพล์ด้วย

โดย pbuf ที่เข้ามาจะเป็น pbuf_pool และ pbuf ที่ออกไปจะเป็น pbuf_ram หรือ pbuf_rom โครงสร้างของ pbuf นั้นประกอบด้วย 2 pointers, 2 length fields, Flag field, และตัวนับ next คือพอยเตอร์ไปยัง pbuf ถัดไปในกรณีที่มีการ chain "Payload pointer" เป็นตัวชี้จุดเริ่มของข้อมูล "Len field" เก็บความยาวของข้อมูลใน pbuf "Tot_len field" เก็บผลรวมของความยาวปัจจุบันและความยาวทั้งหมดใน chain ในอีกความหมายหนึ่งคือ tot_len field เป็นผลรวมความยาวของฟิงค์และค่าของ tot_len ในการตาม pbuf ใน pbuf chain "Flag

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ให้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

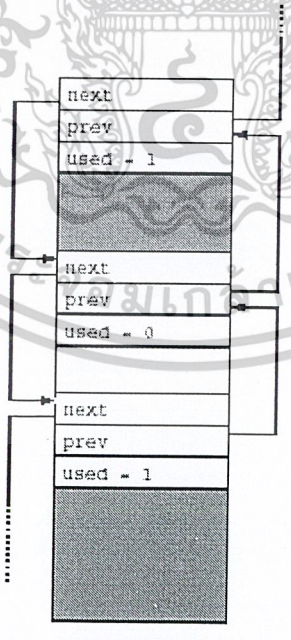
field" ใช้แสดงชนิดของ pbuf และ "ref field" จะประกอบด้วย reference count โดยที่ "next" และ "payload field" นั้นเป็น pointer ที่มีมาตั้งแต่แรก และขนาดจะแปรตามสถาปัตยกรรมของโปรเซสเซอร์ที่ใช้ "two len field" เป็น 16 บิตอันไซน์อินทิเจอร์และ flag กับ ref เป็น 4 บิต ขนาดทั้งหมดของ pbuf ขึ้นอยู่กับขนาดของพอยเตอร์ในสถาปัตยกรรมของโปรเซสเซอร์ สำหรับในสถาปัตยกรรม 32 บิต และ 4 ไบต์อะไลน์เมนท์ ขนาดทั้งหมด 16 ไบต์และบนสถาปัตยกรรมที่เป็น 16 บิตพอยเตอร์และ 1 ไบต์จะมีขนาดรวม 9 ไบต์

3.3.3.2 การจัดการกับหน่วยความจำ (Memory management)

ใช้ memory manager ทำหน้าที่กำหนดและยกเลิกกำหนดพื้นที่ที่ติดกับของหน่วยความจำและสามารถลดขนาดที่ได้กำหนดไว้ก่อนหน้านี้ได้ memory manager ใช้บางส่วนของหน่วยความจำทั้งหมดที่ได้รับในระบบ โอเปอร์เรชันของโปรแกรมอื่นๆจะไม่เข้ามากวนได้ ถ้าเน็ตเวิร์คซิสเต็มใช้หน่วยความจำทั้งหมด

Memory manager จะติดตามการกำหนดหน่วยความจำ โดยการเอาโครงสร้างขนาดเล็กไปแปะไว้กับด้านบนของแต่ละบล็อกของหน่วยความจำที่กำหนดไว้ โครงสร้างนี้มี 2 พอยเตอร์ชี้ไปยังตัวต่อไปและตัวก่อนหน้า และยังใช้ flag ในการบอกว่าบล็อกนั้นถูกกำหนดไปแล้วหรือยัง

หน่วยความจำจะถูกกำหนดโดยการบล็อกที่กำหนดไว้ที่ยังไม่ถูกใช้ที่มีขนาดใหญ่เพียงพอตามที่ขอมาโดยใช้อัลกอริทึม "first fit" เมื่อมีการยกเลิกกำหนดพื้นที่แล้ว use flag ก็จะถูกกำหนดเป็น 0 เพื่อป้องกันการเฟรกเมนท์ use flag ของก่อนหน้านี้และถัดไป ตัวบล็อกจะถูกตรวจสอบก่อน ถ้ามีอันใดก็ตามที่มีสถานะเป็น unused อยู่ บล็อคก็จะถูกรวมเป็นบล็อกของ unused ที่ใหญ่ขึ้น



รูปที่ 3-7 โครงสร้างของการกำหนดหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 โปรโตคอลที่สนับสนุน

เนื่องจากไลท์เวทที่ซีพี/ไอพีแสต็กนั้นมีการสนับสนุนการทำงานของโปรโตคอลไม่ครบทุกอย่าง จึงสามารถจำแนกได้ดังตารางต่อไปนี้

Protocol	Feature support	หมายเหตุ
ARP	ARP request	ARP packet จะต้องเป็น local broadcast packet บน local link เท่านั้น
	ARP reply	
	RFC 3220 4.6 IP Mobility Support for IPv4	
SLIP		จะต้องมี serial I/O เพราะเป็นการส่ง IP packet บน Dial-up line module
PPP	Link Control Protocol	PPP IP Control Protocol
	Network Control Protocols (NCPs)	
	Authentication and Phase control	
	Network Challenge Handshake Authentication Protocol (CHAP)	
	Network Password Authentication Protocol (PAP)	
	MD5	
IPv4	Fixed IP address	ทำ local network binding โดยใช้ wildcard
	Class A	
	Class B	
	Class C	
	Class D	
	Broadcast address	
	Loop back address	
	IP fragmentation	
	IP reassemble	
	IP checksum	
ICMP	Network Unreachable	
	Host Unreachable	
	Protocol Unreachable	
	Port Unreachable	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Fragmentation needed but DF set	
	Source route failed	
	Time to live exceeded in transit	ต้องอนุญาตให้มีการทำ IP forward
	Fragment reassembly time exceeded	ต้องอนุญาตให้มีการทำ IP forward
IPv6	IPv6 header	
	IP checksum	
ICMPv6	Network Unreachable	
	Host Unreachable	
	Protocol Unreachable	
	Port Unreachable	
	Fragmentation needed and DF set	
	Time to live exceeded in transit	ต้องอนุญาตให้มีการทำ IP forward
	Fragment reassembly time exceeded	ต้องอนุญาตให้มีการทำ IP forward
TCP	Multiple TCP connections	
	TCP options	
	Variable TCP Maximum Segment Size	
	Round Trip Time Estimation	
	TCP Flow Control	
	Sliding TCP Window	
	TCP Congestion Control	
	Out-of-Sequence TCP data	
	TCP urgent data	
	Data buffered for retransmit	
UDP	UDP lookup	Incomplete
DHCP	Client mode	
SNMP		Define function แต่ไม่มี code ในการทำงาน

ตารางที่ 3-1 โปรโตคอลที่สนับสนุน

3.3.5 โครงสร้างข้อมูลที่เกี่ยวข้อง

3.3.5.1 eth_addr

เป็นโครงสร้างของแม็คแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5.2 eth_hdr

เป็นโครงสร้างของ header ของเออาร์พีแพ็กเก็ต แสดงโครงสร้างได้ดังรูป

รูปที่ 3-8 โครงสร้างของ eth_hdr

ซึ่งประกอบด้วย source, destination และ type

3.3.5.3 ip_addr

เป็นโครงสร้างของไอพีแอดเดรสในรูปแบบของเน็ตเวิร์คไบต์ออร์เดอร์ (Network byte order) ที่มีดาต้าฟิลด์เป็น u32_t addr

3.3.5.4 ip_hdr

เป็นเฮดเดอร์ของไอพีแพ็กเก็ต โดยมีโครงสร้างดังนี้

รูปที่ 3-9 โครงสร้างของ ip_hdr

ซึ่งประกอบด้วย

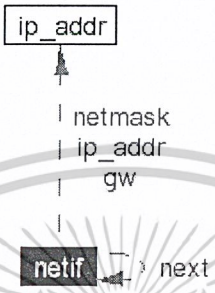
_v_hl_tos	version, header length, type of service
len	total length
_id	identification
_offset	fragment offset
_ttl_proto	time to live, protocol
_chksum	checksum
src	source IP address
dest	destination IP address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5.5 netif

เป็นตัวแทนเน็ตเวิร์คอินเทอร์เฟส โดยจะเป็นตัวที่เชื่อมโยงระหว่างแสต็กกับเน็ตเวิร์คอินเทอร์เฟส โดยที่เน็ตเวิร์คอินเทอร์เฟสอาจจะเป็นดีไวซ์ไดรเวอร์ไปยังฟิสิคัลเน็ตเวิร์คฮาร์ดแวร์ก็ได้ หรืออาจจะเป็นเวรปเปอร์ไปยังอินเทอร์เฟสอื่น เช่น /dev/tun หรือ /dev/tap (UNIX/Linux)

โดยที่เน็ตเวิร์คอินเทอร์เฟสจะถูกเก็บไว้บนโกลบอลลิ้งค์ลิส (Global linked list) ซึ่งเชื่อมโยงโดยพอยท์เตอร์ดังโครงสร้างต่อไปนี้



รูปที่ 3-10 โครงสร้างของ netif

ซึ่งประกอบด้วย

next	อินเทอร์เฟสถัดไปใน global connected list
num	จำนวนของอินเทอร์เฟสใน global connected list
ip_addr	โลคอลไอพีแอดเดรสในแบบเน็ทเวิร์คไบต์ออร์เดอร์
netmask	เน็ทมาสก์ในแบบเน็ทเวิร์คไบต์ออร์เดอร์
gw	ไอพีแอดเดรสของเกตเวย์ในแบบเน็ทเวิร์คไบต์ออร์เดอร์
hwaddr	ฮาร์ดแวร์แอดเดรสของดีไวซ์
name	ชื่อของแต่ละเน็ทเวิร์คอินเทอร์เฟส
state	ระบุถึงสถานะของอุปกรณ์โดยดีไวซ์ไดร์เวอร์

3.3.5.6 pbuf

โครงสร้างของเฮดเดอร์ของแพ็กเก็ตบัฟเฟอร์ ออกแบบมาเพื่อให้เหมาะกับแสต็กขนาดเล็กโดยเฉพาะ ซึ่งคล้ายกับ mbuf ของ BSD ซึ่งรองรับทั้งการกำหนดพื้นที่หน่วยความจำแบบไดนามิกซ์และแบบคงที่ สามารถเชื่อมต่อกันใน 1 ลิสต์ได้ เรียกว่า pbuf chain ดังที่ได้กล่าวไว้แล้วข้างต้น

รูปที่ 3-11 โครงสร้างของ pbuf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

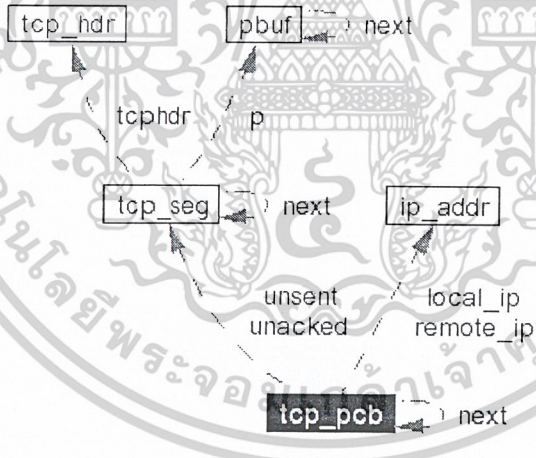
3.3.5.7 tcp_hdr

โครงสร้างของเฮดเดอร์ของทีซีพีแพ็กเก็ต ซึ่งประกอบด้วย

src	ต้นทาง
dest	ปลายทาง
seqno	เลขลำดับ (sequence number)
ackno	หมายเลขตอบรับ (Acknowledge number)
_offest_flags	ออฟเซ็ทและเฟลก
wnd	วินด์โดว์
chksum	checksum
urgp	urgent pointer

3.3.5.8 tcp_pcb

โครงสร้างของโปรโตคอลคอนโทรลบล็อกสำหรับทีซีพีคอนเน็คชัน ซึ่งประกอบด้วยข้อมูลสำหรับทีซีพีคอนเน็คชัน และเก็บอยู่บนลิงค์ลิส นอกจากนี้ยังมีพอยท์เตอร์ไปยัง PCB ถัดไปใน global linked list ของ TCP PCBs ด้วย สามารถแสดงโครงสร้างได้ดังนี้



รูปที่ 3-12 โครงสร้างของ tcp_pcb

ซึ่งประกอบด้วย

next	พอยท์เตอร์ไปยัง PCB ถัดไปใน global linked list ของ TCP PCBs
callback_arg	สำหรับ callback ฟังก์ชัน
local_ip	โลคอลไอพีแอดเดรส
local port	โลคอลพอร์ต
remote_ip	รีโมทไอพีแอดเดรส
remote_port	รีโมทพอร์ต
state	TCP state
rtime	รีทรานสมิทชันไทม์เมอร์

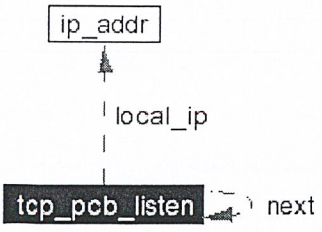
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mss	maximum segment size
rto	retransmission time-out
nrtx	จำนวนในการรีทรานสมิท
snd_buf	ขนาดของบัฟเฟอร์ที่มีอยู่สำหรับใช่ส่งเป็นไบต์
snd_queuelen	ขนาดของบัฟเฟอร์ที่มีอยู่สำหรับใช่ส่งเป็นทีซีพีเซกเมนต์
unsent	คิวของเซกเมนต์ที่ยังไม่ได้ส่งโดยเรียงตามเลขลำดับ
unacked	ที่ส่งแล้วแต่ยังไม่ได้รับ ACK กลับเรียงตามเลขลำดับ
rcv_nxt	เลขลำดับถัดไปที่คาดหวังว่าจะได้
rcv_wnd	วินโดว์ของผู้รับ
tmr	ไทม์เมอร์
polltmr	ไทม์เมอร์สำหรับการพูล
pollinterval	ช่วงเวลาในการพูล
flags	ค่าเฟรกต่างๆ
rttest	ค่าประมาณของราวทริปไทม์
rtseq	มีการจับเวลาของเลขลำดับนั้น
sa	ค่าประมาณของราวทริปไทม์
sv	ค่าประมาณของราวทริปไทม์
lastack	ค่าเลขลำดับของ ACK สูงสุดที่ได้รับ
dupacks	ค่าเลขลำดับของ ACK สูงสุดที่ได้รับ
cwnd	congestion avoidance/contol
ssthresh	congestion avoidance/contol
snd_nxt	เลขลำดับถัดไปที่จะส่ง
snd_max	ค่าเลขลำดับสูงสุดที่ส่ง
snd_wnd	วินโดว์ของผู้ส่ง
snd_wl1	เลขลำดับและเลขตอบรับของวินด์โดว์สุดท้ายที่ได้รับการแก้ค่า
snd_wl2	เลขลำดับและเลขตอบรับของวินด์โดว์สุดท้ายที่ได้รับการแก้ค่า
snd_lbb	เลขลำดับของไบต์ถัดไปที่จะถูกบัฟเฟอร์

3.3.5.9 tcp_pcb_listen

โครงสร้างของสถานะของทีซีพีในโหมด LISTEN และ TIME_WAIT เพราะมีความจำเป็นที่จะต้องเก็บสถานะน้อยกว่าสถานะอื่นๆเลยสามารถใช้ PCB ที่มีขนาดเล็กกว่าปกติในการเก็บได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



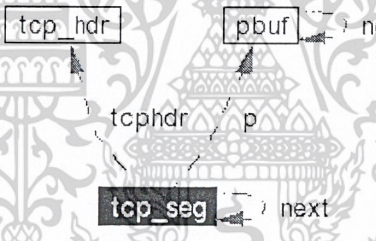
รูปที่ 3-13 โครงสร้างของ tcp_pcb_listen

ซึ่งประกอบด้วย

next	พอยท์เตอร์ไปยัง PCB ถัดไปใน global linked list ของ TCP PCBs
callback_arg	สำหรับ callback ฟังก์ชัน
local_ip	โลคอลไอพีแอดเดรส
local_port	โลคอลพอร์ต

3.3.5.10 tcp_seg

เป็นโครงสร้างของทีซีพีเซกเมนต์เมื่ออยู่ในคิว



รูปที่ 3-14 โครงสร้างของ tcp_seg

ซึ่งประกอบด้วย

next	ใช้เมื่อมีการเอาเซกเมนต์ไปใส่คิว
p	บัฟเฟอร์ที่ประกอบด้วยข้อมูลและเฮดเดอร์ของทีซีพี
dataptr	พอยต์เตอร์ที่ชี้ไปยังทีซีพีดาต้าภายใน pbuf
len	ความยาวของทีซีพีในเซกเมนต์นี้
tcphdr	เฮดเดอร์ของทีซีพี

3.3.5.11 udp_hdr

โครงสร้างของเฮดเดอร์ของยูดีพี ซึ่งประกอบด้วย

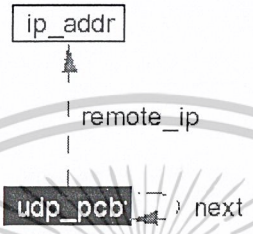
src	source
dest	destination
len	ความยาวของยูดีพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

chksum checksum

3.3.5.12 udp_pcb

โครงสร้างของโปรโตคอลคอนโทรลบล็อกของยูดีพีคอนเนคชัน โดยจะเก็บข้อมูลเกี่ยวกับ 1 ยูดีพีคอนเนคชันใดๆ ซึ่งจะเก็บบนลิงค์ลิสที่ตรงกับยูดีพีดาต้าแกรมที่เข้ามา นอกจากนี้ยังมีพอยต์เตอร์ที่ชี้ไปยัง PCB ถัดไปที่อยู่บน global linked list ของ UDP PCBs ซึ่งสามารถแสดงโครงสร้างได้ดังนี้



รูปที่ 3-15 โครงสร้างของ udp_pcb

ซึ่งประกอบด้วย

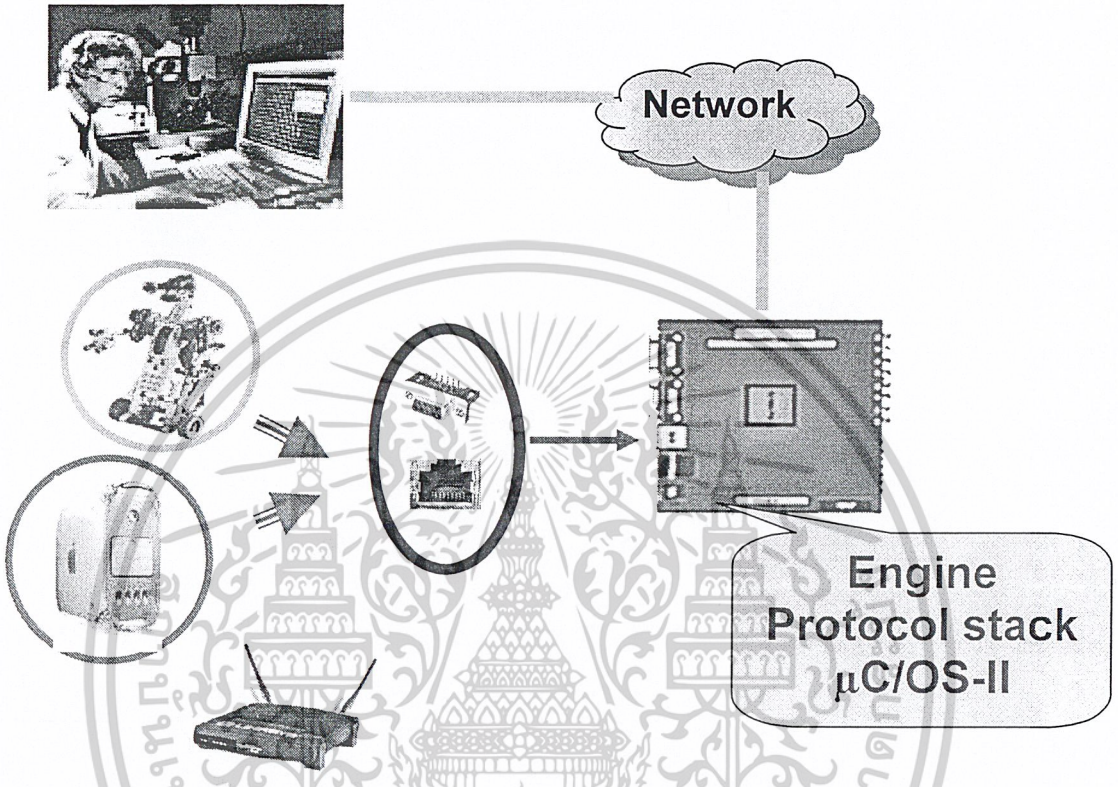
next	พอยต์เตอร์ที่ชี้ไปยัง PCB ถัดไปใน global linked list ของ UDP PCBs
remote_ip	ไอพีแอดเดรสของปลายทางของยูดีพีเซสชัน
local_port	หมายเลขพอร์ตของต้นทางของยูดีพีเซสชัน
remote_port	หมายเลยพอร์ตของปลายทางของยูดีพีเซสชัน
flags	ค่าเฟรกที่ระบุถึงข้อกำหนดของ checksum ของยูดีพีนั้นต้องใช้แซสชันนี้
chksum_len	ใช้ในการบอกขนาด checksum ในโหมด UDP lite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบและพัฒนา

4.1 ขอบเขตของโครงงาน



รูปที่ 4-1 ขอบเขตของโครงงาน

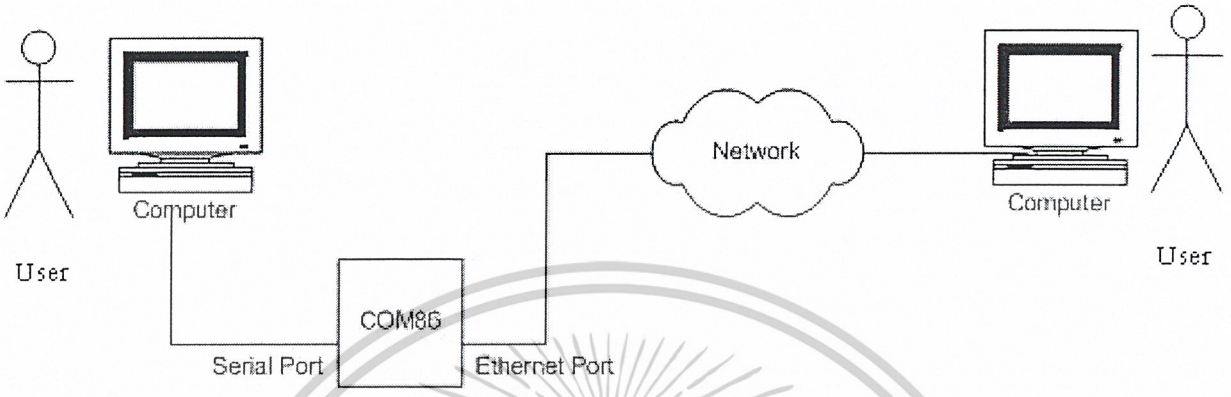
นอกจากการใช้ lwIP (Light weight TCP/IP protocol stack) ในการพัฒนาแล้วการทำงานยังอยู่ภายใต้ความสามารถของชุดพัฒนาคอม 86 ซึ่งสำหรับการทำงานของระบบนั้นมีส่วนที่สามารถติดต่อได้หลายส่วน และได้แบ่งออกเป็น

1. ส่วนคอนโซลพอร์ตอนุกรม ซึ่งผู้ใช้สามารถติดต่อกับอุปกรณ์จากเครื่องคอมพิวเตอร์ผ่านโปรแกรมไฮเปอร์เทอร์มินัลได้
2. ส่วนอีเทอร์เน็ทพอร์ต โดยผู้ใช้สามารถติดต่อกับอุปกรณ์ผ่านทางระบบเครือข่ายได้
3. ส่วนพอร์ตอื่นๆ ของชุดพัฒนาคอม 86 ผู้ใช้สามารถที่จะเพิ่มเติมส่วนต่างๆ ของระบบเข้าไปได้เอง

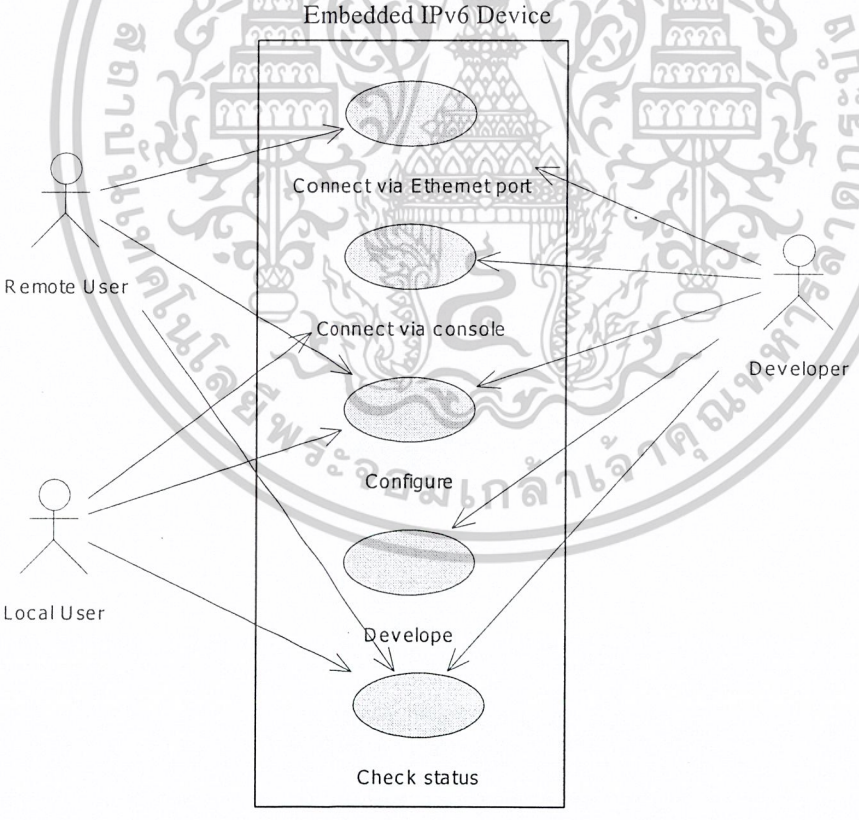
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โครงสร้างและองค์ประกอบของระบบ

โครงสร้างและองค์ประกอบของระบบโดยรวมที่ใช้งานรวมทั้งลักษณะการเชื่อมต่อแสดงดังรูปที่ 4-2 และรูปที่ 4-3



รูปที่ 4-2 ภาพรวมของระบบและการเชื่อมต่อ



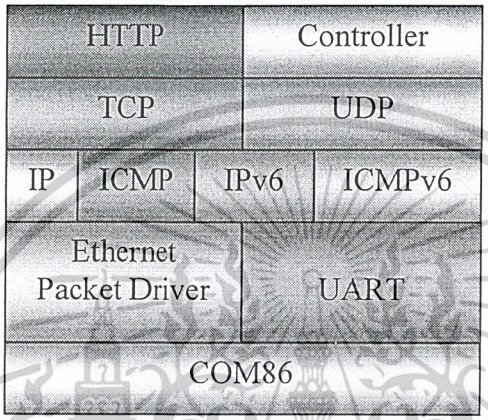
รูปที่ 4-3 แสดงการออกแบบลักษณะการใช้งานอุปกรณ์

ลักษณะการใช้งานอุปกรณ์นี้นั้นสามารถใช้งานได้ทั้งจากระยะไกลและในระยะใกล้ คือการเชื่อมต่อเพื่อใช้งานในระยะไกลนั้นสามารถเชื่อมต่อได้ผ่านระบบเครือข่ายเข้ามาทางอีเทอร์เน็ทพอร์ต ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายเคเบิลอาร์เจ 45 (RJ-45 cable) ส่วนการเชื่อมต่อเพื่อใช้งานในระยะใกล้นั้นสามารถกระทำได้โดยผ่านทางคอนโซลพอร์ตอนุกรมโดยการใช้โปรแกรมไฮเปอร์เทอร์มินัลด้วยสายซีเรียลอาร์เอส-232 โดยผู้ใช้สามารถเข้าไปกำหนดค่าคุณสมบัติต่างๆอุปกรณ์ได้จากทั้ง 2 ทางคือทางเวบเพจหรือทางไฮเปอร์เทอมินัลก็ได้ และนักพัฒนาสามารถใช้อุปกรณ์นี้เป็นอุปกรณ์ต้นแบบในการพัฒนาเพื่อทำงานอื่นต่อไปได้

4.2.1 โครงสร้างของสถาปัตยกรรมของระบบไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86



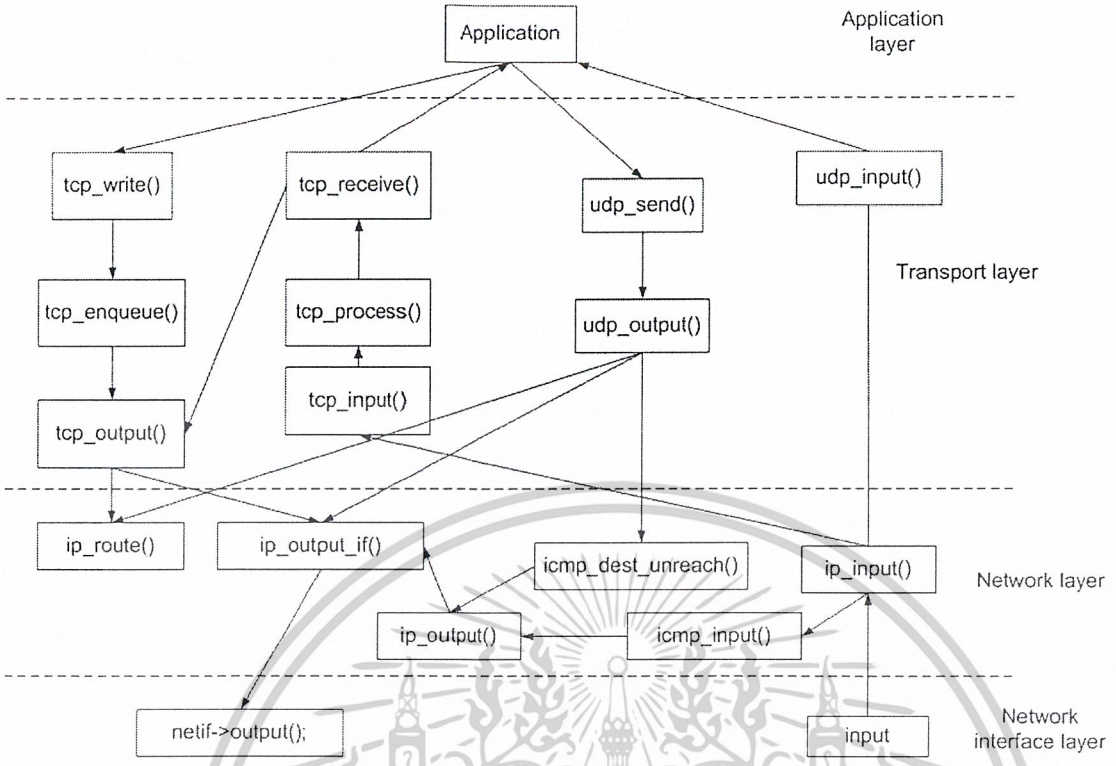
รูปที่ 4-4 แสดงโครงสร้างสถาปัตยกรรมของระบบไอพีรุ่น 6 ฝังตัวในอุปกรณ์คอม 86

จากรูปที่ 4-4 โครงสร้างสถาปัตยกรรมของระบบสามารถเทียบกับแบบอ้างอิงโอเอสไอได้ดังนี้

ชั้นฟิสิคัล	ชุดพัฒนาคอม 86 รวมทั้งสายเชื่อมต่อแบบต่างๆ
ชั้นเดทาลิงค์	อีเทอร์เน็ทแพ็กเก็ตไดร์เวอร์และยูอาร์ต
ชั้นเน็ตเวิร์ค	ไอพี, ไอซีเอ็มพี, ไอพีรุ่น 6, ไอซีเอ็มพีรุ่น 6
ชั้นทรานสพอร์ต	ทีซีพี, ยูดีพี
ชั้นแอพพลิเคชัน	เอชทีทีพีและส่วนที่ทำการควบคุมการทำงานของระบบ

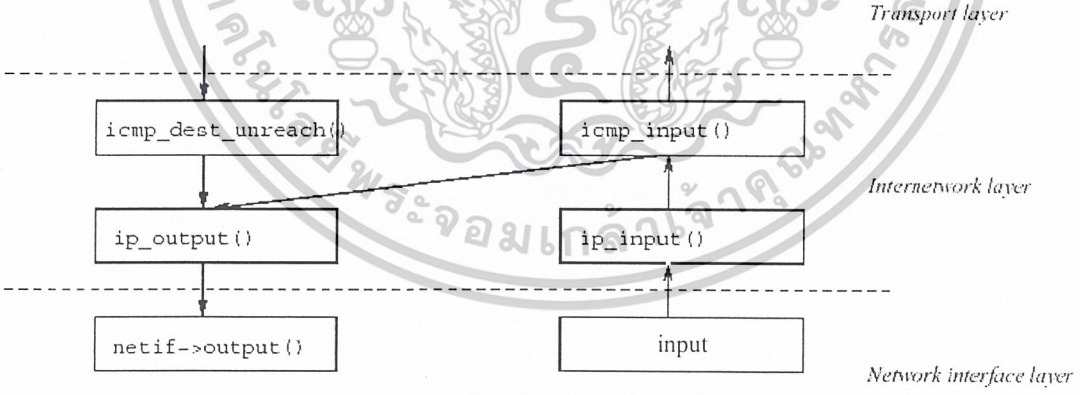
จากรูปที่ 4-5 เป็นการแสดงโครงสร้างการทำงานของระบบระหว่างเลเยอร์เมื่อมีแพ็กเก็ตเข้ามาทางอีเทอร์เน็ทพอร์ต และการส่งข้อมูลข้ามเลเยอร์กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 แสดงโครงสร้างการทำงานระหว่างเลเยอร์

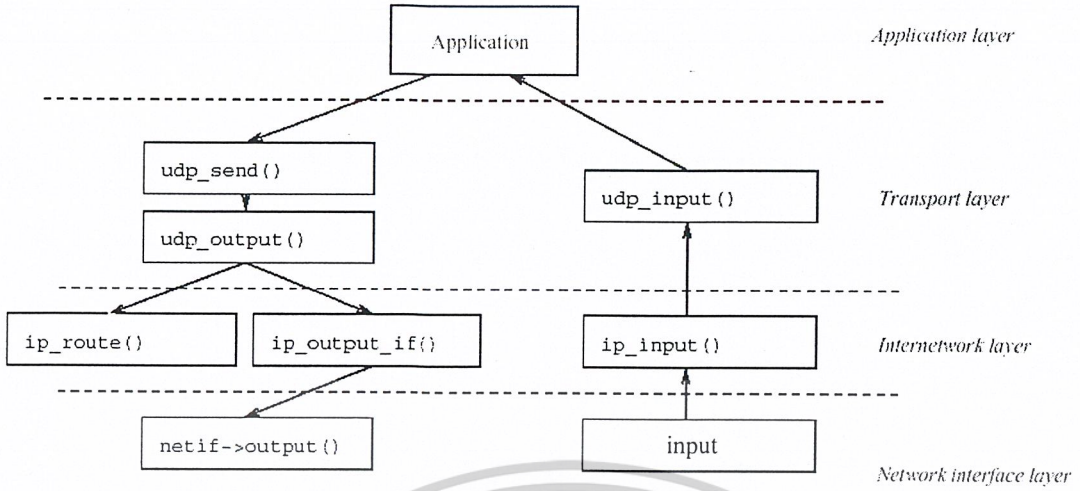
ซึ่งสามารถแยกรายละเอียดได้ดังนี้



รูปที่ 4-6 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของไอซีเอ็มพี

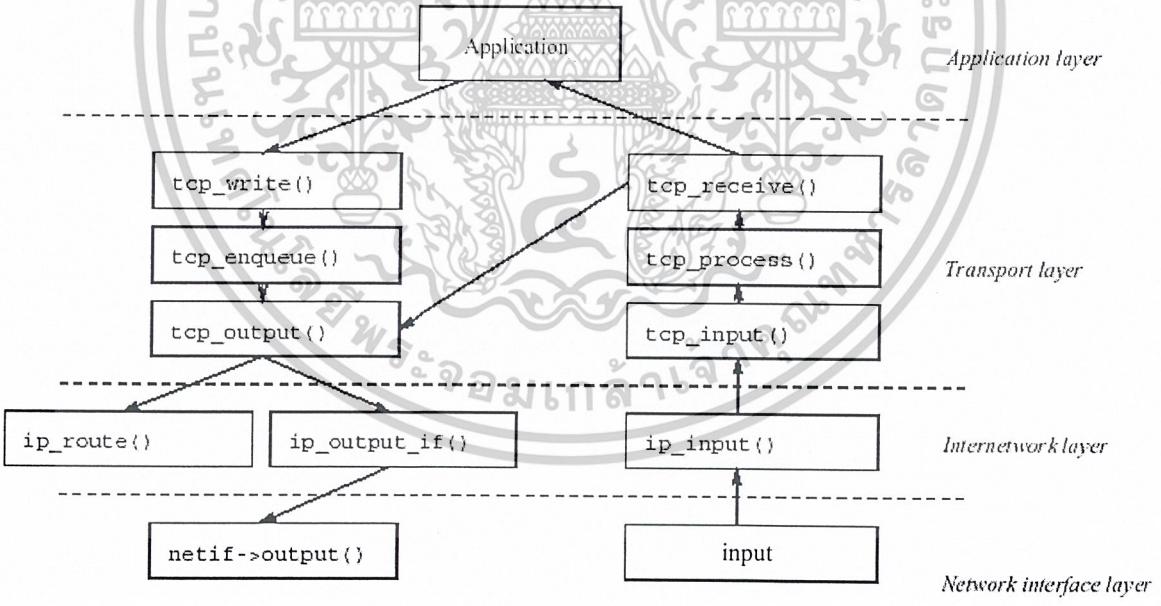
เมื่อมีไอซีเอ็มพีแพ็กเก็ตเข้ามาที่ ip_input() ก็จะถูกส่งเข้าไปที่ icmp_input() ซึ่งเป็นส่วนที่ใช้ติดต่อกันระหว่างไอซีเอ็มพีและไอพี เมื่อมีการกระทำการกับไอซีเอ็มพีแพ็กเก็ตแล้วก็จะรู้ว่าต้องทำอะไรต่อไปหรือส่งให้ใคร เช่น โปรโตคอลยูดีพีจะมีการใช้ icmp_dest_unreach() ในการส่งเดสทิเนชันอันรีชเอเบิลแมสเซสแล้วก็จะส่งออกทาง ip_output() ต่อไป เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของยูดีพี

ในการส่งข้อมูลโปรแกรมประยุกต์ก็จะเรียกใช้ udp_send() ซึ่งจะไปเรียก udp_output() อีกทีเพื่อคำนวณเช็คซัมและเติมเฮดเดอร์เข้าไป อาจจะมีเรียก ip_route() ขึ้นมาเพื่อเลือกเน็ทเวิร์คอินเทอร์เฟสที่เหมาะสมแล้วจึงเรียก ip_output_if() เพื่อที่จะทำการส่งต่อไป



รูปที่ 4-8 แสดงโครงสร้างการทำงานระหว่างเลเยอร์ของทีซีพี

เมื่อโปรแกรมประยุกต์มีความต้องการที่จะส่งข้อมูลแบบทีซีพีก็จะมีการเรียกใช้งาน tcp_write() ซึ่งจะไปเรียก tcp_enqueue() ขึ้นมาทำงาน โดยจะแบ่งข้อมูลออกเป็นทีซีพีเซกเมนต์ที่มีขนาดที่เหมาะสมแล้วเอาไปเข้าคิวสำหรับการติดต่อสื่อสาร tcp_output() ก็จะตรวจสอบความเป็นไปได้ที่จะส่งข้อมูล เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดวินด์โดว์ของฝั่งผู้รับ แล้วถึงจะส่งข้อมูลโดยเรียกใช้ ip_route() และ ip_output_if() ขึ้นมาทำงานต่อไป

เมื่อมีข้อมูลเข้ามา ip_input() ก็จะตรวจสอบเฮดเดอร์ของไอพีก่อนแล้วถึงจะส่งผ่านไปที่ tcp_input() ซึ่งจะตรวจสอบเกี่ยวกับเช็คซัมและออพชัน รวมทั้งตรวจสอบก่อนว่าเป็นของคอนเนคชันไหนก่อนที่จะส่งผ่านไปยัง tcp_process() ซึ่งทำงานตามทีซีพีสเตทแมชชีนและสเตททรานสิชันที่จำเป็นใดๆ แล้ว tcp_receive() ก็จะถูกเรียกเพื่อที่จะส่งต่อข้อมูลไปให้กับโปรแกรมประยุกต์ที่ถูกต้อง จากนั้นก็จะต้องมีการส่ง ACK กลับซึ่งจะมีการเรียกใช้งาน tcp_output() ขึ้นมาทำงานนี้

4.2.2 องค์ประกอบของระบบ

1. ระบบปฏิบัติการ

เนื่องจากการทำงานของระบบต้องสามารถทำงานแบบมัลติทาสก์ได้ เพื่อที่จะสามารถทำงานหลายๆ อย่างพร้อมกันได้และสามารถตอบสนองต่อการทำงานได้อย่างรวดเร็ว ระบบจึงจำเป็นที่จะต้องใช้ระบบปฏิบัติการที่สนับสนุนการทำงานแบบมัลติทาสก์ด้วย ซึ่งผู้พัฒนาได้เลือกใช้ uC/OS-II ซึ่งได้กล่าวถึงแล้วในบทที่ 3

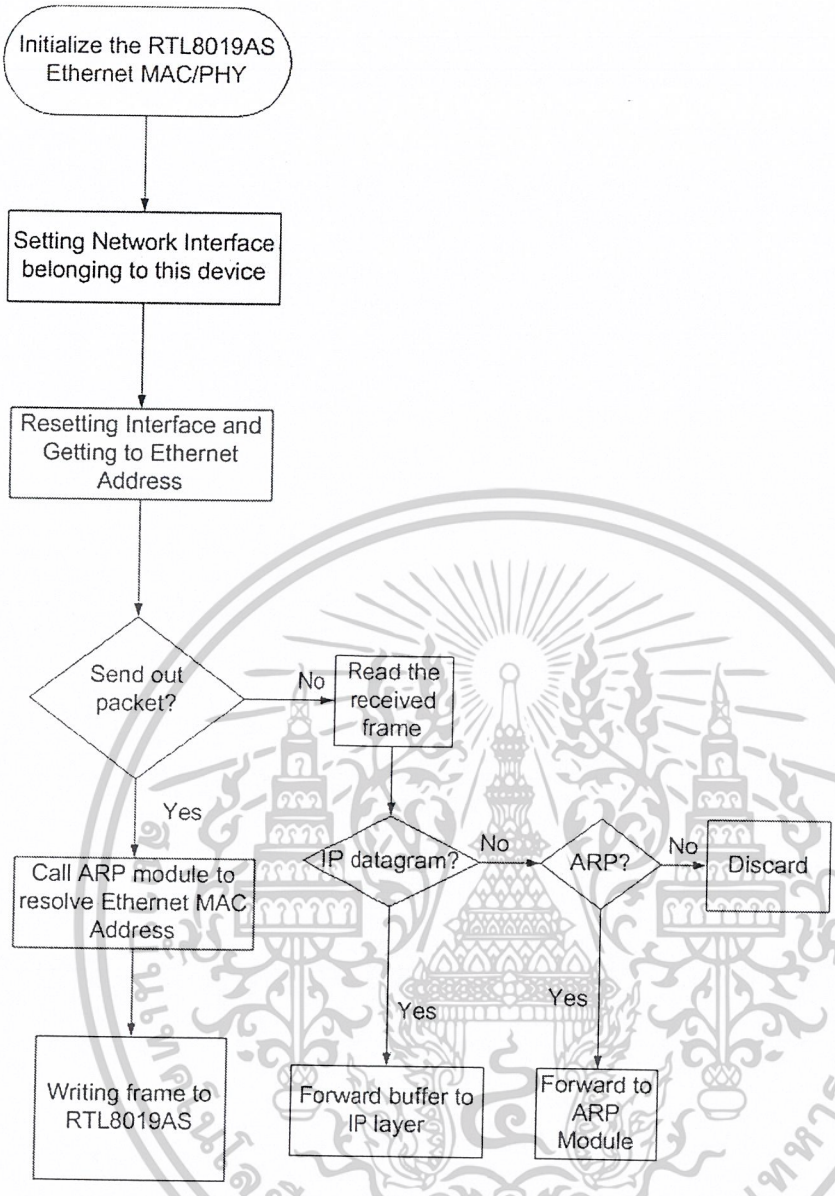
2. โปรโตคอลแสต็ก

ในการสื่อสารในระบบเครือข่ายนั้นจำเป็นต้องอาศัยการทำงานของโปรโตคอลแสต็ก ซึ่งโปรโตคอลแสต็กนั้นต้องเป็นโปรโตคอลแสต็กที่เหมาะสมกับการทำงานบนระบบฝังตัวด้วย เพราะระบบฝังตัวนั้นมีหน่วยความจำน้อย ซึ่งผู้พัฒนาได้เลือกใช้ไลท์เวทโปรโตคอลแสต็ก ซึ่งได้กล่าวถึงแล้วในบทที่ 3

3. ดีไวซ์ไดรเวอร์

การทำงานร่วมกับอุปกรณ์ต่างๆนั้นต้องมีการทำงานของไดรเวอร์เข้ามาเกี่ยวข้องด้วย ซึ่งชุดพัฒนาคอม 86 นั้นใช้ RTL8019AS ในการทำงานเกี่ยวการรับส่งข้อมูลผ่านทางอีเทอร์เน็ตพอร์ต ดังนั้นการที่จะให้อุปกรณ์ทำงานกับโปรโตคอลแสต็กได้อย่างถูกต้องต้องมีการแก้ไขไดรเวอร์หรือเขียนไดรเวอร์ขึ้นมาใหม่ เพื่อให้ทำงานร่วมกันได้ โดยสามารถแสดงลักษณะการทำงานของไดรเวอร์ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-9 แสดงการทำงานของดีไวซ์ไดรเวอร์

4. เอ็นจิน

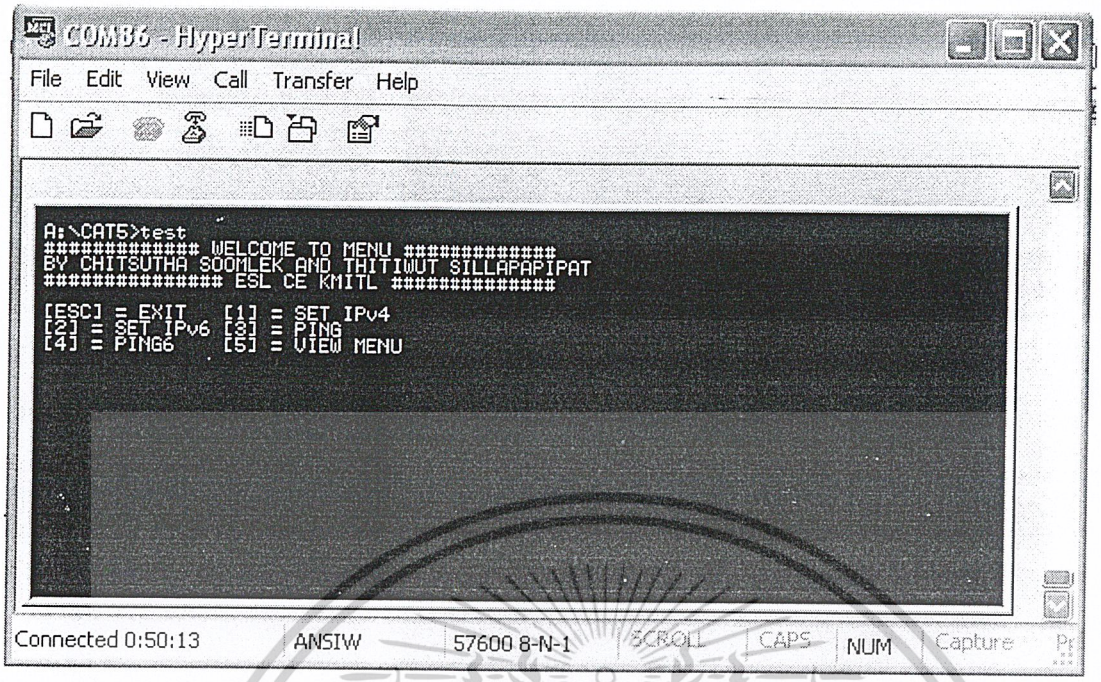
การทำงานของเอ็นจินที่ควบคุมระบบมีส่วนประกอบหลักๆคือ ควบคุมการทำงานและการสื่อสารผ่านทางคอนโซลพอร์ตอนุกรม อีเทอร์เน็ทพอร์ต การทำงานกับระบบปฏิบัติการและโปรโตคอลแสต็ก ให้ทั้งระบบสามารถทำร่วมกันได้อย่างถูกต้องเหมาะสม

5. ส่วนที่ติดต่อกับผู้ใช้ (User Interface)

ส่วนที่ติดต่อกับผู้ใช้นั้นแบ่งออกเป็น 2 ส่วนด้วยกัน เนื่องจากผู้ใช้สามารถติดต่อกับระบบได้ผ่านทั้งทางคอนโซลพอร์ตอนุกรมและอีเทอร์เน็ทพอร์ต จึงมีการสร้างส่วนที่ติดต่อกับผู้ใช้ 2 แบบ ดังนี้

(1) ส่วนที่ติดต่อกับผู้ใช้ทางคอนโซลพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-10 ส่วนที่ติดต่อกับผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม

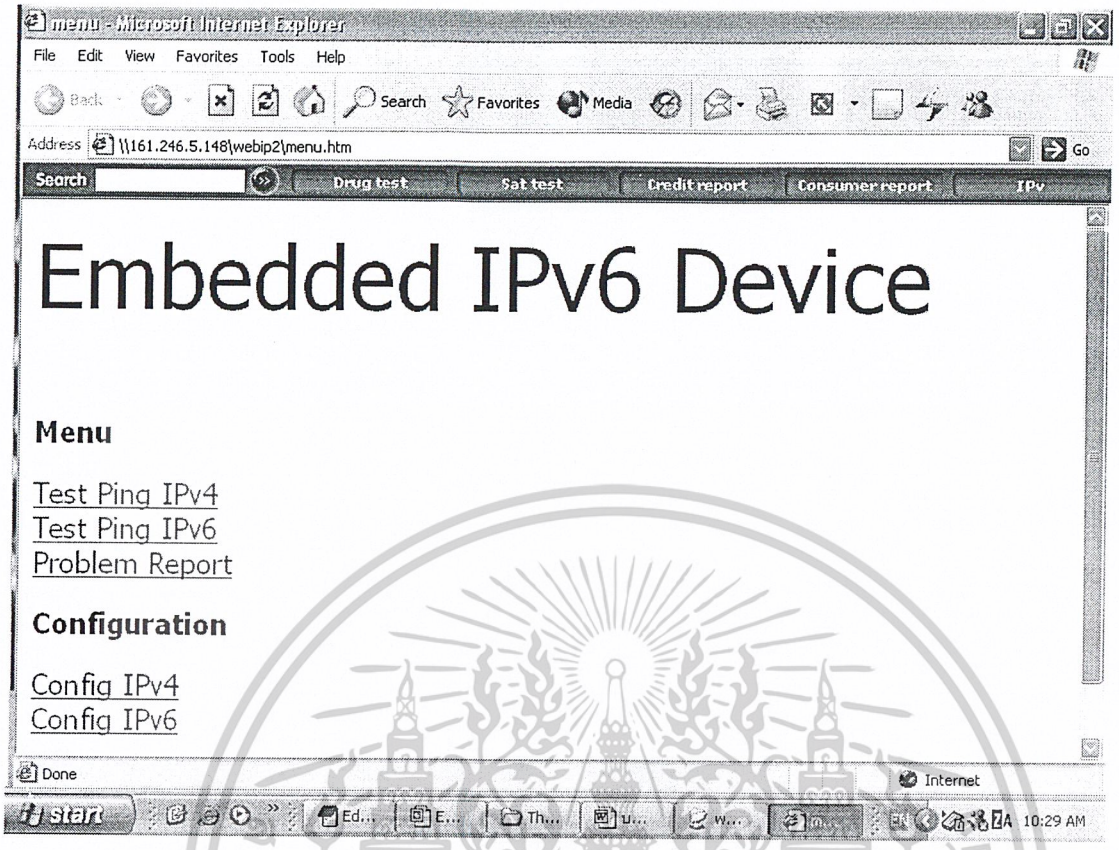
โดยผู้ใช้ที่ทำการติดต่อกับอุปกรณ์ผ่านทางคอนโซลพอร์ตอนุกรมนั้นจะต้องติดต่อผ่านเครื่องคอมพิวเตอร์ส่วนบุคคลที่เชื่อมต่อกับอุปกรณ์นี้อยู่ผ่านโปรแกรมเทอร์มินัล ซึ่งส่วนที่ติดต่อกับผู้ใช้นั้นแสดงดังภาพที่ 4-10 จะเห็นได้ว่าผู้ใช้สามารถเรียกใช้บริการจากเมนูที่ผู้พัฒนาได้กำหนดไว้ให้แล้ว

(2) ส่วนที่ติดต่อกับผู้ใช้ทางอีเทอร์เน็ท

ผู้ใช้ที่ทำการติดต่อกับอุปกรณ์ผ่านทางอีเทอร์เน็ทนั้นสามารถติดต่อสื่อสารจะที่ไหนก็ได้ในระบบเครือข่าย ดังภาพที่ 4-11 ผู้ใช้สามารถติดต่อขอใช้บริการโดยใช้โปรแกรมบราวเซอร์ทั่วไปเรียกส่วนที่ติดต่อกับผู้ใช้ขึ้นมาทำงาน

เนื่องจากข้อจำกัดในเรื่องของหน่วยความจำของตัวชุดพัฒนาคอม 86 และความเร็วในการประมวลผลจึงต้องทำส่วนที่ติดต่อกับผู้ใช้ออกมาในรูปแบบของเท็กซ์เบส (Text base) ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 ส่วนที่ติดต่อกับผู้ใช้ผ่านทางอีเทอร์เน็ท

4.3 การพัฒนาไอพีรุ่น 6 บนระบบฝังตัว

เนื่องจากระบบฝังตัวที่ไม่ใช่พีซีนั้นมีข้อจำกัดในหลายๆด้านด้วยกัน ไม่ว่าจะเป็นเรื่องของหน่วยความจำ และการประมวลผล ซึ่งมีความแตกต่างจากคอมพิวเตอร์ส่วนบุคคลอยู่พอสมควร การที่จะพัฒนาไอพีรุ่น 6 ทั้งหมดตามที่มาตรฐานกำหนดนั้นเป็นเรื่องยากมาก ซึ่งผู้วิจัยได้เล็งเห็นว่าการพัฒนาอุปกรณ์ฝังตัวที่เป็นอุปกรณ์ปลายทาง (Terminal device) ซึ่งมีความสามารถเฉพาะงานนั้นๆและเป็นโฮสต์ไม่ใช่เราเตอร์ให้สามารถที่จะทำงานบนเครือข่ายไอพีรุ่น 6 นั้น ควรมีการพัฒนาสิ่งต่างๆ ดังนี้

4.3.1 การปรับค่าต่างๆของฮาร์ดแวร์

การปรับค่าต่างๆของฮาร์ดแวร์ (Hardware configuration) ควรมีการคำนึงถึงสิ่งเหล่านี้

- อุปกรณ์ดังกล่าวอาจมีหนึ่งเน็ตเวิร์คอินเทอร์เฟสหรือหลายเน็ตเวิร์คอินเทอร์เฟส แต่ถ้ามีหลายเน็ตเวิร์คอินเทอร์เฟสจะต้องมีการคำนึงถึงสิ่งอื่นๆ เพิ่มเติม ซึ่งจะกล่าวต่อไป
- ในที่นี้ถือว่าอีเทอร์เน็ท (Ethernet) และโปรโตคอลพีพีพี (Point-to-Point Protocol) นั้นเป็นส่วนพื้นฐานที่ต้องมีอยู่แล้ว
- ไม่จำเป็นที่จะต้องมีการบู้ตผ่านระบบเครือข่าย เพราะถือว่าอุปกรณ์เหล่านี้ทำงานแบบแสตนอะโลน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าต่างๆที่ควรจะกำหนดไว้ตั้งแต่อุปกรณ์เริ่มทำงานจนหยุดทำงานนั้น ควรจะเป็นค่าที่ต้องใช้ตลอดการทำงานได้แก่ ค่า hop limit ในเฮดเดอร์ แอดเดรสของดีเอ็นเอสเซิร์ฟเวอร์ เป็นต้น

4.3.2 Internet Protocol, Version 6 (IPv6) Specification [rfc2460]

ตัวอุปกรณ์นั้นสามารถที่จะรับแพ็กเก็ตที่มีเฮดเดอร์ส่วนขยายแบบใดๆก็ได้ โดยการพัฒนานั้นควรที่จะคำนึงถึงสิ่งเหล่านี้

- ถ้าตัวอุปกรณ์นั้นไม่มีความจำเป็นที่จะต้องใช้ฟังก์ชันของเฮดเดอร์ส่วนขยาย (extension header function) เราก็สามารถที่ตัดการพัฒนาส่วนที่ใช้ในการตรวจสอบเฮดเดอร์ส่วนขยายได้
- ถ้ารับแพ็กเก็ตที่มีเฮดเดอร์ส่วนขยายเข้ามาควรมีการตรวจสอบก่อนเสมอว่าเป็นเฮดเดอร์ส่วนขยายที่เรารองรับหรือไม่ ถ้าไม่ก็สามารถทิ้งแพ็กเก็ตนั้นไปได้

โดยการจัดการกับเฮดเดอร์ส่วนขยายแบบต่างๆนั้น ควรมีการกระทำดังนี้

4.3.2.1 Hop-by-Hop Options Header

- ตัวอุปกรณ์ต้องสามารถที่จะส่ง Hop-by-Hop Options header สำหรับในกรณีที่ต้องการสนับสนุน Multicast Listener Discovery (MLD) เท่านั้น
- ตัวอุปกรณ์นั้นควรเก็บเลขออพชันไว้ด้วย และจะมีการกระทำเก็บแพ็กเก็ตที่มีเลขออพชันที่ตัวเองสนับสนุนเท่านั้น

4.3.2.2 Routing Header

- อุปกรณ์จะไม่มีการสร้าง Routing header จากตัวเอง
- ในกรณีที่ได้รับแพ็กเก็ตที่มีค่า RH type เป็น 0 และค่าใน segment_left ไม่เป็น 0 แล้วให้ฟอร์เวอร์ดไปที่อินเทอร์มิเดียดโหนดหรือตอบกลับด้วยไอซีเอ็มพีพารามิเตอร์พรอเบลิ้มแทนก็ได้

4.3.2.3 Fragment Header

- ถ้าผู้พัฒนาอุปกรณ์มีการกำหนดไว้และสามารถรับประกันได้ว่าไม่ให้ส่งแพ็กเก็ตที่ขนาดเกินเอ็มทียูที่น้อยที่สุดแล้ว ไม่จำเป็นที่จะต้องมีการทำ fragment header sending function
- ตัวอุปกรณ์ต้องสนับสนุนการ reassemble ให้กับแพ็กเก็ตที่ถูกเฟรกเมนต์มาและทำได้คราวละ 1 แพ็กเก็ตเท่านั้น โดยที่จะต้องมีบัฟเฟอร์ในการเก็บได้อย่างน้อย 2 ส่วนโดยขนาดน้อยกว่าหรือเท่ากับลิ้งค์เอ็มทียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2.4 Destination Options Header

ในการส่งนั้นอุปกรณ์ต้องไม่มีการสร้าง destination options header จากตัวเอง สำหรับการรับแพ็กเก็ตที่เข้ามานั้นเราต้องสนับสนุนออพชันนั้นด้วยถึงจะมีการนำแพ็กเก็ตนั้นไปใช้ต่อไป

4.3.2.5 ไม่มี Next Header

ไม่มีการกระทำใดๆกับเฮดเดอร์ชนิดนี้ทั้งนั้น

4.3.3 Path MTU Discovery for IP version 6

เราต้องสามารถรับประกันได้ว่าจะไม่มีการส่งแพ็กเก็ตที่มีขนาดใหญ่เกินค่าเอ็มทียูที่น้อยที่สุดของไอพีรุ่น 6 จึงจะสามารถเลือกที่จะไม่ทำฟังก์ชันนี้ได้

4.3.4 Neighbor Discovery for IP Version 6 (IPv6) [rfc2461]

สืบเนื่องจากที่กล่าวไว้ในข้างต้นเกี่ยวกับตัวอุปกรณ์ ดังนั้นฟังก์ชันที่เกี่ยวกับเราเตอร์เหล่านี้ไม่จำเป็นที่จะต้องทำขึ้นมา

1. ส่ง Router advertisement messages
2. รับ Router solicitation messages
3. ส่ง Redirect messages

สำหรับการรับ Redirect message นั้นเป็นฟังก์ชันของโฮสต์อยู่แล้ว แต่จะอิมพลีเมนต์หรือไม่ก็ได้ เช่นเดียวกันกับการทำ destination cache ที่จะทำหรือไม่ทำก็ได้ ซึ่งถ้าต้องการที่จะสร้าง destination cache แล้วเรามีความจำเป็นที่จะต้องคำนึงถึงการทำงานของมันร่วมกันกับเนเอบร์อันรีชเอเบิลดีเทคชัน (Neighbor Unreachable Detection : NUD) ด้วย

ถ้าตัวอุปกรณ์นั้นมีหลายเน็ตเวิร์คอินเทอร์เฟสก็ต้องใช้ทรัพยากรมากขึ้นเพราะเราต้องคำนึงสิ่งเหล่านี้เพิ่มเติม

1. ชนิดของข้อมูลในการเราท์ เช่น destination cache และ routing table
2. จำนวนข้อมูลที่เก็บอยู่ในแคชก็มากขึ้นไปด้วย เช่น Neighbor Discovery cache, Default Router list, Prefix list เป็นต้น

4.3.5 IPv6 Stateless Address Autoconfiguration [rfc2462]

ในการทำ Duplicated Address Detection (DAD) ทุกฟังก์ชันนั้นต้องอาศัยฟังก์ชันเนเบอร์ดิสเคิร์ฟเวอรี่ ดังนั้นการที่จะพัฒนาส่วนนี้ขึ้นมานั้นไม่มีผลกระทบต่อทรัพยากรของระบบมาก อาจจะพัฒนาขึ้นมาหรือไม่ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification [rfc2463]

ดังที่กล่าวมาแล้วเกี่ยวกับตัวอุปกรณ์ เราจึงไม่มีความจำเป็นที่จะต้องทำไอซีเอ็มพีแมสเซสที่ใช้โดยกับเราเตอร์ แต่ในส่วนของ ICMP Echo Request/Reply นั้นเป็นพื้นฐานต่ำสุดที่พึงกระทำนอกจากนี้เรายังสามารถเพิ่มฟังก์ชันการทำงานของ ICMPv6 เข้าไปได้ตามความจำเป็น ดังตารางนี้

Type	Code	Necessary
Destination unreachable message	0 : No route to destination	No
	1: Administratively Prohibited	No
	3 : Address unreachable	No
	4 : Port unreachable	Yes
Packet Too Big Message	0 : Packet Too Big	No
Time Exceeded Message	0: Hop limit exceeded	No
	1: Fragment reassembly time exceeded	No
Parameter Problem Message	0: Erroneous header field	Yes
	1: Unrecognized next heade	Yes
	2: Unrecognized IPv6 option	Yes
Echo Request Message	0: Echo request	Yes
Echo Reply Message	0: Echo reply	Yes

ตารางที่ 4-1 แสดงเกี่ยวกับไอซีแมสแซสรุ่น 6 สำหรับระบบฝังตัว

4.3.7 IP Version 6 Addressing Architecture [rfc2373]

โดยปกติแล้วตัวอุปกรณ์นั้นจะต้องมีแอดเดรสเหล่านี้

1. Loopback (::1)
2. Node-local all nodes multicast (ff01::1)
3. Link-local all nodes multicast (ff02::1)

แต่ถ้าต้องการให้มีการทำ address autoconfiguration ตัวอุปกร์ควรจะมีสนับสนุนแอดเดรสต่อไปนี้ด้วยเช่นกัน

1. Link-local unicast (fe80/64 host ID)
2. Solicited-node multicast ที่เกี่ยวข้องกับ address autoconfiguration
3. Unicast (prefix/64 host ID) ที่เกี่ยวข้องกับพรีฟิกส์ออพชันของ router advertise messages
4. Solicited-node multicast ที่เกี่ยวข้องกับ address autoconfiguration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต้องการให้มีการทำ manual configuration ด้วยควรมีการสนับสนุนแอดเดรสดังต่อไปนี้

1. Arbitrary unicast
2. Solicited-node multicast ที่เกี่ยวข้องกับ manual configuration

4.3.8 การพิจารณาเกี่ยวกับขอบเขต

ถึงแม้ว่าตัวอุปกรณ์จะมีแค่ 1 ฟิสิคัลอินเทอร์เฟส เราก็สามารถที่จะกำหนดให้มีได้หลายลอจิคัลอินเทอร์เฟสก็ได้ เช่น ลูปแบ็คอินเทอร์เฟส (Loopback interface) เป็นต้น ได้เช่นกัน

4.3.9 การเลือกแอดเดรสหลักสำหรับไอพีรุ่น 6

ตัวอุปกรณ์จะต้องสนับสนุนการเลือกแอดเดรสหลักสำหรับไอพีรุ่น 6 โดยควรพิจารณา

1. ถ้าตัวอุปกรณ์ไม่มีการสนับสนุน ICMP redirect message เราสามารถตัดส่วนของกระบวนการตรวจสอบออกไปได้
2. ในการเลือกแอดเดรสต้นทางหลักควรพิจารณาถึงว่าเรามีการสนับสนุนโมบิลิตี้หรือไม่ มีกี่อินเทอร์เฟส มีการทำ policy table หรือไม่ รวมทั้งสิ่งอื่นที่สนับสนุนและไม่สนับสนุนด้วยเช่นกัน
3. ในการเลือกแอดเดรสปลายทางหลักนั้นก็ต้องมีการคำนึงถึงเรื่องการสนับสนุนโมบิลิตี้ policy table รวมทั้งสิ่งอื่นที่สนับสนุนและไม่สนับสนุนด้วยเช่นกัน

4.3.10 Multicast Listener Discovery (MLD) for IPv6 [rfc2710]

เมื่อตัวอุปกรณ์นี้มีความจำเป็นที่จะต้องเข้าไปร่วมกับกลุ่มมัลติคาสต์ฟังก์ชันนี้เป็นสิ่งที่พึงกระทำ

4.3.11 IPsec สำหรับระบบฝังตัว

ไอพีรุ่น 6 ใช้โครงสร้างของ IPsec ในการเป็นพื้นฐานทางด้านความปลอดภัย ซึ่งมีด้วยกันหลายอย่างได้กัน แต่สำหรับระบบฝังตัวนั้นไม่มีความจำเป็นที่ต้องนำทั้งหมดมาใช้ โดยสามารถพิจารณาได้ดังนี้

4.3.11.1 IPsec specification สำหรับระบบฝังตัว

ตัวอุปกรณ์นั้นต้องมีการทำ IPsec ในโหมดทรานสปอร์ต โดยอาจจะทำในหมดเทอร์นเนลด้วยก็ได้ แต่ไม่จำเป็นสำหรับอุปกรณ์ที่เป็นโฮสต์ที่ไม่ได้มีการสื่อสารผ่านเซคเคียวริตี้เกตเวย์ และพารามิเตอร์บางอย่างของ Security Assocication อาจจะไม่ต้องทำก็ได้ เช่น automated key ซึ่งส่วนนี้ขึ้นอยู่กับความต้องการของนักพัฒนา

4.3.11.2 Security Policy Database

อย่างน้อยที่สุดต้องมีการทำในส่วนที่เกี่ยวข้องกับไอพีแอดเดรสขึ้นมา สำหรับส่วนอื่นๆนั้นจะทำหรือไม่ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.11.3 Security Protocol

นักพัฒนาจะต้องมีการพัฒนา IP Encapsulating Security Payload (ESP) [rfc2406] และ IP Authentication Header (AH) ขึ้นมาโดยการใช้นั้นจะใช้แค่อันใดอันหนึ่งหรือทั้งคู่ก็ได้

4.3.11.4 การทรานสฟอร์มและอัลกอลิทึม

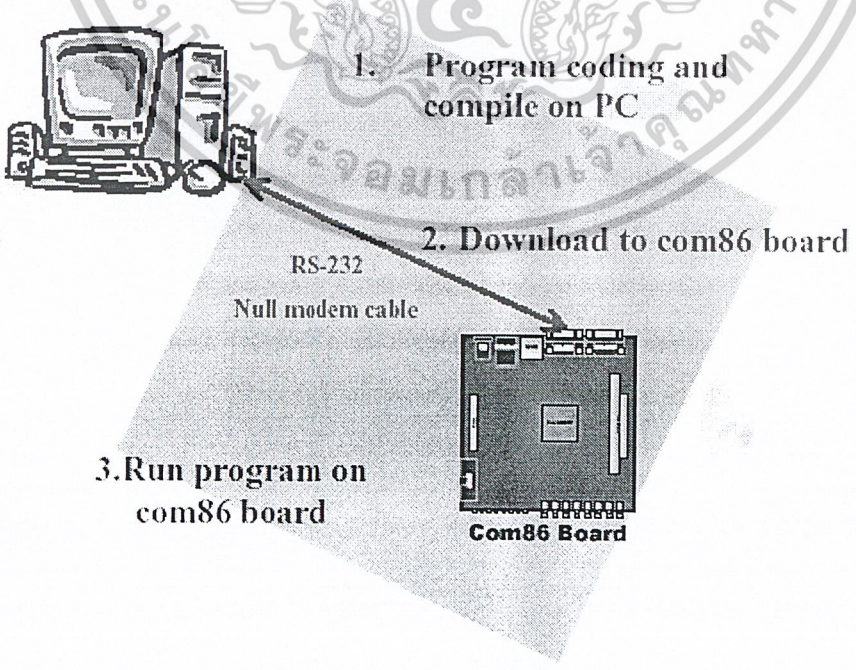
ในกรณีของการพัฒนาเอ็นคริปท์ อัลกอริทึม AES-CBC นั้นเป็นสิ่งที่ต้องกระทำ แต่สำหรับอัลกอริทึมอื่นๆ เช่น DES-CBC และ 3DES-CBC นั้นจะทำเฉพาะ backward compatibility หรือไม่ทำก็ได้ สำหรับอัลกอริทึม NULL Encryption อาจจะทำไว้ใช้ในการดีบักก็ได้ ส่วนการทำออเทนทิเคชันนั้นมีอัลกอลิทึมที่แนะนำว่าน่าจะทำ คือ HMAC-SHA-1-96 [rfc2404] และ HMAC-MD5-1-96 [rfc2403]

4.3.11.5 SA และการจัดการเกี่ยวกับคีย์

สำหรับ manual key management นั้นเป็นสิ่งที่ต้องพัฒนาขึ้นมา โดยมีจุดประสงค์ที่จะสุ่มค่าของอินเทอร์เฟสไอดีขึ้นมา ในส่วนของ automated key management นั้นอาจจะทำ IKE ก็ได้ ตามแต่ต้องการ

4.4 การพัฒนาโปรแกรมบนชุดพัฒนาคอม86

การพัฒนาโปรแกรมบนชุดพัฒนาคอม86 สามารถพัฒนาได้จากหลายภาษาด้วยกันอาทิเช่น ภาษาซี (C) หรือซีพลัสพลัส (C++) ภาษาปาสคาล (Pascal) ภาษาแอสเซมบลี (Assembly) เป็นต้น โดยการพัฒนาโปรแกรมสามารถทำได้บนเครื่องคอมพิวเตอร์บุคคล (Personal computer) แล้วทำการส่งไฟล์ที่ผ่านการคอมไพล์และสร้างเป็น ไฟล์ที่สามารถรันได้บนระบบปฏิบัติการดอส (executable file) ไปยังบอร์ดชุดพัฒนา แล้วทำการรัน รูปที่4-12



รูปที่ 4-12 ขั้นตอนพื้นฐานในการพัฒนาโปรแกรมทั่วไปบนชุดพัฒนาคอม 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่จะส่งไฟล์ไปไว้บนบอร์ดชุดพัฒนาได้นั้น จะส่งผ่านทางโปรแกรมเทอร์มินัลทั่วไป โดยในชุดพัฒนาคอม86 ได้มีโปรแกรมยูทิลิตี้ตัวหนึ่งในการรองรับการทำงานในส่วนนี้เป็นชื่อโปรแกรมว่า transfer.exe เป็นโปรแกรมที่ช่วยในการรับส่งไฟล์ต่างๆ ผ่านทางพอร์ตอนุกรม ซึ่งจะใช้โปรโตคอล เอ็กซ์-โมเด็ม (X-modem) ในการรับส่งไฟล์ที่ความเร็ว 57600 bps ซึ่งรูปแบบการใช้งานโปรแกรมแสดงดังรูปที่ 4-10

```
TRANSFER [/S | /R| [/H]|/?] filename
where   /S   To send file
        /R   To receive file
             /?   Show this help
             /H   Show this help
the default option is /R
```

ตัวอย่างการเรียกใช้งาน

```
A:\>Transfer program1.exe
```

รูปที่ 4-13 รูปแบบการใช้โปรแกรมในการรับส่งไฟล์กับชุดพัฒนาคอม 86

4.4.1 การใช้งานพอร์ตอนุกรมของชุดพัฒนาคอม86

สำหรับพอร์ตอนุกรมบนชุดพัฒนาคอม86 ในที่นี้หมายถึงพอร์ตคอม2(COM2) ซึ่งโดยมากจะใช้ไปควบคุมหรือติดต่อกับอุปกรณ์อื่นๆ ต่อไป โดยคุณสมบัติทั่วไปของพอร์ตอนุกรมนี้ได้แก่

- สามารถรับส่งข้อมูลได้ที่อัตราเร็วสูงสุด 115200 bps
- การส่งข้อมูลเป็นแบบสองทิศทางในเวลาเดียวกัน(Full-duplex)
- ใช้บิตที่เป็นสต็อปบิต 1 หรือ 2 บิตก็ได้

ในการใช้งานพอร์ตอนุกรมนี้ผู้เขียนโปรแกรมจำเป็นต้องทราบรายละเอียดของรีจิสเตอร์ตัวต่างๆ ของพอร์ตว่าต้องมีการส่งค่าใดไปให้บ้าง นอกจากนี้ยังจะต้องทราบอัตราเร็วของการรับส่งข้อมูลผ่านทางพอร์ตอนุกรมของอุปกรณ์ที่ชุดพัฒนาจะไปต่อเชื่อมด้วย สำหรับรายละเอียดของรีจิสเตอร์ต่างๆที่เกี่ยวข้องกับการรับส่งข้อมูลผ่านซีเรียลพอร์ต แสดงดังรูป 4-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPCON0	Serial port control 0
SPCON1	Serial port control 1
SPSTAT	Serial port status
SPIMSK	Serial port Interrupt Mask
SPTXD	Serial port Transmit data
SPRXD	Serial port Receive data
SPRXDP	Serial port Receive data peak
SPBDV	Serial port Baud Rate Divisor

รูปที่ 4-14 รีจิสเตอร์ที่เกี่ยวข้องในการเขียนโปรแกรมเพื่อติดต่อผ่านทางพอร์ตอนุกรม

4.4.2 การเขียนโปรแกรมเพื่อให้ชุดพัฒนาคอม86 สามารถเชื่อมต่อเข้าสู่เครือข่าย

การที่จะให้ชุดพัฒนาคอม86 สามารถเชื่อมต่อเข้าสู่ระบบเครือข่ายท้องถิ่น (Local Area Network) นั้น ในขั้นแรกจะต้องทำการต่อสายยูทีพีเข้ากับชุดพัฒนาคอม86 จากนั้นต่อสายซัพพลายเข้ากับชุดพัฒนา และทำการโหลดโปรแกรมแพ็กเกตไดร์เวอร์ โดยพิมพ์คำว่า "pnppd 0x60" ที่คอมมานไลน์ เมื่อสิ้นสุดขั้นตอนนี้ ชุดพัฒนาคอม86 จะสามารถเชื่อมต่อ เข้าสู่เครือข่ายได้ โดยสามารถทดสอบว่าชุดพัฒนาคอม86 เชื่อมต่อเข้าสู่เครือข่ายได้ โดย โปรแกรม ping ที่อยู่ในไดเร็กทอรี a:/demo โดยการใช้โปรแกรม ping มีรูปแบบของคำสั่งดังนี้ "ping [หมายเลขไอพีที่ต้องการทดสอบ]"

4.5 การพัฒนาโปรแกรมบนไลท์เวททีซีพี/ไอพีแสต็ก

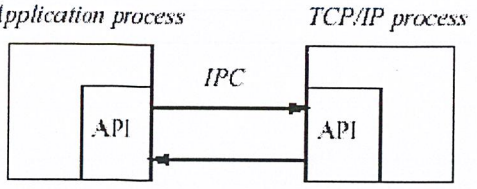
การที่จะติดต่อสื่อสารกับตัวโปรโตคอลแสต็กนั้นมีด้วยกัน 2 วิธี คือ

1. การเรียกฟังก์ชันในการทำงานโดยตรง ซึ่งวิธีนี้จะทำให้โปรแกรมที่พัฒนาขึ้นมีความซับซ้อนและเข้าใจได้ยาก

2. ใช้เอลดับเบิลยูไอพีเอพีไอ (lwIP API) ซึ่งมีการทำงานคล้ายคลึงกันกับบีเอสดีซ็อกเก็ตเอพีไอ (BSD socket API) โดยตัวเอพีไอนั้นแบ่งออกเป็น 2 ส่วนด้วยกันตามลักษณะของโพรเซสโมเดลดังที่ได้กล่าวมาก่อนหน้านี้ โดยส่วนหนึ่งจะเป็นของทีซีพี/ไอพีโพรเซสอีกส่วนหนึ่งจะเป็นของแอพพลิเคชันโพรเซส

ซึ่งการสื่อสารจะใช้วิธีอินเทอร์โพรเซสคอมมิวนิเคชัน (interprocess communication: IPC) ซึ่งทำงานโดยอีมูเลชันเลเยอร์ของระบบปฏิบัติการของไลท์เวททีซีพีไอพีแสต็กทำให้ไม่ต้องอาศัยการทำงานของระบบปฏิบัติการ โดยอินเทอร์โพรเซสคอมมิวนิเคชันนั้นได้ใช้กระบวนการของแชร์เมมโมรี่ (shared memory) แมสเซสพาสซิ่ง (message passing) และเซมาฟอร์ (semaphores)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



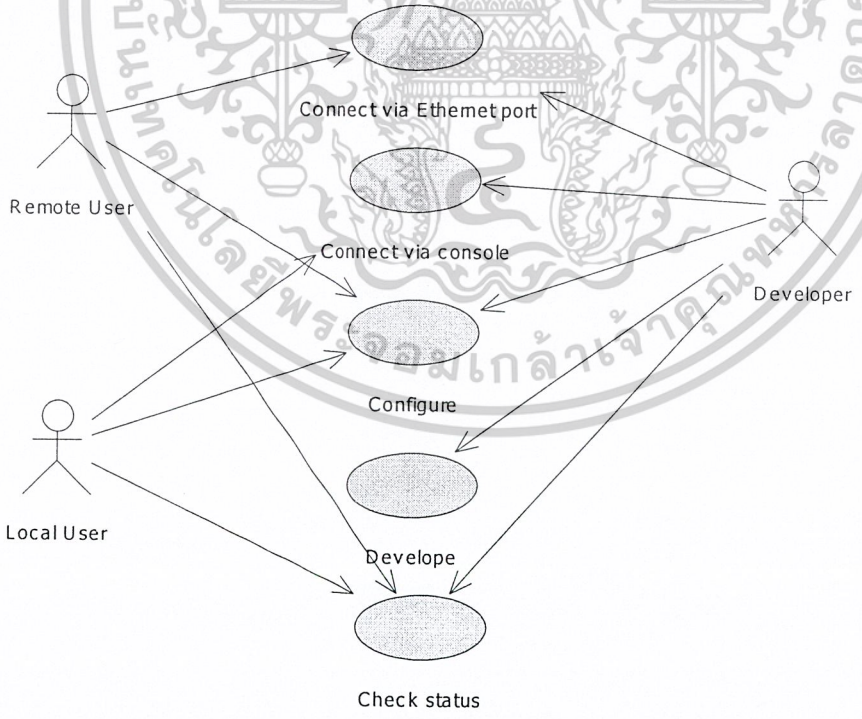
รูปที่ 4-15 การสื่อสารกันผ่านเอพีไอ

โดยใช้แชร์เมมโมรี่ในการส่งข้อมูลในบัฟเฟอร์ข้ามกันระหว่างแอพพลิเคชันโพรเซสและทีซีพี/ไอพีโพรเซส ซึ่งมีการใช้เซมาฟอร์เพื่อทำเมมโมรี่โพรเทคชัน สำหรับฟังก์ชันที่ทำงานเกี่ยวกับเน็ทเวิร์คคอนเนคชันนั้นจะสื่อสารกันด้วยการใช้กระบวนการของแมสเซสพาสซิ่งในการส่งเมสแซสเกี่ยวกับโอเปอร์เรชันต่างๆและอาร์กิวเมนต์ของโอเปอร์เรชันนั้นๆ ให้แก่กันและส่งผลลัพธ์กลับมาด้วยวิธีการเดียวกัน

4.6 การออกแบบระบบ

4.6.1 Use case diagram

ไดอะแกรมนี้แสดงฟังก์ชันการทำงานหลักๆ ที่สำคัญของระบบ และมีรายละเอียดแสดงไว้ดังนี้



รูปที่ 4-16 Use case diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connect via Ethernet port

Use case	Connect via Ethernet port
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้ต้องสื่อสารผ่านโปรโตคอลเอชทีทีพีด้วยการใช้บราวเซอร์

Connect via console

Use case	Connect via console
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้ต้องสื่อสารผ่านพอร์ตอนุกรมด้วยโปรแกรมเทอร์มินัล

Configure

Use case	Configure
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้งานจะตั้งค่าคุณสมบัติให้กับระบบได้ก็ต่อเมื่อมีการเชื่อมต่อเข้ามาในระบบแล้วเท่านั้น โดยการปรับแต่งค่าคุณสมบัตินั้นจะแยกออกเป็น 2 ส่วนคือ - ผ่านทางคอนโลพอร์ตอนุกรม - ผ่านทางอีเทอร์เน็ต

Develope

Use case	develop
ผู้ใช้	Developer
เงื่อนไข	การนำไปพัฒนาต่อนั้นขึ้นกับลักษณะงานที่นำไปใช้และข้อจำกัดของระบบ

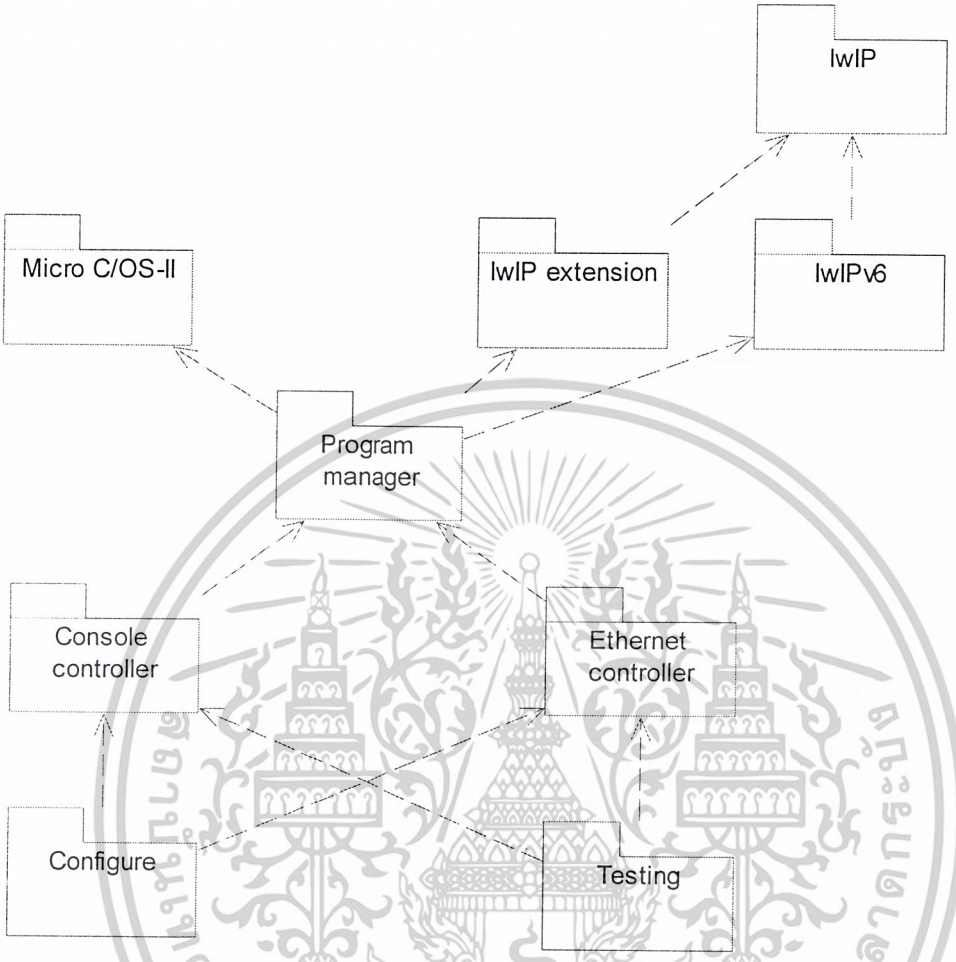
Check status

Use case	Check status
ผู้ใช้	Remote user, Local user, Developer
เงื่อนไข	ผู้ใช้สามารถตรวจสอบสถานะของการทำงานได้โดยใช้เครื่องมือที่มีให้หรือแก้ไขค่าในไฟล์ lwipopts.h เพื่อให้แสดงสถานะอื่นๆเพิ่มเติมได้

ตารางที่ 4-2 อธิบาย Use case diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 Component diagram



รูปที่ 4-17 Component diagram ของระบบ

แผนภาพด้านบนนั้นแสดงถึงองค์ประกอบต่างๆของซอฟแวร์ของระบบ ซึ่งประกอบด้วย

1. Micro C/OS-II เป็นส่วนของเรียลไทม์โอเอส จัดการเกี่ยวกับทาสก์ต่างๆ ที่สร้างขึ้น รวมทั้งการสื่อสารระหว่างทาสก์โดยวิธีการต่างๆ เช่น semaphore, message boxes เป็นต้น

2. lwIP เป็นส่วนของโปรโตคอลแสต็ก ซึ่งทำหน้าที่ให้บริการเกี่ยวกับการสื่อสารผ่านเครือข่าย และการทำงานที่เกี่ยวข้องกับระบบเครือข่าย ตัวอย่างโปรโตคอลที่สนับสนุนได้แก่ ไอพี ทีซีพี ยูดีพี เอชทีทีพี เป็นต้น

3. Console controller จัดการเกี่ยวกับการติดต่อสื่อสารกับผู้ใช้กับระบบผ่านทางคอนโซลพอร์ตอนุกรม รวมทั้งการรับอินพุตเข้ามาและส่งเอาท์พุตออกไปแสดงผลด้วย

4. Ethernet controller จัดการติดต่อสื่อสารกับผู้ใช้กับระบบผ่านทางอีเทอร์เน็ท โดยการใช้โปรโตคอลเอชทีทีพี รวมทั้งการรับอินพุตเข้ามาและส่งเอาท์พุตออกไปแสดง

5. Configure ทำหน้าที่ในการกำหนดค่าคุณสมบัติต่างๆ ให้กับระบบ โดยรับค่าต่างๆ จากผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรมหรืออีเทอร์เน็ท เช่น การกำหนดไอพีแอดเดรสให้กับเน็ทเวิร์คอินเทอร์เฟส

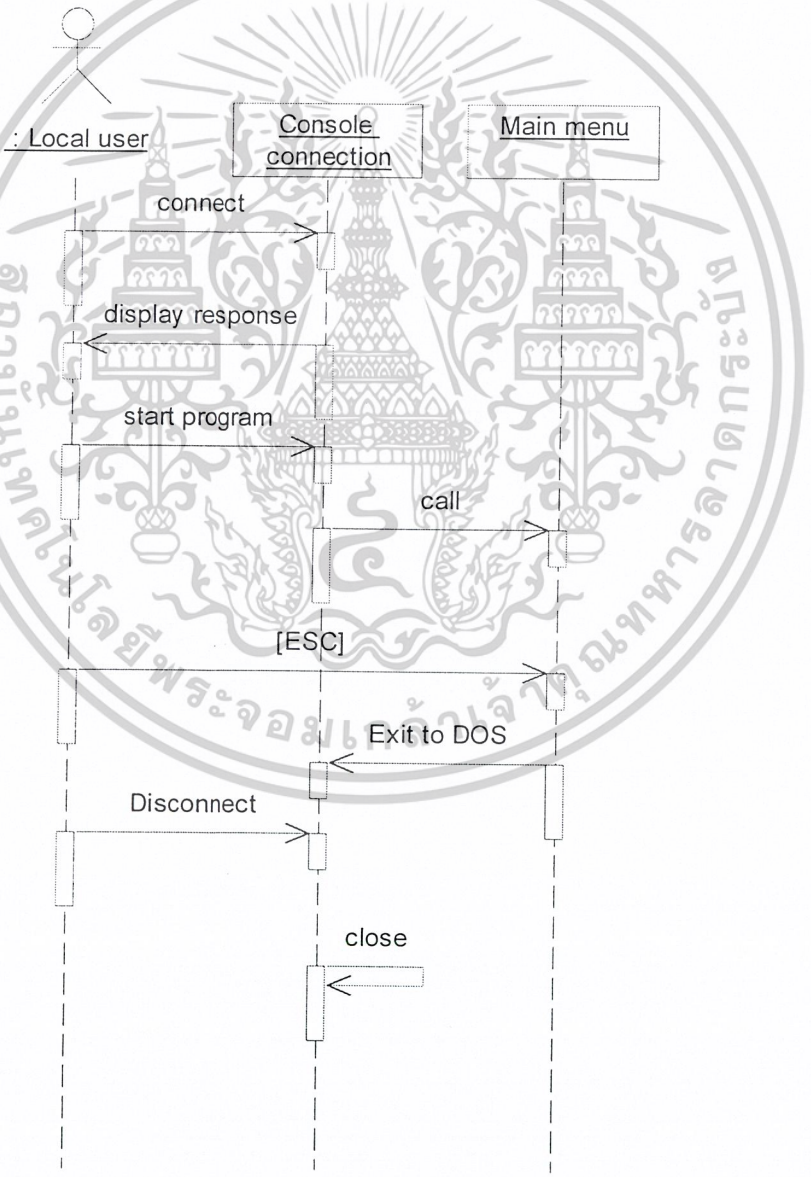
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Testing เป็นส่วนที่ใช้ในการทดสอบการทำงานว่าระบบสามารถทำงานได้ถูกต้องตามต้องการหรือไม่ เช่น ทดสอบว่าระบบสามารถทำงานในเครือข่ายได้ โดยการใช้โปรแกรมประยุกต์ ping มาที่ไอพีแอดเดรสที่กำหนดไว้ให้กับอุปกรณ์ แล้วดูว่าได้รับการตอบกลับหรือไม่ เป็นต้น

4.6.3 Sequence diagram

เป็นไดอะแกรมที่จำลองการทำงานของระบบ เพื่อแสดงให้เห็นถึงการทำงานที่เป็นลำดับขั้นตอนและการทำงานร่วมกันของอ็อบเจ็คต่างๆ ในระบบ

4.6.3.1 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม

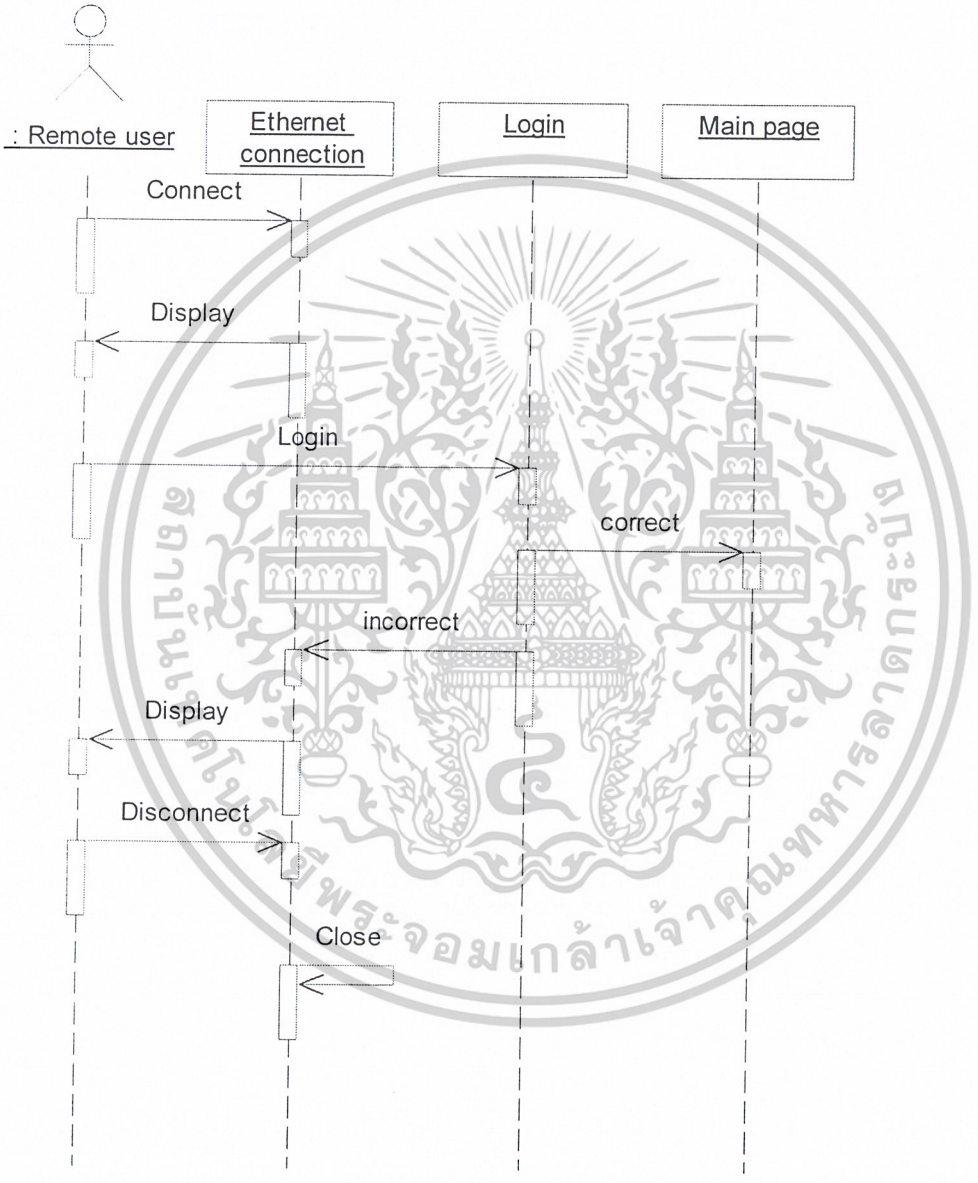


รูปที่ 4-18 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางคอนโซลพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-18 เมื่อผู้ใช้ติดต่อเข้ามาในระบบผ่านทางโปรแกรมเทอร์มินัลแล้ว ผู้ใช้สามารถเลือกการใช้บริการจากเมนูหลักก่อน แล้วทำตามขั้นตอนที่ระบบได้วางไว้ โดยถ้าต้องการออกจากระบบสามารถกดปุ่ม ESC ได้

4.6.3.2 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางอีเทอร์เน็ท



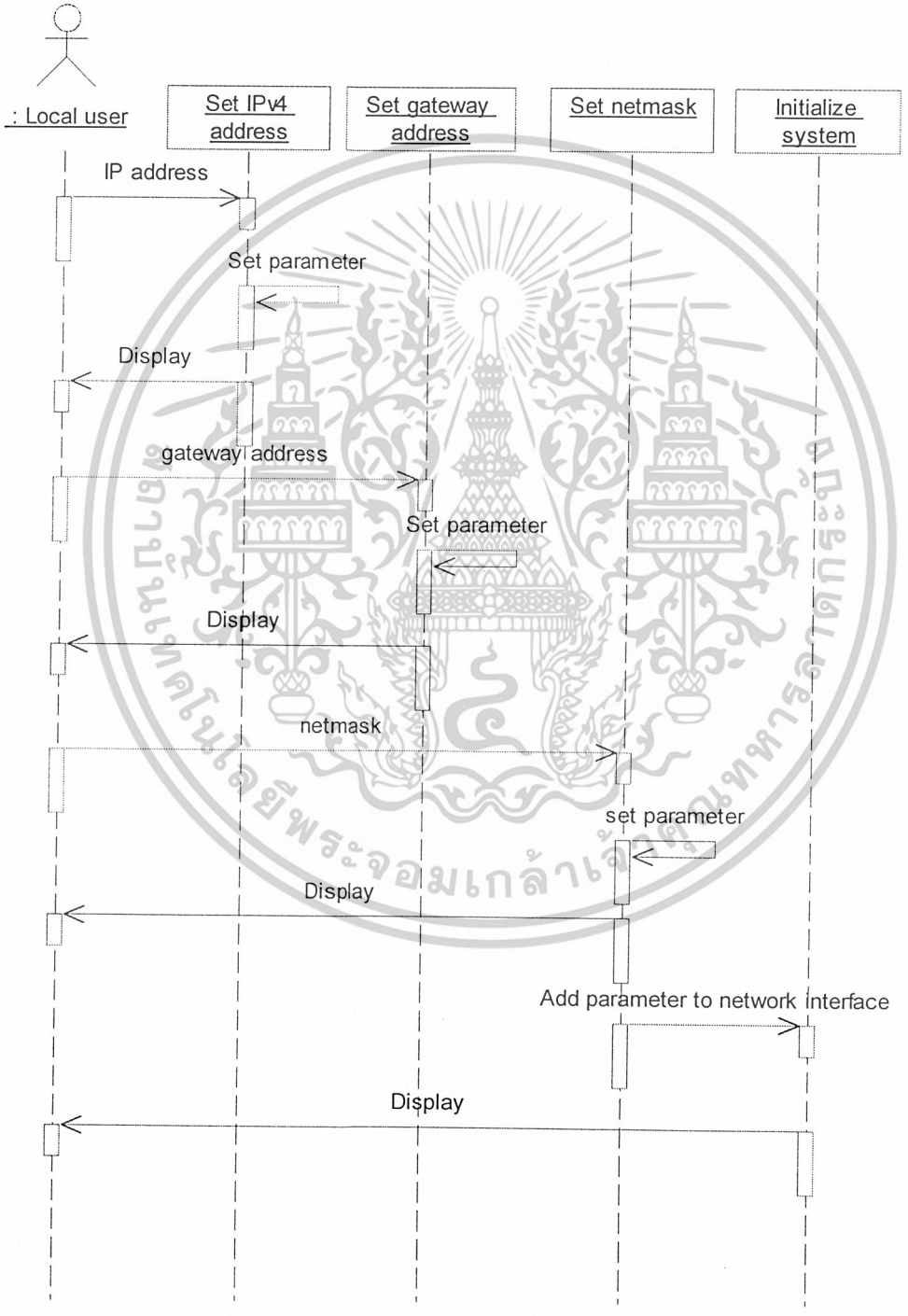
รูปที่ 4-19 Sequence diagram แสดงการติดต่อเข้ามาใช้งานของผู้ใช้ผ่านทางอีเทอร์เน็ท

เมื่อผู้ใช้ต้องการติดต่อเข้ามาในระบบจะต้องมีการ log in จากภายนอกเข้ามาโดยการใช้โปรแกรมบราวเซอร์ขึ้นมาทำงาน เมื่อเข้ามาในระบบแล้วก็จะมีเมนูหลักให้เลือกใช้งาน เมื่อเลือกใช้งานส่วนไหนก็จะต้องทำงานตามขั้นตอนของส่วนนั้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.3.3 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 4

จากภาพที่ 4-20 แสดงขั้นตอนทำงานในการกำหนดค่าคุณสมบัติเกี่ยวกับไอพีรุ่น 4 ให้กับระบบ ผู้ใช้ต้องกระทำการป้อนค่า ไอพีแอดเดรส ตามด้วยค่าของเกตเวย์ และเน็ตมาสก์ตามลำดับ หลังจากที่ป้อนค่าทุกอย่างครบแล้ว ก็จะนำค่าเหล่านี้ไปกำหนดให้กับเน็ตเวิร์คอินเทอร์เฟส และอินนิเชียลค่าระบบขึ้นมาใหม่เพื่อแก้ไขค่าและทำงาน



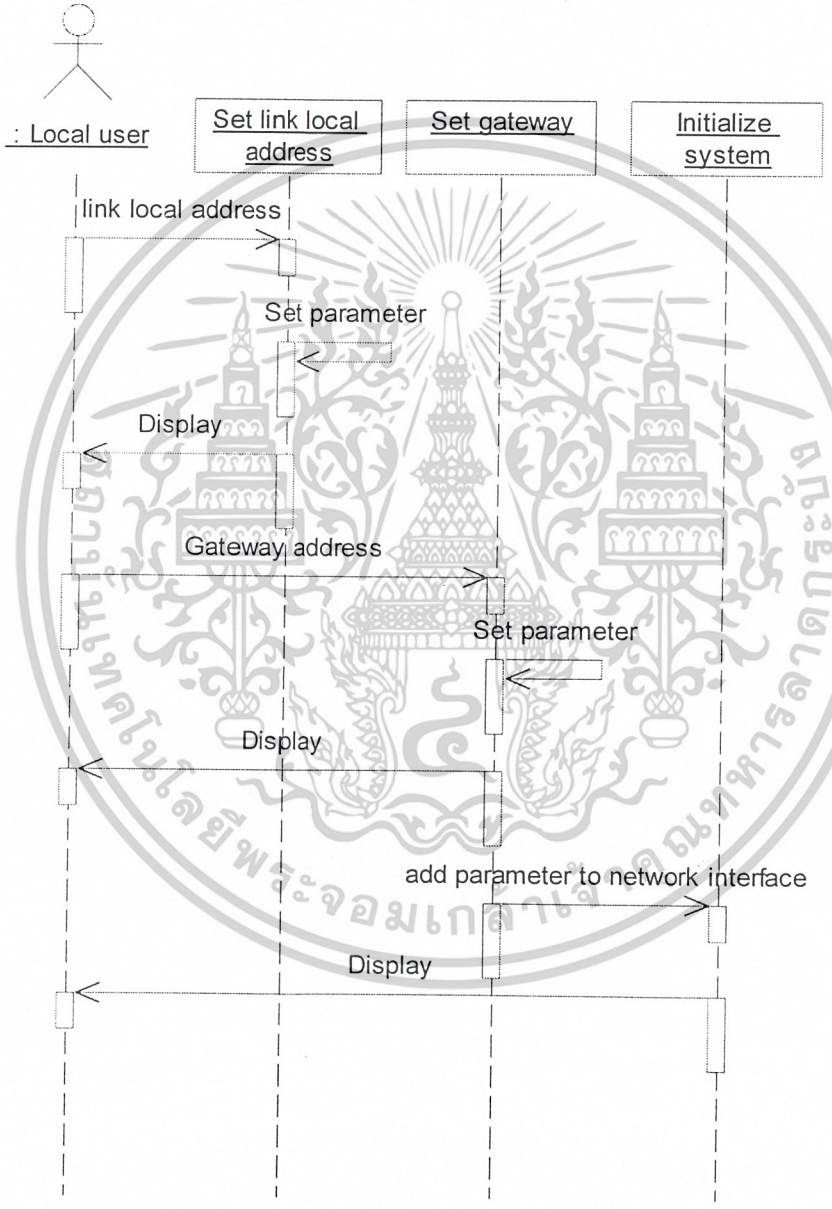
รูปที่ 4-20 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.3.4 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น 6

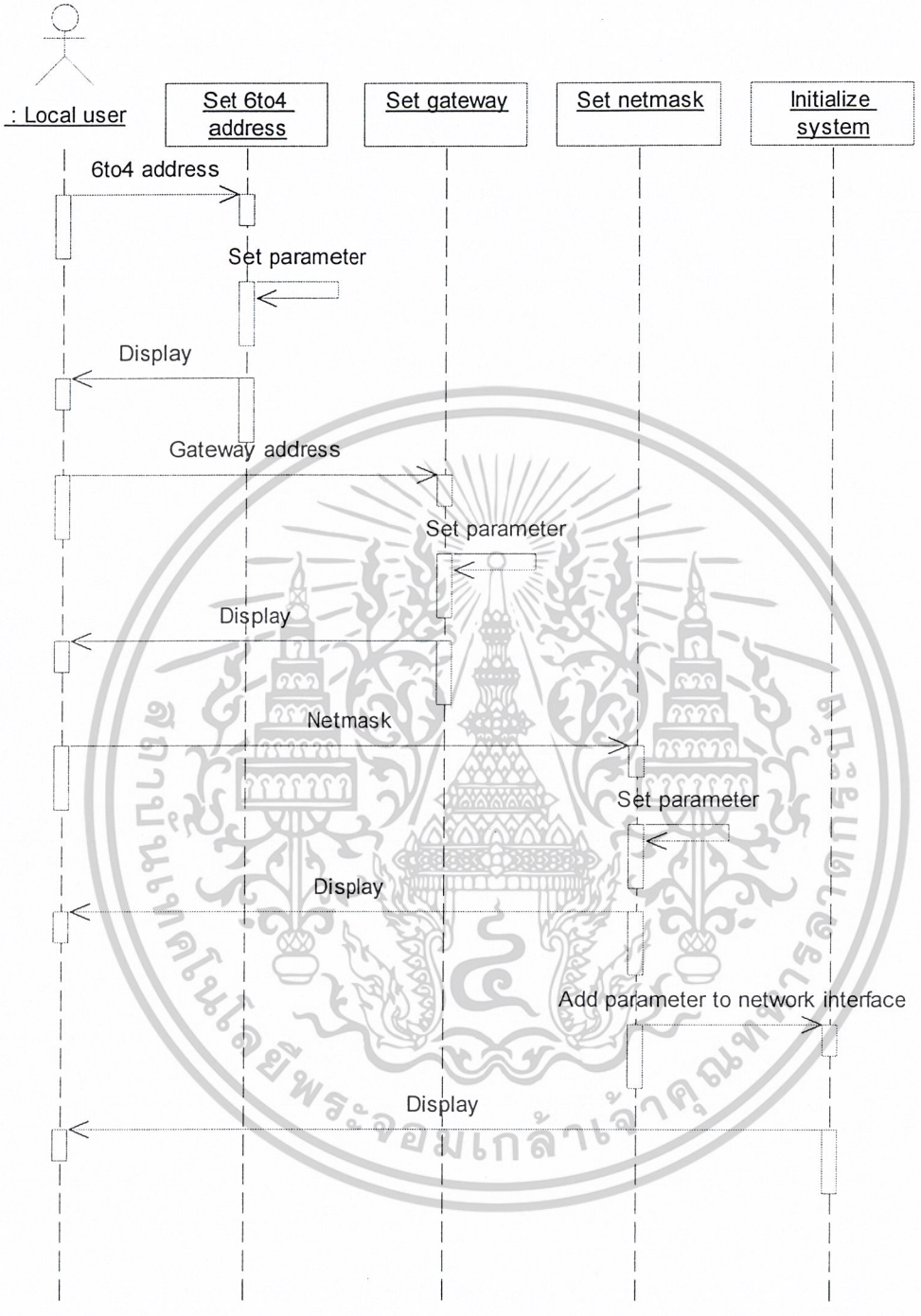
การตั้งค่าคุณสมบัติเกี่ยวกับไอพีรุ่น 6 นั้นแบ่งออกเป็นดังนี้

1. การตั้งค่าคุณสมบัติเกี่ยวกับลิงค์โลคอลแอดเดรส
2. การตั้งค่าคุณสมบัติเกี่ยวกับคอมแพททิเบิลแอดเดรส
   - แบบ 6to4
   - แบบ IPv4 compatible



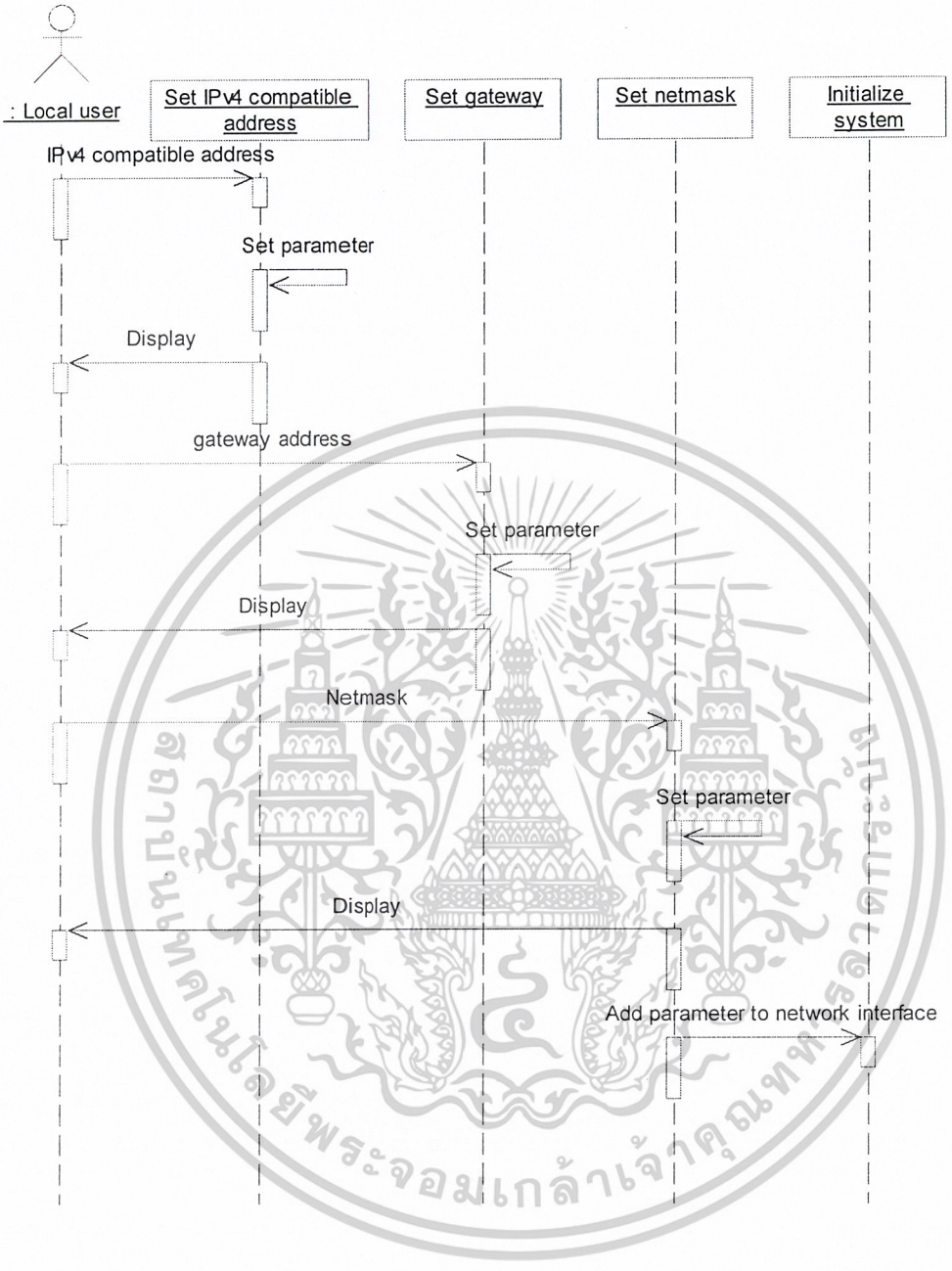
รูปที่ 4-21 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบลิงค์โลคอลแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-22 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบ 6to4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-23 Sequence diagram แสดงการกำหนดค่าพื้นฐานเกี่ยวกับไอพีรุ่น6 แบบ IPv4 compatible

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ลักษณะการใช้งานระบบ

ลักษณะการใช้งานระบบสามารถแบ่งออกได้ดังนี้

1. นักพัฒนา
2. ผู้เรียกใช้บริการจากระบบฝังตัว
3. ผู้พัฒนาไอพีรุ่น 6 บนระบบฝังตัว

5.1 นักพัฒนา

ในลักษณะการใช้งานนี้นักพัฒนาสามารถที่จะนำซอร์ดโค้ดไปพัฒนาต่อ โดยจะสามารถแก้ไขและเพิ่มเติมส่วนต่างๆ ได้อย่างอิสระ และสามารถเพิ่มเติมอุปกรณ์ฮาร์ดแวร์กับซอฟแวร์ที่ช่วยควบคุมการทำงานเข้าไปได้ด้วย โดยโครงสร้างของไดเร็คทอรี่ในการเก็บซอร์ดโค้ดเป็นดังนี้

```
<ROOT DIRECTORY>
+---contrib                 contribution sources tree
|  +---apps
|  \---ports
|     \---am186cx
|        +---proj            project
|        |  \---ucos
|        |     +---apps      lwIP application sources
|        |     |  \---fs
|        |     |     \---img
|        |     \---obj
|        +---include         port include directory
|        |  +---netif        network interface include files
|        |  \---arch         arch dependent include files
|        +---netif           network interface sources
|        +---ucos-ii         MicroC/OS-II sources directory
|        |  +---Port         am186cx ports
|        |  \---Source       MicroC/OS-II sources
|        \---blocks          COM86 support library
\---lwip                     lwIP 0.6.5 sources tree
```

รูปที่ 5-1 โครงสร้างของไดเร็คทอรี่ในการเก็บซอร์ดโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้านักพัฒนาได้พัฒนาในส่วนของโปรแกรมประยุกต์เพิ่มเติมควรนำโปรแกรมประยุกต์เหล่านั้นไปไว้ในส่วนของ \apps โดยให้ส่วนของ main program อยู่ที่ \proj\ucos และต้องทำการแก้ไขไฟล์ที่สร้างและลิงค์ส่วนต่างๆเข้าด้วยกันด้วย และหากต้องการแก้ไขหรือเพิ่มเติมบริการจากระบบที่มีอยู่สามารถเข้าไปแก้ไขไฟล์ lwipopts.h และ os_cfg.h ใน \proj\ucos ได้

```
#define SYS_DEBUG               DBG_OFF

#define MEM_DEBUG               DBG_ON      //DBG_OFF
#define MEMP_DEBUG              DBG_ON      //DBG_OFF
#define PBUF_DEBUG              DBG_ON      //DBG_OFF
#define API_LIB_DEBUG           DBG_OFF
#define API_MSG_DEBUG           DBG_OFF
#define TCPIP_DEBUG             DBG_OFF
#define NETIF_DEBUG             DBG_OFF
#define SOCKETS_DEBUG           DBG_OFF
#define DEMO_DEBUG              DBG_OFF
#define IP_DEBUG                DBG_ON      //DBG_OFF
#define IP_REASS_DEBUG          DBG_OFF
#define RAW_DEBUG               DBG_OFF
#define ICMP_DEBUG              DBG_OFF
#define UDP_DEBUG               DBG_OFF
#define TCP_DEBUG               DBG_ON      //DBG_OFF
#define TCP_INPUT_DEBUG         DBG_OFF
#define TCP_OUTPUT_DEBUG        DBG_OFF
```

รูปที่ 5-2 ตัวอย่างการแก้ไขบางส่วนในไฟล์ lwipopts.h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define OS_MAX_EVENTS                64
#define OS_MAX_FLAGS                 5
#define OS_MAX_MEM_PART              8
#define OS_MAX_QS                    32
#define OS_MAX_TASKS                 32

#define OS_SCHED_LOCK_EN             1
#define OS_TASK_IDLE_STK_SIZE        256

#define OS_TASK_STAT_EN              1
#define OS_TASK_STAT_STK_SIZE        256
#define OS_TASK_STAT_STK_CHK_EN      1

#define OS_TICK_STEP_EN              1
#define OS_TICKS_PER_SEC             200
```

รูปที่ 5-3 ตัวอย่างการกำหนดค่าบางส่วนในไฟล์ os_cfg.h

```
COMPILER_PATH = C:\LANGS\BC
CC = $(COMPILER_PATH)\BIN\BCC +TEST.CFG
TASM = $(COMPILER_PATH)\BIN\TASM
TLIB = $(COMPILER_PATH)\BIN\TLIB
TLINK = $(COMPILER_PATH)\BIN\TLINK
LIBPATH = $(COMPILER_PATH)\LIB
DEFSINCPATH = $(COMPILER_PATH)\INCLUDE
ARCHINCPATH = .\;$(ARCH_BLOCK_DIR);$(ARCH_LWIP_DIR)\INCLUDE;$(UCOS_DIR)\PORT;
LWIPINCPATH =
$(LWIP_DIR)\INCLUDE;$(LWIP_DIR)\INCLUDE\IPV4;$(LWIP_DIR)\INCLUDE\IPV6
UCOSINCPATH = $(UCOS_DIR)\SOURCE
APPSINCPATH = $(APPS_LWIP_DIR)
IP6PATH = $(LWIP_DIR)\CORE\IPV6
```

รูปที่ 5-4 ตัวอย่างการกำหนดค่าบางส่วนในไฟล์ makefile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
REM Must be edit tools path
mkdir .\obj
c:\langs\bc\bin\make -fMakefile
c:\utils\upx\upx test.exe
```

รูปที่ 5-5 ตัวอย่างการกำหนดในไฟล์ maketest.bat

5.2 ผู้เรียกใช้บริการจากระบบฝังตัว

ผู้ใช้ที่ต้องการใช้งานอุปกรณ์นี้เป็นอุปกรณ์ปลายทาง (Terminal device) ต้องเขียนโปรแกรมประยุกต์เพื่อที่จะเรียกใช้งานบริการต่างๆ จากระบบ หรือถ้าต้องการใช้อุปกรณ์นี้เป็นเว็บเซิร์ฟเวอร์ที่ไม่มีการประมวลผลอย่างอื่นที่นอกเหนือจากความสามารถของระบบก็สามารถสร้างเว็บเพ็จ และตั้งค่าต่างๆในไดเร็คทอรี่ apps ให้ทำงานได้ แต่ผู้ใช้ต้องทำการปรับแต่งค่าคุณสมบัติต่างๆให้กับอุปกรณ์ก่อน

```
static err_t
http_recv(void *arg, struct tcp_pcb *pcb, struct pbuf *p, err_t err)
{
    ...
    fs_open("/index.html", &file); หรือ
    fs_open("/404.html", &file);
    ...
}
```

รูปที่ 5-6 ตัวอย่างส่วนที่ต้องปรับแต่งเมื่อต้องการใช้งานหน้าเว็บเพ็จอื่น-1

```
#!/usr/bin/perl
    open(OUTPUT, "> fsdata.c");
    chdir("fs");
    open(FILES, "find . -type f |");
```

รูปที่ 5-7 ตัวอย่างส่วนที่ต้องปรับแต่งเมื่อต้องการใช้งานหน้าเว็บเพ็จอื่น-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ผู้พัฒนาไอพีรุ่น 6 บนระบบฝังตัว

ในลักษณะการใช้งานนี้ผู้ที่สร้างไอพีรุ่น 6 บนระบบฝังตัวสามารถที่จะทำสิ่งเหล่านี้ได้

1. พัฒนาแอพพลิเคชันเซอร์วิส โดยอาจใช้ฟังก์ชันที่มีอยู่ ไลบารี่มาตรฐานหรือไลบารี่ที่สร้างขึ้นเองก็ได้ เช่น การพัฒนาไลบารี่ของ SOAP บนไอพีรุ่น6 โมไบล์ไอพี (Mobile IP) เป็นต้น
2. พัฒนาโปรโตคอลแสต็กให้สามารถให้บริการอื่นเพิ่มเติมได้ เช่น ดีเอชซีพีรุ่น 6 (DHCPv6) ชนิดของแอดเดรสอื่นๆ เป็นต้น
3. พัฒนาอุปกรณ์ฮาร์ดแวร์ส่วนขยายและซอฟแวร์ควบคุมที่เข้ามาทำงานร่วมกันกับอุปกรณ์นี้ เพื่อให้สามารถทำงานเป็นอุปกรณ์ชนิดอื่นได้ เช่น เชื่อมต่อกับแอ็กเซสพอยท์เพื่อทำงานกับระบบเครือข่ายไร้สาย สร้างเกตเวย์บนระบบเครือข่ายที่สนับสนุนไอพีรุ่น 6 เป็นต้น
4. พัฒนาให้เป็นอุปกรณ์ปลายทางที่มีความสามารถอื่นๆ เพิ่มเติม
5. นำเอาเฉพาะซอฟแวร์ของระบบไปปรับปรุงและประยุกต์ใช้กับอุปกรณ์ฝังตัวอื่นๆ เช่น ARM-7 ARM-9 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดสอบและวิเคราะห์ผล

6.1 การทดสอบเบื้องต้น

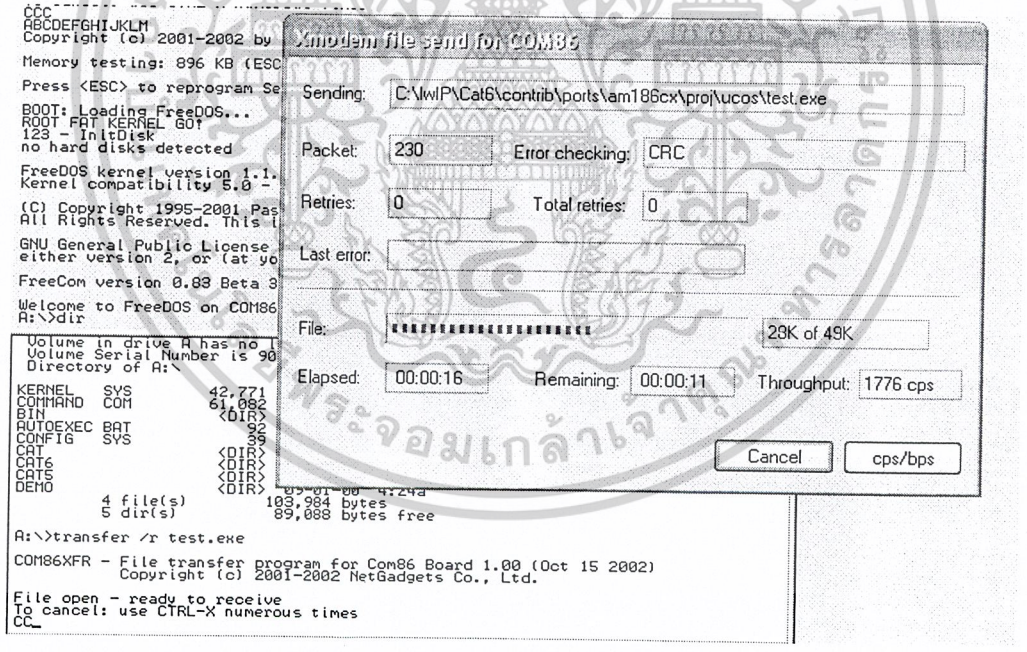
เนื่องจากโครงงานนี้มีการติดต่อใช้งานกับอุปกรณ์ฮาร์ดแวร์ จึงมีความจำเป็นที่ต้องทดสอบทดลองใช้งานอุปกรณ์ก่อนว่าสามารถทำงานได้อย่างถูกต้องตามที่ทราบมาจริงๆ ซึ่งแบ่งการทดสอบออกเป็นดังนี้

1. ทดลองส่งไฟล์เข้าไปในบอร์ดคอม 86 และรันโปรแกรม
2. ทดลองรับ-ส่งข้อมูลกับบอร์ดคอม 86 ผ่านทางพอร์ตอนุกรม
3. ทดลองให้บอร์ดคอม 86 ทำงานบนระบบเครือข่าย

นอกจากนี้แล้วโครงงานนี้ยังต้องมีการทดสอบทดลองการทำงานของไอพีรุ่น 6 ด้วย จึงต้องศึกษาและทดลองใช้งานกับระบบที่ให้บริการไอพีรุ่น 6 ได้จริง

4. ทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการวินโดว์ เอ็กซ์พี (Window XP)
5. ทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการลีนุกซ์

6.1.1 ทดลองส่งไฟล์เข้าไปในบอร์ดคอม 86 และรันโปรแกรม



รูปที่ 6-1 การทดลองส่งไฟล์เข้าไปในอุปกรณ์คอม 86

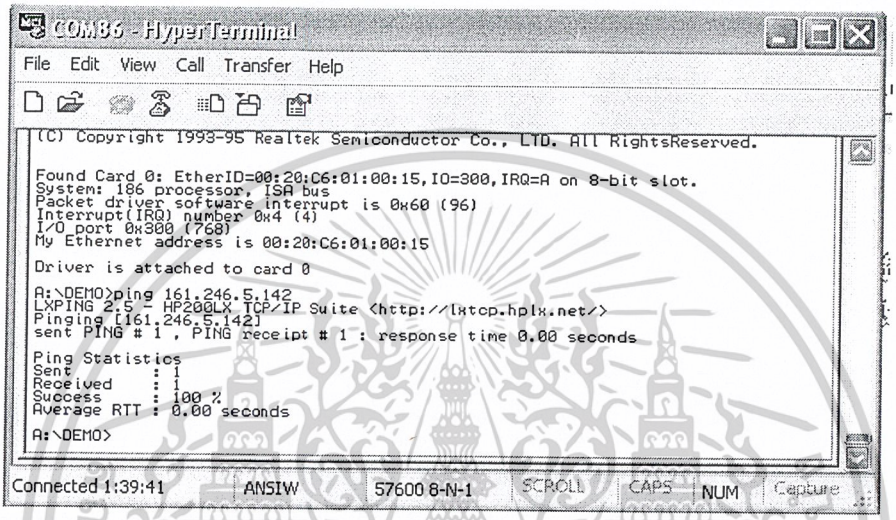
การทดลองส่งไฟล์เข้าไปในอุปกรณ์คอม 86 แล้วรัน เป็นการทดลองเพื่อทำความเข้าใจการทำงานกับชุดพัฒนา โดยโปรแกรมที่ใช้ส่งไปเพื่อทดสอบการทำงานนั้นเป็นโปรแกรมง่ายๆ นอกจากนี้ยังเพื่อศึกษาการทำงานของระบบปฏิบัติการ FreeDOS ว่ามีการทำงานอย่างไรด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 ทดลองรับ-ส่งข้อมูลกับบอร์ดคอม 86 ผ่านทางพอร์ตอนุกรม

เป็นการทดลองรับ-ส่งข้อมูลผ่านทางพอร์ตอนุกรม โดยการเชื่อมต่อกับคอมพิวเตอร์อีกเครื่อง แล้วรันโปรแกรม จากนั้นทำการส่งข้อความไปยังคอมพิวเตอร์อีกเครื่อง โดยจะรับข้อความผ่านทางโปรแกรมไฮเปอร์เทอร์มินัล

6.1.3 ทดลองให้บอร์ดคอม 86 ทำงานบนระบบเครือข่าย



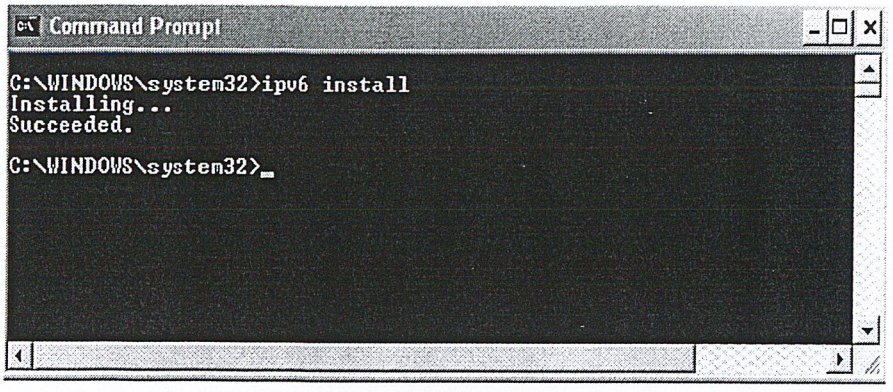
รูปที่ 6-2 การทดลองให้ชุดพัฒนาคอม 86 ทำงานบนเครือข่าย

การทดลองนี้ได้เรียกแพ็กเก็ตไดร์เวอร์ขึ้นมาทำงาน และทดสอบด้วยการ ping ไปยังโฮสต์อื่นบนระบบเครือข่าย ว่าสามารถได้รับการตอบกลับมาหรือไม่ ถ้าได้แสดงว่าสามารถทำงานบนอีเทอร์เน็ทได้ ถ้าไม่ได้แสดงว่าความผิดพลาด ณ จุดใดจุดหนึ่ง เช่น การทำงานของแพ็กเก็ตไดร์เวอร์ ใช้งานไม่ถูกต้องตามขั้นตอน เป็นต้น

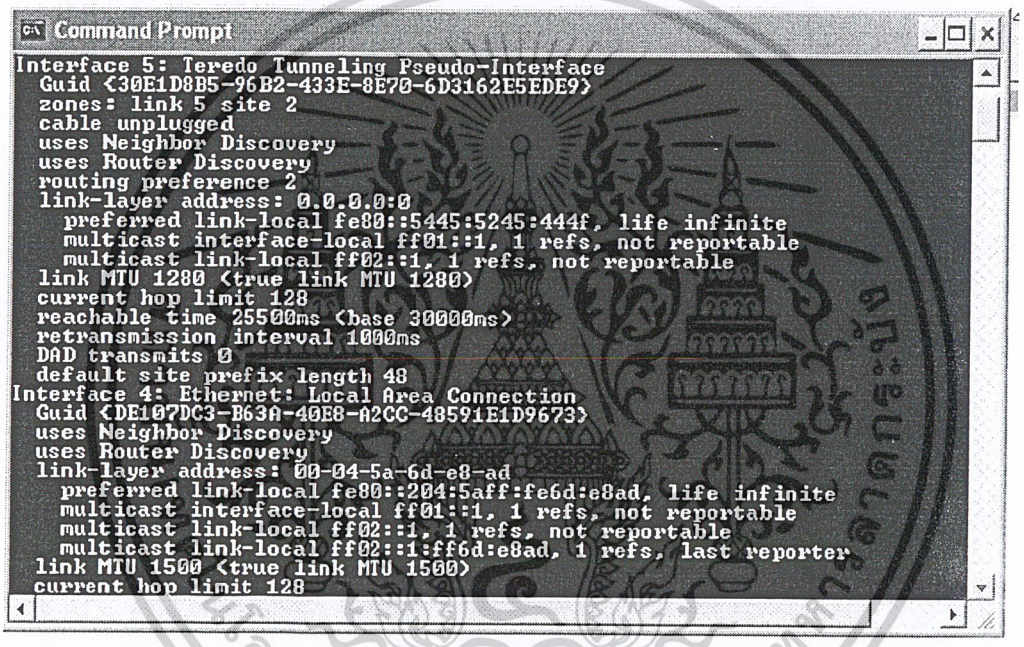
6.1.4 ทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการวินโดว์ เอ็กซ์พี (Window XP)

การจะทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการวินโดว์ เอ็กซ์พีนั้นก่อนอื่นต้องมีการติดตั้งซอฟแวร์ของไอพีรุ่น 6 เสียก่อน ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

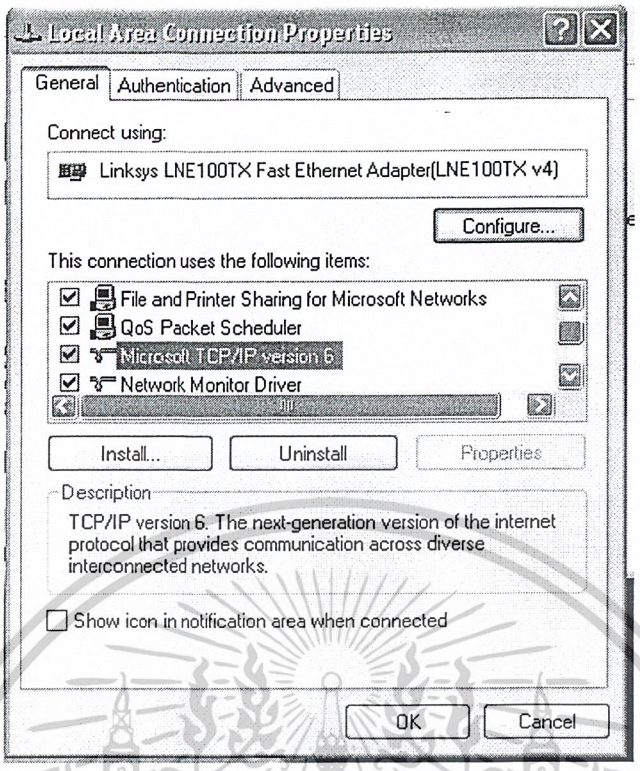


รูปที่ 6-3 การติดตั้งไอพีรุ่น 6 ลงบนระบบปฏิบัติการวินโดว์



รูปที่ 6-4 ผลลัพธ์ที่ได้บางส่วนหลังจากการติดตั้งไอพีรุ่น 6 ลงบนระบบปฏิบัติการวินโดว์ -1

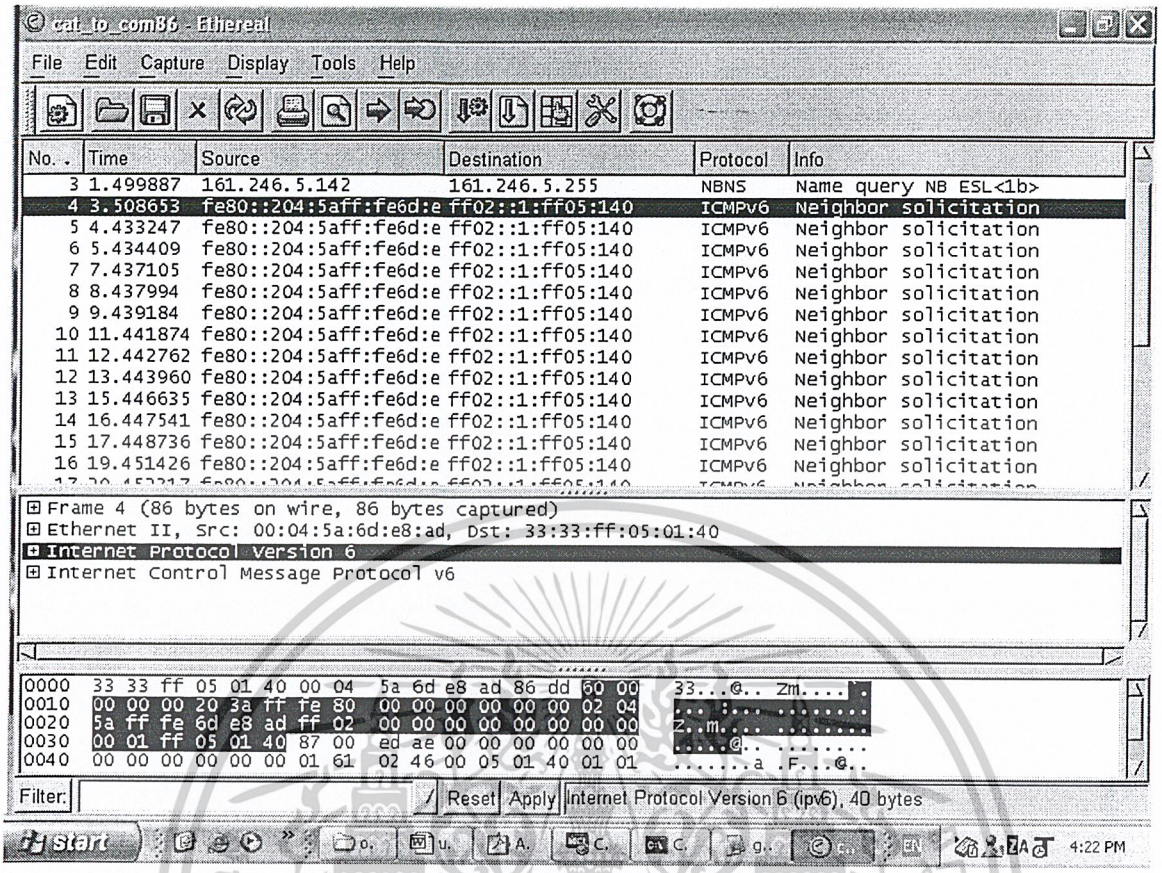
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-5 ผลลัพธ์ที่ได้บางส่วนหลังจากการติดตั้งไอพีรุ่น 6 ลงบนระบบปฏิบัติการวินโดว์ -2

หลังจากทำการติดตั้งแล้วจึงทำการทดสอบ โดยให้เครื่องที่ได้รับการติดตั้งแล้ว 2 เครื่องทำงานอยู่บนระบบเครือข่าย แล้วใช้โปรแกรมประยุกต์ ping6 ในการทดสอบการทำงานโดยการส่งแพ็กแก็ตไปยังเครื่องเป้าหมาย และใช้โปรแกรมประเภทโปรโตคอลอนาไลซ์เซอร์ (protocol analyzer) ในการตรวจสอบแพ็กแก็ตอีกทีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



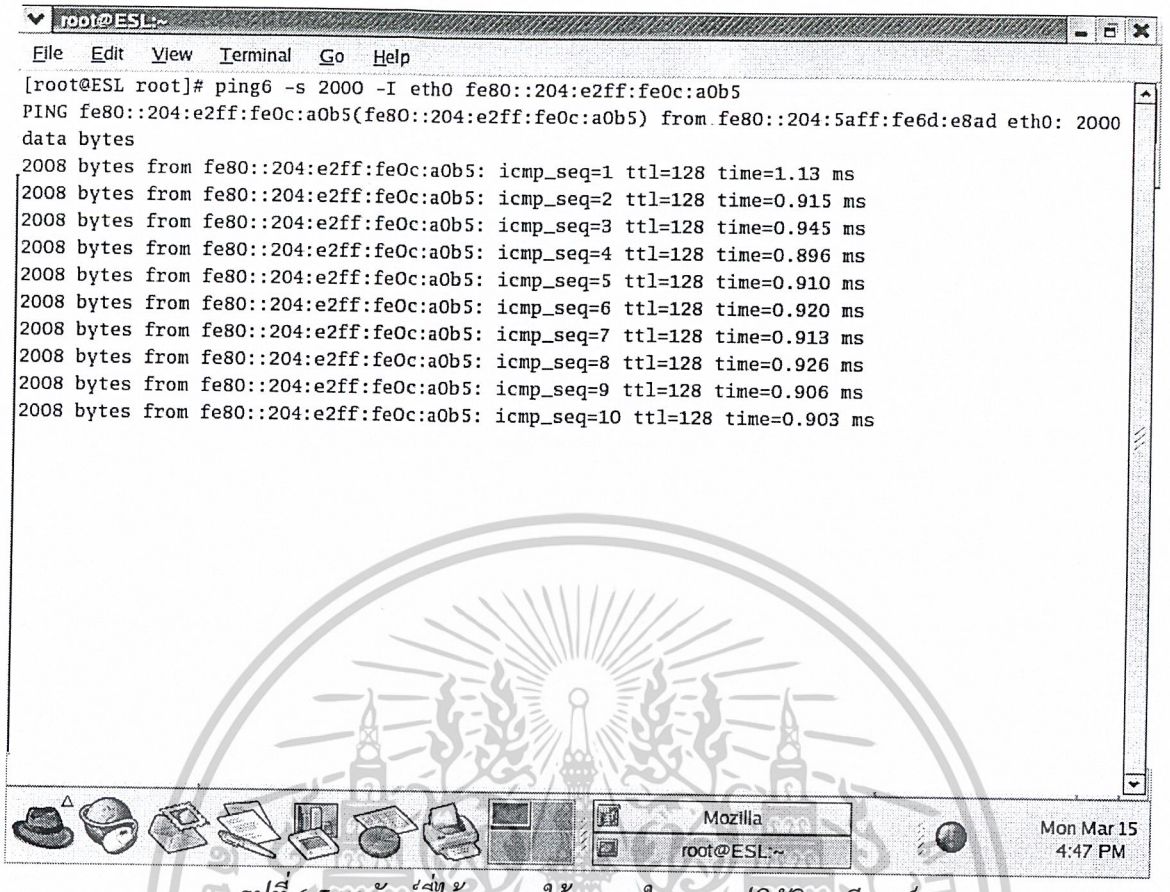
รูปที่ 6-6 ผลลัพธ์ที่ได้บางส่วนที่ได้จากโปรแกรมอีเทอร์เรียลและการ ping6

6.1.5 ทดสอบการให้บริการไอพีรุ่น 6 บนระบบปฏิบัติการลีนุกซ์

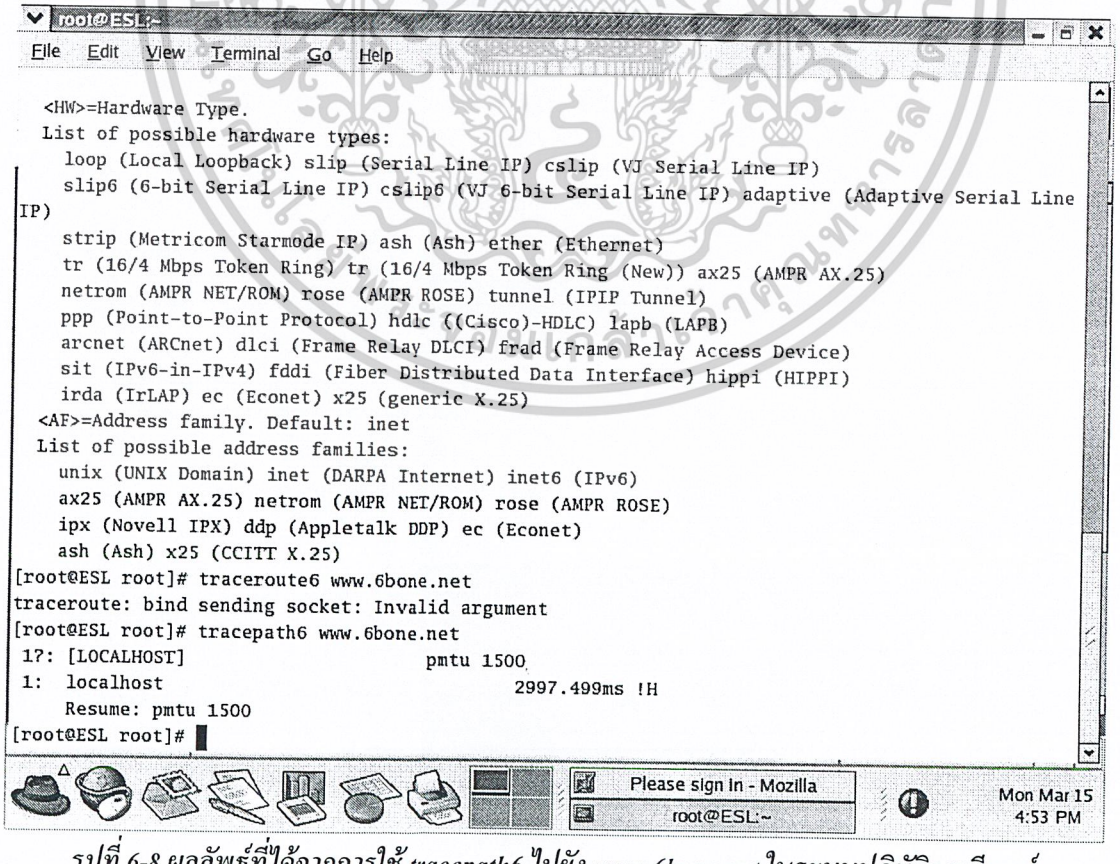
ในที่นี้ผู้พัฒนาได้เลือกใช้ลีนุกซ์เรดแฮ็ท 9.0 เช่นเดียวกันกับระบบปฏิบัติการวินโดว์ที่ต้องมีการติดตั้งไอพีรุ่น 6 ลงไปก่อนเช่นกันถึงจะสามารถใช้งานได้ แต่ต่างกันตรงที่ระบบปฏิบัติการลีนุกซ์นั้นต้องมีการรีคอมไพล์เคอร์เนลใหม่ด้วย เพราะจากการตรวจสอบแล้วพบว่าในเคอร์เนล 2.4.20-8 ไม่รองรับไอพีรุ่น 6 จึงต้องเปลี่ยนไปใช้เคอร์เนล 2.4.23-0 แทน

จากนั้นจึงทำการทดสอบการทำงานโดยการใช้เครื่องที่ติดตั้งไอพีรุ่น 6 แล้วสองเครื่องซึ่งทำงานอยู่บนระบบเครือข่าย โดยทดสอบด้วยการใช้โปรแกรมประยุกต์ ping6 ไปยังเครื่องปลายทางและ tracepath6 ไปยัง www.6bone.net ซึ่งเป็นทำงานอยู่บนระบบเครือข่ายไอพีรุ่น6 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-7 ผลลัพธ์ที่ได้จากการใช้ ping6 ในระบบปฏิบัติการลีนุกซ์



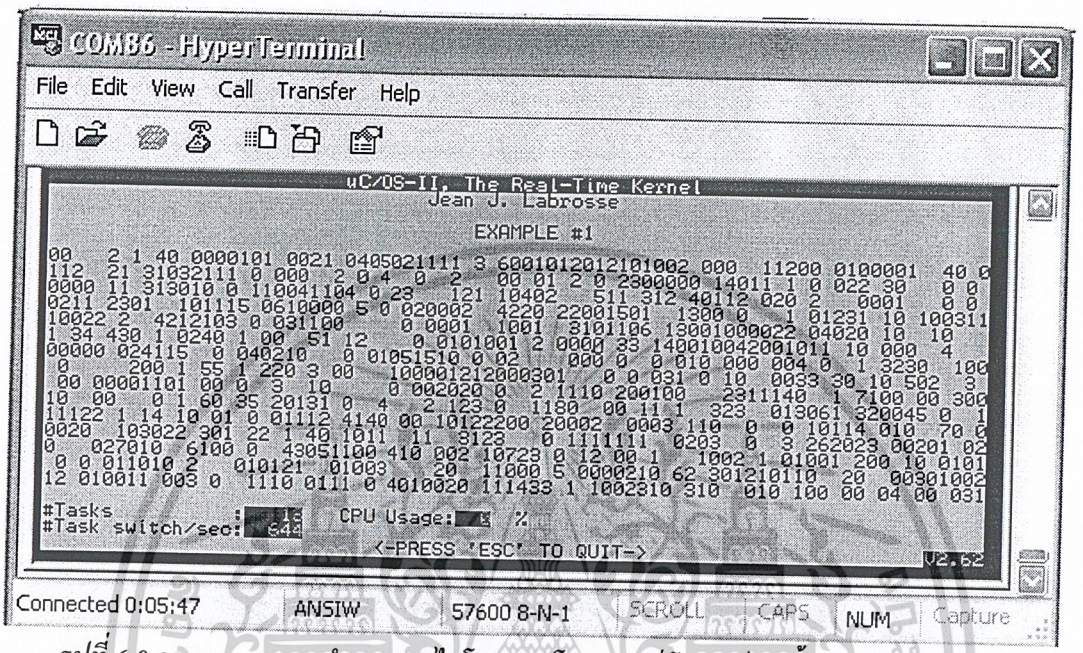
รูปที่ 6-8 ผลลัพธ์ที่ได้จากการใช้ tracepath6 ไปยัง www.6bone.net ในระบบปฏิบัติการลีนุกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การทดสอบระบบ

6.2.1 การทดสอบการทำงานแบบมัลติทาสก์กิ้ง

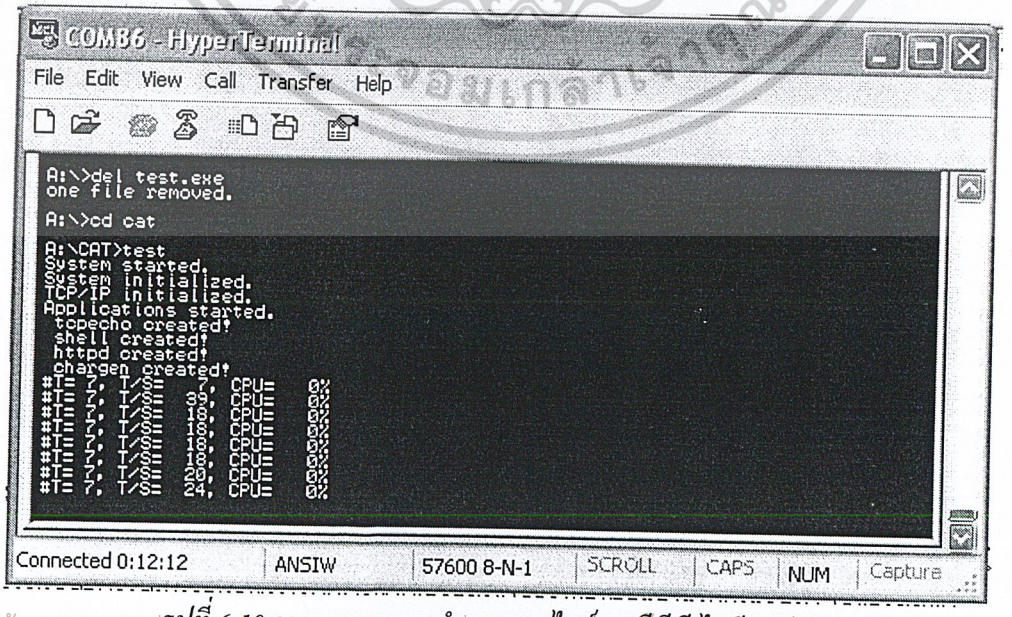
ขั้นตอนนี้คือการทดสอบการทำงานของไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2 ที่พอร์ตลงบนบอร์ดแล้วว่าสามารถทำงานได้อย่างถูกต้องหรือไม่ โดยใช้โปรแกรมประยุกต์ที่มีการทำงานหลายทาสก์ด้วยกัน เพื่อทดสอบการทำงาน ดังนี้



รูปที่ 6-9 การทดสอบการทำงานของไมโครคอนโทรลเลอร์ โอเปอร์เรทติ้ง ซิสเต็มเวอร์ชัน 2

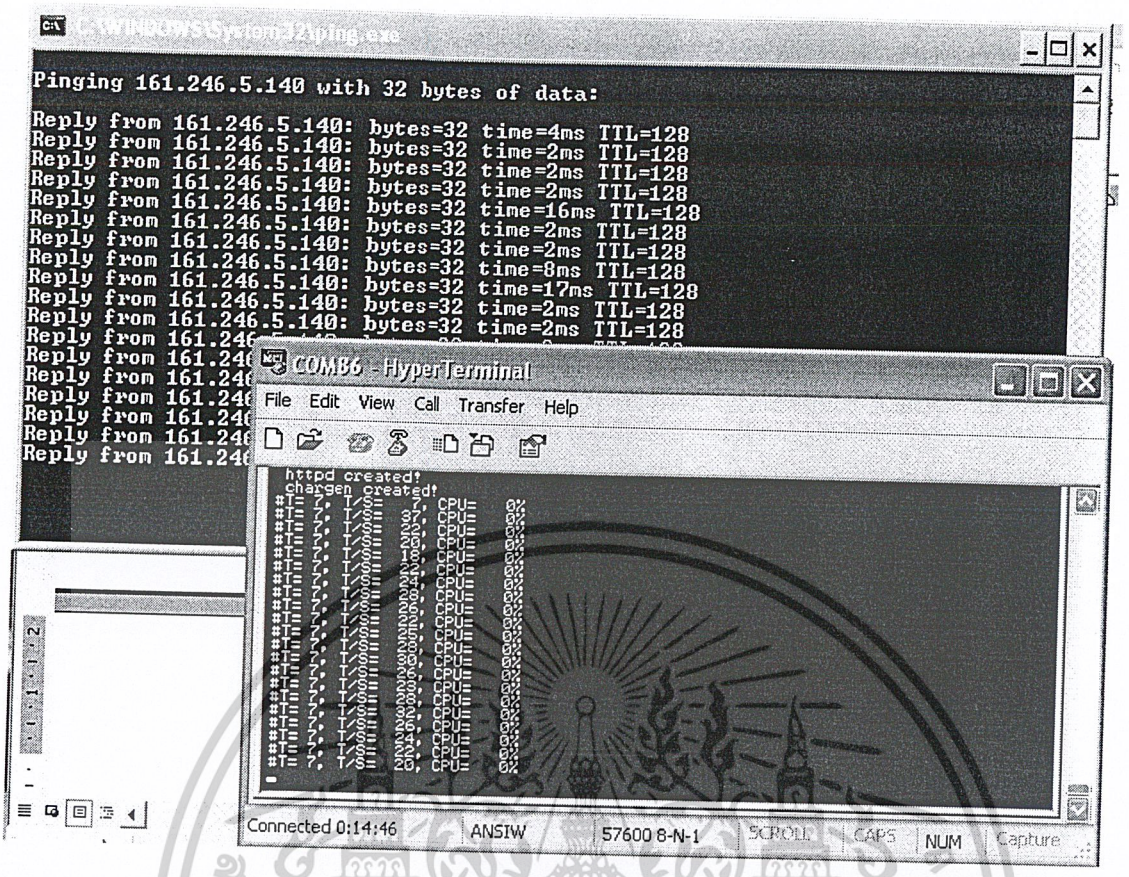
6.2.2 การทดสอบการทำงานของไลท์เวททีซีพี/ไอพีแสต็ก

เป็นการทดสอบการทำงานของโปรโตคอลแสต็กหลังจากทำการพอร์ตลงบนบอร์ดคอม 86 แล้ว โดยเป็นการทดสอบว่าสามารถที่จะทำงานบนระบบเครือข่ายได้จริงและถูกต้องตามมาตรฐาน โดยการเขียนโปรแกรมประยุกต์เพื่อทดสอบการทำงาน ดังนี้

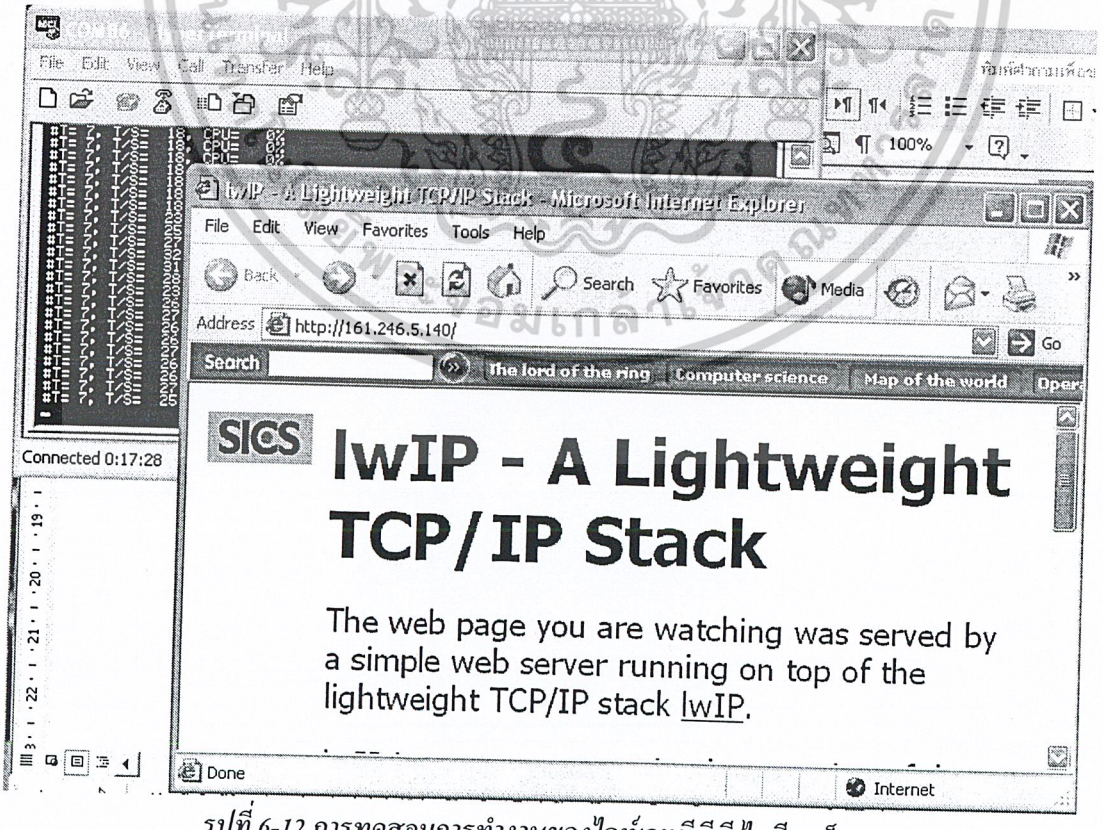


รูปที่ 6-10 การทดสอบการทำงานของไลท์เวททีซีพี/ไอพีแสต็ก-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-11 การทดสอบการทำงานของไลท์เวทที่ซีพี/ไอพีแสต็ก-2

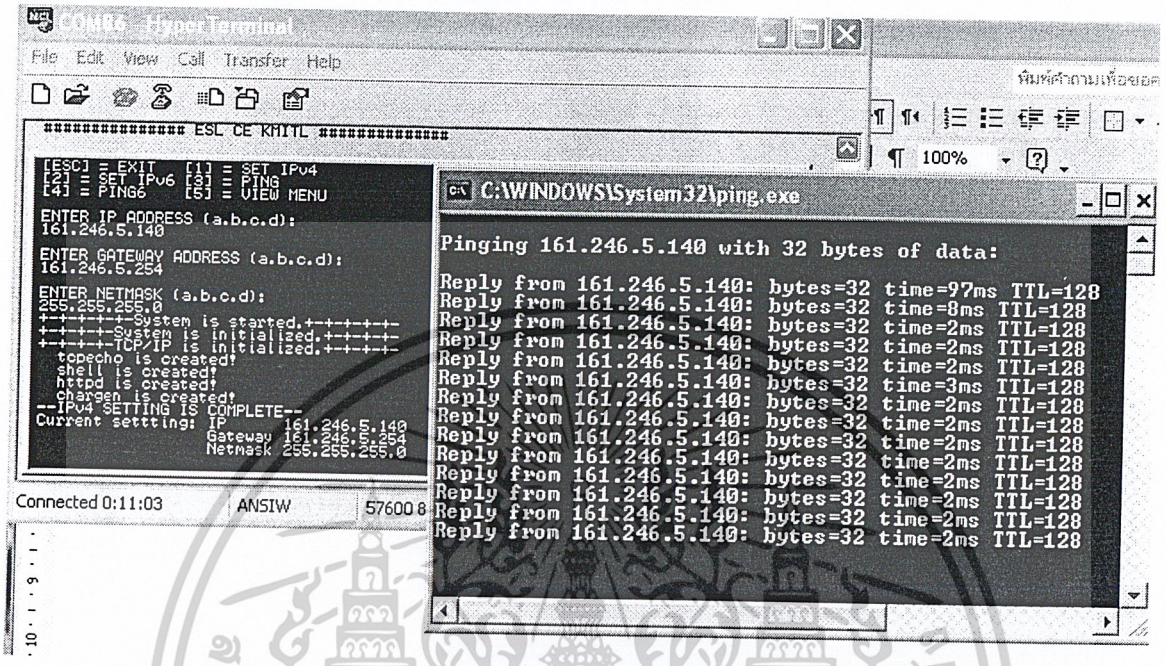


รูปที่ 6-12 การทดสอบการทำงานของไลท์เวทที่ซีพี/ไอพีแสต็ก-3

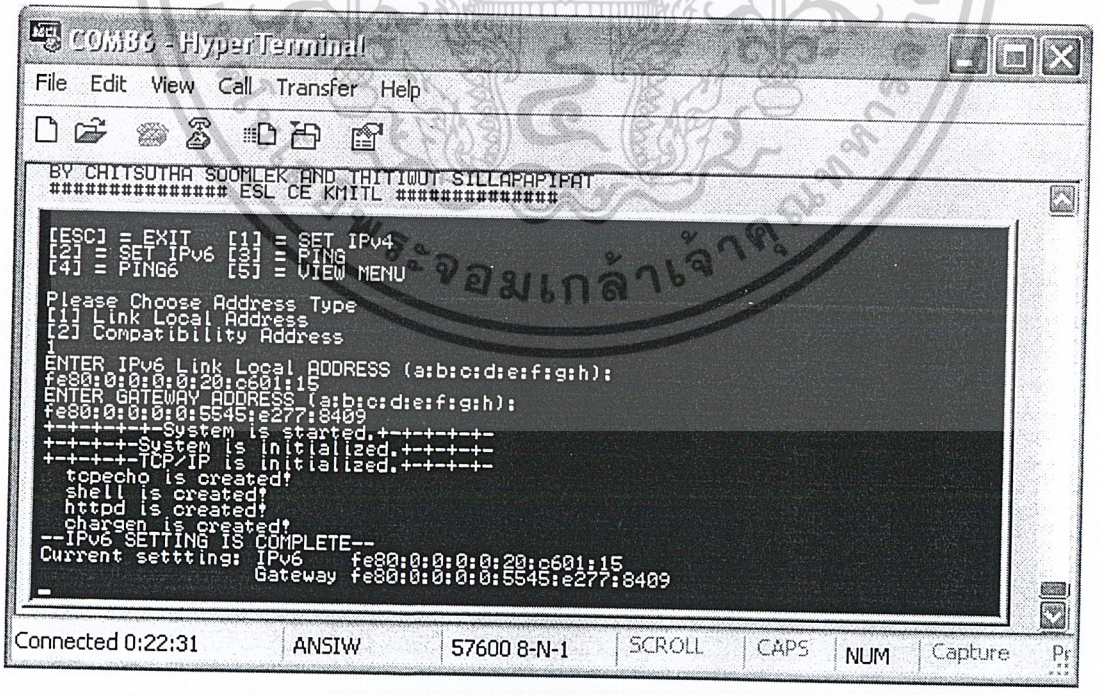
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 การทดสอบการการกำหนดค่าคุณสมบัติผ่านคอนโซลพอร์ตอนุกรม

การทดสอบนี้เป็นการทดสอบว่าระบบที่พัฒนาขึ้นสามารถทำงานได้ถูกต้องหรือไม่ สามารถกำหนดค่าคุณสมบัติให้กับระบบได้อย่างถูกต้องหรือไม่ เมื่อกำหนดค่าคุณสมบัติไปแล้วสามารถที่จะทำงานได้ถูกต้องหรือไม่ด้วย

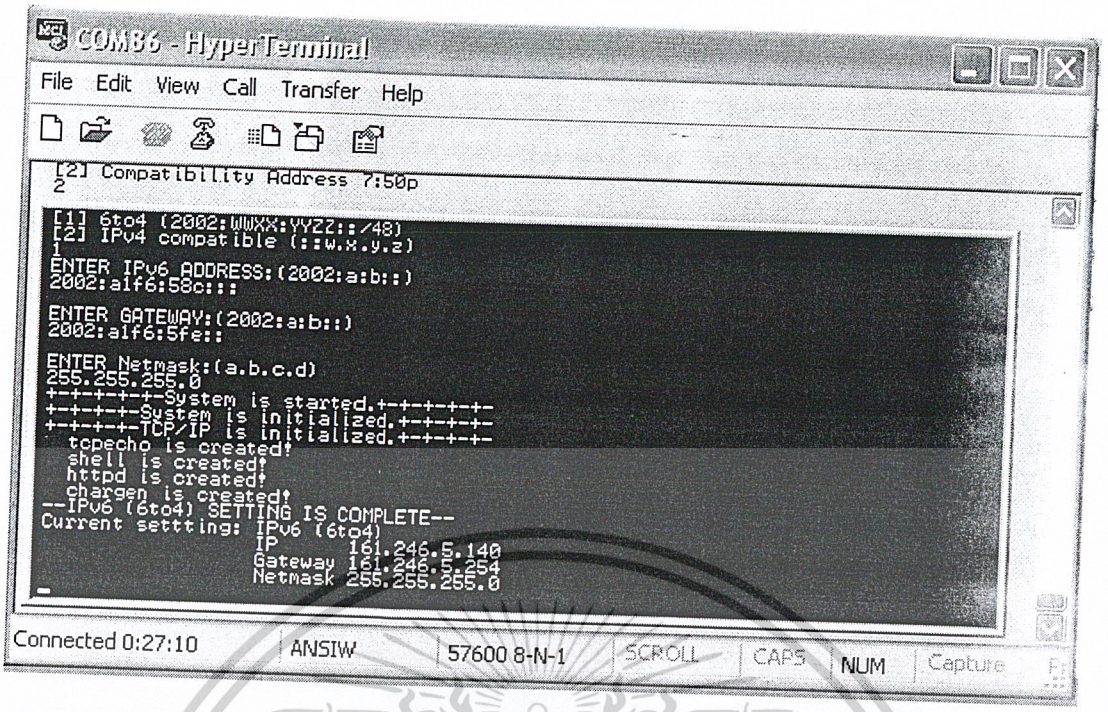


รูปที่ 6-13 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น 4 ให้กับระบบ

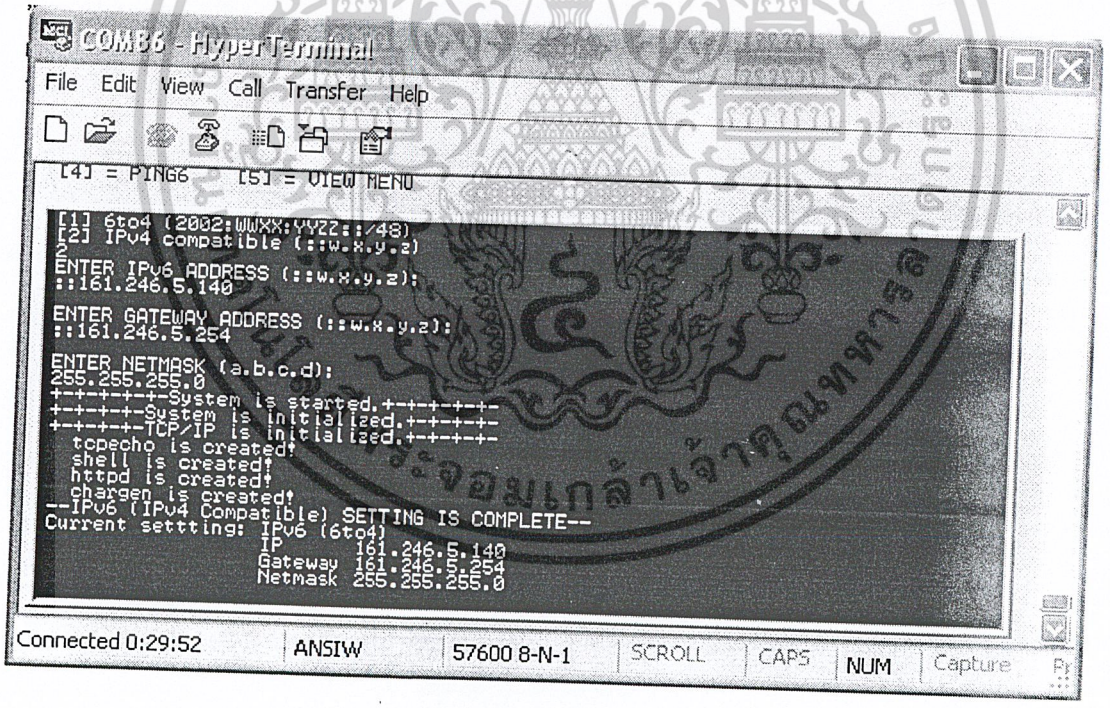


รูปที่ 6-14 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น6 แบบลิงค์โลคอลแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-15 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น6 แบบ Compatibility address (6to4)

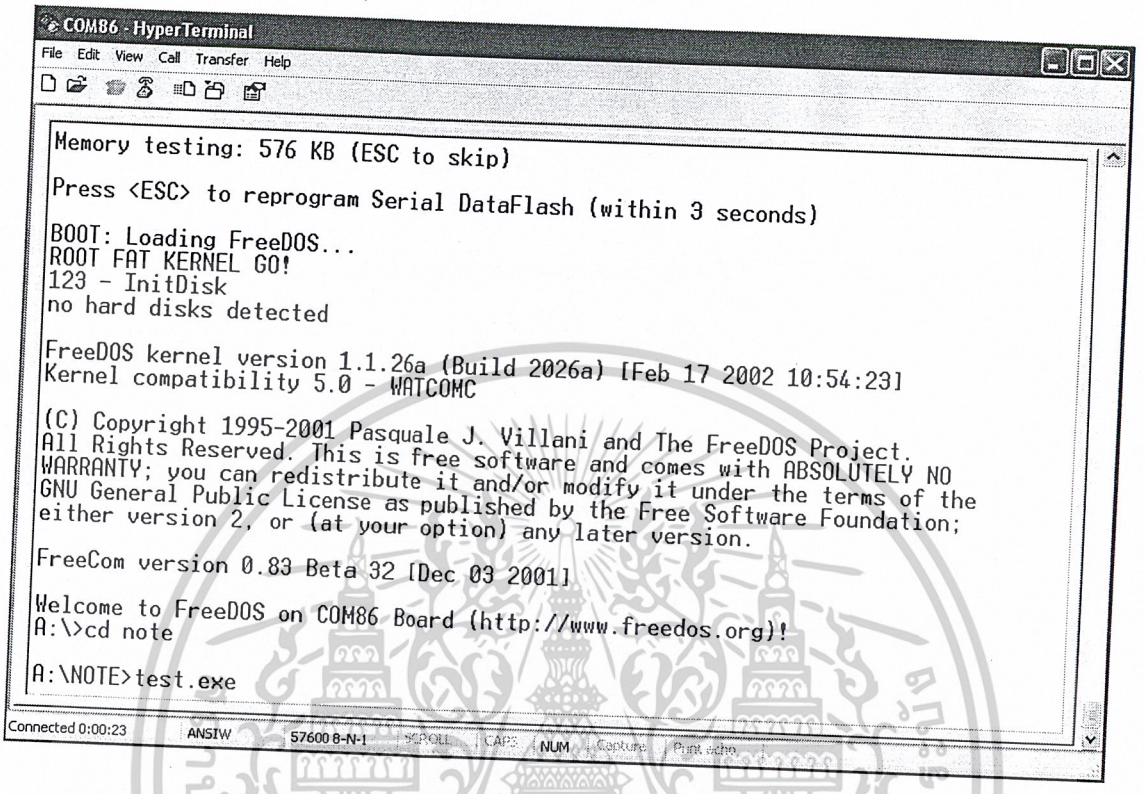


รูปที่ 6-16 การทดสอบการกำหนดค่าคุณสมบัติของไอพีรุ่น6 แบบ Compatibility address (IPv4 compatible)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

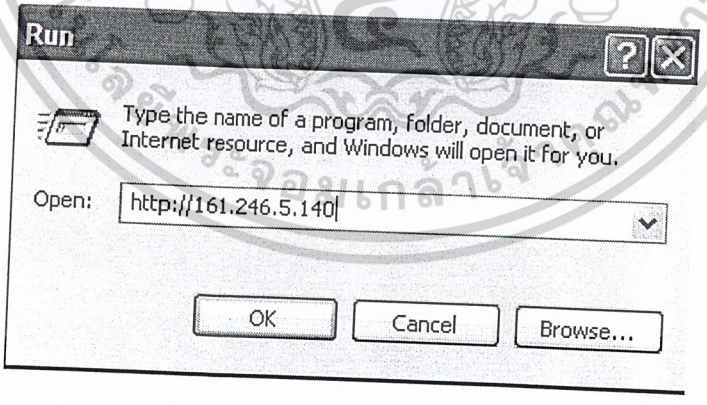
6.2.4 การทดสอบการการกำหนดค่าคุณสมบัติผ่านระบบเครือข่ายอีเทอร์เน็ท

เป็นการทดสอบการกำหนดค่าคุณสมบัติต่างๆโดยมีรูปแบบการกำหนดค่าต่างๆผ่านเวบบราวน์เซอร์ ก่อนจะทำการรันเว็บบราวน์เซอร์นั้นจะต้องมีการรันโปรแกรมบางตัวที่อยู่บนคอม86ซะก่อน



รูปที่6-17 มีการรัน test.exe ก่อนที่จะไปกำหนดค่าผ่านหน้าเว็บโดยในที่นี้กำหนดให้รันที่

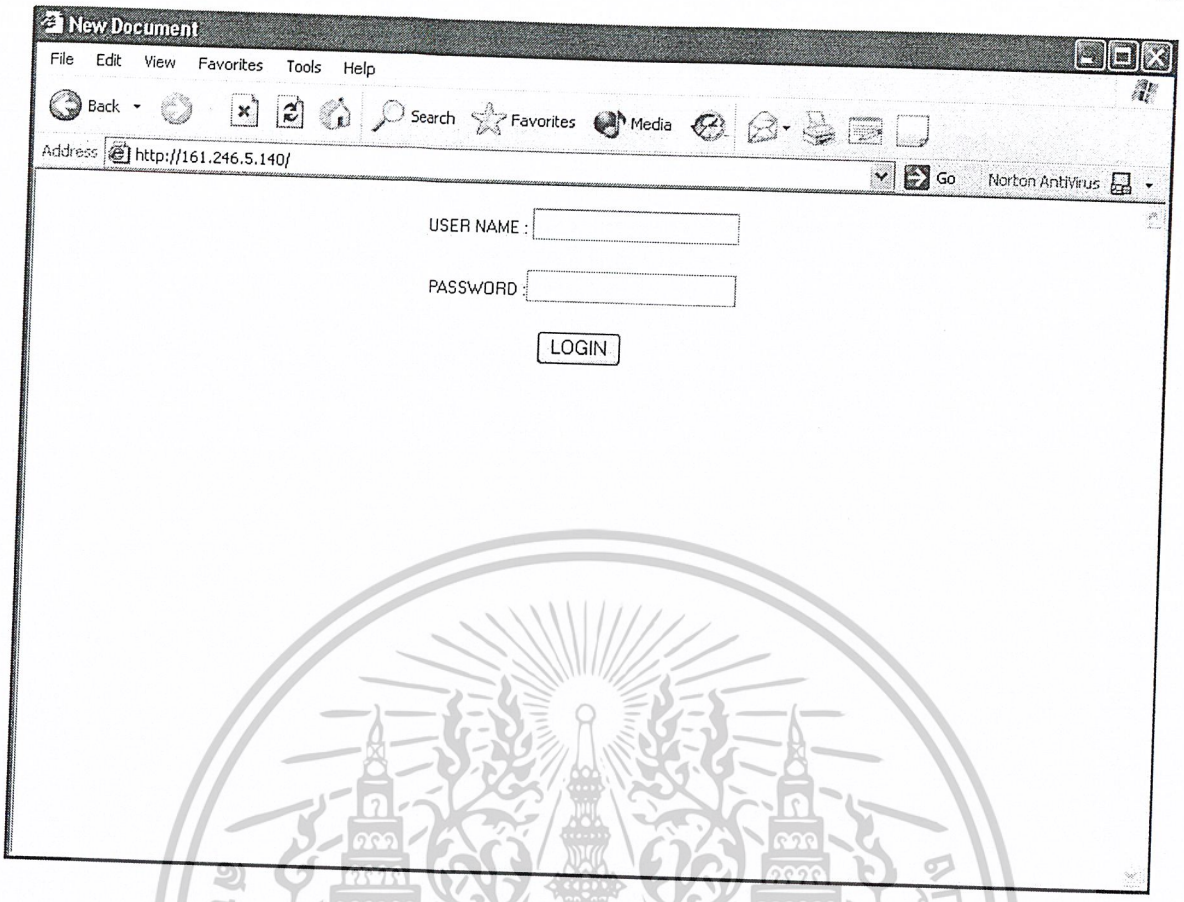
HTTP://161.246.5.140



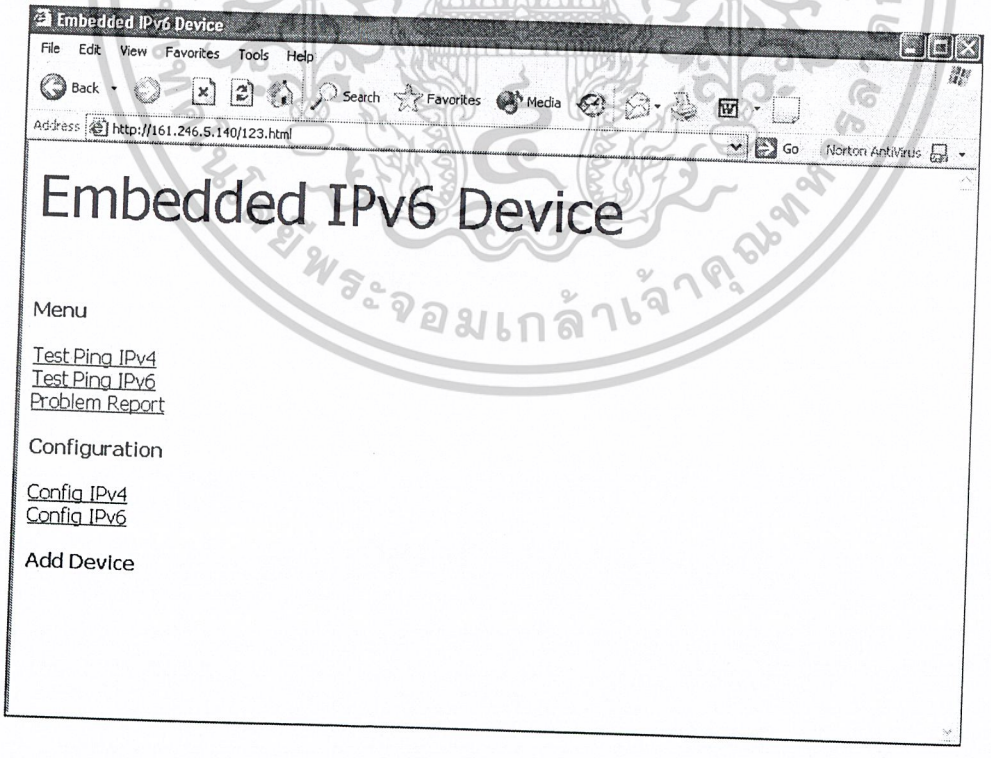
รูปที่ 6-18 การเข้าถึงระบบด้วยโปรโตคอลเอชทีทีพี

ในหน้าเว็บที่ไปกำหนดค่าไอพีนั้นจะต้องมีการกำหนดผู้ให้เข้าไปใช้เพื่อป้องกันการเข้าไปกำหนดค่าต่างๆจากคนภายนอกจึงจำเป็นต้องมีการป้อนค่าเพื่อล็อกอินเข้าสู่ระบบ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6-19 หน้าที่ใช้สำหรับล็อคอินเข้าสู่ระบบ



รูปที่6-20 หน้าเว็บที่แสดงออกมาหลังจากล็อกอินเข้ามาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเข้าสู่ระบบได้แล้วจะมีหน้าเว็บดังรูปที่ 6-20 นี้ซึ่งมีรายละเอียดดังนี้

ส่วน Menu

Test Ping IPv4: เป็นการทดสอบการ Ping ในแบบไอพีรุ่นสี่

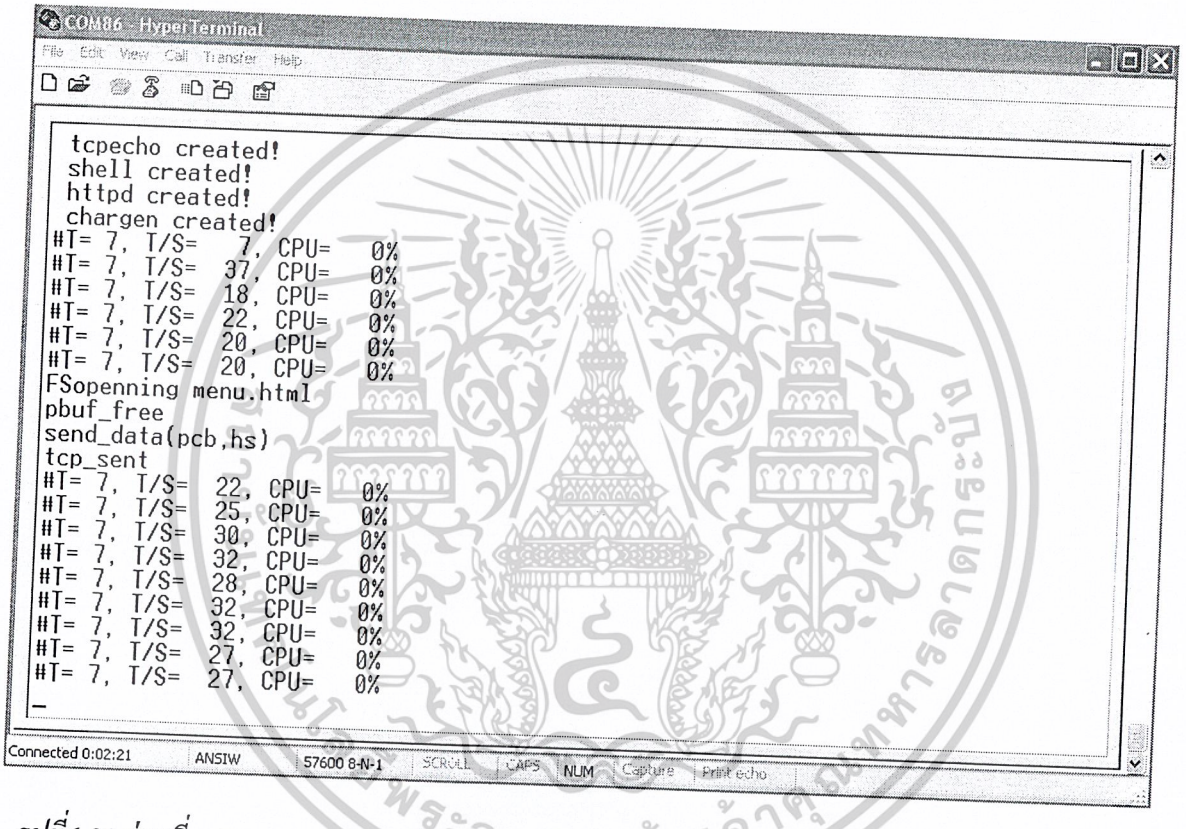
Test Ping IPv6 : เป็นการทดสอบการ Ping ในแบบไอพีรุ่นหก

Problem Report : เป็นการแสดงค่าข้อผิดพลาดที่เกิดขึ้น

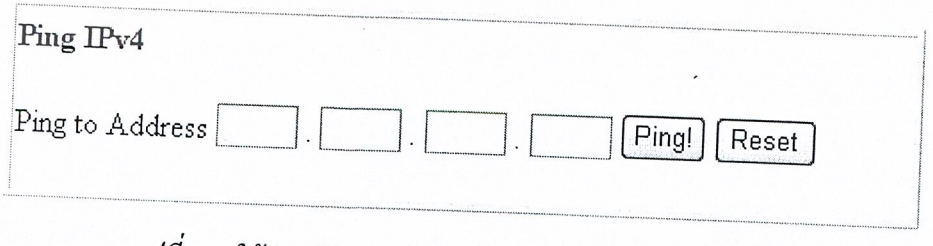
ส่วน Configuration

Config IPv4 : เป็นการกำหนดค่าต่างๆที่ใช้ในแบบไอพีรุ่นสี่

Config IPv6 : เป็นการกำหนดค่าต่างๆที่ใช้ในแบบไอพีรุ่นหก

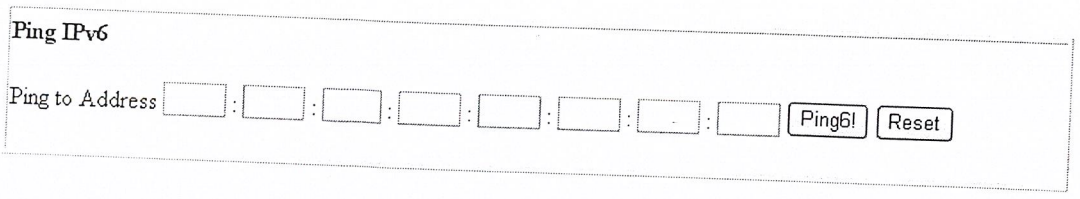


รูปที่6-21 ส่วนที่แสดงผลออกมาขณะล็อคอินเข้ามาในเวบแล้วถูกแสดงผ่านคอม86โดยจะมีการบอกว่ามีการเปิดหน้าเวบนี้ขึ้นมาแล้ว (FSopenning menu.html)

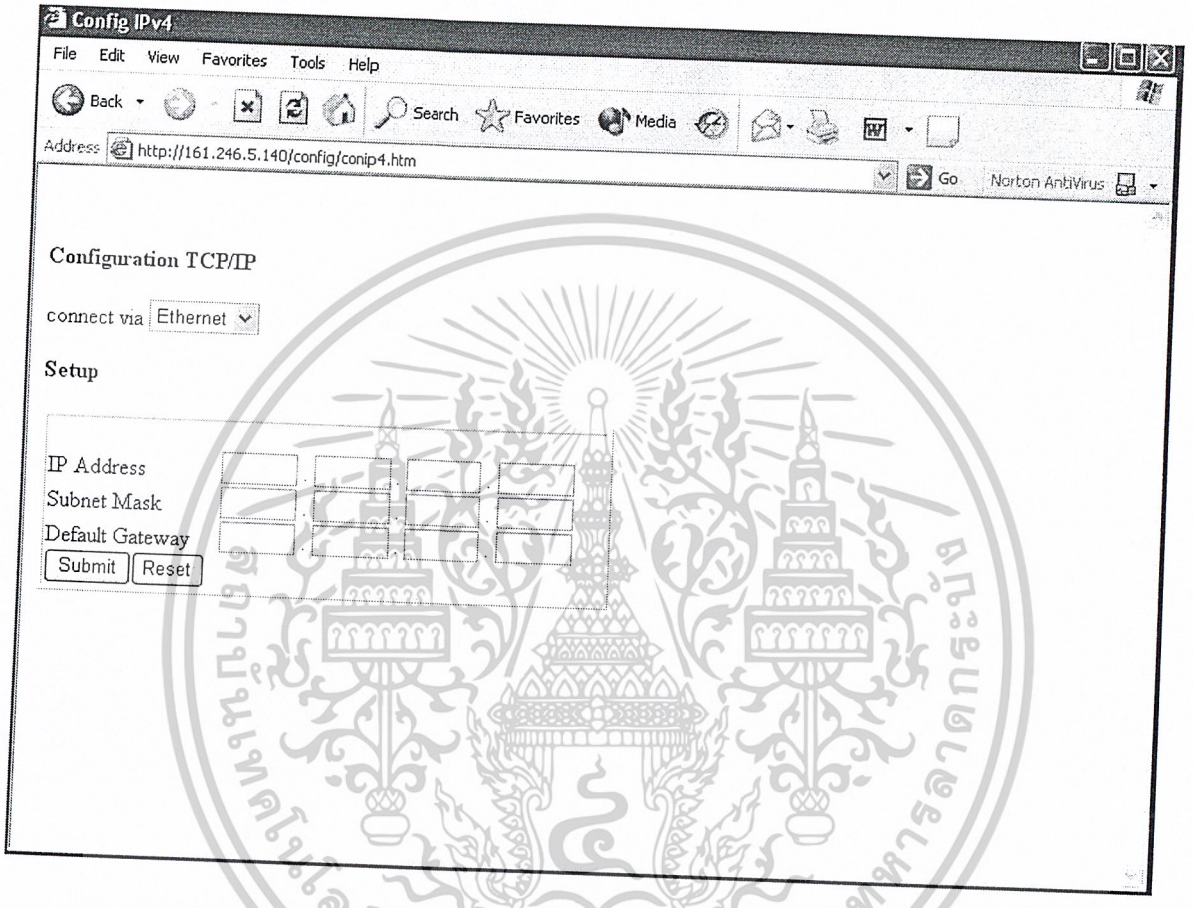


รูปที่6-22 ใช้สำหรับกรอกค่าในการทดสอบการ Ping ในไอพีรุ่นสี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

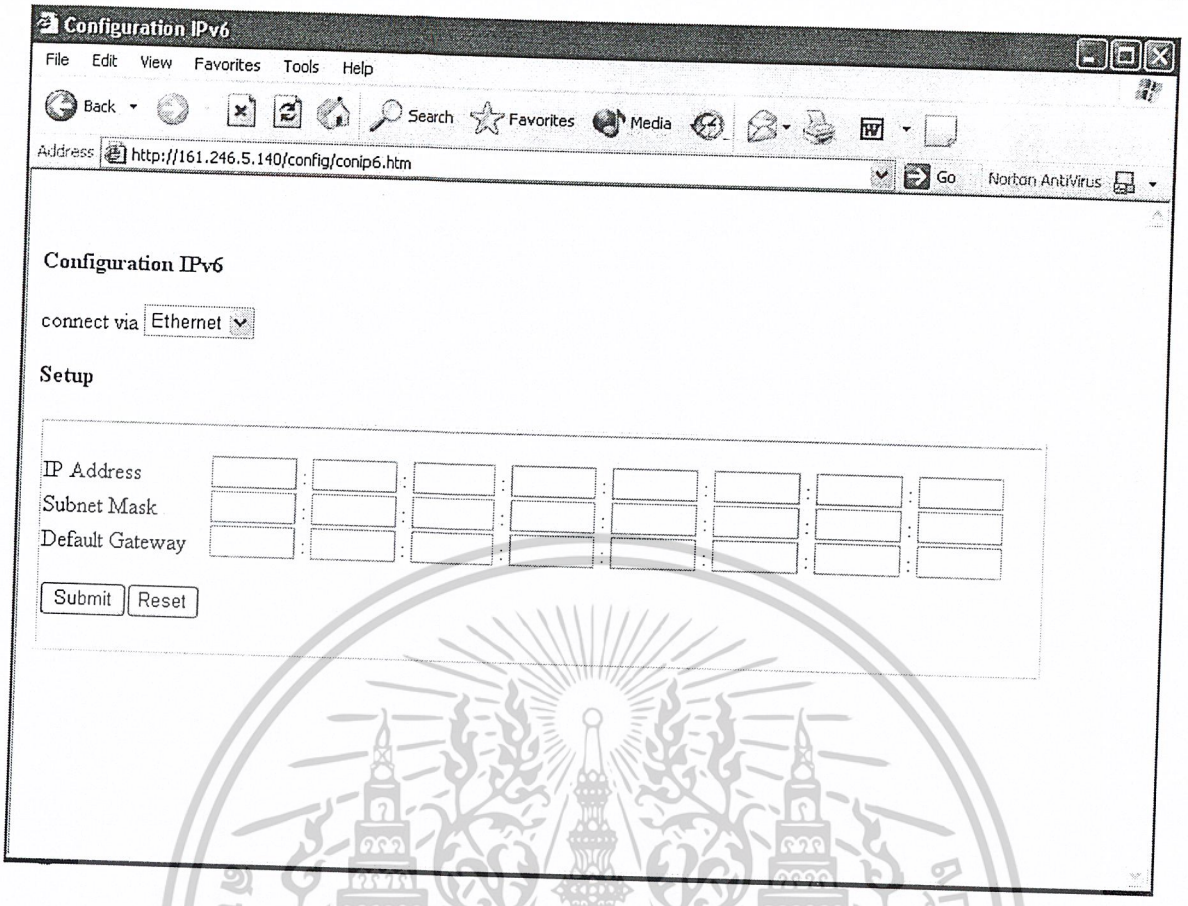


รูปที่6-23 ใช้สำหรับกรอกค่าในการทดสอบการ Ping ในไอพีรุ่นหก



รูปที่6-24 ใช้สำหรับกรอกค่าการกำหนดค่าต่างๆให้กับระบบในแบบไอพีรุ่นสี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

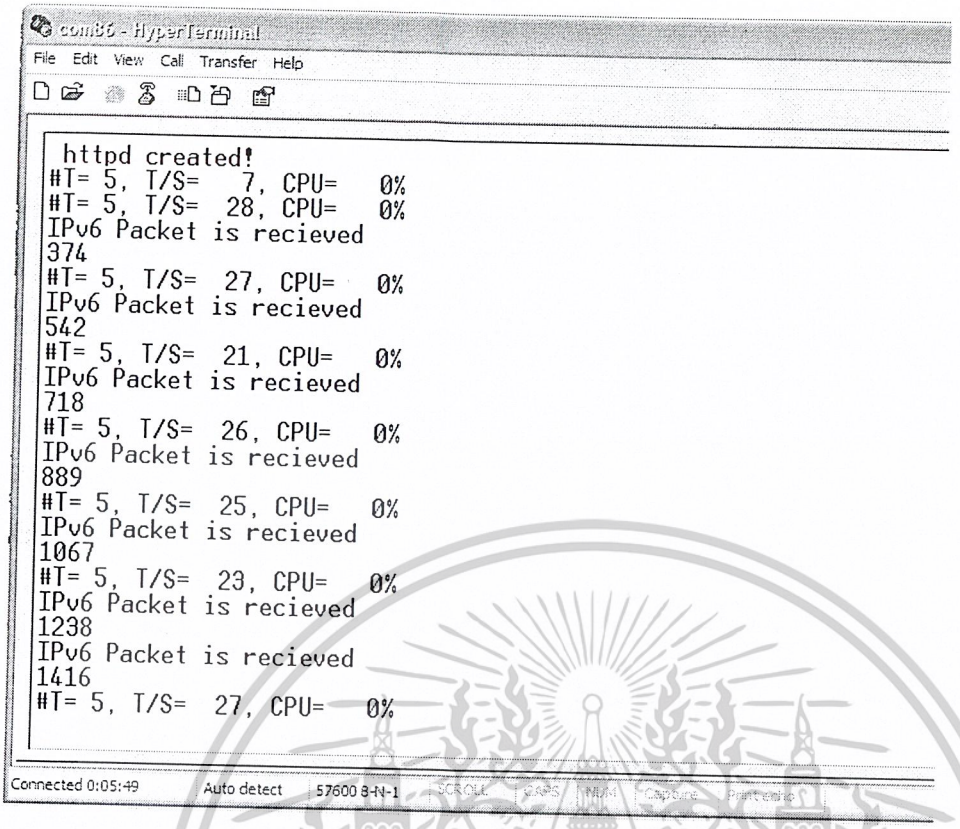


รูปที่6-25 ใช้สำหรับกรอกค่าการกำหนดค่าต่างๆให้กับระบบในแบบไอพีรุ่นหก

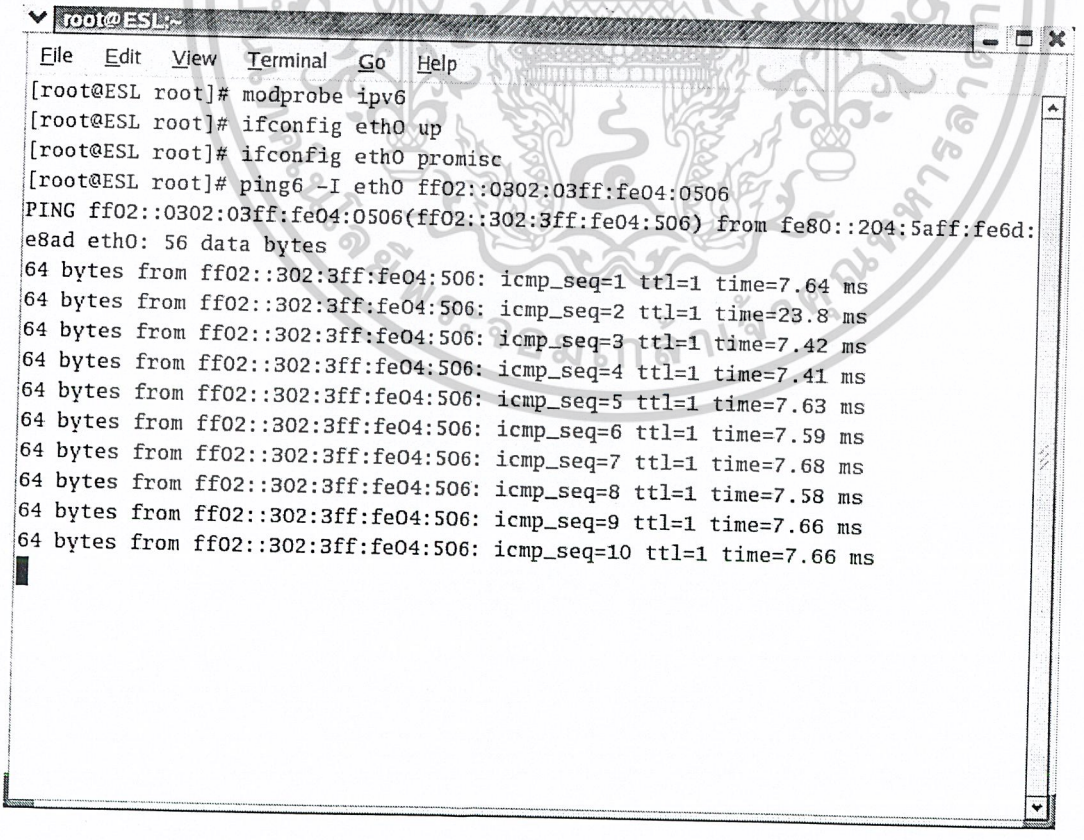
6.2.5 การทดสอบ ping6

เป็นการทดสอบโปรโตคอลแสต็กว่าสามารถทำงานเกี่ยวกับไอพีรุ่น 6 และไอซีเอ็มพีรุ่น 6 ได้ โดยจะส่งไอซีเอ็มพีแพ็กเก็ตจากระบบปฏิบัติการลีนุกซ์ไปหาคอม 86 และรอการตอบกลับ โดยในที่นี้ใช้โปรแกรมประยุกต์ ping6 ในการทดสอบและได้ผลลัพธ์ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่6-26 แสดงสถานะว่าได้รับ IPv6 packet เข้ามา



รูปที่6-27 แสดงการส่งและรับ ICMPv6 packet จาก Linux กับ COM86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทสรุปและวิจารณ์

7.1 วิเคราะห์ผลจากการออกแบบและสร้างระบบ

เนื่องจากปัจจุบันเทคโนโลยีของระบบเครือข่ายมีการเปลี่ยนแปลงอยู่ตลอดเวลา และมีความต้องการที่สูงขึ้นเรื่อง มีการผลิตอุปกรณ์ที่สามารถเข้าไปทำงานในระบบเครือข่ายได้ออกมามากมาย เช่น โทรทัศน์ อุปกรณ์พกพาต่างๆ อุปกรณ์ในภาคอุตสาหกรรม เป็นต้น ซึ่งอุปกรณ์ที่พัฒนาขึ้นมาใช้งานกับระบบเครือข่ายนั้นส่วนใหญ่อยู่บนพื้นฐานของระบบฝังตัวแทบทั้งนั้น ด้วยการพัฒนาที่รวดเร็วดังกล่าวทำให้การที่จะนำไอพีรุ่น 6 มาใช้นั้นเป็นสิ่งที่น่าจะพึงกระทำในเวลาอันใกล้นี้

โครงงานชิ้นนี้ได้มีวัตถุประสงค์เพื่อลดขั้นตอนในการศึกษาพัฒนาและค่าใช้จ่ายในการลงทุนเพื่อสนับสนุนให้เกิดการศึกษาและพัฒนาเกี่ยวกับเทคโนโลยีไอพีรุ่น 6 ซึ่งผู้ดำเนินโครงงานได้รวมเอาเทคโนโลยีของเรียลไทม์โอเอส ซึ่งในที่นี้ก็คือ ไมโครคอนโทรลเลอร์โอเปอร์เรทติ้งซิสเต็มเวอร์ชัน 2 และโปรโตคอลแสต็ก ซึ่งในที่นี้ก็คือ ไลท์เวททีซีพี/ไอพีแสต็ก ไว้ในชุดพัฒนาแล้ว ผู้ที่ต้องการนำอุปกรณ์นี้ไปพัฒนาต่อก็ไม่จำเป็นที่จะต้องพอร์ตระบบปฏิบัติการและโปรโตคอลแสต็กลงไปใหม่ สามารถนำไปใช้งานได้เลย โดยต้องศึกษาเกี่ยวกับการเขียนโปรแกรมกับไมโครคอนโทรลเลอร์โอเปอร์เรทติ้งซิสเต็มเวอร์ชัน 2 และการทำงานของไลท์เวททีซีพี/ไอพีแสต็กเพิ่มเติมเท่านั้น นอกจากนี้ยังได้มีการสร้างเอ็นจินที่ควบคุมการทำงานบางอย่างของระบบ คือ การกำหนดค่าคุณสมบัติของไอพีรุ่น 4 และไอพีรุ่น 6 เฉพาะลิงค์โลคอลแอดเดรส, 6to4 address, IPv4 compatible address โดยผู้ใช้สามารถติดต่อสารกับระบบได้โดยผ่านทางคอนโซลพอร์ตอนุกรมและทางอีเทอร์เน็ท นอกจากนี้เรายังสามารถทดสอบการทำงานของไอพีรุ่น 6 โปรโตคอลแสต็กได้ด้วยการใช้คำสั่ง ping6 จากโฮสต์ที่สามารถให้บริการเกี่ยวกับไอพีรุ่น 6 เพื่อทดสอบสถานะได้

แต่เนื่องจากการพัฒนาโปรโตคอลไอพีรุ่น 6 นั้นมีรายละเอียดและองค์ประกอบค่อนข้างมาก ผู้ดำเนินโครงงานจึงไม่สามารถที่จะสร้างอุปกรณ์ให้ให้บริการเกี่ยวกับไอพีรุ่น 6 ได้ครบทุกอย่างตามที่มาตรฐานกำหนด และการทำงานบางอย่างก็ไม่สามารถสร้างและทดสอบได้เพราะต้องอาศัยการทำงานอื่นร่วมด้วย เช่น การกำหนดโกลบอลแอดเดรสนั้นต้องกระทำผ่านเราท์เตอร์แอดเวอร์ไทส์เมนท์ (router advertisement) หรือ ดีเอชซีพีรุ่น 6 (DHCPv6) เป็นต้น และการให้บริการบางอย่างนั้นอาจต้องมีการแก้ไขไดรเวอร์เพื่อให้สามารถทำงานได้ ตัวอย่างเช่น มัลติคาสต์แอดเดรส เนเบอร์ดิสเคิร์ฟเวอรี่ (Neighbor Discovery) เป็นต้น

7.2 ปัญหาและอุปสรรค

จากการศึกษาสามารถสรุปปัญหาที่เกิดจากการทำโครงงานได้ดังนี้

1) เนื่องจากไอพีรุ่น6 เป็นโปรโตคอลที่มีเทคโนโลยีใหม่อยู่ ยังมีแหล่งข้อมูลไม่มากพอ และมักจะเป็นข้อมูลที่ซ้ำกัน จึงใช้เวลาในการศึกษาหาข้อมูลค่อนข้างนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เนื่องจากโครงงานนี้เป็นการนำเทคโนโลยีหลายชนิดมารวมกัน การจะทำให้เทคโนโลยีแต่ละอย่างสามารถที่จะทำงานร่วมกันได้นั้นใช้เวลานาน และตรวจหาข้อผิดพลาดได้ยาก

3) การที่จะทดสอบการทำงานของไอพีรุ่น 6 นั้นค่อนข้างยุ่งยาก เพราะยังไม่ได้ใช้แพร่หลาย ทำให้ต้องติดตั้งอุปกรณ์หลายอย่าง ถึงจะทำการทดสอบได้ แต่ก็ไม่ใช่สภาพแวดล้อมทำงานจริงของระบบเครือข่ายไอพีรุ่น 6

4) ชุดพัฒนาคอม86 ได้มานั้นมีปัญหาทางด้านฮาร์ดแวร์ เมื่อนำไปเปลี่ยนเอาตัวใหม่มาใช้ก็ยังมีปัญหาเกี่ยวกับการค้างค่อนข้างบ่อย หรือกดรีเซ็ตแล้วก็ไม่ทำงาน ต้องถอดปลั๊กออก เป็นต้น

7.3 แนวทางการพัฒนา

1) นักพัฒนาสามารถที่จะนำซอร์ดโค้ดไปพัฒนาต่อได้โดยจะสามารถแก้ไขและเพิ่มเติมส่วนต่างๆได้อย่างอิสระโดยนักพัฒนาสามารถพัฒนาในส่วนโปรแกรมประยุกต์เพิ่มเติมโดยนำไปไว้ที่ \app โดยให้ส่วนของ main program อยู่ที่ \proj\ucos เป็นต้น

2) นักพัฒนาสามารถพัฒนาโปรโตคอลแสต็กให้สามารถให้บริการอื่นเพิ่มเติมได้ เช่น ดีเอชซีพีรุ่น 6 (DHCPv6) ชนิดของแอดเดรสอื่นๆ

3) นักพัฒนาสามารถพัฒนาอุปกรณ์ฮาร์ดแวร์ส่วนขยายและซอฟแวร์ควบคุมที่เข้ามาทำงานร่วมกันกับอุปกรณ์นี้ เพื่อให้สามารถทำงานเป็นอุปกรณ์ชนิดอื่นได้ เช่น เชื่อมต่อกับแอ็กเซสพอยท์เพื่อทำงานกับระบบ เครือข่ายไร้สาย สร้างเกตเวย์บนระบบเครือข่ายที่สนับสนุนไอพีรุ่น 6 เป็นต้น

4) เนื่องจากไอพีรุ่น 6 เป็นโปรโตคอลที่สามารถทำงานร่วมกับระบบอื่นๆได้ จึงสามารถที่จะพัฒนาตัวอุปกรณ์ให้ทำงานกับระบบอื่นได้ด้วย

5) เนื่องจากระบบฝังตัวนั้นไม่ได้จำกัดอยู่แค่เพียงชุดพัฒนาคอม 86 เท่านั้น จึงสามารถที่จะนำเอาเฉพาะซอฟแวร์ของระบบไปปรับปรุงและประยุกต์ใช้กับอุปกรณ์ฝังตัวอื่นๆ เช่น ARM-7 ARM-9 เป็นต้น

6) พัฒนาไลท์เวททีซีพี/ไอพีแสต็กให้เป็นแสต็กแบบคู่ เพื่อที่จะได้สามารถให้บริการทั้งไอพีรุ่น 4 และไอพีรุ่น 6 พร้อมๆกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

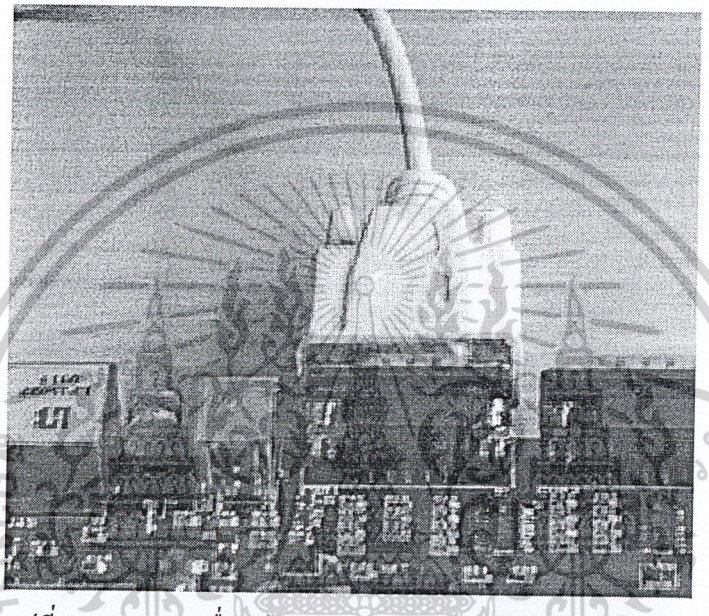
ภาคผนวก ก.

1. การใช้งานบอร์ดคอม 86

ในการติดตั้งใช้ชุดพัฒนาคอม86 ให้ปฏิบัติตามขั้นตอนต่างๆ ดังนี้

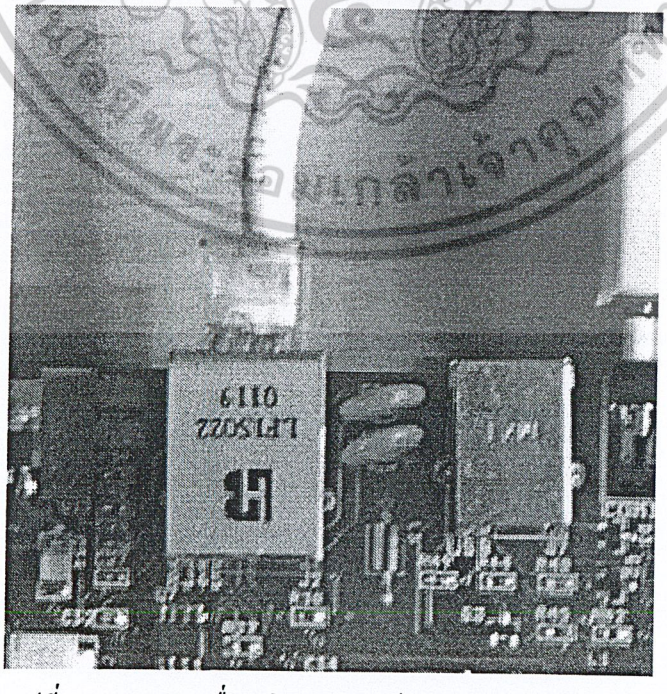
1. นำบอร์ดคอม86 ออกมาจากซองที่ใช้บรรจุเพื่อการขนส่งและตรวจสอบสภาพของบอร์ดในเบื้องต้น

2. เชื่อมต่อสายของ Serial cable ที่ให้มากับพอร์ตอนุกรมของคอมพิวเตอร์ที่ต้องการใช้งาน และเชื่อมต่อสายอีกด้านเข้ากับบอร์ดคอม86



รูปที่ ก-1 ภาพการเชื่อมต่อสาย serial console เข้ากับบอร์ดคอม86

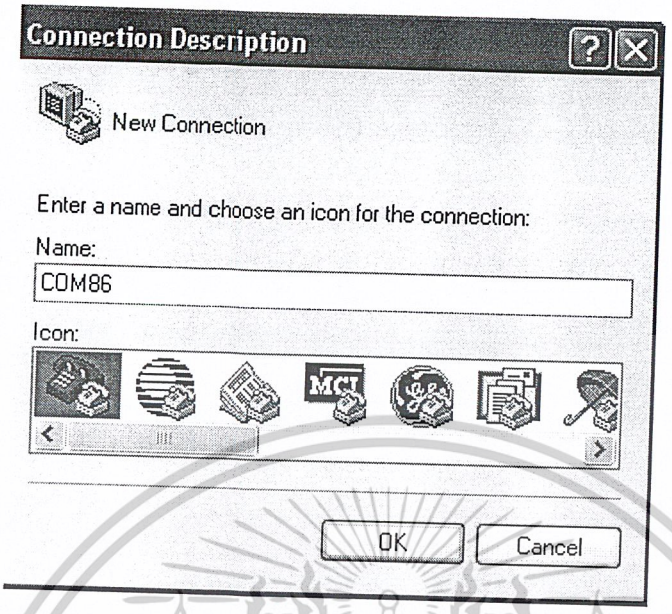
3. เชื่อมต่อสาย Ethernet เข้าบอร์ดคอม86 และเครื่องคอมพิวเตอร์หรือเน็ตเวิร์กฮับ*



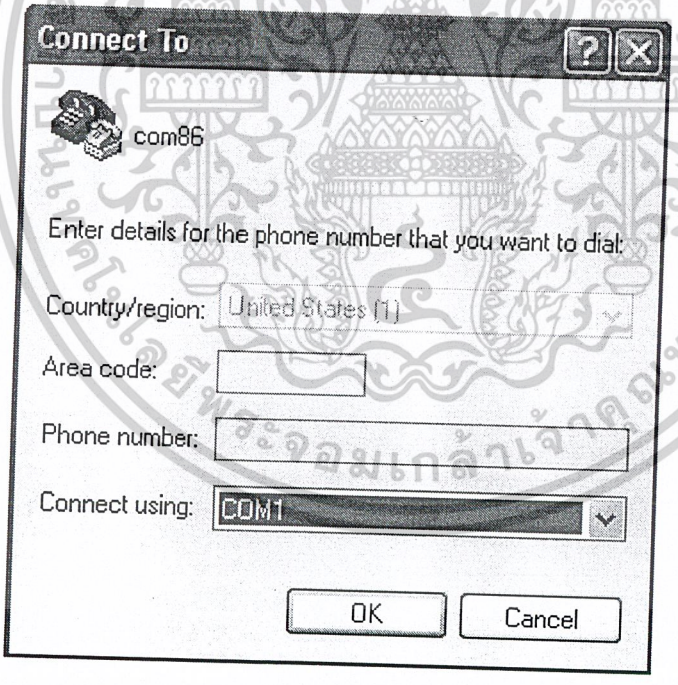
รูปที่ ก-2 ภาพการเชื่อมต่อสายอีเทอร์เน็ทเข้ากับบอร์ดคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เปิดโปรแกรม terminal ที่ใช้งานขึ้นมาและทำการกำหนดค่าต่างๆ ดังนี้

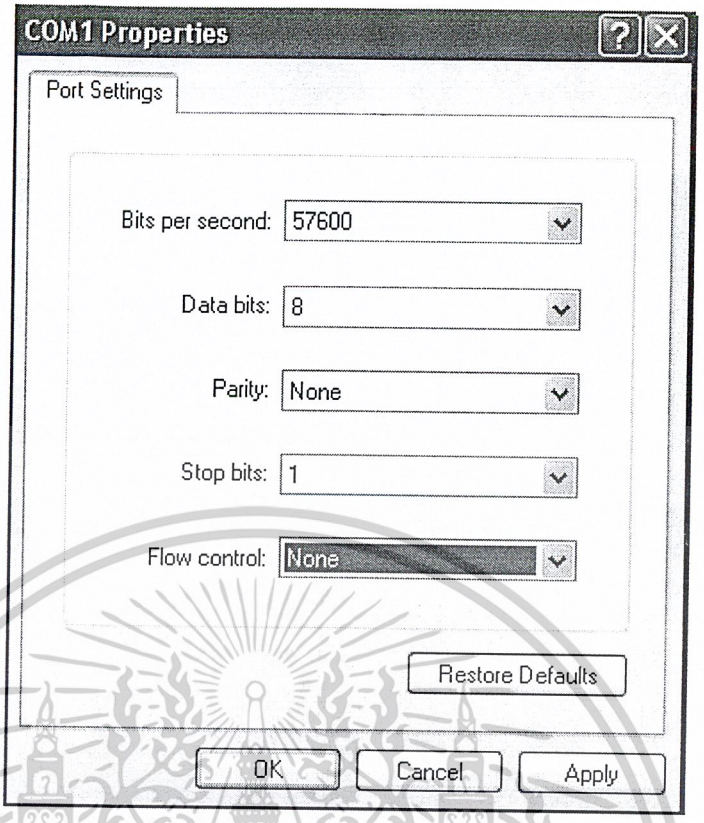


รูปที่ ก-3 ใส่ชื่อรูปแบบการเชื่อมต่อ



รูปที่ ก-4 เลือกพอร์ตอนุกรมที่มาเชื่อมต่อ

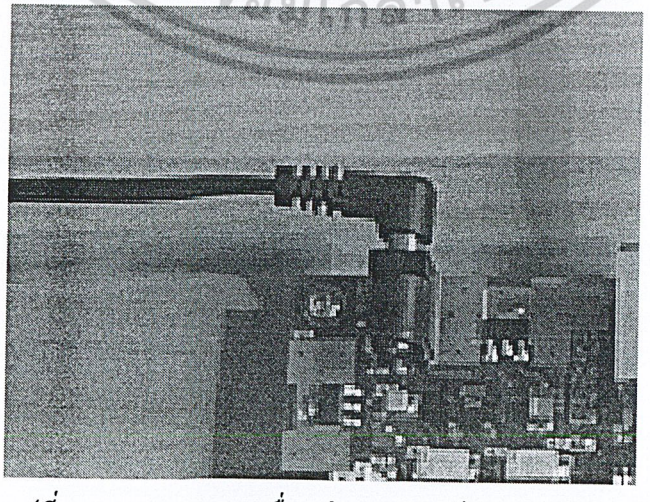
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-5 ภาพการตั้งค่าต่างๆ ของโปรแกรมเทอร์มินัล

- เลือกพอร์ตการเชื่อมต่อของคอมพิวเตอร์ ตามที่ได้ทำการเชื่อมต่อในขั้นตอนที่ 2
- กำหนดอัตราการรับส่งข้อมูลอยู่ที่ 57600 บิตต่อวินาที
- ข้อมูลเป็นแบบ 8 บิต
- ไม่มีพาริตี้
- Stop บิตเท่ากับ 1
- ยกเลิกการทำโฟลว์คอนโทรลทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ และสั่งเริ่มการเชื่อมต่อ

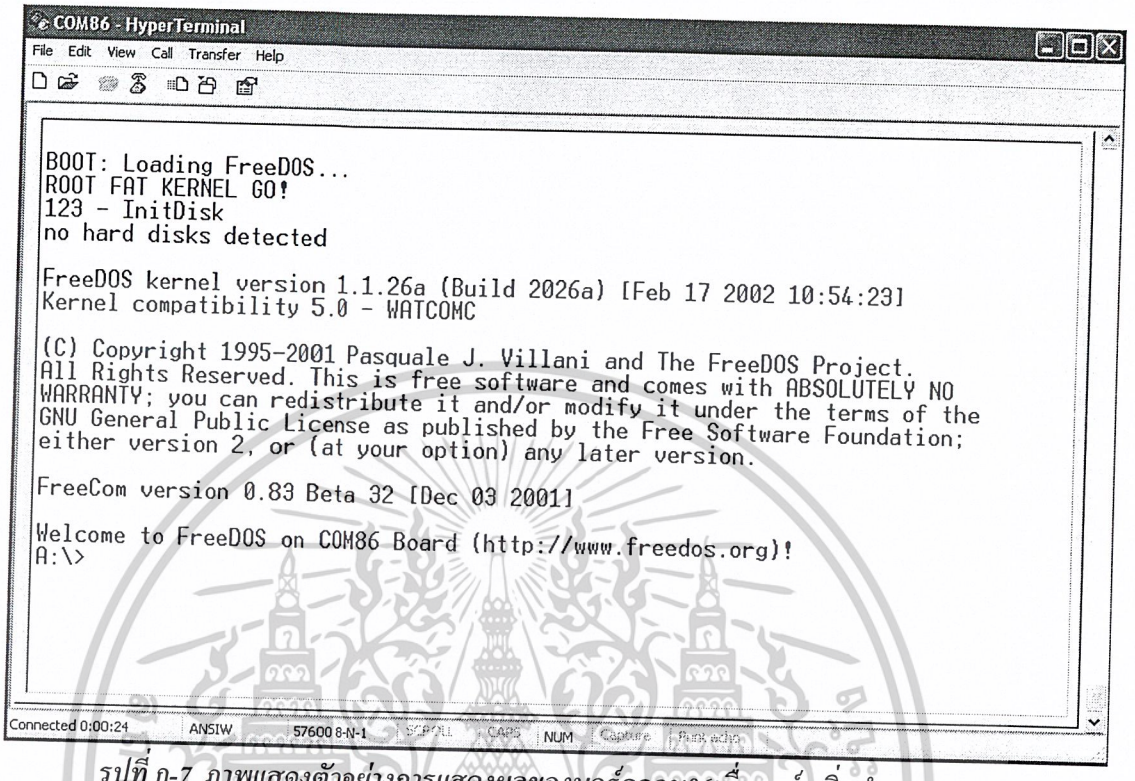
5. ต่อปลั๊กของอแดปเตอร์เข้ากับบอร์ดคอม86 และต่ออแดปเตอร์เข้ากับแหล่งจ่ายไฟของอาคาร



รูปที่ ก-6 ภาพแสดงการเชื่อมต่ออแดปเตอร์กับบอร์ดคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กดรีเซตสวิตช์จะเห็นโปรแกรมบนบอร์ดเริ่มทำงานโดยการแสดงผลผ่านโปรแกรมเทอร์มินัลที่ใช้งานและสามารถใช้งานชุดคำสั่งบนดอสได้



รูปที่ ก-7 ภาพแสดงตัวอย่างการแสดงผลของบอร์ดคอม86เมื่อบอร์ดเริ่มทำงาน

2. การติดตั้งโปรแกรม

1. การ transfer file ลงไปในชุดพัฒนาคอม 86

สามารถทำได้โดยการเรียกใช้ program transfer.exe ของชุดพัฒนาคอม 86 ดังนี้

transfer [/s][/r][/h][/?] filename

/s = to send file

/r = to receive file

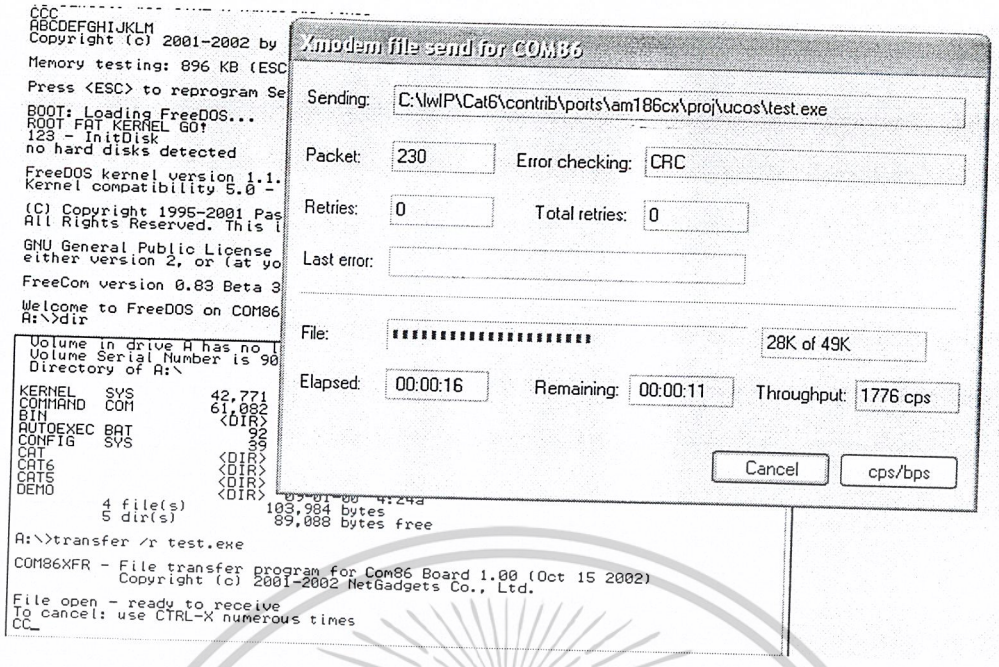
/?,/h = to show help

โดยที่ default option คือ /r

```
A:\>Transfer program1.exe
```

รูปที่ ก-8 การ transfer fileกับอุปกรณ์คอม 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

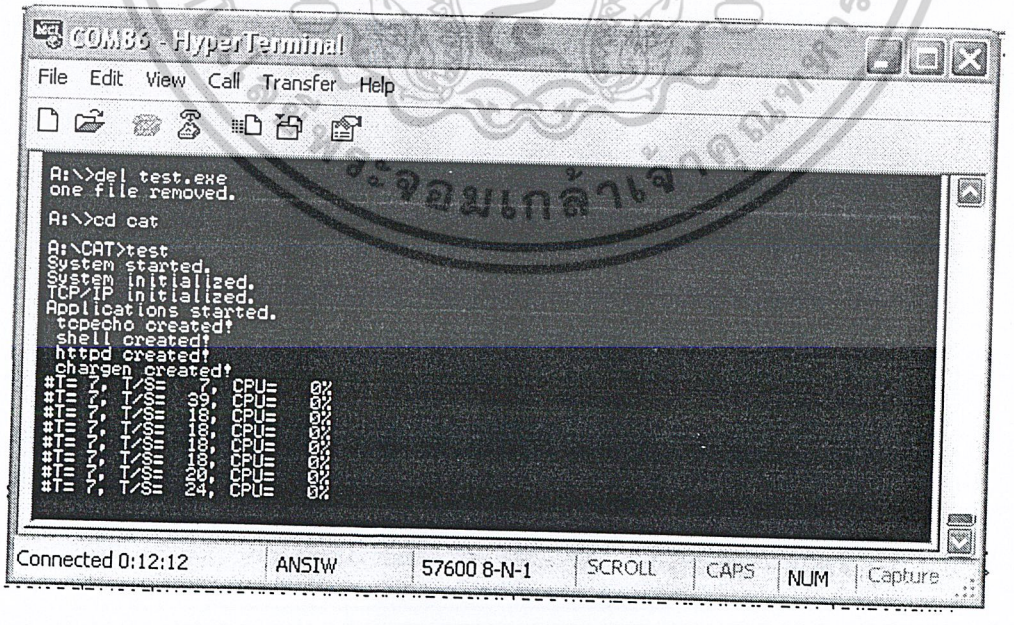


รูปที่ ก-9 การส่งไฟล์โดยใช้เครื่องมือของโปรแกรมไฮเปอร์เทอร์มินัล

โดยหลังจากใช้คำสั่งนี้ที่ folder ที่ต้องการที่จะติดตั้งโปรแกรมแล้วให้ส่งไฟล์ test.exe จากเครื่องคอมพิวเตอร์ส่วนบุคคลมาที่บอร์ดคอม 86 โดยใช้เครื่องมือในการส่ง file ของโปรแกรมไฮเปอร์เทอร์มินัล ซึ่งถ้าต้องการใช้งานโปรแกรมเกี่ยวกับไอพีรุ่น 4 ก็ให้ส่งไฟล์ test.exe ของไอพีรุ่น4 ถ้าต้องการใช้งานเกี่ยวกับไอพีรุ่น 6 ก็ให้ส่งไฟล์ test.exe ของไอพีรุ่น 6 มา

2.รันโปรแกรม

สั่งรันโปรแกรม test.exe ก็สามารถที่จะใช้งานได้ทันที



รูปที่ ก-10 การรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

1. การติดตั้งระบบเพื่อทำการทดสอบเกี่ยวกับไอพีรุ่น 6

ในที่นี้ใช้เครื่องคอมพิวเตอร์ส่วนบุคคลพร้อมการ์ดแลน ติดตั้งระบบปฏิบัติการ Linux Red hat 9.0 kernel 2.4.23-0

2. ทดสอบว่า kernel ปัจจุบันที่ใช้อยู่มี IPv6 module อยู่แล้วหรือไม่

1. ตรวจสอบใน /proc-file-system โดยจะต้องมี /proc/net/if_inet6 อยู่
2. ใช้คำสั่ง #test –f /proc/net/if_net6 && echo "Running kernel is IPv6 ready" ถ้า fail แสดงว่าไม่ได้มีการติดตั้ง IPv6 module ไว้

3. การโหลด IPv6 module

1. ใช้คำสั่ง #modprobe ipv6 ถ้าสำเร็จ module จะอยู่ใน list สามารถตรวจสอบได้โดยการใช้คำสั่ง #lsmod | grep –w 'ipv6'&& echo "IPv6 module successfully loadded" ถ้า fail แสดงว่าการติดตั้ง IPv6 module ไม่สำเร็จ
2. ใช้วิธี automatically loading of module โดยการเติมบรรทัดต่อไปนี้ใน configuration file ของ kernel module loader ซึ่งโดยปกติจะอยู่ที่ /etc/modules.conf หรือ /etc/conf.modules
   alias net-pf-10 ipv6 on แล้วทดสอบเหมือนข้อ 1
3. ถ้าทำทั้ง 2 ข้อที่ผ่านมาแล้วไม่สำเร็จ ต้องใช้วิธีคอมไพล์ kernel ใหม่ที่มี IPv6 อยู่ด้วย โดย
   - update ตัว contribution ให้มีตัวที่รองรับ IPv6 out-of-box
   - compile kernel ใหม่
   - recompile kernel sources ที่ได้มาจาก Linux distribution
   - Compile kernel ด้วย USAGI extensions

4. เรียกใช้งาน IPv6 จากระบบปฏิบัติการ Linux เพื่อทำการทดสอบ

หลังจากสั่งรันโปรแกรม test.exe ของไอพีรุ่น 6 แล้วให้ใช้คำสั่งดังต่อไปนี้ ตามลำดับ

1. #modprobe ipv6
2. #ifconfig eth0 up
3. #ifconfig eth0 promisc
4. #ping6 -I eth0 ff02::0302:03ff:fe04:0506

โดยที่ ff02::0302:03ff:fe04:0506 เป็น IPv6 address ที่กำหนดให้กับอุปกรณ์คอม 86 ซึ่งถ้าไม่มีข้อผิดพลาดต้องได้รับการตอบกลับ โดยสามารถทดสอบโดยการ #ping6 –I ff02::1 เลยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

ในการคอมไพล์โปรแกรมนั้นจะต้องมีการสร้าง makefile ขึ้นมาเพื่อรันโปรแกรมที่ใช้งานซึ่งภายใน makefile ซึ่งประกอบด้วย

1. สร้างตัวแปรที่ใช้เก็บค่าไดเร็คทอรี่

```
#                    *Translator Definitions*
ARCH_BLOCK_DIR=..\..\BLOCKS
ARCH_LWIP_DIR=..\..
LWIP_DIR=..\..\..\..\..\LWIP\SRC
UCOS_DIR=..\..\UCOS-II
APPS_LWIP_DIR=.\APPS
OBJ_DIR=OBJ
```

2.ใส่ตำแหน่งไดเร็คทอรี่ที่จะไปคอมไพล์

```
# Please specified your compiler path
COMPILER_PATH = D:\LANGS\BC
CC = $(COMPILER_PATH)\BIN\BCC +TEST.CFG
TASM = $(COMPILER_PATH)\BIN\TASM
TLIB = $(COMPILER_PATH)\BIN\TLIB
TLINK = $(COMPILER_PATH)\BIN\TLINK
LIBPATH = $(COMPILER_PATH)\LIB
DEFSINCPATH = $(COMPILER_PATH)\INCLUDE
ARCHINCPATH =
.\;$(ARCH_BLOCK_DIR);$(ARCH_LWIP_DIR)\INCLUDE;$(UCOS_DIR)\PORT;
LWIPINCPATH = $(LWIP_DIR)\INCLUDE;$(LWIP_DIR)\INCLUDE\IPV4
UCOSINCPATH = $(UCOS_DIR)\SOURCE
APPSINCPATH = $(APPS_LWIP_DIR)
```

3. อิมพลิซิทรูล์

```
#                    *Implicit Rules*
.c.obj:
 $(CC) -c {$< }
.cpp.obj:
 $(CC) -c {$< }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.สร้างรายการที่จะไปบอกว่าจะรันได้ก็ต่อเมื่อมีการติดต่อกับออบเจ็กเหล่านี้

```
#                   *List Macros*


EXE_dependencies = \
 $(OBJ_DIR)\ucos_ii.obj      \
 $(OBJ_DIR)\os_cpu_a.obj     \
 $(OBJ_DIR)\os_cpu_c.obj     \
 $(OBJ_DIR)\os_dbg.obj       \
 $(OBJ_DIR)\com86nic.obj     \
 $(OBJ_DIR)\sys_arch.obj     \
 $(OBJ_DIR)\pbuf.obj         \
 $(OBJ_DIR)\stats.obj        \
 $(OBJ_DIR)\sys.obj          \
 $(OBJ_DIR)\tcp.obj          \
 $(OBJ_DIR)\tcp_in.obj       \
 $(OBJ_DIR)\tcp_out.obj      \
 $(OBJ_DIR)\udp.obj          \
 $(OBJ_DIR)\api_lib.obj      \
```

5. สร้างลิงค์ที่จะไปติดต่อกับออบเจ็กที่ได้กล่าวข้างต้น

```
#                   *Explicit Rules*
test.exe: test.cfg $(EXE_dependencies)
 $(TLINK) /s /c /P- @&&|
 $(LIBPATH)\c0l.obj      +
 $(OBJ_DIR)\test.obj     +
 $(OBJ_DIR)\ucos_ii.obj      +
 $(OBJ_DIR)\os_cpu_a.obj     +
 $(OBJ_DIR)\os_cpu_c.obj     +
 $(OBJ_DIR)\os_dbg.obj       +
 $(OBJ_DIR)\com86nic.obj     +
 $(OBJ_DIR)\sys_arch.obj     +
 $(OBJ_DIR)\pbuf.obj         +
 $(OBJ_DIR)\stats.obj        +
 $(OBJ_DIR)\sys.obj          +
 $(OBJ_DIR)\tcp.obj          +
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
$(LIBPATH)\emu.lib      +
$(LIBPATH)\mathl.lib    +
$(LIBPATH)\cl.lib
```

6.จะมีการกำหนดค่าสำหรับไฟล์อิสระต่างๆว่าขึ้นอยู่กับส่วนไหน

```
#                   *Individual File Dependencies*
$(OBJ_DIR)\os_cpu_a.obj: test.cfg $(UCOS_DIR)\PORT\os_cpu_a.asm
$(TASM) /MX /ZI /O $(UCOS_DIR)\PORT\OS_CPU_A.ASM, $(OBJ_DIR)\OS_CPU_A.OBJ

$(OBJ_DIR)\os_cpu_c.obj: test.cfg $(UCOS_DIR)\PORT\os_cpu_c.c
$(CC) -c $(UCOS_DIR)\PORT\os_cpu_c.c

$(OBJ_DIR)\os_dbg.obj: test.cfg $(UCOS_DIR)\PORT\os_dbg.c
$(CC) -c $(UCOS_DIR)\PORT\os_dbg.c

$(OBJ_DIR)\ucos_ii.obj: test.cfg $(UCOS_DIR)\source\ucos_ii.c
$(CC) -c $(UCOS_DIR)\source\ucos_ii.c

$(OBJ_DIR)\com86nic.obj: test.cfg $(ARCH_BLOCK_DIR)\com86nic.asm
$(TASM) /MX /ZI /O $(ARCH_BLOCK_DIR)\COM86NIC.ASM, $(OBJ_DIR)\com86nic.obj
```

7.มีการสร้างรูปแบบคำสั่งว่าให้คอมไพล์แบบใด

```
#                   *Compiler Configuration File*
test.cfg: Makefile
 copy &&|
-ml
-1
-a-
-G
-O
-Ogemvlbpi
-Z
-d
-k-
-w-
-v-
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

[1] Jean J. Labrosse : MicroC/OS-II The Real-Time Kernel : Miller Freeman, 1999

[2] อภิเนตร อูนากูล : Object-Oriented Analysis and Design : โครงการผลิตตำราของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543

[3] นุกูล กระจาย : การเขียนโปรแกรมในดอสและวินโดวส์ด้วยบอร์แลนด์ C++5.0 : ซีเอ็ดยูเคชั่น , 2540

[4] William Stallings,Ph.D. : Operating Systems Internals and Design Principles : Prentice-Hall International, Inc. , 2001

[5] Douglas E. Comer : Internetworking With TCP/IP Vol I: Prentice-Hall International, Inc. , 2000

[6] เทพพิทักษ์ การุญบุญญานันท์ : กลเม็ดและเทคนิคในภาษา C/C++ : ซีเอ็ดยูเคชั่น , 2539

[7] W.Richard Stevens Gary R. Wright : TCP/IP Illustrated, Volume 2 The Implementation:Addison-Wesley Publishing Company,1995

[8] สุรศักดิ์ สงวนพงษ์ : สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี : ซีเอ็ดยูเคชั่น, 2543.

[9] Anthony Jones,Jim Ohlund : Network Programming For Microsoft Windows second Edition: Microsoft

[10] Phillip M.Adams : การเขียน ดอสดีไวซ์ไดรเวอร์ด้วยภาษา C : ซีเอ็ดยูเคชั่น , 2538

เวบไซด์ที่เกี่ยวข้อง

[1] http://www.sics.se/~adam/lwip/ : "lwIP"

[2] http://www.ucos-ii.com/ : "Micro C/OS-II"

[3] www.ipv6.org/ : "IPv6"

[4] http://www.webopedia.com/TERM/p/protocol_stack.html : "Protocol stack"

[5] www.6bone.net : "IPv6 และ IPv6 Testbed"

[6] www.ipv6.nectec.or.th : "IPv6 และ IPv6 Testbed"

[7] www.ietf.org : "มาตรฐาน RFC ต่างๆ"

[8] http://geocities.com/michaelanburaj/Embedded.html :"Lightweight TCP/IP stack for uC/OS-II"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้