

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

กล้องผ่านเครือข่ายอินเทอร์เน็ต

Internet Protocol Camera



T104183

โดย

นายกิตติศักดิ์

สุขวัฒนาภิรมย์

นายสุวรรต

บุตรกริม

นายอัศวิน

กีกก้อง

เลขหมู่.....

เลขทะเบียน 104183

วัน,เดือน,ปี 30 ต.ค. 2552



ปฏิญานีพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องผ่านเครือข่ายอินเทอร์เน็ต

Internet Protocol Camera

โดย

นายกิตติศักดิ์ สุขวัฒนาภิรมย์ 49015044

นายสุวรรต บุตรกริม 49015076

นายอัศวิน กีก้อง 49015081

อาจารย์ที่ปรึกษา

ผศ.อักรพล ตริรัตน์

อ.ศรวต์น์ ชิวปรีชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง กล้องผ่านเครือข่ายอินเทอร์เน็ต

Internet Protocol Camera

ผู้จัดทำ

นายกิตติศักดิ์ สุขวัฒนาภิรมย์ 49015044

นายสุวรรต บุตรกริม 49015076

นายอัศวิน กีก้อง 49015081

อาจารย์ที่ปรึกษา
(ผศ.อักรพล ตริรัตน์)

อาจารย์ที่ปรึกษา
(อ.สรวัฒน์ ชิวปรีชา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องผ่านเครือข่ายอินเทอร์เน็ต
Internet Protocol Camera

โดย นาย กิตติศักดิ์ สุขวัฒนาภิรมย์ 49015044
นายสุวรรต บุตรกริม 49015076
นายอัสวิน กีก้อง 49015081

อาจารย์ที่ปรึกษา

ผศ.อัครพล ตีร์รัตน์
อ.ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

โครงการนี้นำเสนอกำลังผ่านระบบเครือข่ายอินเทอร์เน็ต (IP CAMERA) โดยผู้ใช้บริการสามารถดูภาพและควบคุมการเคลื่อนไหวของตัวกล้องผ่านทางหน้าเว็บเพจของคอมพิวเตอร์ที่ต่ออยู่กับระบบเครือข่ายอินเทอร์เน็ตจากระยะไกลได้ โดยในโครงการนี้ประกอบด้วย ตัวกล้อง, X-PORT (ใช้ในการเชื่อมต่อกับระบบเครือข่ายอินเทอร์เน็ต) รวมทั้งไมโครคอนโทรลเลอร์เพื่อใช้ในการควบคุมการเคลื่อนไหวของตัวกล้อง

ABSTRACT

This project presents an implementation of Internet camera (IP CAMERA). This IP camera let users can see pictures and can control the position of this camera a predestined web-page. This project comprises of three main parts: camera, X-PORT module and Micro-controller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้จะไม่สำเร็จลงได้หากมิได้รับความช่วยเหลือและการแนะนำจาก
ท่าน อ.ศรวัฒน์ ชิวปรีชา รวมทั้งอาจารย์ทุกท่านที่ได้สั่งสอนและอบรมมา คณะผู้จัดทำขอขอบพระคุณ
ท่านอาจารย์ทุกท่าน ตลอดจนผู้ให้คำปรึกษาและแนะนำทุกท่านล้วนมีส่วนในความสำเร็จของปริญญา
นิพนธ์ฉบับนี้

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| | หน้า |
|--|------|
| สารบัญรูปภาพ | I |
| สารบัญตาราง | IV |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความสำคัญและที่มา | 1 |
| 1.2 วัตถุประสงค์ของโครงการ | 1 |
| 1.3 โครงสร้างพื้นฐานของระบบ | 2 |
| บทที่ 2 ทฤษฎีและหลักการ | 3 |
| 2.1 อินเทอร์เน็ต (Internet) | 3 |
| 2.1.1 ความหมายของอินเทอร์เน็ต | 3 |
| 2.1.1.1 หน้าที่และความสำคัญของอินเทอร์เน็ต | 3 |
| 2.1.2 อินเทอร์เน็ตในประเทศไทย | 4 |
| 2.1.3 เทคโนโลยีเครือข่าย | 5 |
| 2.1.3.1 ความหมายของระบบเครือข่าย | 5 |
| 2.1.4 ระบบอ้างอิงโมเดล OSI | 6 |
| 2.1.4.1 Application Layer | 7 |
| 2.1.4.2 Presentation Layer | 8 |
| 2.1.4.3 Session Layer | 9 |
| 2.1.4.4 Transport Layer | 9 |
| 2.1.4.5 Network Layer | 10 |
| 2.1.4.6 Data Link Layer | 10 |
| 2.1.4.7 Physical Layer | 10 |
| 2.1.5 เลขหมายอินเทอร์เน็ต (IP Address) | 11 |
| 2.1.5.1 IPv4 (Internet Protocol Address Version 4) | 12 |
| 2.1.5.2 IPv6 (Internet Protocol Address Version 6) | 12 |
| 2.1.6 การแบ่ง Subnet | 12 |
| 2.1.7 โครงสร้างของโปรโตคอล TCP/IP | 13 |
| 2.1.8 Process Layer หรือ Application Layer | 14 |
| 2.1.9 Host – To – Host Layer หรือ Transport Layer | 15 |
| 2.1.9.1 โปรโตคอล TCP | 16 |
| 2.1.9.2 โปรโตคอล UDP | 16 |

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | หน้า |
|---|------|
| 2.1.9.4 UDP Checksum | 18 |
| 2.1.10 Internetwork Layer | 18 |
| 2.1.10.1 IP Datagram | 18 |
| 2.1.10.2 โพรโทคอล IP | 19 |
| 2.1.10.3 การกำหนด IP address ให้กับอุปกรณ์ | 20 |
| 2.1.10.4 การ Bind IP address | 20 |
| 2.1.10.5 โพรโทคอล ARP | 21 |
| 2.1.10.6 โพรโทคอล ARP ย้อนกลับ RARP | 21 |
| 2.1.11 Network Interface Layer | 22 |
| 2.1.11.1 MAC Address | 22 |
| 2.1.11.2 Domain Name System (DNS) | 22 |
| 2.1.11.3 DHCP (Dynamic Host Configuration Protocol) | 24 |
| 2.1.12 เวิลด์ไวด์เว็บและ โพรโทคอลเอชทีทีพี (The World Wide Web; HTTP) | 24 |
| 2.1.12.1 HTTP message format | 25 |
| 2.1.12.2 NAT (Network Address Translation) | 28 |
| 2.1.13.3 การทำงานของ NAT | 29 |
| 2.1.12.4 Static NAT | 29 |
| 2.1.12.5 Dynamic NAT | 30 |
| 2.1.13 ไมโครชิป TCP/IP Stack (MICROCHIP) | 31 |
| 2.1.13.1 การใช้งานไมโครชิป TCP/IP Stack | 33 |
| 2.1.13.2 การเชื่อมต่อกับระบบเน็ตเวิร์ก | 33 |
| 2.1.13.3 โพรโทคอล HTTP2 และ อิมเมจไฟล์ MPFS2 | 34 |
| 2.2 สเต็ปเปอร์มอเตอร์ (STEPPER MOTOR) | 36 |
| 2.2.1 ชนิดของสเต็ปเปอร์มอเตอร์ | 36 |
| 2.2.2 เฟสของสเต็ปเปอร์มอเตอร์ | 38 |
| 2.2.3 การกระตุ้นและการควบคุมการหมุนของStepping Motor | 41 |
| 2.2.3.1 ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave) | 41 |
| 2.2.3.2 ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส | 41 |
| 2.2.3.3 ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป | 42 |
| 2.3 ET-XPORT V3 | 43 |
| 2.3.1 คุณสมบัติของ ET-XPORT V3 | 43 |
| 2.3.2 ข้อกำหนดด้านการใช้งานของ ET-XPORT V3 | 44 |
| 2.3.3 โครงสร้างของ ET-XPORT V3 | 44 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | หน้า |
|--|-----------|
| 2.3.4 การกำหนดค่าการใช้งาน ET-XPORT V3 กับระบบเครือข่าย Ethernet LAN | 45 |
| 2.3.5 โปรแกรมที่ใช้ในการพัฒนาของ ET-XPORT V3 | 46 |
| บทที่ 3 การคำนวณและการสร้างวงจร | 47 |
| 3.1 ส่วนประกอบฮาร์ดแวร์ของระบบ | 47 |
| 3.2 ส่วนเชื่อมต่อกล้อง C328 และ XPORT | 48 |
| 3.3 ส่วนของโปรแกรมการแสดงผลภาพผ่านหน้าจอกอมพิวเตอร์ | 50 |
| 3.3.1 การออกแบบหน้าการแสดงผลของภาพ | 50 |
| 3.3.2 ส่วนของโปรแกรมควบคุมการทำงานทั้งหมด | 50 |
| 3.4 ขั้นตอนการโหลดโปรแกรมเข้า XPORT | 52 |
| 3.5 การตั้งค่าการทำงานให้กับ XPORT | 58 |
| 3.6 การควบคุมการเคลื่อนไหวกของตัวกล้อง | 60 |
| 3.6.1 ส่วนการเชื่อมต่อคอมพิวเตอร์กับไมโครคอนโทรลเลอร์เบอร์ PIC16F876A ผ่านพอร์ตอนุกรม RS-232 | 61 |
| 3.6.2 ส่วนการเชื่อมต่อไมโครคอนโทรลเลอร์เบอร์ PIC16F876A กับวงจรขับกระแส | 61 |
| 3.6.3 วงจรรวมการควบคุมการเคลื่อนไหวกของตัวกล้องด้วยคอมพิวเตอร์ผ่านพอร์ตอนุกรม | 62 |
| 3.6.4 ส่วนตัว Ethernet Module | 65 |
| บทที่ 4 ผลการทดลอง และวิเคราะห์ผลการทดลอง | 66 |
| 4.1 การทดลองวัดแรงดันที่เฟสแต่ละเฟสของสเต็ปเปอร์มอเตอร์ | 66 |
| 4.2 การทดลองการควบคุมการเคลื่อนไหวกของกล้องผ่านทางหน้าเว็บเพจ | 74 |
| 4.3 โปรแกรมแสดงผลภาพผ่านทางหน้าจอกอมพิวเตอร์ | 76 |
| 4.4 การติดตั้งและการนำไปใช้งาน | 78 |
| บทที่ 5 บทสรุปและวิจารณ์ | 79 |
| 5.1 สรุปผลการทดลอง | 79 |
| 5.1.1 การแสดงผลภาพ | 79 |
| 5.1.2 การควบคุมการเคลื่อนไหวกของฐานกล้อง | 79 |
| 5.2 ปัญหาที่พบในการทดลองและแนวทางแก้ไข | 79 |
| 5.2.1 การแสดงผลภาพ | 79 |
| 5.2.2 การควบคุมการเคลื่อนไหวกของฐานกล้อง | 79 |

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญรูปภาพ

| | หน้า |
|--|------|
| 1.1 โครงสร้างพื้นฐานของระบบ | 2 |
| 2.1 แสดงชั้นของการเชื่อมต่อระหว่างคอมพิวเตอร์ 2 เครื่อง | 6 |
| 2.2 Data Structures | 14 |
| 2.3 แสดงการใช้งาน port ของแต่ละโปรโตคอล | 15 |
| 2.4 แสดงการส่งข้อมูลจาก Application ไปยัง Host – To – Host Layer | 16 |
| 2.5 แสดงกลไกการส่งข้อมูลด้วยโปรโตคอล UDP | 17 |
| 2.6 แสดงการ Bind IP address | 21 |
| 2.7 HTTP request-response behavior | 25 |
| 2.8 General format of a request message | 26 |
| 2.9 General format of a response message | 28 |
| 2.10 เป็นการทำงานแบบ Static NAT | 30 |
| 2.11 เป็นการทำงานแบบ Dynamic NAT | 30 |
| 2.12 การเชื่อมต่อ ENC28J60 | 32 |
| 2.13 PIC18F97J60 ไมโครคอนโทรลเลอร์ที่มีโมดูล Ethernet ในตัว | 32 |
| 2.14 แผนภาพของการทำงานของโปรโตคอล HTTP2 | 34 |
| 2.15 โครงสร้างสเต็ปเปอร์มอเตอร์ชนิดวาริเอเบิลรีลักเตนซ์และสัญลักษณ์ | 37 |
| 2.16 สเต็ปเปอร์มอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต | 37 |
| 2.17 สเต็ปเปอร์มอเตอร์ชนิดไฮบริดขนาด 5 เฟส | 38 |
| 2.18 ลักษณะการพันขดลวดบนสเตเตอร์ | 39 |
| 2.19 สัญลักษณ์ โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบไบ โฟลา 2 เฟส | 39 |
| 2.20 สัญลักษณ์ โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบยูนิ โฟลา 2 เฟส | 40 |
| 2.21 สัญลักษณ์ โครงสร้างมอเตอร์ 2 เฟสที่พันลวดแบบไบ โฟลาและแยกปลายของแต่ละขด | 40 |
| 2.22 สัญลักษณ์การต่อสายแบบต่างๆ | 40 |
| 2.23 แสดงการกระตุ้นหรือ ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave) | 41 |
| 2.24 แสดงการกระตุ้นหรือควบคุมแบบ Full Step 2 เฟส หรือ แบบ 2 เฟส | 42 |
| 2.25 แสดงการกระตุ้นหรือควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป | 42 |
| 2.26 แสดงแผนผัง โครงสร้างของ ET-XPORT V3 | 44 |
| 3.1 แสดงส่วนประกอบหลักของฮาร์ดแวร์ทั้งหมด | 47 |
| 3.2 การแสดงการเชื่อมต่อระหว่างกล่อง C328 และ X-Port | 48 |
| 3.3 โฟลว์ชาร์ตการทำงานของโปรแกรมของกล่อง C328 และ XPORT | 49 |
| 3.4 หน้าจอการแสดงผลภาพ | 50 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปร่าง(ต่อ)

| | หน้า |
|--|------|
| 3.5 โฟลว์ชาร์ตการทำงานของโปรแกรมดึงภาพและแสดงภาพ | 51 |
| 3.6 วิธีการตั้งค่า Internet Protocol (TCP/IP) | 52 |
| 3.7 วิธีการตั้ง Internet Protocol เพื่อการเชื่อมต่อ | 53 |
| 3.8 แสดงโปรแกรม Deviceinstaller ที่ใช้ร่วมกับ XPORT | 54 |
| 3.9 วิธีการโปรแกรมตัว XPORT หน้าที่ 1 | 54 |
| 3.10 วิธีการ โปรแกรมตัว XPORT หน้าที่ 2 | 55 |
| 3.11 วิธีการ โปรแกรมตัว XPORT หน้าที่ 3 | 55 |
| 3.12 วิธีการ โปรแกรมตัว XPORT หน้าที่ 4 | 56 |
| 3.13 วิธีการ โปรแกรมตัว XPORT หน้าที่ 5 | 56 |
| 3.14 วิธีการ โปรแกรมตัว XPORT หน้าที่ 6 | 57 |
| 3.15 วิธีการ โปรแกรมตัว XPORT หน้าที่ 7 | 57 |
| 3.16 การแสดงการตั้งค่าการใช้งาน | 58 |
| 3.17 การแสดงการตั้งค่าการใช้งาน | 59 |
| 3.18 ตั้งค่าผ่าน Port UART 1 | 59 |
| 3.19 ตั้งค่าผ่าน Port UART 2 | 60 |
| 3.20 การป้อนคำสั่งให้กับไมโครคอนโทรลเลอร์ด้วยโปรแกรม HyperTerminal | 60 |
| 3.21 วงจรการเชื่อมต่อคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรม | 61 |
| 3.22 วงจรการเชื่อมต่อไมโครคอนโทรลเลอร์กับวงจรขับเคลื่อน | 62 |
| 3.23 วงจรรวมการควบคุมการเคลื่อนไหวกของตัวกลิ้งด้วยคอมพิวเตอร์ผ่านพอร์ตอนุกรม | 63 |
| 3.24 โฟลว์ชาร์ตการควบคุมสเต็ปมอเตอร์ผ่านทางพอร์ตอนุกรม | 64 |
| 4.1 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนซ้าย | 66 |
| 4.2 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนขวา | 67 |
| 4.3 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนขึ้น | 68 |
| 4.4 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนลง | 69 |
| 4.5 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนซ้าย | 70 |
| 4.6 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนขวา | 71 |
| 4.7 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนขึ้น | 72 |
| 4.8 การป้อนสัญญาณให้แก่ละเฟสของมอเตอร์ให้หมุนลง | 73 |
| 4.9 หน้าเว็บเพจควบคุมการเคลื่อนไหวกของกลิ้ง | 74 |
| 4.10 การเชื่อมต่อตัว Ethernet Module เข้ากับคอมพิวเตอร์ | 75 |
| 4.11 ผลการทดลองเมื่อทำการกดปุ่มควบคุมต่างๆบนหน้าเว็บเพจ | 75 |
| 4.12 หน้าจอโปรแกรมทางด้านผู้ใช้งาน | 76 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

| | หน้า |
|--|------|
| 4.13 ภาพเคลื่อนไหวออกทางหน้าจอทางด้านผู้ใช้งาน | 76 |
| 4.14 ขาสัญญาณ RS-232 ส่งของตัว C328 และ XPORT | 77 |
| 4.15 ขาสัญญาณ RS-232 รับของตัว C328 และ XPORT | 77 |
| 4.16 การเชื่อมต่ออุปกรณ์ส่วนต่างๆ | 78 |
| 4.17 การเชื่อมต่ออุปกรณ์ส่วนต่างๆ เข้ากับกล้อง | 78 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

| | หน้า |
|--|------|
| 2.1 แสดงชุดโปรโตคอล OSI ที่ทำงานในแต่ละเลเยอร์ | 7 |
| 2.2 UDP Header | 17 |
| 2.3 รายละเอียด UDP Header | 17 |
| 2.4 Pseudo Header | 18 |
| 2.5 แสดงโครงสร้าง IP Header | 19 |
| 2.6 ค่าต่าง ๆ ของ IP Protocol Field | 20 |
| 2.7 ลำดับชั้นของโปรโตคอล TPC/IP | 32 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

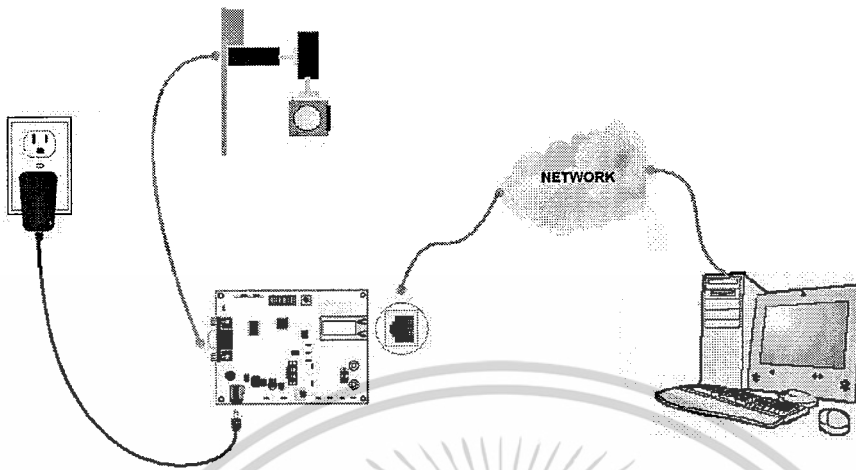
1.1 ความสำคัญและที่มา

ในปัจจุบันนี้ประเทศของเรากำลังประสบกับปัญหาทั้งทางด้านเศรษฐกิจและสังคม และจากปัญหาราคาน้ำมันแพงทำให้ราคาสินค้าต่างๆปรับตัวสูงขึ้น ส่งผลให้ค่าครองชีพในปัจจุบันสูงขึ้นด้วย และเศรษฐกิจที่มีการแข่งขันกันมากขึ้นทำให้หลายบริษัทต้องปิดกิจการลง ปล่อยให้คนตกงานมากขึ้นเป็นสาเหตุที่ทำให้เกิดกลุ่มมิจฉาชีพเพิ่มมากขึ้นทำให้ความปลอดภัยในชีวิตและทรัพย์สินทุกวันนี้ลดน้อยลง จึงทำให้อุปกรณ์รักษาความปลอดภัยหรือการเฝ้าระวังทรัพย์สินมีความสำคัญมาก และในปัจจุบันระบบเครือข่ายอินเทอร์เน็ตได้มีการใช้งานกันอย่างแพร่หลาย จึงได้เกิดแนวคิดที่จะสร้าง “กล้องผ่านระบบเครือข่ายอินเทอร์เน็ต (IP CAMERA)” โดยนำเอาประโยชน์ของอินเทอร์เน็ตมาประยุกต์เข้ากับการรักษาความปลอดภัย โดยเมื่อเครื่องคอมพิวเตอร์ปลายทางที่เชื่อมต่ออยู่กับเครือข่ายอินเทอร์เน็ตทำการติดต่อเข้ามา กล้องก็จะทำการส่งสัญญาณภาพผ่านทางอินเทอร์เน็ตโปรโตคอล โดยมีอุปกรณ์ X-PORT ทำหน้าที่เป็น Embedded Web Server ซึ่งผู้ใช้บริการสามารถดูภาพและควบคุมการเคลื่อนไหวของตัวกล้องเพื่อดูภาพในมุมต่างๆผ่านทางหน้าเว็บเพจของคอมพิวเตอร์ที่ต่ออยู่กับระบบเครือข่ายอินเทอร์เน็ตจากระยะไกลได้

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการสื่อสารผ่าน IP (Internet Protocol)
2. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์
3. เพื่อศึกษาและออกแบบชุดควบคุมกล้องโดยอาศัยหลักการของ Stepping Motor
4. เพื่อศึกษาการเขียนโปรแกรมเชิงวัตถุ
5. เพื่อนำไปประยุกต์ใช้งานด้านการรักษาความปลอดภัย

1.3 โครงสร้างพื้นฐานของระบบ



รูปที่ 1.1 โครงสร้างพื้นฐานของระบบ

จากรูปที่ 1.1 ข้างต้นสามารถที่จะอธิบายได้ดังนี้

1. ส่วนการแสดงผลภาพที่มีอุปกรณ์ X-PORT เป็น Embedded Web Server ทำหน้าที่ร้องขอข้อมูลภาพจากกล้องเพื่อส่งต่อข้อมูลผ่านเครือข่ายไปแสดงผลยังผู้ใช้งาน
2. ส่วนควบคุมการเคลื่อนไหวของกล้อง (Camera Control) จะเป็นส่วนที่รับคำสั่งควบคุมการเคลื่อนไหวจาก Ethernet Module โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมให้สเต็ปเปอร์มอเตอร์หมุนฐานกล้องไปตามทิศที่ต้องการ
3. ส่วนผู้ใช้งาน (Client Control) จะใช้คอมพิวเตอร์ที่เชื่อมต่ออยู่กับระบบเครือข่ายอินเทอร์เน็ตในการเข้าดูภาพและควบคุมการเคลื่อนไหวของตัวกล้องผ่านทางหน้าเว็บเพจ

บทที่ 2

ทฤษฎีและหลักการ

2.1 อินเทอร์เน็ต (Internet)

ในสังคมยุคข่าวสารเช่นปัจจุบันนี้ แทบจะไม่มีใครไม่เคยได้ยินคำว่า “อินเทอร์เน็ต” เหตุเพราะอินเทอร์เน็ตได้กลายเป็นส่วนหนึ่งในชีวิตประจำวันของคนจำนวนมากในโลกนี้ไปแล้ว ประมาณกันว่าในแต่ละวันมีผู้คนมากกว่า 50 ล้านคนในประเทศต่างๆ กว่า 150 ประเทศทั่วโลกกำลังใช้อินเทอร์เน็ตกันอยู่ อาจเป็นนักศึกษาคนหนึ่งในประเทศไทยที่กำลังสืบค้นข้อมูลจากห้องสมุดแห่งหนึ่ง หรือเป็นอาจารย์มหาวิทยาลัยคนหนึ่งในประเทศญี่ปุ่นกำลังสั่งซื้อหนังสือจากประเทศไทย เป็นต้น การประกอบกิจกรรมต่างๆ ในอินเทอร์เน็ตดังที่ได้กล่าวมานี้ เป็นตัวอย่างที่สะท้อนให้เห็นภาพของการสื่อสารที่ไร้พรมแดนได้อย่างชัดเจน

การใช้อินเทอร์เน็ตในปัจจุบันได้ขยายวงกว้างออกไปมากขึ้น โดยได้ก้าวล่วงเข้าไปในทุกสาขาอาชีพไม่ได้จำกัดอยู่เฉพาะด้านการศึกษาหรือการวิจัยเหมือนเมื่อเริ่มมีการใช้อินเทอร์เน็ตใหม่ๆ ด้วยคุณสมบัติการเข้าถึงกลุ่มเป้าหมายจำนวนมากๆ ได้ในเวลาอันรวดเร็วและใช้ต้นทุนในการลงทุนต่ำทำให้อินเทอร์เน็ตเป็นสิ่งที่พึงปรารถนาขององค์กรทั้งหลาย ได้มีความพยายามนำอินเทอร์เน็ตมาใช้เพื่อประโยชน์สำหรับหน่วยงานของตนในรูปแบบต่างๆ อาทิ การประชาสัมพันธ์องค์กร การโฆษณาสินค้า การค้าขาย การติดต่อสื่อสาร ฯลฯ นอกจากนี้อินเทอร์เน็ตยังกลายเป็นอีกสื่อหนึ่งของความบันเทิงภายในครอบครัวไปด้วยไม่ว่าจะเป็นการฟังวิทยุ ดูโทรทัศน์ หรืออ่านหนังสือพิมพ์ก็ตาม ล้วนแล้วแต่สามารถกระทำผ่านอินเทอร์เน็ตได้ทั้งสิ้น

2.1.1 ความหมายของอินเทอร์เน็ต

“อินเทอร์เน็ต” มาจากคำว่า International Network เป็นเครือข่ายของการสื่อสารข้อมูลขนาดใหญ่ อันประกอบด้วยเครือข่ายคอมพิวเตอร์จำนวนมาก เชื่อมโยงแหล่งข้อมูลจากองค์กรต่างๆ ทั่วโลกเข้าด้วยกัน

คำว่า “เครือข่าย” หมายถึง

1. การที่มีคอมพิวเตอร์ตั้งแต่ 2 เครื่องขึ้นไป เชื่อมต่อเข้าด้วยกันด้วยสายเคเบิล (ทางตรง) และหรือสายโทรศัพท์ (ทางอ้อม)
2. มีผู้ใช้คอมพิวเตอร์
3. มีการถ่ายเทข้อมูลระหว่างกัน

2.1.1.1 หน้าที่และความสำคัญของอินเทอร์เน็ต

การสื่อสารในยุคปัจจุบันที่กล่าวมานี้ว่าเป็นยุคไร้พรมแดนนั้น การเข้าถึงกลุ่มเป้าหมายจำนวนมากๆ ได้ในเวลาอันรวดเร็ว และใช้ต้นทุนในการลงทุนต่ำ เป็นสิ่งที่พึงปรารถนาของทุกหน่วยงาน และอินเทอร์เน็ตเป็นสื่อที่สามารถตอบสนองต่อความต้องการดังกล่าวได้ จึงเป็นความจำเป็นที่ทุกคนต้องให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสนใจและปรับตัวให้เข้ากับเทคโนโลยีใหม่นี้ เพื่อจะได้ใช้ประโยชน์จากเทคโนโลยีดังกล่าวอย่างเต็มที่

อินเทอร์เน็ต ถือเป็นระบบเครือข่ายคอมพิวเตอร์สากลที่เชื่อมต่อเข้าด้วยกัน ภายใต้มาตรฐานการสื่อสารเดียวกันเพื่อใช้เป็นเครื่องมือสื่อสารและสืบค้นสารสนเทศจากเครือข่ายต่างๆ ทั่วโลก ดังนั้น อินเทอร์เน็ตจึงเป็นแหล่งรวมสารสนเทศจากทุกมุมโลก ทุกสาขาวิชา ทุกด้าน ทั้งบันเทิงและวิชาการ ตลอดจนการประกอบธุรกิจต่างๆ

เหตุผลสำคัญที่ทำให้อินเทอร์เน็ตได้รับความนิยมแพร่หลายคือ

1. การสื่อสารบนอินเทอร์เน็ต ไม่จำกัดระบบปฏิบัติการของเครื่องคอมพิวเตอร์ คอมพิวเตอร์ที่ต่างระบบปฏิบัติการกันก็สามารถติดต่อสื่อสารกันได้
2. อินเทอร์เน็ตไม่มีข้อจำกัดในเรื่องของระยะทางไม่ว่าจะอยู่ในอาคารเดียวกันห่างกันคนละทวีปข้อมูลก็สามารถส่งผ่านถึงกันได้
3. อินเทอร์เน็ตไม่จำกัดรูปแบบของข้อมูล ซึ่งมีได้ทั้งข้อมูลที่เป็นข้อความอย่างเดียว หรืออาจมีภาพประกอบรวมไปถึงข้อมูลชนิดมัลติมีเดีย คือมีทั้งภาพเคลื่อนไหวและมีเสียงประกอบด้วยได้ คำอื่นที่ใช้ในความหมายเดียวกับอินเทอร์เน็ต คือ Information Superhighway และ Cyberspace

2.1.2 อินเทอร์เน็ตในประเทศไทย

ประเทศไทยได้เริ่มมีการติดต่อเชื่อมโยงเข้าสู่อินเทอร์เน็ตใน พ.ศ. 2535 โดยเริ่มที่สำนัก วิทยุบริการจุฬาลงกรณ์มหาวิทยาลัยซึ่งได้เช่าวงจรสื่อสารความเร็ว 9600 บิตต่อวินาทีจากการสื่อสารแห่งประเทศไทย ต่อมาใน พ.ศ. 2536 เนคเทคได้เช่าวงจรสื่อสารความเร็ว 64 กิโลบิตต่อวินาที ซึ่งช่วยเพิ่มความสามารถในการขนถ่ายข้อมูลทำให้ประเทศไทยมีวงจรสื่อสารระหว่างประเทศ 2 วงจร หน่วยงานต่างๆ ที่เข้าร่วมเชื่อมโยงเครือข่ายในระยะแรกๆ ได้แก่สถาบันอุดมศึกษาต่างๆ และต่อมาได้ขยายไปยังหน่วยงานราชการอื่นๆ

สำหรับภาคเอกชน ได้มีการก่อตั้งบริษัทสำหรับให้บริการอินเทอร์เน็ตแก่เอกชนและบุคคลทั่วไป ที่นิยมเรียกกันว่า ISP (Internet Service Providers) หลายราย เช่น ศูนย์บริการอินเทอร์เน็ตแห่งประเทศไทย (Internet Thailand) บริษัทเคเอสซีคอมเมอร์เชียลอินเทอร์เน็ตจำกัด (Internet KSC) บริษัทล็อกซเลย์ อินฟอร์เมชันจำกัด (Loxinfo) เป็นต้น โดยในการพิจารณาเลือกใช้บริการจาก ISP เอกชนเหล่านี้ สิ่งที่ต้องคำนึงถึงคือ

1. อัตราค่าใช้จ่ายโดยรวมทั้งค่าสมัครเป็นสมาชิกและค่าใช้จ่ายเป็นรายครั้งรายเดือนหรือรายปี
2. จำนวนคู่สายโทรศัพท์ว่ามีให้ใช้ติดต่อมากเพียงพอหรือไม่ เพราะถ้ามีไม่มากก็จะเสียเวลารอคอยนานกว่าจะเชื่อมต่อได้
3. ความเร็วของสายที่ใช้
4. พื้นที่ในการให้บริการควรเลือกใช้ ISP ที่อยู่ในจังหวัด หรือพื้นที่ใกล้เคียงจะเหมาะสมกว่า

เพราะ ISP ส่วนใหญ่มักให้บริการในเขตกรุงเทพมหานคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 เทคโนโลยีเครือข่าย

การพัฒนาอย่างรวดเร็วของเทคโนโลยีด้าน Internet Protocol (IP) ทำให้อินเทอร์เน็ตซึ่งเป็นเครือข่ายคอมพิวเตอร์ที่มีแหล่งข้อมูลที่ใหญ่ที่สุด ในปัจจุบันมีความจำเป็นสำหรับประชาชนทุกคนที่สามารถเข้าถึงหาข้อมูลและข่าวสารต่างๆ ที่ต้องการได้อย่างรวดเร็ว และยังรวมถึงการใช้ติดต่อสื่อสารระหว่างบุคคลและแหล่งข้อมูลต่างๆ ได้ทั่วทุกมุมโลกตลอดเวลา

IP Network หรือ โครงข่ายไอ-พี เป็นโครงข่ายมาตรฐานที่นิยมใช้ในการติดต่อสื่อสารที่มีความทันสมัย และความเร็วสูงสามารถรับสัญญาณได้ทั้งเสียงข้อมูลภาพวีดิโอและภาพเคลื่อนไหวโดยอยู่ในรูปสัญญาณดิจิทัลที่เรียกว่า ไอ-พี แพ็กเกจ โครงข่ายนี้รองรับการใช้งานร่วมกับโครงข่ายโทรคมนาคมใหม่ๆ และทันสมัย ได้แก่ โครงข่ายใยแก้วความเร็วสูงแบบ SDH (Synchronous Digital Hierarchy) โครงข่ายดิจิทัลสวิตซ์ที่รองรับหลายความเร็วแบบ ATM (Asynchronous Transfer Mode) โครงข่ายใยแก้วความเร็วสูงแบบ Optical Internet Protocol โดยสัญญาณที่ระดับความเร็วสูงตั้งแต่ 2Mbps-2.5Gbps และพร้อมพัฒนาใช้งานร่วมกับเครือข่ายอินเทอร์เน็ตแบบไร้สายในโครงข่าย โทรศัพท์เคลื่อนที่ยุคที่ 3 (Third-Generation Digital Mobile Communication Systems)

2.1.3.1 ความหมายของระบบเครือข่าย

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) หรือเรียกสั้น ๆ ว่า ระบบเครือข่าย (Network) ประกอบไปด้วยเครื่องคอมพิวเตอร์หลาย ๆ เครื่องที่สามารถติดต่อสื่อสารกันเพื่อแลกเปลี่ยนข้อมูลกัน การติดต่อจะผ่านทางช่องการสื่อสารต่าง ๆ เช่น สายโทรศัพท์ สายไฟหรือผ่านทางสื่อแบบอื่น ๆ ได้แก่ โมเด็ม ไมโครเวฟ สัญญาณอินฟราเรด เป็นต้น ซึ่งโดยทั่วไปจะหมายถึงการที่เรานำเครื่องคอมพิวเตอร์อย่างน้อย 2 เครื่องมาเชื่อมต่อกัน วัตถุประสงค์ที่ต้องต่อกันนี้ มักเกิดจากความต้องการที่จะใช้ทรัพยากรของระบบร่วมกัน เช่น ใช้นโยบายข้อมูลในคลังร่วมกันใช้งานเครื่องพิมพ์เลเซอร์ที่มีอยู่เครื่องเดียวร่วมกันต้องการส่งข้อมูลให้กับบุคคลอื่นในระบบไปใช้งานหรือต้องการติดต่อสื่อสารระหว่างกัน เป็นต้น ฉะนั้น ระบบเครือข่ายคือ ระบบที่นำเอาเครื่องคอมพิวเตอร์ส่วนบุคคล (PC หรือ Personal Computer) แต่ละเครื่องมาต่อเชื่อมกันด้วยกลวิธีทางระบบคอมพิวเตอร์นั่นเอง

การแบ่งชนิดของระบบเครือข่ายสามารถดูได้จากลักษณะการติดตั้งใช้งานทางภูมิศาสตร์จึงสามารถแบ่งระบบเครือข่ายได้เป็น 3 แบบด้วยกัน คือ

1. ระบบเครือข่ายระดับท้องถิ่น (Local Area Network หรือ LAN) เป็นระบบเครือข่ายที่ใช้งานอยู่ในบริเวณไม่กว้างอาจจะใช้ภายในอาคารเดียวกันหรืออาคารที่อยู่ใกล้กัน เช่น ภายในมหาวิทยาลัย อาคารสำนักงาน คลังสินค้าหรือโรงงาน เป็นต้น การส่งข้อมูลสามารถทำได้ด้วยความเร็วสูงและมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อผิดพลาดน้อยระบบเครือข่ายระดับท้องถิ่นจึงถูกออกแบบมาให้ช่วยลดต้นทุนและเพื่อเพิ่มประสิทธิภาพในการทำงานและใช้งานอุปกรณ์ต่าง ๆ ร่วมกัน

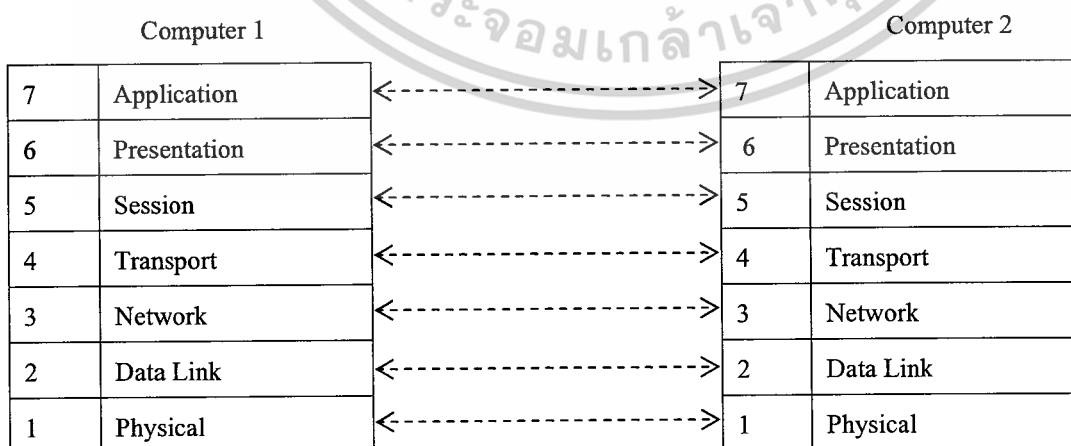
2. ระบบเครือข่ายระดับประเทศ (Wide Area Network หรือ LAN) เป็นระบบเครือข่ายที่ติดตั้งใช้งานอยู่ในบริเวณกว้าง เช่น ระบบเครือข่ายที่ติดตั้งใช้งานทั่วโลก โดยปกติมีอัตราการส่งข้อมูลที่ต่ำและมีโอกาสเกิดข้อผิดพลาดการส่งข้อมูลอาจใช้อุปกรณ์ในการสื่อสาร เช่น โมเด็ม มาช่วย

3. ระบบเครือข่ายระดับเมือง (Metropolitan Area Network หรือ MAN) เป็นระบบเครือข่ายที่มีขนาดอยู่ระหว่าง LAN และ WAN เป็นระบบเครือข่ายที่ใช้ภายในเมืองหรือจังหวัดเท่านั้น

2.1.4 ระบบอ้างอิงโมเดล OSI

องค์การมาตรฐานสากลหรือ ISO (International Organization for Standardization) ได้มีการศึกษาและหาแนวทางในการกำหนดมาตรฐานของเครือข่าย ซึ่งจะทำให้การเชื่อมต่อระหว่างเครือข่ายสามารถทำได้ ISO ค้นพบว่าระบบเครือข่ายจะมีกิจกรรมพื้นฐานต่าง ๆ เช่น การรับส่งข้อมูลการเข้าใช้งานในเครือข่าย การส่งพิมพ์ที่เครื่องพิมพ์ในเครือข่าย เป็นต้น ดังนั้น ISO ได้จัดแบ่งกิจกรรมเหล่านั้นออกเป็นงานย่อยและกำหนดเป็นโมเดลแบ่งเป็นชั้น ๆ ตามลำดับ เรียกว่า มาตรฐาน Open System Interconnection หรือ OSI ด้วยวิธีการแบ่งกิจกรรมที่ซับซ้อนในเครือข่ายออกเป็นงานย่อย ๆ เพื่อจะช่วยให้การออกแบบและใช้งานเครือข่ายรวมถึงการเชื่อมโยงกันเป็นไปได้อย่างความสะดวก และมีวิธีการทำงานอยู่ในกรอบเดียวกัน โมเดล OSI นี้เป็นต้นแบบแนวคิดในการสร้างเครือข่ายระบบย่อยของการสื่อสารข้อมูลเป็นส่วนที่มีความซับซ้อนมากไม่ว่าจะเป็นทางด้าน Hardware หรือ Software ในสมัยแรก ๆ นั้นการพยายามสร้างโปรแกรมเพื่อการติดต่อสื่อสารนั้นจะสร้างโปรแกรมที่ค่อนข้างซับซ้อน ด้วยการมีส่วนประกอบหลายส่วนใน 1 โปรแกรม โปรแกรมประเภทนี้โดยมากจะเขียนด้วยภาษาแอสเซมบลี

OSI Reference Model



รูปที่ 2.1 แสดงชั้นของการเชื่อมต่อระหว่างคอมพิวเตอร์ 2 เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดโพรโทคอล OSI ประกอบด้วยโพรโทคอลมาตรฐานหลายโพรโทคอลโพรโทคอลเหล่านี้เป็นส่วนหนึ่งของโครงการนานาชาติเพื่อพัฒนาโพรโทคอลและมาตรฐานอื่น ๆ เพื่อเอื้ออำนวยให้อุปกรณ์เครือข่ายที่ผลิตจากบริษัทต่าง ๆ สามารถทำงานร่วมกันได้ข้อกำหนดมาตรฐาน OSI ถูกจัดทำโดย 2 องค์กรคือ ISO และ ITU-T

ตารางที่ 2.1 แสดงชุดโพรโทคอล OSI ที่ทำงานในแต่ละเลเยอร์

| OSI Reference Model | | OSI Protocols |
|---------------------|--------------|--|
| 7 | Application | CMIP, DS, FTAM, MHS, VTP |
| 6 | Presentation | Presentation Service/Presentation Protocol |
| 5 | Session | Session Service/Session Protocol |
| 4 | Transport | TP0, TP1, TP2 TP3, TP4, TP5 |
| 3 | Network | CONP/CMNS, CLNP/CLNS, IS-IS, ES-IS |
| 2 | Data Link | IEEE 802.2, IEEE 802.3, IEEE802.5, FDDI, X.25 |
| 1 | Physical | IEEE 802.2 Hardware, IEEE802.3 Hardware, IEEE802.5 Hardware, FDDI, x.25 Hardware |

OSI Reference Model ถูกออกแบบมาให้ประกอบด้วย 7 ลำดับชั้น ได้แก่

2.1.4.1 Application Layer

เป็นชั้นบนสุดของโมเดลจะเป็นส่วนที่ทำการติดต่อระหว่างแอปพลิเคชันของเครือข่ายกับใช้งาน ตัวอย่างแอปพลิเคชันของเครือข่ายก็เช่น ระบบจดหมายอิเล็กทรอนิกส์ การโอนถ่ายเพิ่มข้อมูล (File Transfer) การขอเข้าใช้ระบบคอมพิวเตอร์ในเครือข่าย (Host Terminal) การจัดเพิ่มข้อมูลในลักษณะต่าง ๆ เป็นต้น นอกจากนี้ยังรวมถึงการบริการทางด้านการแลกเปลี่ยนเอกสารข้อความต่าง ๆ Application Layer จะทำหน้าที่จัดการเรื่องต่างๆ ของเครือข่ายตามที่คุณต้องการนั่นเองโดยจะอยู่ในระดับบนที่ใกล้ชิดกับผู้ใช้ที่สุดการเข้าไปใช้บริการระดับ Application Layer โดยปกติจะใช้โดยการพิมพ์คำสั่งต่าง ๆ ผ่านทางระบบโปรแกรมควบคุมเครื่อง (Operating System หรือ OS) โดย OS จะถือว่าเป็นโปรแกรมในการสื่อสารนั่นเองการบริการของลำดับชั้นนี้จะแสดงให้เห็นให้ผู้ใช้งานได้เข้าใจในทันที โดยไม่ว่าการเอ็กสแตร์นเป็นเอ็กสแตร์นที่ส่งในเวลาสำหรับการใช้งานเพื่อการศึกษานี้ เมื่ออนุญาตเห็นไปไซเบรประเยชนดานการค้ำไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารจะประสบความสำเร็จหรือไม่และหากมีข้อผิดพลาดประการใด ผู้ใช้ก็สามารถทราบได้จากข้อความที่แสดงออกมานอกเหนือจากบริการด้านต่าง ๆ ที่กล่าวข้างต้นแล้ว ยังมีบริการอื่น ๆ ที่ Application Layer จัดให้ดังนี้คือ

- การระบุระบบคอมพิวเตอร์ปลายทางโดยอาจจะใช้วิธีการเรียกตามที่อยู่ก็ได้
- กำหนดว่าเครื่องปลายทางที่ต้องการติดต่อนั้น อยู่ในสถานะที่พร้อมสำหรับการสื่อสาร
- กำหนดอำนาจหน้าที่ความสำคัญในการติดต่อสื่อสาร
- การร่วมตกลงในระบบการใส่รหัสเพื่อความปลอดภัยของข้อมูล
- การรับรองเครื่องคอมพิวเตอร์ที่ต้องการสื่อสารด้วยว่า อยู่ในสถานะที่ต้องการจะสื่อสารได้หรือไม่
- กำหนดและเลือกข้อมูลประเภทต่าง ๆ ที่จำเป็นต้องใช้เมื่อมีการสื่อสาร
- การร่วมตกลงเกี่ยวกับความรับผิดชอบ กรณีการกู้ข้อมูลที่เกิดความเสียหาย
- กำหนดข้อจำกัดต่าง ๆ เกี่ยวกับรูปแบบของข้อมูลที่จะใช้ส่ง เช่น รูปแบบตัวอักษรที่จะใช้หรือ โครงสร้างของข้อมูล

2.1.4.2 Presentation Layer

เป็นชั้นที่มีการกำหนดหน้าที่ไม่ชัดเจนนักและมีการนำไปใช้ไม่มาก หน้าที่หลัก คือ เป็นส่วนที่จัดรูปแบบและนำเสนอข้อมูลระหว่างการสื่อสารให้เป็นไปตามที่ต้องการ โดยมีการกำหนดรูปแบบการส่งข้อมูลสำหรับใช้ในการแลกเปลี่ยนรูปแบบที่จะมีการกำหนดไว้ใน 2 ลักษณะ ทั้งนี้ยังรวมไปถึงการจัดแปลงข้อมูลในรูปมาตรฐาน ASCII หรือ EBCDIC, การลดขนาดข้อมูล (Data Compression) การเข้ารหัสหรือถอดรหัสข้อมูลเพื่อความปลอดภัยในการสื่อสารแต่ส่วนใหญ่แล้วแอปพลิเคชันจะเป็นตัวจัดการแทนได้ตัวอย่างของ Presentation Layer ที่เข้าใจง่าย ๆ เช่น การพูดโทรศัพท์ที่ระหว่างบุคคลที่พูดภาษาฝรั่งเศสและบุคคลที่พูดภาษาสเปน บุคคลทั้งสองใช้ล่ามในการแปลภาษาและภาษากลางที่ใช้ก็คือ ภาษาอังกฤษ ล่ามแต่ละฝ่ายก็จะแปลภาษาทางฝ่ายของตนเองออกเป็นภาษาอังกฤษ แล้วฝ่ายที่รับก็จะแปลภาษาอังกฤษนั้นกลับเป็นภาษาของตนเองอีกต่อหนึ่ง บุคคลผู้พูดทางโทรศัพท์ก็เปรียบเสมือนกับโปรแกรม 2 โปรแกรมที่ต้องการสื่อสารกัน ตัวล่ามก็เปรียบเสมือนกับ Presentation Layer ในเครื่องคอมพิวเตอร์ทั้ง 2 ฝ่าย ภาษาฝรั่งเศสและภาษาสเปนเปรียบเสมือนกับ Abstract Data Syntax ส่วนภาษาอังกฤษเปรียบได้กับ Transfer หรือ Concrete Syntax ทั้งนี้ต้องนึกเสมอว่า ในการสื่อสารนั้นจะต้องมีการใช้ภาษากลางอันเป็นที่เข้าใจกันโดยทั่วไปภาษาหนึ่งเสมอ นอกจากนั้นตัวกลางในการแปลภาษา จากต้นทางและปลายทางให้เป็นภาษากลางก็ไม่จำเป็นจะต้องเข้าใจในความหมายของข้อมูลที่สื่อสารนั้นเสมอไปหน้าที่อีกประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่งของ Presentation Layer ก็คือเรื่องความปลอดภัยของข้อมูลโดยมีการเข้ารหัสข้อมูลก่อนส่งออกไป และเมื่ออีกฝ่ายหนึ่งได้รับข้อมูลแล้ว ก่อนนำมาใช้ก็จะมีการถอดรหัสตามรูปแบบตามที่ได้กำหนดไว้ล่วงหน้าทั้งสองฝ่ายแม้ว่าวิธีการเข้ารหัสข้อมูลนี้จะมีได้มีการกำหนดไว้ในมาตรฐานใด ๆ เลขก็ตามแต่ก็ได้มีผู้นำมาใช้กันอย่างแพร่หลาย

2.1.4.3 Session Layer

เป็นชั้นที่จัดการในเรื่องของการสร้างการติดต่อแต่ละครั้ง หรือ Session ให้ระบบคอมพิวเตอร์ทั้งสองฝ่ายกล่าวง่าย ๆ คือ จะให้บริการแก่โปรแกรมสื่อสารได้จัดการรวบรวมข้อมูลต่าง ๆ ที่จะใช้ในการติดต่อสื่อสาร โดยทำหน้าที่ตั้งแต่เริ่มการติดต่อจนถึงการส่งผ่านข้อมูลในการติดต่อนั้น ๆ เป็นไปได้โดยไม่มีปัญหาจนถึงการเลิกติดต่อเมื่อเสร็จงาน ซึ่ง Session Layer นี้จะรับผิดชอบเกี่ยวกับการจัดสรรช่องการสื่อสารระหว่าง Application Layer ทั้งสองฝ่ายให้สามารถสื่อสารกันได้จนเสร็จสมบูรณ์โดยต้องอาศัยความช่วยเหลือจาก Presentation Layer นอกจากนี้ยังให้บริการด้านต่าง ๆ คือ

- ให้บริการจัดเงื่อนไขได้ตอบข้อมูลซึ่งจะให้บริการทั้งในแบบส่งข้อมูลไม่พร้อมกัน (Half - Duplex) และแบบพร้อมกัน (Full - Duplex)
- การสื่อสารในทิศทางเดียวกันในการสื่อสารผ่านระบบเครือข่ายระยะไกล ถ้าหากเกิดข้อผิดพลาดขึ้นระหว่างการสื่อสารในจุดใดจุดหนึ่ง Session Layer จะถูกอนุญาตให้ผู้ใช้เลือกที่จะทำการรับ-ส่งข้อมูลใหม่อีกครั้งในเวลาใดก็ได้
- การรายงานเกี่ยวกับข้อผิดพลาดถ้าในระหว่างการสื่อสารเกิดข้อผิดพลาดที่ไม่สามารถแก้ไขได้ Session Layer จะทำการส่งสัญญาณแจ้งให้ Application Layer รู้ถึงข้อผิดพลาดนั้น

2.1.4.4 Transport Layer

ทำหน้าที่ควบคุมปริมาณและรายละเอียดวิธีการรับส่งข้อมูลให้เป็นไปตามกำหนดที่ได้ตั้งไว้ และจัดการให้การเชื่อมโยงเครือข่ายเป็นไปอย่างราบรื่น Transport Layer เป็นชั้นสุดท้ายที่จัดการเรื่องของการเดินทางในการส่งข้อมูล และจัดการตรวจสอบความผิดพลาดของข้อมูล ซึ่งส่วนของ TCP (Transmission Control Protocol) ใน TCP/IP แบบที่ใช้ในอินเทอร์เน็ตจะทำงานที่ชั้นนี้ Transport Layer จะทำหน้าที่เป็นตัวเชื่อมระหว่าง 2 Layer คือ Application – Oriented Layer ซึ่งอยู่เหนือกว่า กับ Network Dependent Protocol Layer ซึ่งอยู่ต่ำกว่าและมีหน้าที่ในการเตรียมข้อความต่าง ๆ ในการสื่อสารแบบ Session Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.5 Network Layer

ทำหน้าที่ควบคุมวิธีการส่งผ่านข้อมูลระหว่างเครือข่ายให้ถูกต้อง และเป็นไปตามเส้นทางที่กำหนดทำหน้าที่ในการเชื่อมต่อเส้นทางการสื่อสารระหว่าง Transport Layer ของเครือข่ายต้นทางและปลายทาง โดยจะจัดคำสั่งการทำงานเกี่ยวกับการค้นหาที่หมายปลายทางและความคลุมเครือของข้อมูลในการเชื่อมต่อระหว่างคอมพิวเตอร์กับเครือข่ายและในกรณีติดต่อระหว่างเครือข่าย Network Layer ก็จะมีการจัดเตรียมคำสั่งการทำงานเพื่อให้การติดต่อเป็นไปอย่างราบรื่นสมบูรณ์ ในปัจจุบันเมื่อมีการใช้เครือข่ายมากขึ้นขนาดใหญ่มากขึ้นจะมีการจัดการแบ่งเครือข่ายนั้นให้เป็นส่วนย่อย หรือ Network Segment หรือเรียกว่า Subnetwork โดยใช้อุปกรณ์ Bridge หรือ Router ทั้งนี้ Network Layer จะจัดส่งผ่าน Packet ข้อมูล คือข้อมูลที่ถูกรวบรวมให้อยู่ในรูปแบบที่กำหนดไว้แล้วนั่นเองผ่านอุปกรณ์ต่าง ๆ ไปยังเครือข่ายย่อยให้อย่างถูกต้องตามที่ต้องการ นอกจากนี้ Network Layer จะจัดการดูแลเส้นทางในการส่งข้อมูล (Routing Table) และค้นหาหรือกรอง Packet ข้อมูลที่ส่งไปยังที่หมายในเครือข่ายย่อยเดียวกันไม่ให้ข้ามไปยังเครือข่ายย่อยอื่นซึ่งจะช่วยลดปริมาณข้อมูลที่วิ่งบนเครือข่ายได้ส่วนหนึ่ง

2.1.4.6 Data Link Layer

จะจัดเตรียมข้อมูลเกี่ยวกับการเชื่อมต่อให้กับ Network Layer และจะทำหน้าที่เรียกใช้หรือกำหนดช่องทางในการส่งข้อมูลที่ถูกต้อง เช่น Ethernet, Token Ring หรือ FDDI (Fiber Distributed Data Interface) เป็นต้น รวมถึงการควบคุมลำดับและอัตราการรับส่งข้อมูลและสถานที่ ที่จะส่งข้อมูลไป ทั้งนี้ Data Link Layer เป็นชั้นแรกที่มีการแปลงข้อมูลจากบิตให้อยู่ในรูปของ Packet โดยจะมีการเพิ่มข้อมูลเพื่อตรวจสอบความถูกต้องในกรณีส่งข้อมูลออกไป หรือในกรณีอ่านข้อมูลเข้ามาก็จะตรวจสอบส่วน Checksum เพื่อความข้อมูลที่รับมาถูกต้องครบถ้วนและถ้าได้รับ Packet ข้อมูลที่ไม่ถูกต้องก็จะไม่เอาข้อมูลนั้นไปใช้งานต่อและบอกไปยังต้นทางให้ส่งมาใหม่ ตัวอย่างเช่น คำสั่งในการตรวจสอบความผิดพลาดหรือมีข้อผิดพลาดแล้วต้องการส่งข้อมูลใหม่ก็จะมีคำสั่งในการส่งข้อมูลซ้ำ โดยปกติแล้วจะมีการให้บริการใน 2 รูปแบบ ได้แก่

1. Connection Less ถ้าหากมีการตรวจสอบและพบข้อผิดพลาดก็จะตัดการสื่อสารทันที
2. Connection Oriented ให้บริการในลักษณะพยายามไม่ให้เกิดข้อผิดพลาดใด ๆ ทั้งสิ้น

2.1.4.7 Physical Layer

จะให้บริการเกี่ยวกับวิธีการส่งข้อมูลเป็นบิตระหว่างอุปกรณ์ 2 ชนิด ได้แก่ เครื่องมือและอุปกรณ์ทางฝ่ายผู้รับและผู้ส่งและอุปกรณ์ของเครือข่ายโดยข้อมูลต่าง ๆ จะถูกส่งให้กับ Data Link Layer นอกจากนั้นยังรับผิดชอบดูแลในรายละเอียดการส่งข้อมูลในฮาร์ดแวร์จริง เช่น การควบคุม Network Interface Card การส่งสัญญาณผ่าน สายสัญญาณแบบต่าง ๆ การเชื่อมต่อเข้าเครือข่ายแบบต่าง ๆ โดย Physical Layer

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะจัดสัญญาณทางไฟฟ้าสัญญาณเสียงหรือสัญญาณแสงที่จำเป็นในการสื่อสาร โดยตรงการส่งถ่ายข้อมูลระหว่างชั้นโปรโตคอลในแต่ละชั้นล้วนมีหน้าที่เกี่ยวข้องในการส่งผ่านข้อมูลจากสถานีต้นทาง ไปยังสถานีปลายทางข้อมูลจะถูกส่งผ่านจากโปรโตคอลระดับบนสุด จากสถานีต้นทางไปยังระดับล่างจนกระทั่งข้อมูลถูกแปลงให้อยู่ในรูปของสัญญาณไฟฟ้า แล้วเดินทางผ่านสถานีเครือข่ายไปยังสถานีปลายทาง โปรโตคอลระดับล่างสุดที่สถานีปลายทางจะรับสัญญาณและส่งผ่านขึ้นไปยังโปรโตคอลระดับบนต่อไปเมื่อข้อมูลผ่านแต่ละระดับชั้นโปรโตคอล ในชั้นนั้น ๆ จะผนวกข่าวสารกำกับการทำงานประจำโปรโตคอลซึ่งเรียกว่า โปรโตคอลเฮดเดอร์เข้ากับข้อมูลเฮดเดอร์และข้อมูลจากระดับบนจะถูกส่งผ่านไปยังระดับล่าง โปรโตคอลระดับล่างจะมองเฮดเดอร์และตัวข้อมูลรวมเป็นเสมือนข้อมูลและเพิ่มเฮดเดอร์ประจำชั้นเข้าไปข้อมูลเดิมจึงมีแค่เฮดเดอร์หุ้มเป็นชั้น ๆ กระบวนการนี้เรียกว่า การเอ็นแคปซูลเลตในอินเทอร์เน็ตเมื่อสถานีปลายทางได้รับแพ็คเกจก็จะดำเนินการส่งไปตามลำดับชั้น โปรโตคอลประจำชั้น จะทำการถอดเฮดเดอร์ออกและส่งส่วนที่เหลือไปยังชั้นถัดไป

2.1.5 เลขหมายอินเทอร์เน็ต (IP Address)

IP address ถูกกำหนดขึ้นมาให้เป็นหมายเลขอ้างอิงประจำตัวของอุปกรณ์ต่าง ๆ ที่เชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ตโดยการกำหนด IP address ให้แต่ละเครื่องหรือแต่ละอุปกรณ์นี้จะต้องไม่ซ้ำกัน ซึ่ง IP address นี้จะไม่ถูกผูกติดกับฮาร์ดแวร์แต่อย่างใดจึงสามารถกำหนดใหม่หรือแก้ไขเปลี่ยนแปลงได้เมื่อมีการเปลี่ยนตัวฮาร์ดแวร์การทำงานของโปรโตคอล IP จำเป็นต้องอาศัย IP address เพื่อระบุและอ้างถึงอุปกรณ์ต่าง ๆ ที่ต่ออยู่ในเครือข่ายไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์ เมล์เซิร์ฟเวอร์ อุปกรณ์ Router ฯลฯ IP address จะเป็นค่าตัวเลขขนาด 32 บิต ถูกแบ่งออกเป็นส่วนละ 8 บิต รวมเป็น 4 ส่วนและคั่นละส่วนด้วยเครื่องหมายจุด (.) ดังนั้นค่าตัวเลขในแต่ละส่วนจะมีได้ตั้งแต่ 0 ถึง 255 (2^8) เช่น ที่ทำหน้าที่เป็น Gateway อาจมี LAN card อยู่ 2 ถึง 3 card หมายเลข IP มักนิยมเขียนในรูปเลขฐาน 10 สี่กลุ่มโดยใช้จุดคั่นแต่ละช่วงมีค่าตั้งแต่ 0 ถึง 255 เช่น 161.246.18.44 เป็นต้นแต่เวลาเครื่องคอมพิวเตอร์นำเลขเหล่านี้ไปประมวลมันจะมองเป็นเลขฐานสอง 32 บิต ต่อเนื่องกัน เลข IP นี้แบ่งออกเป็นสองส่วนคือ ส่วนที่ชี้เน็ตเวิร์ก กับส่วนที่ชี้เครื่อง แต่ละส่วนจะใช้กบิตนั้นขึ้นอยู่กับ Class ของเลข IP นั้น ๆ ซึ่ง Class ของเลข IP นั้นจะมีอยู่ 5 Class ด้วยกันคือ

1. Class A : ช่วงตั้งแต่ 0.0.0.0 ถึง 127.255.255.255
2. Class B : ช่วงตั้งแต่ 128.0.0.0 ถึง 191.255.255.255
3. Class C : ช่วงตั้งแต่ 192.0.0.0 ถึง 223.255.255.255
4. Class D : ช่วงตั้งแต่ 224.0.0.0 ถึง 239.255.255.255
5. Class E : ช่วงตั้งแต่ 240.0.0.0 ถึง 247.255.255.255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเลขหมายอินเทอร์เน็ตมีจำกัดจึงจำเป็นต้องมีการจำกัดบางหมายเลขเพื่อใช้ในการภายในเพื่อให้ผู้ที่ต้องการใช้เลขหมายอินเทอร์เน็ตที่ไม่ต้องการติดต่อกับเครือข่ายภายนอกได้ใช้ ได้แก่

Class A: ช่วงตั้งแต่ 0.0.0.0 ถึง 10.255.255.255

Class B: ช่วงตั้งแต่ 172.16.0.0 ถึง 127.31.255.255

Class C: ช่วงตั้งแต่ 192.168.0.0 ถึง 192.168.255.255

2.1.5.1 IPv4 (Internet Protocol Address Version 4)

IP address ทุกค่าจะมีรูปแบบที่เขียนให้เข้าใจเหมือนกันคือ เป็นตัวเลข 4 ชุดคั่นด้วยจุด เพื่อให้อ่านและจดจำได้ง่าย IP address มีขนาด 32 บิต เช่น 204.283.255.20 เป็นต้น ในปัจจุบันเรียกรูปแบบดังกล่าวนี้ว่า IP version 4 หรือ IPv4 ซึ่งมีการใช้งานกันทั่วไปและถูกใช้หมดไปเรื่อย ๆ จนอาจจะไม่เพียงพอสำหรับใช้งานในอนาคตอันใกล้นี้เนื่องจาก IPv4 มีขนาดเพียง 32 บิต ทำให้การกำหนดค่า IP address ทำได้ไม่มากพอกับอุปกรณ์และเครือข่ายใหม่ ๆ ที่เชื่อมต่อเข้ามาในเครือข่ายอินเทอร์เน็ต

2.1.5.2 IPv6 (Internet Protocol Address Version 6)

IP address V6 แบบ unicast คือ การกำหนดเลขหมาย IP address ให้กับเครื่องเซิร์ฟเวอร์ที่ให้บริการหรือกำหนดให้อุปกรณ์ที่เชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ตเหมือนปกติแต่แบ่งเป็น 3 ลักษณะ

- Provider-based address ใช้กำหนดให้กับผู้ให้บริการอินเทอร์เน็ตหรือบริษัท ISP ซึ่งข้อมูลของ IP address จะมีส่วนของ Provider ID และ Subscriber ID อยู่ด้วยเพื่อบอกให้ทราบว่าเป็นของ ISP ไດ
- Site – Local – use address เป็นการกำหนด IP Address V6 ให้ใช้งานกับองค์กรที่ต้องการใช้โปรโตคอล TCP/IP แต่ไม่ได้เชื่อมต่อเข้ากับเครือข่ายอินเทอร์เน็ตเรียกว่าการใช้งานแบบอินทราเน็ตในเบื้องต้นและต้องการเชื่อมเข้าอินเทอร์เน็ตได้ในอนาคต
- Link – Local – use address ใช้กำหนดให้กับเครือข่ายที่เชื่อมต่อในอินเทอร์เน็ตตลอดเวลาขององค์กรหรือบริษัทใดๆ

2.1.6 การแบ่ง Subnet

เราสามารถแบ่งหนึ่งเน็ตเวิร์คของเราออกเป็นโครงข่ายย่อย ๆ โดยเรามักจะทำเมื่อ IP ของเราสามารถอ้างอิงเครื่องภายในหนึ่งเน็ตเวิร์คได้มากกว่าความต้องการจริง ๆ ของเรามาก ๆ วิธีการแบ่ง Subnet นั้นเราจะอาศัยความรู้เรื่องของเลขอินเทอร์เน็ตที่ว่าเลข IP แบ่งออกเป็นสองส่วนคือ ส่วนซึ่งเน็ตเวิร์คกับส่วนซึ่งเครื่องในเมื่อจำนวนบิตที่เราใช้ในการชี้เครื่องมากเกินไป เราก็ทำการลดจำนวนโดยนำไปเพิ่มให้กับส่วนซึ่งเน็ตเวิร์คทั้งหมดที่กล่าวมาจะสามารถทำได้โดยการใช้เน็ตเวิร์คซึ่งเป็นเลข 32 บิตเหมือนเลขหมายเลขอินเทอร์เน็ตมีหน้าทีระบุขอบเขตของเลข IP ว่าบิตใดใช้ชี้เน็ตเวิร์คและบิตใดใช้ชี้เครื่องเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติเน็ตเวิร์ค Class A,B,C จะมีเนตมาร์ค ที่ตายตัวอยู่แล้วคือ 255.0.0.0 , 255.255.0.0 และ 255.255.255.0 ตามลำดับวิธีการใช้เนตมาร์คนั้นเราจำนำมา And กับ IP Address ผลที่ได้จากการ And คือ หมายเลขเน็ตเวิร์ค เช่น 161.246.18.44 จะมีเนตมาร์คมาตรฐานเป็น 255.255.0.0 (เพราะเป็น IP Class B) เมื่อนำมา And กันจะได้ 161.246.0.0 ซึ่งก็คือหมายเลขเน็ตเวิร์คนั่นเองบิตที่ถูกตัดทิ้งไปก็คือหมายเลขเครื่องเราสามารถทำการเปลี่ยนแปลงเนตมาร์คมาตรฐานได้โดยการเพิ่มจำนวนบิตที่ใช้เน็ตเวิร์คนั้น คือเราจะเพิ่มจำนวนบิตที่เป็น 1 ของ เนตมาร์ค เช่น เราอาจเพิ่มบิตที่ใช้ในการชี้เน็ตเวิร์คของ IP Class B ขึ้นอีก 8 บิต โดยดึงมาจากบิตที่เคยใช้ในการชี้เครื่อง ในกรณีนี้เราจะได้เนตมาร์คเป็น 255.255.255.0 ทำให้เมื่อนำไป And กับ IP 161.246.18.44 จะได้ 161.246.12.18.0 เป็นหมายเลขเน็ตเวิร์คจะเห็นว่าจะได้เน็ตเวิร์คเพิ่มขึ้นมาเป็น 256 เน็ตเวิร์ค และแต่ละ เน็ตเวิร์คมีหมายเลขเครื่อง 254 เครื่อง

2.1.7 โครงสร้างของโปรโตคอล TCP/IP

โปรโตคอล TCP/IP (Transmission Control Protocol / Internet Protocol) มีการจัดกลไกการทำงานเป็นชั้นหรือ Layer เรียงต่อกัน โดยในแต่ละ Layer จะมีการทำงานเทียบได้กับ OSI model มาตรฐานแต่บาง Layer ปนกันซึ่งในแต่ละ Layer ของโปรโตคอล TCP/IP จะประกอบด้วย

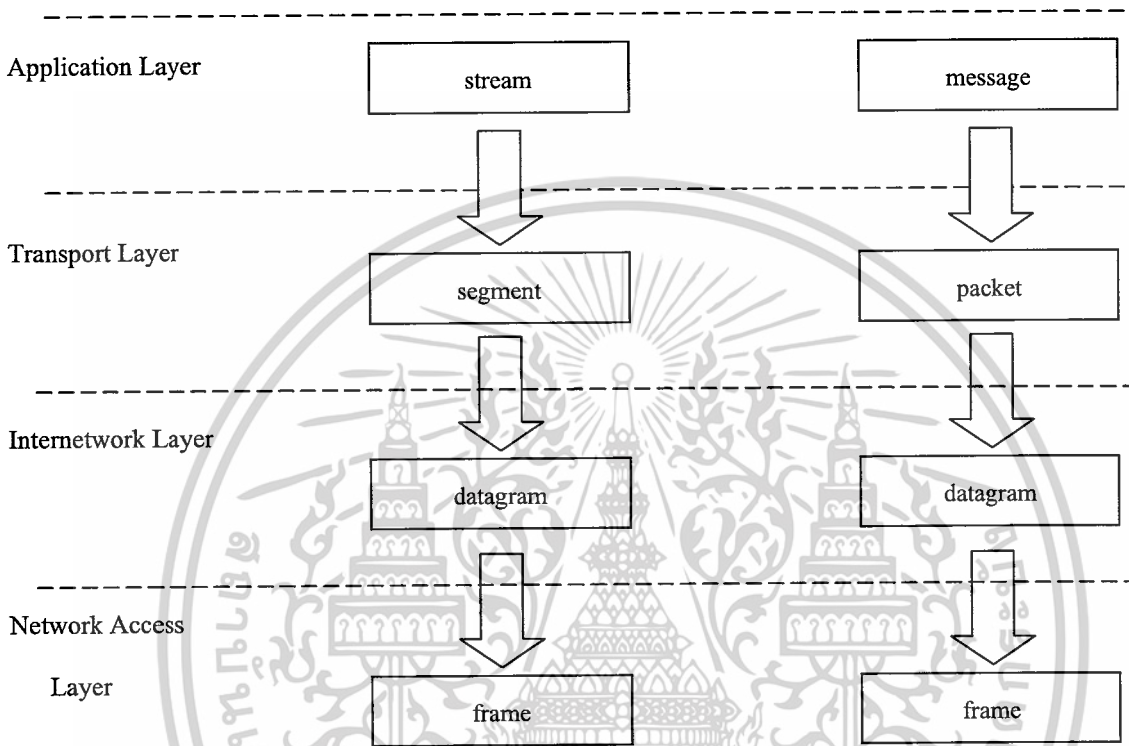
- Process Layer หรือ Application Layer
- Host-to-Host Layer หรือ Transport Layer
- Internetwork Layer
- Network Interface Layer

โดยเมื่อเทียบกับมาตรฐาน OSI model ดังรูปที่ 2.2 ซึ่งเราจะเห็นว่าบางกลไกของโปรโตคอล TCP/IP เทียบได้กับมาตรฐาน OSI model สองชั้นหรือบางกลไกก็จะทำงานคาบเกี่ยวกันระหว่างบางชั้นของ OSI model ตัวอย่างเช่น กลไกการทำงานของโปรโตคอล TCP/IP ในส่วน Network Interface Layer เมื่อเทียบกับมาตรฐาน OSI model จะเทียบได้กับ Data Link Layer และ Physical Layer 2 ชั้นรวมกัน เป็นต้น ในแต่ละกลไกของโปรโตคอล TCP/IP และโปรโตคอลอื่น ๆ ในชุดของ TCP/IP ร่วมทำงานอยู่ด้วยกันสำหรับการศึกษาโปรโตคอล TCP/IP นั้นเราจะไม่อ้างอิง OSI Reference Model นี้เพราะจะเข้าใจได้ยาก ดังนั้นเราจึงจะสร้าง Model ขึ้นมาใหม่ ลักษณะการทำงานของ model ในลักษณะนี้คือ ข้อมูลจะถูกส่งลงมาจากชั้นข้างบนลงมายังชั้นข้างล่างสุดซึ่งมีหน้าที่จัดการเกี่ยวกับการส่งข้อมูลผ่านสายสัญญาณ ไปยังจุดหมายปลายทาง เมื่อข้อมูลไปถึงจุดหมายแล้วก็จะย้อนกลับจากชั้นล่างขึ้นไปชั้นบนสุด ซึ่งเป็นชั้นที่โปรแกรมใช้งานต่าง ๆ ทำงานอยู่ขณะที่ข้อมูลถูกส่งผ่านจากชั้นบนลงมายังชั้นล่างแต่ละชั้นจะทำการเพิ่มข้อมูลควบคุมเข้าไปเพื่อให้การส่งข้อมูลถูกต้อง และเป็นการส่งพารามิเตอร์ที่จำเป็น ไปให้กับชั้นของมัน

ในเครื่องปลายทางข้อมูลควบคุมเหล่านี้เราเรียกว่า Header ที่มีรูปแบบเป็นของตัวเองการเพิ่ม Header เข้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปกับข้อมูลนั้นเราเรียกว่า Data Encapsulation หน่วยของข้อมูลที่จะทำการส่งนั้นจะมีชื่อเรียกต่างกันเมื่อมันเดินทางผ่านแต่ละ Layer ซึ่งจะเห็นว่า การส่งใน TCP/IP นั้นมีอยู่ 2 โพรโทคอล ย่อย คือ TCP กับ UDP ชื่อของข้อมูลที่ผ่าน โพรโทคอลทั้งสองนั้นแตกต่างกันในชั้นบน ๆ เท่านั้นชื่อสำคัญ ๆ ที่เราจะต้องพบและใช้ต่อไปบ่อย ๆ ได้แก่ datagram และ frame



รูปที่ 2.2 Data Structures

2.1.8 Process Layer หรือ Application Layer

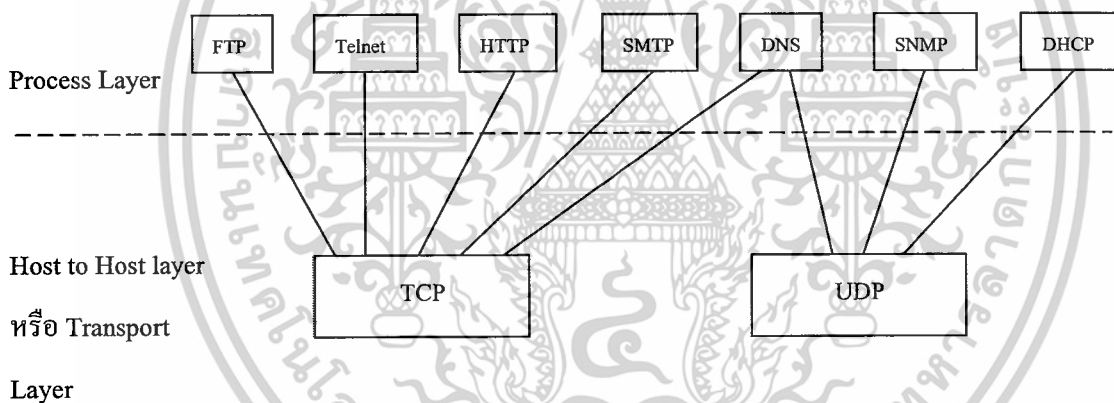
การแสดงลำดับชั้นการทำงานของโปรโตคอล TCP/IP เทียบกับมาตรฐาน OSI model นั้น ในชั้นบนสุดเรียกว่า Process Layer ทำงาน 2 หน้าที่เทียบได้กับ Application Layer และ Presentation Layer ในชั้นนี้จะรองรับการทำงานของแอปพลิเคชันต่างๆ ที่ทำงานเป็นโปรเซสอยู่ในเครื่องเซิร์ฟเวอร์ที่ให้บริการและเครื่องที่ขอใช้บริการ หรือ client ซึ่งจะติดต่อกันผ่านโปรโตคอลเฉพาะแอปพลิเคชันอีกทีหนึ่ง ตัวอย่างเช่น เมื่อผู้ใช้งานอินเทอร์เน็ตต้องการโอนถ่ายไฟล์หรือ download ข้อมูลจากเครื่องเซิร์ฟเวอร์ที่ให้บริการโดยอาจจะเรียกใช้โปรแกรม ftp client ทั่วไป เช่น โปรแกรม WS ftp ติดต่อกับโปรเซส ftp ที่กำลังให้บริการอยู่ที่เครื่องเซิร์ฟเวอร์ จากนั้นตัวโปรเซส ftp ก็จะเรียกใช้โปรโตคอล FTP (File Transfer Protocol) เพื่อทำการโอนถ่ายไฟล์นี้หรือถ้าผู้ใช้ต้องการเรียกใช้งานคอมพิวเตอร์ที่อยู่ห่างไกลออกไปด้วยการใช้โปรแกรม telnet ที่ทำงานอยู่ก็จะเรียกใช้โปรโตคอล Telnet เพื่อติดต่อกันหรือในกรณีที่มีการเรียกใช้โปรแกรม web browser เช่น Netscape Navigator เพื่อเรียกดูเว็บไซต์ CNN ที่เครื่องซึ่งให้บริการ

เอ็กสทราเนตเป็นเอ็กสทราเนตที่ส่งมาเพื่อการศึกษาเท่านั้น เมื่อนูญเตเห็นาเบเซบระเอชชานการค้ำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บของ CNN ก็จะมีโปรเซส HTTP (HyperText Transfer Protocol) ทำงานอยู่และจะติดต่อกับผู้ใช้ผ่านโปรโตคอล HTTP เป็นต้น โปรโตคอลหลัก ๆ ที่ทำงานใน Process Layer ซึ่งผู้ใช้อาจจะคุ้นเคยกันได้แก่ FTP (File Transfer Protocol), Telnet, HTTP (HyperText Transfer Protocol) และ SMTP (Simple Mail Transfer Protocol)

2.1.9 Host – To – Host Layer หรือ Transport Layer

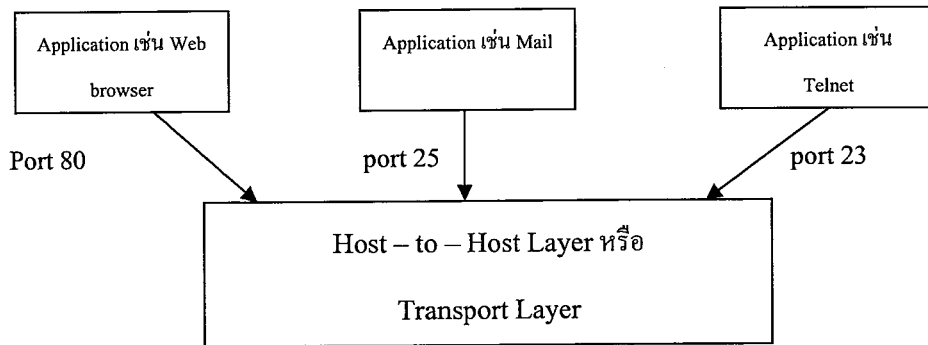
การทำงานที่ชั้นของ Host – To – Host Layer นี้จะมีบทบาทในการจัดการต่อจาก Process Layer บางครั้งเรามักเรียกชั้น Host – To – Host Layer ว่าเป็น Transport Layer ซึ่งไม่ใช่ชั้นของ Transport Layer ในมาตรฐาน OSI model การทำงานของ Host – To – Host Layer นี้จะมีการสร้าง connection หรือการเชื่อมต่อกันระหว่างแอปพลิเคชันกับ Host – To – Host Layer โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลที่เราเรียกว่า port หรือ socket (คำว่า port ในที่นี้ไม่ได้หมายถึง port ทางฮาร์ดแวร์) และในแต่ละแอปพลิเคชันที่อยู่ในชั้น Process Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้และแต่ละโปรโตคอลจะมีการใช้งาน port หมายเลขต่าง ๆ ไม่ซ้ำกัน



รูปที่ 2.3 แสดงการใช้งาน port ของแต่ละโปรโตคอล

เมื่อแอปพลิเคชันทำงานผ่านโปรโตคอลในชั้น Process Layer จะมีการส่งผ่านข้อมูลไปยัง Host – To – Host Layer ที่ชั้นนี้จะมีการเชื่อมต่อผ่าน port ที่กำหนดทำให้การรับส่งข้อมูลในแต่ละโปรโตคอลทำได้ถูกต้อง ถึงแม้ว่าในเครื่องเซิร์ฟเวอร์ที่ให้บริการจะมีการทำงานอยู่หลายโปรเซสที่แตกต่างกันก็ตาม หรือมีผู้ใช้บริการเข้ามาใช้งานพร้อมกันจำนวนมากและหลายแอปพลิเคชันในเวลาเดียวกันในชั้น Host – To – Host Layer หรือ Transport Layer ของ TCP/IP นี้ จะมีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกัน คือ โปรโตคอล TCP และโปรโตคอล UDP (User Datagram Protocol) ในการส่งผ่านข้อมูลลงไปชั้นถัดไปเราจะเห็นว่าโปรโตคอล TCP และ UDP จะถูกผนึกเข้าไปในโปรโตคอล IP อีกทีหนึ่งและส่งต่อไปยังเครือข่ายอินเทอร์เน็ตต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงการส่งข้อมูลจาก Application ไปยัง Host - To - Host Layer

2.1.9.1 โพรโทคอล TCP

โพรโทคอล TCP (Transmission Control Protocol) เป็นโพรโทคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไปแต่ละแบ่งข้อมูลเป็นส่วนย่อย ๆ ก่อนแล้วจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูลในกรณีที่มีข้อมูลส่วนใดส่วนหนึ่งสูญหายไปก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูล datagram ดังนั้นแอปพลิเคชันหรือโปรเซสใดที่อาศัยการส่งผ่านข้อมูลด้วยโพรโทคอล TCP จะต้องใช้หน่วยความจำและขนาดของช่องสัญญาณ (bandwidth) มากกว่า UDP

การติดต่อระหว่างกันจะต้องเป็นแบบ connection - oriented คือต้องมีการสร้างการติดต่อกันเป็น session ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (full duplex) เหมือนกับการใช้โทรศัพท์ติดต่อกัน เมื่อผู้ติดต่อต้นทางเรียกให้ฝ่ายตรงข้ามรับสายแล้วจึงเริ่มการสนทนา เช่น พูดคำว่า “ สวัสดี” หรือ “ ฮัลโหล” กันก่อนเพื่อให้แน่ใจว่าฝ่ายตรงข้ามพร้อมจะติดต่อด้วยจากนั้นจึงเริ่มต้นติดต่อกันและเมื่อต้องการจะเลิกการติดต่อก็จะมีการพูดคำว่า “สวัสดี” ให้ฝ่ายตรงข้ามทราบว่าจะเลิกการติดต่อและวางสายไปซึ่งในการเชื่อมโยงระหว่างทั้งสองด้านยังคงมีอยู่ไม่ขาดไปจนกว่าฝ่ายใดฝ่ายหนึ่งจะวางสาย เช่นเดียวกับการติดต่อกันด้วยกลไกโพรโทคอล TCP เมื่อแอปพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โพรโทคอลที่เหมาะสมในชั้น Process Layer ติดต่อกันและมีการสร้างช่องส่งข้อมูลผ่านพอร์ตที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโพรโทคอล TCP

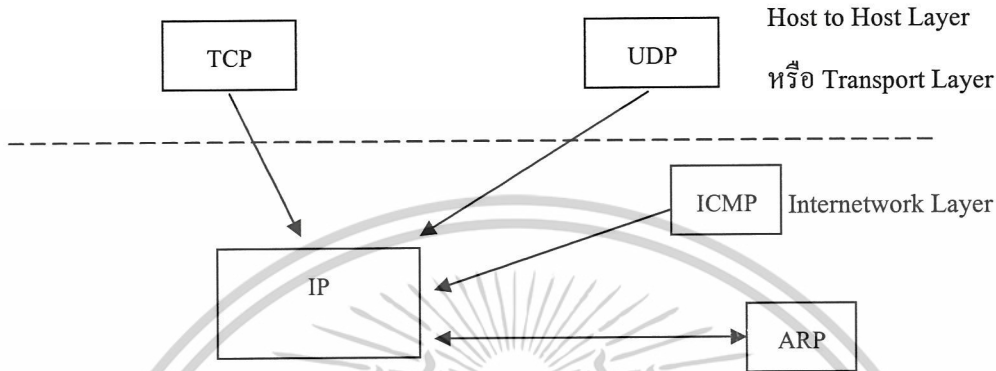
2.1.9.2 โพรโทคอล UDP

ใน Host - To - Host Layer นอกจากจะมีโพรโทคอล TCP ทำงานแล้วก็มีโพรโทคอล UDP (User Datagram Protocol) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอปพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโพรโทคอล UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่าง

เครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มี การส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอปพลิเคชันหรือ โปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูล ขึ้นมาเอง



รูปที่ 2.5 แสดงกลไกการส่งข้อมูลด้วยโปรโตคอล UDP

2.1.9.3 UDP Header

ตารางที่ 2.2 UDP Header

| | |
|--------------------|-------------------------|
| 16-bit Source port | 16-bit Destination port |
| Length | Checksum |
| Data | |

ตารางที่ 2.3 รายละเอียด UDP Header

| ตำแหน่ง | ชื่อ | อธิบาย |
|-----------|-------------------------|--|
| บิต 0-15 | Source port number | หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้ มีความยาว 16 บิต |
| บิต 16-31 | Destination port number | หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม มีความยาว 16 บิตเช่นกัน |
| บิต 32-47 | UDP length | ความยาวของดาต้าแกรม ทั้งส่วน Header และ Data นั้น หมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header |
| บิต 48-63 | Checksum | เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9.4 UDP Checksum

Checksum เป็นเลข 16 บิตถูกคำนวณด้วยวิธี one's complement โดยนำ Pseudo Header และข้อมูลทั้งหมดใน UDP Datagram มาคำนวณ Pseudo Header เป็นข้อมูลที่อยู่ในส่วนของ IP Header ประกอบด้วยฟิลด์ source IP address , destination IP address, zero , protocol, UDP length

ตารางที่ 2.4 Pseudo Header

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

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

2.1.10 Internetwork Layer

ในระดับล่างต่อมาในชั้น Internetwork Layer มีหน้าที่ส่งผ่านข้อมูลในระหว่างเครือข่าย โดยมีโปรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ บนอินเทอร์เน็ต คือ โปรโตคอล IP (Internet Protocol) นอกจากนี้ในชั้น Internetwork Layer ยังมีโปรโตคอลทำงานอยู่ด้วยอีก 2 ชนิดคือ โปรโตคอล Internet Control Message Protocol (ICMP) และโปรโตคอล Address Resolution Protocol (ARP) แต่ที่ใช้ในการทำโครงงานนี้คือ โปรโตคอล IP

2.1.10.1 IP Datagram

IP Datagram ประกอบด้วย IP Header และ IP Payload

IP Header เป็นขนาดที่เปลี่ยนแปลงได้ระหว่าง 20 และ 60 ไบต์ ในการเพิ่มขึ้น 4 ไบต์ มันจะจัดการเตรียมการสนับสนุน routing การแสดงตัว payload การชี้ให้เห็นถึงขนาด IP Header และ Datagram การสนับสนุน fragmentation

ตารางที่ 2.5 แสดง โครงสร้าง IP Header

| | | | | | | |
|-----------------------------|------------------------------|------------------------------|--------------------------------|-------------------------------------|------------------------------------|------------------------------|
| Version 4 bit | IP Header Length 4 bit | Type of Service 8 bit | Total Length 16 bit | Identifier 16 bit | Flags 3 bit | Fragment Offset 13 bit |
| Time – to- Live 8 bit | Protocol 8 bit | Header Checksum 16 bit | Source IP Address 32 bit | Destination IP Address 32 bit | IP Option And Padding 32 bit | |

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

2.1.10.2 โพรโทคอล IP

โพรโทคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host-Layer เพื่อส่งข้ามไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง แม้ว่าจะมีเครือข่ายเชื่อมต่อกันอยู่ในอินเทอร์เน็ตเป็นล้าน ๆ เครือข่ายก็ตาม เนื่องจากโพรโทคอล IP มีข้อมูลตำแหน่ง IP ปลายทางที่จะส่งข้อมูลไปให้โดยทำงานร่วมกับอุปกรณ์ router เพื่อส่งข้อมูลข้ามเครือข่ายออกไปได้ตัวโพรโทคอล IP จะทำงานแบบ packet switching คือ มีการส่งข้อมูลผ่านสวิตช์ (switch) ไปยังปลายทางโดยข้อมูลจะเดินทางไปยังเครือข่ายต่าง ๆ ผ่านสวิตช์นี้ไปเรื่อย ๆ จนกว่าจะถึงปลายทางตัวจริงผ่านหรือ switch นี้อาจเป็น gateway หรือ router ในระบบเครือข่ายก็ได้ซึ่งในข้อมูลของโพรโทคอล IP ที่จะส่งข้อมูลไปและเมื่อถึงเครือข่ายปลายทางแล้วจะมีกลไกแปลงหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์ประจำเครื่องที่ถูกต้องอีกทีหนึ่งด้วยโพรโทคอล ARP

โพรโทคอล Field มีขนาดยาว 1 ไบต์ และใช้เป็นตัวบ่งบอกในการบรรจุข้อมูลไปยัง Layer ที่สูงขึ้นกับใน IP Payload และยังเป็นตัวบ่งบอกลูกข่ายโพรโทคอลที่ชัดเจนโดยปกติค่าของ IP Protocol Field จะเป็น 1 สำหรับ ICMP 6 สำหรับ TCP และ 17 (0x11) สำหรับ UDP โพรโทคอล field จะแสดงตัวได้อย่างมากมาย ดังนั้น payload สามารถที่จะผ่านไปยัง Layer ที่สูงขึ้นไปได้ถูกต้องโดยจะได้รับที่ Destination

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 ค่าต่าง ๆ ของ IP Protocol Field

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

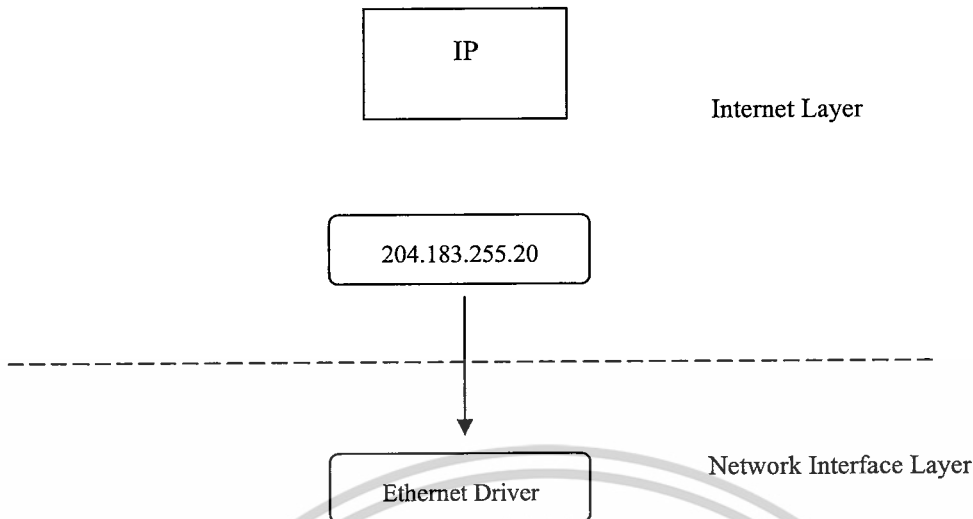
2.1.10.3 การกำหนด IP address ให้กับอุปกรณ์

ต้องกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเข้ากับเครือข่ายทุกจุดเชื่อมต่อหรือ Interface อาจหมายถึง Network Interface card (Lan การ์ด) ที่ติดตั้งในเซิร์ฟเวอร์หรือ LAN port, Ethernet port ที่ Router ใช้เชื่อมต่อเข้ากับเครือข่ายเป็นต้น การกำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อนี้ทำให้เราเข้าใจได้ว่าในบางอุปกรณ์ที่มีจุดเชื่อมต่อเข้ากับเครือข่ายมากกว่าหนึ่งจุดต้องกำหนดหมายเลข IP address ให้ครบ

2.1.10.4 การ Bind IP address

เมื่อได้กำหนดหมายเลข IP address ให้กับจุดเชื่อมต่อเช่น Lan card แล้วที่เครื่องเซิร์ฟเวอร์จะต้องมีการ bind หรือผนวกค่า IP address ดังกล่าวเข้ากับ Ethernet driver เพื่ออ้างอิงหมายเลข IP กับฮาร์ดแวร์ ให้ทำหน้าที่ติดต่อส่งข้อมูลในระดับ Network Interface ได้ต่อไป ดังตัวอย่างในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการ Bind IP address

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

2.1.10.5 โปรโตคอล ARP

โปรโตคอล ARP (Address Resolution Protocol) ถูกเรียกใช้งานโดยโปรโตคอล IP เพื่อช่วยแปลงหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น เว็บเซิร์ฟเวอร์เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ตและในการเชื่อมต่อนี้ต้องอาศัย Network Interface Card (NIC) หรือ LAN card ติดตั้งอยู่ที่ LAN card นี้เองจะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานในโปรโตคอล TCP/IP ก็จะต้องมีการกำหนดหมายเลข IP address ประจำตัวเพื่อใช้อ้างอิงกันและโปรโตคอล ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงให้ในระดับการทำงานที่ Internetwork Layer นี้ ซึ่งกลไกการแปลงนี้เรียกว่า address resolution

2.1.10.6 โปรโตคอล ARP ย้อนกลับ RARP (Reverse Address Resolution Protocol)

วิธีการ ARP ช่วยแก้ปัญหาในการค้นหาที่อยู่ของข้อมูลที่ให้การกำหนดที่อยู่ฮาร์ดแวร์ แบบ IP แต่ถ้าทราบว่าที่อยู่แบบฮาร์ดแวร์แล้วต้องการแปลงที่อยู่เป็น IP จะทำอย่างไร ปัญหานี้มักเกิดขึ้นกับเครื่องคอมพิวเตอร์ที่เริ่มทำงานด้วยการอ่านข้อมูลทั้งหมดจากเครื่อง Host เครื่องประเภทนี้จะทราบเพียงที่อยู่ของตนเองจากอุปกรณ์สื่อสารเครือข่ายเท่านั้น การค้นหาคำตอบสามารถทำได้โดยวิธีควบคุมการสื่อสาร เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ ARP ย้อนกลับ หรือ RARP (Reverse Address Resolution Protocol) วิธีการนี้คอมพิวเตอร์ที่เพิ่งจะเริ่มทำงาน (หรือเครื่องใดก็ได้แล้วแต่) จะส่งคำถามออกไปในทำนอง “ที่อยู่ขนาด 48 bits แบบ ฮาร์ดแวร์ของฉันคือ 14.04.05.18.01.25 มีใครทราบที่อยู่ IP ของฉันบ้าง” เครื่องที่ให้บริการ RARP จะตรวจสอบข้อมูลในตารางข้อมูลของตนเองแล้วจึงส่งหมายเลข IP กลับไปให้วิธีการนี้ช่วยให้เกิดความอ่อนตัวและประสิทธิภาพในการใช้หมายเลข IP เนื่องจากผู้ใช้ไม่มีหมายเลข IP เป็นของตนเอง ผู้ควบคุมระบบสามารถกำหนดหมายเลข IP ใด ๆ ที่ไม่มีผู้ใช้งานในขณะนั้นให้ใช้ได้หมายเลข IP ในที่นี้จึงเป็นเสมือนสมบัติส่วนกลางที่ทุกคนใช้ร่วมกัน

2.1.11 Network Interface Layer

การทำงานระดับล่างสุดต่อจาก Internetwork Layer จะเป็นการแปลงข้อมูล IP datagram ให้อยู่ในรูปที่เหมาะสมและแปลงสัญญาณไฟฟ้าส่งไปยังเครือข่ายต่อไป ซึ่งในชั้น Network Interface Layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นการรวม 2 Layer เข้าด้วยกัน คือ Data link และ Physical Layer กล่าวโดยสรุปคือ การทำงานในชั้นต่าง ๆ ตามโครงสร้างของโปรโตคอล TCP/IP

2.1.11.1 MAC Address

เนื่องจากคอมพิวเตอร์แต่ละเครื่องสามารถแชร์ข้อมูลกันได้ ในระบบเครือข่ายเดียวกันดังนั้นแต่ละเครื่องควรมีสิ่งที่ชี้ลักษณะเฉพาะตัวของมัน เช่น เราต้องมีบัตรประจำตัวประชาชน ซึ่งในทางคอมพิวเตอร์นี้เราจะใช้เลขฐาน 16 จำนวน 12 digits เป็นตัวบ่งชี้ลักษณะเฉพาะนั้น ๆ ซึ่งเราเรียกว่า MAC address เนื่องจาก MAC address เป็นตัวบ่งชี้ลักษณะเฉพาะของแต่ละ machine ดังนั้นจึงต้องเป็นค่าที่ไม่ซ้ำกัน (unique) MAC address เป็นเลข 48 bits โดยแบ่งออกเป็น 2 ส่วน โดย 24 bits แรกเป็นค่าที่แสดงบริษัทที่ผลิตการ์ดนั้น ๆ ส่วน 24 bits หลังเป็น serial number ที่ทางบริษัทกำหนดให้ซึ่งแต่ละตัวต้องไม่ซ้ำกันเราเรียกเลข 24 bits นี้ว่า OUI (Organizationally Unique Identifier) ซึ่ง OUI จะใช้เพียง 22 bits เท่านั้น ส่วนอีก 2 bits ที่เหลือจะถูกใช้เพื่อวัตถุประสงค์อื่น โดย bits หนึ่งจะใช้เพื่อแสดงว่า address นั้นเป็น broadcast/multicast address ส่วนอีก bit หนึ่งนั้นไว้แสดงว่า adapter นั้นถูกกำหนด locally administered address ซึ่ง admin ของระบบจะทำการกำหนด MAC address เพื่อความเหมาะสมของนโยบายระบบ

2.1.11.2 Domain Name System (DNS)

ระบบ Domain Name System หรือ DNS นี้เป็นระบบจัดการแปลงชื่อไปเป็นหมายเลข IP Address (name-to-IP Address mapping) โดยมีโครงสร้างฐานข้อมูลแบบลำดับชั้นเพื่อใช้เก็บข้อมูลที่เรียกได้อย่างรวดเร็วทั่วโลกหลักของระบบ DNS คือ ทำหน้าที่แปลงข้อมูลชื่อและหมายเลข IP Address หรือทำกลับกันได้ นอกจากนี้ยังมีฟังก์ชันเพิ่มเติมอื่น ๆ อีก เช่น แจกชื่อของอีเมลเซิร์ฟเวอร์ใน domain ที่เอ็กส่าร์นเป็นเอ็กส่าร์ที่ส่งวันเวลาสำหรับการใช้งานเพื่อการศึกษาค้นหา เอนูญาตให้หาไปไซประเยชนดานการค้ำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับผิดชอบ ในระบบ DNS จะมีการกำหนด name space ที่มีกฎเกณฑ์อย่างชัดเจนมีกลไกการเก็บข้อมูลเป็นฐานข้อมูลแบบกระจาย ทำงานในลักษณะของไคลเอนต์เซิร์ฟเวอร์ (Client/Server) โดยมี DNS sever ให้บริการเรียกค้นชื่อและแปลงข้อมูลให้ตามที่เครื่องลูกข่าย (DNS Client) เรียกเข้ามา เช่น เครื่องลูกข่ายที่ต้องการรับส่งอีเมลล์หรือโอนไฟล์ข้อมูลให้กับเครื่องอื่น อาจจะทราบเพียงชื่อของเครื่องเซิร์ฟเวอร์ที่ให้บริการเท่านั้น แต่ไม่ทราบหมายเลข IP address กลไก DNS จะทำหน้าที่แปลงข้อมูลและแจ้งให้เครื่องลูกข่ายทราบตามที่ได้ส่งคำสั่งขอข้อมูลมาและกลไกการทำงานแบบนี้เองทำให้เครื่องซึ่งทำหน้าที่ DNS สามารถเป็นเซิร์ฟเวอร์และไคลเอนต์ของ DNS ในเครื่องเดียวกัน ดังนั้นเครื่องที่ให้บริการในเครือข่ายอินเทอร์เน็ต 1 เครื่องจะมีการอ้างถึงได้หลายแบบดังนี้

- อ้างตามชื่อ domain เช่น mail.provision.co.th
- อ้างตาม IP Address เช่น 204.183.255.30
- อ้างตามหมายเลขฮาร์ดแวร์หรือ MAC address เช่น 00:a0:7c:4d:f2

ข้อกำหนดที่สำคัญของ DNS คือ ชื่อในลำดับชั้นที่สองที่ต่อจาก root ได้มีการกำหนดชื่อเฉพาะที่ระบุรายละเอียดของกลุ่มเอาไว้ชัดเจนแล้วดังนี้

| | |
|------|---|
| .mil | แทนกลุ่มของหน่วยงานทางทหารของสหรัฐอเมริกา |
| .gov | แทนกลุ่มของหน่วยงานรัฐบาล |
| .com | แทนกลุ่มขององค์กรหรือบริษัท |
| .net | แทนองค์กรที่ทำหน้าที่เป็นผู้ให้บริการเครือข่าย |
| .edu | แทนสถาบันการศึกษา |
| .org | แทนองค์กรหรือสมาคมต่างๆ |
| .xx | ตัวอักษร 2 ตัวแทนชื่อประเทศ เช่น .th สำหรับประเทศไทยหรือ .ca สำหรับประเทศแคนาดา เป็นต้น |

ต่อมามีการแก้ไขเพิ่มเติมกลุ่มของ domain name ในลำดับที่สองอีกมากมายหลายกลุ่มด้วยกัน ประกอบด้วย

| | |
|---------|-----------------------------------|
| .aero | แทนกลุ่มอุตสาหกรรมการบิน |
| .biz | แทนองค์กรหรือบริษัทห้างร้านทั่วไป |
| .coop | แทนองค์กรในลักษณะของสหกรณ์ |
| .museum | แทนพิพิธภัณฑ์สถาน |
| .firm | แทนองค์กรหรือบริษัทห้างร้านทั่วไป |
| .store | แทนบริษัทที่มีธุรกรรมทางการค้า |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|-------|--|
| .web | แทนเว็บไซต์ที่ให้ข้อมูลต่าง ๆ |
| .arts | แทนกลุ่มที่มีกิจกรรมทางด้านประเพณีและวัฒนธรรม |
| .rec | แทนองค์กรหรือหน่วยงานที่ทำด้านนันทนาการ |
| .info | แทนองค์กรที่เป็นผู้ให้บริการข้อมูล |
| .nom | สำหรับบุคคลทั่วไป |
| .name | สำหรับบุคคลทั่วไป |
| .pro | สำหรับนักบัญชี, หนายความ, หมอ และผู้เชี่ยวชาญเฉพาะด้าน |

2.1.11.3 DHCP (Dynamic Host Configuration Protocol)

การกำหนดแอดเดรสสำหรับ Internet Protocol หรือ IP address ให้กับเครื่องคอมพิวเตอร์และอุปกรณ์ต่าง ๆ ภายในเครือข่าย TCP/IP เป็นสิ่งที่ผู้บริหารเครือข่ายต้องวางแผนและจัดการอย่างเป็นระบบเพื่อไม่ให้เกิดการใช้งานแอดเดรสที่ซ้ำกัน หรือการกำหนดแอดเดรสผิดทำให้ไม่สามารถสื่อสารถึงกันได้ ในกรณีที่เครือข่ายมีขนาดเล็กการเพิ่มเติมหรือแก้ไขสามารถที่จะกระทำได้เองโดยง่าย แต่หากเครือข่ายมีขนาดใหญ่มา ๆ แล้วจะกลายเป็นงานที่ลำบากและเสียเวลามากเลยทีเดียว

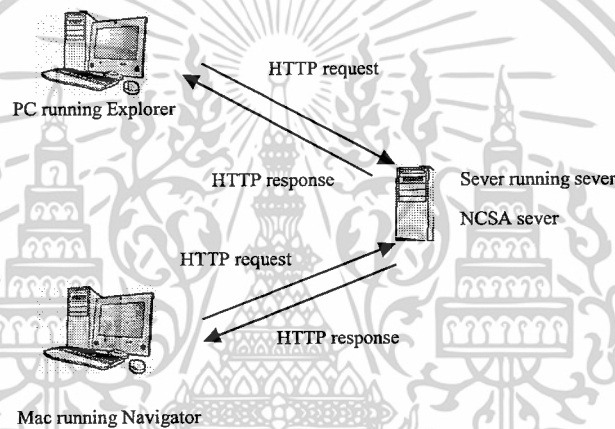
ด้วยเหตุนี้จึงได้มีผู้คิดค้นโปรโตคอล DHCP (Dynamic Host Configuration Protocol: RFC 1531, RFC2131 และ RFC 2132) สำหรับกำหนดแอดเดรสแบบไดนามิกขึ้นมาใช้งาน โดยจะทำหน้าที่แจกจ่ายแอดเดรสและพารามิเตอร์ที่จำเป็นให้กับคอมพิวเตอร์และอุปกรณ์ภายในเครือข่ายโดยอัตโนมัติ การทำงานของโปรโตคอล DHCP ได้รับการปรับปรุงจากโปรโตคอล Bootstrap กล่าวคือ ได้เพิ่มความสามารถในการนำแอดเดรสที่กำหนดให้เครื่องคอมพิวเตอร์ หรืออุปกรณ์นำไปใช้งานกลับมาแล้วแจกจ่ายออกไปใหม่ได้นั่นเอง หลังจากที่มีโปรโตคอล DHCP ผู้ดูแลระบบสามารถวางแผนและกำหนดค่าแอดเดรสรวมทั้งพารามิเตอร์ต่างๆ ที่จำเป็นต้องใช้ได้จากศูนย์กลางเพียงจุดเดียว หากต้องการเปลี่ยนแปลงแก้ไขก็สามารถทำได้ง่ายและรวดเร็วขึ้น

2.1.12 เวิลด์ไวด์เว็บและโปรโตคอลเอชทีทีพี (The World Wide Web: HTTP)

HTTP เป็น protocol ใน Application Layer ซึ่งเป็นหัวใจสำคัญของเว็บซึ่งถูกแบ่งการทำงานออกเป็น 2 ส่วน คือ Client program และ Server program โดยอาจจะมีการทำงานบนระบบที่ต่างกันมีการแลกเปลี่ยน HTTP message กันด้วย ซึ่ง HTTP นั้นจะทำกรกำหนดรูปแบบของข้อมูล (message) และวิธีการแลกเปลี่ยนข้อมูลระหว่าง client และ server Web page ประกอบด้วยออบเจกต์ (Object) คือ ไฟล์ประเภทต่างๆ เช่น HTML, JPEG, GIF, Java applet, audio clip เป็นต้น ซึ่งไฟล์เหล่านี้จะมี URL เป็นตัวบอกที่อยู่ของไฟล์และจะเก็บอยู่ในรูปแบบของไฟล์ HTML สมมติว่าเว็บเพจประกอบด้วยไฟล์รูป JPEG 5 ไฟล์และเท็กซ์ไฟล์ HTML 1 ไฟล์ ดังนั้นเว็บเพจนี้จะประกอบด้วย 6 ออบเจกต์ ทำให้เวลาที่เว็บเพจจะอ้างถึงออบเจกต์ต่างๆ จึงต้องมีการใช้ด้วย URL ของแต่ละออบเจกต์นั่นเอง ในแต่ละ URL จะประกอบด้วยด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ส่วนคือ Host name ของ server และ path ของออบเจ็กต์นั้น ๆ ตัวอย่างเช่น www.someSchool.edu/someDepartment/picture.gif โดยมี www.someSchool.edu เป็น Host name และ /someDepartment/picture.gif เป็น path name เป็นต้น

ส่วน Browser ก็คือโปรแกรมที่ทำหน้าที่ติดต่อระหว่างผู้ใช้ (user) และ web server ที่เป็นที่ยอมรับได้แก่ Netscape Communication และ Microsoft Internet Explorer นอกจากนี้ Browser ยังทำหน้าที่ implement HTTP ของฝั่ง Client อีกด้วย ในขณะที่ Web server นั้นที่เก็บของออบเจ็กต์ต่างๆ ซึ่งในแต่ละออบเจ็กต์จะระบุที่อยู่โดย URL ซึ่ง Web Server ที่เป็นที่ยอมรับได้แก่ Apache, Microsoft Internet Information Server และ Netscape Enterprise Server โดยที่ Web Server จะทำหน้าที่ implement HTTP ของฝั่ง Server ด้วยเมื่อผู้ใช้ต้องการข้อมูลของเว็บเพจ (เช่นคลิก Hyperlink) browser จะส่ง HTTP request เพื่อขอรายละเอียดของ web page



รูปที่ 2.7 HTTP request-response behavior

จากรูป 2. แสดงการติดต่อสื่อสารกันระหว่าง client และ server เมื่อ user ร้องขอเว็บเพจ browser จะทำการส่ง HTTP request message ไปยัง server และเมื่อ server ได้รับ request ก็จะส่ง HTTP response message และออบเจ็กต์ที่ร้องขอกลับไปยัง client HTTP เป็น protocol ใน Application layer โดยใช้บริการของ TCP ดังนั้น browser ต้องทำการสร้าง connection โดยการสร้าง socket ขึ้นมาและติดต่อไปยัง server ที่ port เบอร์ 80 ทางด้าน server เมื่อได้รับคำร้องขอ connection ก็จะตอบกลับไปยัง client เพื่อเริ่มการส่ง message ต่อไป โดย message จะมีการแลกเปลี่ยนกันระหว่าง browser กับ server และเมื่อทำการแลกเปลี่ยน message ทั้งหมดกันเสร็จเรียบร้อยแล้วก็จะทำการปิด connection นั้น HTTP เป็น protocol แบบ stateless คือมันจะไม่เก็บรักษาสถานะต่างๆ ของ client เอาไว้ เช่น Client ใดทำการร้องขอเว็บเพจมาเมื่อไรร้องขออะไรบ้าง เป็นต้น

2.1.12.1 HTTP message format

ทั้ง HTTP/1.0 และ HTTP/1.1 กำหนดรูปแบบของ HTTP message ไว้ 2 แบบด้วยกันคือ HTTP request message และ HTTP response message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTTP Request Message

รูปแบบของ HTTP Request Message เป็นดังนี้

```
GET /somedir/page.html HTTP/1.1
```

```
HOST: www.someschool.edu
```

```
Connection: Close
```

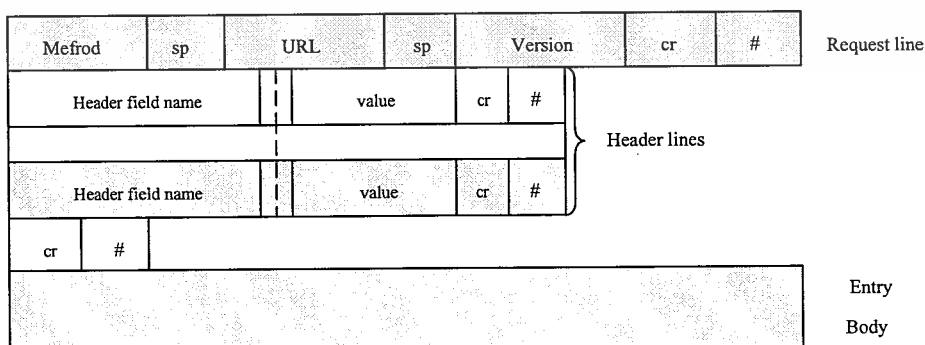
```
User-agent: Mozailla/4.0
```

```
Accept-language: fr
```

```
(Extra carriage return, line feed)
```

จะเห็นว่า Message ประกอบด้วย 5 บรรทัด โดยแต่ละบรรทัดจะจบด้วย carriage return, line feed ซึ่งจริงแล้วใน message อาจมีมากกว่า 5 บรรทัด หรือมีเพียง 1 บรรทัดก็ได้ โดยบรรทัดแรกใน HTTP request message เรียกว่า request line ส่วนบรรทัดอื่นๆ เรียกว่า header line ใน request line จะประกอบด้วย 3 field ได้แก่ method field, URL field และ HTTP version field ซึ่ง HTTP request message ส่วนใหญ่จะใช้ GET method โดยถ้าใช้ GET method ข้อความที่ user กรอกใน Form ในเว็บเพจจะปรากฏอยู่ข้างหลัง URL และถ้ายาวเกินไปจะถูกตัดออกไป แต่ถ้าใช้ Post method ข้อความที่ user กรอกจะปรากฏในส่วน body ของ request message ในส่วนของ Header line ในตัวอย่างนั้นจะมีความหมาย ดังนี้

- Host: ใช้ระบุ host (หรือ URL) ที่ทำการเก็บไฟล์ (Object ที่ต้องการ)
- Connection: close นั้น browser จะบอก server ว่ามันไม่ต้องการใช้ persistent connection แต่ต้องการให้ server ทำการปิด connection หลังจาก server ส่งอบเจ็คต์ที่ร้องขอมายัง client แล้ว ดังนั้นจะพบว่า browser ที่สร้าง request message นี้ถูก implement โดย HTTP/1.1 แต่ไม่ต้องการใช้ persistent connection
- User-agent: ใช้ทำการระบุชนิดของ browser ที่สร้างการร้องขอไปยัง server ในกรณีนี้ user agent คือ Mozilla/4.0 ซึ่งก็คือ Netscape browser
- Accept-language: ใช้บอก server ว่าชนิดของออบเจ็คต์ที่ browser สามารถรับได้นั้นเป็นภาษาอะไรบ้าง



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.8 แสดงรูปแบบทั่วไปของ request message ข้างต้น สังเกตได้ว่าหลังบรรทัดว่างจะตามด้วย “Entity body โดย Entity Body จะไม่ใช้ใน GET method แต่จะใช้ใน POST method โดยที่ HTTP client จะใช้ POST method เมื่อ user มีการใส่ข้อความเข้ามา ตัวอย่างเช่นเมื่อ user ใส่คำที่ต้องการหาใน search engine เช่น Yahoo โดยใช้ POST method ซึ่งก็คือ user ทำการร้องขอ web page จาก server แต่ web page จะถูกกำหนดโดยขึ้นอยู่กับคำที่ user ใส่เข้ามาถ้าใน Method field คือ POST ในส่วนของ Entity Body จะบรรจุคำที่ user ใส่เข้ามา ส่วน HEAD method ก็คล้ายกับ POST method คือ เมื่อ server รับ request ด้วย HEAD method แล้ว server จะทำการตอบ HTTP message กลับมายัง client โดยไม่ส่ง object ที่ client ร้องขอกลับมาด้วย (จะส่งแค่ HTTP header กลับมา) HEAD method จะถูกใช้โดย HTTP server ที่พัฒนาขึ้นมาเพื่อทำการ debugging

HTTP Response Message

รูปแบบของ HTTP Response Message เป็นดังนี้

```
HTTP/1.1 200 OK
Connection: Close
Data: Thu, 06 Aug 1998 10:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Connection-Length: 6821
Content-Type: text/html
```

HTTP response message ประกอบด้วย 3 ส่วน ได้แก่ status line, header line และ Entity Body โดย Entity Body คือเนื้อหาของ message ซึ่งบรรจุ object ที่ client ร้องขอใน Status line มีทั้งหมด 3 field ได้แก่ protocol response field, status code และ status message ของ status code นั้น จากตัวอย่างนี้ status line HTTP/1.1 200 OK จะบอกว่า server ใช้ HTTP/1.1 และทุกอย่าง OK หมายถึง server พบออบเจกต์ที่ร้องขอแล้วและกำลังจัดส่งออบเจกต์ไปให้ client ในส่วนของ Header line server มีรายละเอียด ดังนี้

- Connection: close header line เพื่อบอก Client ว่ามันจะทำการปิด TCP connection ทันทีหลังจากทำการส่ง message เสร็จสิ้นแล้ว

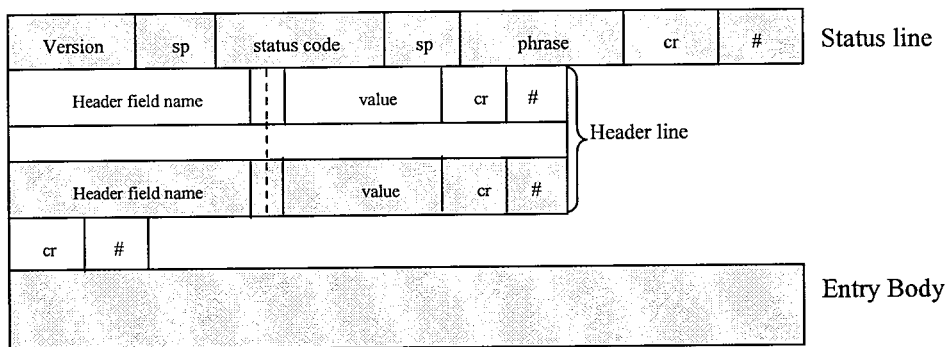
- Date: header line ระบุเวลาและวันที่ที่ server สร้างและส่ง HTTP response นี้ ขอให้สังเกตว่าเวลาดังกล่าว ไม่ใช่เวลาที่สร้างออบเจกต์หรือเวลาที่เก็บออบเจกต์แต่เป็นเวลาที่ server ดึงออบเจกต์ออกจากระบบไฟล์ แล้วใส่ออบเจกต์ไปใน response message และส่ง response message

- Server: header line ระบุว่าใช้ Web server อะไร ตามตัวอย่างเป็นการบอก client ว่าใช้ Apache web server ซึ่งคล้าย User-agent: header line ใน HTTP request message

- Content-Length: header line ระบุจำนวน byte ของออบเจกต์ที่จะส่งไป

- Content-Type: header line ระบุว่าออบเจกต์ในส่วน entity body เป็น text ชนิด HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 General format of a response message

Status code และ Phrase ที่ระบุใน response message ที่พบทั่วไปมีดังนี้

- 200 OK: การร้องขอสำเร็จและข้อมูลถูกส่งกลับไปใน response message
- 301 Moved Permanently: ออบเจ็กต์ที่ client ร้องขอถูกเคลื่อนย้ายไปและ URL ใหม่จะระบุใน

Location: header ของ response message และ client software อาจจะทำการดึงออบเจ็กต์มาจาก URL ใหม่ให้โดยอัตโนมัติ

- 400 Bad Request: เป็นการบอกว่า server ไม่เข้าใจคำร้องขอจาก Client
- 404 Not found: เป็นการบอกว่า document ที่ร้องขอไม่อยู่ใน server นี้
- 505 HTTP Version Not Supported: เป็นการบอกว่า version ของ HTTP protocol ที่ร้องขอไม่

support โดย server

2.1.12.2 NAT (Network Address Translation)

เราจำเป็นต้องมี Public IP address เพื่อใช้ในการเชื่อมต่อเข้ากับ Internet เนื่องจากในปัจจุบันการเติบโตของอินเทอร์เน็ตขยายเพิ่มขึ้นอย่างต่อเนื่องและการเติบโตของอินเทอร์เน็ตทำให้ Public IP Address ที่มีอยู่เดิมแล้วนั้นไม่เพียงพอต่อการใช้งาน ดังนั้น NAT จึงเป็นตัวช่วยหนึ่งในการจัดสรร IP Address ให้เพียงพอในแต่ละองค์กรให้สามารถใช้ private IP เชื่อมต่อและใช้งานอินเทอร์เน็ตได้และเพิ่มความปลอดภัยในเครือข่ายอีกด้วย การแบ่ง IP Address นั้นไม่มีเพียงแต่การทำด้วยวิธีการ NAT เท่านั้นแต่ยังมีการทำด้วย VLSM, SubNetmark

NAT เป็นตัวแรกที่มีชื่อเสียงในการแก้ไขการขาดแคลน IPv4 และช่วยแทนที่ Private Address ที่ใช้งานภายในให้กลายเป็น Public Address ที่ใช้ติดต่อกับโลกอินเทอร์เน็ตได้ซึ่งเราเตอร์ในเครือข่ายอินเทอร์เน็ตต้องใช้ Public Address จึงจะส่งเรดแพ็คเก็ตกลับมาได้ NAT เหมาะสำหรับภายในบ้านและหน่วยงานองค์กรสำหรับเชื่อมต่ออินเทอร์เน็ต โดยที่เครื่องลูกข่ายไม่จำเป็นต้องการออกไปภายนอกเครือข่ายพร้อม ๆ กันในเวลาเดียวกัน NAT เพิ่มความปลอดภัยเครือข่ายภายในด้วยการแปลงเลข IP ทราฟฟิกภายนอกทั้งหมดก่อให้เกิดจากการทำงานของ gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.12.3 การทำงานของ NAT

NAT จะเริ่มสร้างตารางภายในขึ้นมาหนึ่งตารางเพื่อไว้ใช้ในการเก็บหมายเลขไอพีแอดเดรสของเครื่องภายในเครือข่ายที่ส่งแพ็คเกต ผ่าน NAT device และต่อมาจะสร้างตารางไว้สำหรับเก็บหมายเลขพอร์ต (port number) ที่ถูกใช้ไปโดย outside IP address และเมื่อมีการส่งแพ็คเกตจากเครือข่ายภายในไปยังเครือข่ายภายนอก NAT device จะมีกระบวนการทำงานเมื่อต้องส่งแพ็คเกตดังนี้

1. NAT จะบันทึกข้อมูล source IP address และ source port number ไว้ในตารางที่เกี่ยวข้อง (NAT Table)

2. NAT จะแทนที่ IP ของ packet ด้วย IP ขาออกของ NAT device เอง

3. NAT จะ assign หมายเลขพอร์ตใหม่ให้กับ packet และบันทึกค่าพอร์ตนี้ไว้ในตารางและกำหนดค่านี้ลงไป source port number ของ packet นั้น

4. จากนั้นจะคำนวณหา IP, TCP checksum อีกครั้งเพื่อตรวจสอบความถูกต้องและเมื่อ NAT device ได้รับ packet ย้อนกลับมาจาก external network นั่นก็คือ Internet มันจะตรวจสอบ destination port number ของ packet นั้นๆ แล้วนำมาเปรียบเทียบกับข้อมูล source port number ในตารางที่บรรจุข้อมูลไว้ ถ้าเจอข้อมูลที่ตรงกันมันก็จะเขียนทับ destination port number, destination IP address ของ packet นั้นๆ แล้วจึงส่ง packet นั้น ไปยังเครื่องที่อยู่ในเครือข่ายภายในที่เป็นผู้สร้าง packet นี้ขึ้นมาในครั้งแรก

อุปกรณ์เครือข่ายหรือโปรแกรมที่ใช้ในการทำ NAT จะต้องมีความสามารถในการทำงานต่างๆ เหล่านี้คือ

1. สามารถกำหนดหมายเลขไอพีแอดเดรสได้ (Transparent address assignment)

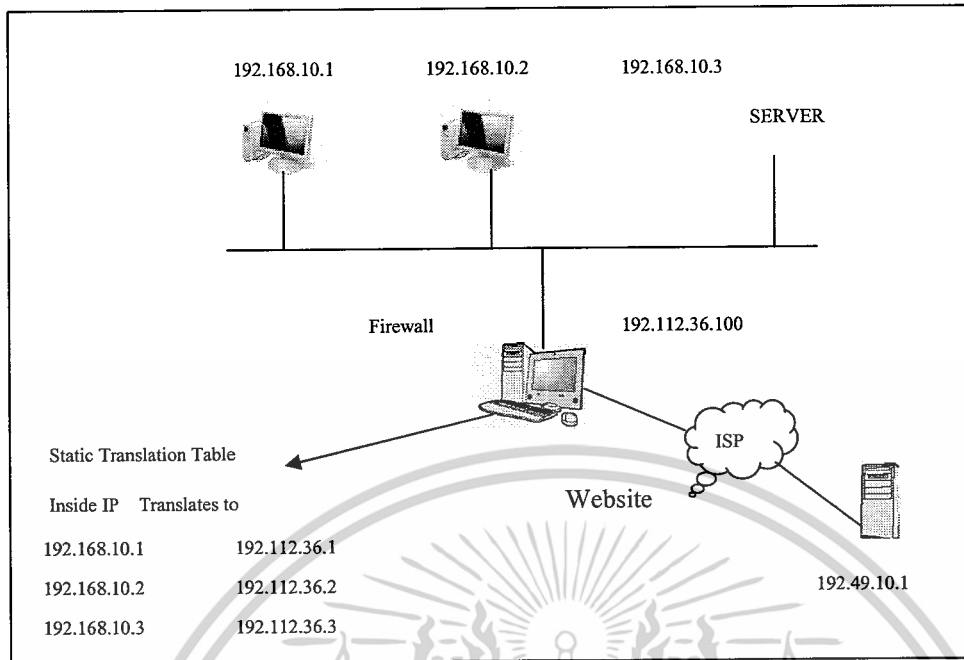
2. สามารถส่งผ่านแพ็คเกตของข้อมูลที่มีการเปลี่ยนแปลงแอดเดรสได้ (Transparent address routing address transition)

3. สามารถเปลี่ยนแปลงข้อมูลของ ICMP payload ได้ (ICMP error message payload translation)

อุปกรณ์ที่จะทำ NAT นั้นต้องสามารถทำการเปลี่ยน IP Address ภายในเครือข่ายเพื่อให้สามารถติดต่อสื่อสารกับโลกอินเทอร์เน็ตได้

2.1.12.4 Static NAT

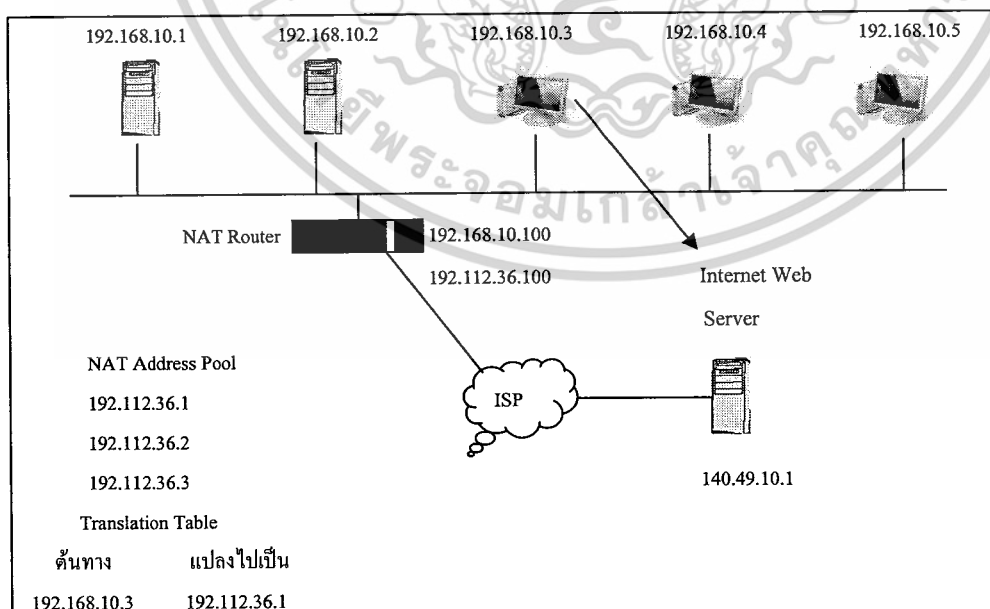
เป็นการทำ NAT แบบหนึ่งต่อหนึ่ง (One to One) คือ การกำหนดให้มีการแมปหมายเลข Private Address ที่อยู่ในเครือข่ายภายในหนึ่งเบอร์จะถูกแทนที่ออกไปเป็นหมายเลข Public Address หนึ่งเบอร์เมื่อทำการเชื่อมต่ออินเทอร์เน็ตและหมายเลขไอพีแอดเดรสจะเป็นหมายเลขที่มีค่าคงที่ไม่เปลี่ยนแปลง



รูปที่ 2.10 เป็นการทำงานแบบ Static NAT

2.1.12.5 Dynamic NAT

เป็นการทำ NAT แบบหลายๆ แอดเดรสต่อหนึ่งแอดเดรส (Many to One) ก็คือ การสร้าง NAT Pool ขึ้นมาหนึ่ง Pool เพื่อมาแชร์แอดเดรสใน Pool นั้นจะมี IP Address หลายเบอร์ เมื่อแพ็คเกจ IP ที่มี Private Address ต่าง ๆ จากภายในเครือข่ายวิ่งผ่านอุปกรณ์ที่ทำ NAT เข้ามา แพ็คเกจ IP แรกที่ผ่านเข้า จะถูกแทนที่ด้วยเครือข่ายด้วยแอดเดรสใน Pool



รูปที่ 2.11 เป็นการทำงานแบบ Dynamic NAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dynamic NAT แบ่งออกเป็น 3 ชนิด

1. many to One คือ มีคอมพิวเตอร์ในเครือข่ายหลายเครื่องแต่จะใช้ไอพีแอดเดรสเพียงหมายเลขเดียว
2. many to many คือ มีคอมพิวเตอร์หลายเครื่องและก็มีหมายเลขไอพีแอดเดรสหลายเบอร์เช่นกัน
3. many to many (Overload) โดยเครื่องคอมพิวเตอร์ในเครือข่ายที่เป็น Private IP เมื่อติดต่อไปยังเครือข่ายอื่นๆ จะถูกเปลี่ยนเป็นหมายเลขไอพีแอดเดรสเพียงหมายเลขเดียวแต่มีการเปลี่ยนแปลงหมายเลขพอร์ตต้นทางในการเชื่อมต่อแทน เป็นการทำงานของการทำงานการทำ PAT

2.1.13 ไมโครชิป TCP/IP Stack (MICROCHIP)

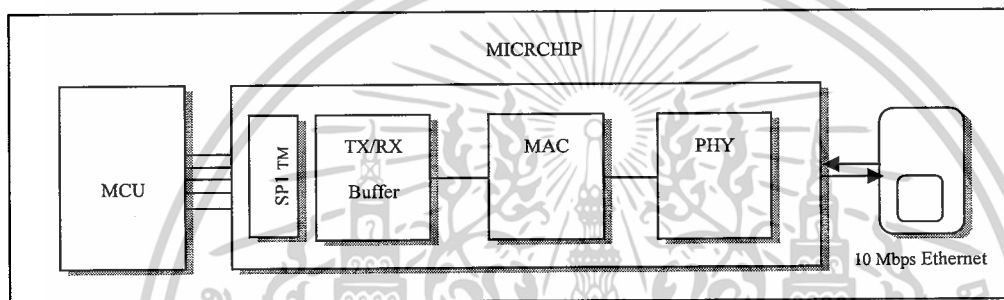
ไมโครชิป TCP/IP stack จะเป็นการนำเอาโปรโตคอล TCP/IP ไปติดตั้งลงบนไมโครคอนโทรลเลอร์ตระกูล PIC เพื่อพัฒนาทางด้าน Embedded Controller ให้มีประสิทธิภาพมากขึ้น ซึ่งภายในโปรโตคอล TCP/IP จะรองรับการทำงานของลำดับชั้นต่างๆ กันในระบบเครือข่าย

ชั้นที่อยู่บนสุดของโปรโตคอลคือ แอปพลิเคชันเลเยอร์ (Application Layer) ผู้ใช้งานสามารถเรียกใช้โปรโตคอลในระดับการใช้งานได้ ตัวอย่างเช่น DHCP, SNMP, HTTP เป็นต้น ลำดับชั้นต่อมาเรียกว่า ทรานสปอร์ตเลเยอร์ (Transportation Layer) ซึ่งทำหน้าที่ในส่วนของการรับส่งข้อมูลหรือแพ็กเกจที่เกิดขึ้นภายในระบบการสื่อสารแบ่งได้สองรูปแบบคือ UDP และ TCP ลำดับชั้นที่สามคือ อินเทอร์เน็ตเวิร์กกิ้งโปรโตคอลเลเยอร์ (Internetworking Protocol Layer) รับผิดชอบช่องทางของการขนย้ายข้อมูลในรูปแบบแพ็กเกจในระบบ ลำดับชั้นที่สี่เรียกว่า แอดเดรสโซลูชันโปรโตคอลเลเยอร์ (Address Solution Protocol Layer) ทำหน้าที่ในการค้นหาค่า Physical Address หรือค่า MAC Address ที่สัมพันธ์กับค่า IP Address ภายในระบบเครือข่าย และลำดับชั้นล่างสุดคือ Ethernet Layer กับ Physical Layer รับผิดชอบในส่วนของอุปกรณ์ที่เป็นฮาร์ดแวร์ทั้งหมด

ไมโครชิปได้สนับสนุนการใช้งานโปรโตคอล TCP/IP อย่างเต็มที่ซึ่งจะให้การรองรับการทำงานของ Application Layer มากขึ้น เช่น HTTP, FTP, TFTP, SMTP เป็นต้น สำหรับ TCP/IP stack ของไมโครชิปจะรองรับการทำงานกับไมโครคอนโทรลเลอร์ตระกูล PIC18, PIC24, และ dsPIC30/33 ซึ่งไมโครคอนโทรลเลอร์ที่จะลงซอฟต์แวร์ก็จะต้องมีหน่วยความจำโปรแกรมอย่างน้อย 64 กิโลไบต์ สำหรับในส่วนของผู้กรณ์ฮาร์ดแวร์ ไมโครชิปได้ออกแบบชิปสำหรับใช้งานในส่วน Physical Layer ซึ่งสามารถทำงานร่วมกับซอฟต์แวร์ตัวอื่นคือ ENC28J60 เป็น Ethernet Controller ทำงานที่ความเร็ว 10Mbps การติดต่อกับชิปตัวนี้ทำได้โดยผ่านพอร์ต SPI กล่าวโดยสรุปว่า ENC28J60 เป็นชิป Stand-Alone ซึ่งเราสามารถที่จะเลือกไมโครคอนโทรลเลอร์บอร์ดก็ได้มาติดต่อกับชิปตัวนี้ รูปที่ 2 แสดงการเชื่อมต่อของ ENC28J60 กับไมโครคอนโทรลเลอร์

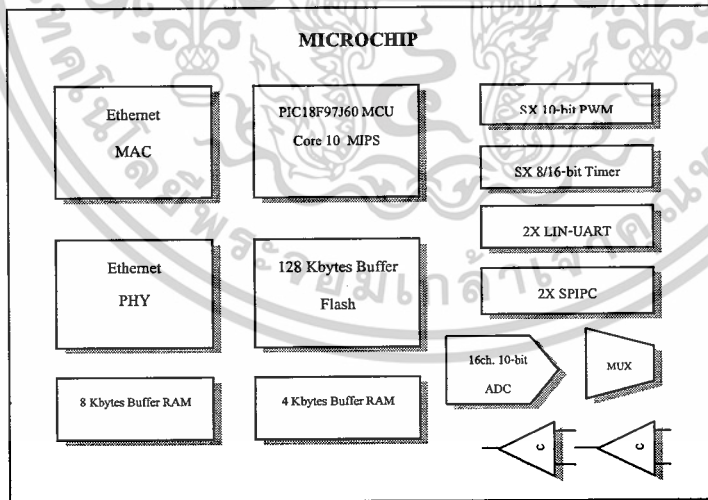
ตารางที่ 2.7 ลำดับชั้นของโปรโตคอล TCP/IP

| | | | | |
|----------|------|------|-----|------|
| DHCP | SNMP | HTTP | FTP | TFTP |
| UDP | | TCP | | |
| ICMP | | | | |
| IP | | | | |
| ARP | | | | |
| Ethernet | | | | |



Stand-Alone Ethernet Controller

รูปที่ 2.12 การเชื่อมต่อ ENC28J60



รูปที่ 2.13 PIC18F97J60 ไมโครคอนโทรลเลอร์ที่มีโมดูล Ethernet ในตัว

นอกจากชิป Ethernet Controller ที่ต่อภายนอกแล้วไมโครชิปยังได้ออกแบบและพัฒนาโมดูล Ethernet สำหรับ PIC18 ซึ่งทำให้ขนาดของการออกแบบแผ่นวงจรและราคาอุปกรณ์ในการเชื่อมต่อภายนอก โดยรูปที่ 2.14 เป็นการแสดงภาพของ PIC18 ที่มีการรวมโมดูล Ethernet ไว้ในชิปตัวเดียวกัน ซึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP/IP Stack ยังรองรับการทำงานของโมดูล Ethernet ใน PIC18 อีกด้วย การใช้งาน PIC18 กับโมดูล Ethernet จึงเป็นเรื่องที่ง่ายมากและไม่ซับซ้อนอีกต่อไป

2.1.13.1 การใช้งานไมโครชิป TCP/IP Stack

การใช้งานสแต็ก ไฟล์ TCP/IPConfig.h และ HardwareProfile.h หลังจากติดตั้งซอฟต์แวร์สแต็กเรียบร้อยแล้ว จะพบกับไฟล์เฮดเดอร์ที่ชื่อว่า ‘TCP/IPConfig’ และ HardwareProfile.h สำหรับไฟล์ตัวแรก มีประโยชน์ในการปรับแต่งค่าคอนฟิกต่าง ๆ เช่น โพรโตคอลที่ต้องการใช้งาน ขนาดของหน่วยความจำที่ใช้เก็บอิมเมจไฟล์ ค่าตำแหน่งของ IP, Subnet, และ Gateway ค่าตำแหน่งของ MAC เป็นต้น ตัวอย่างเช่น หากต้องการปรับเปลี่ยนชื่อของบอร์ดทดลองให้ไปแก้ที่มาโคร MY_DEFAULT_HOST_NAME (ค่าเริ่มต้นคือ http://mhboard/) สามารถตั้งได้สูงสุด 15 ตัวอักษร ซึ่งชื่อของบอร์ดจะอ้างได้แต่เฉพาะระบบเน็ตเวิร์กแบบโลคอลเท่านั้นเพราะใช้ NetBIOS name server ไม่สามารถอ้างกับอินเทอร์เน็ตภายนอกได้ ตัวอย่างที่สองเป็นการปรับเปลี่ยนค่าตำแหน่งของ MAC Address ซึ่งในที่นี้จะขอเรียกย่อ ๆ ว่า MAC โดยปกติค่า MAC จะไม่สามารถแก้ไขได้จะถูกกำหนดมากับตัวฮาร์ดแวร์ของผู้ผลิตเพื่อไม่ให้เกิดซ้ำกัน โดยจะมีหน่วยงานสากลที่ทำหน้าที่ในการจัดลำดับหมายเลข MAC ให้กับผู้ผลิตอีกทีดังนั้นหากใครต้องการจะสร้างอุปกรณ์ที่ใช้ในระบบเน็ตเวิร์กจะต้องไปขอหมายเลข MAC จากหน่วยงานดังกล่าวเพื่อป้องกันการซ้ำซ้อนของตัวเลข สำหรับมาตรฐานที่ใช้กำหนดค่า MAC มีชื่อเรียกว่า Organizationally Unique Identifier (OUI) ซึ่งจะอ้างถึงตัวเลขทั้งหมด 6 ไบต์โดยปกติจะอ้างเป็นเลขฐานสิบหกเพื่อให้ง่ายต่อการจำ

ตัวอีเทอร์เน็ตคอลโทรลเลอร์ของไมโครชิปจะสามารถเปลี่ยนค่า MAC ได้ทั้งหมดไบต์ แต่ไม่แนะนำให้ทำเช่นนั้นเนื่องจากหมายเลขอาจจะไปชนกับอุปกรณ์ตัวอื่น โดยในการใช้งานให้กำหนดค่าในสามไบต์แรกเป็น 00-04-A3 ซึ่งจะเป็หมายเลขที่ไม่โครชิปลงทะเบียนไว้แล้ว ดังนั้นเราจะแก้ไขเฉพาะสามไบต์สุดท้าย และจำนวนสูงสุดของอุปกรณ์ที่อ้างได้ในหนึ่งวงคือ 65,535 ตัว ถ้าหมายเลขของอุปกรณ์คือ 12345 ค่า MAC จะมีค่า 00-04-A3-00-30-39 ซึ่งจะแก้ไขที่ MY_DEFAULT_MAC_BYTEX (X คือเลข 1 ถึงเลข 6

2.1.13.2 การเชื่อมต่อกับระบบเน็ตเวิร์ก

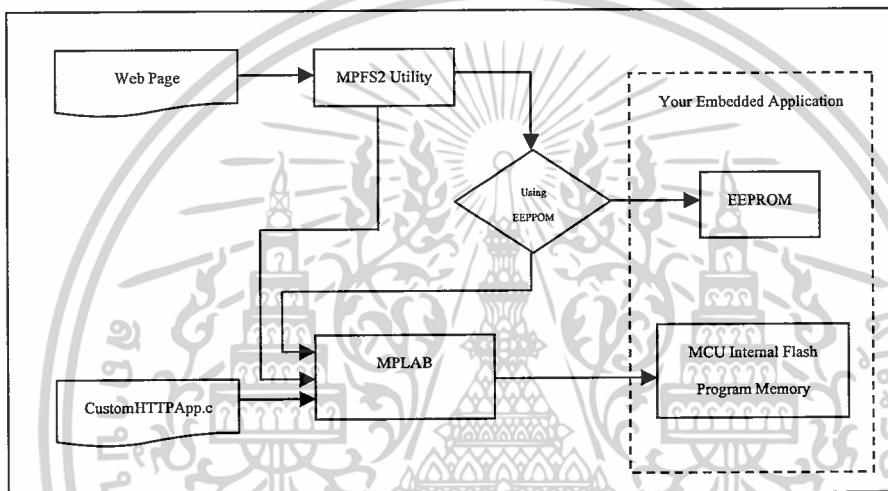
ในการเชื่อมต่อกันของระบบเน็ตเวิร์ก TCP/IP จะอาศัยการกำหนดค่า IP แอดเดรส ซึ่งเป็นซอฟต์แวร์แอดเดรส สำหรับฮาร์ดแวร์จะกำหนดผ่านค่า MAC แอดเดรส โพรโตคอล DHCP (Dynamic Host Configuration Protocol) จะเป็นตัวกำหนดค่า IP แอดเดรสอัตโนมัติให้กับอุปกรณ์ในระบบเน็ตเวิร์กภายในโปรแกรมทดลองของสแต็กจะมีการกำหนดการใช้งาน DHCP server และ DHCP Client ซึ่งทำให้การเชื่อมต่อบอร์ดทดลองกับระบบเน็ตเวิร์กทำได้โดยปราศจากการคอนฟิก ตัวบอร์ดโดยจะรับค่า IP แอดเดรสของอัตโนมัติจาก DHCP server ของระบบเน็ตเวิร์ก

ถ้าหากเชื่อมต่อบอร์ดทดลองกับคอมพิวเตอร์โดยตรงไม่ผ่านสับหรือสวิตช์ โดยใช้สายเคเบิลแบบ Crossover บอร์ดทดลองจะทำตัวเป็น DHCP server และแจกค่า IP แอดเดรสให้กับคอมพิวเตอร์อัตโนมัติ สำหรับการสังเกตว่าระบบเน็ตเวิร์กที่เชื่อมต่อทำงานถูกต้องหรือไม่ ให้ดูได้จาก LED ที่อยู่หน้าการตั้งค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้างๆ ตัวอีเทอร์เน็ตคอลโทรลเลอร์ โดยปกติหากมีการเสียบสายเคเบิลถูกต้อง LED ทั้งสองดวงจะติดและกระพริบ

2.1.13.3 โพรโทคอล HTTP2 และ อิมเมจไฟล์ MPFS2

โพรโทคอล HTTP2 จะรองรับการทำงานของ Web server ดังนั้นในการสร้างอิมเมจไฟล์จึงจำเป็นต้องเปลี่ยนโครงสร้างโดยใช้เครื่องมือ MPFS2 ซึ่งจะสร้างระบบไฟล์แบบใหม่ให้สามารถรองรับการทำงานของ HTTP2 ได้ โพรโทคอล HTTP2 ก็คือ โพรโทคอล HTTP เวอร์ชันใหม่ โมดูลใหม่จะรองรับการทำงานของ Authentication (ระบบรักษาความปลอดภัย), POST form, GZIP Compression, Cookies เป็นต้น



รูปที่ 2.14 แผนภาพของการทำงานของ โพร โทคอล HTTP2

ในส่วนแรกคือ เว็บเพจซึ่งประกอบไปด้วย HTML และไฟล์รูปทั้งหมดรวมถึงสไตลชีต CCS และจาวาสคริปต์ไฟล์สำหรับการแสดงผลเว็บไซต์ ไฟล์ที่กล่าวถึงมาทั้งหมดจะถูกเก็บอยู่ที่โฟลเดอร์ที่มีชื่อว่า WebPages2

ส่วนที่สองคือ MPFS2 เป็นโปรแกรมสำเร็จรูปที่ทางไมโครชิปจัดเตรียมไว้ให้สำหรับแปลงเว็บเพจให้กลายเป็นอิมเมจไฟล์เวอร์ชัน 2 โปรแกรมตัวนี้จะทำการชี้ตำแหน่งของตัวแปรแบบ dynamic ที่เขียนอยู่ในเว็บเพจและเก็บค่าที่ชี้ไว้ในไฟล์ HTTPPrint.h ซึ่งจะต้องถูกคอมไพล์โดย MPLAP IDE อีกที ถ้าเก็บอิมเมจไฟล์ไว้ในหน่วยความจำข้อมูลภายนอกโปรแกรมจะสร้างอิมเมจไฟล์นามสกุล BIN ถ้าอิมเมจไฟล์เก็บไว้ในหน่วยความจำโปรแกรมจะเป็นนามสกุล C ซึ่งจะต้องคอมไพล์พร้อมกับโปรเจกต์ทั้งหมดโดย MPLAP IDE

ถ้ามีการแก้ไขตัวแปรแบบไดนามิก ไม่ว่าจะเพิ่มหรือลบออกจากแอปพลิเคชันโปรแกรม MPFS2 จะแก้ไขไฟล์ HTTPPrint.h ให้อัตโนมัติ หากมีการแก้ไขเกิดขึ้นโปรเจกต์ทั้งหมดจะต้องถูกรีคอมไพล์ใหม่เพื่อให้แน่ใจว่าตัวแปรใหม่จะชี้ค่าได้ถูกต้องตามแอปพลิเคชัน

ไฟล์ CustomHTTPApp.c มีไว้สำหรับเว็บแอปพลิเคชันโดยเฉพาะ ซึ่งจะแสดงผลค่าเอาต์พุตผ่านตัวแปรไดนามิกจากฟังก์ชัน (HTTPPrint*callbacks) หรือจัดการข้อมูลที่ถูกส่งผ่านฟอร์ม ด้วยฟังก์ชันไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HTTPExe-cuteGet และ HTTPExecutePost และการตรวจสอบรับรองการเข้าถึงข้อมูลผ่านฟังก์ชัน HTTPAuthenticate เป็นต้น ตัวการใช้งานเว็บแอปพลิเคชันจะอยู่ในเว็บเพจของสแต็กซึ่งสามารถเข้าใจได้ไม่ยากเพราะจะมีคำอธิบายขั้นตอนและวิธีการใช้งานอย่างละเอียด

ไฟล์สุดท้ายคือ HTTPPrint.h ถูกสร้างจากโปรแกรม MPFS2 ซึ่งจะเก็บค่าอินเด็กซ์ของตัวแปรไดนามิกและเป็นไฟล์ที่เชื่อมตัวแปรภายในเว็บเพจกับตัวแปรที่ฟังก์ชัน HTTPPrint ซึ่งเก็บอยู่ที่ไฟล์ CustomHTTPApp.c ในการใช้งานเราไม่จำเป็นต้องแก้ไข HTTPPrint.h ด้วยตนเองแต่โปรแกรม MPFS2 จะเป็นตัวจัดการให้ทั้งหมด

คุณสมบัติที่เพิ่มเข้ามาใน HTTP2 และ MPFS2

โปรโตคอล HTTP2 สามารถรองรับการทำงานในส่วนของเว็บแอปพลิเคชันได้มากขึ้น โดยสรุปจะมีคุณสมบัติเพิ่มเติมจากตัวก่อนอยู่ 7 ข้อ คือ

1. การอ้างตัวแปรแบบไดนามิกระหว่างเว็บแอปพลิเคชันกับ HTML

ข้อมูลเอาต์พุตของแอปพลิเคชันจะแทนด้วยตัวแปรไดนามิก จากวิธีการดังกล่าวจะอนุญาตให้ข้อมูลของตัวแปรภายในไมโครคอนโทรลเลอร์สามารถแสดงค่าผ่านทางหน้าเว็บเพจได้ วิธีการสร้างตัวแปรแบบไดนามิกในเว็บเพจทำได้โดยการประกาศชื่อตัวแปรให้อยู่ระหว่างเครื่องหมาย “~” โปรแกรม MPFS2 จะสร้างค่าอินเด็กซ์สำหรับตัวแปรแบบไดนามิกภายในไฟล์ HTTPPrint.h และตัวแปรไดนามิกจะทำงานผ่านฟังก์ชัน HTTPPrint_myVariable ซึ่งอยู่ภายในไฟล์ CustomHTTPApp.c

2. การทำงานของ HTML Form

HTML Form คือ ตัวรับข้อมูลภายในแอปพลิเคชัน ซึ่งจะอนุญาตให้แอปพลิเคชันส่งข้อมูลผ่านฟอร์มโดยวิธี GET หรือ POST เมื่อมีการส่งข้อมูลภายในฟอร์มเกิดขึ้น ฟังก์ชัน HTTPExecuteGet หรือ HTTPExe-cutePost จะทำหน้าที่ในการจัดการกับตัวแปรที่เราอ้างถึง

3. การรับรองสิทธิ์ในการเข้าถึงหน้าเว็บเพจ (Authentication)

ไมโครชิป HTTP2 เซิร์ฟเวอร์รองรับวิธีการรับรองสิทธิ์ขั้นพื้นฐานในการเข้าสู่หน้าเว็บเพจ เว็บแอปพลิเคชันจะถาม username และ password สำหรับการเข้าถึงหน้าเว็บเพจบางหน้าต่างที่ต้องการเก็บเป็นความลับ ฟังก์ชันที่ถูกเรียกใช้งานคือ HTTPAuthenticate

4. การใช้คุกกี้ (Cookies)

โดยทั่วไปโปรโตคอล HTTP ถูกออกแบบให้ใช้งานในลักษณะ session-less และ state-less ซึ่งทำให้การใช้งานสามารถจำค่าสถานะต่างๆ ได้ ถ้าหากต้องการใช้ session สามารถทำได้โดยการตั้งค่า Cookies ในโปรแกรมเว็บเบราว์เซอร์ได้

5. ไฟล์อัปโหลด (File Uploads)

เนื่องจาก ไมโครชิป HTTP เซิร์ฟเวอร์รองรับการทำงานของ POST ดังนั้นเราจึงสามารถใช้งานไฟล์อัปโหลดผ่านโปรโตคอล HTTP ได้ ประโยชน์ของการใช้งานไฟล์อัปโหลด เช่น การคอนฟิกระบบผ่านโปรโตคอล HTTP หรือการอัปเดตเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การบีบอัดข้อมูลแบบ GZIP (GZIP Compression)

โปรแกรมเว็บเบราว์เซอร์สามารถรับไฟล์ที่ถูกเข้ารหัสด้วยการบีบอัดข้อมูลแบบ GZIP สำหรับไฟล์ที่ไม่มีตัวแปรแบบไดนามิกสามารถที่จะลดขนาดของการส่งข้อมูลได้ถึง 60% โปรแกรม MPFS2 จะทำการค้นหาไฟล์ที่สามารถบีบอัดได้อัตโนมัติ และจะเก็บไฟล์ที่บีบอัดไว้ในอิมเมจไฟล์ HTTP เซิร์ฟเวอร์จะส่งข้อมูลที่ถูกระบุบีบอัดกลับไปยังเบราว์เซอร์ปกติเนื่องที่ในการเก็บข้อมูลจะใช้น้อยลงทำให้การส่งข้อมูลกลับไปยังโคลแอนทำได้รวดเร็วมากขึ้น

7. การอัปโหลดไฟล์อิมเมจ (MPFS2 Uploads)

หากใช้งาน EEPROM ภายนอกสำหรับเก็บเว็บเพจ เราสามารถอัปโหลดอิมเมจไฟล์นามสกุล BIN ลงบนบอร์ดทดลองผ่านโปรแกรมเบราว์เซอร์ ซึ่งไม่จำเป็นจะต้องใช้ FTP เซิร์ฟเวอร์อีกต่อไป ทำให้ประหยัดการใช้งาน TCP Sockets หรือ หน่วยความจำข้อมูลประมาณ 1 กิโลไบต์ของอีเทอร์เน็ตคอนโทรลเลอร์ เราสามารถปิดการใช้งานในส่วนของอิมเมจไฟล์อัปโหลดได้โดยการคอมเมนต์มาโคร HTTP_MPFS2_UPLOAD ในไฟล์ TCP/IPConfig.h หรือ หากต้องการเปิดการใช้งานการรับรองสิทธิ์ของการอัปโหลดไฟล์เพื่อถาม username และ password ก็สามารถทำได้เช่นเดียวกัน

2.2 สเต็ปเปอร์มอเตอร์ (STEPPER MOTOR)

สเต็ปเปอร์มอเตอร์เป็นมอเตอร์ที่มีการหมุนเป็นสเต็ป (STEP) เมื่อจ่ายกระแสไฟฟ้าไปยังขดลวดที่พันอยู่บนสเตเตอร์ (stator) ในลักษณะเป็นพัลส์ (Pulse) ในกรณีที่จ่ายกระแสไฟฟ้าให้กับขดลวดขดนั้นของสเต็ปเปอร์มอเตอร์ตลอดเวลาจะเกิดการหมุนเพียงหนึ่งสเต็ปเท่านั้นซึ่งต่างจาก DC มอเตอร์เมื่อจ่ายกระแสไฟฟ้าให้ก็จะหมุนตามจนกว่าจะหยุดจ่ายกระแสไฟฟ้าให้

ข้อดีของสเต็ปเปอร์มอเตอร์เมื่อเปรียบเทียบกับ DC มอเตอร์

1. สามารถใช้ในงานควบคุมตำแหน่งในลักษณะวงจรควบคุมแบบเปิด (Open Loop Control) ได้ โดยที่ไม่ต้องการสัญญาณป้อนกลับ (FeedBack signal) แต่อาศัยการนับจำนวนของพัลส์ที่ส่งไปควบคุมการหมุนแทน

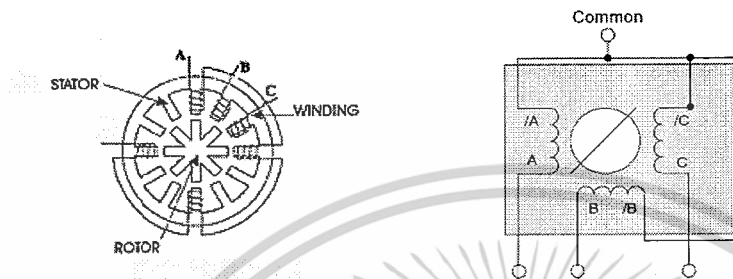
2. ไม่มีส่วนของแปรงถ่านที่จะสึกหรอและไม่เกิดการสปาร์คที่แปรงถ่านซึ่งอาจก่อให้เกิดสัญญาณรบกวน

2.2.1 ชนิดของสเต็ปเปอร์มอเตอร์

1. วาริเอเบิลรีลักแตนซ์ (Variable Reluctance : VR) โรเตอร์ (Rotor) ทำด้วยเหล็กอ่อนรูปทรงกระบอกและทำเป็นลักษณะฟัน (teeth) สเตเตอร์ (Stator) จะมีขดลวดพันและจะทำการเป็นลักษณะของฟันเช่นกัน เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันของสเตเตอร์ และเหนี่ยวนำให้ฟันของโรเตอร์ เกิดเป็นขั้วแม่เหล็กที่มีขั้วตรงกันข้ามกับสเตเตอร์ทำให้ดึงดูดกันเกิดการหมุนของโรเตอร์ขึ้นมอเตอร์ชนิดนี้โดยปกติจะมีขนาด 3 เฟส ในบางครั้งอาจพบ ถึง 4 เฟส มอเตอร์ชนิดนี้ถ้าไม่จ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์ ตัวโรเตอร์จะไม่เกิดแรงดึงดูดกับสเตเตอร์ มอเตอร์ชนิดนี้ไม่นิยมนำไปใช้งานอุตสาหกรรมแต่จะถูกนำไปใช้กับงานที่มีขนาดเล็ก เช่น Micro-positioning table เป็น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

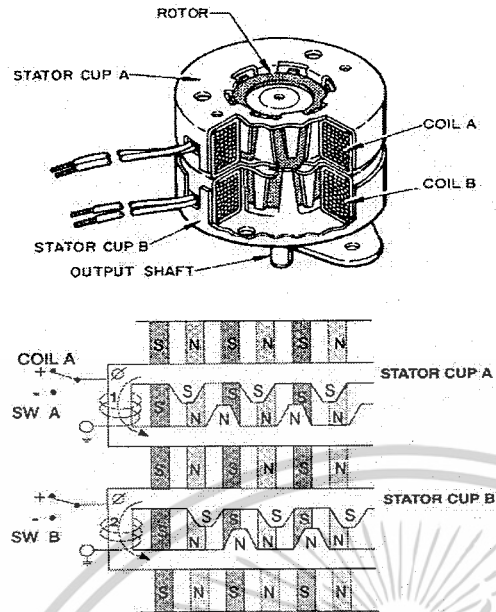
ต้นเพราะไม่มีส่วนที่เป็นแม่เหล็กถาวร ดังนั้นในขณะที่ไม่จ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จึงไม่เกิดแรงดึงดูด วิธีการขับ(Driving)หรือการกระตุ้นเฟส(Phase Excitation)จะทำดังนี้คือ ต่อปลายด้าน common เข้ากับแหล่งจ่ายไฟขั้วบวก(+) แล้วทำการสวิตซ์ให้ปลายด้าน A , B , C ต่อลงกราวด์ (Ground) ตามลำดับ ทีละปลายแล้วทำเช่นนี้เรื่อยไป แต่ถ้าต้องการให้หมุนกลับก็สวิตซ์ย้อนกลับ และในการอธิบายต่อจากนี้ไปจะไม่ขอก้าวถึงมอเตอร์ชนิดนี้อีก



รูปที่ 2.15 โครงสร้างสเต็ปเปอร์มอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์และสัญญาณลักษณะ

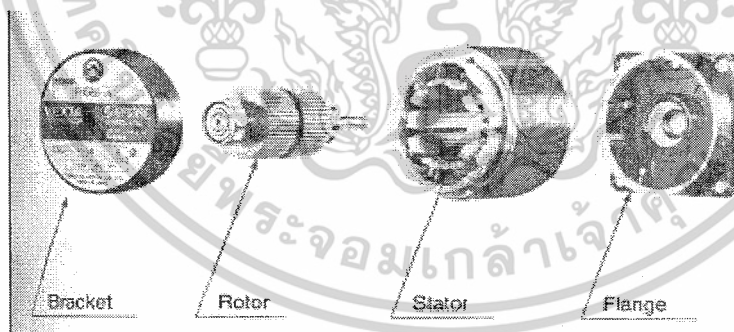
2. แบบแม่เหล็กถาวร (Permanent Magnet : PM) โรเตอร์(Rotor) ทำด้วยแม่เหล็กถาวร รูปทรงกระบอกเรียบสเตเตอร์ (Stator) จะมีขดลวดพันและก็จะทำเป็นฟันเมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันของสเตเตอร์ และจะดึงดูดกับขั้วของแม่เหล็กถาวรที่โรเตอร์ ทำให้เกิดการหมุนของโรเตอร์ขึ้น มอเตอร์ชนิดนี้โดยจะมีตั้งแต่ขนาด 2 เฟสขึ้นไปมอเตอร์ชนิดนี้ไม่นิยมนำไปใช้ในงานอุตสาหกรรมแต่จะถูกนำไปใช้กับอุปกรณ์คอมพิวเตอร์เช่นตัวขับวงล้อที่ใช้หมุนเพื่อเลื่อนกระดาษของเครื่องพิมพ์ เป็นต้น เพราะความเร็วต่ำแรงบิดต่ำ และนอกจากนี้ด้วยโครงสร้างของมอเตอร์ชนิดนี้ทำให้มุมที่หมุนไปแต่ละสเต็ป (Step Angle) ไม่ละเอียดเช่น สเต็ปละ 3.6, 7.5 , 15 , 18 องศาเป็นต้น มอเตอร์ชนิดนี้ถึงไม่จ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์ (Stator) ตัวโรเตอร์จะเกิดแรงดึงดูดกับสเตเตอร์ซึ่งเกิดจากอำนาจของแม่เหล็กถาวรที่โรเตอร์ทำให้หมุนได้ยากจำนวนขั้วแม่เหล็กที่โรเตอร์สามารถนับได้จากจำนวนขั้วแม่เหล็กที่จะเกิดขึ้นจากกระแสไฟฟ้าไหลผ่านขดลวดที่สเตเตอร์ชุดใดชุดหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 สเต็ปเปอร์มอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต

3. แบบผสม (Hybrid : HB) ใช้หลักการการทำงานของทั้งสองแบบมาออกแบบโดยที่สเตเตอร์จะคล้ายกับแบบ VR ส่วนโรเตอร์จะคล้ายแบบ PM แต่จะทำเป็นฟันมอเตอร์แบบนี้นิยมใช้งานอุตสาหกรรมเพราะแรงบิดสูงความละเอียดของสเต็ปในการหมุนสูง ความเร็วสูงกว่าสองแบบที่กล่าวมาแล้ว มอเตอร์ชนิดนี้โดยปกติจะมีขนาด 2 เฟส ถึง 5 เฟส และมอเตอร์ชนิดนี้ได้มีการพัฒนาให้มีประสิทธิภาพเหนือกว่าเดิมไปอีกโดยให้ชื่อว่า “Enhanced Hybrid” ซึ่งจะไม่มีขายโครงสร้างในเอกสารนี้



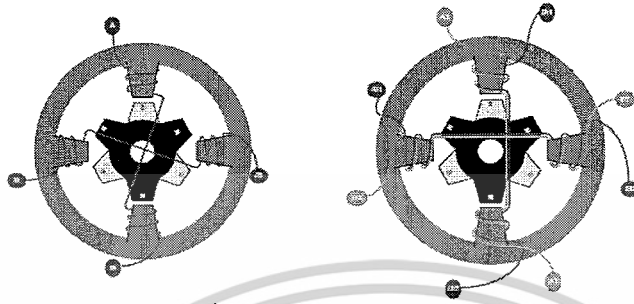
รูปที่ 2.17 สเต็ปเปอร์มอเตอร์ชนิดไฮบริดขนาด 5 เฟส

2.2.2 เฟสของสเต็ปเปอร์มอเตอร์ (Stepper Motor Phase) หมายถึง จำนวนขดลวดที่พันอยู่บนสเตเตอร์ซึ่งแยกออกจากกันอย่างอิสระ รูปที่ 1 แสดงตัวอย่างมอเตอร์ขนาด 3 เฟส ในกรณีของมอเตอร์แบบยูนิโพลาร์ 2 เฟสนั้นมักถูกจะเรียกเป็นมอเตอร์ขนาด 4 เฟสก็เพราะขดลวดที่พันอยู่บนสเตเตอร์แต่ละขดจะมี 2 ขด จึงเข้าใจว่ามี 4 ขดลวด แต่ถ้าพิจารณากันจริงจะพบว่าขดลวดทั้งสองนั้นเป็นขดลวดขดเดี่ยวแต่มีจุดต่อตรงกลางขดเท่านั้นดังแสดงในรูปที่ 2.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพันลวดบนสเตเตอร์ของสเต็ปเปอร์มอเตอร์แบ่งออกได้ 2 แบบคือ

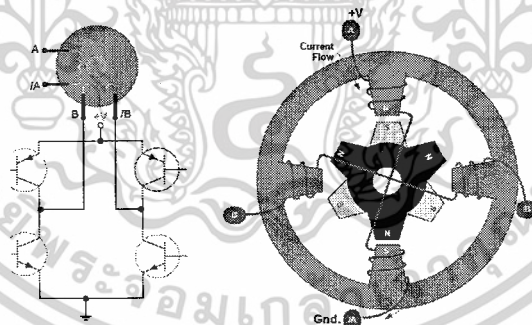
1. การพันแบบใช้ลวดเส้นเดียว (Unifilar)
2. การพันแบบใช้ลวด 2 เส้นพันไปด้วยกัน (Bifilar)



รูปที่ 2.18 ลักษณะการพันขดลวดบนสเตเตอร์

ชนิดของสเต็ปเปอร์มอเตอร์แบ่งตามลักษณะสายที่ใช้ต่อกับวงจรขับเคลื่อนออกได้ 2 แบบคือ

1. แบบไบโพลาร์ (Bipolar) ขดลวดที่สเตเตอร์แต่ละขดจะไม่มีจุดร่วม การต่อเข้ากับวงจรขับเคลื่อนจะใช้ปลายทั้งสองด้านของขดลวดแต่ละขด การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟจากปลายด้านหนึ่งไปยังปลายอีกด้านหนึ่งของขดลวดและการเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ขดเดียวกันนี้ก็ทำได้โดยสลับทิศทางการไหลของกระแสไฟฟ้านั้นเอง ดังนั้นวงจรขับที่ใช้จึงจำเป็นต้องสามารถกลับทิศทางการไหลของกระแสได้ กรณีเป็นมอเตอร์ 2 เฟสจะมีสายที่ใช้ต่อกับวงจร 4 สาย

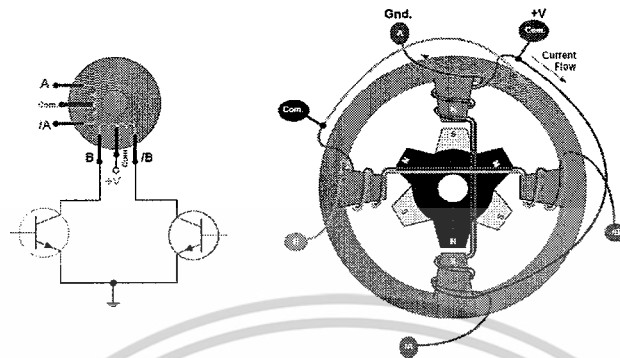


รูปที่ 2.19 สัญลักษณ์ โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบไบโพลาร์ 2 เฟส

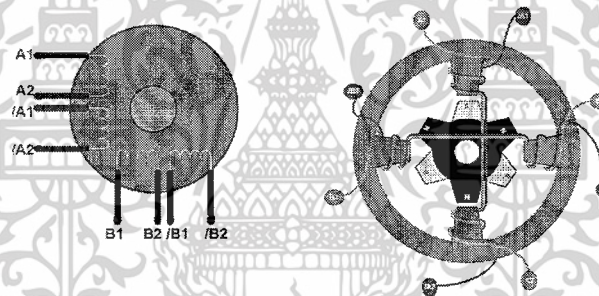
2. แบบยูนิโพลาร์ (Unipolar) ขดลวดที่สเตเตอร์แต่ละขดจะมีจุดร่วม การพันขดลวดจะพันในแบบ Bifilar การต่อเข้ากับวงจรขับเคลื่อนจะใช้ปลายของขดลวดแต่ละด้านต่อเข้ากับวงจรขับและใช้จุดร่วมต่อเข้ากับขั้วบวกของแหล่งจ่ายไฟ การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟให้ไหลจากร่วมลงกราวด์มาครบวงจรที่ปลายด้านหนึ่งของขดลวดการเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ขดเดียวกันนี้ก็ทำได้โดยเปลี่ยนการจ่ายกระแสไฟจากขดหนึ่งไปยังอีกขดหนึ่งของขดลวดที่พันอยู่บนสเตเตอร์ขด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดียวกัน ดังนั้นวงจรขั้วจึงเป็นวงจรสวิตช์เพื่อให้จ่ายกระแสไฟที่ผ่านขดลวดครบวงจรเท่านั้นกรณีเป็นมอเตอร์ 2 เฟสจะมีสายที่ใช้ต่อเข้ากับวงจร 5 หรือ 6 สาย

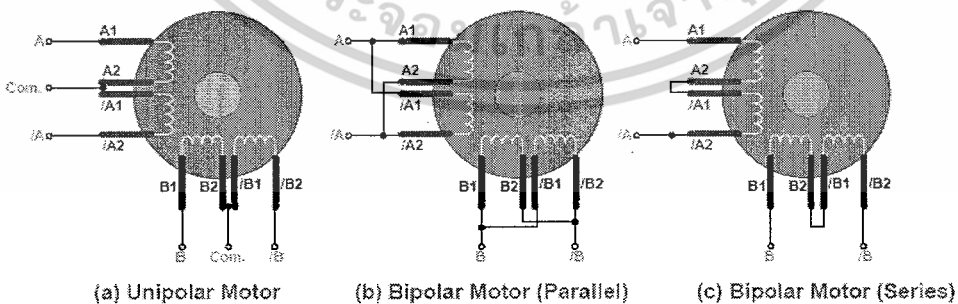


รูปที่ 2.20 สัญลักษณ์ โครงสร้าง และวงจรขั้วที่ใช้กับมอเตอร์แบบยูนิโพลาร์ 2 เฟส สเต็ปเปอร์มอเตอร์ 2 เฟสที่มีการพันลวดแบบไบโพลาร์และแยกปลายแต่ละขดออกจากกันดังนั้นสามารถเลือกต่อเป็นแบบต่างๆได้ดังนี้



รูปที่ 2.21 สัญลักษณ์ โครงสร้างมอเตอร์ 2 เฟสที่พันลวดแบบไบโพลาร์และแยกปลายของแต่ละขด

ตัวอย่างการต่อสาย



รูปที่ 2.22 สัญลักษณ์การต่อสายแบบต่างๆ

มอเตอร์ที่ผลิตออกมาจำหน่ายและมีการพันลวดแบบไบโพลาร์บางรุ่นให้ผู้ใช้เลือกต่อเองเป็นแบบต่างๆแต่บางรุ่นก็ต่อภายในมาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การกระตุ้นและการควบคุมการหมุนของ Stepping Motor

ในการควบคุมการทำงานของ Stepping Motor สามารถแบ่งได้ออกเป็น 3 รูปแบบ คือ

- ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)
- ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส
- ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

2.2.3.1 ควบคุมแบบ Full Step 1 เฟส หรือแบบเวฟ (wave)

ในการควบคุมการหมุนของ Stepping นั้นเราจะต้องกระตุ้นให้มอเตอร์หมุนไปแต่ละ Step โดยจ่ายกระแสไฟฟ้าให้กับ Stepping ทีละเฟสตามลำดับ หลักการคือเริ่มจากจ่ายกระแสให้กับขดลวด Stator เฟสที่ 1 จากนั้นกระตุ้นเฟสที่ 2 และ เฟสที่ 3 ไปเรื่อยๆ ตามลำดับ จากนั้นก็วนกลับมาที่ขดลวด Stator เฟสที่ 1 อีกครั้งและวน Loop ไปเรื่อยๆ ก็จะทำให้ Stepping Motor หมุนและในทางกลับกันถ้าต้องการให้ Stepping Motor หมุนกลับทางก็ต้องกระตุ้นขดลวด Stator เฟส 4 เฟส 3 เฟส 2 และ เฟส 1 ตามลำดับ สามารถเขียนขั้นตอนการทำงานเป็นตารางออกมาได้ดังนี้

| Step | เฟส 4 | เฟส 3 | เฟส 2 | เฟส 1 |
|--------|---------------------|-------|-------|-------|
| Step 1 | ON | OFF | OFF | OFF |
| Step 2 | OFF | ON | OFF | OFF |
| Step 3 | OFF | OFF | ON | OFF |
| Step 4 | OFF | OFF | OFF | ON |
| Step 5 | ย้อนกลับ Step ที่ 1 | | | |
| | | | | |

รูปที่ 2.23 แสดงการกระตุ้นหรือควบคุมแบบ Full Step 1 เฟสหรือแบบเวฟ (wave)

2.2.3.2 ควบคุมแบบ Full Step 2 เฟส หรือแบบ 2 เฟส

ในการควบคุม Stepping Motor แบบ 2 เฟสนั้น เราจะต้องจ่ายกระแสไฟฟ้าเพื่อกระตุ้นขดลวดของมอเตอร์ทีละ 2 เฟส ในเวลาเดียวกันและเรียงกันไปตามลำดับซึ่งได้แสดงดังตารางด้านล่าง โดย Stepping Motor จะหมุนเหมือนกับการควบคุมแบบเวฟแต่การควบคุมแบบ 2 เฟสจะให้แรงบิดที่สูงกว่าแบบเวฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Step | เฟส 4 | เฟส 3 | เฟส 2 | เฟส 1 |
|--------|---------------------|-------|-------|-------|
| Step 1 | ON | ON | OFF | OFF |
| Step 2 | OFF | ON | ON | OFF |
| Step 3 | OFF | OFF | ON | ON |
| Step 4 | ON | OFF | OFF | ON |
| Step 5 | ย้อนกลับ Step ที่ 1 | | | |
| | | | | |

รูปที่ 2.24 แสดงการกระตุ้นหรือควบคุมแบบ Full Step 2 เฟส หรือ แบบ 2 เฟส

2.2.3.3 ควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

การควบคุม Stepping Motor แบบครึ่งสเต็ปจะทำให้เราสามารถเพิ่มความละเอียดในการควบคุมการหมุนของ Stepping Motor ได้แม่นยำมากขึ้นซึ่งเป็นการผสมผสานระหว่างการควบคุมแบบเวฟ และแบบ Full Step 2 เฟสเข้าด้วยกัน ลักษณะการจ่ายกระแสไฟ เพื่อกระตุ้นขดลวดจะแสดงดังตารางด้านล่าง

| Step | เฟส 1 | เฟส 2 | เฟส 3 | เฟส 4 |
|--------|---------------------|-------|-------|-------|
| Step 1 | ON | ON | OFF | OFF |
| Step 2 | ON | ON | OFF | OFF |
| Step 3 | OFF | ON | OFF | OFF |
| Step 4 | OFF | ON | ON | OFF |
| Step 5 | OFF | OFF | ON | OFF |
| Step 6 | OFF | OFF | ON | ON |
| Step 7 | OFF | OFF | OFF | ON |
| Step 8 | ON | OFF | OFF | ON |
| Step 9 | ย้อนกลับ Step ที่ 1 | | | |
| | | | | |

รูปที่ 2.25 แสดงการกระตุ้นหรือควบคุมแบบ Half Step หรือแบบครึ่งสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ET-XPORT V3

ET-XPORT V3 เป็นชุดอุปกรณ์ที่ใช้สำหรับการแปลงข้อมูลระหว่าง Protocol ของ TCP/IP และ RS232 โดยด้านหนึ่งจะใช้การติดต่อสื่อสารด้วยพอร์ตสื่อสารอนุกรมแบบ RS232 ส่วนอีกด้านหนึ่งจะสื่อสารผ่านระบบ LAN แบบ Ethernet ด้วย TCP/IP Protocol โดยที่ ET-XPORT V3 จะทำหน้าที่เป็น Data Protocol Converter สำหรับแปลงข้อมูลของการสื่อสารระหว่างอุปกรณ์ที่เป็น RS232 กับอุปกรณ์ที่เป็น Ethernet LAN ซึ่งจะช่วยลดความยุ่งยากสำหรับผู้ใช้งานที่ต้องการติดต่อสื่อสารข้อมูลกับระบบเครือข่ายของ Ethernet LAN โดยผู้ใช้งานไม่จำเป็นต้องเสียเวลาไปศึกษารายละเอียดของการรับส่งข้อมูลผ่าน Protocol แบบ TCP/IP มากมายนักก็สามารถทำการเชื่อมต่อสื่อสารกับเครือข่าย Ethernet LAN ด้วย TCP/IP Protocol ได้แล้ว

โดย ET-XPORT V3 มีความเหมาะสมในการนำไปประยุกต์ใช้งานที่เกี่ยวข้องกับการสื่อสารผ่าน TCP/IP Protocol โดยเฉพาะระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์อื่นๆในเครือข่าย โดยการเชื่อมต่อ ET-XPORT V3 ด้านที่เป็นพอร์ตสื่อสารอนุกรม RS232 เข้ากับระบบไมโครคอนโทรลเลอร์ ส่วนด้าน Ethernet Port (RJ45) ก็เชื่อมต่อเข้ากับเครือข่าย จากนั้นก็จะทำให้ระบบไมโครคอนโทรลเลอร์สามารถติดต่อสื่อสารข้อมูลกับอุปกรณ์อื่นๆที่อยู่ภายในเครือข่ายได้แล้ว ซึ่งรูปแบบและข้อกำหนดในการสื่อสารก็สามารถเลือกกำหนดได้มากมายหลายรูปแบบ

ซึ่งนอกจากจะนำ ET-XPORT V3 ไปประยุกต์ใช้งานในด้านของการสื่อสารกับ TCP/IP ร่วมกับระบบไมโครคอนโทรลเลอร์หรืออุปกรณ์อื่นๆที่เป็น RS232 แล้วยังสามารถนำไปประยุกต์ใช้งานในด้านอื่นๆ โดยไม่จำเป็นต้องเชื่อมตัวกับไมโครคอนโทรลเลอร์ก็ได้ เช่น ใช้เป็นระบบเตือนภัยหรือตรวจจับสัญญาณ Sensor ต่างพร้อมทั้งส่งผลการตรวจจับผ่านเครือข่าย LAN ไปยัง Server ในรูปแบบของ Alarm Email ก็ได้ โดยการกำหนดให้ ET-XPORT V3 ตรวจจับสัญญาณ Input

2.3.1 คุณสมบัติของ ET-XPORT V3

1. มีจุดต่อพอร์ตสื่อสารอนุกรม (Serial Port) แบบ RS232 โดยใช้ขั้วต่อแบบ DB9 ตัวเมีย สำหรับเชื่อมต่อกับอุปกรณ์ที่ใช้การรับส่งข้อมูลด้วย RS232 เช่น คอมพิวเตอร์ หรือ ไมโครคอนโทรลเลอร์ต่างๆ
2. มีจุดต่อพอร์ต Ethernet LAN แบบ Modula RJ45 จำนวน 1 ช่อง สำหรับเชื่อมต่อกับเครือข่ายระบบ LAN แบบ Ethernet พร้อม LED แสดงสถานะของการสื่อสารข้อมูลของระบบ Ethernet LAN
3. มีจุดต่อสัญญาณ Input แบบ OPTO-ISOLATE สำหรับต่อ Sensor จำนวน 3 Input
4. มี LED แสดงสถานะ การทำงานดังนี้
 - LED สีแดง ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยงวงจร (Power)
 - LED สีเขียว ใช้แสดงสถานะของ Input ซึ่งมี 3 ชุด (In0, In1 และ In2)
5. สามารถตั้งค่า Configuration การทำงานผ่านเว็บ Browser หรือ Telnet หรือ RS232 ได้
6. มีระบบป้องกันความปลอดภัยด้วย Password Protection
7. มีระบบป้องกันความปลอดภัยของการสื่อสารข้อมูลด้วยการเข้ารหัส Encryption

แบบ Rijndael algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. รองรับการสื่อสาร TCP/IP Protocol หลายรูปแบบ ตัวอย่างเช่น

- ARP, UDP, TCP, ICMP, Telnet, TFTP, Auto IP, DHCP, HTTP, SNMP สำหรับ Network
- TCP, UDP, Telnet สำหรับการรับส่งผ่าน Serial Port (RS232)
- User Datagram Protocol (UDP)
- SMTP สำหรับการรับส่ง E-mail

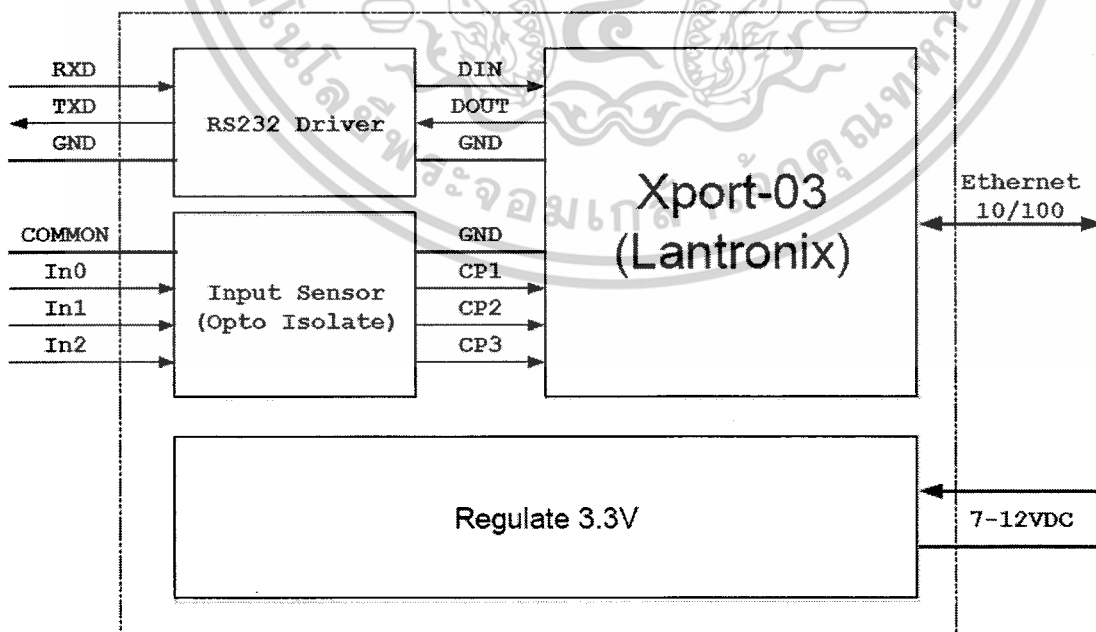
9. มี MAC Address หรือ Ethernet Address ที่เป็นมาตรฐานสากลประจำอยู่ในชุด

10. ใช้ Power Supply 3.3VDC

2.3.2 ข้อจำกัดด้านการใช้งานของ ET-XPORT V3

1. ในการสื่อสารด้าน RS232 ไม่รองรับการตรวจสอบความพร้อมแบบ Hardware Handshake ด้วยสัญญาณ CTS/RTS/DTR/DSR ซึ่งถ้าต้องการใช้การ Handshake ต้องใช้การ Handshake ด้วยเทคนิคทางด้าน Software เช่น XON/XOFF แทน
2. ในการสื่อสารด้าน RS232 ในโหมด Modem Emulate นั้นการสื่อสารกับ Modem จะไม่รองรับสัญญาณ Modem Control ซึ่งจะมีเฉพาะ RXD, TXD และ GND เท่านั้น
3. สัญญาณ GPIO ทั้ง 3 สัญญาณ สามารถกำหนดหน้าที่การใช้งานเป็น Input ได้อย่างเดียว
4. ไม่สามารถใช้เงื่อนไขการทำงานที่เกี่ยวกับ Hardware Handshake และการกระตุ้นการทำงานด้วยสัญญาณทาง Hardware ต่างๆ เช่น การกำหนดให้เกิดการ Auto Start ด้วยสัญญาณ DTR ซึ่งจะไม่สามารถใช้ได้กับระบบฮาร์ดแวร์ของ ET-XPORT V3

2.3.3 โครงสร้างของ ET-XPORT V3



เอกสารนี้เป็นเอกสารที่ลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การกำหนดค่าการใช้งาน ET-XPORT V3 กับระบบเครือข่าย Ethernet LAN

สำหรับการนำ ET-XPORT V3 ไปประยุกต์ใช้งานนั้นสามารถทำได้หลายรูปแบบขึ้นอยู่กับผู้ใช้จะนำไปประยุกต์ดัดแปลง แต่ในที่นี้จะขอเน้นเฉพาะในส่วนของการใช้งานในส่วนของการเชื่อมต่อสัญญาณเพื่อใช้ในการรับส่งข้อมูลระหว่างอุปกรณ์ที่เป็น RS232 และอุปกรณ์ที่ต่อกันอยู่ในเครือข่ายของ Ethernet LAN เป็นหลัก

โดยในโหมดของการเชื่อมต่อสัญญาณเพื่อรับส่งข้อมูลระหว่าง RS232 กับเครือข่ายของ Ethernet LAN นั้น ET-XPORT V3 จะสามารถรองรับการเชื่อมต่อกับอุปกรณ์ในเครือข่าย โดยใช้ TCP/IP Protocol ได้ 3 รูปแบบด้วยกัน คือ TCP, UDP และ Telnet ซึ่งผู้ใช้สามารถเลือกกำหนดรูปแบบของ Protocol ที่จะใช้ในการเชื่อมต่อกับระบบเครือข่ายได้ตามต้องการ โดยก่อนที่จะนำ ET-XPORT V3 ไปเชื่อมต่อเข้ากับระบบเครือข่ายได้นั้นผู้ใช้จะต้องทำการกำหนดค่าที่จำเป็นต่อใช้งานในระบบเครือข่ายให้กับ ET-XPORT V3 ให้เรียบร้อยเสียก่อนโดยในระบบเครือข่ายของ Ethernet LAN นั้น มีความจำเป็นต้องกำหนดค่าต่างๆ ให้กับอุปกรณ์ที่จะนำมาเชื่อมต่อเพื่อใช้งานร่วมกันในระบบเครือข่ายดังนี้

- ค่า **Hardware Address** หรือ **MAC Address** ซึ่งมีค่าเป็นเลข HEX ขนาด 2 หลัก ระหว่าง 00-FF จำนวน 6 ชุด โดยในส่วนนี้ อุปกรณ์แต่ละตัวที่จะนำมาเชื่อมต่อกันในเครือข่ายต้องมีค่ารหัสที่ไม่ซ้ำกัน แต่สำหรับ ET-XPORT V3 นั้นค่ารหัสนี้จะถูกกำหนดไว้ที่ตัว ET-XPORT V3 แต่ละตัวอยู่แล้ว ดังนั้นในการอ้างอิงค่ารหัส MAC Address ของ ET-XPORT V3 นั้นผู้ใช้สามารถตรวจสอบได้จากรหัสตัวเลขที่กล่องของ ET-XPORT V3 ได้โดยตรง โดย ET-XPORT V3 ทุกๆตัวจะมี MAC Address ของ 3 ชุดแรกเหมือนกัน คือ 00-20-4A ส่วน 3 ชุดสุดท้ายจะมีความแตกต่างกัน เช่น 00-20-4A-86-AB-AB

- ค่า **IP Address** ซึ่งเป็นหมายเลข IP Address ที่จะใช้ในการสื่อสารข้อมูลกับอุปกรณ์อื่นๆในเครือข่ายโดยใช้ TCP/IP Protocol มีค่าเป็นตัวเลขฐานสิบขนาด 3 หลัก ระหว่าง 0-255 จำนวน 4 ชุด ซึ่งค่าของ IP Address นี้ผู้ใช้ต้องกำหนดให้กับ ET-XPORT V3 เอง และต้องมีค่า IP Address ที่ไม่ซ้ำกับอุปกรณ์อื่นๆที่อยู่ในเครือข่ายด้วย เช่น 192.168.1.50 เป็นต้น

- ค่า **Port Numbers** ซึ่งเป็นค่า Port ที่ใช้อ้างอิงในการรับส่งข้อมูลของ TCP และ UDP โดยสามารถกำหนดได้ตามต้องการ แต่ต้องไม่ตรงกับหมายเลข Port ที่สงวนไว้ใช้ในระบบ โดย ET-XPORT V3 นั้นจะมีการสงวนค่าหมายเลข Port ไว้ใช้งานเป็นการเฉพาะ (ห้ามใช้งานโดย User) ดังต่อไปนี้

- 1-1024 จะถูกสงวนไว้ใช้เกี่ยวกับการติดต่อสื่อสารทั่วไป
- 9999 จะถูกสงวนไว้เพื่อใช้สำหรับการสั่ง Setup ค่า Configuration ให้กับ ET-XPORT V3 ผ่านระบบเครือข่ายด้วย Telnet Protocol
- 14000-14009 จะถูกสงวนไว้สำหรับใช้งานในการสื่อสารร่วมกับโปรแกรม Redirector
- 30704 จะถูกสงวนไว้ใช้ในระบอบด้านการรักษาความปลอดภัย (77F0H)
- 30718 จะถูกสงวนไว้ใช้ในระบอบด้านการรักษาความปลอดภัย (77FEH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 โปรแกรมที่ใช้ในการพัฒนาของ ET-XPORT V3

จะเห็นได้ว่าก่อนที่จะเริ่มต้นใช้งาน ET-XPORT V3 ได้นั้นในอันดับแรกจะต้องทำการกำหนดค่า Configuration ต่างๆให้กับตัวเครื่องให้เรียบร้อยเสียก่อนไม่ว่าจะเป็นค่า IP Address รวมไปถึงโหมดการทำงานของเครื่องและค่าตัวเลือกต่างๆ เช่น ค่าความเร็วของ Baud rate ของการสื่อสารด้าน RS232 และลักษณะ Protocol ที่จะนำมาใช้ในการรับส่งข้อมูลกับเครือข่ายของ Ethernet LAN ว่าจะใช้แบบ Telnet หรือ TCP หรือ UDP และจะให้ทำงานแบบเป็นฝ่ายรับหรือจะให้เป็นฝ่ายร้องขอการเชื่อมต่อกับ TCP/IP เป็นต้น ซึ่งการกำหนดค่าการทำงานต่างๆเหล่านี้จะใช้โปรแกรมสำเร็จรูปของ “Lantronix” ซึ่งเป็นผู้ผลิตโมดูล XPORT จัดทำไว้ให้ใช้งานโดยโปรแกรม Utility ที่จะใช้สำหรับกำหนดการทำงานของ ET-XPORT V3 คือ โปรแกรม ”DeviceInstaller” ดังนั้น ในอันดับแรกผู้ใช้จะต้องทำการติดตั้งโปรแกรม “DeviceInstaller” ไว้ในเครื่องคอมพิวเตอร์ของผู้ใช้ให้เรียบร้อยเสียก่อน แต่เนื่องจากว่าโปรแกรม “DeviceInstaller” นั้นต้องทำงานร่วมกับทรัพยากรบางส่วน of โปรแกรม Microsoft Dot Net Framework V1.1.4322 ด้วยและในกรณีที่ทางผู้ที่มีความประสงค์ที่จะทำการตั้ง Setup ค่าการทำงานให้กับ ET-XPORT V3 ผ่านทาง Web Server ด้วย ก็จะต้องทำการติดตั้งโปรแกรม JAVA J2SE ไว้ด้วย ดังนั้นผู้ใช้จึงจำเป็นต้องทำการติดตั้งโปรแกรมทั้ง 3 ชุดนี้ให้กับเครื่องคอมพิวเตอร์ PC ให้เรียบร้อย ตามลำดับคือ

1. โปรแกรม “Microsoft Dot Net Framework Version 1.1.4322”
2. โปรแกรม JAVA J2SE (ในกรณีที่ต้องการใช้การ Setup ด้วย Web Manager)
3. โปรแกรม “DeviceInstaller Version 3.6.0.6”

บทที่ 3

การคำนวณและการสร้างวงจร

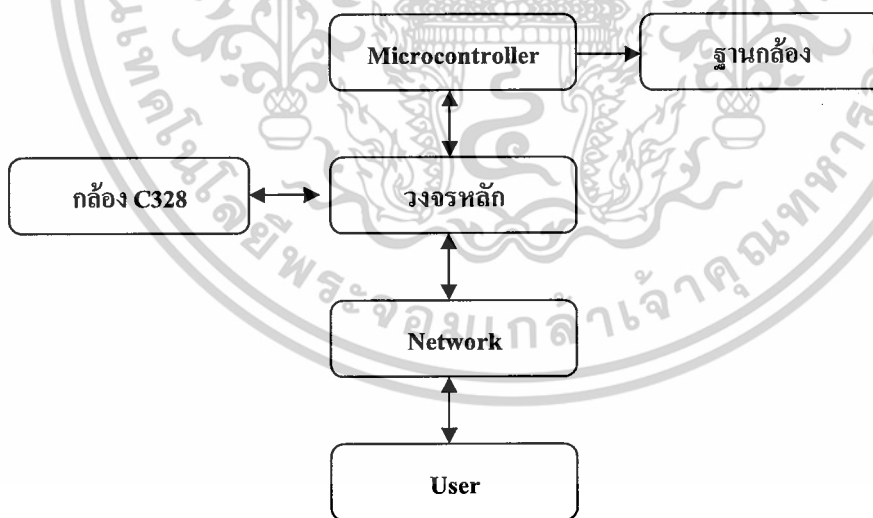
หลังจากที่ทราบถึงทฤษฎีหรือหลักการในบทที่ 2 แล้ว เราก็จะมาทำการคำนวณและสร้างวงจรในส่วนของฮาร์ดแวร์ของระบบ ประกอบไปด้วยวงจรควบคุมการรับ-ส่งข้อมูลผ่านอินเทอร์เน็ตซึ่งใช้ตัว XPORTทำหน้าที่เป็น Embedded Web server โดยการแสดงผลและการควบคุมนั้นผู้ใช้งานจะใช้งานผ่านทางหน้าเว็บเพจทั่วไป โดยอาศัยอุปกรณ์ควบคุมการเชื่อมต่อระบบเครือข่าย (Ethernet Controller) รวมทั้งอธิบายถึงการควบคุมหน่วยการรับเข้าออกข้อมูล (I/O Port) และกระบวนการรับ-ส่งข้อมูลของฮาร์ดแวร์

และในส่วนไมโครคอนโทรลเลอร์ควบคุมการเคลื่อนไหวของฐานกล้องผ่านทางหน้าเว็บเพจ ขั้นตอนการออกแบบในแต่ละส่วนของกล้องผ่านระบบเครือข่ายอินเทอร์เน็ต (IP CAMERA) ที่ใช้ในการทดลองจริง ซึ่งสามารถแบ่งออกเป็น 2 ส่วนที่สำคัญ คือ

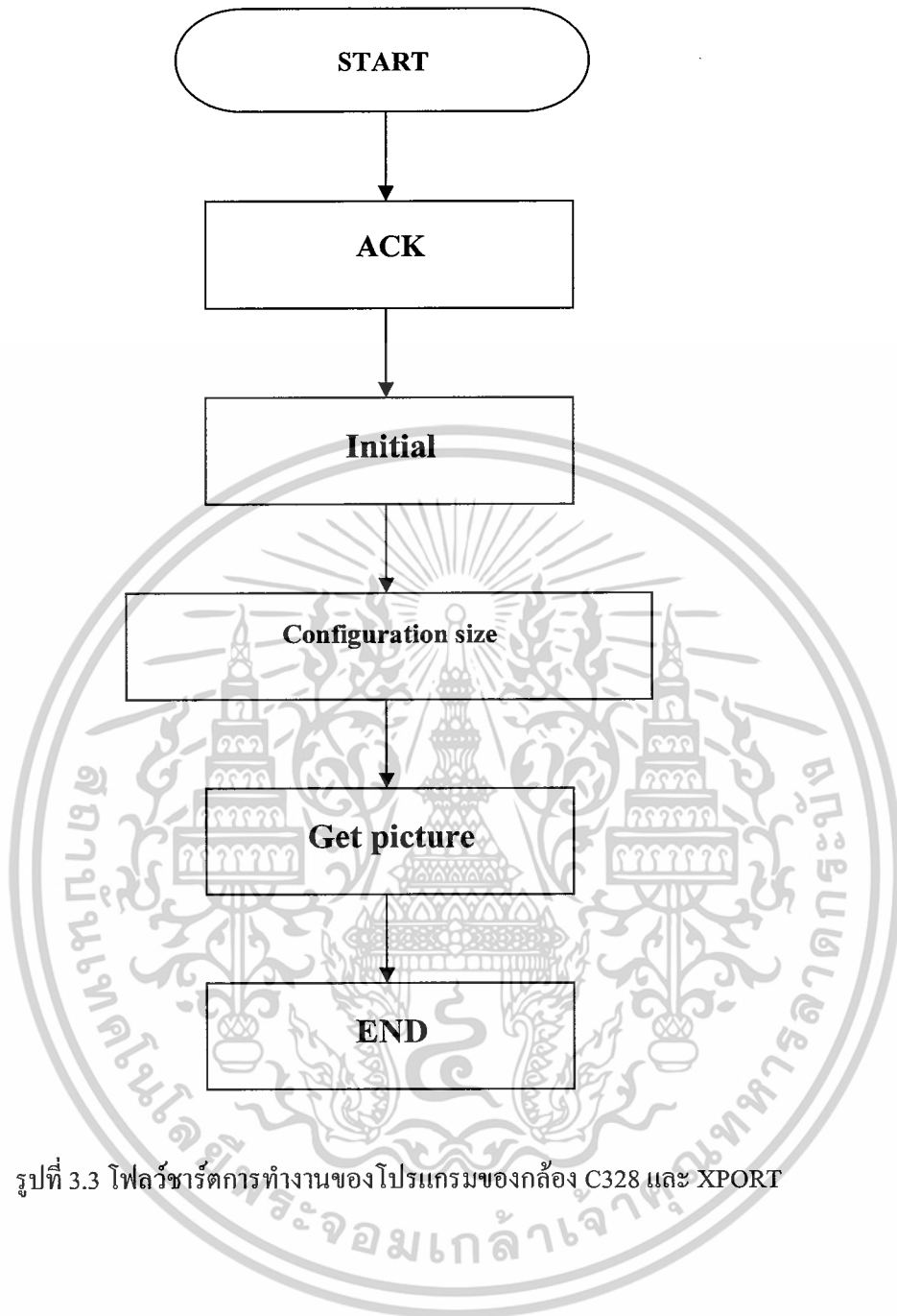
- การแสดงภาพ
- การควบคุมการเคลื่อนไหวของฐานกล้อง

3.1 ส่วนประกอบฮาร์ดแวร์ของระบบ

ส่วนประกอบหลักของฮาร์ดแวร์ทั้งหมดประกอบด้วย วงจรเชื่อมต่อกับกล้อง C328 และตัวควบคุมการเคลื่อนไหวของฐานกล้องซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ รวมทั้งวงจรเชื่อมต่อกับระบบเครือข่าย



รูปที่ 3.1 โฟลว์ชาร์แสดงส่วนประกอบหลักของฮาร์ดแวร์ทั้งหมด

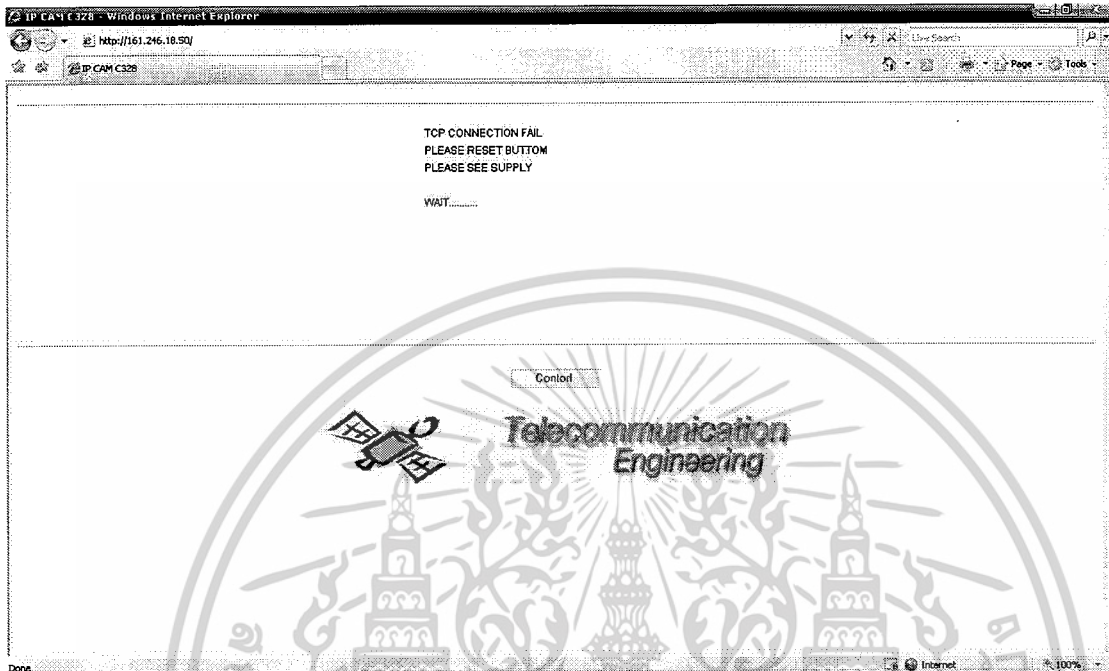


รูปที่ 3.3 โฟลว์ชาร์ตการทำงานของโปรแกรมของกล้อง C328 และ XPORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนของโปรแกรมการแสดงผลผ่านหน้าจอกอมพิวเตอร์

3.3.1 การออกแบบหน้าการแสดงผลของภาพ



รูปที่ 3.4 หน้าจอการแสดงผลภาพ

รายละเอียดของหน้าจอดังนี้

1. ช่องใส่หมายเลข IP Address
2. หน้าจอแสดงภาพจากกล้อง
3. ปุ่มควบคุมการเคลื่อนที่ของกล้อง

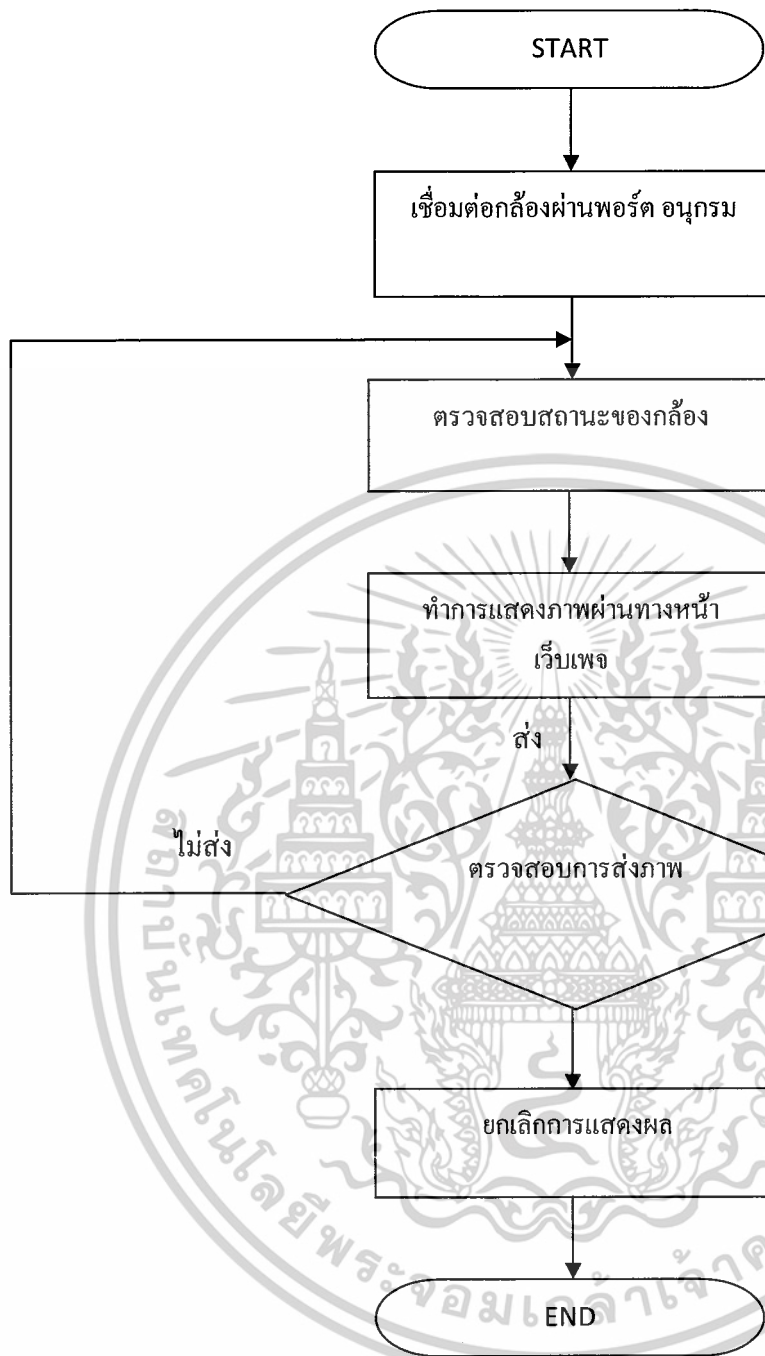
3.3.2 ส่วนของโปรแกรมควบคุมการทำงานทั้งหมด

ในส่วนของโปรแกรมนั้น เราสามารถแบ่งออกได้เป็น 2 ส่วน

1. ส่วนของโปรแกรมดึงภาพและแสดงผล
2. ส่วนของโปรแกรมควบคุมการเคลื่อนที่ของกล้อง

ในการดึงภาพนั้นเราได้ใช้ LAN พอร์ต ในการดึงสัญญาณภาพจากกล้องเพื่อนำมาแสดงบนหน้าจอซึ่งสามารถแสดงไฟล์วีราร์ดการทำงานของโปรแกรมแสดงผลภาพได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

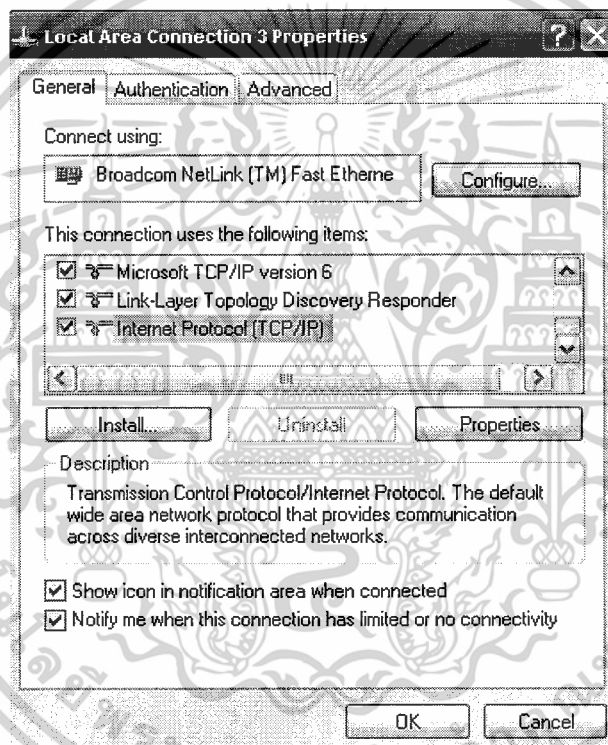


รูปที่ 3.5 โฟลว์ชาร์ตการทำงานของโปรแกรมดึงภาพและแสดงภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนการโหลดโปรแกรมเข้า XPORT

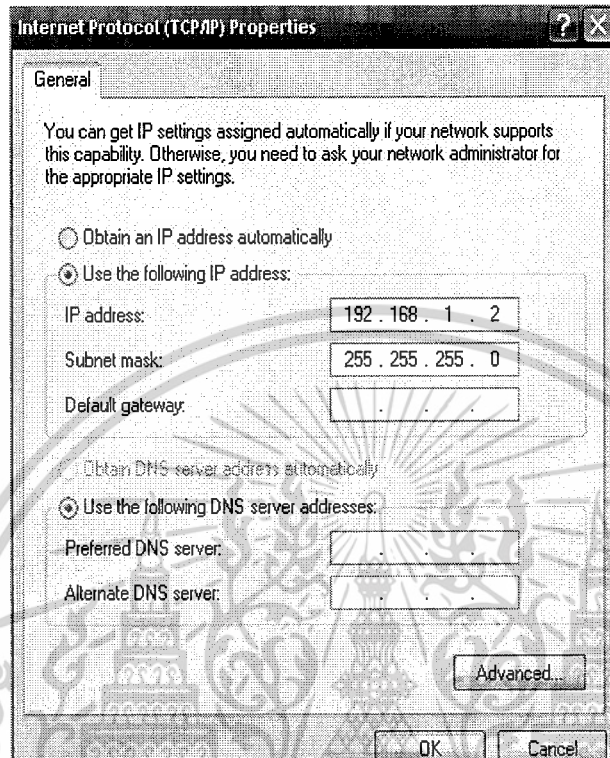
ก่อนอื่นเริ่มจากให้ป้อนไฟเข้าวงจรที่เราประกอบเสร็จแล้ว จากนั้นให้นำสาย LAN ที่เป็นสายไขว้มาทำการเชื่อมต่อระหว่างคอมพิวเตอร์กับ XPORT จากนั้นทำการติดตั้งโปรแกรมที่ใช้ในการ Flash ให้กับ XPORT โดยเราจะใช้โปรแกรม DeviceInstaller โดยสามารถ download ได้จาก http://txfaq.custhelp.com/cgibin/txfaq.cfg/php/enduser/std_adp.php?p_faaid=644 เมื่อทำการติดตั้งเสร็จแล้วให้ทำการ FIX IP ที่จะใช้ในเครื่องคอมพิวเตอร์ของเรา เนื่องจากว่าตัว XPORT นั้นเรายังไม่ทำการตั้ง DHCP ดังนั้นเราจึงต้องทำการตั้ง IP ที่เครื่องคอมพิวเตอร์ก่อน โดยให้เราคลิกที่ Start->Setting->Network Connection แล้วดับเบิลคลิก จากนั้นให้คลิกขวาที่ Local Area Connection แล้วเลือกแถบ Properties จะปรากฏหน้าต่างดังรูป



รูปที่ 3.6 วิธีการตั้งค่า Internet Protocol (TCP/IP)

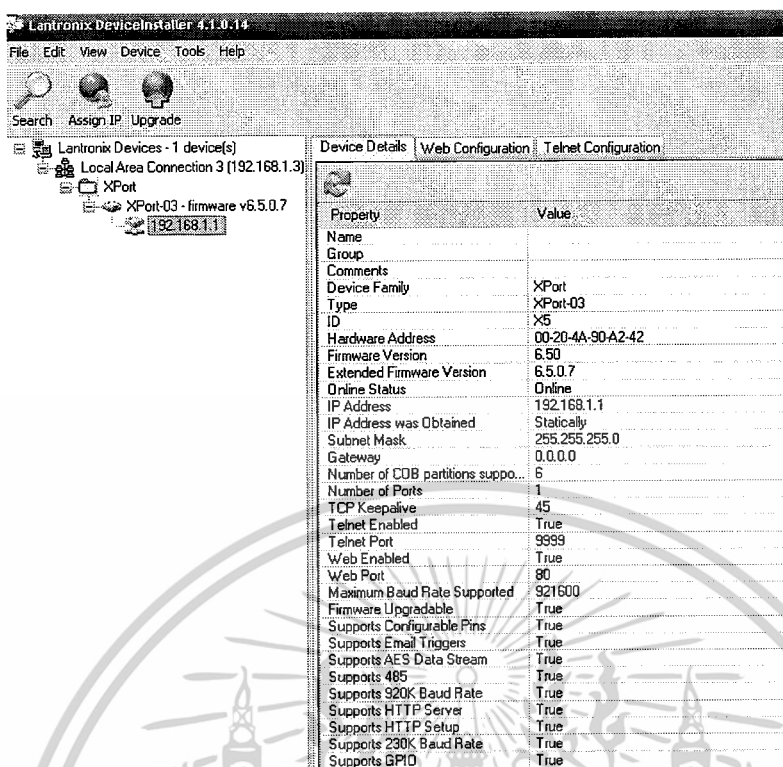
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นดับเบิลคลิกที่ Internet protocol (TCP/IP) จะปรากฏหน้าต่างดังรูป ให้ทำการใส่หมายเลข IP ตามรูป



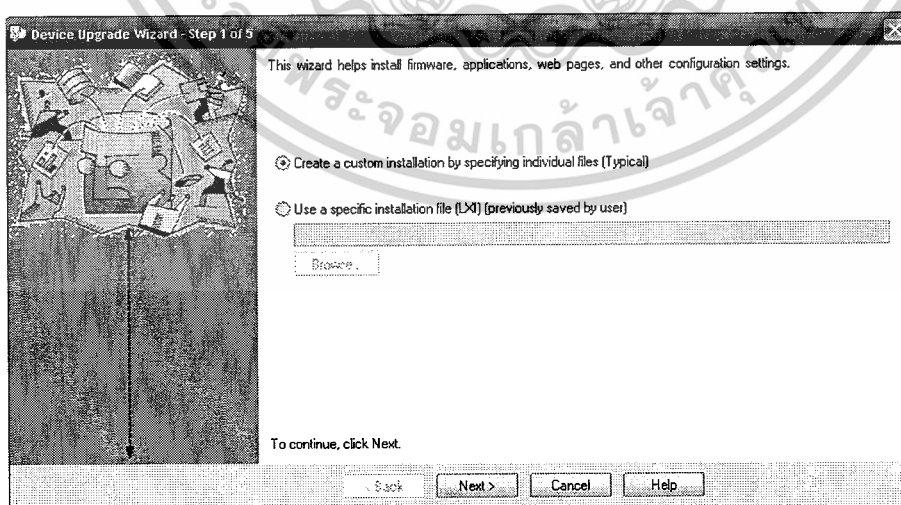
รูปที่ 3.7 วิธีการตั้ง Internet Protocol เพื่อการเชื่อมต่อ

จากนั้นให้ทำการใส่หมายเลข IP ตามรูปด้านบน แล้วก็คลิกที่ OK เป็นอันเสร็จในส่วนของการติดตั้งหมายเลข IP ให้กับคอมพิวเตอร์ คราวนี้เรามาเริ่มการตั้งค่าตัว XPORT กันต่อเลย ให้เราเปิดโปรแกรม DeviceInstaller ขึ้นมา จากนั้นตัวโปรแกรมจะทำการค้นหา XPORT ที่อยู่ในระบบเครือข่าย โดยหากพบตัว XPORT ก็จะรายงานออกมาดังรูปด้านล่าง



รูปที่ 3.8 แสดงโปรแกรม Deviceinstaller ที่ใช้ร่วมกับ XPORT

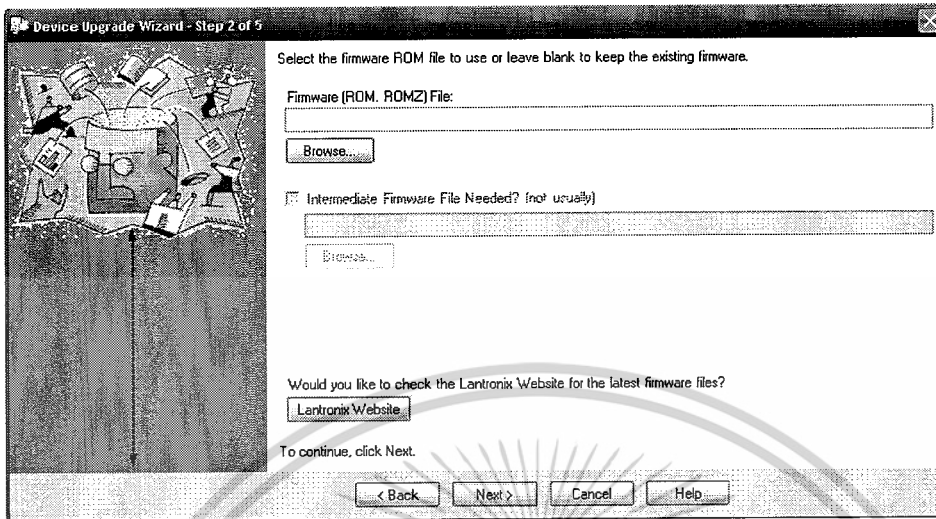
เมื่อหน้าต่างของโปรแกรมขึ้นตรงรูปด้านบนแล้วก็แสดงว่าตอนนี้ตัวโปรแกรมติดต่อกับ XPORT ได้แล้ว ตามรูปด้านบนจะเห็นว่าตอนนี้ XPORT ของเรามี IP เป็น 192.168.1.1 และมี Subnet เป็น 255.255.255.0 (ในส่วนของ การเปลี่ยน IP นั้นสามารถทำได้โดยผ่านทาง TELNET) ต่อมาจะทำการโหลดข้อมูลที่ใช้ร่วมงานกับกล่องลงไป ใน XPORT โดยให้คลิกที่ Upgrade จากนั้นจะปรากฏหน้าต่างด้านล่างขึ้นมา



รูปที่ 3.9 วิธีการโปรแกรมตัว XPORT หน้าที่ 1

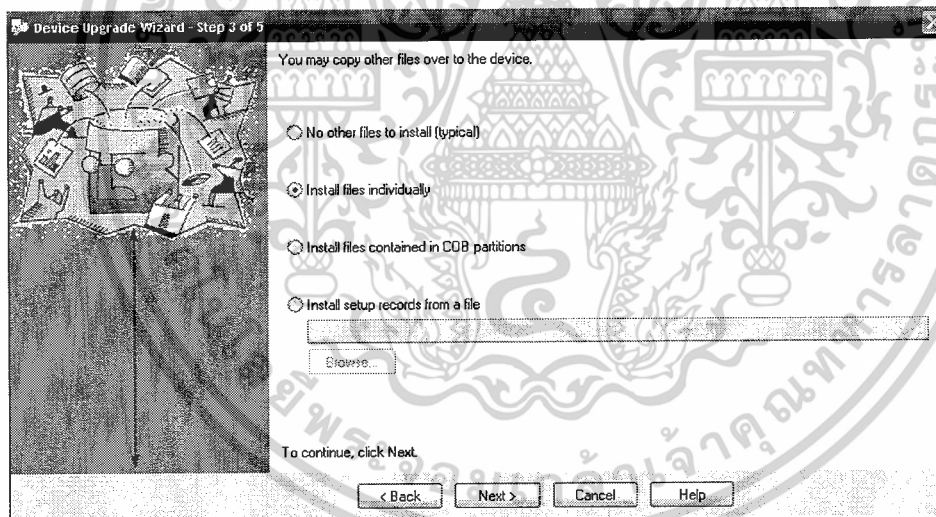
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้เลือกหัวข้อแรกเหมือนตามรูปด้านบน จากนั้นให้คลิกที่ Next



รูปที่ 3.10 วิธีการโปรแกรมตัว XPORT หน้าที 2

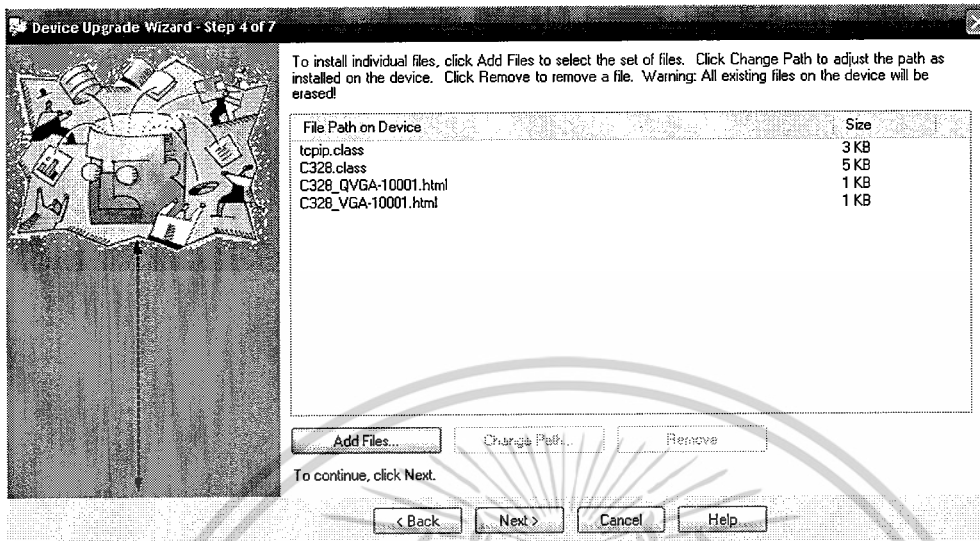
จากนั้นจะปรากฏหน้าต่างดังรูปด้านบน ในหน้าต่างนี้ไม่ต้องไปทำอะไรทั้งสิ้น ให้คลิกที่ Next



รูปที่ 3.11 วิธีการโปรแกรมตัว XPORT หน้าที 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้าต่างนี้ให้คลิกเลือกที่มีข้อความว่า Install files individually จากนั้นให้คลิก Next

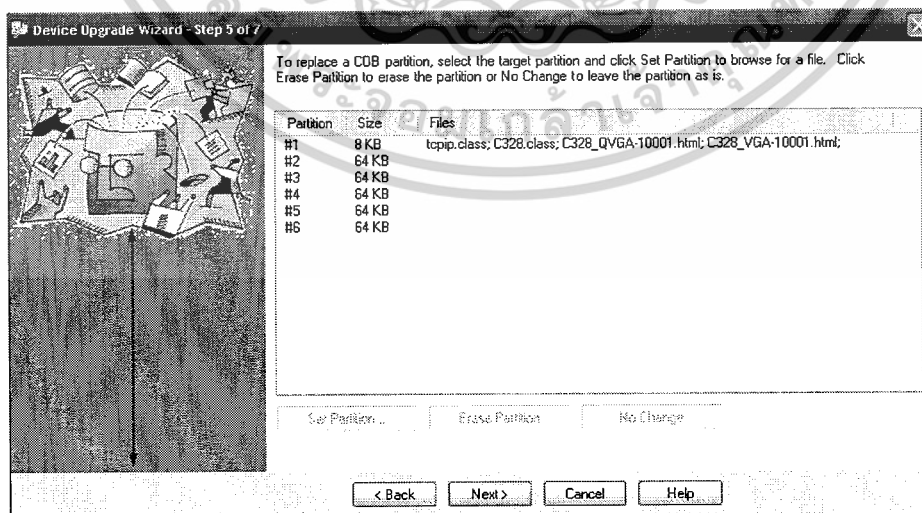


รูปที่ 3.12 วิธีการ โปรแกรมตัว XPORT หน้าที 4

ในหน้าต่างนี้ เริ่มต้นด้วยการคลิกที่ Add files จากนั้นให้หาไฟล์ที่จะมาโปรแกรมให้กับตัว XPORT จากนั้นจะโปรแกรมไฟล์ทั้ง 4 ไฟล์ โดยไฟล์ทั้ง 4 ไฟล์จะประกอบไปด้วย

1. tcpip.class
2. C328.class
3. C328_QVGA-10001.html
4. C328_VGA-10001.html

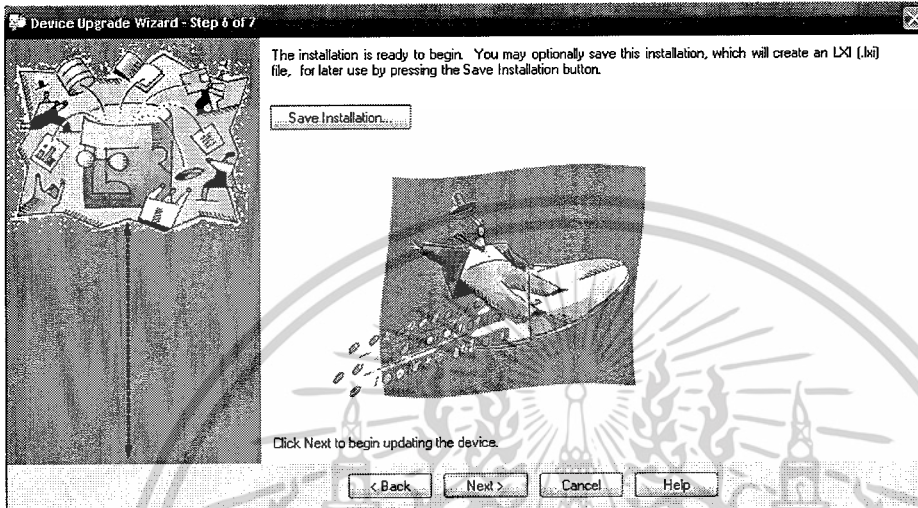
เมื่อทำการ Add files ทั้งหมดครบแล้วให้คลิกที่ Next



รูปที่ 3.13 วิธีการ โปรแกรมตัว XPORT หน้าที 5

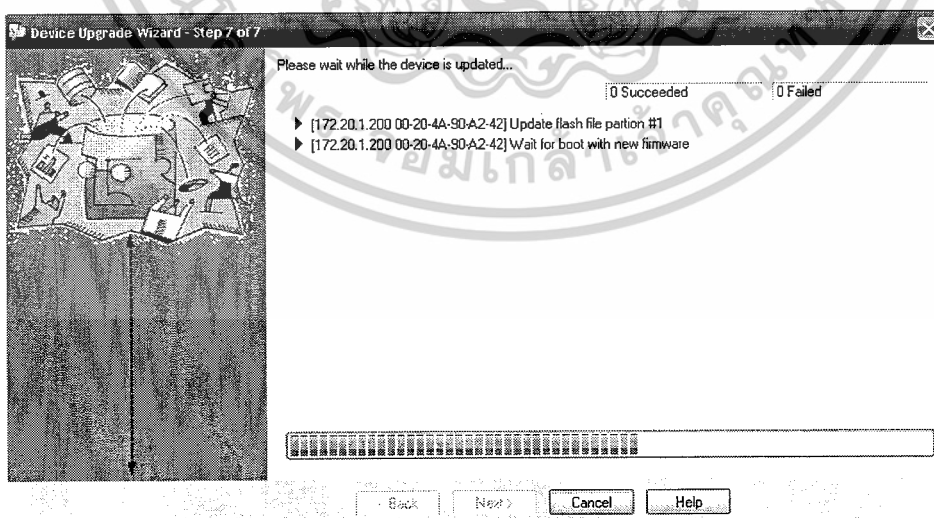
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้าต่างนี้ตัวโปรแกรมจะรายงานการใช้พื้นที่ทั้งหมดของตัวว่าตอนนี้เก็บข้อมูลไว้ที่ Petition ไหนบ้าง และขนาดของไฟล์รวมกันทั้งหมดเป็นเท่าไร ให้สังเกตว่าตัว XPORT นั้นจะมี Petition ทั้งหมด 6 Petition และในแต่ละ Petition มีขนาดเท่ากับ 64KB นั่นก็หมายความว่าสามารถที่จะเขียนโปรแกรมหรือแก้ไขหน้าตาของเว็บเพิ่มเติมได้ จากนั้นให้คลิกที่ Next



รูปที่ 3.14 วิธีการ โปรแกรมตัว XPORT หน้าที 6

ในรูปด้านบนตัวโปรแกรมจะให้เลือกว่าจะทำการเซฟไฟล์ทั้งหมดที่ผ่านมาเป็นไฟล์ .LXI หรือไม่ เพื่อในการโปรแกรมครั้งต่อไปไม่ต้องทำการเลือกไฟล์ทั้ง 4 ไฟล์ เราเพียงแค่ Add files *.LXI เท่านั้นเอง และตรงนี้ตัวโปรแกรมจะให้คลิกที่ Next อีกครั้งเพื่อเป็นการยืนยันการ Flash ข้อมูลทั้งหมดลงไปในตัว XPORT ถ้าทุกอย่างเสร็จเรียบร้อยแล้วก็คลิกที่ Next



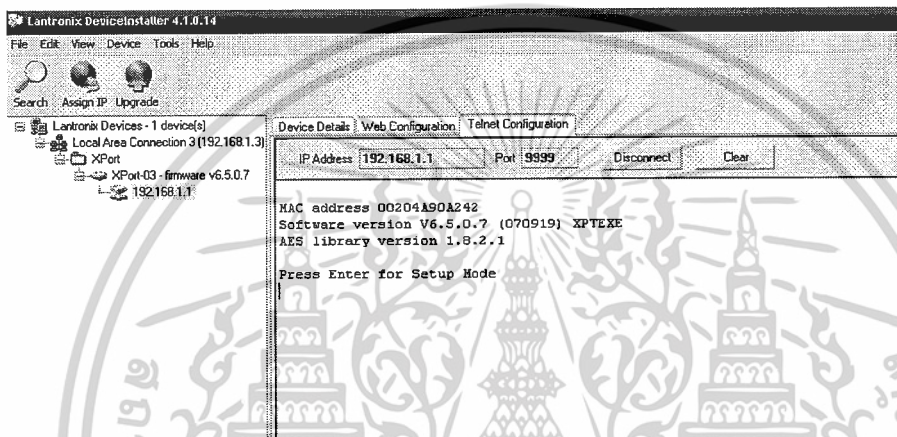
รูปที่ 3.15 วิธีการ โปรแกรมตัว XPORT หน้าที 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปด้านบนนี้ตัวโปรแกรมกำลังทำการ Update Flash files ให้กับตัว XPORT อยู่ให้รอนจนกว่าจะเสร็จ เมื่อเสร็จแล้วตัวโปรแกรมจะมีปุ่มให้เราคลิกคำว่า Close เป็นอันเสร็จสิ้นการโปรแกรม

3.5 การตั้งค่าการทำงานให้กับ XPORT

ในหัวข้อนี้จะเป็นการตั้งค่าในส่วนของ Port UART ในตัว XPORT ให้ตรงกับกรเขียนโปรแกรม โดยการตั้งค่าทั้งหมดนั้นสามารถกระทำได้ผ่านทาง TELNET โดยการใช้นั้นให้เลือกที่หัวข้อ Telnet Configuration จากนั้นให้คลิกที่ Connect จากนั้นตัว XPORT จะถามหารหัสผ่านที่ใช้ในการเข้าไปเปลี่ยนแปลงแก้ไขค่าต่างๆ



รูปที่ 3.16 การแสดงการตั้งค่าการใช้งาน

โดยรหัสผ่านที่มาจากโรงงานนั้นคือ 1234 โดยในตอนใต้จะไม่มีหมายเลขหรืออะไรขึ้น จะเป็นการรักษาความปลอดภัยอย่างหนึ่งของการ TELNET เมื่อใส่รหัสผ่านเสร็จก็กด Enter หากรหัสผ่านที่ใส่ถูกต้องจะปรากฏดังรูปด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

- Trigger 2
Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

- Trigger 3
Serial trigger input: disabled
Channel: 1
Match: 00,00
Trigger input1: X
Trigger input2: X
Trigger input3: X
Message :
Priority: L
Min. notification interval: 1 s
Re-notification interval : 0 s

```

```

Change Setup:
0 Server
1 Channel 1
3 E-mail
5 Expert
6 Security
7 Defaults
8 Exit without save
9 Save and exit
Your choice ? |

```

รูปที่ 3.17 การแสดงการตั้งค่าการใช้งาน

ต่อไปจะทำการเปลี่ยนแปลงค่าของ Channel1 ซึ่งในที่นี้ก็คือ Port UART นั้นเอง ให้ใส่หมายเลข 1 ลงไปจากนั้นก็กด Enter จะปรากฏหน้าต่างขึ้นมาดังรูป

```

Change Setup:
0 Server
1 Channel 1
3 E-mail
5 Expert
6 Security
7 Defaults
8 Exit without save
9 Save and exit
Your choice ? 1

Baudrate {9600} ?

```

รูปที่ 3.18 ตั้งค่าผ่าน Port UART 1

ในส่วนของ Baud rate ให้เราเปลี่ยนจาก 9600 เป็น 115200 ครับ โดยให้เราใส่ 115200 ลงไปจากนั้นก็กด Enter คราวนี้ตัวโปรแกรมจะให้เรารับค่า โดยการตั้งค่าทั้งหมดให้เลือกตามรูปด้านล่าง โดยสิ่งที่ต้องเปลี่ยนแปลงอีกอันก็คือต้อง Flushmode จากเดิมเป็น 00 ให้เปลี่ยนเป็น 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Baudrate (9600) ? 115200
I/F Mode (4C) ?
Flow (00) ?
Port No (10001) ? 10001
ConnectMode (C0) ?
Send '+++' in Modem Mode (Y) ?
Show IP addr after 'RING' (Y) ?
Auto increment source port (N) ?
Remote IP Address : (000) .(000) .(000) .(000)
Remote Port (0) ?
DisConnMode (00) ?
FlushMode (00) ? 80
Pack Cntrl (00) ?
DisConnTime (00:00) ? :
SendChar 1 (00) ?
SendChar 2 (00) ?

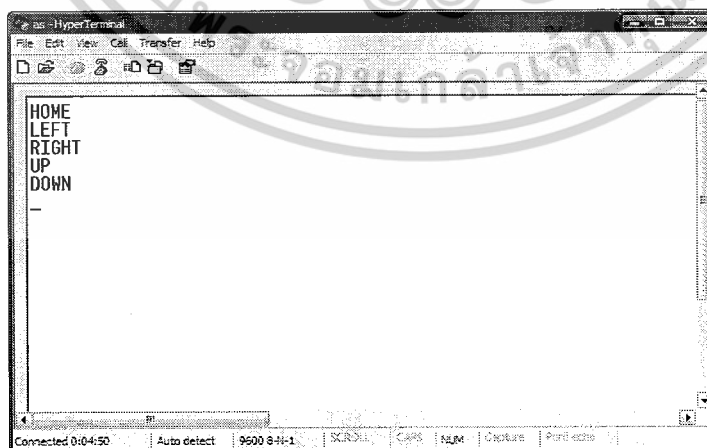
```

รูปที่ 3.19 ตั้งค่าผ่าน Port UART 2

3.6 การควบคุมการเคลื่อนไหวของฐานกลิ้ง

ในการออกแบบการควบคุมในส่วนนี้ จะออกแบบโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานของสเต็ปเปอร์มอเตอร์เพื่อควบคุมการเคลื่อนไหวของฐานกลิ้ง โดยในการทดลองจะออกแบบจะให้ไมโครคอนโทรลเลอร์รับคำสั่งจากคอมพิวเตอร์ก่อน เพื่อให้ง่ายต่อการออกแบบ และเมื่อสามารถควบคุมการเคลื่อนไหวของฐานกลิ้งได้แล้ว จึงจะนำไปเชื่อมต่อกับส่วนของตัว Ethernet Module เพื่อรับคำสั่งงานผ่านทางหน้าเว็บเพจของคอมพิวเตอร์ต่อไป

ในการออกแบบจะใช้ไมโครคอนโทรลเลอร์เบอร์ PIC16F876A โดยจะรับคำสั่งอินพุตจากคอมพิวเตอร์ผ่านทางพอร์ต RS-232 โดยใช้โปรแกรม HyperTerminal เพื่อป้อนคำสั่งให้กับไมโครคอนโทรลเลอร์ดังรูปที่ 3.20



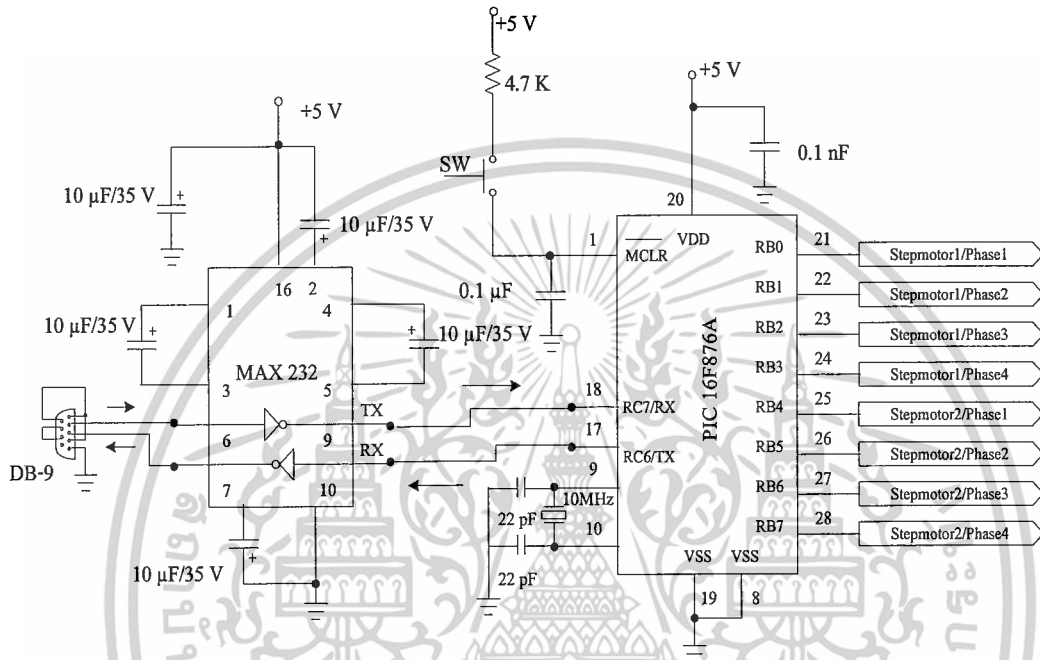
รูปที่ 3.20 การป้อนคำสั่งให้กับไมโครคอนโทรลเลอร์ด้วยโปรแกรม HyperTerminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.1 ส่วนการเชื่อมต่อคอมพิวเตอรืกับไมโครคอนโทรลเลอร์เบอร์ PIC16F876A ผ่าน

พอร์ตอนุกรม RS-232

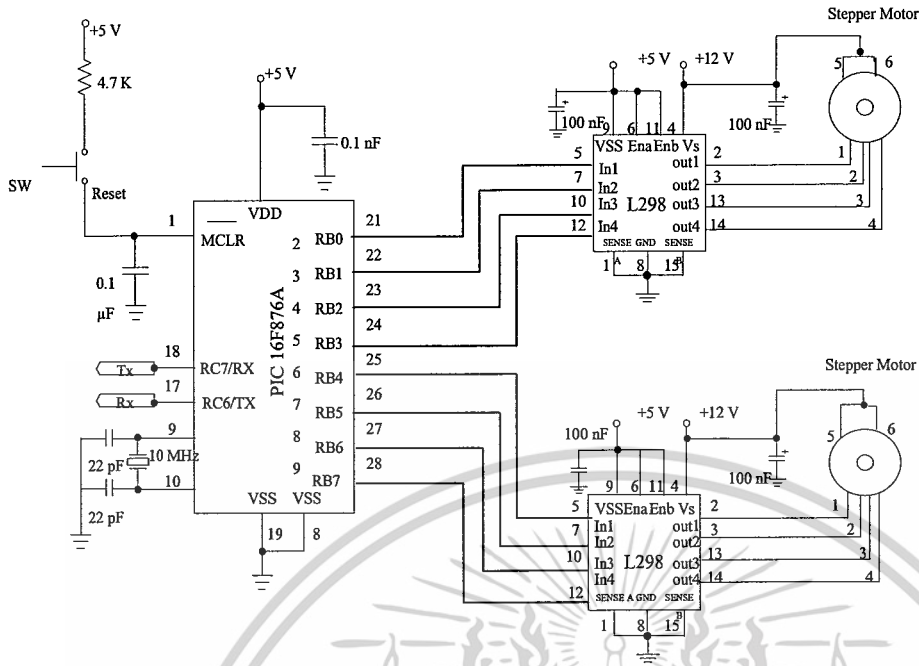
ในส่วนนี้จะเป็นการเชื่อมต่อคอมพิวเตอรืกับไมโครคอนโทรลเลอร์เบอร์ PIC16F876A ผ่านทางพอร์ต RS-232 โดยมีไอซีเบอร์ MAX-232 ทำหน้าที่เป็นตัวแปลงระดับสัญญาณจากพอร์ตอนุกรมของคอมพิวเตอรืให้เข้ากับระดับสัญญาณของไมโครคอนโทรลเลอร์



รูปที่ 3.21 วงจรการเชื่อมต่อคอมพิวเตอรืกับไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรม

3.6.2 ส่วนการเชื่อมต่อไมโครคอนโทรลเลอร์เบอร์ PIC16F876A กับวงจรขับเคลื่อน

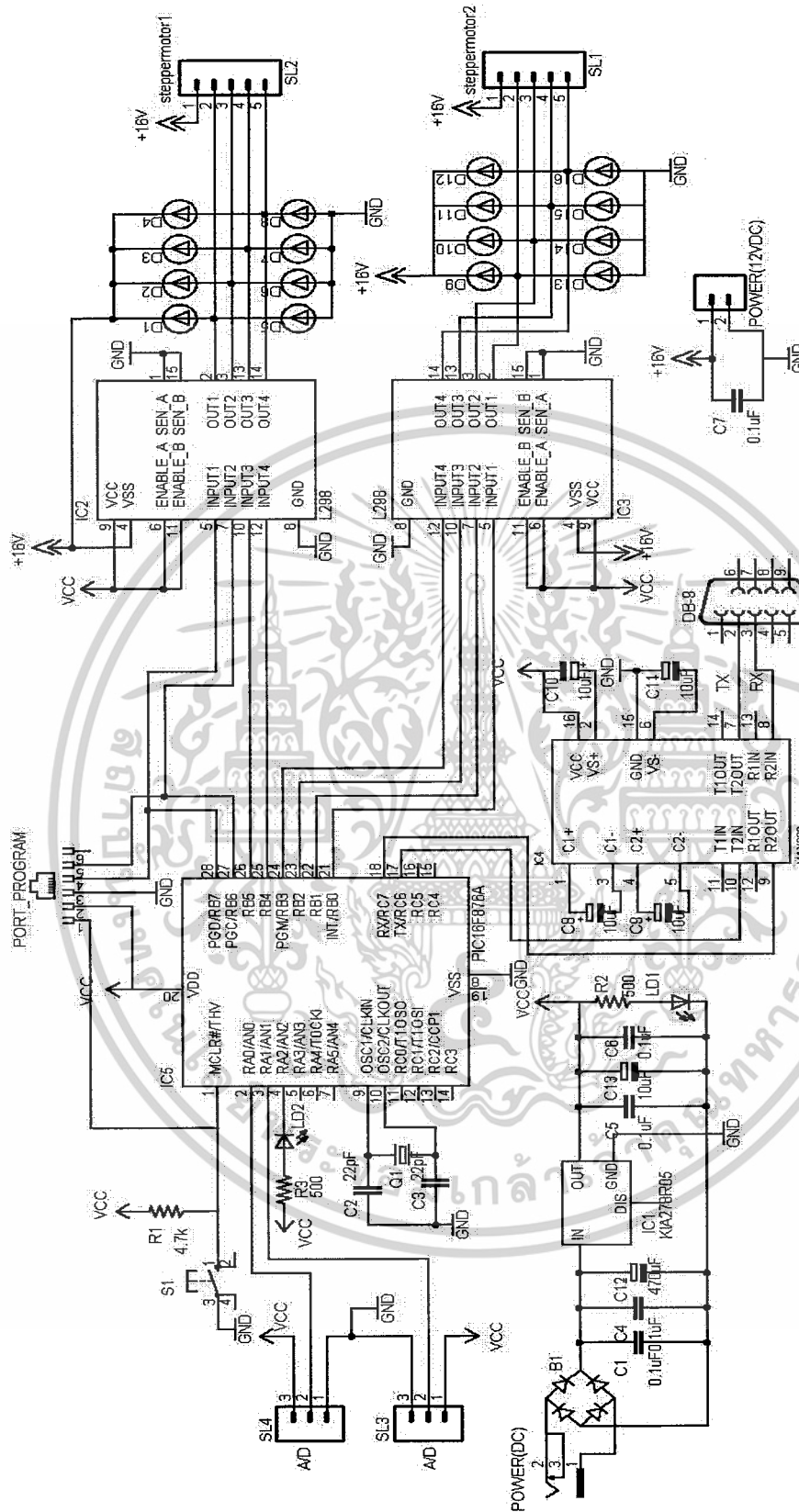
ในส่วนนี้จะเป็นการเชื่อมต่อไมโครคอนโทรลเลอร์เบอร์ PIC16F876A กับวงจรขับเคลื่อน ในการออกแบบจะใช้ไอซีเบอร์ L298 ทำหน้าที่ขับเคลื่อนเพื่อป้อนให้กับสเต็ปเปอร์มอเตอร์ เนื่องจากสามารถให้กระแสได้สูงสุดถึง 4 แอมแปร์



รูปที่ 3.22 วงจรการเชื่อมต่อไมโครคอนโทรลเลอร์กับวงจรขับเคลื่อน

3.6.3 วงจรรวมการควบคุมการเคลื่อนไหวนของตัวถ่วงด้วยคอมพิวเตอร์ผ่านพอร์ตอนุกรม

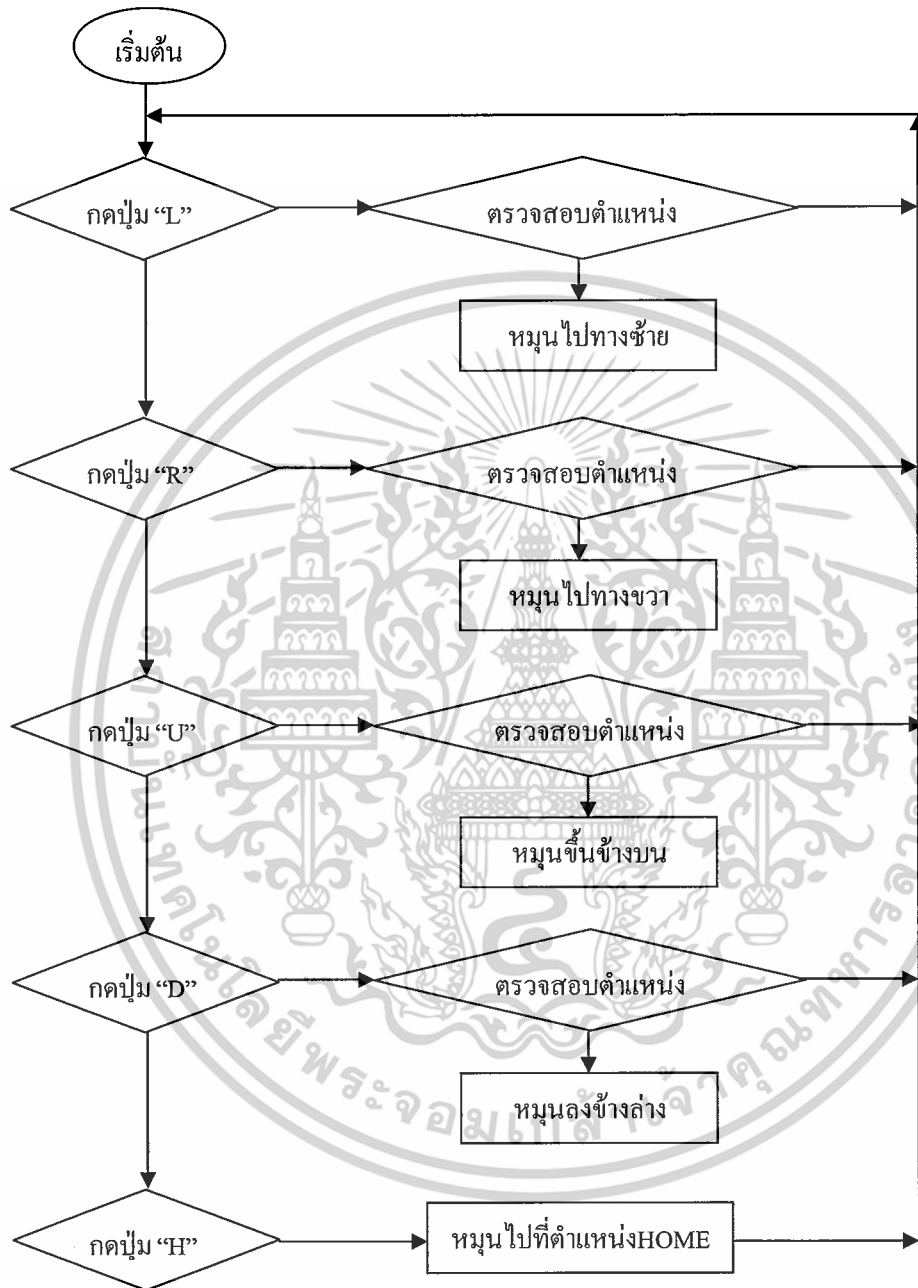
ประกอบด้วยไมโครคอนโทรลเลอร์เบอร์ PIC16F876A ,ไอซีเบอร์ MAX-232 และไอซีขับเคลื่อนเบอร์ L298 เมื่อมีคำสั่งอินพุตเข้ามาทางพอร์ตอินพุต ไมโครคอนโทรลเลอร์ก็จะทำการประมวลผลคำสั่งที่ได้รับมาว่าเป็นคำสั่งให้เคลื่อนฐานถ่วงไปที่ทิศทางใดแล้วก็จะทำตามคำสั่งนั้น โดยจะส่งแรงดันออกทางพอร์ตเอาต์พุต โดยมีไอซีเบอร์ L298 เป็นตัวขับเคลื่อนก่อนที่จะป้อนให้กับสเต็ปเปอร์มอเตอร์เพื่อให้ฐานถ่วงหมุนไปตามทิศทางที่ต้องการ



รูปที่ 3.23 วงจรรวมการควบคุมการเคลื่อน ไหวของตัวกลิ้งด้วยคอมพิวเตอร์ผ่านพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการควบคุมการเคลื่อนไหวของฐานกลิ้งนั้นเราจะใช้การกดตัวอักษรบนแป้นคีย์บอร์ด เพื่อป้อนคำสั่งให้กับไมโครคอนโทรลเลอร์ โดยมีโฟลว์ชาร์ตการทำงานดังนี้



รูปที่ 3.24 โฟลว์ชาร์ตการควบคุมสเต็ปเปอร์มอเตอร์ผ่านทางพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 ส่วนตัว Ethernet Module

ในส่วนนี้จะเป็นการนำไมโครคอนโทรลเลอร์มาทำการเชื่อมต่อกับตัว Ethernet Module เพื่อรับค่าการควบคุมกล้องผ่านทางหน้าเว็บเพจ โดยตัว Ethernet Module ซึ่งใช้ไอซีเบอร์ PIC18F66J65 ทำหน้าที่เป็น Data Protocol Converter สำหรับแปลงข้อมูลของการสื่อสารระหว่างอุปกรณ์ที่เป็น RS232 กับอุปกรณ์ที่เป็น Ethernet LAN ซึ่งจะช่วยลดความยุ่งยากสำหรับผู้ใช้งานที่ต้องการติดต่อสื่อสารข้อมูลกับระบบเครือข่ายของ Ethernet LAN

โดยจะแทนปุ่มคำสั่งควบคุมต่างๆด้วยรหัสแอสกี(ASCII) ดังนี้

ปุ่ม Left แทนด้วยรหัสแอสกี “A0860B”

ปุ่ม Right แทนด้วยรหัสแอสกี “A0870B”

ปุ่ม Up แทนด้วยรหัสแอสกี “A0850B”

ปุ่ม Down แทนด้วยรหัสแอสกี “A0880B”

ปุ่ม Home แทนด้วยรหัสแอสกี “A0890B”



บทที่ 4

ผลการทดลอง และวิเคราะห์ผลการทดลอง

ในบทนี้เป็นส่วนของการแสดงผลการทดลองที่เกิดขึ้นจริงของส่วนการแสดงผลและส่วนการควบคุมการเคลื่อนไหวของกล้องที่ได้ทำการสร้างขึ้นมาว่าสามารถทำงานได้จริงและมีประสิทธิภาพตามที่ต้องการหรือไม่

4.1 การทดลองวัดแรงดันที่เฟสแต่ละเฟสของสเต็ปเปอร์มอเตอร์

การทดลองจะแบ่งเป็น 2 ตอน

ตอนที่ 1 การกระตุ้นเฟสเดียว (Single Phase Excitation)

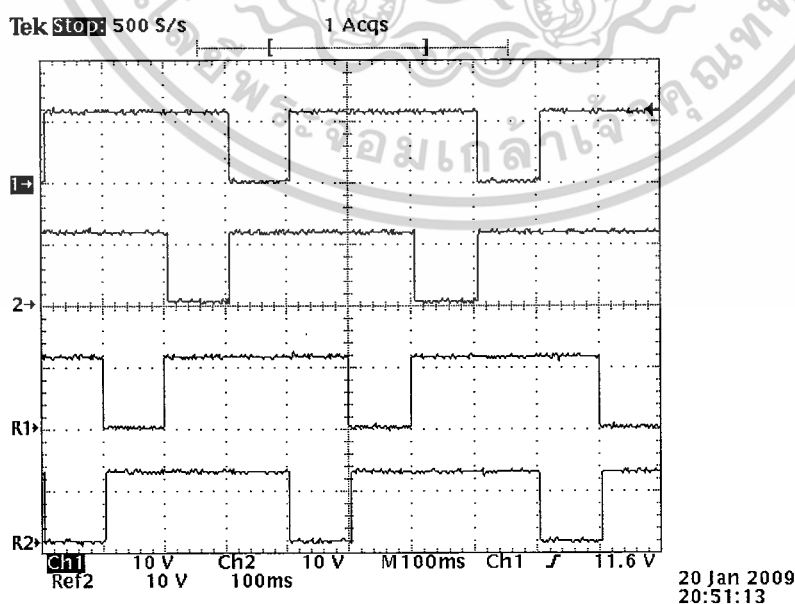
ตอนที่ 2 การกระตุ้น 2 เฟส (Two Phase Excitation)

สามารถทดลองได้โดยทำการต่อวงจรดังรูปที่ 3.23 ที่ประกอบด้วยไมโครคอนโทรลเลอร์เบอร์ PIC16F876A และไอซีขับกระแสเบอร์ L298 โดยทำการป้อนคำสั่งกับไมโครคอนโทรลเลอร์โดยใช้โปรแกรม HyperTerminal เพื่อควบคุมให้สเต็ปเปอร์มอเตอร์หมุนไปในทิศทางต่างๆแล้วทำการวัดสัญญาณที่เอาต์พุตทั้งสี่ขาของไอซีขับกระแสเบอร์ L298

ตอนที่ 1 การกระตุ้นเฟสเดียว (Single Phase Excitation)

เป็นการป้อนสัญญาณกระตุ้นให้กับขดลวดของสเต็ปเปอร์มอเตอร์ทีละขดดังรูปที่ 2.23 โดยจะใช้สัญญาณระดับศูนย์โวลต์ในการกระตุ้น เนื่องจากสเต็ปเปอร์มอเตอร์ที่ใช้เป็นชนิดยูนีโพล่า

- เมื่อทำการป้อนคำสั่งให้มอเตอร์หมุนซ้ายโดยการกดปุ่มตัวอักษร “L” บนแป้นคีย์บอร์ดแล้วทำการวัดสัญญาณเอาต์พุต จะเห็นว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนซ้าย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ซึ่งเน้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

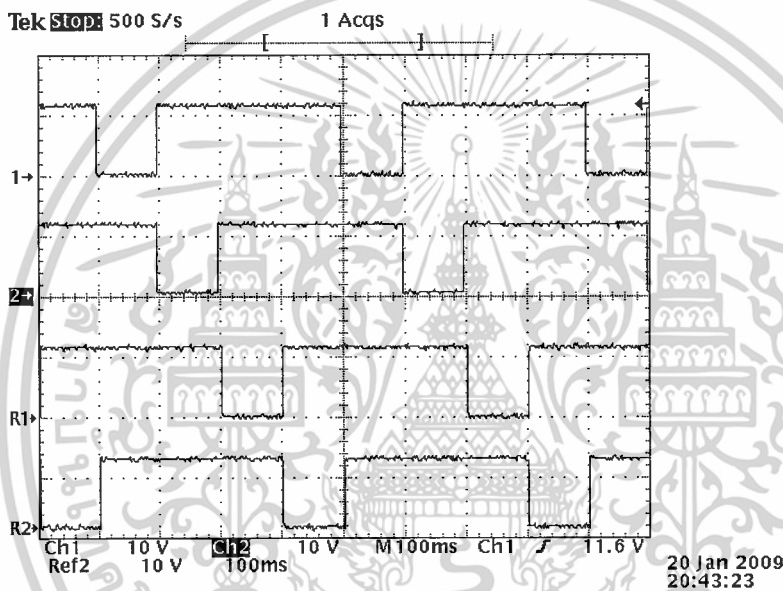
R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

• เมื่อเราป้อนคำสั่งให้มอเตอร์หมุนขวาโดยการกดที่ปุ่มตัวอักษร “R” บนแป้นคีย์บอร์ด แล้วทำการวัดสัญญาณเอาต์พุตจะเห็นได้ว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ โดยการสลับของพัลส์จะเป็นการสลับทิศทางที่ตรงกันข้ามกันกับกรณีที่เราป้อนคำสั่งให้มอเตอร์หมุนซ้ายซึ่งจะสังเกตความแตกต่างได้จากรูปที่ 4.2



รูปที่ 4.2 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนขวา

R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

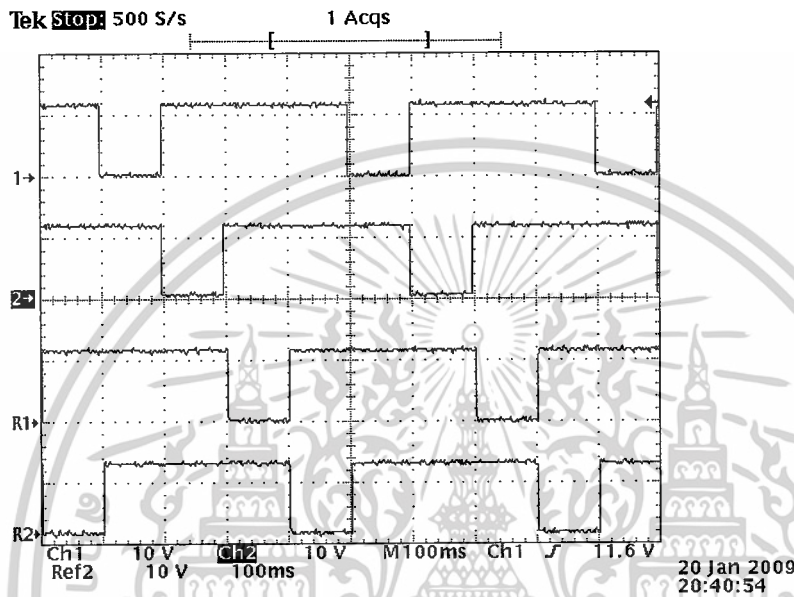
R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

ในทำนองเดียวกันการหมุนมอเตอร์ให้ไปในทิศทางขึ้นและลงก็มีหลักการที่เหมือนกันกับการหมุนมอเตอร์ไปในทิศทางซ้ายและขวาซึ่งสามารถแสดงได้ดังนี้

• เมื่อเราป้อนคำสั่งให้มอเตอร์หมุนขึ้นโดยการกดที่ปุ่มตัวอักษร “U” บนแป้นคีย์บอร์ด แล้วทำการวัดสัญญาณเอาต์พุต จะเห็นว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนขึ้น

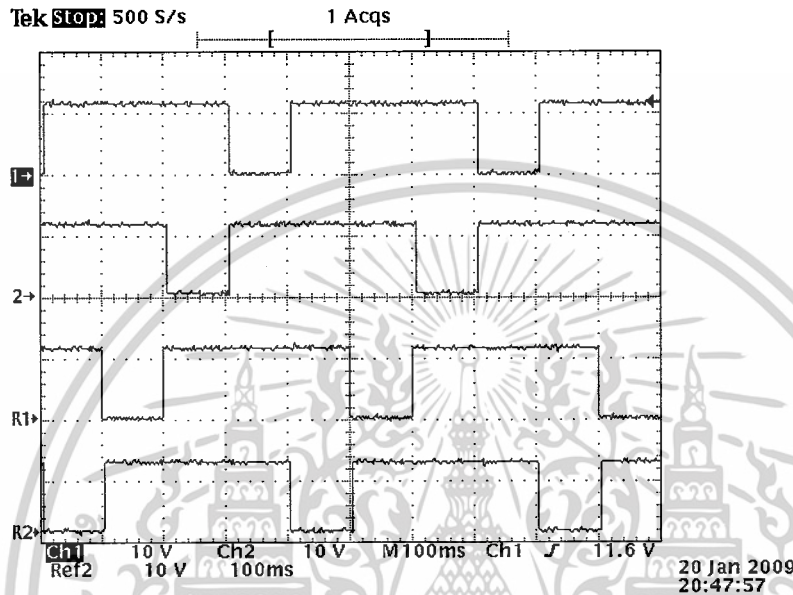
R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

• เมื่อเราป้อนคำสั่งให้มอเตอร์หมุนลงโดยการกดที่ปุ่มตัวอักษร “D” บนแป้นคีย์บอร์ด แล้วทำการวัดสัญญาณเอาต์พุต จะเห็นว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ โดยการสลับของพัลส์จะเป็นการสลับในทิศทางที่ตรงข้ามกันกับในกรณีที่เราป้อนคำสั่งให้มอเตอร์หมุนขึ้นซึ่งจะสังเกตความแตกต่างได้จากรูปที่ 4.4



รูปที่ 4.4 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนลง

R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

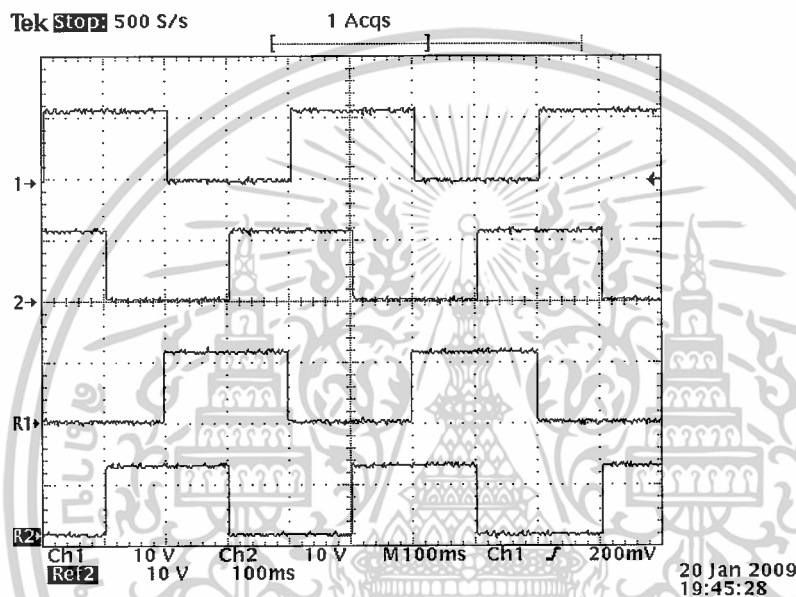
1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

ตอนที่ 2 การกระตุ้น 2 เฟส (Two Phase Excitation)

เป็นการป้อนสัญญาณกระตุ้นให้กับขดลวดของสเต็ปเปอร์มอเตอร์ที่ละ 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไป ดังรูปที่ 2.24 โดยจะใช้สัญญาณระดับศูนย์โวลต์ในการกระตุ้น เนื่องจากสเต็ปเปอร์มอเตอร์ที่ใช้เป็นชนิดยูนีโพลาร์

• เมื่อทำการป้อนคำสั่งให้มอเตอร์หมุนซ้ายโดยการกดที่ปุ่มตัวอักษร “L” บนแป้นคีย์บอร์ด แล้วทำการวัดสัญญาณเอาต์พุต จะเห็นว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนซ้าย

R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

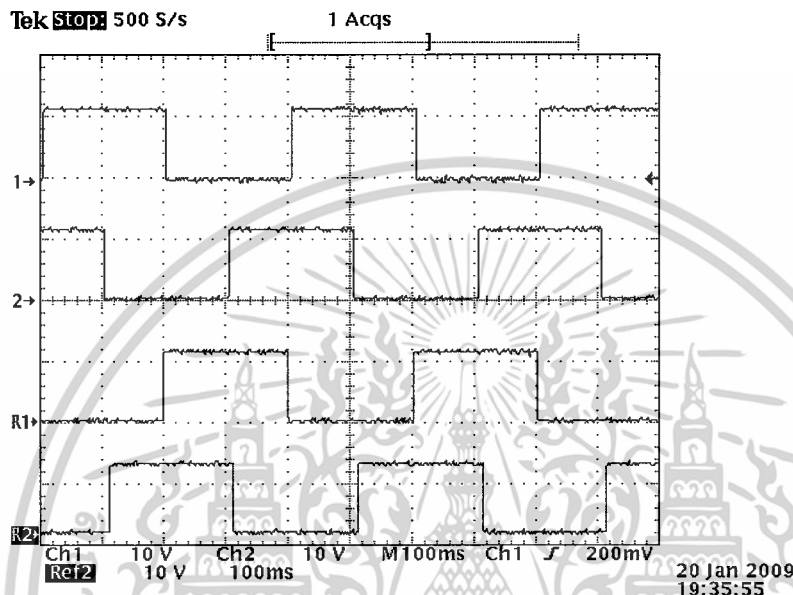
R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

ในการทำงานเกี่ยวกับการหมุนมอเตอร์ให้ไปในทิศทางขึ้นและลงก็มีหลักการที่เหมือนกันกับการหมุนมอเตอร์ไปในทิศทางซ้ายและขวาซึ่งสามารถแสดงได้ดังนี้

• เมื่อเราป้อนคำสั่งให้มอเตอร์หมุนขึ้นโดยการกดที่ปุ่มตัวอักษร “U” บนแป้นคีย์บอร์ด แล้วทำการวัดสัญญาณเอาต์พุต จะเห็นว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนขึ้น

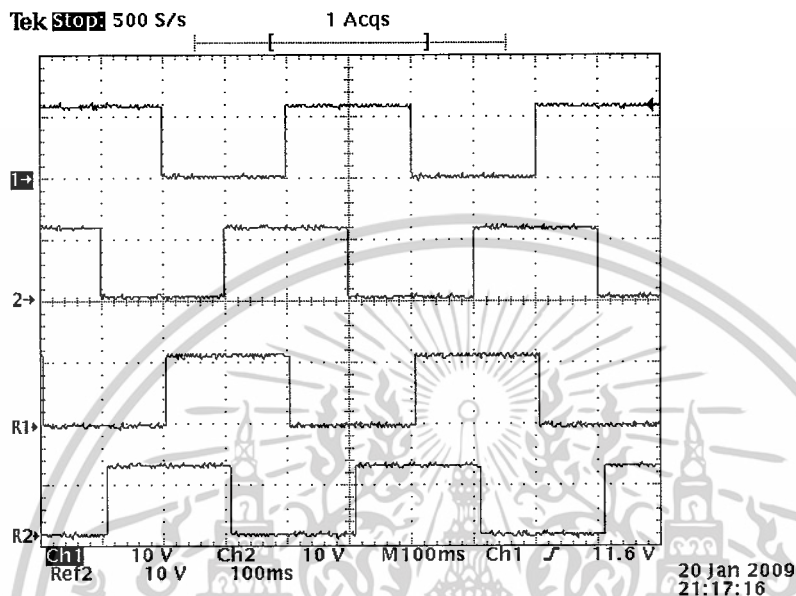
R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

• เมื่อเราป้อนคำสั่งให้มอเตอร์หมุนลงโดยการกดที่ปุ่มตัวอักษร “D” บนแป้นคีย์บอร์ด แล้วทำการวัดสัญญาณเอาต์พุต จะเห็นว่าสัญญาณที่ได้จะออกมาเป็นสัญญาณพัลส์สลับกันไปในแต่ละเฟสของมอเตอร์ โดยการสลับของพัลส์จะเป็นการสลับในทิศทางที่ตรงข้ามกันกับในกรณีที่เราป้อนคำสั่งให้มอเตอร์หมุนขึ้นซึ่งจะสังเกตความแตกต่างได้จากรูปที่ 4.8



รูปที่ 4.8 การป้อนสัญญาณให้แต่ละเฟสของมอเตอร์ให้หมุนลง

R1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 1 ของสเต็ปเปอร์มอเตอร์

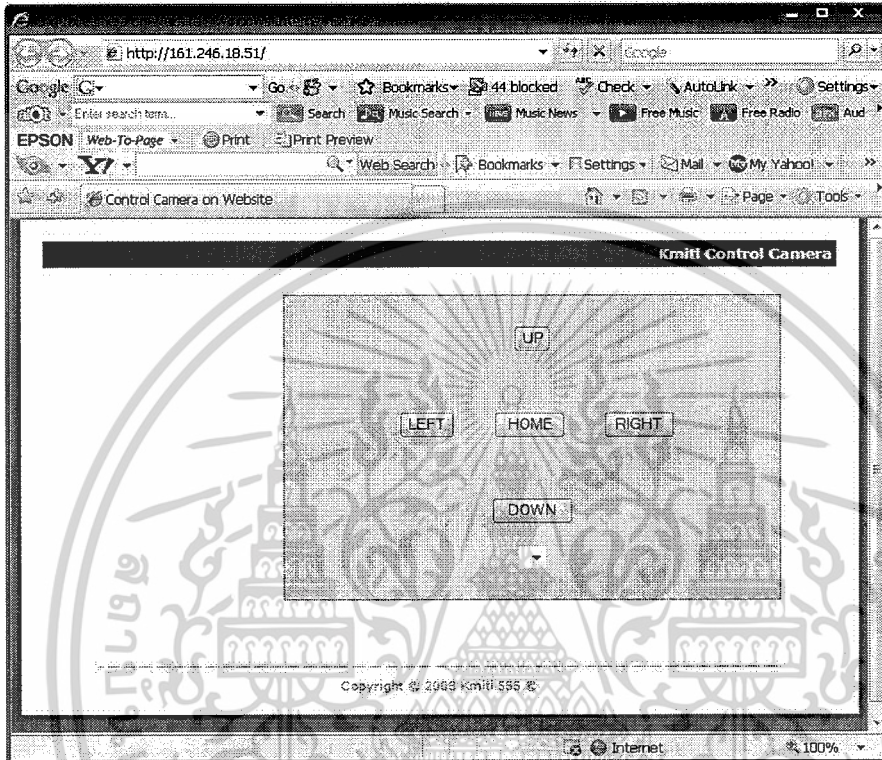
R2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 2 ของสเต็ปเปอร์มอเตอร์

1: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 3 ของสเต็ปเปอร์มอเตอร์

2: แสดงรูปสัญญาณที่วัดได้จากเฟสที่ 4 ของสเต็ปเปอร์มอเตอร์

4.2 การทดลองการควบคุมการเคลื่อนไหวของกล้องผ่านทางหน้าเว็บเพจ

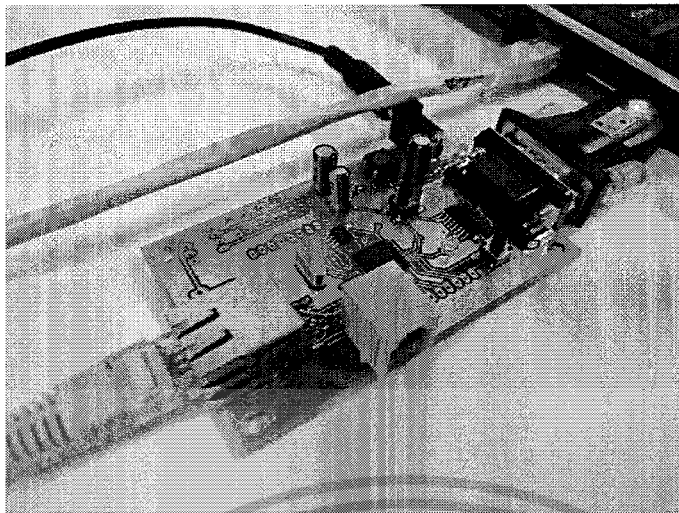
การควบคุมกล้องนั้นในตอนแรกเราจะรับคำสั่งการควบคุมจากคอมพิวเตอร์ แต่ตอนนี้เราจะมาทำการทดลองรับคำสั่งการควบคุมผ่านทางหน้าเว็บเพจดังรูปที่ 4.9 ซึ่งได้ทำการ โปรแกรมไว้ในตัว Ethernet Module เรียบร้อยแล้ว โดยมีไอพีเป็น 161.248.18.51



รูปที่ 4.9 หน้าเว็บเพจควบคุมการเคลื่อนไหวของกล้อง

ในการทดลองจะใช้โปรแกรม HyperTerminal รับค่าการกดปุ่มควบคุมต่างๆบนหน้าเว็บเพจจากเอาต์พุตของตัว Ethernet Module โดยในการทดลองจะใช้สาย LAN แบบตรงในการเชื่อมต่อตัว Ethernet Module เข้ากับคอมพิวเตอร์ดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 การเชื่อมต่อตัว Ethernet Module เข้ากับคอมพิวเตอร์

- ผลการทดลองเมื่อกดปุ่ม Left จะได้รับรหัสแอสกี “A0860B” ดังรูปที่ 4.11
- ผลการทดลองเมื่อกดปุ่ม Right จะได้รับรหัสแอสกี “A0870B” ดังรูปที่ 4.11
- ผลการทดลองเมื่อกดปุ่ม Up จะได้รับรหัสแอสกี “A0850B” ดังรูปที่ 4.11
- ผลการทดลองเมื่อกดปุ่ม Down จะได้รับรหัสแอสกี “A0880B” ดังรูปที่ 4.11
- ผลการทดลองเมื่อกดปุ่ม Home จะได้รับรหัสแอสกี “A0890B” ดังรูปที่ 4.11



รูปที่ 4.11 ผลการทดลองเมื่อทำการกดปุ่มควบคุมต่างๆบนหน้าเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

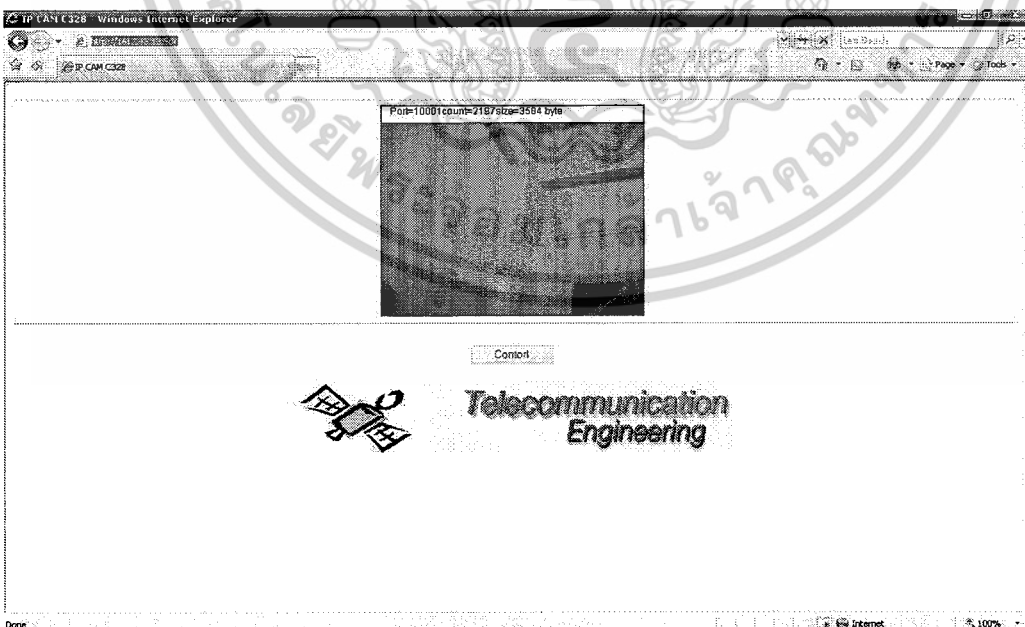
4.3 โปรแกรมแสดงภาพผ่านทางหน้าจคอมพิวเตอร์

ในส่วนนี้จะเป็นส่วนของการแสดงภาพผ่านทางหน้าจคอมพิวเตอร์ทางด้านผู้ใช้งาน ซึ่งเป็นการนำภาพในขณะนั้นมาแสดงผลผ่านทางโปรแกรม หน้าเว็บเพจที่ได้เขียนขึ้น เพื่อให้สามารถแสดงภาพจากกล้อง

เมื่อเปิด โปรแกรม จะแสดงหน้าจอหลักทางด้านผู้ใช้งาน ดังรูปที่ 4.12



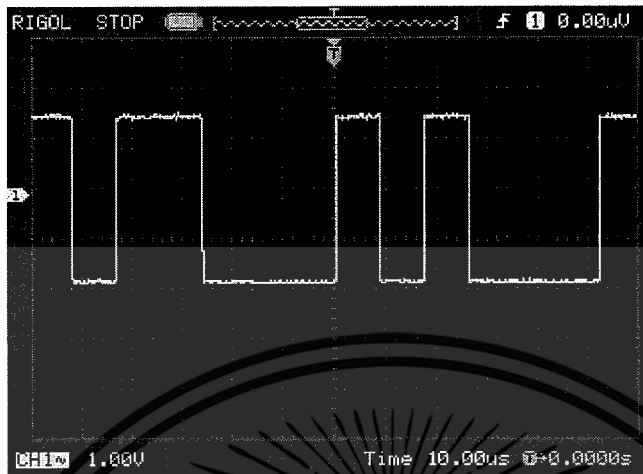
รูปที่ 4.12 หน้าจอโปรแกรมทางด้านผู้ใช้งาน



รูปที่ 4.13 ภาพเคลื่อนไหวออกทางหน้าจอทางด้านผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองบนหน้าเว็บเพจ จากคำสั่งของตัว C328 และ XPORT โดยในการทดลองจะใช้สาย LAN แบบตรงเข้ากับคอมพิวเตอร์ดัง รูปที่ 4.14



รูปที่ 4.14 ขาสัญญาณ TTL ส่งของตัว C328 และ XPORT

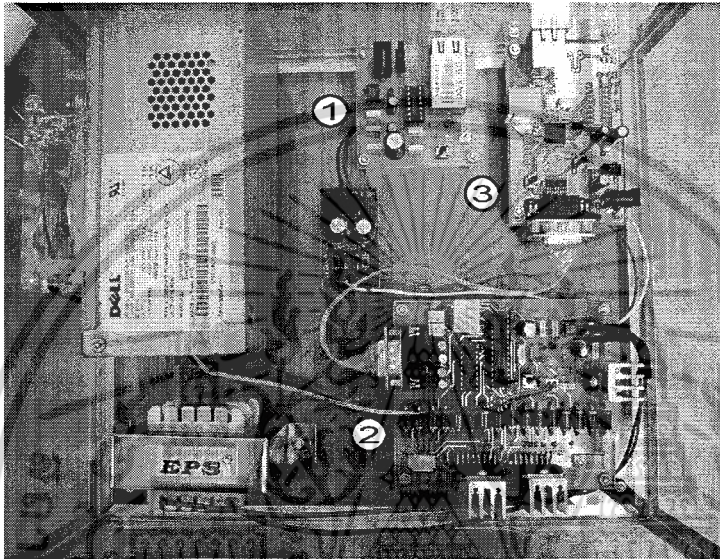


รูปที่ 4.15 ขาสัญญาณ TTL รับของตัว C328 และ XPORT

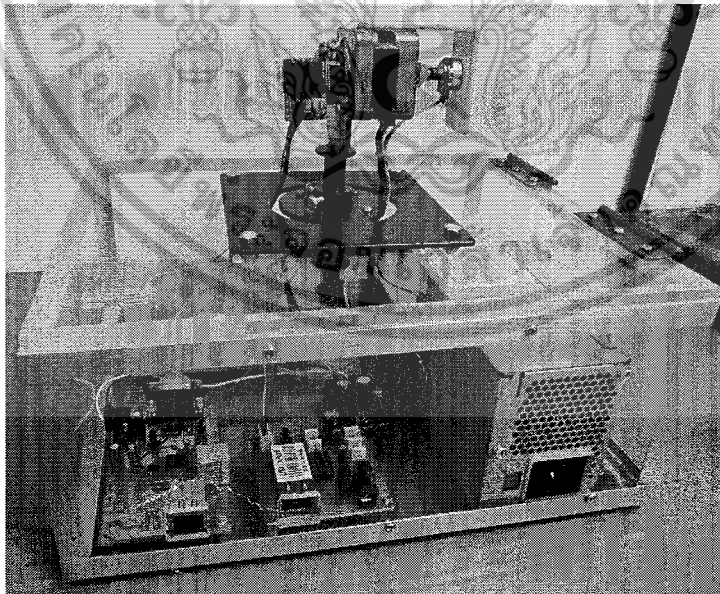
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การติดตั้งและการนำไปใช้งาน

เริ่มแรกจะทำการต่อโมดูลกล้อง C328 เข้ากับบอร์ด XPORT (หมายเลข 1 ซึ่งทำหน้าที่ส่งข้อมูลภาพไปยังเครือข่ายอินเทอร์เน็ต) และต่อมาทำการจะเชื่อมต่อบอร์ดอีเทอร์เน็ตโมดูล (หมายเลข 3 ซึ่งทำหน้าที่รับคำสั่งจากหน้าเว็บเพจเพื่อส่งไปให้ไมโครคอนโทรลเลอร์ทำการควบคุมการเคลื่อนไหวของกล้อง) เข้ากับบอร์ดไมโครคอนโทรลเลอร์และวงจรขับเคลื่อน (หมายเลข 2 ซึ่งทำหน้าที่ควบคุมการหมุนของสเต็ปเปอร์มอเตอร์ให้หมุนกล้องไปในทิศทางที่ต้องการ) ดังรูปที่ 4.16 และรูปที่ 4.17



รูปที่ 4.16 การเชื่อมต่ออุปกรณ์ส่วนต่างๆ



รูปที่ 4.17 การเชื่อมต่ออุปกรณ์ส่วนต่างๆ เข้ากับกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปผลการทดลอง

การทดลองจะแบ่งออกเป็น 2 ส่วนคือ

5.1.1 การแสดงภาพ

ในส่วนการแสดงผลภาพจะเห็นได้ว่าภาพที่แสดงบนหน้าเว็บเพจที่ใช้ จะมีอยู่ในระบบปฏิบัติการคอมพิวเตอร์ทั่วไปซึ่งสะดวกต่อการใช้งาน โดยผู้ใช้งานจะต้องเชื่อมต่ออยู่ในเครือข่ายเดียวกันหรือเชื่อมต่อผ่านเครือข่ายอินเทอร์เน็ตเพื่อสามารถดูภาพได้จากกระยะไกล

5.1.2 การควบคุมการเคลื่อนไหวกของฐานกล้อง

การทดลองจะแบ่งเป็น 2 ตอนคือการกระตุ้นแบบเฟสเดียวและการกระตุ้นแบบ 2 เฟส จากผลการทดลองในบทที่ 4 จะพบว่าการกระตุ้นแบบ 2 เฟสจะให้แรงบิดหรือ Torque ที่มากกว่าการกระตุ้นแบบเฟสเดียว แต่จะใช้กระแสมากกว่า ดังนั้นในการใช้งานจริงจึงเลือกใช้การกระตุ้นแบบ 2 เฟส และจากการทดลองควบคุมตัวกล้องผ่านทางหน้าเว็บเพจจะเห็นว่าสามารถควบคุมการเคลื่อนไหวได้ตามต้องการ

5.2 ปัญหาที่พบในการทดลองและแนวทางแก้ไข

5.2.1 การแสดงผลภาพ

จากการทดลองในส่วนของการแสดงผลภาพ จะมีปัญหาในเรื่องการกระตุกของภาพเนื่องจากกล้อง C328 ส่งภาพเป็นแบบ JPEG ข้อมูลมาเป็นเฟรมที่ละ 512 ไบต์ (เฟรมที่สูงที่สุด)

5.2.2 การควบคุมการเคลื่อนไหวกของฐานกล้อง

จากการทดลองในส่วนของการควบคุมการเคลื่อนไหวกของฐานกล้อง ปัญหาที่พบคือ ตัวกล้องเกิดการสั่นในเวลาหมุนทำให้ภาพสั่นตามไปด้วย เนื่องจากตัวกล้องถูกยึดติดกับสเต็ปเปอร์มอเตอร์โดยตรงและสเต็ปเปอร์มอเตอร์ที่ใช้มีองศาการหมุนต่อสเต็ปกว้างเกินไป แนวทางแก้ไขคือ ออกแบบฐานกล้องให้มีการทดเฟืองเพื่อลดการสั่นของกล้องให้น้อยลงเวลาเคลื่อนที่

ภาคผนวก

Java source code.

1.C328 source code

```
//      C328-7640 Camera Java Applet
//      XPort serial setting  Menu Channel 1/2
//      Baudrate (115200) ? <---- change from factory settings
//      I/F Mode (4C) ?
//      Flow (00) ?
//      Port No (10001) ?
//      ConnectMode (C0) ?
//      Send '+++ ' in Modem Mode (Y) ?
//      Auto increment source port (N) ?
//      Remote IP Address : (000).(000).(000).(000)
//      Remote Port (0) ?
//      DisConnMode (00) ?
//      FlushMode (80) ? <---- change from factory settings
//      Pack Cntrl (00) ?
//      DisConnTime (00:00) ? :
//      SendChar 1 (00) ?
//      SendChar 2 (00) ?
```

```
import java.applet.Applet;
```

```
import java.net.*;
```

```
import java.awt.*;
```

```
public class C328 extends Applet implements Runnable {
```

```
    public tcpip gtp=null;
```

```
    public InetAddress target_ip=null;
```

```
    public Image img=null;
```

```
    public int image_count=0, jpeg_size=0;
```

```
    public byte[] jpeg_data =new byte[0xFFFF];
```

```
    private volatile Thread th=null;
```

```
    private int target_port=0, status=-1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private int i=0,j=0,k=0, loop=0, data_size=0, get_size_flag=0;
private String str=null;
private byte[] data =new byte[1024];

// Commands of C328-7640 Camera
byte[] sync    ={-86,13, 0, 0, 0, 0 };//0xAA,0x0D,00,00,00,00
byte[] ack     ={-86,14,13, 0, 0, 0 };//0xAA,0x0E,0x0D,00,00,00
byte[] initial  ={-86, 1, 0, 7, 7, 7 };//0xAA,0x01,JPEG, size
byte[] package_size={ -86, 6, 8, 0, 2, 0 };//0xAA,0x06,512byte
byte[] get_picture ={-86, 4, 5, 0, 0, 0 };//0xAA,0x04,Get Preview Picture
byte[] ack_with_id ={-86,14, 0, 0, 0, 0 };//0xAA,0x0E,Package ID
byte[] ack_last_id ={-86,14, 0, 0, -16, -16 };//0xAA,0x0E,Package LastID
// byte[] reset    ={-86, 8, 0, 0, 0, 0 };//0xAA,0x08,Reset whole system
byte[] Power_Off  ={-86, 9, 0, 0, 0, 0 };//0xAA,0x09,

public void init() {
    gtp = null; target_ip = null;
    image_count=0;
    target_port = Integer.parseInt(getParameter("port_munber")); //Get from *.html
    initial[3]=(byte)(Integer.parseInt(getParameter("image_size")));//Get from *.html
    // size parameter 3=160*128, 5=320*240, 7=640*480
    try { //Get IP-address
        target_ip = InetAddress.getByName(getCodeBase().getHost());
    }
    catch (UnknownHostException e){ }
}

public void start(){
    if(th==null){
        th=new Thread(this); th.start();
    }
}

public void run(){
    Thread thisThread=Thread.currentThread();

    while( th==thisThread){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( status==1 ){
    img=null;
    try{
        if(target_ip != null){
            if(gtp == null){
                gtp = new tcpip(target_ip, target_port);
                if(gtp.s != null)
                    status=0;
            }
        }
    }
    catch(Exception e){}
    repaint();
    try{
        if( status==0 ){
//            gtp.bsend(reset);//Send reset
//            thisThread.sleep(3000);
            i=0; data=gtp.receives(gtp.length());//buffer clear
            while( i<60 ){//sending sync 60 count
                gtp.bsend(sync); thisThread.sleep(40);//sending sync & wait
                if( gtp.length()==12 ){
                    gtp.bsend(ack);//sending sync
                    status=1; i=60;
                }
                ++i;
            }
        }
    }
    catch(Exception e){}
    repaint();
    try{
        if( status==1 ){
            i=0; data=gtp.receives(gtp.length());
            gtp.bsend(initial);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while( i<2000 ){
    thisThread.sleep(2);
    if( gtp.length()>=6 ){
        status=2; i=2000;//inital OK
    }
    ++i;
}
}
}
}
catch(Exception e){}
repaint();
try{
    if( status==2 ){
        i=0; data=gtp.receive(gtp.length());//buffer clear
        gtp.bsend(package_size);
        while( i<2000 ){
            thisThread.sleep(2);
            if( gtp.length()>=6 ){
                status=3; i=2000;//inital OK
                thisThread.sleep(1000);
                //wait for 1-2seconds befor capturing the first picture
            }
            ++i;
        }
    }
}
}
}
catch(Exception e){}
} // if( status==0 )
try{
    if( status==3 ){
        i=0; data=gtp.receive(gtp.length());//buffer clear
        gtp.bsend(get_picture);//Send get_picture
        while( i<2000 ){
            thisThread.sleep(2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( gtp.length()>=12 ){
    status=4; i=2000;//get picture OK
}
++i;
}
}
}
catch(Exception e){}
try{
    loop=0; j=0;
    while( status==4 ){
        data=gtp.receive(gtp.length());//buffer clear
        get_size_flag=0; i=0; ack_with_id[4]=(byte)loop;
        gtp.send(ack_with_id);// send ack with ID
        while( i<5000 ){
            thisThread.sleep(2);
            if( get_size_flag==0 && gtp.length()>=4 ){
                data=gtp.receive(4);
                data_size=((data[3]<<8) & 0xFF00)+(data[2] & 0xFF)+2;//+VerifyCode 2byte
                if( data_size > 508 || 2 > data_size ) break;
                get_size_flag=1;
            }
            if( get_size_flag==1 ){
                if(gtp.length()>=data_size){
                    data=gtp.receive(data_size);
                    for(k=0; k<(data_size-2); ++k){
                        jpeg_data[j]=data[k]; ++j;
                    }
                    i=9999;
                    if( data_size<508 ){
                        status=5; gtp.send(ack_last_id);
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

g.drawString("PLEASE SEE SUPPLY",10,70);
g.drawString("WAIT.....",10,110);
}
if( status==1 || status==2 )
    g.drawString("wait... load image",10,110);
if( image_count>0 && status>=3 ){
    g.drawImage(img ,0, 20, this );
    if( (image_count % 10)!=0 )
        str="Port="+target_port+"count="+image_count+"size="+jpeg_size+" byte";
    else
        str="count="+image_count+"size="+jpeg_size+" byte";
    g.drawString(str,10,10);
}
}
public void stop(){
    th = null;
    if (gtp != null)
        gtp.disconnect();
    gtp = null;
}
public void destroy(){
    th = null;
    if (gtp != null)
        gtp.disconnect();
    gtp = null;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.TCP/IP source code

```
// There is not support about this program.
import java.*;
import java.lang.*;
import java.net.*;
import java.util.*;
import java.io.*;

//
// This class opens a TCP connection, and allows reading and writing of byte arrays.
//
public class tcpip{
    protected Socket s = null;
    public DataInputStream dis = null;
    protected DataOutputStream dos = null;

    public tcpip(InetAddress ipa, int port){
        Socket s1 = null;
        try{ // Open the socket
            s1 = new Socket(ipa.getHostAddress(), port);
        }
        catch (IOException e) {
            System.out.println("Error opening socket");
            return;
        }
        s = s1;
        try{ // Create an input stream
            dis = new DataInputStream( new BufferedInputStream(s.getInputStream()));
        }
        catch (Exception ex) {
            System.out.println("Error creating input stream");
        }
        try{ // Create an output stream
            dos = new DataOutputStream( new BufferedOutputStream(s.getOutputStream()));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
catch (Exception ex) {
    System.out.println("Error creating output stream");
}
}

```

```

public synchronized void disconnect(){
    if (s != null) {
        try{
            s.close();
        }
        catch (IOException e){ }
    }
}

```

```

public synchronized void bsend(byte[] temp){
    try{
        dos.write(temp, 0, temp.length);
        dos.flush();
    }
    catch(Exception ex) {
        System.out.println("Error sending data : " + ex.toString());
    }
}

```

```

public synchronized byte[] receives(int length){
    byte[] retval = new byte[length];
    try{
        dis.read(retval);
    }
    catch (IOException e){ }
    return(retval);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public synchronized int length(){
    int len = 0;
    try{
        len = dis.available();
    }
    catch (IOException e){ }
    return len;
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมคำสั่งควบคุมฐานกล้อง

```
#include <16F876A.h>
#device ADC=10
#FUSES
NOWDT,HS,NOPUT,NOPROTECT,NODEBUG,NOBROWNOUT,NOLVP,NOCPD,NOWRT
#use delay(clock=10000000)
#use rs232(baud=19200,xmit=PIN_C6,rcv=PIN_C7)

#define Start 'A'
#define End 'B'
#define LEFT '6'
#define RIGHT '7'
#define UP '5'
#define DOWN '8'
#define HOME '9'

/*****

signed int i=0,j=0;
int stepmotor_1[4] = {0x3F,0x6F,0xCF,0x9F};
int stepmotor_2[4] = {0xF3,0xF6,0xFC,0xF9};
int16 read0,read1;

int buffer[30],sw_task=0;
int16 ledst=0;
int ledack=0,a;

*****/

#int_rda
void serial_isr()
{
```

char ch;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static int a;

ch = getc();

switch(ch)
{
case Start:
    a = 0; sw_task = 0;
    break;

case End:
    sw_task = buffer[2];
    break;

default:
    buffer[a++] = ch;
}
}

/*****

void Control_Steptomotor(int select)
{
if(select==1)
{
    if(read0<0x03F0)
    {
        i++;
        if(i>3)
            i=0;
        output_B(stepmotor_1[i]);
        delay_ms(100);
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(select==2)
{
  if(read0>0x000F)
  {
    i--;
    if(i<0)
      i=3;
    output_B(stepmotor_1[i]);
    delay_ms(100);
  }
}

```

```

if(select==3)
{
  if(read1<0x03F0)
  {
    j++;
    if(j>3)
      j=0;
    output_B(stepmotor_2[j]);
    delay_ms(100);
  }
}

```

```

if(select==4)
{
  if(read1>0x000F)
  {
    j--;
    if(j<0)
      j=3;
    output_B(stepmotor_2[j]);
    delay_ms(100);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
if(select==5)
{
    if(read0>0x020F)
    {
        i--;
        if(i<0)
            i=3;
        output_B(stepmotor_1[i]);
        delay_ms(100);
    }
    if(read0<0x01F0)
    {
        i++;
        if(i>3)
            i=0;
        output_B(stepmotor_1[i]);
        delay_ms(100);
    }
    if(read1>0x020F)
    {
        j--;
        if(j<0)
            j=3;
        output_B(stepmotor_2[j]);
        delay_ms(100);
    }
    if(read1<0x01F0)
    {
        j++;
        if(j>3)
            j=0;
        output_B(stepmotor_2[j]);
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        delay_ms(100);
    }
}

/*****

void ledack_st(void)
{
    ledst++;
    if(ledst>=10000)
    {
        ledst=0;
        ledack++;
        if(ledack==1)
        {
            output_low(PIN_A2);
            delay_ms(10);
        }
        if(ledack==2)
        {
            ledack=0;
            output_high(PIN_A2);
            delay_ms(10);
        }
    }
}

*****/

```

```
#ZERO_RAM
```

```
void main(void)
```

```
{
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

int k=0;

delay_ms(100);
set_tris_a(0xFF);
set_tris_b(0xFF);
set_tris_c(0xFF);

sw_task = 0;
port_b_pullups(true);

enable_interrupts(GLOBAL);
enable_interrupts(int_rda);
setup_adc_ports(RA0_ANALOG);
setup_adc(ADC_CLOCK_INTERNAL);

While(true)
{

set_adc_channel(0); // (AN0 or RA0)
delay_us(10);
read0 = read_adc();
set_adc_channel(1); // (AN1 or RA1)
delay_us(10);
read1 = read_adc();

switch(sw_task)
{
case LEFT:
sw_task=0;
for(k=0;k<3;k++)
{
Control_Steptomotor(1);
}
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case RIGHT:
    sw_task=0;
    for(k=0;k<3;k++)
    {
        Control_Steptomotor(2);
    }
break;

```

```

case UP:
    sw_task=0;
    for(k=0;k<3;k++)
    {
        Control_Steptomotor(3);
    }
break;

```

```

case DOWN:
    sw_task=0;
    for(k=0;k<3;k++)
    {
        Control_Steptomotor(4);
    }
break;

```

```

case HOME:
    Control_Steptomotor(5);

```

```

default:
ledack_st();
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C328-7640 User Manual

Release Note:

1. Jan 28, 2004 – official released v1.0

2. Feb 24, 2004 – official released v1.1

- Fix the definition of verify code
- Fix the bug of unable jump to power save mode
- Fix the incorrect connection speed after wake up from power save mode

3. Apr 24, 2004 – official released v2.0

- Add auto baud-rate detection
- Add support of 9600bps, 19200bps, 38400bps
- Disable the 8-bit colour for uncompressed picture

4. Apr 12, 2005 – official released v2.1

- Add command to change the light frequency between 50/60 Hz
- Add more descriptions of the resolution selection

5. Aug 19, 2005 – official released v3.0

- Add description of the auto power mode
- Add FAQ section



Rm 802, Nan Fung Ctr, Castle Peak Rd, Tsuen Wan NT, Hong Kong

Tel: (852) 2498 6248

Fax (852) 2414 3050

Email: sales@comedia.com.hk

<http://www.comedia.com.hk>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

General Description

The C328 module is a highly integrated serial camera board that can be attached to a wireless or PDA host performing as a video camera or a JPEG compressed still camera. It provides a serial interface (RS-232) and JPEG compression engine to act as a low cost and low powered camera module for high-resolution serial bus security system or PDA accessory applications.

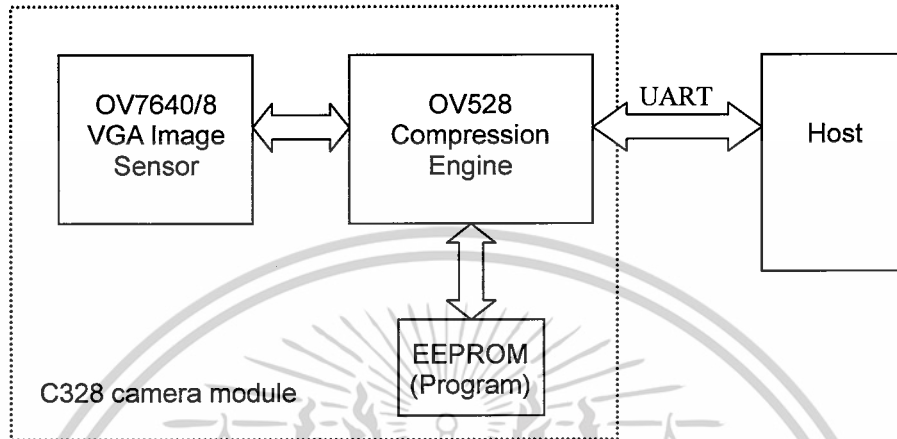


Figure 1 – System block diagram

Features

- Small in size, low cost and low powered (3.3V) camera module for high-resolution serial bus security system or PDA accessory applications.
- On-board EEPROM provides a command-based interface to external host via RS-232.
- UART: 115.2Kbps for transferring JPEG still pictures or 160x128 preview @8bpp with 0.75fps.
- On board OmniVision OV7640/8 VGA color sensor.
- Built-in JPEG CODEC for different resolutions.
- Built-in down sampling, clamping and windowing circuits for VGA, QVGA, 160x120 or 80x60 image resolutions.
- Built-in color conversion circuits for 2-bit gray, 4-bit gray, 8-bit gray, 12-bit RGB, 16-bit RGB or standard JPEG preview images.
- No external DRAM required.

System Configuration

1. Camera Sensor
The C328-7640 module uses OmniVision OV7640/8 VGA color digital CameraChips with an 8-bit YCbCr interface.
2. OV528 Serial Bridge
The OV528 Serial Bridge is a JPEG CODEC embedded controller chip that can compress and transfer image data from CameraChips to external device. The OV528 takes 8-bit YCbCr 422 progressive video data from an OV7640/8 CameraChip. The camera interface synchronizes with input video data and performs down sampling, clamping and windowing functions with desired resolution, as well as color conversion that is requested by the user through serial bus host commands.
The JPEG CODEC can achieve higher compression ratio and better image quality for various image resolutions.
3. Program EEPROM
A serial type program memory is built-in for C328-7640 to provide a set of user-friendly command interfacing to external host.

Board Layout

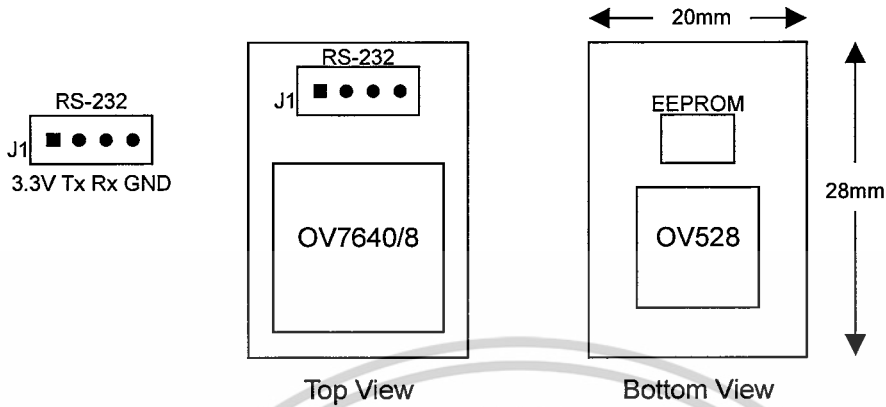


Figure 2 – C328-7640 board layout and serial interface pin

Serial Interface

1. Single Byte Timing Diagram

A single byte RS-232 transmission consists of the start bit, 8-bit contents and the stop bit. A start bit is always 0, while a stop bit is always 1. LSB is sent out first and is right after the start bit.

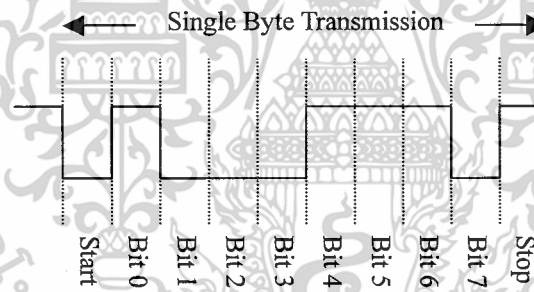


Figure 3 – RS-232 single byte timing diagram

2. Command Timing Diagram

A single command consists of 6 continuous single byte RS-232 transmissions. The following is an example of SYNC (AA0D00000000h) command.

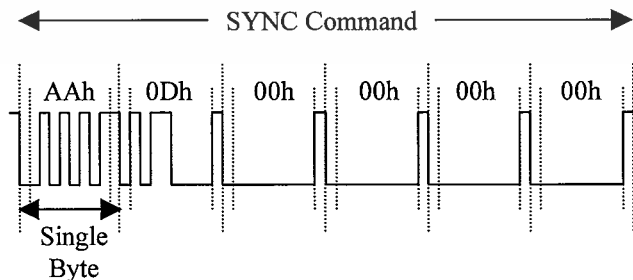


Figure 4 – RS-232 SYNC command timing diagram

Command Set

The C328-7640 module supports total 11 commands for interfacing to host as following:

| Command | ID Number | Parameter1 | Parameter2 | Parameter3 | Parameter4 |
|------------------|-----------|----------------|-----------------------|-----------------------------------|-------------------------|
| Initial | AA01h | 00h | Color Type | RAW Resolution (Still image only) | JPEG Resolution |
| Get Picture | AA04h | Picture Type | 00h | 00h | 00h |
| Snapshot | AA05h | Snapshot Type | Skip Frame Low Byte | Skip Frame High Byte | 00h |
| Set Package Size | AA06h | 08h | Package Size Low Byte | Package Size High Byte | 00h |
| Set Baudrate | AA07h | 1st Divider | 2nd Divider | 00h | 00h |
| Reset | AA08h | Reset Type | 00h | 00h | xxh* |
| Power Off | AA09h | 00h | 00h | 00h | 00h |
| Data | AA0Ah | Data Type | Length Byte 0 | Length Byte 1 | Length Byte 2 |
| SYNC | AA0Dh | 00h | 00h | 00h | 00h |
| ACK | AA0Eh | Command ID | ACK counter | 00h / Package ID Byte 0 | 00h / Package ID Byte 1 |
| NAK | AA0Fh | 00h | NAK counter | Error Number | 00h |
| Light Frequency | AA13h | Frequency Type | 00h | 00h | 00h |

* If the parameter is 0xFF, the command is a special Reset command and the firmware responds to it immediately.

1. Initial (AA01h)

The host issues this command to configure the preview image size and color type. After receiving this command, the module will send out an ACK command to the host if the configuration success. Otherwise, an NACK command will be sent out.

1.1 Color Type

C328-7640 can support 7 different color types as follow:

| | |
|------------------|-----|
| 2-bit Gray Scale | 01h |
| 4-bit Gray Scale | 02h |
| 8-bit Gray Scale | 03h |
| 12-bit Color | 05h |
| 16-bit Color | 06h |
| JPEG | 07h |

1.2 Preview Resolution

| | |
|---------|-----|
| 80x60 | 01h |
| 160x120 | 03h |

1.3 JPEG Resolution

Since the Embedded JPEG Code can support only multiple of 16, the JPEG preview mode can support following image sizes. It is different from normal preview mode.

| | |
|---------|-----|
| 80x64 | 01h |
| 160x128 | 03h |
| 320x240 | 05h |
| 640x480 | 07h |

2. Get Picture (AA04h)

The host gets a picture from C328-7640 by sending this command.

2.1 Picture Type

| | |
|----------------------|-----|
| Snapshot Picture | 01h |
| Preview Picture | 02h |
| JPEG Preview Picture | 05h |

3. Snapshot (AA05h)

C328-7640 keeps a single frame of JPEG still picture data in the buffer after receiving this command.

3.1 Snapshot Type

| | |
|----------------------|-----|
| Compressed Picture | 00h |
| Uncompressed Picture | 01h |

3.2 Skip Frame Counter

The number of dropped frames can be defined before compression occurs. "0" keeps the current frame, "1" captures the next frame, and so forth.

4. Set Package Size (AA06h)

The host issues this command to change the size of data package which is used to transmit JPEG image data from the C328-7640 to the host. This command should be issued before sending Snapshot command or Get Picture command to C328-7640. It is noted that the size of the last package varies for different image.

4.1 Package Size

The default size is 64 bytes and the maximum size is 512 bytes.



- ID -> Package ID, starts from zero for an image
- Data Size -> Size of image data in the package
- Verify Code -> Error detection code, equals to the lower byte of sum of the whole package data except the verify code field. The higher byte of this code is always zero. i.e. verify code = lowbyte(sum(byte[0] to byte[N-2]))

Note: As the transmission of uncompressed image is not in package mode, it is not necessary to set the package size for uncompressed image.

5. Set Baudrate (AA07h)

Set the C328-7640 baud rate by issuing this command. As the module can auto-detect the baud rate of the incoming command, host can make connection with one of the following baud rate in the table. The module will keep using the detected baud rate until physically power off

5.1 Baudrate Divider

Baudrate = 14.7456MHz / 2 x (2nd Divider + 1) / 2 x (1st Divider + 1)

| Baudrate | 1 st Divider | 2 nd Divider | Baudrate | 1 st Divider | 2 nd Divider |
|-----------|-------------------------|-------------------------|------------|-------------------------|-------------------------|
| 7200 bps | ffh | 01h | 28800 bps | 3fh | 01h |
| 9600 bps | bffh | 01h | 38400 bps | 2fh | 01h |
| 14400 bps | 7fh | 01h | 57600 bps | 1fh | 01h |
| 19200 bps | 5fh | 01h | 115200 bps | 0fh | 01h |

6. Reset (AA08h)

The host reset C328-7640 by issuing this command.

6.1 Reset Type

“00h” resets the whole system. C328-7640 will reboot and reset all registers and state machines.
 “01h” resets state machines only.

7. Power Off (AA09h)

C328-7640 will go into sleep mode after receiving this command. SYNC command (AA0Dh) must be sent to wake up C328-7640 for certain period until receiving ACK command from C328-7640.

8. Data (AA0Ah)

C328-7640 issues this command for telling the host the type and the size of the image data which is ready for transmitting out to the host.

8.1 Data Type

| | |
|----------------------|-----|
| Snapshot Picture | 01h |
| Preview Picture | 02h |
| JPEG Preview Picture | 05h |

8.2 Length

These three bytes represent the length of data of the Snapshot Picture, Preview Picture or JPEG Preview Picture.

9. SYNC (AA0Dh)

Either the host or the C328-7640 can issue this command to make connection. An ACK command must be sent out after receiving this command.

10. ACK (AA0Eh)

This command indicates the success of last operation. After receiving any valid command, ACK command must be sent out except when getting preview data. The host can issue this command to request image data package with desired package ID after receiving Data command from C328-7640. The host should send this command with package ID F0F0h after receiving a package to end the package transfer. Note that the field “command ID” should be 00h when request image data package.

10.1 Command ID

The command with that ID is acknowledged by this command.

10.2 ACK Counter

No use.

10.3 Package ID

For acknowledging Data command, these two bytes represent the requested package ID. While for acknowledging other commands, these two bytes are set to 00h.

11. NAK (AA0Fh)

This command indicates corrupted transmission or unsupported features.

11.1 NAK Counter

No use.

11.2 Error Number

| | | | |
|----------------------|-----|---------------------------------|-----|
| Picture Type Error | 01h | Parameter Error | 0bh |
| Picture Up Scale | 02h | Send Register Timeout | 0ch |
| Picture Scale Error | 03h | Command ID Error | 0dh |
| Unexpected Reply | 04h | Picture Not Ready | 0fh |
| Send Picture Timeout | 05h | Transfer Package Number Error | 10h |
| Unexpected Command | 06h | Set Transfer Package Size Wrong | 11h |
| SRAM JPEG Type Error | 07h | Command Header Error | F0h |
| SRAM JPEG Size Error | 08h | Command Length Error | F1h |
| Picture Format Error | 09h | Send Picture Error | F5h |
| Picture Size Error | 0ah | Send Command Error | ffh |

12. Light Frequency (AA13h)

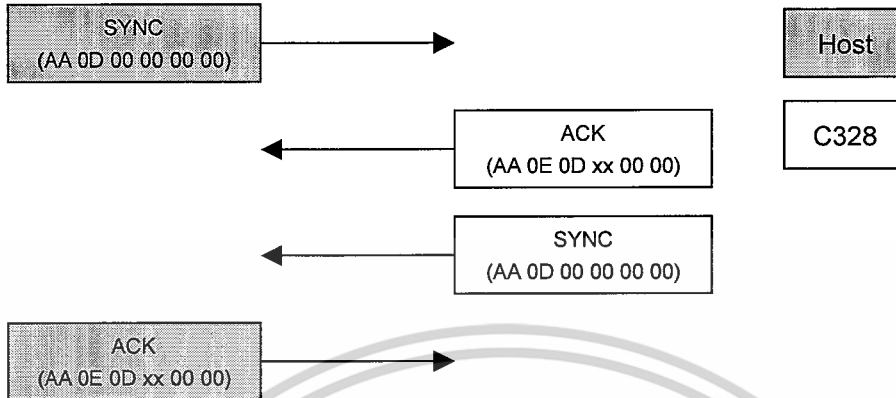
The host issues this command to change the light frequency of the C328-7640.

12.1 Light Frequency Type

| | |
|------|-----|
| 50Hz | 00h |
| 60Hz | 01h |

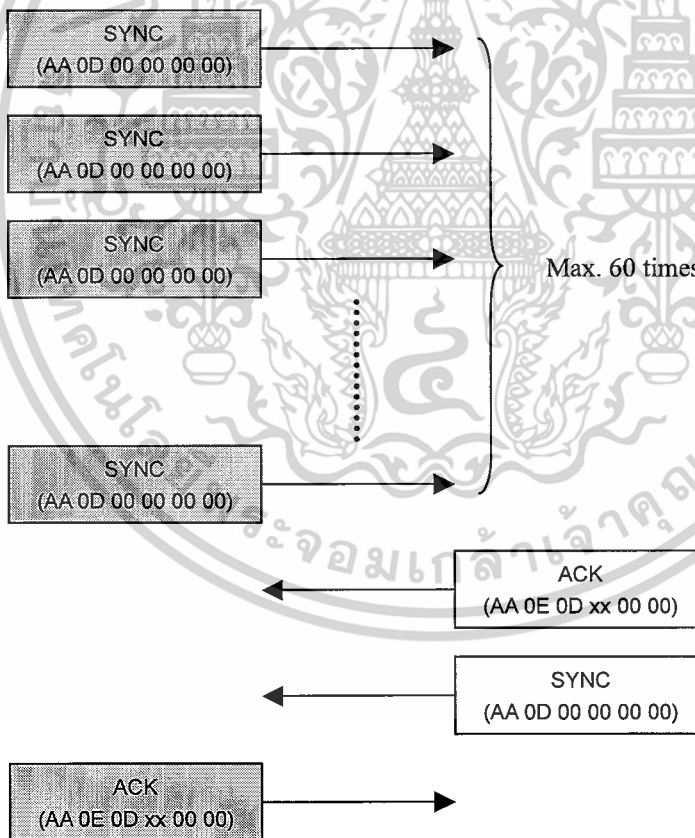
Command Protocol

1. SYNC Command

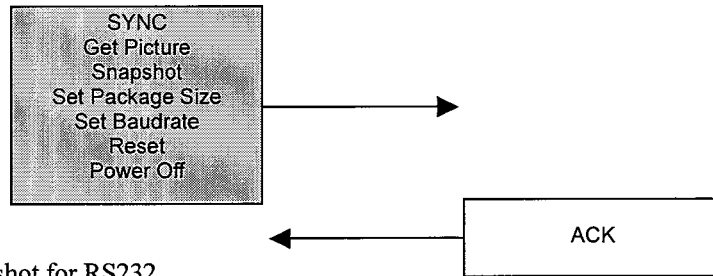


2. Make Connection with C328-7640

Send the SYNC command (at 14400bps) until receiving ACK command from C328-7640 (usually an ACK command is receive after sending 25 times of SYNC command). This must be done after power up.



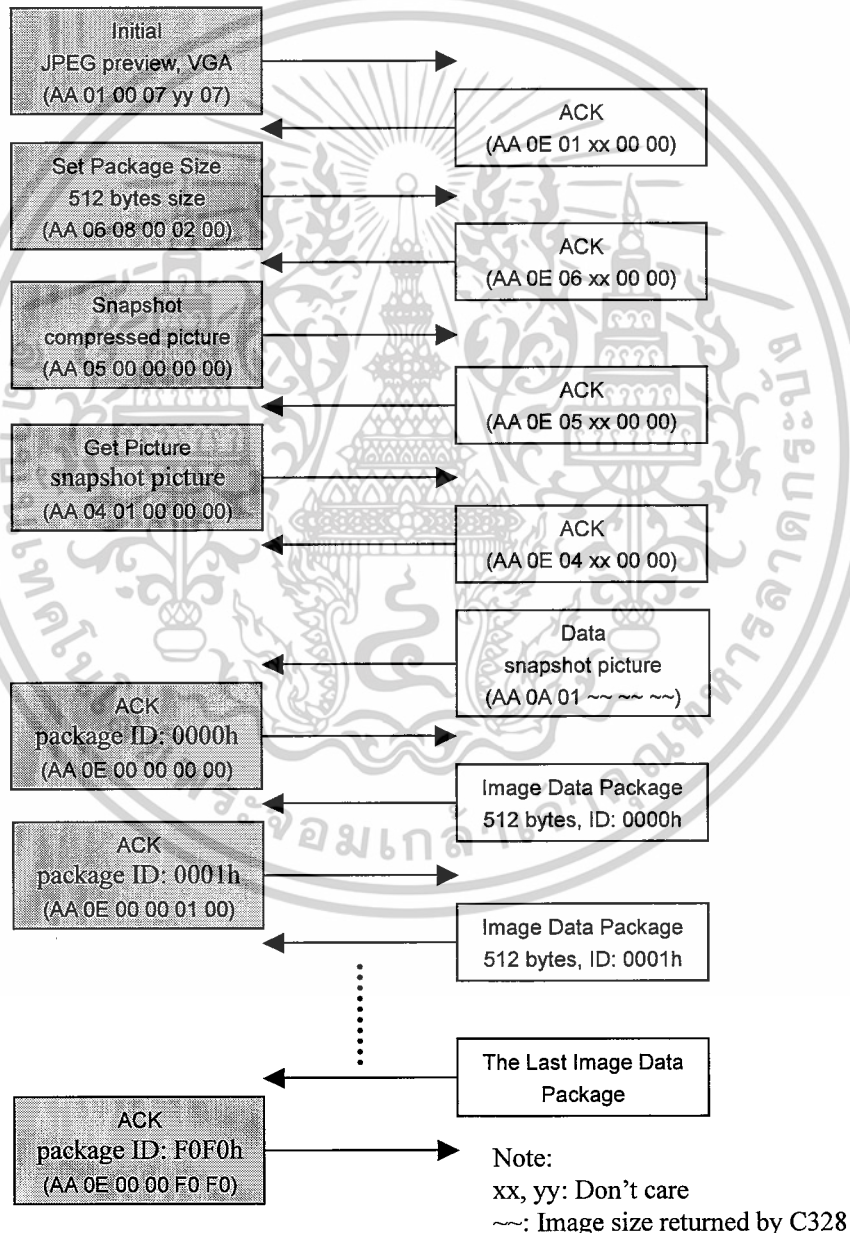
3. Initial, Get Picture, Snapshot, Set Package Size, Set Baudrate, Reset and Power Off Command



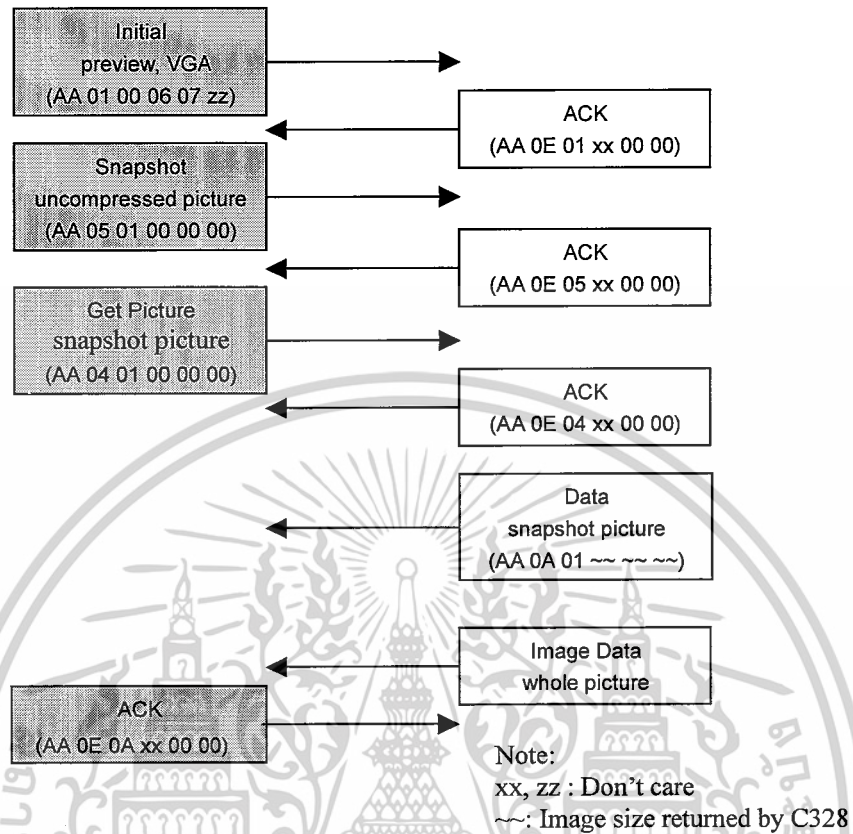
4. Getting a Snapshot for RS232

Make sure connection is made before the following communication.

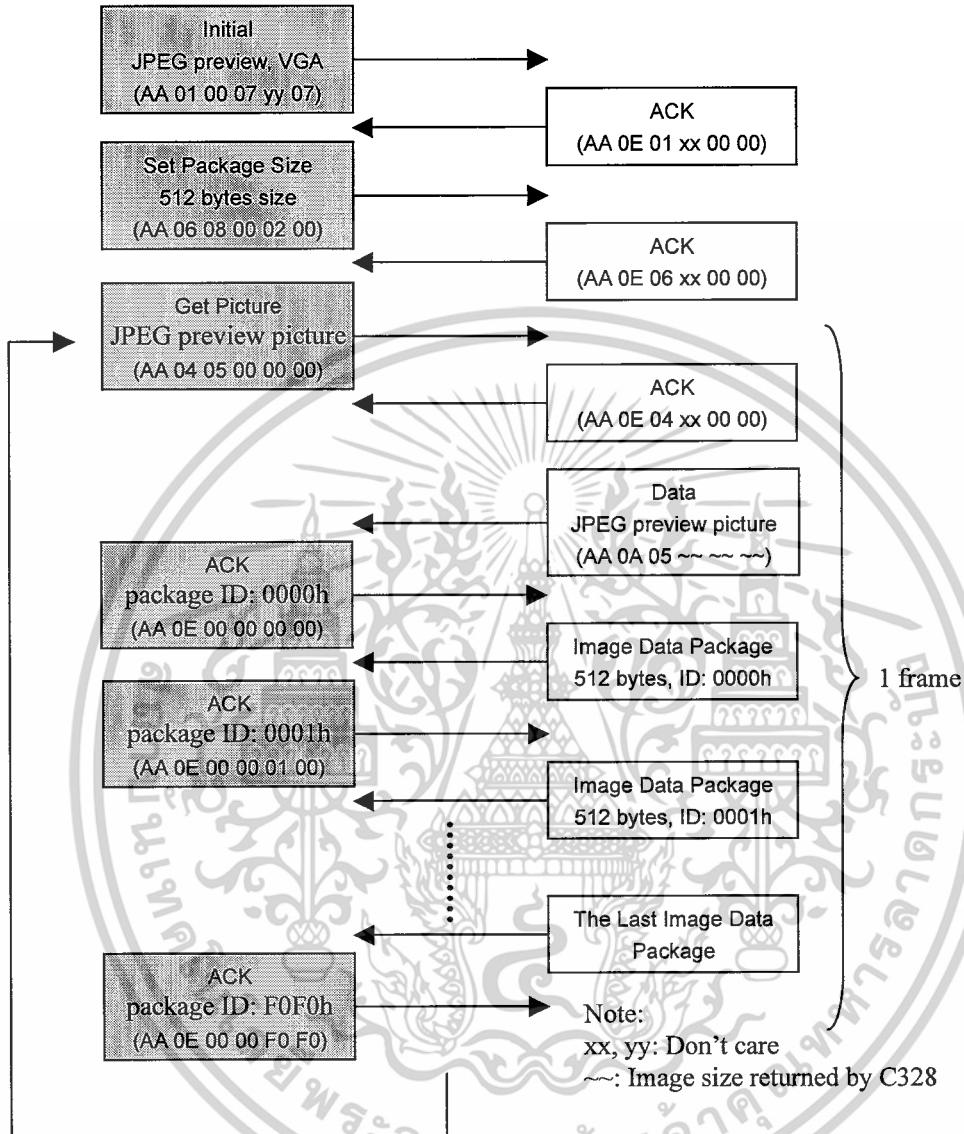
4.1 JPEG Snapshot Picture (eg. 640x480 resolution)



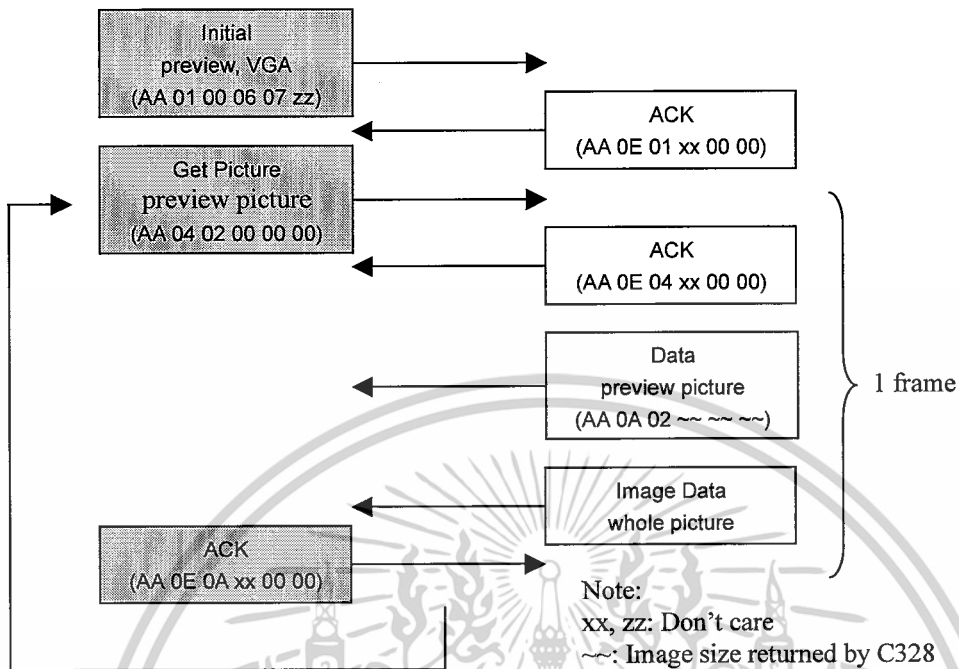
4.2 Snapshot Picture (uncompressed snapshot picture)



5. Getting JPEG preview pictures (video) for RS232
Make sure connection is made before the following communication.
- 5.1 JPEG Preview Picture

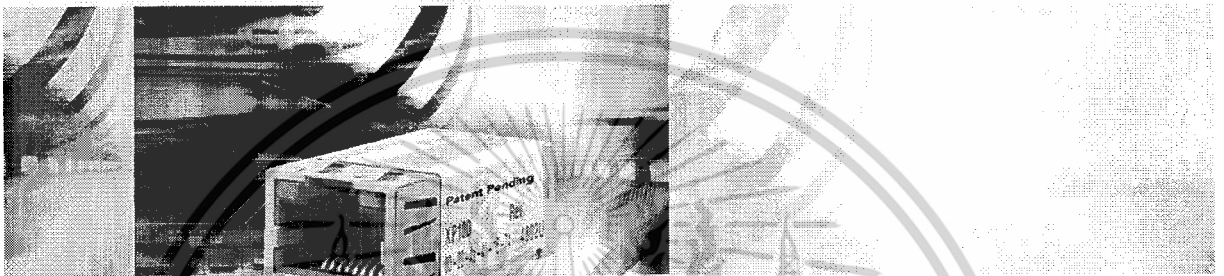


5.2 Preview Picture (uncompressed preview picture)



FAQ

- Q: What is the power range of the camera module?
A: The range is **+3.0V - +3.6V**.
- Q: I want to establish the connection between a PC and the camera module. Is there any configuration should be done?
A: To connection with a PC, a **RS-232 transceiver set-up** should be used as a communication interface.
- Q: I have sent an SYNC command to camera, but it has no response. How can I synchronize with the module?
A: Users should send the SYNC commands one by one continuously until receiving the ACK and SYNC commands from the module. Normally, **25-60 SYNC** commands are required. After that, users should reply with an ACK command.
- Q: What is the baud rate to synchronize with the camera? Will the baud rate change after SYNC?
A: C328 supports **7200, 9600, 14400, 19200, 28800, 38400, 57600 and 115200bps**. Users can synchronize with the camera at one of the baud rate above. Once synchronizing with camera successfully, the baud rate will not be changed until users change it with the "Set Baud rate" command.
- Q: When will the baud rate be changed after receiving the "Set Baud rate" command?
A: The baud rate will be changed after the module reply with the ACK command. Users must use the new baud rate after this.
- Q: After sending "Getpicture" command to the camera, what will the users receive?
A: After sending "Getpicture" command to the module, users will receive an "ACK", a "Data" command, "AA 0A 01 XX YY ZZ" telling you the image size, and then the first package of image data. .
- Q: How to use the image size returned? Also, how many packages must be received to get the captured image?
A: Users can use the image size to calculate the number of packages will be received according to the package size set. The equation is shown in the following:
Number of package = Image size / (Package size - 6)
- Q: According to the flow diagram, the ACK command for the first package is AA 0E 00 00 00 00 and that for the second one AA 0E 00 00 01 00. Is the third one AA 0E 00 00 02 00 or AA 0E 00 00 01 01?
A: For the third package, it should be AA 0E 00 00 02 00. Those for the other package are shown in the following:
AA 0E 00 00 L'L H'H.
L'L is the low byte of package ID
H'H is the high byte of package ID
i.e. ID = H'H L'L in hex
- Q: After synchronization, I got the first picture with too low to too high luminance. What's wrong with it?
A: After synchronization, the camera needs a little time for AEC and AGC to be stable. Users should wait for **1-2 seconds** before capturing the first picture.
- Q: What are the formats of the uncompressed pictures?
A: The formats are shown in the following:
2-bit Gray Scale: 2-bit for Y only
4-bit Gray Scale: 4-bit for Y only
8-bit Gray Scale: 8-bit for Y only
12-bit Color: 444 (RGB)
16-bit Color: 565 (RGB)



XPort™ Integration Guide

Copyright and Trademark

© 2005, Lantronix. All rights reserved. No part of the contents of this book may be transmitted or reproduced in any form or by any means without the written permission of Lantronix. Printed in the United States of America.

XPort, with its patent-pending technology, is a trademark of Lantronix.

Ethernet is a trademark of XEROX Corporation. UNIX is a registered trademark of The Open Group. Windows 95, Windows 98, Windows 2000, Windows NT, and Windows XP are trademarks of Microsoft Corp. Netscape is a trademark of Netscape Communications Corporation.

Contacts

Lantronix Corporate Headquarters

15353 Barranca Parkway
Irvine, CA 92618, USA
Phone: 949-453-3990
Fax: 949-453-3995

Technical Support

Fax: 949-450-7226
Online: www.lantronix.com/support
Email/Chat: www.lantronix.com/support

Sales Offices

For a current list of our domestic and international sales offices, go to the Lantronix web site at www.lantronix.com/about/contact/.

Disclaimer and Revisions

Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his or her own expense, will be required to take whatever measures may be required to correct the interference.

Note: This product has been designed to comply with the limits for a Class B digital device pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with this guide, may cause harmful interference to radio communications.

Changes or modifications to this device not explicitly approved by Lantronix will void the user's authority to operate this device.

Note: With the purchase of XPort, the OEM agrees to an OEM firmware license agreement that grants the OEM a non-exclusive, royalty-free firmware license to use and distribute the binary firmware image provided, only to the extent necessary to use the XPort hardware. For further details, please see the XPort OEM firmware license agreement.

| Date | Rev. | Comments |
|-------|------|--|
| 11/03 | A | Initial Release |
| 4/04 | B | Firmware 1.6 features; information to support XPort-03 |
| 6/04 | C | Technical specifications updated |
| 8/04 | D | Firmware 1.8 features; added XPort-485 information |
| 10/04 | E | Removed out of date manual references |
| 3/05 | F | Updated illustration |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Contents

| | |
|---|-----------|
| Copyright and Trademark | 2 |
| Contacts | 2 |
| Disclaimer and Revisions | 2 |
| 1: Introduction | 5 |
| About the Integration Guide | 5 |
| Additional Documentation | 5 |
| 2: Description and Specifications | 6 |
| The XPort | 6 |
| XPort Block Diagram | 6 |
| PCB Interface | 7 |
| Ethernet Interface | 9 |
| LEDs | 9 |
| Dimensions | 10 |
| Recommended PCB Layout | 11 |
| Product Information Label | 11 |
| Technical Specifications | 12 |
| 3: Development Kit | 14 |
| Contents of the Kit | 14 |
| Evaluation Board Description | 14 |
| Evaluation Board Block Diagram | 15 |
| Major Components | 15 |
| Serial RS-232 Interface | 15 |
| RS-232 Port | 16 |
| Power Supply | 16 |
| General Control PLD | 16 |
| Configuration Switch Bank | 16 |
| SW1-2 Options | 17 |
| SW1-3 Options | 18 |
| SW1-4, SW1-5 Options | 19 |
| Timer | 20 |
| Board Layout | 21 |
| Component Identification | 21 |
| A: XPort-485 Connection Diagram | 22 |
| B: Compliance and Warranty Information | 23 |
| Compliance Information | 23 |
| Warranty | 24 |

List of Figures

| | |
|--|----|
| Figure 2-1. Side View of the -03 XPort | 6 |
| Figure 2-2. XPort Block Diagram | 7 |
| Figure 2-3. XPort LEDs | 9 |
| Figure 2-4. Front View | 10 |
| Figure 2-5. Bottom View | 10 |
| Figure 2-6. Side View | 10 |
| Figure 2-7. PCB Layout | 11 |
| Figure 2-8. Product Label | 12 |
| Figure 3-1. SW1-2 OFF | 17 |
| Figure 3-2. SW1-2 ON | 18 |
| Figure 3-3. SW1-3 OFF | 18 |
| Figure 3-4. SW1-3 ON | 18 |
| Figure 3-5. SW1-5 OFF and SW1-4 ON | 19 |
| Figure 3-6. SW1-5 ON and SW1-4 ON | 19 |
| Figure 3-7. JP4 Configurable Pins Interface Header Footprint | 20 |
| Figure 3-8. Board Layout | 21 |

List of Tables

| | |
|--|----|
| Table 2-1. PCB Interface Signals | 7 |
| Table 2-2. Ethernet Interface Signals (Industry Standards) | 9 |
| Table 2-3. XPort LED Functions (XPort-03 and greater only) | 9 |
| Table 2-4. Technical Specifications | 12 |
| Table 3-1. RS-232 Signals | 15 |
| Table 3-2. Configuration Switch Settings | 17 |
| Table 3-3. CP1 Status Indicator | 18 |
| Table 3-4. LED States | 19 |
| Table 3-5. JP4 Connections | 20 |
| Table 3-6. Board Components | 21 |

1: Introduction

About the Integration Guide

This guide provides the information needed to test the XPort™ device server on the XPort Evaluation Board. This manual is intended for engineers responsible for integrating the XPort into their product.

Note: This document covers XPort™ Device Server versions XP1001000-01 (XPort-01), XP1001000-03 (XPort-03), and XP1004000-03 (XPort-485).

Additional Documentation

The following guides are available on the product CD and the Lantronix Web site (www.lantronix.com)

| | |
|---------------------------------------|--|
| XPort™ Getting Started | Provides the steps for getting the hardware and software up and running. |
| XPort™ User Guide | Provides information needed to configure, use, and update the XPort firmware. |
| DeviceInstaller User Guide | Provides instructions for using the Windows-based utility to configure the XPort and other Lantronix device servers. |
| Com Port Redirector User Guide | Provides information on using the Windows-based utility to create a virtual com port. |

In addition to this documentation, the supplied CD contains a configurable pins control web applet that lets you view the state of the configurable GPIO pins. This applet is provided in the Sample Code and Solutions folder.

For firmware versions 1.3 and earlier, the corresponding applet file is `xptcpctl1102.cob`. For later firmware versions (greater than version 1.3), the corresponding applet file is `xpt_gpioc1_154.cob`.

2: Description and Specifications

The XPort embedded device server is a complete network-enabling solution enclosed within an RJ45 package. This miniature serial-to-Ethernet converter empowers original equipment manufacturers (OEMs) to quickly and easily go to market with networking and web page serving capabilities built into their products.

The XPort

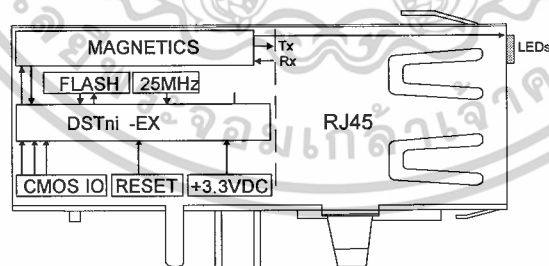
The XPort contains Lantronix's own DSTni controller, with 256 Kbytes of SRAM, 2 Kbytes of boot ROM for the XPort-01 or 16 Kbytes of boot ROM for the XPort-03 or greater, and integrated AMD 10/100 PHY.

The XPort also contains the following:

- ◆ 3.3-volt serial interface
- ◆ All I/O pins are 5V tolerant (XPort-03 or greater only)
- ◆ 4-Mbit flash memory
- ◆ Ethernet magnetics
- ◆ Power supply filters
- ◆ Reset circuit
- ◆ +1.8V regulator
- ◆ 25-MHz crystal and Ethernet LEDs

The XPort requires +3.3-volt power and is designed to operate in an extended temperature range (see technical data).

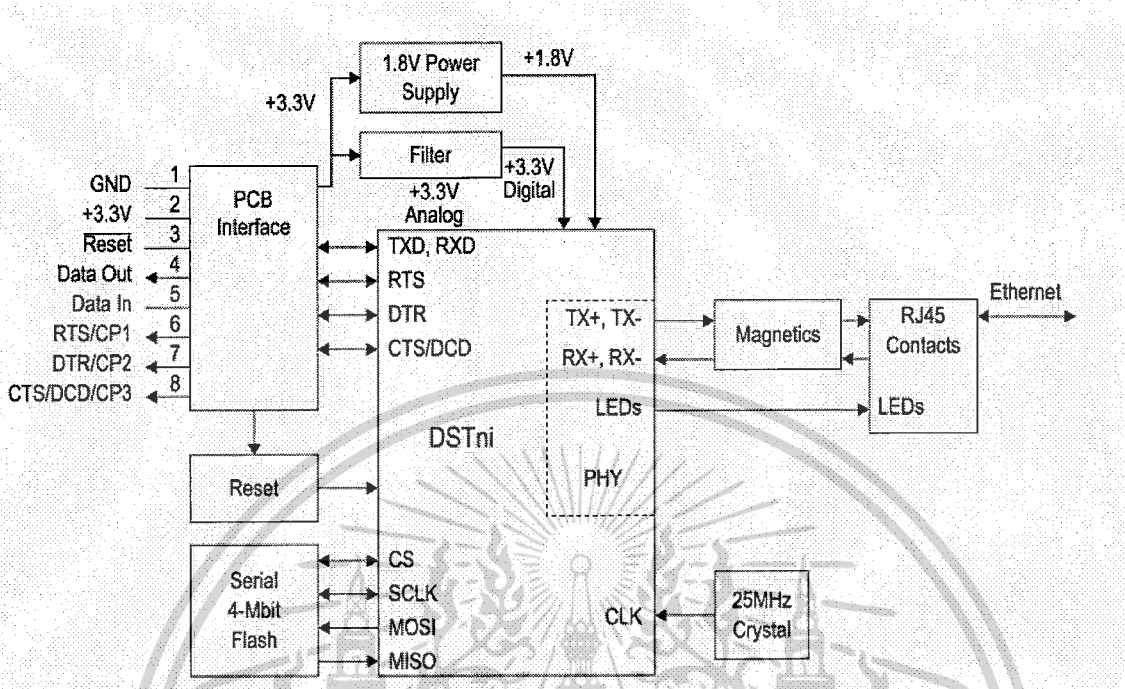
Figure 2-1. Side View of the -03 XPort



XPort Block Diagram

The following drawing is a block diagram of the XPort showing the relationships of the components.

Figure 2-2. XPort Block Diagram



PCB Interface

The XPort has a serial port compatible with data rates up to 920 kbps (in high-performance mode) for the XPort-03 and greater, and at 230 kbps for the XPort-01. The serial signals (pins 4–8) are 3.3V CMOS logic level (5V tolerant for the XPort-03 and greater only). The serial interface pins include +3.3V, ground, and reset. The serial signals usually connect to an internal device, such as a UART. For applications requiring an external cable running with RS-232 or RS422/485 voltage levels, the XPort must interface to a serial transceiver chip. We supply an RS-232 transceiver on the XPort Evaluation Board for this purpose. There is an XPort option available for interfacing with RS422/485.

Table 2-1. PCB Interface Signals

| Signal Name | XPort Pin # | Primary Function |
|---------------------------------|-------------|---|
| GND | 1 | Circuit ground |
| 3.3V | 2 | +3.3V power in |
| Reset | 3 | External reset in |
| Data Out | 4 | Serial data out (driven by DSTni's built-in UART) |
| Data In | 5 | Serial data in (read by DSTni's built-in UART) |
| CP1/RTS (Configurable Pin 1) | 6 | CP1 can be configured as follows: <ul style="list-style-type: none"> • Flow control: RTS (Request to Send) output driven by DSTni's built-in UART for connection to CTS of attached device. • Programmable input/output: CP1 can be driven or read through software control, independent of serial port activity. |

Description and Specifications

| Signal Name | XPort Pin # | Primary Function |
|-------------------------------------|-------------|---|
| CP2/DTR (Configurable Pin 2) | 7 | CP2 can be configured as follows: <ul style="list-style-type: none">• Modem control: DTR (Data Terminal Ready) output driven by DSTni's built-in UART for connection to DCD of attached device.• Programmable input/output: CP2 can be driven or read through software control, independent of serial port activity. |
| CP3/CTS/DCD (Configurable Pin 3) | 8 | CP3 can be configured as follows: <ul style="list-style-type: none">• • Flow control: CTS (Clear to Send) input read by DSTni's built-in UART for connection to RTS of attached device.• • Modem control: DCD (Data Carrier Detect) input read by DSTni's built-in UART for connection to DTR of attached device.• Programmable input/output: CP3 can be driven or read through software control, independent of serial port activity. |



Ethernet Interface

The Ethernet interface magnetics, RJ45 connector, and Ethernet status LEDs are all in the device server shell.

Table 2-2. Ethernet Interface Signals (Industry Standards)

| Signal Name | DIR | Contact | Primary Function |
|-------------|-----|---------|---------------------------------------|
| TX+ | Out | 1 | Differential Ethernet transmit data + |
| TX- | Out | 2 | Differential Ethernet transmit data - |
| RX+ | In | 3 | Differential Ethernet receive data + |
| RX- | In | 6 | Differential Ethernet receive data - |
| Not used | | 4 | Terminated |
| Not used | | 5 | Terminated |
| Not used | | 7 | Terminated |
| Not Used | | 8 | Terminated |
| SHIELD | | | Chassis ground |

LEDs

The XPort contains the following LEDs:

- ◆ Link (bi-color, left LED) — XPort-03 and greater only
- ◆ Activity (bi-color, right LED)

Figure 2-3. XPort LEDs

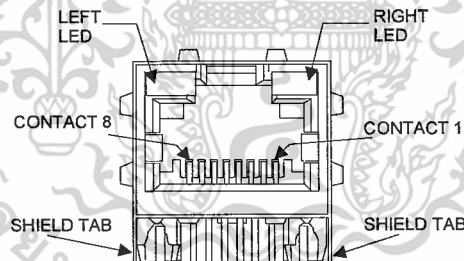


Table 2-3. XPort LED Functions (XPort-03 and greater only)

| Link LED Left Side | | Activity LED Right Side | |
|-----------------------|----------|----------------------------|-------------|
| Color | Meaning | Color | Meaning |
| Off | No Link | Off | No Activity |
| Amber | 10 Mbps | Amber | Half Duplex |
| Green | 100 Mbps | Green | Full Duplex |

Dimensions

The XPort dimensions are shown in the following drawings.

Figure 2-4. Front View

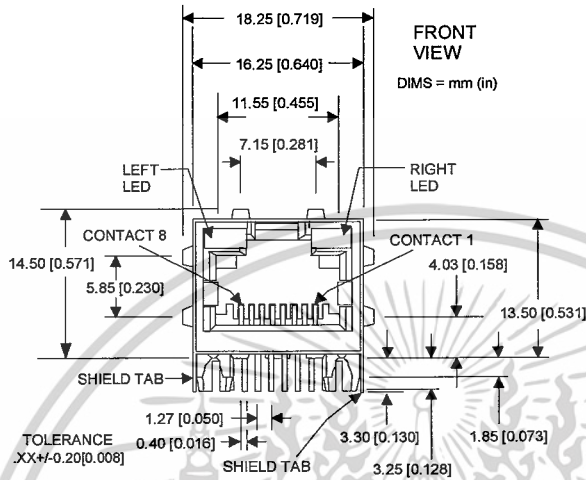


Figure 2-5. Bottom View

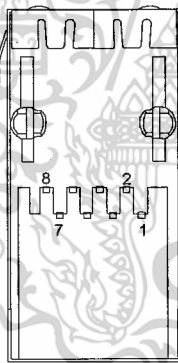
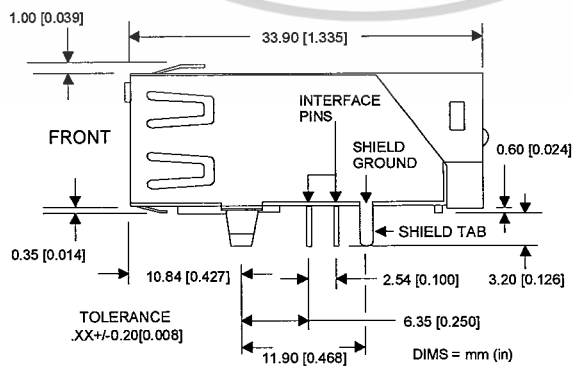


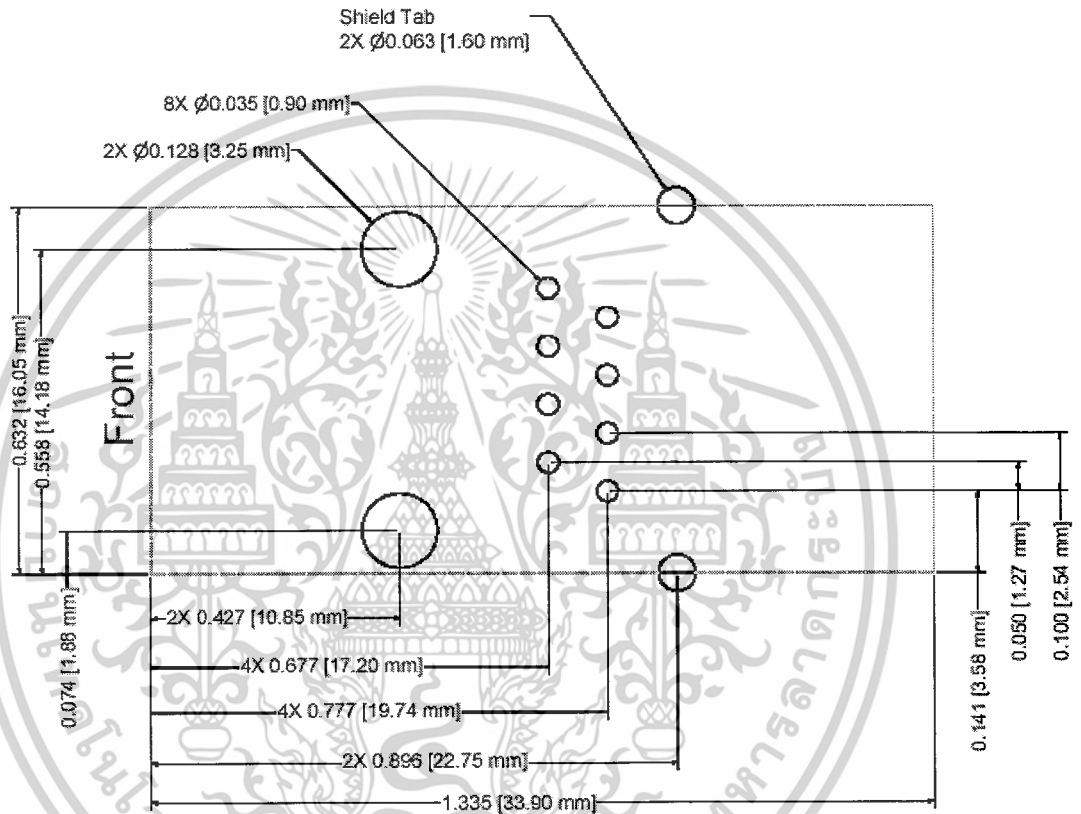
Figure 2-6. Side View



Recommended PCB Layout

The hole pattern and mounting dimensions for the XPort device server are shown in the following drawing. For proper heat dissipation, it is recommended that the PCB have approximately 1 square inch of copper attached to the shield tabs. The shield tabs are an important source of heat sinking for the device.

Figure 2-7. PCB Layout



Product Information Label

The product information label contains important information about your specific unit, such as its product ID (name), bar code, part number, and Ethernet (MAC) address.

Figure 2-8. Product Label



Technical Specifications

Table 2-4. Technical Specifications

| Category | Description |
|---------------------|--|
| CPU, Memory | Lantronix DSTni-EX 186 CPU, 256-Kbyte zero wait state SRAM, 512-Kbyte flash, 2-Kbyte boot ROM (XPort-01), 16-Kbyte boot ROM (XPort-03 and greater) |
| Firmware | Upgradeable via TFTP and serial port |
| Reset Circuit | Internal 200ms power-up reset pulse. Power-drop reset triggered at 2.6V. External reset input causes an internal 200ms reset. |
| Serial Interface | CMOS (Asynchronous) 3.3V-level signals Rate is software selectable: XPort-01: 300 bps to 230400 bps XPort-03 and greater: 300 bps to 921600 bps |
| Serial Line Formats | Data bits: 7 or 8 Stop bits: 1 or 2 Parity: odd, even, none |
| Modem Control | DTR/DCD, CTS, RTS |
| Flow Control | XON/XOFF (software), CTS/RTS (hardware), None |
| Programmable I/O | 3 PIO pins (software selectable), sink or source 4mA max. |
| Network Interface | RJ45 Ethernet 10Base-T or 100Base-TX (auto-sensing) |
| Compatibility | Ethernet: Version 2.0/IEEE 802.3 (electrical), Ethernet II frame type |
| Protocols Supported | ARP, UDP/IP, TCP/IP, Telnet, ICMP, SNMP, DHCP, BOOTP, TFTP, Auto IP, SMTP, and HTTP |
| LEDs | 10Base-T and 100Base-TX Link Activity, Full/half duplex. Software generated status & diagnostic signals can optionally drive external LEDs through CP1 & CP3. |
| Management | Internal web server, SNMP (read only) Serial login, Telnet login |
| Security | Password protection, locking features, optional Rijndael 256-bit encryption |
| Internal Web Server | Serves static web pages and Java applets Storage capacity: 384 Kbytes |
| Weight | 0.34 oz (9.6 grams) |
| Material | Metal shell, thermoplastic case |
| Temperature | Operating range: XPort-01: -40°C to +85°C (-40°F to 185°F) XPort-03 and greater: -40°C to +85°C (-40°F to 185°F) normal mode, -40°C to +75°C (-40°F to 167°F) high-performance mode |
| Shock/Vibration | Non-operational shock: 500 g's Non-operational vibration: 20 g's |

| Category | Description |
|-------------------|---|
| Warranty | One year limited warranty (-01 models) Two year limited warranty (-03 models) |
| Included Software | Windows™ 98/NT/2000/XP-based Device Installer configuration software and Windows™-based Com Port Redirector |
| EMI Compliance | Radiated and conducted emissions - complies with Class B limits of EN 55022:1998 Direct & Indirect ESD - complies with EN55024:1998 RF Electromagnetic Field Immunity - complies with EN55024:1998 Electrical Fast Transient/Burst Immunity - complies with EN55024:1998 Power Frequency Magnetic Field Immunity - complies with EN55024:1998 RF Common Mode Conducted Susceptibility - complies with EN55024:1998 |



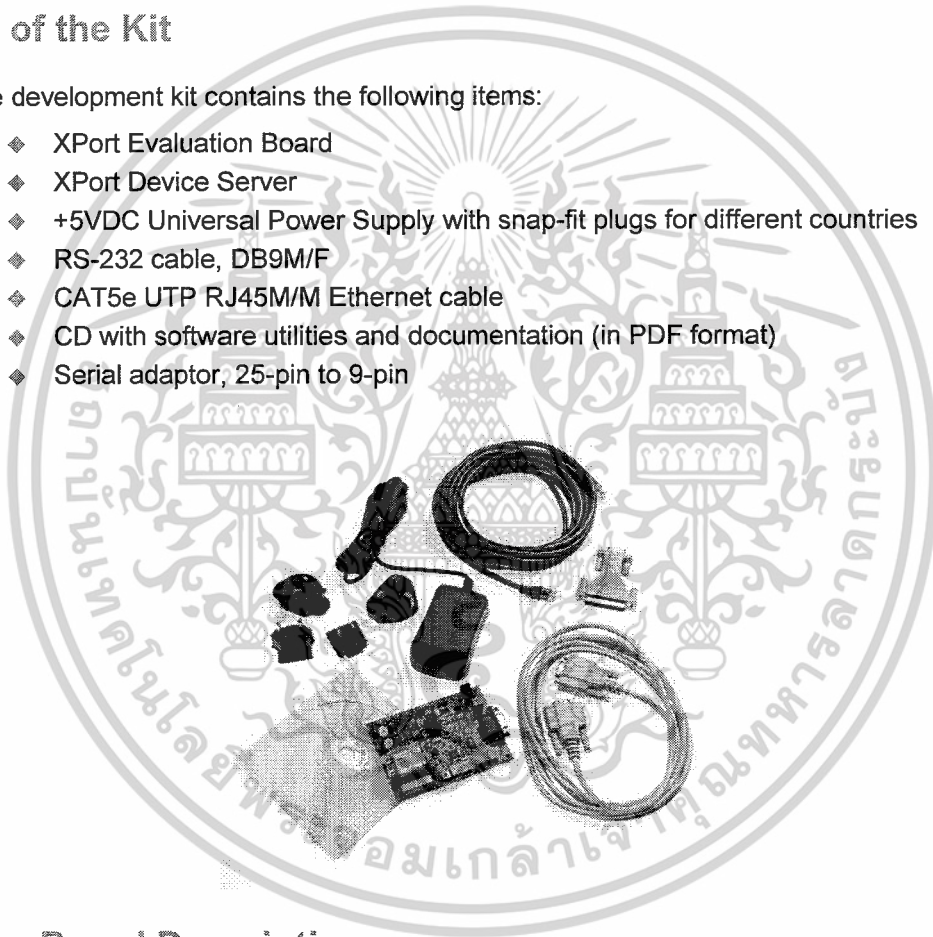
3: Development Kit

The XPort Development Kit includes everything you need to begin to integrate the XPort into your product design.

Contents of the Kit

The development kit contains the following items:

- ◆ XPort Evaluation Board
- ◆ XPort Device Server
- ◆ +5VDC Universal Power Supply with snap-fit plugs for different countries
- ◆ RS-232 cable, DB9M/F
- ◆ CAT5e UTP RJ45M/M Ethernet cable
- ◆ CD with software utilities and documentation (in PDF format)
- ◆ Serial adaptor, 25-pin to 9-pin



Evaluation Board Description

The XPort Evaluation Board provides a test platform for the Lantronix XPort device server.

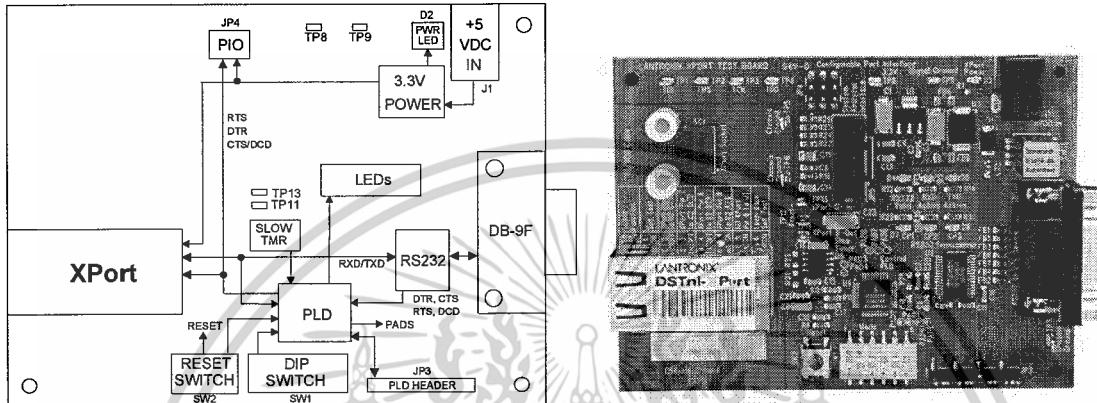
The XPort Evaluation Board supplies an RS-232 serial interface to 3.3V CMOS level interface connection for the XPort. It also supplies the XPort device server with +3.3V. Evaluation board controls include a programmable logic device (PLD), reset push button switch, DIP switch, timer circuit, and reset circuit.

In addition to the status LEDs on the XPort, the evaluation board contains multiple LEDs. The power supply drives the red LED. The PLD drives the remaining LEDs; in the basic mode of operation, these LEDs indicate whether the RS-232 interface is

valid, whether the serial interface transmit data and serial interface receive data are active, and the state of the three configurable pins, CP1, CP2, and CP3.

The PLD and a six-position DIP switch control the evaluation board mode of operation. The switch outputs are inputs to the PLD to select the desired mode of operation.

Evaluation Board Block Diagram



Major Components

The major components of the evaluation board include:

- ◆ RS-232 transceiver
- ◆ Universal Power Adapter
- ◆ Reset switch
- ◆ DIP switch
- ◆ DB9 interface connector (DCE)
- ◆ Test points to monitor all XPort pins
- ◆ 6-pin header (JP4) for connecting user devices or circuits to CP1, CP2, and CP3.
- ◆ Power, signal ground, and reset inputs are also available on JP4.

Serial RS-232 Interface

The RS-232 level serial interface is implemented with a transceiver that converts 3.3V CMOS levels to RS-232 levels. A straight-through serial cable with 9-pin connectors (M/F) is all you need to connect to a DTE device such as a PC.

The table below lists the RS-232 signals. Note that XPort signal pins 6, 7, and 8 are configurable pins you can optionally set for functions other than RS-232 control. The configuration DIP switch determines whether or not to route the XPort configurable pins to the RS-232 converter.

Table 3-1. RS-232 Signals

| XPort Signal | XPort Pin # | Direction | DB9 Pin # | DTE Device Signal |
|--------------|-------------|-----------|-----------|-------------------|
| DTR | 7 | Out | 1 | DCD |
| Data Out | 4 | Out | 2 | RXD |
| Data In | 5 | In | 3 | TXD |
| DCD | 8 | In | 4 | DTR |

| XPort Signal | XPort Pin # | Direction | DB9 Pin # | DTE Device Signal |
|--------------|-------------|-----------|-----------|-------------------|
| Ground | 1 | -- | 5 | GND |
| CTS | 8 | In | 7 | RTS |
| RTS | 6 | Out | 8 | CTS |

Note: The supplied CD contains a configurable pins control web applet that lets you view the state of the configurable GPIO pins. This applet is provided in the Sample Code and Solutions folder.

RS-232 Port

The evaluation board uses a TI SN75C3238 IC to convert all RS-232 interface signals to 3.3V CMOS-level XPort signals.

We selected the interface IC for its capability of true RS-232-level performance from a +3.3V power source and for its 15 kV ESD protection. The part also includes an RS-232 valid output that lights an LED by means of the PLD circuit.

Power Supply

The evaluation board contains a +3.3V regulator that receives input power by means of a 2.5mm input power jack. The input power should be 5VDC from a regulated DC source. The evaluation kit provides a +5VDC power module.

We recommend that you also connect the evaluation board to an earth ground. You can use the chassis ground rectangle (copper tape), TP7, or the shell of the DB9 connector.

General Control PLD

The purpose of the PLD is to provide general evaluation board mode control and LED signal-level monitoring. The PLD works with the 6-position mode control switch (SW1) and the timer and reset circuits to provide mode control.

The PLD is a Lattice ISPLSI2032VE part, which is contained in a 48-pin TQFP package. This part contains 32 I/O and 32 microcells and is in-circuit programmable. A 1x8-pin header (JP3) provides for standard Lattice in-circuit programming.

Configuration Switch Bank

Switch module SW1 contains six independent switches for configuring the PLD on the evaluation board. The PLD controls how the signals from the XPort are routed on the evaluation board. Positions 1 and 6 are for self-test. Positions 2 through 5 are for configuring XPort pins for various operations.

Three XPort pins are configurable pins. You can configure these pins for hardware flow control or as LED status indicators on the evaluation board. For information about configuring these pins, see the *XPort User Guide*.

On the evaluation board, configurable pins are shown as CP1, CP2, CP3, and SW1 switch positions are shown as SW1-1 through SW1-6.

Note: In the following sections, the terms LED1 and LED3 refer to diagnostic modes. You can put control of two of the XPort pins into these modes.

Table 3-2. Configuration Switch Settings

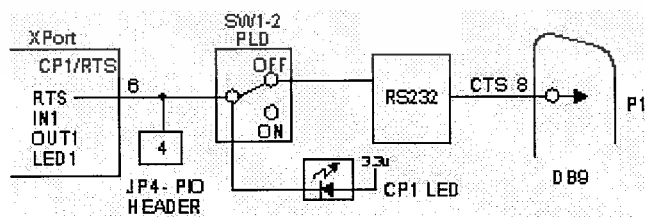
| SW1 | POS | Function |
|-----|-----|--|
| 1 | ON | SW1-1 and SW1-6 ON for self-test mode. (Factory test) |
| | OFF | Normal mode. |
| 6 | ON | SW1-1 and SW1-6 ON for self-test mode. (Factory test) |
| | OFF | Normal mode. |
| 2 | OFF | XPort CP1/RTS (pin 6) is connected to the RS-232 transceiver, which connects to the CTS of an attached device through DB9 pin 8. Use this setting if CP1 is set up for hardware flow control. |
| | ON | XPort CP1/RTS (pin 6) is <i>not</i> connected to an RS-232 transceiver. Use this setting if CP1 is set up as an IN1, OUT1, or LED1. |
| 3 | OFF | XPort CP2/DTR (pin 7) is connected to the RS-232 transceiver, which connects to DCD of an attached device through DB9 pin 1. Use this setting if CP2 is set up for hardware handshaking. |
| | ON | XPort CP2/DTR (pin 7) is <i>not</i> connected to the transceiver. Use this setting if CP2 is set up as an IN2 or OUT2. |
| 4 | OFF | Use this setting when XPort CP3/CTS/DCD (pin 8) is set up for hardware handshaking. With SW1-5 configured appropriately, CP3 will connect to DTR of an attached device. |
| | ON | Use this setting when XPort CP3/CTS/DCD (pin 8) is set up for hardware flow control. With SW1-5 configured appropriately, CP3 will connect to RTS of an attached device. |
| 5 | OFF | XPort CP3/CTS/DCD (pin 8) is connected to the RS-232 transceiver, which connects to either the DTR (DB9 pin 4) or the RTS (DB9 pin 7) of an attached device, depending on the setting on SW1-4. Use this setting if CP3 is set up for hardware flow control. |
| | ON | XPort CP3/CTS/DCD (pin 8) is not connected to the transceiver. Use this setting if CP3 is set up as an IN3, OUT3, or LED3. |

SW1-2 Options

SW1-2 controls the routing of the CP1/RTS (configurable pin 1) signal from the XPort. CP1 is connected to pin 6 of the XPort and can be software configured as CTS, IN1, OUT1, or LED1. SW1-2 is an input to the PLD, which does the actual switching. The drawings represent the logical switching function.

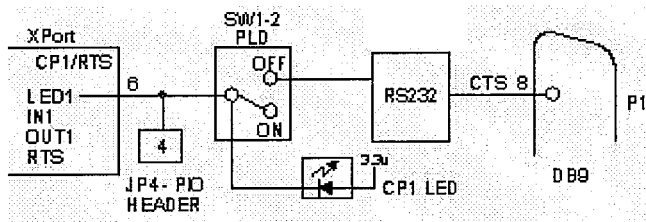
In this drawing, SW1-2 is OFF, which connects XPort pin 6 to the RS-232 transceiver. The XPort configurable pin 1 (CP1) is configured for CTS.

Figure 3-1. SW1-2 OFF



In the next drawing, SW1-2 is ON, which disconnects pin 6 from the RS-232 transceiver. The XPort configurable pin 1 (CP1) is configured for LED1.

Figure 3-2. SW1-2 ON



When configurable pin 1 is configured for LED1, it functions as a status indicator for the serial port.

Table 3-3. CP1 Status Indicator

| Condition | CP1 LED State |
|----------------------|--------------------------|
| Idle channel | Solid on |
| Connected to network | 4 blinks every 4 seconds |

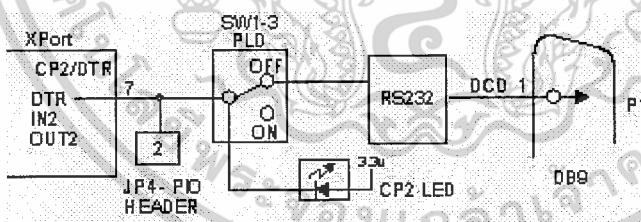
CTS and RTS work together for hardware flow control. Configure CP3 as RTS when CP1 is configured as CTS. Select hardware flow control as described in *Flow* in the *XPort User Guide*. See the *Device Installer User Guide* for information on configuring CP1 as IN1 or OUT1.

SW1-3 Options

SW1-3 controls the routing of the CP2/DTR (configurable pin 2) signal from the XPort. CP2 is connected to pin 7 of the XPort and can be software-configured as DCD, IN2, or OUT2. SW1-3 is an input to the PLD that does the actual switching. The drawings represent the logical switching function.

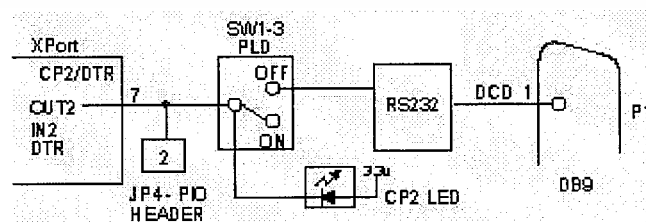
In this drawing, SW1-3 is OFF, which connects XPort pin-7 to the RS-232 transceiver. The XPort CP2 is configured for DCD.

Figure 3-3. SW1-3 OFF



In the drawing below, SW1-3 is ON, which disconnects XPort pin-7 from the RS-232 transceiver. The XPort CP2 is configured for OUT2.

Figure 3-4. SW1-3 ON



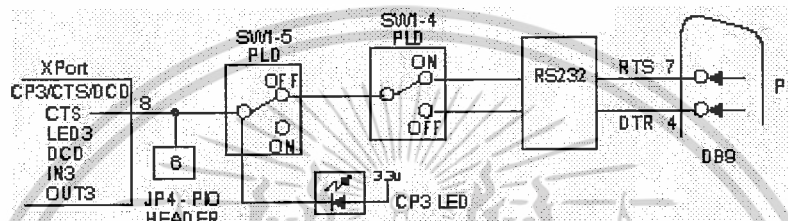
See the *Device Installer User Guide* for information on configuring CP2 as IN2 or OUT2.

SW1-4, SW1-5 Options

SW1-5 controls the routing of the CP3/CTS/DCD signal from the XPort. CP3 is connected to pin 8 of the XPort, and you can software configure it as RTS, LED3, DTR, IN3, or OUT3. With SW1-5 in the OFF position, the CP3 signal is routed to SW1-4. SW1-4 and SW1-5 are inputs to the PLD, which does the actual switching. The drawings represent the logical switching functions.

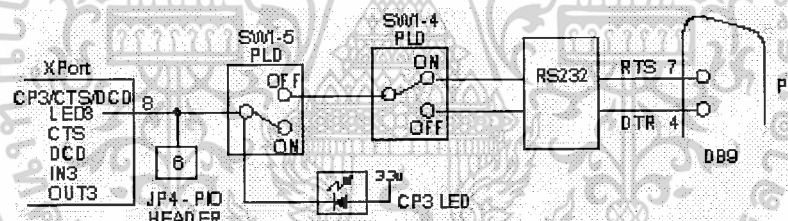
CP3 options are a little more complicated because both SW1-4 and SW1-5 are used in the configuration setup. In this drawing, SW1-5 is OFF, which connects XPort pin-8 to the RS-232 transceiver. The XPort CP3 is configured for RTS. SW1-4 is ON, routing the signal from XPort pin-8 to P1 pin-7 (RTS).

Figure 3-5. SW1-5 OFF and SW1-4 ON



In the next drawing, SW1-5 is ON, which disconnects XPort pin-8 from the RS-232 transceiver. The XPort CP3 is configured for LED3.

Figure 3-6. SW1-5 ON and SW1-4 ON



When CP3 is configured for LED3, it functions as a diagnostic indicator. The LED3 signal in combination with the LED1 signal indicates diagnostic information as shown in the following table.

Note: CP1 must be configured for LED1, and CP3 must be configured for LED3 for diagnostic mode.

Table 3-4. LED States

| Condition | CP3 LED (LED3) | CP1 LED (LED1) |
|------------------------------|----------------|--------------------|
| No errors | OFF | ON |
| Network controller error | ON | Blink 3x/4 sec OFF |
| Duplicate IP address present | ON | Blink 5x/4 sec OFF |
| No DHCP response | Blink 2x/sec | Blink 5x/4 sec OFF |
| Setup menu active | Blink 2x/sec | See Note. |

Note: During a Telnet connection, CP1 LED (LED1) is ON. For a serial port connection, CP1 LED (LED 1) blinks for 2 seconds, then OFF for 2 seconds. (It appears as 4 blinks, then OFF for 2 seconds)

CTS and RTS work together for hardware flow control. Configure CP3 as RTS when CP1 is configured as CTS. Select hardware flow control as described in *Flow* in the *XPort User Guide*.

See the *Device Installer User Guide* for information on configuring CP3 as IN3 or OUT3. JP4 can be used to monitor XPort pins 6 (CP1), 7 (CP2), and 8 (CP3). You can also use it to connect external circuitry or LEDs to the evaluation board.

Figure 3-7. JP4 Configurable Pins Interface Header Footprint

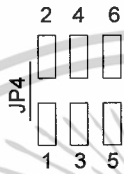


Table 3-5. JP4 Connections

| JP4 Pin # | Connects To |
|-----------|------------------------------|
| 1 | 3.3V |
| 2 | XPort pin 7, CP2 |
| 3 | Signal ground |
| 4 | XPort pin 6, CP1 |
| 5 | External reset – XPort pin 3 |
| 6 | XPort pin 8, CP3 |

Timer

The timer circuit is a National LMC555 IC. Its purpose is to generate a timer clock (approximately 10 counts per second). This clock is routed to the input clock pin of the PLD to be used for state machine and counter functions.

Board Layout

Component Identification

Figure 3-8. Board Layout

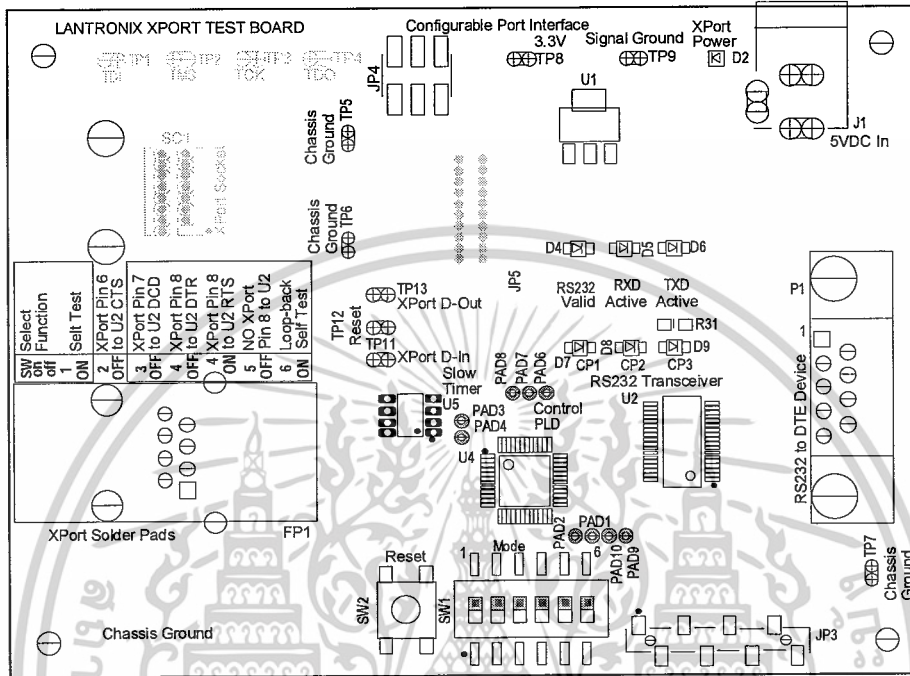
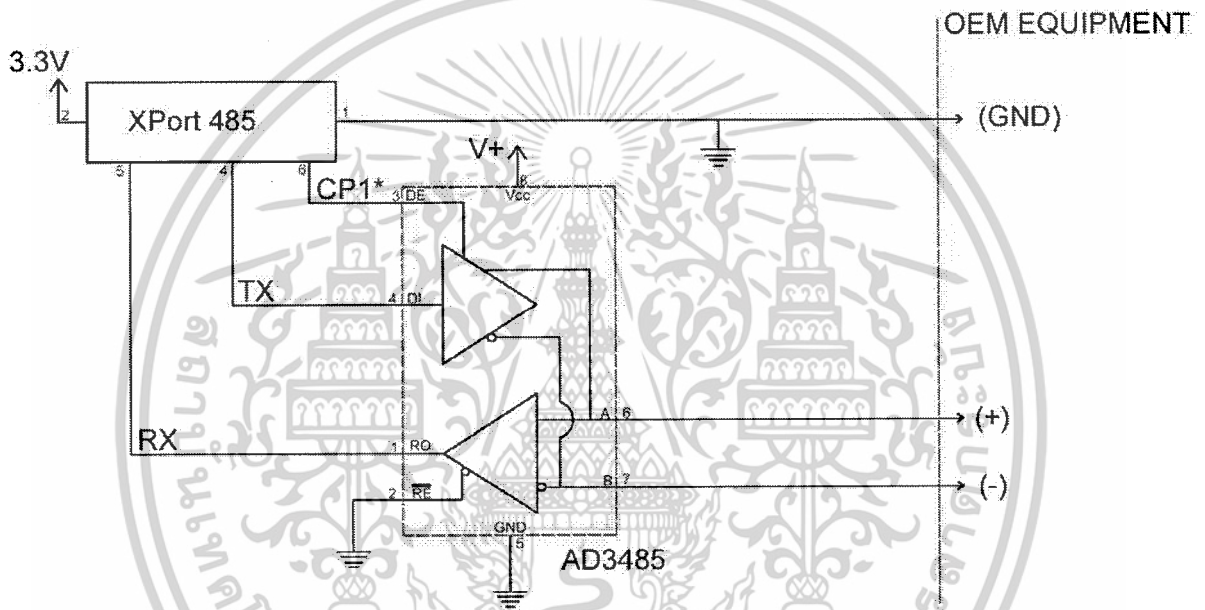


Table 3-6. Board Components

| Label | Function | Label | Function |
|-------|--|-------|-----------------------|
| P1 | RS-232 interface. DB9F connector | TP1 | NA |
| J1 | +5VDC input connector | TP2 | NA |
| SC1 | NA | TP3 | NA |
| SW1 | Mode switch | TP4 | NA |
| SW2 | Reset switch | TP5 | Chassis ground |
| D2 | XPort power (red) (LED) | TP6 | Chassis ground |
| D4 | RS-232 valid (green) (LED) | TP7 | Chassis ground |
| D5 | RXD active (green) (LED) | TP8 | XPort 3.3VDC (3V3) |
| D6 | TXD active (green) (LED) | TP9 | Signal ground |
| D7 | CP1- XPort pin 6 goes low (LED) | | |
| D8 | CP2- XPort pin 7 goes low (LED) | TP11 | XPort pin 5, data in |
| D9 | CP3- XPort pin 8 goes low (LED) | TP12 | XPort pin 3, reset |
| JP3 | Programming for PLD | TP13 | XPort pin 4, data out |
| JP4 | Configurable pins interface header connector | FP1 | XPort solder pads |
| JP5 | Factory test – no connector | | |

A: XPort-485 Connection Diagram

The following example illustrates a connection between the XPort-485 to an external transceiver IC:



* Any one of CP1, CP2 or CP3 can be used to control the TX enable.

B: Compliance and Warranty Information

Compliance Information

(According to ISO/IEC Guide 22 and EN 45014)

Manufacturer's Name & Address:

Lantronix 15353 Barranca Parkway, Irvine, CA 92618 USA

Declares that the following product:

Product Name Model: XPort Embedded Device Server

Conforms to the following standards or other normative documents:

Electromagnetic Emissions:

EN55022: 1998 (IEC/CSP1R22: 1993) Radiated RF emissions, 30MHz-1000MHz

Conducted RF Emissions – Telecom Lines – 150 kHz – 30 MHz

FCC Part 15, Subpart B, Class B

IEC 1000-3-2/A14: 2000

IEC 1000-3-3: 1994

Electromagnetic Immunity:

EN55024: 1998 Information Technology Equipment-Immunity Characteristics

Direct ESD, Contact Discharge

Indirect ESD

Radiated RF Electromagnetic Field Test

Electrical Fast Transient/Burst Immunity

RF Common Mode Conducted Susceptibility

Power Frequency Magnetic Field Test

Manufacturer's Contact:

Director of Quality Assurance, Lantronix

15353 Barranca Parkway, Irvine, CA 92618 USA

Tel: 949-453-3990

Fax: 949-453-3995

Warranty

Lantronix warrants each Lantronix product to be free from defects in material and workmanship for a period of ONE YEAR (-01 models) or TWO YEARS (-03 models). During this period, if a customer is unable to resolve a product problem with Lantronix Technical Support, a Return Material Authorization (RMA) will be issued. Following receipt of a RMA number, the customer shall return the product to Lantronix, freight prepaid. Upon verification of warranty, Lantronix will -- at its option -- repair or replace the product and return it to the customer freight prepaid. If the product is not under warranty, the customer may have Lantronix repair the unit on a fee basis or return it. No services are handled at the customer's site under this warranty. This warranty is voided if the customer uses the product in an unauthorized or improper way, or in an environment for which it was not designed.

Lantronix warrants the media containing its software product to be free from defects and warrants that the software will operate substantially according to Lantronix specifications for a period of 60 DAYS after the date of shipment. The customer will ship defective media to Lantronix. Lantronix will ship the replacement media to the customer.

In no event will Lantronix be responsible to the user in contract, in tort (including negligence), strict liability or otherwise for any special, indirect, incidental or consequential damage or loss of equipment, plant or power system, cost of capital, loss of profits or revenues, cost of replacement power, additional expenses in the use of existing software, hardware, equipment or facilities, or claims against the user by its employees or customers resulting from the use of the information, recommendations, descriptions and safety notations supplied by Lantronix. Lantronix liability is limited (at its election) to:

- ◆ Refund of buyer's purchase price for such affected products (without interest).
- ◆ Repair or replacement of such products, provided that the buyer follows the above procedures.

There are no understandings, agreements, representations or warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose, other than those specifically set out above or by any existing contract between the parties. Any such contract states the entire obligation of Lantronix. The contents of this document shall not become part of or modify any prior or existing agreement, commitment or relationship.

For details on the Lantronix warranty replacement policy, go to our web site at www.lantronix.com/support/warranty.

หนังสืออ้างอิง

1. สุวัฒน์ ปุณณชัยยะ, ต้น ต้นศัพท์สุทธิวงศ์, สุพจน์ ปุณณชัย, “เปิดโลก TCP/IP และโปรโตคอลของอินเทอร์เน็ต”, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), กรุงเทพฯ , 2545
2. ทีมงานสมาร์ทเลิร์นนิ่ง, “PIC Microcontroller Learning-By-Doing ด้วยภาษา C ฉบับรวมเครื่องโปรแกรมและอุปกรณ์ครบชุด”, สมาร์ทเลิร์นนิ่ง , กรุงเทพฯ, 2550
3. ดอนสัน ปงผาบ, ทิพวัลย์ คำน้ำนอง, “ไมโครคอนโทรลเลอร์ PIC และการประยุกต์ใช้งาน”, สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น), กรุงเทพฯ, 2550
4. ประจัน พลังสันติกุล, “PIC Works Example and C Source Code”, บริษัท แอปซอพท์เทค จำกัด, กรุงเทพฯ, 2547
5. http://www.ett.co.th/product/LAN/ET_XPORT_V1/et-xport_v1_00275.htm



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้