

ระบบควบคุมความปลอดภัยระยะไกลผ่านทางอินเทอร์เน็ต

Remote Security Control System Via Internet



โดย
นายเชxonันต์ สุโนนทอง
นายตุลา เล็กบุญญาสิน
นายปัญญา มลาลภรณ์

เลขที่.....
เลขทะเบียน 46298
วัน, เดือน, ปี 21 ส.ค. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมความปลอดภัยระยะไกลผ่านทางอินเทอร์เน็ต

Remote Security Control System Via Internet

โดย

นายเดชนันต์ สุโนนทอง เลขประจำตัว 41014145

นายศุภา เต็กบุญญาสิน เลขประจำตัว 41014152

นายปัญญา มาลาภรณ์ เลขประจำตัว 41014266

อาจารย์ที่ปรึกษา

รศ.ดร.รัตติกกร วรากุลศิริพันธุ์

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์เรื่อง

ระบบควบคุมความปลอดภัยระยะไกลผ่านทางอินเทอร์เน็ต

Remote Security Control System Via Internet

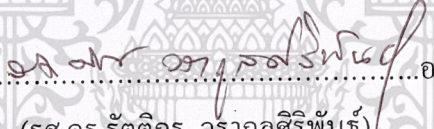
จัดทำโดย

นายเชxonันต์ สุโนนทอง รหัสประจำตัว 41014145

นายตุลา เต็กบุญญาสิน รหัสประจำตัว 41014152

นายปัญญา มาลาภรณ์ รหัสประจำตัว 41014266

ปริญญานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ..........อาจารย์ที่ปรึกษา

(รศ.ดร.รัตติกร วรากุลศิริพันธุ์)

วันที่ 27 มี.ค. ๕๕

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมความปลอดภัยระยะไกลผ่านทางอินเทอร์เน็ต

นายเชชนันต์ สุโนนทอง รหัส 41014145
นายศุลา เล็กบุญญาสิน รหัส 41014152
นายปัญญา มาลาภรณ์ รหัส 41014266
รศ.ดร.รัตติกร วรากุลศิริพันธ์ (อาจารย์ที่ปรึกษา)
ปีการศึกษา 2544

บทคัดย่อ

โครงการนี้นำเสนอระบบควบคุมความปลอดภัยระยะไกลผ่านทางอินเทอร์เน็ต ซึ่งในปัจจุบันนี้ อินเทอร์เน็ต ได้เข้ามามีบทบาทในชีวิตประจำวันของเรามากขึ้น จากประโยชน์ที่สามารถติดต่อสื่อสารกันได้อย่างง่ายดาย ไม่ว่าจะอยู่ที่ไหนในโลก เราจึงได้นำมาประยุกต์ใช้งานในด้านการสื่อสารของโครงการนี้โดยนำมาใช้ในการควบคุมกล้องวีดีโอระยะไกลและการส่งข้อมูลภาพและเสียง เพื่อใช้สำหรับระบบรักษาความปลอดภัย รายละเอียดของโครงการนี้ได้แก่ ส่วนไมโครคอนโทรลเลอร์ซึ่งใช้ติดตั้งกล้องวีดีโอ โดยใช้สเต็ปมอเตอร์ในการควบคุมตำแหน่งของกล้องและมีเซนเซอร์อัลตราโซนิกคอยตรวจจับการเคลื่อนไหวของเป้าหมาย และส่วนของ Client กับ Server ซึ่งใช้ในการส่งข้อมูลเพื่อควบคุมตำแหน่งของกล้องและใช้ในการส่งภาพและเสียงจาก Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote security control system via internet

Mr.Det-Anan Sunonthong 41014145

Mr.Tula Lekboonyasin 41014152

Mr.Panya Malaporn 41014266

Mr.Rattikorn Waragulsiripan (Advisor)

Education Year 2001

Abstract

This project present remote security system via internet ,because nowsaday the internet extremely become useful in every day life from useful in communication easily , wherever we are .so then we will apply it in this project. For remote control the video camera and transmit picture and voice , use in security system . This project consist of a microcontroller (MCS-51) for install the video camera and have two stepping motors for control the video camera and the ultrasonic sensors for detect the motion of object and part of client and server for control the microcontroller and transmit picture and voice.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี ต้องขอขอบพระคุณ คุณพ่อ คุณแม่และครอบครัวที่
อบอุ่นของคณะผู้จัดทำเป็นอย่างยิ่ง ที่คอยให้ความสนับสนุนเรื่องราวต่างๆด้วยความรักมาโดย
ตลอด

ขอขอบคุณอาจารย์ทุกๆ ท่านที่ช่วยอบรมบ่มเพาะมาตั้งแต่วัยเยาว์นับย้อนไปตั้งแต่ลืมตา
โลก พวกผมเชื่อว่าทุกท่านมีส่วนในความเป็นพวกผมทุกวันนี้จริงๆ

ขอขอบคุณเป็นพิเศษสำหรับอาจารย์ที่ปรึกษา รศ.ดร. รัตติกร วรากุลศิริพันธุ์ สำหรับการ
สนับสนุนเป็นอย่างดีในเรื่องต่างๆ และสำหรับความใจดีมีเมตตาของอาจารย์ต่อความผิดพลาดของ
พวกผม ผมซาบซึ้งเป็นอย่างยิ่งครับ

ขอขอบคุณรุ่นพี่ทุกๆท่าน ประสบการณ์และความรู้ที่เหนือกว่าของพวกพี่ช่วยพวกผมได้
มากจริงๆ และสำหรับการดูแลอย่างใกล้ชิดพวกผมซาบซึ้งมากครับ ถ้าไม่มีพี่ผมคงจะแย่เลย

ขอบคุณเพื่อนๆทุกคนที่อยู่ทำงานเป็นเพื่อนกันมาตลอด รวมถึงให้ความช่วยเหลือแนะนำ
ทุกๆครั้งที่ต้องการ และทำให้พวกผมได้เรียนรู้ว่าคนเราเก่งกันได้ด้วยความพากเพียรและใจรัก...

ความพยายามของทุกคนมีค่าสำหรับพวกผมจริงๆครับ

สุดท้ายนี้ต้องขอขอบคุณทุกๆท่านอีกครั้ง แม้ว่าจะพยายามไม่เอ่ยชื่อด้วยเกรงว่าจะตกหล่น
ด้วยเชื่อว่าทุกท่านมีส่วนในประสบการณ์อันมีค่าของพวกผม พวกผมซึ้งใจและขอบคุณเป็นอย่าง
ยิ่งครับ

นาย เศษอนันต์ สุโนนทอง

นาย ตูลา เล็กบุญญาสิน

นาย ปิฎญา มาลาภรณ์

สารบัญ

บทคัดย่อ	หน้า
ABSTRACT	
กิตติกรรมประกาศ	
สารบัญ	
สารบัญภาพ	
สารบัญตาราง	
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 เครือข่ายอินเทอร์เน็ต	3
2.1 สถาปัตยกรรม อินเทอร์เน็ต	3
2.2 OSI โมเดล	4
2.3 โครงสร้างชั้น Network	4
2.4 รูปแบบมาตรฐาน โปรโตคอลของอินเทอร์เน็ต	7
2.5 Internet IP	7
บทที่ 3 หลักการ โปรโตคอลของ TCP/IP	12
3.1 Process Layer	13
3.2 Host-to-Host Layer	14
3.3 โปรโตคอล TCP	16
3.4 โปรโตคอล UDP	17
3.5 Internetwork Layer	18
3.6 โปรโตคอล IP	19
3.7 โปรโตคอล ICMP	19
3.8 โปรโตคอล ARP	20
3.9 Network Interface Layer	20
3.10 กลไกของโปรโตคอล IP	22
บทที่ 4 พื้นฐาน ไมโครคอนโทรลเลอร์ MCS-51 และสตีปเปอร์มอเตอร์	26
4.1 ความหมายของไมโครคอนโทรลเลอร์	26

เอกสารนี้เป็น 4.2 โครงสร้างของ MCS-51 ที่ใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 คุณสมบัติที่สำคัญของ MCS-51	27
4.4 หน้าที่การใช้งานของแต่ละขาของ MCS-51	27
4.5 โครงสร้างหน่วยความจำใน MCS-51	28
4.6 รีจิสเตอร์ฟังก์ชันพิเศษ	29
4.7 สเต็ปเปอร์มอเตอร์	31
4.8 ชนิดของสเต็ปเปอร์มอเตอร์	32
4.9 การกระตุ้นและการควบคุมการหมุนของสเต็ปเปอร์มอเตอร์	32
บทที่ 5 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51	34
5.1 การสื่อสารข้อมูลแบบอะซิงโครนัส	34
5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51	35
5.3 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51	36
5.4 อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51	37
5.5 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์	38
บทที่ 6 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรมของคอมพิวเตอร์และ การส่งข้อมูลภาพและเสียงรวมทั้งคำสั่งควบคุมผ่านทางอินเทอร์เน็ต	40
6.1 คอนโทรล MComm	40
6.2 คุณสมบัติของ MComm	40
6.3 การเขียนโปรแกรมเพื่อควบคุมการเชื่อมต่อผ่านอินเทอร์เน็ต	44
บทที่ 7 หลักการออกแบบและการสร้าง	51
7.1 ชุดควบคุมกล้อง	51
7.2 การโปรแกรมด้วย Visual Basic	60
7.3 หลักการออกแบบเซนเซอร์อัลตราโซนิกเพื่อใช้ตรวจจับสิ่งกีดขวาง	65
บทที่ 8 สรุปและวิเคราะห์ผลการทดลอง	69
8.1 การทดลองที่ 1 การรับ-ส่งรหัสควบคุมมอเตอร์และการหมุนของมอเตอร์	69
8.2 การทดลองในส่วนของการส่งภาพและเสียง	70
8.3 การทดลองในส่วนของเซนเซอร์	73
บรรณานุกรม	

สารบัญภาพ

หน้า

รูปที่ 1.1 แสดงการเชื่อมต่ออุปกรณ์ความคุมกับเครือข่ายอินเทอร์เน็ต	1
รูปที่ 2.1 สถาปัตยกรรมของ internet	3
รูปที่ 2.2 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model	4
รูปที่ 2.3 Network layer structure	6
รูปที่ 2.4 Internetwide IP schematic	7
รูปที่ 2.5 โครงสร้าง ของ address ที่ใช้ใน class ต่างๆ ของเครือข่ายทั้งหมดความยาว 32 bits	8
รูปที่ 2.6 Internet datagram format and contents	9
รูปที่ 2.7 Internet fragmentation and reassembly	10
รูปที่ 3.1 แสดง TCP/IP stack เปรียบเทียบกับมาตรฐาน OSI	13
รูปที่ 3.2 แอปพลิเคชันหรือโปรเซสต่างๆ สื่อสารกับ Host-to-Host ผ่านจุดเชื่อมต่อ	14
รูปที่ 3.3 โปรเซสต่างๆที่เรียกใช้ Transport layer เพื่อส่งผ่านข้อมูลโดยอาศัย Port	16
รูปที่ 3.4 รูปแบบของ TCP packet	17
รูปที่ 3.5 รูปแบบของ UDP packet	18
รูปที่ 3.6 โปรโตคอล TCP และ UDP อาศัย IP เพื่อส่งผ่านข้อมูล	19
รูปที่ 3.7 โครงสร้างของโปรโตคอล TCP/IP	21
รูปที่ 3.8 แสดง IP Header	23
รูปที่ 4.1 โครงสร้างภายในของ MCS-51	26
รูปที่ 4.2 แสดงตำแหน่งขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51	27
รูปที่ 4.3 ตำแหน่งหน่วยความจำของ MCS-51	29
รูปที่ 4.4 แสดงการกระตุ้นแบบเฟสเดียว	32
รูปที่ 4.5 แสดงการกระตุ้นแบบ 2 เฟส	33
รูปที่ 4.6 แสดงการกระตุ้นแบบครึ่งสเต็ป	33
รูปที่ 5.1 รูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส	35
รูปที่ 5.2 รายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรม ของคอมพิวเตอร์	39
รูปที่ 5.3 วงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ และไมโครคอนโทรลเลอร์ MCS-51	39
รูปที่ 7.1 แสดง โครงสร้างภายในของ AT89C2051	51
รูปที่ 7.2 แสดงตำแหน่งขาต่างๆของ AT89C2051	52

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.3 แสดงการต่อคริสตัลให้กับ AT89C2051	53
รูปที่ 7.4 แสดงรายละเอียดของ ULN2003A	54
รูปที่ 7.5 แสดงวงจรการเชื่อมต่อของ MCS-51 และวงจรการขับเคลื่อนมอเตอร์	55
รูปที่ 7.6 แสดง Flowchart โปรแกรมหลักที่อยู่ในชุดควบคุมกล้อง	56
รูปที่ 7.7 แสดง Flowchart การทำงาน โปรแกรมโหมด Manual ที่อยู่ในชุดควบคุมกล้อง	57
รูปที่ 7.8 แสดง Flowchart โปรแกรมย่อยของโปรแกรมหมุนมอเตอร์โหมด Manual	58
รูปที่ 7.9 แสดง Flowchart โปรแกรมย่อยของโปรแกรมหมุนมอเตอร์โหมด Auto	59
รูปที่ 7.10 แสดงโครงสร้างการทำงานของระบบ	61
รูปที่ 7.11 กลไกการเชื่อมต่อผ่านทางอินเทอร์เน็ตของคอลโทรล Active XLEADTOOL ที่ชื่อว่า Inet	62
รูปที่ 7.12 แสดงหน้าจอหลักทางด้านฝั่ง Server	63
รูปที่ 7.13 แสดงหน้าจอควบคุมการติดต่อทางด้านฝั่ง ไคลเอนต์	64
รูปที่ 7.14 บล็อกไดอะแกรมการทำงานของวงจรส่วนเซนเซอร์	65
รูปที่ 7.15 รูปวงจรทั้งหมดของวงจรส่วนเซนเซอร์ที่จะนำไปต่อกับ ไมโครคอนโทรลเลอร์	66
รูปที่ 7.16 สัญญาณที่จุดทดสอบต่างๆ แสดงการทำงานของวงจร	67
รูปที่ 7.17 ก. เมื่อเป้าหมายเคลื่อนที่ออกจากกึ่งกลางกล้อง ข. เมื่อเซนเซอร์ตรวจจับเป้าหมายได้และควบคุมให้กึ่งกลางกล้องย้ายตำแหน่ง	68
รูปที่ 8.1 ก. เมื่อเป้าหมายอยู่ตรงกลาง ข. เมื่อเป้าหมายเคลื่อนไปทางซ้าย ค. เมื่อเซนเซอร์ควบคุมกล้องให้เคลื่อนมาหาเป้าหมาย	73
รูปที่ 8.2 ก. เมื่อเป้าหมายอยู่ตรงกลาง ข. เมื่อเป้าหมายเคลื่อนที่ออกไปทางขวา ค. เมื่อเซนเซอร์ควบคุมกล้องให้เคลื่อนมาหาเป้าหมาย	74

สารบัญตาราง

	หน้า
ตารางที่ 7.1 แสดงหน้าที่พิเศษของพอร์ต3	53
ตารางที่ 8.1 ผลการทดลองวัดตำแหน่งมอเตอร์แกน X ที่หมุนจริงกับค่าตำแหน่งที่แสดงบน หน้าจอโปรแกรม Visual Basic	69
ตารางที่ 8.2 ผลการทดลองวัดตำแหน่งมอเตอร์แกน Y ที่หมุนจริงกับค่าตำแหน่งที่แสดงบน หน้าจอโปรแกรม Visual Basic	70
ตารางที่ 8.3 แสดงผลการทดลองในการส่งภาพออกทางด้าน client	71
ตารางที่ 8.4 แสดงผลการทดลองการส่งเสียงผ่านเครือข่าย	72



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

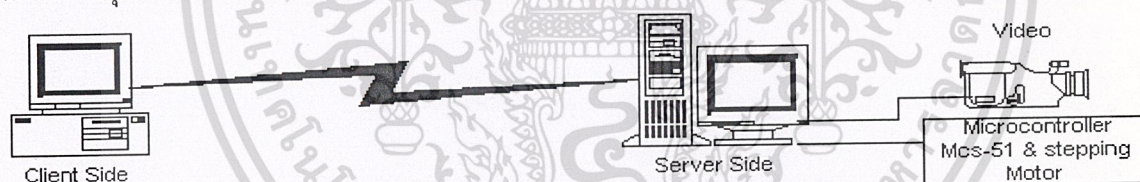
บทนำ

1.1 ความสำคัญและที่มา

ทุกวันนี้ Internet ได้เข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมากโดยเฉพาะการติดต่อสื่อสารทุกรูปแบบเช่น การสนทนา, การสั่งซื้อสินค้า, การจองตั๋วเครื่องบินและการควบคุมระยะไกล(Re เป็นต้น ไม่ว่าเราจะอยู่ที่ใดก็ตามก็จะสามารถเข้าถึงอินเทอร์เน็ตได้ เนื่องจากอินเทอร์เน็ตนั้นครอบคลุมทั่วทุกมุมโลก

จากข้อดีของอินเทอร์เน็ตหลายข้อทำให้เกิดแนวความคิดนำการนำอินเทอร์เน็ตมาผสมผสานกับระบบอิเล็กทรอนิกส์เพื่อประยุกต์ใช้ในการควบคุมระยะไกลซึ่งประกอบด้วยส่วน Hard ware และ Software เพียงเท่านี้ก็ยังสามารถนำไปใช้ในที่ต่างๆที่มีความยากลำบากต่อการทำงานเช่นในสถานที่ที่มีความร้อนสูง, ในที่มีสารกัมมันตภาพรังสี ที่เป็นอันตรายต่อมนุษย์ และใช้ในการรักษาความปลอดภัยซึ่งจะเป็นประโยชน์มากอีกทั้งทำให้ประหยัดทรัพยากรอีกด้วย

โครงการนี้ได้นำเอาแนวความคิดข้างต้นมาทำการศึกษาและสร้างเครื่องมือชุดนี้ขึ้นมาเพื่อให้ได้มาซึ่งวัตถุประสงค์ตามที่ต้องการ โดยหลักการทำงานอย่างคร่าวๆ สามารถดูได้จากบล็อกรูปที่ 1.1



รูปที่ 1.1 แสดงการเชื่อมต่ออุปกรณ์ควบคุมกับเครือข่ายอินเทอร์เน็ต

จากบล็อกข้างต้นสามารถที่จะอธิบายได้ดังนี้

1. ส่วนควบคุม(Client Control) ประกอบไปด้วยคอมพิวเตอร์ที่เชื่อมต่อเข้ากับเครือข่ายอินเทอร์เน็ตและบรรจุโปรแกรมควบคุมการทำงานของส่วนควบคุมกล้องและส่วนการรับภาพและเสียงผ่านเครือข่ายอินเทอร์เน็ตในส่วนของโครงการนี้เอาไว้

2. ส่วนประมวลผล (Server Process) จะเป็นคอมพิวเตอร์ที่ทำให้เป็น Server ที่เอาไว้สำหรับติดต่อกับส่วนควบคุมโดยผ่านเครือข่ายอินเทอร์เน็ต และนำผลของการประมวลผลไปส่งยังส่วนควบคุมกล้องรวมทั้งใช้สำหรับการส่งภาพและเสียง

3. ส่วนควบคุมกล้อง (Camera Control Unit) จะเป็นส่วนที่รับคำสั่งควบคุมการทำงานจากส่วนประมวลผลมาปฏิบัติตามแล้วและจะรายงานผลตำแหน่งของกล้องด้วยว่าขณะนี้ที่ตำแหน่งใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กล้องถ่ายภาพ (Video Camera) ทำหน้าที่ แปลงภาพให้เป็นสัญญาณ VDO เพื่อส่งไปให้คอมพิวเตอร์

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาการรับ-ส่งข้อมูลผ่านพอร์ตอนุกรมของคอมพิวเตอร์
2. ศึกษาการส่งภาพและเสียงผ่านเครือข่ายอินเทอร์เน็ต
3. ทำการปรับปรุงคุณภาพของภาพและเสียง
4. ทำการปรับปรุงโปรแกรมการทำงานให้มีลักษณะง่ายต่อการใช้งาน

1.3 ขอบเขตของโครงการ

1. ส่วน Hardware ประกอบด้วยสเต็ปมอเตอร์ 2 ตัวเพื่อใช้สำหรับติดตั้งกล้องวิดีโอเพื่อการควบคุมตำแหน่งและวงจรถูดควบคุมมอเตอร์ซึ่งจะมีวงจรในส่วนของ การรับส่งข้อมูลผ่านพอร์ตอนุกรมและวงจรในส่วนของอุลตราโซนิกเซนเซอร์ (Ultrasonic Sensor) สำหรับตรวจจับการเคลื่อนไหวด้วย โดยติดตั้งอุลตราโซนิกเซนเซอร์ไว้ 2 ตัวทางด้านซ้ายและขวาของฐานกล้อง เมื่อเซนเซอร์ด้านใดตรวจจับพบสิ่งกีดขวางก็จะทำให้มอเตอร์หมุนไปทางด้านนั้น
2. ส่วน Software ประกอบด้วยโปรแกรมสำหรับการใช้งาน ซึ่งพัฒนาขึ้นโดยใช้โปรแกรม Visual Basic เวอร์ชัน 6 โดยหน้าที่ของโปรแกรมมีดังนี้
 - 2.1 รับส่งรหัสควบคุมสเต็ปมอเตอร์ผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์
 - 2.2 การควบคุมสเต็ปมอเตอร์ผ่านเครือข่ายอินเทอร์เน็ต
 - 2.3 การส่งภาพและเสียงผ่านเครือข่ายอินเทอร์เน็ต

1.4 วิธีการดำเนินงาน

โครงการนี้เริ่มต้นจากการศึกษาทฤษฎีต่างๆในการสื่อสารผ่านพอร์ตอนุกรม การใช้งานไมโครคอนโทรลเลอร์และอุลตราโซนิกเซนเซอร์ ต่อมาจึงได้ศึกษาทฤษฎีต่างๆที่ใช้ในการเขียนโปรแกรมบนระบบปฏิบัติการวินโดวส์ และศึกษาชุดคำสั่งต่างๆ ของโปรแกรม Visual Basic แล้วจึงได้ออกแบบการทำงานและส่วนอินเตอร์เฟซ (Interface) ของโปรแกรม

หลังจากนั้นจึงได้ทำการออกแบบวงจรในส่วน Hardware และเขียนโปรแกรมโดยใช้คอนโทรลต่างๆ ของ Visual Basic ตามรูปแบบของโปรแกรมที่ออกแบบไว้ ทำการทดลองรับส่งรหัสควบคุมมอเตอร์ผ่านพอร์ตอนุกรม แล้วจึงทำการทดลองติดต่อกันระหว่างคอมพิวเตอร์ 2 เครื่องเพื่อทดลองการทำงานในส่วนต่างๆ ได้แก่ การเข้าสู่ระบบ , การติดต่อกันระหว่างเครื่อง , การควบคุมกล้องผ่านเครือข่าย, การส่งภาพและเสียง แล้วพัฒนาปรับปรุงคุณภาพให้ดีที่สุดเพื่อนำไปใช้งานอย่างมีประสิทธิภาพ

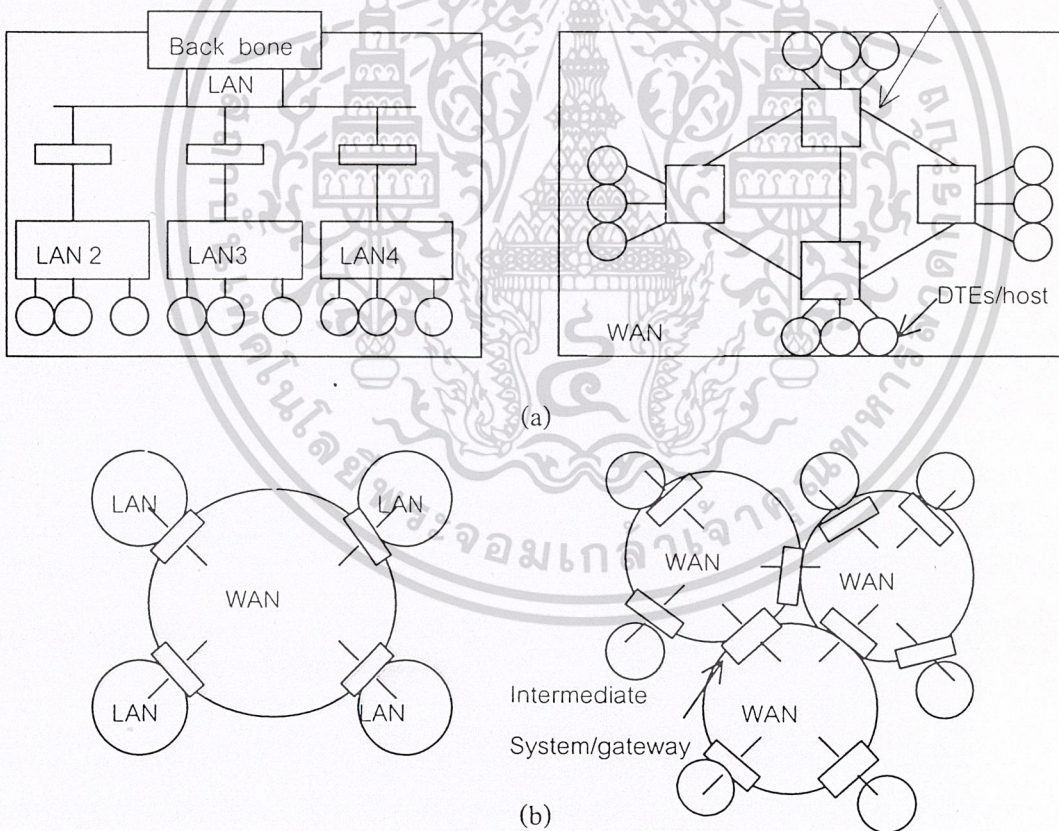
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

เครือข่ายอินเทอร์เน็ต

อินเทอร์เน็ต (Internet) คือการที่เครือข่าย 2 หรือมากกว่า เชื่อมต่อกัน และการทำงานเสมือนเป็นเครือข่ายเดียวกัน โดย network ที่เป็นส่วนประกอบของ internet คือ Subnetwork (subnet) ซึ่งการจะเป็นเครือข่าย Local Area Network (LAN) หรือ Wide Area network (WAN) อุปกรณ์ที่ใช้ในการเชื่อมต่อ 2 เครือข่ายเข้าด้วยกันก็คือ intermediate system (IS) หรือ internetworking unit (IWU) การเชื่อมโยงระหว่างระบบที่แตกต่างกัน จำเป็นต้องมีมาตรฐานการติดต่อกัน ซึ่งเรียกเป็นศัพท์เฉพาะว่า โปรโตคอล (protocol)

2.1 สถาปัตยกรรม อินเทอร์เน็ต (Internet Architectures)



รูปที่ 2.1 สถาปัตยกรรมของ internet(a)single LAN and WAN (b) Internetconnected

LAN/WAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูป (a) แสดงตัวอย่าง 2 ตัวอย่างของเครือข่ายเดี่ยว (Single network) ซึ่งอย่างแรกเป็น site-wide LAN ซึ่งประกอบขึ้นมาจากชุดของ LANs ซึ่งถูกต่อเข้ากับเครือข่ายหลัก (backbone) ซึ่งอุปกรณ์ที่ใช้ต่อ LAN เข้ากับเครือข่ายหลักถ้า LAN ทุกเครือข่ายมีระบบเดียวกันก็จะใช้ bridge ถ้าเป็น LAN ที่แตกต่างกันก็จะใช้ router ตัวอย่างที่ 2 เป็นตัวอย่างของ WAN เดี่ยว ๆ ในรูป (b) แสดงถึงเครือข่ายอินเทอร์เน็ต ซึ่งประกอบด้วย network ทั้ง 2 ชนิดข้างต้น

2.2 OSI โมเดล

องค์ประกอบมาตรฐานสากล ISO (International Organization for Standardization) ได้กำหนดมาตรฐานของเครือข่าย โดยจัดแบ่งกิจกรรมของเครือข่าย ออกเป็นงานย่อย ๆ และกำหนดโมเดลแบ่งเป็นชั้น ๆ ตามลำดับเรียกว่ามาตรฐาน OSI (Open System Interconnection) โดยที่จะแบ่งกิจกรรมที่ซับซ้อนในเครือข่าย ออกเป็นงานย่อย ๆ ก็จะช่วยในการออกแบบ และการใช้งานเครือข่ายรวมถึงการเชื่อมต่อโดยกัน เป็นไปได้ด้วยความสะดวก และมีวิธีการทำงานอยู่ในกรอบเดียวกัน ดังรูปที่ 2.2

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Datalink Layer
Physical Layer

รูปที่ 2.2 แสดงการแบ่งการทำงานของเครือข่ายออกเป็น OSI model

2.3 โครงสร้างชั้น Network

ในแต่ละชั้นของ OSI model จะมีการติดต่อสื่อสารกันเป็นชั้น ตามลำดับลงมาเช่น

Application Layer ก็จะติดต่อสื่อสารกับ Presentation Layer ตามลำดับไปจนถึงชั้นแรกสุด คือ Physical Layer เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Layer เป็นชั้นบนสุดของโมเดล เป็นส่วนที่จะทำให้การติดต่อระหว่างเครือข่ายกับผู้ใช้เป็นไปตามต้องการตัวอย่างแอปพลิเคชันของเครือข่าย เช่น ระบบ e-mail , การโอนถ่ายข้อมูล (File transfer) การขอใช้ระบบคอมพิวเตอร์ในเครือข่าย เป็นต้น Presentation Layer มีการกำหนดหน้าที่ไม่ชัดเจนและมีการนำไปใช้ไม่มาก ซึ่งหน้าที่หลักก็คือเป็นส่วนที่จัดรูปและนำเสนอข้อมูล ให้เป็นไปตามต้องการ รวมไปถึงการจัดแปลงข้อมูล ในรูปแบบมาตรฐาน ASCII หรือ EBCDIC การลดขนาดข้อมูล (data compression) การเข้ารหัส หรือ session ให้ระบบคอมพิวเตอร์ทั้งสองฝั่ง โดยทำหน้าที่ตั้งแต่เริ่มการติดต่อ ดูแลในการส่งผ่านข้อมูล ในการติดต่อครั้งนั้น ๆ เป็นไปได้โดยไม่มีปัญหา จนถึงเลิกการติดต่อเมื่อเสร็จงาน

Transport Layer ทำหน้าที่ควบคุมปริมาณ และรายละเอียดการรับส่งข้อมูล ให้เป็นไปตามกำหนดที่ตั้งไว้และจัดการให้การเชื่อมโยงเครือข่ายเป็นไปด้วยความราบรื่น Transport Layer จะเป็นชั้นสุดท้ายที่จัดการเรื่องเส้นทางในการส่งข้อมูล และจัดการตรวจสอบความผิดพลาดของข้อมูล ซึ่งส่วนของ TCP (Transmission Control Protocol) ในโปรโตคอล TCP/IP ทำงานที่ระดับนี้

Network Layer ทำหน้าที่ควบคุมวิธีการส่งผ่านข้อมูลการระหว่างเครือข่ายให้ถูกต้อง และเป็นไปตามเส้นทางที่กำหนด โดยจะจัดการส่งผ่าน packet ข้อมูล ผ่านอุปกรณ์ต่าง ๆ ไปยังเครือข่ายย่อยได้อย่างถูกต้องตามที่ต้องการ นอกจากนี้ยังจัดการดูแลเส้นทางในการส่งข้อมูล (Routing table) และกั้นกรอง packet ข้อมูลที่ส่งไปยังเครือข่ายเดียวกันไม่ให้ข้ามไปยังเครือข่ายอื่น ซึ่งจะช่วยลดปริมาณข้อมูลที่จะวิ่งบนเครือข่ายได้ส่วนหนึ่ง โปรโตคอล IP, TCP/IP และ Ipx เป็นโปรโตคอลที่ทำงานอยู่ใน layer นี้

Datalink Layer ทำหน้าที่เรียกใช้หรือกำหนดช่องทาง ในการส่งข้อมูลที่ต้องการ เช่น Ethernet, Tokenring หรือ FDDI เป็นต้น รวมถึงการลำดับและอัตราการรับส่งข้อมูลหรือ flow control และสถานที่ที่จะส่งข้อมูลไป (address) ทั้งนี้ Datalink layer จะเป็นชั้นแรกที่จัดการแปลงข้อมูลจาก bit ให้เป็น packet โดยจะมีการเพิ่มข้อมูลเพื่อตรวจสอบผ่าน checksum เพื่อดูว่าข้อมูลที่ได้รับมาถูกต้องครบถ้วน และถ้าได้รับ packet ข้อมูลที่ไม่ถูกต้องก็จะไม่เอาข้อมูลนี้ไปใช้งานและบอกให้ค้นหาทางส่งข้อมูลเดิมมาใหม่

Physical Layer รับผิดชอบดูแลในรายละเอียดในการส่งข้อมูลในด้าน hardware เช่น การควบคุม Network Interface Card การส่งสัญญาณแบบต่าง ๆ การเชื่อมต่อเข้ากับเครือข่ายต่าง ๆ โดยใช้ Physical Layer จะจัดสร้างสัญญาณทางไฟฟ้า , สัญญาณเสียง หรือสัญญาณที่จำเป็นในการสื่อสารโดยตรง

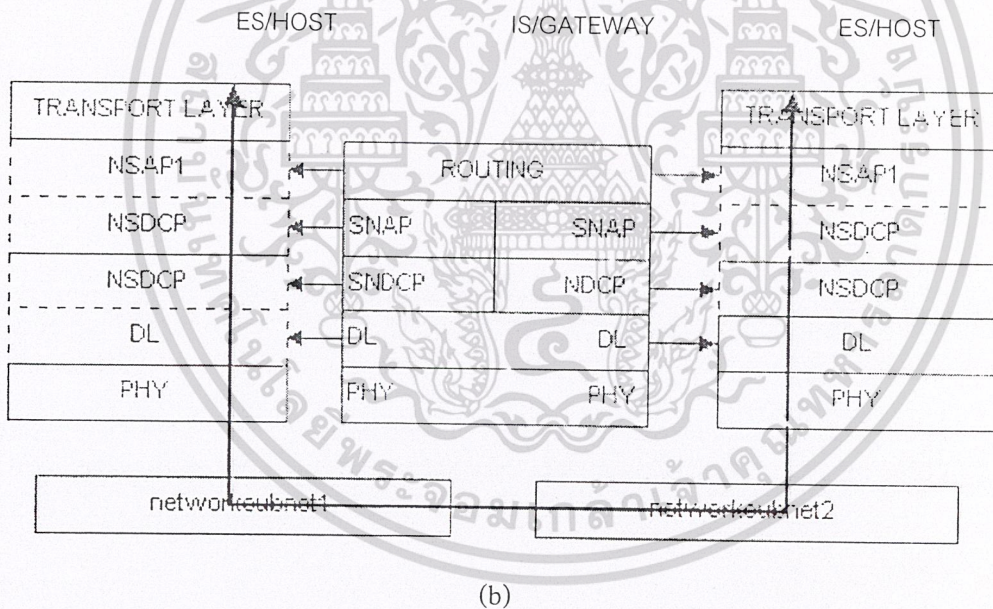
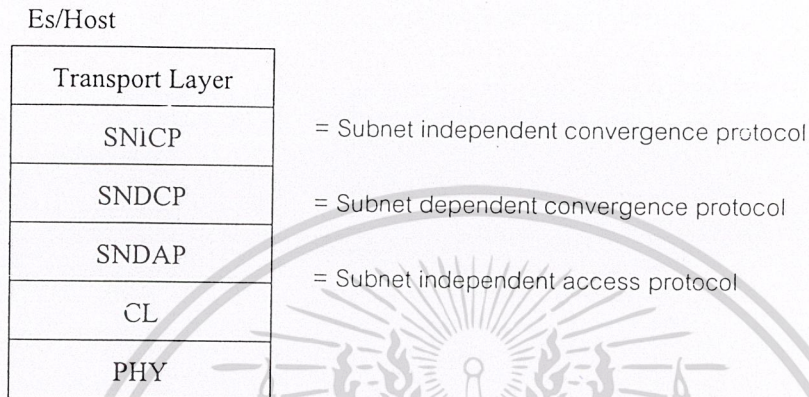
เนื่องจาก Network Layer ในแต่ละ End System (ES) จะเป็นตัวจัดการการติดต่อ แบบ end to end ของการให้บริการ internetwide ไปยังผู้ใช้บริการ (NS-User)

โดย ISO ได้จัด network layer เป็น 3 (sub layer) Protocol ซึ่งจะทำงานร่วมกัน เพื่อให้บริการใน network layer ได้แก่ การให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Subnetwork Independent Convergence Protocol (SNICP)

Subnetwork Dependent Convergence Protocol (SNDACP)

Subnetwork Dependent Access Protocol (SNDAP)



รูปที่ 2.3 network layer structure (a) Sub layer Protocol ; (b) IS structure

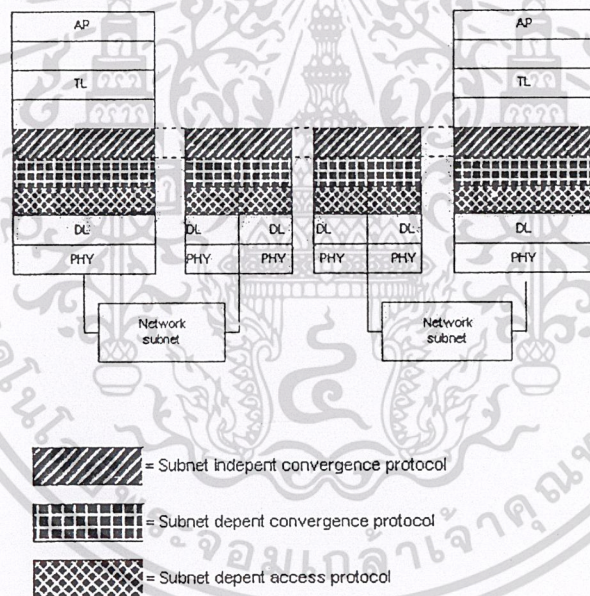
โดยที่ SNICP จะเป็นตัวสนับสนุนจัดการให้ผู้ใช้บริการ (NS-user)สามารถ interface กับ internet ซึ่งมันจะมีหน้าที่เป็นตัวประสานฟังก์ชันต่าง ๆ ที่จำเป็นในการเลือกเส้นทางและถ่ายทอดข้อมูลของผู้ใช้งาน Internet ซึ่งการทำงานของมันก็ขึ้นอยู่กับคุณสมบัติเฉพาะของเครือข่าย

(subnet) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SNDAP จะเป็นตัวโปรโตคอลที่ติดต่อกับเครือข่ายย่อย (subnet) ที่มีลักษณะเฉพาะใน Internet เช่น X.25 packet layer protocol สำหรับเครือข่าย X.25 ซึ่งใช้บ่อยใน LAN เพราะว่าการบริการประกอบการทำงาน SNDAP แตกต่างจาก network แบบอื่น ๆ sublayer ที่อยู่ตรงกลางคือ SNDCP จะเป็นตัวตัดการระหว่าง SNICP และ SNDAP

2.4 รูปแบบมาตรฐานโปรโตคอลของอินเทอร์เน็ต (Internet Protocol standards)

อินเทอร์เน็ตโปรโตคอลซึ่งถูกใช้ในอินเทอร์เน็ต TCP/IP (Transfer Control Protocol / Internet Protocol) ซึ่งรวมถึง transport และ application โปรโตคอล ซึ่งทั้งหมดของ TCP/IP จะกำหนดให้เหมาะสมกับการใช้ในเชิงสาธารณะ ซึ่งรูปแบบโดยทั่วไปแสดงดังรูปที่ 2.4 IP เป็น internetwide protocol ซึ่งทำให้สอง transport protocol ที่ต่างสถานที่และต่าง Ess/hosts กันสามารถแลกเปลี่ยนหน่วยข้อมูล (NSDUS) กันได้ ซึ่งหมายถึงว่า หลาย ๆ network / subnet และ ISO/Gateway ที่แตกต่างกันสามารถติดต่อสื่อสารได้อย่างสมบูรณ์



รูปที่ 2.4 Internetwide IP schematic

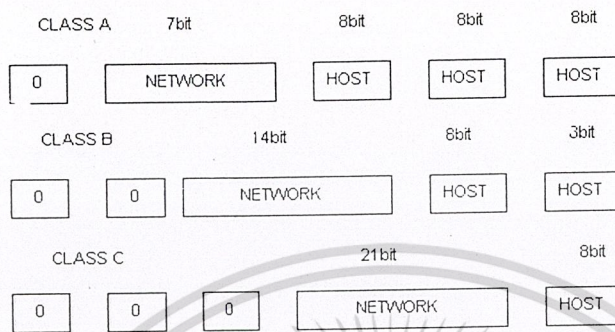
2.5 Internet IP

2.5.1 address structure

ในศัพท์ของ ISO เมื่อ 2 network ติดต่อกันด้วย host/Es ที่ต่อกับอินเทอร์เน็ต network เหล่านี้ ติดต่อกันได้โดยใช้ network service access point (NSAP) address และ subnet point of attachment (SNPA) สำหรับใน TCP/IP ก็จะมี IP address และ NPA address ตามลำดับ โดย NPA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

address จะแตกต่างกันในแต่ละชนิดของ network/ subnetwork ขณะที่ IP address ตามลำดับ โดย NPA address จะเป็นรูปแบบเดียวกัน โครงสร้าง ของ IP address แสดงดังรูปที่ 2.5



รูปที่ 2.5 โครงสร้าง ของ address ที่ใช้ใน class ต่าง ๆ ของเครือข่ายทั้งหมดความยาว 32 bits

IP address นี้มีการจัดแบ่งออกเป็นทั้งหมด 5 ระดับ (class) แต่ที่ใช้งานทั่วไปจะมีเพียง 3 ระดับ คือคอมพิวเตอร์ที่เชื่อมต่ออยู่มาก จะมีหมายเลขอยู่ใน Class A และลดหลั่นกันมาใน Class B และ Class C ตามลำดับ

จากรูปจะเห็นว่าหมายเลข IP ของ Class A มีตัวแรกเป็น 0 และหมายเลขเครือข่าย (network number) ขนาด 7 Bit และมีหมายเลขเครื่องคอมพิวเตอร์ (Host number) ขนาด 24 bit ทำให้ในหนึ่งเครือข่ายของ Class A สามารถมีคอมพิวเตอร์เชื่อมต่ออยู่ในเครือข่ายได้ถึง $2^{24} = 16$ ล้านเครื่อง แต่ใน Class A จะหมายเลขเครือข่ายได้ 128 ตัวเท่านั้นทั่วโลก ซึ่งก็คือจะมีเครือข่ายใหญ่แบบนี้เพียง 124 เครือข่ายเท่านั้น

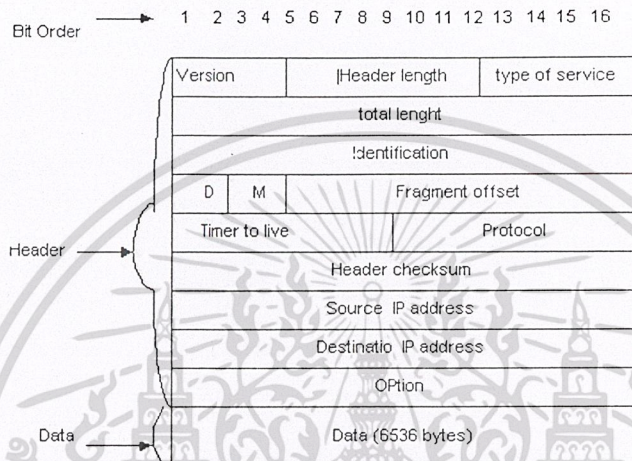
สำหรับ Class B จะมีหมายเลขเครือข่ายแบบเครื่องคอมพิวเตอร์แบบ 16 bits (ส่วนอีก 2 bit ที่เหลือบังคับว่าต้องขึ้นต้นด้วย 10₂) ดังนั้นจึงสามารถมีคอมพิวเตอร์เชื่อมต่อในเครือข่าย Class B แต่ละเครือข่ายได้ถึง 216 กว่า 65000 เครื่อง และสุดท้ายคือ Class C ซึ่งมีหมายเลขคอมพิวเตอร์แบบ 8 bit และมีหมายเลขเครือข่ายแบบ 21 bit ส่วน 3 bit แรกบังคับว่าต้องเป็น 110₂ ดังนั้นในแต่ละเครือข่าย ($2^8 = 256$ แต่หมายเลข 0 และ 255 จะไม่ถูกใช้งานจึงเหลือเพียง 254)

จะเห็นได้ว่าเมื่อเครือข่ายและเครื่องคอมพิวเตอร์ที่ต่ออยู่ในอินเทอร์เน็ตนี้มีหมายเลข IP address ให้ใช้อ้างอิงได้ไม่ซ้ำกันและมีความหมายให้ทราบถึงขนาดเครือข่ายแล้วการติดต่อผ่านข้อมูลจึงกระทำได้ไม่สับสน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 รูปแบบของข้อมูล (Datagrams)

รูปแบบของ IP data unit ก็คือ datagrams ซึ่งโครงสร้างของ datagrams เป็นดังรูปที่ 2.6



รูปที่ 2.6 Internet datagrams format and contents

หน่วยข้อมูล IP (IP datagrams) แต่ละหน่วยจะประกอบด้วย ส่วนของข้อมูลที่รับมาจาก ส่วนงาน TCP หรือUDP และส่วนของข้อมูลนำทาง(Header) ซึ่งมีรายละเอียดดังนี้

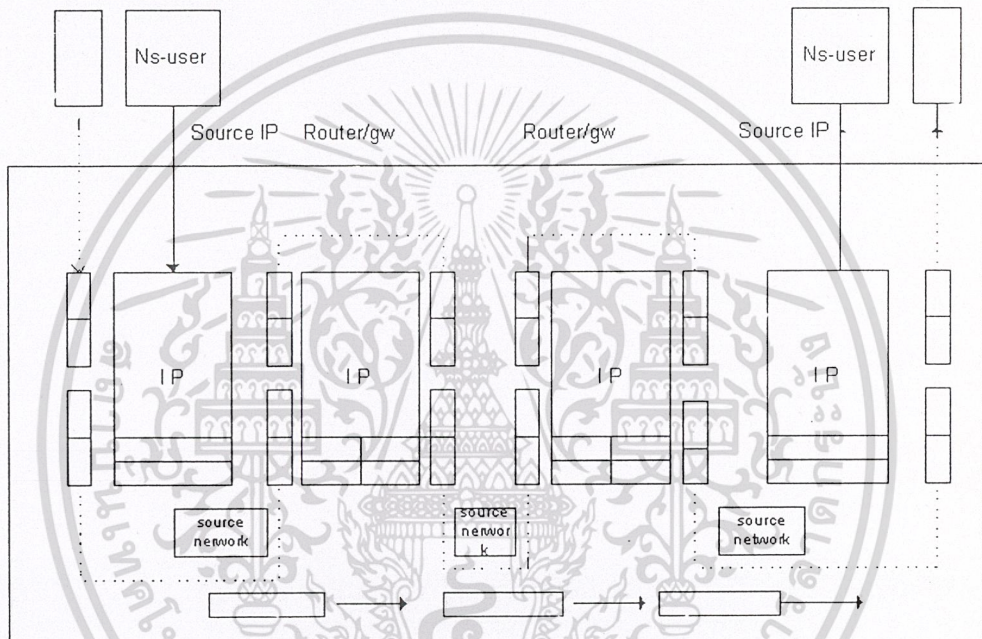
- Version หมายถึงรุ่นของข้อกำหนด IP
- Header Length ความยาวของข้อมูลนำทาง
- Type of service วิธีการจัดการกับข้อมูล
- Total Length ความยาวของหน่วยข้อมูล
- Identification , Flags และ Fragment offset รายละเอียดที่เกี่ยวกับการแบ่งย่อยข้อมูลซึ่ง จะถูกนำมาใช้ในการรวบรวมข้อมูล
- Time to live เวลาสูงสุดที่ใช้ในการเดินทาง ซึ่งกำหนดมาจากต้นทาง เวลานี้จะลดลง เรื่อยในระหว่างทางถ้าลดลงไปถึงศูนย์ หน่วยข้อมูลนั้นจะถูกกำจัดไป
- Protocol ชนิดของข้อมูลเป็น UCP หรือ TCP
- Header Checksum ค่าตรวจสอบข้อมูลนำทาง
- IP address หมายเลข internetwide IP(NSP)ของเครื่องต้นทางและปลายทาง
- Option ข้อมูลอื่น ๆ เช่นข้อมูลที่เกี่ยวข้องกับการรักษาความปลอดภัย บันทึกเส้นทางเดิน

ของข้อมูลและเวลาที่ข้อมูลเดินทางมาถึง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 การแบ่งส่วนข้อมูลและการประกอบขึ้นใหม่ (Fragmentation and Reassembly)

ขนาดข้อมูลของผู้ใช้ซึ่งอ้างอิง NSDU มีความจุได้ถึง 64 k หรือ 65,536 byte ต่อขนาดของหน่วยข้อมูล (packet size) ที่สามารถติดต่อกันในระบบที่ต่างกัน สามารถมีได้ตั้งแต่ 128 byte สำหรับระบบ x.25 packet switching จนถึง 8,000 bytes สำหรับบาง LAN ดังนั้นกระบวนการ Fragmentation และ Reassembly จึงถูกนำมาใช้เพื่อ ทำให้ขนาดของข้อมูลเล็กลง และสามารถส่งไปในระบบได้ และเมื่อถึงปลายทาง IP ก็จะทำการประกอบข้อมูล (reassemble) ขึ้นมาใหม่ก่อนที่จะส่งผ่านไปยังผู้ใช้ปลายทาง ดังรูปที่ 2.7



รูปที่ 2.7 internet fragmentation and reassembly

อันดับแรก IP ใน Host ต้นทางจะแยกข้อมูลของผู้ใช้ (NS-User) , NSDU เป็น Datagram ซึ่งมี Address กำกับเป็นเฉพาะส่วน ๆ ไป ซึ่งจะถูกออกคำสั่งโดย network ที่ในติดต่อกันด้วยและการส่ง Datagram ไปยัง IP ใน Gateway ตัวแรก โดยที่ IP ใน Gateway จะไม่ Reassemble แต่จะปรับปรุงในขอบเขตที่เหมาะสม และส่ง Datagram ที่ได้รับตรงไปยัง Network ที่สอง (ถ้า Network ที่สองสามารถรองรับ ขนาด Datagram นี้) หรือทำการ Fragment datagram ให้มีขนาดเล็กลง ซึ่งขั้นตอนนี้จะถูกทำซ้ำที่ Gateway ตัวต่อไป Network ตัวสุดท้ายสามารถรองรับขนาดของ packet ได้มากกว่า packet ที่มันได้รับข้อมูลจึงถูกส่งได้โดยตรง โดยที่จะมีการปรับปรุงในบางส่วน Header ของ Datagram เท่านั้น จากนั้น IP ใน Host ปลายทางจะทำการประกอบข้อมูล (Reassemble) ที่มันได้รับขึ้นมาใหม่และส่งผลที่ได้รับก็คือ NSDU ไปยังผู้ใช้ (NS-user)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการคิดค่าเวลาสูงสุดที่ Host ต้นทางกำหนดให้ Gateway รอ datagram(NSDU) ระหว่างแต่ละการ Assembly ซึ่งก็คือ time-to-live ซึ่งจะถูกรั้งค่าโดย IP ใน Host ต้นทาง ซึ่งจะมีค่าลดลงเรื่อย ๆ ในแต่ละขั้นตอนของการ Process datagram ถ้า Datagram ถูก Fragment ค่าปัจจุบันจะถูกนำไปใส่ในส่วน Header ของ Datagram ตัวใหม่ ถ้ามันถึงค่า 0 ที่จุดใด ๆ ระหว่างการ Process ใน Gateway (หรือ Host) การ reassembly ก็จะมีผลและทุก ๆ การ Fragment ที่เกี่ยวกับ NSDU ก็จะถูกตัดทิ้ง

ค่า Time-to-live ในแต่ละ datagram จะเป็นจำนวนเท่าของวินาที โดยที่จำนวนของมันจะถูกลดลงโดยแต่ละ IP ซึ่งจะเปลี่ยนแปลงไปตามค่าเวลาจริงในการส่งถ่ายข้อมูลของ network ที่ติดต่อกันด้วย

2.5.4 การเลือกเส้นทาง (Routing)

ในแต่ละ network (หรือ subnet) ใน Internet จะมีชนิดของ PA address ที่แตกต่างกัน ซึ่งระบบ (system Host) หรือ gateway ที่ถูกต่อเข้ากับ network จะสามารถส่ง data gram ไปยังระบบอื่นได้โดยตรงเฉพาะ network ที่เหมือนกันเท่านั้น ในการเลือกเส้นทาง (routing) ให้ datagram ข้ามไปยัง หลาย ๆ network IP ในแต่ละ internetwork gateway ต้องรู้ PA address ของ host ปลายทาง

ซึ่งมี 2 วิธีการพื้นฐานที่ถูกใช้ในการหาเส้นทางภายใน Internet คือ centralized และ distributed ด้วยวิธีการ centralized routing ข้อมูลเกี่ยวกับการเลือกเส้นทางที่เกี่ยวข้องกับแต่ละ gateway จะถูก download จาก site ส่วนกลาง โดยใช้ข้อมูล network และ special network management จะพยายามตรวจสอบ network และ host ที่ถูกเพื่อเข้า และถอดออก และขี้อบกพร่องที่จะถูกวินิจฉัยและตรวจสอบ

ด้วยวิธีการ Distributed routing ทุกๆ host และ gateway จะร่วมกันในการแบ่งปัน วิธีการในการรับประกันว่า ข้อมูลเกี่ยวกับการเลือกเส้นทางจะถูกจัดจำไว้โดยแต่ละระบบ ในรูปของ routing table ซึ่งจะมี NPA address ไว้ใช้ในการส่งแต่ละ datagram ซึ่ง Internet จะใช้วิธีการแบบนี้

ขั้นตอนการ Routing ที่เกี่ยวกับ IP ขั้นตอนแรกจะอ่าน IP address (NSAP) ปลายทางจากภายใน datagram และใช้มันในการหาการตอบสนอง PA address ของ host หรือ gateway จาก routing table ในส่วนที่เพิ่มเติมชุดของ routing protocol จะถูกใช้เพิ่มและรักษาส่วนที่อยู่ในแต่ละ routing table ในรูปแบบของ distributed ซึ่งรูปแบบทั่วไปที่ถูกใช้ภายใน host IP

บทที่ 3

หลักการโปรโตคอลของTCP/IP

ในช่วง 5 ปีที่ผ่านมา มี Programmer จำนวนไม่มากที่ได้รับข่าวของ TCP/IP ซึ่งเป็นชุด Protocol ไว้ใช้ในการส่งผ่านเครือข่าย Internet สำหรับงาน Internet Communication และไม่มีระบบปฏิบัติการใดที่จะสามารถเอาใจใส่ทำให้สมบูรณ์ในเรื่องของ Protocol แม้แต่ระบบปฏิบัติการ Windows ก็ไม่ละเว้น

Application Programming ที่ส่วนมากใช้กันอย่างกว้างขวางสำหรับ TCP/IP Programmer คือ Windows Sockets Library ซึ่งได้ถูกใช้ในการส่งออกที่ Internet สำหรับการใช้สิ่งเคลื่อน TCP/IP Application และเป็นธรรมดาอยู่แล้วเมื่อเวลาผ่านไป Microsoft ได้คิดค้น API สำหรับ TCP/IP ภายใต้ระบบปฏิบัติการของ Windows

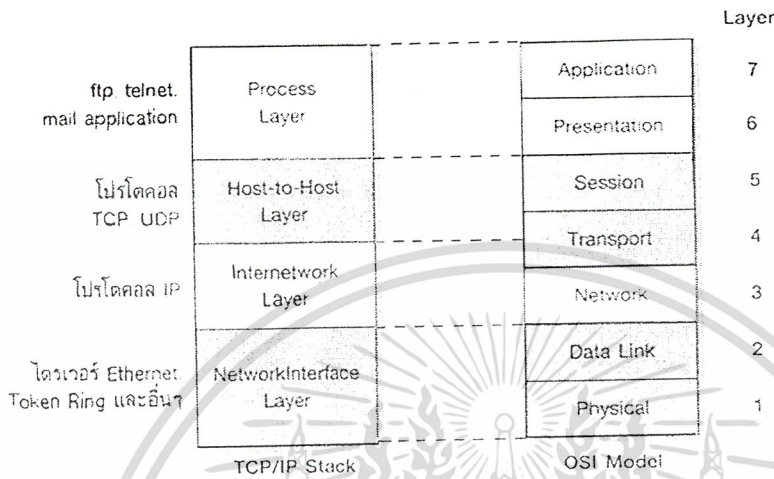
ในการใช้งานของ Windows Sockets จะไม่ถูกจำกัดในเรื่องของ TCP/IP Programming Windows Sockets จะแสดงให้เห็นว่าแม้การติดต่อ ที่เป็นงานหนัก เมื่อใช้ TCP/IP หรือ Protocol อื่น ๆ สำหรับการส่งค่าจริง ๆ ของข้อมูลอย่างไรก็ตามในปัจจุบัน Winsock Implementation จะจำกัดการใช้งานของ Windows Sockets ในการติดต่อกับ TCP/IP

โปรโตคอล TCP/IP มีกลไกการทำงานเป็นชั้นหรือ Layer เรียงต่อกัน โดยในแต่ละ Layer จะมีการทำงานเทียบได้กับ OSI Model มาตรฐาน แต่บาง Layer ของโปรโตคอล TCP/IP จะทำงานเทียบกับ OSI หลาย Layer ปนกันซึ่งในแต่ละ Layer ของโปรโตคอล TCP/IP จะประกอบด้วย

- Process layer
- Host-to-Host layer
- Internetwork layer
- Network Interface layer

โดยเมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นดังรูปที่ 3.1 ซึ่งเราจะเห็นว่าบางกลไกของโปรโตคอล TCP/IP เทียบได้กับมาตรฐาน OSI model สองชั้น หรือบางกลไกก็จะทำงานคาบเกี่ยวกันระหว่างบางชั้นของ OSI model ตัวอย่างเช่น กลไกก็จะทำงาน TCP/IP ในส่วน Network Interface layer เมื่อเทียบกับมาตรฐาน OSI model จะเทียบได้กับ Data Link layer และ Physical layer 2 ชั้นรวมกัน เป็นต้น ในแต่ละกลไกของโปรโตคอล TCP/IP จะมีโปรโตคอลอื่น ๆ ชุดของ TCP/IP ร่วมทำงานอยู่ด้วย ซึ่งจะกล่าวโดยละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดง TCP/IP stack เปรียบเทียบกับมาตรฐาน OSI

3.1 Process layer

จากรูปแสดงลำดับชั้นการทำงาน ของโพรโตคอล TCP/IP เทียบกับมาตรฐาน OSI model นั้น ในชั้นบนสุดเรียกว่า Process layer ทำงาน 2 หน้าที่เทียบได้กับ Application layer และ Presentation layer ในชั้นนี้จะรองรับการทำงาน ของ Application ต่าง ๆ ที่ทำงานเป็นโปรเซส อยู่ในเครื่องเซิร์ฟเวอร์ให้บริการและเครื่องที่ขอใช้บริการ หรือ ไคลเอนต์ (client) ซึ่งจะติดต่อกันผ่านโพรโตคอลเฉพาะแอฟพลิเคชันอีกทีหนึ่ง ตัวอย่างเช่น เมื่อผู้ใช้งานอินเทอร์เน็ตต้องการ โอนถ่ายไฟล์ หรือ download ข้อมูลจากเครื่องเซิร์ฟเวอร์ที่ให้บริการ โดยอาจจะเรียกใช้โปรแกรม ftp client ทั่วไป เช่น โปรแกรม WS ftp ติดต่อกับโปรเซส ftp ที่กำลังให้บริการอยู่ที่เครื่องเซิร์ฟเวอร์จากนั้นตัวโปรเซส ftp ก็จะเรียกใช้โพรโตคอล FTP (ftp transfer protocol) เพื่อทำการโอนถ่ายไฟล์นี้ หรือถ้าผู้ใช้ต้องการเรียกใช้งานคอมพิวเตอร์เครื่องที่อยู่ห่างไกลออกไปด้วยการใช้โปรแกรม telnet ที่เครื่องเซิร์ฟเวอร์ ใช้บริการ ตัวโปรเซส telnet ที่อยู่ก็จะเรียกใช้โพรโตคอล Telnet เพื่อติดต่อกันหรือในกรณีที่มีการเรียกใช้โปรแกรม web browser เช่น Netscape Navigator เพื่อเรียกดูเว็บเพจในเว็บไซด์ CNN ที่เครื่องซึ่งให้บริการเว็บของ CNN ก็จะมีโปรเซส HTTP (HyperText tranfer Protocol) ทำงานอยู่และจะติดต่อกับผู้ใช้ผ่าน โพรโตคอล HTTP เป็นต้น

การทำงานของแอฟพลิเคชันต่าง ๆ จะอยู่ที่ Process layer นี้ และมีการติดต่อกันตามแต่โพร

โตคอลเฉพาะแล้วแต่แอฟพลิเคชันที่ใช้งาน จากการที่ Process layer ของ TCP/IP รองรับให้โพรโตคอลเหล่านี้เป็นที่ยอมรับทั้งในระดับสากลเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอลอื่นทำงานได้หลายโปรเซส และหลายโปรโตคอลได้พร้อมกันนั้น ทำให้ผู้ใช้สามารถ เปิดโปรแกรมใช้งานได้หลาย ๆ อย่างพร้อมกัน เช่น เปิดโปรแกรม Internet Explorer เพื่อเรียกดูเว็บเพจ พร้อมกับใช้งานโปรแกรม Outlook Express เพื่อรับส่งอีเมลไปพร้อมกันได้โดยไม่ต้องรอให้ทำงานอย่างหนึ่งอย่างใดให้เสร็จก่อนหรือในปัจจุบันมีการพัฒนาโปรแกรม web browser ให้สามารถเรียกใช้งานโปรโตคอล FTPO ได้โดยไม่ต้องไปหาโปรแกรมอื่นมาใช้

โปรโตคอลหลัก ๆ ที่ทำงานใน Process layer ซึ่งผู้ใช้อาจจะคุ้นเคยกันได้แก่ FTP (File Transfer Protocol) , Telnet , HTTP (Hyper Text Transfer Protocol) และ SMTP (Simple Mail Transfer Protocol) นอกจากนี้ยังมีโปรโตคอลอื่นมีอยู่เบื้องหลัง ซึ่งทำงานโดยที่ผู้ใช้ไม่สามารถมองเห็นได้จากโปรแกรมหรือไม่ได้มีการใช้งานโดยตรง เช่น

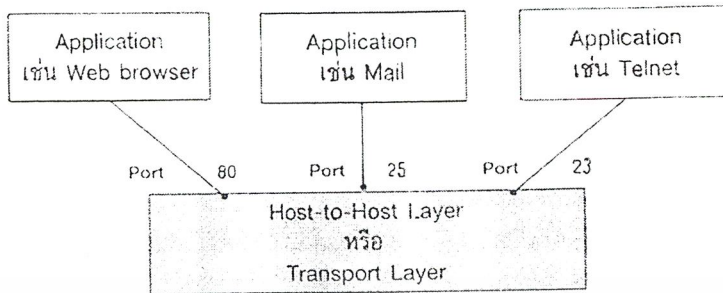
- โปรโตคอล DNS (Domain Name System) ที่ทำหน้าที่แปลงข้อมูลชื่อ domain name หรือชื่อเว็บไซต์ทั้งหลายให้เป็นหมายเลข IP address
- โปรโตคอล SNMP (Simple Network Management Protocol) ใช้ในการควบคุมและตรวจสอบอุปกรณ์ที่อยู่ในเครือข่าย
- โปรโตคอล DHCP (Dynamic Host Configuration Protocol) ทำหน้าที่แจกจ่าย ข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายที่เชื่อมต่ออยู่

3.2 Host-to-Host layer

ผู้ใช้งานอินเทอร์เน็ตเคยสงสัยหรือไม่ว่าเครื่องเซิร์ฟเวอร์ที่ให้บริการต่าง ๆ เช่น เว็บเซิร์ฟเวอร์นั้น เมื่อมีผู้เข้ามาเรียกใช้บริการพร้อมกันหลายคน จะมีวิธีการส่งข้อมูลกลับไปยังต้นทางได้อย่างไรโดยไม่ผิดพลาด ซึ่งบางครั้งผู้ใ้รายหนึ่งอาจจะเปิดโปรแกรม web browser ซ้อนกันเพื่ออ่านข้อมูลจากเว็บเพจอื่น ๆ พร้อมกันไปได้ ดังนั้นระบบจะทราบได้อย่างไรถึงการจัดส่งข้อมูลได้อย่างถูกต้อง

การทำงานที่ชั้นของ Host-to-Host layer นี้จะมีบทบาทในการจัดการต่อจาก Process layer บางครั้งเรามีเรียกชั้น Host-to-Host ว่าเป็น Transport layer ซึ่งไม่ใช่ชั้นของ Transport layer ในมาตรฐาน OSI model การทำงานของ Host-to-Host layer นี้จะมีการสร้าง connection หรือการเชื่อมต่อกันระหว่างแอปพลิเคชัน กับ Host-to-Host layer โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลนี้เรียกว่า port หรือ docket (คำว่า port ในที่ไม่ได้หมายถึง port ทางฮาร์ดแวร์) และในแต่แอปพลิเคชัน ก็จะสร้างการเชื่อมต่อผ่าน port ได้พร้อมกันหลายแอปพลิเคชัน ซึ่งการใช้งาน port ของแต่ละแอปพลิเคชันที่อยู่ในชั้น Process layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้ และแต่ละโปรโตคอลจะมีการใช้งาน port หมายเลขต่าง ๆ ไม่ซ้ำ กัน ตามรูปที่ 3.2

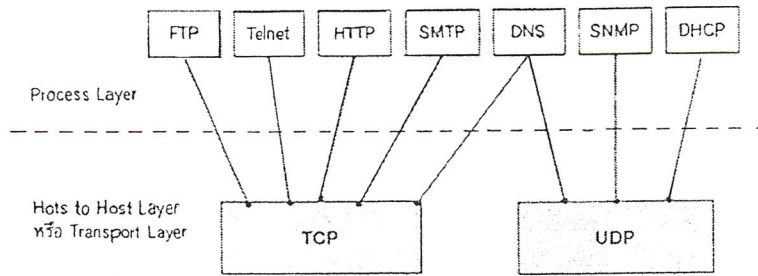
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แอปพลิเคชันหรือโปรเซสต่าง ๆ สื่อสารกับ Host-to-Host layer ผ่านจุดเชื่อมต่อ หรือport ส่วนหมายเลขในรูปคือหมายเลข port ที่โปรเซสใช้งานเช่น เว็บ หรือ โปรเซส http ใช้งาน port 80 ในการส่งผ่านข้อมูล เป็นต้น

เมื่อ แอปพลิเคชันทำงานผ่าน โปรโตคอลในชั้น Process layer จะมีการส่งผ่านข้อมูลไปยัง Host-to-Host layer ที่ชั้นนี้จะมีการเชื่อมต่อผ่าน port ที่กำหนด ทำให้การรับส่งข้อมูลในแต่ละโปรโตคอลทำได้ถูกต้อง ถึงแม้ว่าในเครื่องเซิร์ฟเวอร์ที่ให้บริการจะมีการทำงานอยู่หลายโปรเซสที่แตกต่างก็ตาม หรือมีผู้ใช้บริการเข้ามาใช้งานพร้อมกันจำนวนมากและหลายแอปพลิเคชันในเวลาเดียวกัน ในชั้น Host-to-Host หรือ Transport layer ของ TCP/IP นี้จะมีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกันคือ โปรโตคอล TCP และโปรโตคอล UDP (User Datagram Protocol) ในการส่งผ่านข้อมูลลงไปที่ยื่นถัดๆ ไป เราจะเห็นว่าโปรโตคอล TCP และ UDP (User Datagram Protocol) ในการส่งผ่านข้อมูลลงไปที่ยื่นถัดๆ ไป เราจะเห็นว่าโปรโตคอล TCP และ UDP จะถูกผนึกเข้าไปในโปรโตคอล IP อีกทีหนึ่งและส่งต่อไปยังเครือข่ายอินเทอร์เน็ตต่อไป

ตัวโปรโตคอล TCP และโปรโตคอล UDP จะมีแอปพลิเคชัน เฉพาะเพื่อเรียกใช้งานแยกกันคือ แอปพลิเคชันที่ใช้โปรโตคอล FTP, Telnet, HTTP และ SMTP จะมีการส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล TCP ส่วนแอปพลิเคชันที่ใช้โปรโตคอล UDP และสำหรับโปรโตคอล DNS นั้นจะสามารถเรียกใช้งานได้ ทั้ง TCP และ UDP ดังรูปซึ่งเหตุผลที่มีการเรียกใช้โปรโตคอล TCP และ UDP แตกต่างกัน ก็เนื่องมาจากวิธีการทำงานของทั้งสองโปรโตคอลต่างกันนั่นเอง



รูปที่ 3.3 โพรเซสต่าง ๆ ที่เรียกใช้ Transport layer เพื่อส่งผ่านข้อมูลโดยอาศัย Port ซึ่งในแต่ละ โพรเซสจะเรียกใช้งาน port เฉพาะแตกต่างกันยกเว้น DNS ที่สามารถใช้งานได้ทั้ง TCP และ UDP

3.3 โพรโทคอล TCP

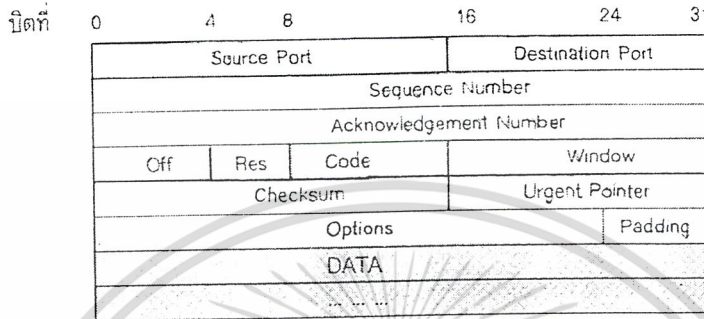
โพรโทคอล TCP (Transmission Control Protocol) เป็น โพรโทคอลที่มีกรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงปริมาณข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อย ๆ ก่อน แล้วจึงจะส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูล Datagram ใหม่ให้ต่อเนื่องกันและประกอบกลับเป็นข้อมูลทั้งหมดได้ ซึ่งจะแยกข้อมูลลบลดตลตลที่ไม่ถูกต้องออกคั้นนั้นแอฟพลิเคชันหรือโพรเซสเซอร์ใดที่อาศัยการส่งผ่านข้อมูลด้วยโพรโทคอล TCP จะต้องใช้หน่วยความจำและขนาดของช่องสัญญาณ (bandwidth) มากกว่า UDP

การติดต่อระหว่างกันจะต้องเป็นแบบ Connection-oriented คือต้องมีการสร้างการติดต่อกันเป็น session ทั้ง 2 ด้านเสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (full duplex) เหมือนกัน การใช้โทรศัพท์ติดต่อกัน เมื่อผู้ติดต่อต้นทางเรียกให้ฝ่ายตรงข้ามรับสายแล้ว จึงเริ่มสนทนา เช่น พูดคำว่า สวัสดี หรือ ฮัลโล่ กันก่อนเพื่อให้แน่ใจว่าฝ่ายตรงข้ามพร้อมจะติดต่อกับ จากนั้นจึงเริ่มต้นติดต่อกัน และเมื่อต้องการจะเลิกการติดต่อก็จะมีการพูดคำว่า สวัสดี , ให้ฝ่ายตรงข้ามทราบว่าจะเลิกการติดต่อและวางสายไป ซึ่งในระหว่างการติดต่อกันนั้น แม้ว่าฝ่ายใดฝ่ายหนึ่งหรือทั้งสองฝ่ายจะเจียบไป คือ ไม่พูดอะไรเป็นเวลานาน ๆ แต่การเชื่อมโยงระหว่างทั้งสองด้านยังคงมีอยู่ไม่ขาดไป จนกว่าฝ่ายหนึ่งจะวางสาย เช่น เดียวกับการติดต่อกันด้วยกลไกโพรโทคอล TCP เมื่อแอฟพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โพรโทคอลที่เหมาะสมในชั้น Process layer ติดต่อกับไปและมีการสร้างช่องส่งข้อมูลผ่าน port ที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโพรโทคอล TCP

ในระหว่างการรับส่งข้อมูลนี้ โพรโทคอล TCP จะเพิ่มขบวนการสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความน่าเชื่อถือของการส่งผ่านข้อมูลโดยโปรโตคอล TCP จะมีมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบมากกว่าในการทำงานเช่นกัน



รูปที่ 3.4 รูปแบบของ TCP packet จะเห็นว่าฟิลด์ Acknowledgement Number และข้อมูล Checksum เพื่อใช้ตรวจสอบการเดินทางของข้อมูล ส่วน Header มีข้อมูลมากทำให้ต้องอาศัยทรัพยากรของระบบทำงานมาก

3.4 โปรโตคอล UDP

ใน Host-to-Host layer นอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ก็ยังมีโปรโตคอล UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วยในการรับส่งข้อมูลผ่าน โปรโตคอล UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลนั้น ๆ ด้วย เนื่องจากโปรโตคอล UDP ไม่สัญญาตรวจสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอปพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง

ตามรูป 3.3 จะเห็นว่าโปรโตคอลชั้นบนขึ้นไปที่ใช้การส่งผ่านข้อมูลโดยโปรโตคอล UDP เช่น โปรโตคอล SNMP (ใช้ควบคุมและจัดการอุปกรณ์ในเครือข่าย หรือโปรโตคอล DHCP (ใช้ส่งข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายได้ใช้งาน) การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูลที่มีการรับส่งจะไปทำในชั้นตอนของโปรโตคอลชั้นที่สูงกว่าแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างขั้นตอนกลไกการทำงานโดยใช้โปรโตคอล UDP มีดังต่อไปนี้

1. ในชั้นของ Process layer เมื่อโปรแกรมควบคุมอุปกรณ์เครือข่ายเช่น โปรแกรม Network management ต้องการส่งข้อมูลไปยังอุปกรณ์ที่ต้องการ แอปพลิเคชันนั้นจะติดต่อผ่านโปรโตคอล SNMP ในชั้น Process layer

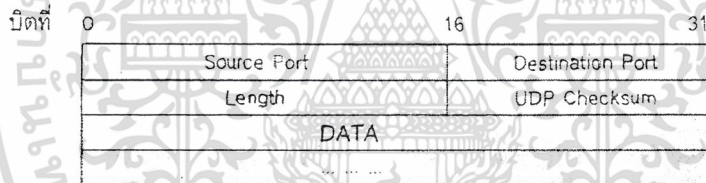
2. โปรโตคอล SNMP จะติดต่อกับโปรโตคอล UDP ในชั้นถัดไป เพื่อขอติดต่อผ่าน port ที่กำหนด

3. โปรโตคอล SNMP เตรียมข้อมูลที่ส่ง รวมทั้งที่อยู่ปลายทาง

4. โปรโตคอล SNMP ส่งผ่านข้อมูลให้โปรโตคอล UDP ที่อยู่ในชั้น Host-to-Host layer

5. โปรโตคอล UDP ทำหน้าที่ผนึกข้อมูลหรือ datagram นั้น ไปกับโปรโตคอล IP ในชั้นถัดลงไป เพื่อส่งข้อมูลออกจากเครื่อง

ซึ่งจะเห็นว่ากลไกที่ต่างจากการส่งข้อมูลด้วยโปรโตคอล TCP ซึ่งจะต้องมีการติดต่อกันก่อน และทั้งสองฝ่ายรับทราบการรับส่งข้อมูลของช่องการส่งข้อมูลนั้น



รูปที่ 3.5 รูปแบบของ UDP packet จะมีฟิลด์ข้อมูลส่วน header น้อยมากและไม่ข้อมูลส่วนการตรวจสอบข้อมูล ทำให้ UDP packet มีขนาดเล็กและใช้หน่วยความจำหรือทรัพยากรขอรระบบน้อย

3.5 Internetwork Layer

ในระดับล่างต่อมาในชั้น Internetwork layer มีหน้าที่ส่งผ่านข้อมูลในระหว่างเครือข่าย โดยมีโปรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ บน อินเทอร์เน็ต คือ โปรโตคอล IP (Internet Protocol) นอกจากนี้ในชั้น Internetwork layer ยังมีโปรโตคอลทำงานอยู่ด้วยอีก 2 ชนิด คือ โปรโตคอล Internet control Message protocol (ICMP) และ โปรโตคอล Address Resolution Protocol (ARP)

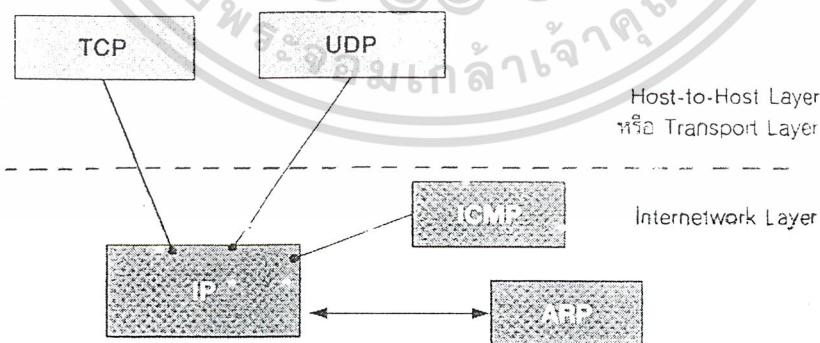
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 โพรโทคอล IP

โพรโทคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host layer เพื่อส่งข้ามไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง แม้ว่าจะมีเครือข่ายเชื่อมต่อกันอยู่ในอินเทอร์เน็ตเป็นล้าน ๆ เครือข่ายก็ตาม เนื่องจากโพรโทคอล IP มีข้อมูลตำแหน่ง IP ปลายทางที่จะส่งข้อมูลไปโดยทำงานร่วมกับอุปกรณ์ Router เพื่อส่งข้อมูลผ่านสวิตช์ (switch) ไปยังปลายทาง ตัววงจรผ่าน หรือ switch นี้อาจเป็น Gateway หรือ Router ในระบบเครือข่ายก็ได้ ซึ่งในข้อมูลของโพรโทคอล IP จะมีข้อมูลของหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์ และจำเครื่องที่ถูกต้องอีกทีหนึ่งด้วยโพรโทคอล ARP ตามรูป 3.6 ที่จะแสดงการติดต่อกันระหว่างโพรโทคอลในชั้นของ Host-to-Host layer และ internetwork layer

3.7 โพรโทคอล ICMP

หน้าที่หลักของโพรโทคอล ICMP (Internet Control Message Protocol) คือการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบคือส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นอกจากนี้โพรโทคอล ICMP ยังถูกเรียกใช้งานจากเครื่องเซิร์ฟเวอร์และ Router อีกด้วยเพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโพรโทคอล ICMP นั้นจะทำงานควบคู่กับโพรโทคอล IP ในระดับเดียวกัน และข้อความต่าง ๆ ที่แจ้งให้ทราบจะถูกผนึกอยู่ภายในข้อมูลของ IP (IP datagram) อีกทีหนึ่ง



รูปที่ 3.6 โพรโทคอล TCP และ UDP อาศัยโพรโทคอล IP ที่อยู่ชั้นล่างเพื่อส่งผ่านข้อมูลระหว่างเครือข่าย และในชั้น Internetwork Protocol ยังมีโพรโทคอล ICMP ทำหน้าที่ส่งข้อความแจ้งเตือนและโพรโทคอล ARP ทำหน้าที่แปลงเลขหมาย IP ไปเป็นหมายเลขของฮาร์ดแวร์จริง โยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความที่โปรโตคอล ICMP ส่งนั้นแบ่งออกได้ 2 แบบ คือ ICMP error message หรือ ข้อความแจ้งข้อผิดพลาด และ ICMP query หรือข้อความเรียกขอข้อมูลเพิ่มเติม ตัวอย่างกลไกการทำงานของโปรโตคอล ICMP เช่นเมื่อ มีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือ ขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ที่ Router จะส่งข้อความแจ้งเป็น ICMP message ที่ชื่อ destination unreachable ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้งข้อความก็จะมี ส่วนข้อมูล IP datagram ที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูล ได้รับข้อความแจ้งแล้วก็จะทราบได้ว่าจุดที่เกิดปัญหานั้นอยู่ที่ใด

ดังนั้นโปรโตคอล ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบว่าเครื่องเซิร์ฟเวอร์ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในเครือข่าย อินเทอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง ping มีการเรียกใช้งานโปรโตคอล ICMP แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง

3.8 โปรโตคอล ARP

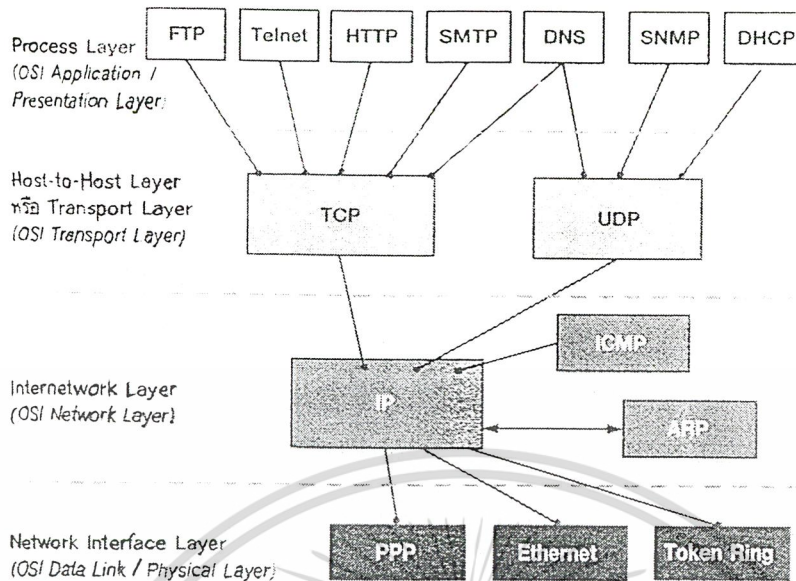
โปรโตคอล ARP (Address Resolution Protocol) ถูกเรียกใช้งานโดยโปรโตคอล IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทางตัวอย่างเช่น เว็บเซิร์ฟเวอร์เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต และในการเชื่อมต่อนี้ต้องอาศัย Network Interface Card (NIC) หรือ LAN card ติดตั้งอยู่ที่ LAN card นี้เองจะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำ กับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานในโปรโตคอล TCP/IP ก็จะต้องมีการ กำหนดหมายเลข IP address ประจำตัวเพื่อใช้อ้างอิงกัน และโปรโตคอล ARP จะทำหน้าที่แปลงค่า หมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงให้ในระดับการทำงานที่ Internetwork layer นี้ซึ่งกลไกการแปลงนี้เรียกว่า Address resolution

3.9 Network Interface Layer

เนื่องจากในด้านกายภาพของเครือข่ายนั้นมีหลายวิธีการและหลายรูปแบบในการเชื่อมต่อระบบให้เป็นเครือข่าย แต่อย่างไรก็ตามในเครือข่ายอินเทอร์เน็ตนี้ ข้อมูลหรือ IP datagram จะถูกถ่ายทอด และส่งผ่านไปยังปลายทางโดยไม่คำนึงถึงรูปแบบการเชื่อมต่อทางกายภาพ ไม่ว่าจะเป็นการใช้เครือข่ายใยแก้วนำแสงหรือเครือข่ายสาย Unshielded Twist Pair (UTP) เชื่อมต่อเป็นแบบเครือข่าย Ethernet ธรรมดาหรือเครือข่าย Token Ring , ATM , ISDN ฯลฯ ก็ตาม

การทำงานระดับต่ำสุดต่อจาก Internetwork layer จะเป็นการแปลงข้อมูล IP datagram ให้อยู่ในรูปที่เหมาะสม และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่ายต่อไปซึ่งในชั้น Network Interface layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นการรวม 2 Layer เข้าด้วยกัน คือ Data link layer และ Physical layer กล่าวโดยสรุปคือการทำงานในชั้นต่าง ๆ ตามโครงสร้างของโปรโตคอล TCP/IP จะมีลักษณะดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 โครงสร้างของโปรโตคอล TCP/IP ในแต่ละชั้นหรือ layer จะมีโปรโตคอลหลักทำหน้าที่ต่าง ๆ และส่งผ่านข้อมูลไปยังเครือข่าย และ ออกสู่อินเทอร์เน็ต

กล่าวโดยสรุปก็คือ โปรโตคอล TCP/IP ทำงานโดยแบ่งเป็นชั้นเทียบกับ OSI model ได้กลไกในการทำงานของโปรโตคอล TCP/IP มี 4 ชั้น ซึ่งในชั้นแรกคือ Process layer ทำหน้าที่ติดต่อกับแอปพลิเคชันและโปรโตคอลที่แอปพลิเคชันนั้น ๆ ใช้งานและ ส่งต่อมาให้ชั้น Host-to-Host layer เพื่อติดต่อกันระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องผู้ขอใช้บริการ ในชั้นนี้จะมีการสร้าง Session หรือการเชื่อมต่อระหว่างระบบขึ้นตามแต่ละโปรโตคอลที่ต้องการ ต่อมาเป็นการผนึกข้อมูลไปเป็น IP datagram ที่ชั้น Internetwork layer โดยอาศัยโปรโตคอล IP เพื่อให้สามารถติดต่อส่งข้อมูลข้ามเครือข่ายไปยังเครือข่ายและเครื่องที่ถูกต้องได้ และสุดท้ายการส่งข้อมูลออกสู่โลกภายนอกต้องอาศัยกลไกในชั้น Network Interface layer เพื่อแปลงข้อมูลใหม่ เพิ่มข้อมูลใหม่ที่จำเป็นในการอ้างอิงตำแหน่งและแปลงข้อมูลเป็นสัญญาณไฟฟ้าส่งออกไปเครือข่าย และอาจจะออกไปยัง Gateway หรือ Router เพื่อข้ามเครือข่ายออกไปยังเส้นที่กำหนดไว้เพื่อผ่านข้อมูลไปยังระดับล่างและออกสู่เครือข่ายอินเทอร์เน็ตในที่สุด

ตารางสรุปหมายเลขบางส่วนของ port ที่ใช้งาน โดย TCP และ UDP เราจะเห็นว่าในแต่ละชั้นของโครงสร้าง TCP/IP Stack มีการใช้งาน โปรโตคอลต่างอยู่หนึ่งโปรโตคอล

โปรโตคอลที่ใช้งาน	Port หรือ socket เชื่อมต่อ (โดยฐาน 10)	โปรโตคอลในระดับ Host-to-Host	รายละเอียด
BootP	67	UDP	BOOTstrap Protocol ด้านเซิร์ฟเวอร์
BootP	68	UDP	BOOTstrap Protocol ด้านไคลเอนต์
DHCP	67	UDP	Dynamic Host Configuration Protocol ด้านเซิร์ฟเวอร์
DHCP	68	UDP	Dynamic Host Configuration Protocol ด้านไคลเอนต์
DNS	53	UDP/TCP	Domain Name System
FTP	21	TCP	File Transfer Protocol ด้านเซิร์ฟเวอร์ที่ควบคุม
FTP	20	TCP	File Transfer Protocol ด้านเซิร์ฟเวอร์ที่ส่งข้อมูล
HTTP	80	TCP/UDP	Hyper Text Transfer Protocol ด้านเซิร์ฟเวอร์

3.10 กลไกของโปรโตคอล IP

ในการส่งผ่านข้อมูลหรือ IP datagram ไปยังเครือข่ายอินเทอร์เน็ตนั้น โปรโตคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่ง IP datagram นั้นจะเป็นภายในเครือข่ายของตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยการพิจารณาโปรโตคอล IP address ปลายทางว่าส่วนที่เป็นค่าหมายเลขเครือข่าย (network address) จะเหมือนกับค่าหมายเลขเครือข่ายของ IP address ต้นทางหรือไม่ ถ้าตรงกันแสดงว่าการส่งข้อมูลอยู่ภายในเครือข่ายเดียวกัน แต่ถ้าค่าต่างกันแสดงว่าต้องส่งข้อมูลไปยังปลายทางที่อยู่คนละเครือข่ายกัน

การส่งข้อมูลภายในเครือข่ายเดียวกัน มีกลไกดังนี้

1. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP (address Resolution Protocol) เพื่อแปลงหมายเลข IP ปลายทางให้เป็นค่าหมายเลขฮาร์ดแวร์ เช่น MAC address
2. เมื่อโปรโตคอล IP รับค่าหมายเลขฮาร์ดแวร์แล้ว ก็จะส่งข้อมูลนั้นไปยังฮาร์ดแวร์ที่

ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลข้ามเครือข่าย มีกลไกดังนี้

1. โพรโทคอล IP จะตรวจสอบพบว่าหมายเลข IP address ปลายทางอยู่บนละเครือข่ายกัน โดย โพรโทคอล IP จะอ่านค่า IP address ของ Router เพื่อเตรียมส่งข้อมูลไปที่ Router แทนซึ่งในที่นี้จะมีการกำหนดเป็น default router
2. โพรโทคอล IP จะเรียกใช้บริการ โพรโทคอล ARP เพื่อแปลงค่า IP Address ของ Router ให้เป็นค่าหมายเลขฮาร์ดแวร์
3. โพรโทคอล IP ส่งข้อมูล IP datagram ไปยัง Router ที่กำหนดไว้ จากนั้น Router ส่งข้อมูลข้ามเครือข่ายไปตามขั้นตอน

3.10.1 IP Datagram

หัวใจหลักของการส่งข้อมูล TCP/IP คือ IP datagram ซึ่งจะเป็นชุดของข้อมูลซึ่งประกอบด้วย Source , Address ปลายทาง , Type of Service Information , User data และ Error Correction information

IP datagram จะประกอบด้วยส่วนของ Header Block ของข้อมูลซึ่งข้อมูลสามารถที่จะกำหนดขึ้นอยู่กับชนิดของ Service และความต้องการของผู้ใช้ ในส่วน Header จะประกอบด้วยชุดของ well-Defined Field

3.10.2 IP Header

ในส่วนหัวของ IP datagram หรือ IP Header โดยส่วนมากจะประกอบด้วย 20 bytes ตามรูปที่ 2.2 จะแสดง IP Header

version(4bit)	Header length(4bit)	Type of Service(8bit)
Packet Length (16bit)		
Packets Identifier (16bit)		
Fragmentation Data (16bit)		
Time to live (8bit)	Protocol(8bit)	
Header Checksum(16bit)		
Source Address(32bit)		
Destination address (32 bit)		

รูปที่ 3.8 แสดง IP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรณีสืบค้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามรูปที่ 3.8 ครั้งแรกจะพิจารณาสิ่งที่น่าสนใจในส่วนของ Protocol . Source . Address และ Destination Address Field ในส่วนของ Protocol จะถูกกำหนดวิธีของ IP Packet จะถูกแปลตัว ๆ ไปค่าต่าง ๆ จะถูกกำหนดสำหรับส่วนนี้ ในส่วนของ Address จะแสดงให้เห็นส่วนของ Host Address ซึ่ง Address นี้จะเพียงหมายเลขเดียวกันที่เหมือนกันในระบบ Internet

3.10.3 IP Host Address and Routing

IP Host Address จะเป็นส่วนข้อมูล 32 bits ซึ่งมีเพียงหมายเลขเดียวกันที่เหมือนกันซึ่งใช้ในการติดต่อกับ Internet Host ส่วนของ Gateway (เป็นตัวทำหน้าที่ Interface Network ที่มีมากกว่า 1 Network) จะมี Host อยู่มากมาย

ตามหลักที่ถูกต้องคือ 4 bytes ของ Internet host address มักเขียนให้อยู่ในรูปของเลขฐานสิบ ตัวอย่างเช่น 121.0.0.1

Internet host address ส่วนมากจะถูกแบ่งออกเป็น 2 ส่วน คือ Network address และ Actual host address ถ้าเรียงตามความยาวทั้งสองตาม address โดยมากจะขึ้นอยู่กับข้อกำหนด bytes ใน address เสียเป็นส่วนใหญ่

Class A Network address จะสังเกตได้ว่า byte แรกจะอยู่ระหว่าง 0 ถึง 127 และจะประกอบไปด้วย 8 bit network จะประกอบด้วย 24 bit host address เนื่องจาก address เริ่มต้นที่ 0 และ 127 จะถูกสงวนไว้ ซึ่งสามารถมีค่าสูงสุดคือ 126 Class A Subnets ใน Class A นี้สามารถมีได้ทั้งหมด 16,777,214 Hosts (Address อยู่ในรูป nnn.0.0.0 และ nnn.255.255.255 จะถูกสงวนเอาไว้)

Class B Network address จะสังเกตได้จากตัวเลขหลักแรกอยู่ระหว่าง 128 และ 191 , Class B Network จะประกอบไปด้วย 16 bit Network address และ 16 bit host address และสามารถมี 16,383 Subnets Class B subnet สามารถที่จะบรรจุได้ถึง 65,534 hosts (address อยู่ในรูป nnn.mmm.0.0 และ nnn.mmm.255.255 จะถูกสงวนไว้)

Class C Network address จะสังเกตได้จากตัวเลขหน้าสุดอยู่ระหว่าง 192 และ 223 ซึ่งจะมีได้ 2,097,152 Subnets Class C Subnets สามารถที่จะบรรจุได้ 254 hosts (โดยเริ่มที่ address nnn.mmm.bbb.0 และ nnn.mmm.bbb.255 จะถูกสงวนไว้)

Class D Network address (โดยทั่วไปตัวเลข byte แรกจะอยู่ระหว่าง 224 ถึง 255) จะถูกสงวนเอาไว้เพื่อ IP Multitasking ที่ถูกจำกัดจากในส่วนของ IP broadcastion คือจะไม่ไปยุ่งเกี่ยวกับ Winsock Programmer

3.10.4 Host Names

ในระหว่างการคิดค้นประดิษฐ์ Internet ซึ่งตัวเลขที่จะนำมาแทน Host address ซึ่งจะมีไม่เพียงพอและอีกอย่างหนึ่งคือตัวเลขเหล่านี้เป็นการยากที่จะจดจำ และอีกอย่างถ้าเกิด host address มีการเปลี่ยนแปลงอันเนื่องมาจาก หลาย ๆ สาเหตุ จึงทำให้เกิดความยุ่งยาก ดังนั้นระบบ Naming System จึงได้ถูกสร้างขึ้นมาซึ่งใช้ในการ Map ตัวเลขของ IP address ไปเป็น memories host names (ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทุกวันนี้จะมีอยู่ 5 hosts บน Internet ทุก ๆ hosts จะถูกปรับเปลี่ยนให้อยู่ในรูปของ Files กับข้อมูลของ Internet host ทุกตัวและ address อย่างไรก็ตามอินเทอร์เน็ตก็ได้เจริญเติบโตอย่างต่อเนื่องทำให้ดูว่าจะไม่เพียงพอ อย่างแรกคือการตั้งชื่อ internet host จะทำได้มีมาตรฐานอย่าง 2 คือทางด้าน Hardware สามารถที่จะแยกแยะความต้องการที่จะติดต่อกับทุกเครื่องคอมพิวเตอร์ทุกตัวที่ต่ออินเทอร์เน็ตอยู่

คำตอบของปัญหานี้คือการกำหนด Domain Name System (DNS) ซึ่งจะเป็นตัวจัดระเบียบของการตั้งชื่อ (Naming System) รูปแบบของ Domain Names ก็จะถูกคล้ายกันทุกคนที่ใช้ อินเทอร์เน็ตโดยปกติการตั้งชื่อจะอยู่ในรูปแบบคือ Host Subdomain Domain

ในส่วนนี้จะกำหนดโดยหน่วยงานหนึ่ง (ส่วนมากเป็นหน่วยงานในสหรัฐ) หรือ โดยแต่ละประเทศ โดยจะมีดังต่อไปนี้

GOV: Government Bodies

EDU: Education Institutions

COM: Commercial enterprises

MIL: Military organization

ORG: Other organization

Top-level Domain โดยใช้ชื่อประเทศ มักจะใช้ตามมาตรฐาน ISO 3166 d โดยจะใช้อักษรย่อ 2 ตัวของชื่อประเทศ

การทำงานเกี่ยวกับเครือข่ายคอมพิวเตอร์นั้นเป็นเรื่องค่อนข้างซับซ้อน จนทำให้เราสร้างข้อกำหนดต่าง ๆ ขึ้นมาเพื่อให้วิธีการสื่อสารระหว่างกันเป็นไปได้โดยราบรื่น แล้วก็ทำให้เกิดเป็น โพรโตคอล (Protocol) มากมายหลายชนิดมาใช้งานร่วมกับเครือข่าย ซึ่งเรามีการแบ่ง โพรโตคอลต่าง ๆ ออกเป็นระดับชั้น เพื่อให้งานต่อการใช้งานและง่ายต่อการทำความเข้าใจงานวันเข้าโปรโตคอลที่เราคิดว่ามันทำให้เราทำงานสะดวก นั่นก็มีเพิ่มมากขึ้นจนกลายเป็นความยุ่งยาก ต่างบริษัทก็ต่างมีผลิตภัณฑ์ที่สนับสนุน โพรโตคอลต่าง ๆ ไม่เหมือนกัน ทำให้การใช้งานร่วมกันทำได้ยากขึ้น ทำให้เราต้องมาหาวิธีใช้งานโพรโตคอลมาตรฐานเหล่านั้นมาเอง

บทที่ 4

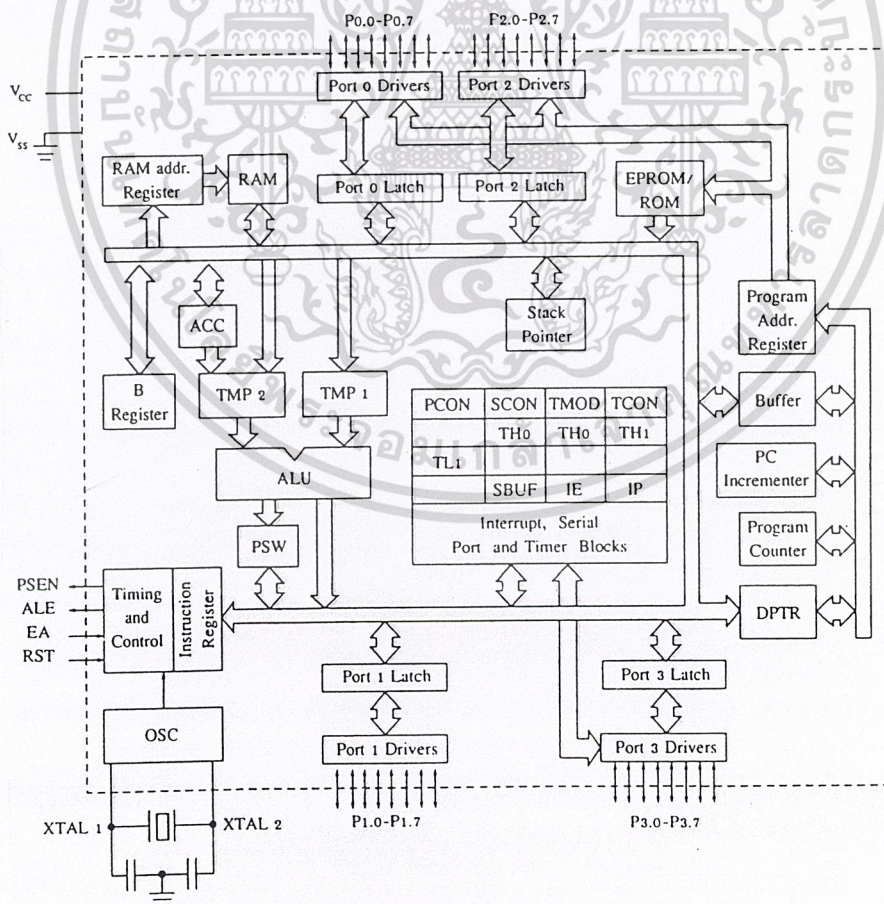
พื้นฐานไมโครคอนโทรลเลอร์ MCS-51 และสตีปเปอร์มอเตอร์

4.1 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่ง ที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรจับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรรีเลย์อิเล็กทรอนิกส์ที่ซับซ้อน ได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ

4.2 โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์ที่เลือกใช้ในครั้งนี้เป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ซึ่งมีอยู่ด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายในของมัน โดยใช้เป็นส่วนควบคุมการทำงานของชุดควบคุมมอเตอร์ ซึ่งไมโครคอนโทรลเลอร์มีหน้าที่รับคำสั่งควบคุมจากคอมพิวเตอร์มาประมวลผลเพื่อกำหนดตำแหน่งให้กับมอเตอร์รวมทั้งส่งค่าตำแหน่งปัจจุบันของมอเตอร์ให้กับคอมพิวเตอร์

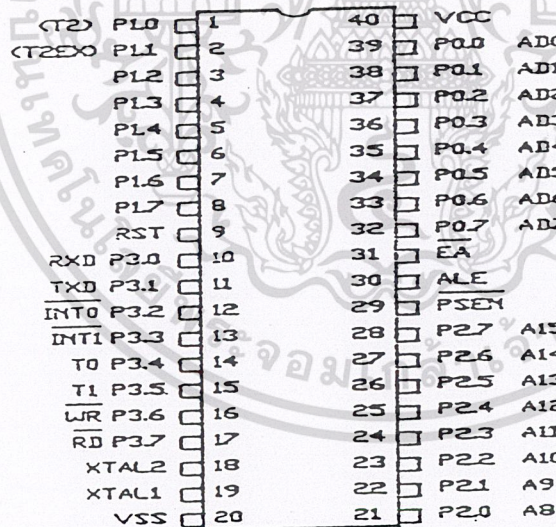


รูปที่ 4.1 โครงสร้างภายในของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม้อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

- มีหน่วยความจำ ROM 4k byte
 - มีหน่วยความจำ RAM 128 byte
 - มี Port I/O ขนาด 8 บิต 4 Port
 - มี Timer 16 บิต 2 ตัว
 - สามารถ Interrupt ได้ 5 แหล่ง
 - มีวงจรรอสซิงเกิลเตอร์และวงจรรักษาพลังงานชีพ
 - มี Port อนุกรมที่สามารถรับส่งข้อมูลแบบ Full duplex ความเร็วสูง
 - อ้าหน่วยความจำโปรแกรมภายนอกได้ 64k
 - อ้าหน่วยความจำข้อมูลภายนอกได้ 64k
 - สามารถประมวลผลทีละบิตได้
 - สามารถอ้าหน่วยความจำแบบบิตได้ 210 ตำแหน่ง
 - หนึ่งวัฏจักรคำสั่ง กินเวลาประมาณ 1 ไมโครวินาที ขณะทำงานด้วย Clock 12 Mhz
- เบอร์พื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วย 8051,8031,8751 ซึ่งแตกต่างกันที่ชนิดและหน่วยความจำภายใน



รูปที่ 4.2 แสดงตำแหน่งขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51

4.4 หน้าที่การใช้งานของแต่ละขาของ MCS-51

- ขา Vss (ขา 20) สำหรับต่อลง Ground
- ขา Vcc (ขา 40) สำหรับต่อแหล่งจ่ายไฟตรง +5 V
- ขาพอร์ต 0 (ขา 32-39) มี 8 ขา ใช้เป็นพอร์ต 0 (P0.0-P0.7) เป็นได้ทั้งเอาต์พุตและอินพุตพอร์ตและใช้ในการติดต่อหน่วยความจำเป็นไบต์ต่ำ (A0-A7)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาพอร์ต 1 (ขา 1-8) มี 8 ขา ใช้เป็นพอร์ต 1 (P1.0-P0.7) เป็นได้ทั้งเอาต์พุตและอินพุตพอร์ต โดยทั่วไปมีวงจรพูลอัพภายใน
- ขาพอร์ต 2 (ขา 21-28) มี 8 ขา ใช้เป็นพอร์ต 2 (P2.0-P2.7) เป็นได้ทั้งเอาต์พุตและอินพุตพอร์ตและใช้ในการติดต่อกับหน่วยความจำเป็นไบต์สูง
- ขาพอร์ต 3 (ขา 10-17) มี 8 ขา ใช้เป็นพอร์ต 3 (P3.0-P3.7) เป็นได้ทั้งเอาต์พุตและอินพุตพอร์ต โดยมีวงจรพูลอัพภายในและยังใช้งานในหน้าที่พิเศษต่างๆ ดังนี้
 - ขา P3.0 ใช้รับข้อมูลภายนอกแบบอนุกรม
 - ขา P3.1 ใช้ส่งข้อมูลออกภายนอกแบบอนุกรม
 - ขา P3.2 ใช้เป็นอินพุตอินเตอร์รัปต์ชนิดที่ 0
 - ขา P3.3 ใช้เป็นอินพุตอินเตอร์รัปต์ชนิดที่ 1
 - ขา P3.4 สัญญาณอินพุตให้เคาน์เตอร์ของไทมเมอร์ 0
 - ขา P3.5 สัญญาณอินพุตให้เคาน์เตอร์ของไทมเมอร์ 1
 - ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำข้อมูลภายนอก
 - ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำข้อมูลภายนอก

ทุกครั้ง

- ขา RST (ขา 9) ใช้ในการรีเซ็ตค่าภายในทั้งหมด
- ขา ALE/PROG (ขา 30) เป็นขาที่ใช้ในการส่งสัญญาณควบคุมการแลตซ์ค่าแอดเดรสไบต์ต่ำ จาก PO ในการติดต่อกับหน่วยความจำภายนอก
- ขา PSEN (ขา 29) ใช้ส่งสัญญาณสโตรบเพื่ออ่านคำสั่งจากโปรแกรม
- ขา EA/Vpp (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมภายในหรือภายนอก
- ขา XTAL (ขา 19) ใช้ต่อคริสตัลภายนอก โดยเป็นขาอินพุต
- ขา XTAL (ขา 18) ใช้ต่อคริสตัลภายนอก โดยเป็นขาเอาต์พุต

4.5 โครงสร้างหน่วยความจำใน MCS-51

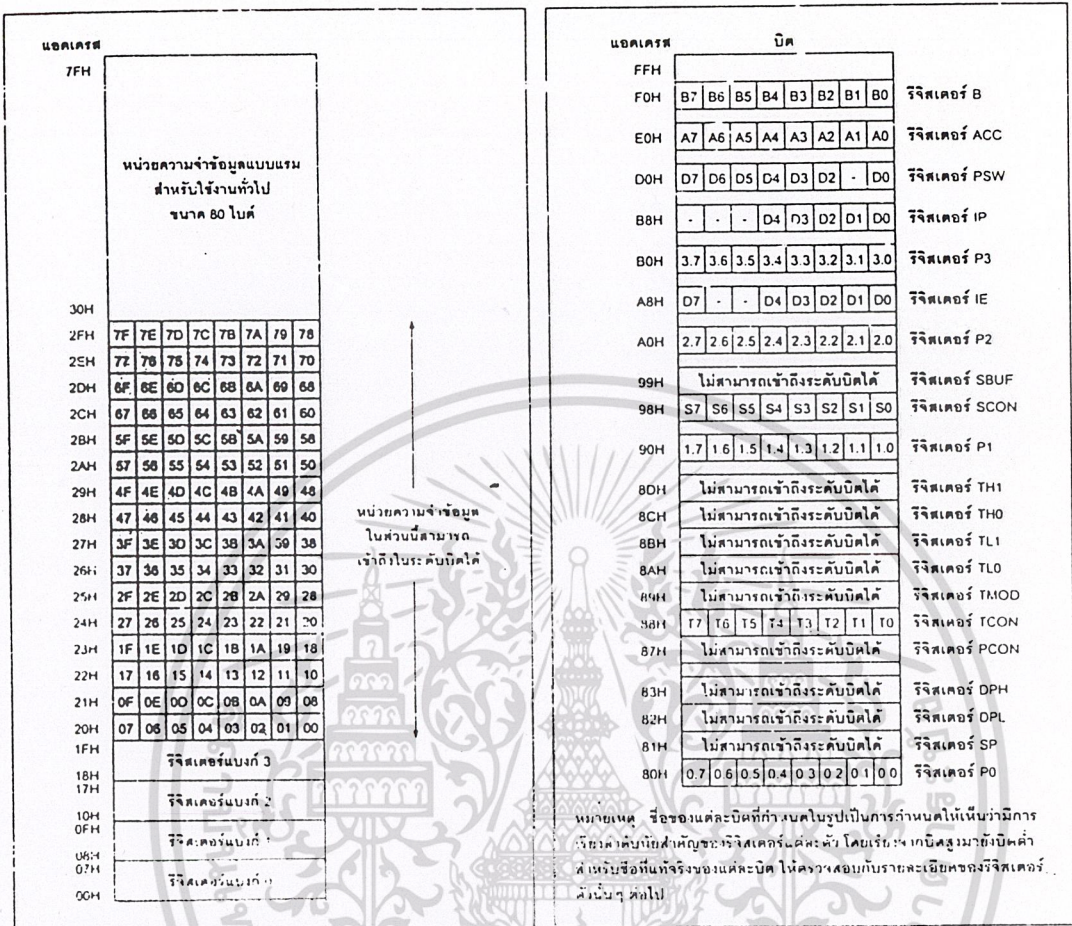
แบ่งออกเป็น 2 ส่วนคือ

1. หน่วยความจำโปรแกรม
2. หน่วยความจำข้อมูล

MCS-51 ทุกเบอร์จะมีหน่วยความจำสำหรับเก็บข้อมูลภายในอย่างน้อย 128 ไบต์ไปจนถึง 256 ไบต์ หน่วยความจำสำหรับเก็บข้อมูลภายในบริเวณ 128 ไบต์แรกเรียกว่า Lower 128 และในบริเวณ 128 ไบต์หลังที่มีเพิ่มในบางเบอร์มีชื่อเรียกว่า Upper 128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำสำหรับเก็บข้อมูลภายในบริเวณ 128 ไบต์แรกและหลังจะมีวิธีการเข้าถึงข้อมูลในหน่วยความจำทั้งสองส่วนไม่เหมือนกัน



รูปที่ 4.3 ตำแหน่งหน่วยความจำของ MCS-51

4.6 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register : SFR)

ใน MCS-51 รีจิสเตอร์จะใช้หน่วยความจำ RAM ภายในชิพ โดยส่วนหนึ่งเป็นรีจิสเตอร์พิเศษต่างๆ จะเริ่มที่หน่วยความจำตั้งแต่ 80H ถึง FFH ซึ่งหน่วยความจำมีทั้งหมด 128 ตำแหน่ง แต่จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษเพียง 2 ตำแหน่ง แต่ถ้าเป็น 8032/8051 จะใช้ 26 ตำแหน่งหรือมี SFR 26 ตัว

สำหรับรายละเอียดเบื้องต้นของ SFR มีดังนี้

4.6.1 รีจิสเตอร์สถานะของโปรแกรม (Program Status Word) รีจิสเตอร์นี้เรียกว่าย่อๆ ว่า PSW จะอยู่ที่ตำแหน่ง D0H ซึ่งสามารถเข้าถึงข้อมูลระดับบิตได้ โดยรีจิสเตอร์ตัวนี้จะเป็นตัวบอกสถานะต่างๆของไมโครคอนโทรลเลอร์ MCS-51 แยกเป็นบิตคือ

1. แฟล็กตัวทค (Carry Flag) เป็นบิตที่ 7 ของ PSW บิตนี้จะมีความสำคัญหากมีการกระทำทางคณิตศาสตร์ โดยบิตนี้จะเซตเมื่อเกิดการทคของบิตที่ 7 ขณะทำการบวกเลขหรือเซตเมื่อเกิดการยืมของบิตที่ 7 เมื่อเกิดการลบเลข
2. แฟล็กตัวทคช่วย (Auxiliary Carry Flag) เป็นบิตที่ 6 เมื่อมีการบวกแบบ Binary Code Decimal (BCD) บิต AC หรือบิตตัวทคช่วยจะถูกเซตเมื่อมีการทคจากบิตที่ 3 ไปบิตที่ 4 หรือถ้าใน Low Nibble มีค่าระหว่าง 0AH-0FH เนื่องจากรหัส BCD มีค่าได้มากที่สุดแค่ 9 ถ้าหากมีการบวกเลขแบบ BCD จะต้องตามด้วยคำสั่ง DAA เพื่อปรับค่าที่เกิน 9 โดยการบวกเพิ่มด้วย 6 เข้าไป จะทำให้เป็นรหัส BCD ได้
3. แฟล็กศูนย์ เป็นบิตที่ 5 ของ PSW เป็นแฟล็กใช้งานทั่วไป เมื่อผู้เขียน โปรแกรม กำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะกระทำคำสั่งใดๆที่ บิตนี้จะไม่มีการเปลี่ยนแปลง
4. บิตเลือก رجิสเตอร์แบบจ (Register Select1) เป็นบิตที่ 4 และ 3 ทำให้เลือกใช้ رجิสเตอร์ R0-R7 ได้ 4 ชุด
5. แฟล็กโอเวอร์โฟลว บิตที่ 2 จะถูกเซตเมื่อกระทำทางคณิตศาสตร์แล้วเกิดโอเวอร์โฟลว คือจำนวนที่ได้จากการกระทำมีค่าเกินกว่าจำนวนไบต์ที่จะเป็น ได้คือมากกว่า 128 หรือน้อยกว่า -128
6. บิตพาริตี (Parity Bit) เป็นบิตที่บอกค่าพาริตีของ رجิสเตอร์ A โดยเป็นบิตที่ 0 ของ PSW

4.6.2 แอควิวมูลเตอร์ (Accumulator : ACC) มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็น رجิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ رجิสเตอร์นี้สามารถเข้าถึงได้ในระดับบิต

4.6.3 رجิสเตอร์ B อยู่ที่ตำแหน่ง F0H เป็น رجิสเตอร์ที่สามารถใช้งานได้ทั่วไปได้ แต่ส่วนใหญ่จะใช้ رجิสเตอร์นี้คูณหรือหารกับ رجิสเตอร์ A

4.6.4 สแต็กพอยน์เตอร์ (Stack Pointer : SP) เป็น رجิสเตอร์ขนาด 8 บิต อยู่ที่ตำแหน่ง 81H رجิสเตอร์ที่ชี้ข้อมูลหรือค่าพอยน์เตอร์ (Data Pointer : DPTR) เป็น رجิสเตอร์ที่ใช้สำหรับชี้ตำแหน่งรหัสโปรแกรมหรือข้อมูลในหน่วยความจำ โดยเป็น رجิสเตอร์ขนาด 16 บิต ซึ่งประกอบด้วย رجิสเตอร์ 2 ตัวคือ DPL อยู่ที่ตำแหน่งที่ 82H โดยจะเก็บเป็น 8 บิตต่ำและ DPH ตำแหน่งที่ 83H โดยจะเก็บค่า 8 บิตสูง

4.6.5 رجิสเตอร์พอร์ต (Port Register) มีขนาด 8 บิตที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มี 4 ตัวคือ رجิสเตอร์พอร์ต 0 หรือ P0 มีแอดเดรสอยู่ที่ 80H , رجิสเตอร์พอร์ต 1 หรือ P1 มีแอดเดรสอยู่ที่ 90H رجิสเตอร์พอร์ต 2 หรือ P2 มีแอดเดรสอยู่ที่ A0H และ رجิสเตอร์พอร์ต 3 หรือ P3 มีแอดเดรสอยู่ที่ B0H رجิสเตอร์ทุกตัวสามารถเข้าถึงได้ในระดับบิต เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการอ่านหรือเขียนข้อมูลออกไปยังพอร์ตของไมโครคอนโทรลเลอร์ จะต้องกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

4.6.6 รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF) รายละเอียดของรีจิสเตอร์นี้จะได้กล่าวในบทต่อไป

4.6.7 รีจิสเตอร์ไทมเมอร์ (Timer Register) ใน MCS-51 เบอร์ 8051 จะมีรีจิสเตอร์ที่ใช้นับเวลาขนาด 16 บิตอยู่ 2 ตัวคือ Timer 0 อยู่ที่ตำแหน่ง 8AH และ 8CH โดยหมายถึง TLO และ TH0 ตามลำดับ รีจิสเตอร์อีกตัวคือ Timer 1 ตำแหน่ง 8BH เป็น TL1 และ 8DH เป็น TH1 การใช้งาน Timer จะต้องกำหนดการทำงานในรีจิสเตอร์ TMOD ซึ่งอยู่ตำแหน่งที่ 88H ก่อน

4.6.8 รีจิสเตอร์ควบคุม (Control Register) แบ่งเป็น

1. รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงานของไมโครคอนโทรลเลอร์ MCS-51
2. รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของวงจรสื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51
3. รีจิสเตอร์ TCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทมเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51
4. รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทมเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51
5. รีจิสเตอร์ IE หรือ IP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการสนองการอินเทอร์รัปต์ โดย IE เป็นรีจิสเตอร์สำหรับเอ็นเอเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเทอร์รัปต์ ในขณะที่ IP เป็นรีจิสเตอร์สำหรับกำหนดลำดับความสำคัญของการตอบสนองการอินเทอร์รัปต์ว่า จะให้ซีพียูตอบสนองการเกิดอินเทอร์รัปต์ในลักษณะใดก่อนหรือหลัง

4.7 สเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์มีความแตกต่างจากมอเตอร์ทั่วไป โดยเมื่อป้อนกำลังไฟฟ้าให้กับมัน มันจะหมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งแตกต่างจากมอเตอร์ซึ่งแตกต่างจากมอเตอร์ทั่วไปซึ่งมันจะหมุนทันทีและตลอดเวลา สเต็ปเปอร์มอเตอร์สามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลขได้อย่างละเอียดโดยการใช้คอมพิวเตอร์เป็นตัวกำหนดและจัดเก็บตัวเลขเหล่านั้นไว้

สเต็ปเปอร์มอเตอร์ สามารถใช้งานในระบบเปิด (Open loop system) นั่นก็คือ มันทำงานได้โดยไม่ต้องการป้อนกลับ (feedback) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งได้อย่างถูกต้องจำเป็นต้อง

ที่ ต้องการการป้อนกลับไปยังระบบให้รับรู้
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกับมอเตอร์ทั่วไป การที่จะทำให้เกิดการหมุนของโรเตอร์ (rotor) ได้ต้องมีการกระทำของสนามแม่เหล็กที่เกิดขึ้นระหว่างโรเตอร์และสเตเตอร์ (stator) ซึ่งขึ้นอยู่กับการจัดวางขั้วแม่เหล็ก การหมุนทำได้ทั้งแบบต่อเนื่องและแบบกลับทิศทางไปมา โดยการจัดวางของแปรงถ่านและการจัดวางคอมมิวเตเตอร์ และทำการสวิตซ์ซึ่งกำลังไฟฟ้าให้เกิดแรงดึงดูดของแม่เหล็ก (Magnetic attraction) ที่ขั้วแม่เหล็ก ผลก็คือเกิดสนามแม่เหล็กหมุนสเตเตอร์โดยการจ่ายกำลังไฟฟ้าทีละขั้วแม่เหล็กในทิศทางตรงข้ามตลอดเวลา และเมื่อต้องการหยุดหมุนทำได้โดยหยุดการเกิดขั้วแม่เหล็กที่จุดหนึ่ง โดยทำการสวิตซ์ซึ่งในลำดับต่อไปเสีย การหมุนสลับทิศทางก็ทำได้เช่นเดียวกับที่กล่าวมา เพียงแต่ทำการสวิตซ์ซึ่งกำลังไฟฟ้าให้เกิดสนามแม่เหล็กหมุนในทิศทางกลับกัน โดยกลับลำดับการสวิตซ์ของมัน

4.8 ชนิดของสเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์แบ่งตามพื้นฐานได้เป็น 3 ชนิดคือ

1. ชนิดวาริเบิลรีลักแตนซ์ (Variable Reluctance : VR) สเต็ปเปอร์มอเตอร์ชนิดนี้มีจุดด้อยในเรื่องของความถูกต้องของตำแหน่งและทำงานได้ไม่ดีนักเมื่อมีสเต็ปในการหมุนสูง
2. ชนิดเพอร์มาเนนต์แมกเน็ต (Permanent magnet : PM) มีข้อดีในเรื่องของความถูกต้องของตำแหน่งและความเร็วมากขึ้นเมื่อเปรียบเทียบกับชนิดอื่น
3. ชนิดไฮบริด (Hybrid) เป็นชนิดที่นิยมใช้งานกันมากที่สุด โดยเฉพาะนำมาใช้กับอุปกรณ์ในเครื่องคอมพิวเตอร์

4.9 การกระตุ้นและควบคุมการหมุนของสเต็ปเปอร์มอเตอร์

การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนแบบซีแควนเชียลในรูปแบบที่ถูกต้องด้วยการกระตุ้นเฟสของขดลวดสเต็ปเปอร์มอเตอร์ มีอยู่ด้วยกัน 3 แบบคือ

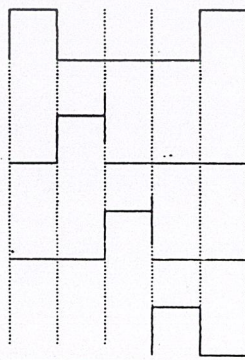
1. การกระตุ้นเฟสเดียว (Single Phase Excitation) เป็นการป้อนสัญญาณให้กับสเต็ปเปอร์มอเตอร์ ทีละขดดังแสดงในรูปที่ 4.4

PHASE 1

PHASE 2

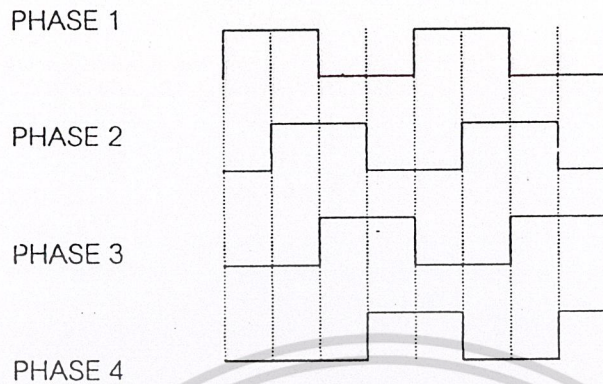
PHASE 3

PHASE 4



รูปที่ 4.4 แสดงการกระตุ้นแบบเฟสเดียว

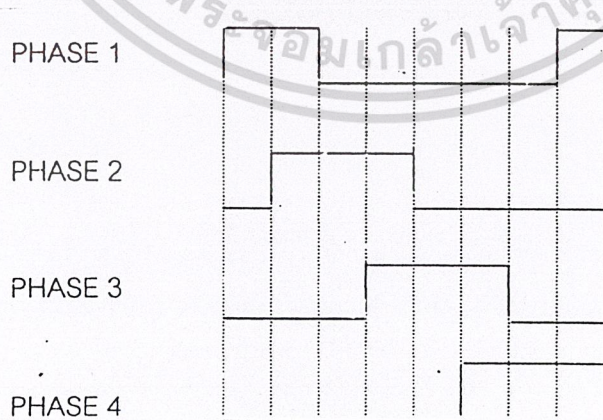
3. การกระตุ้น 2 เฟส (Two Phase Excitation) เป็นการป้อนสัญญาณให้กับขดลวดของ สเต็ปเปอร์มอเตอร์ ทีละ 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปดังรูปที่ 4.5



รูปที่ 4.5 แสดงการกระตุ้นแบบ 2 เฟส

การเพิ่มจำนวนของขดลวดที่ถูกกระตุ้นนี้ทำให้เพิ่มแรงบิดได้มากกว่าแบบเฟสเดียว โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และต่อไปด้วยแรงดึงจาก 2 ขดลวดถัดไป สำหรับข้อเสียก็คือ การกระตุ้นแบบนี้ต้องใช้แหล่งจ่ายกำลังไฟฟ้ามากขึ้น

3.การกระตุ้นแบบครึ่งสเต็ป (Half Step) เป็นรูปแบบที่เกิดจากการผสมผสานระหว่างการกระตุ้นแบบเฟสเดียวและ 2 เฟสเพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง แรงบิดที่ได้จากการกระตุ้นจะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลงและแต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้นการกระตุ้นแบบนี้ต้องทำการหมุนถึง 2 สเต็ปจึงจะได้เท่ากับ 1 สเต็ปเต็มเหมือนกับการกระตุ้น 2 แบบแรก สำหรับแหล่งจ่ายไฟใช้เทียบเท่ากับแบบ 2 เฟส



รูปที่ 4.6 แสดงการกระตุ้นแบบครึ่งสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ 1 ชุด (วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ต 3 คือขา P3.0 เป็นขารับข้อมูลเข้าหรือ RxD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TxD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชเป็นแบบอะซิงโครนัส ปกติแล้วพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 แต่ในปัจจุบันสามารถติดต่อกันในมาตรฐานของ RS-422 หรือ RS-485 ได้แล้ว โดยใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

5.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

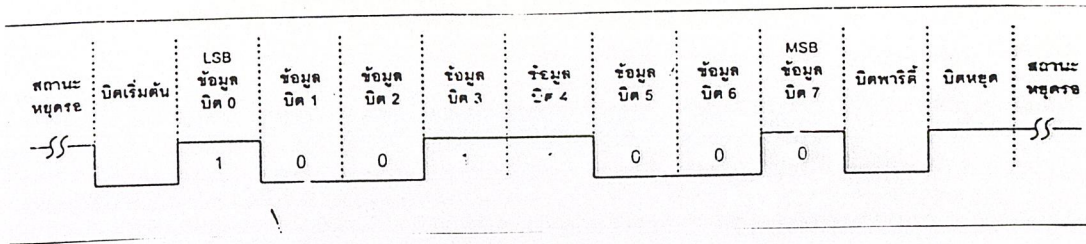
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลโดยไม่ต้องมีสัญญาณนาฬิกา ร่วมด้วย แต่จะใช้การกำหนดค่าอัตราความเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอด หรือ บอดเรต (Baud Rate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี (parity bit) มีขนาด 1 บิต หรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1 บิต

รูปที่ 5.1 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัสเมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น (start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อนซึ่งข้อมูลที่ต้องการส่งมีทั้งหมด 8 บิต จากนั้นตามด้วยบิตพาริตี (parity bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือ บิตปิดท้ายหรือบิตหยุด (stop bit) โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิตหรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อัตราความเร็วในการรับและส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่า ตั้งแต่ 110 ถึง 19,200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์



รูปที่ 5.1 รูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (odd) , แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตีก็ได้ พาริตีคี่หรือพาริตีคู่แสดงถึงจำนวนลอจิก "1" ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมพาริตีว่ามีจำนวนเป็นเลขคู่หรือเลขคี่

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter : เป็นอุปกรณ์ที่ใช้ในการรับส่งข้อมูลอนุกรม) ซึ่งทางภาครับต้องกำหนดการตรวจสอบพาริตีที่ตรงกันเอาไว้ว่าจะทำการตรวจสอบพาริตี คี่หรือคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดพาริตีไว้ว่าเป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้งานทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีเป็น NONE นั้น ทั้งภาครับและส่งจะไม่มีกรตรวจสอบพาริตี

5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังนี้

5.2.1 รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษหรือ SFR มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลและรับข้อมูล เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาจากขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

5.2.2 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial port control register) เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้ งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0-SM1 (Serial port mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2 : ใช้ในการเอ็นเอเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกติฟ ถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกติฟถ้ายังไม่ได้รับบิตหยุดส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception) : ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมใน MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8 : ใช้สำหรับรับข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานในโหมด 1 และบิต SM2 เป็น "0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด สำหรับในการทำงานโหมด 0 บิตนี้ไม่ใช้งาน บิต RB8 นี้สามารถเซตด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag) : ใช้ในการแสดงการเกิดอินเทอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receive Interrupt flag) : ใช้แสดงการเกิดอินเทอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรมสามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นกรณีที่บิต SM2 มีการเซต บิตนี้จะเซต ได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

5.3 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

พอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 สามารถเลือกการทำงานได้ถึง 4 โหมดคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะซีฟต์รีจิสเตอร์ การทำงานในโหมดนี้ของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการเชื่อมต่อกับไอซีรีจิสเตอร์ภายนอกเพื่อทำการขยายพอร์ตอินพุตหรือเอาต์พุต แต่ไม่นิยมใช้งานมากนัก เนื่องจากในไมโครคอนโทรลเลอร์ MCS-51 เองมีพอร์ตอยู่ค่อนข้างมาก และติดต่อกับพอร์ตเหล่านั้นได้ง่ายและเร็วกว่ามาก
2. กำหนดจากอัตราการเกิดโอเวอร์โฟลว(Overflow) ของไทเมอร์ 1 ใน AT89C51 ส่วนในไมโครคอนโทรลเลอร์เบอร์ AT98C52 และในอนุกรม AT89Sxx สามารถเลือกใช้อัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 หรือไทเมอร์ 2 ในการกำหนดอัตราบอดได้การทำงานในโหมดนี้ได้รับความนิยมสูงสุด เนื่องจากมีกระบวนการที่ไม่ซับซ้อนและสามารถทำการรับส่งข้อมูลกับคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้ การเลือกโหมดทำได้ด้วยการกำหนดข้อมูลให้แก่บิต SM0 และ SM1 ในรีจิสเตอร์ SCON

5.4 อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51

โหมด 0

อัตราบอดของโหมดนี้มีค่าคงที่ โดยสามารถคำนวณได้จากสูตร

อัตราบอดในโหมด 0 = ความถี่ของสัญญาณนาฬิกา/2 หน่วยเป็น บิตต่อวินาที

โหมด 1 และ 3

เนื่องจากทั้งสองโหมดนี้สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 แหล่งคือ จากอัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โฟลวของไทเมอร์ 1 จะต้องใช้ค่าบิต SMOD ในรีจิสเตอร์ PCON มาพิจารณาประกอบด้วย สามารถคำนวณค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = (2^{\text{ค่าของบิตSMOD}} / 32) \textcircled{4} \text{อัตราโอเวอร์โฟลวของไทเมอร์ 1}$$

ถ้าหากในไทเมอร์ 1 ไม่ได้เอ็นเอเบิลการอินเตอร์รัปต์ไว้ สามารถคำนวณค่าอัตราบอดได้จาก

$$\text{อัตราบอด} = (2^{\text{ค่าในรีจิสเตอร์ SMOD}} / 32) \textcircled{4} (\text{ความถี่สัญญาณนาฬิกา} / \{ 12 \times [256 - (\text{TH1})] \})$$

การกำหนดอัตราบอดโดยใช้ไทเมอร์ 1 แสดงในตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2

ในโหมดนี้อัตราบอดจะขึ้นอยู่กับค่าของบิต SMOD ในรีจิสเตอร์ PCON ถ้า SMOD เป็น "0" อัตราบอดจะเท่ากับ 1/64 ของสัญญาณนาฬิกา ในกรณีที่ SMOD เป็น "1" อัตราบอดจะเท่ากับ 1/32 ของความถี่สัญญาณนาฬิกา สามารถแสดงเป็นสูตรคำนวณทางคณิตศาสตร์ได้ดังนี้

$$\text{อัตราบอด} = (2^{\text{ค่าของบิต SMOD}} / 64) \times \text{ความถี่สัญญาณนาฬิกา}$$

อัตราบอด (บิตต่อวินาที : bps)	ความถี่ สัญญาณนาฬิกา	SMOD	โหมด 1		
			C/T	โหมด	ค่ารีโหลด
โหมด 0: สูงสุด 1 MHz	12 MHz	X	X	X	X
โหมด 2: สูงสุด 375 kHz	12 MHz	1	X	X	X
โหมด 1,3 : 62.5 kHz	11.0592 MHz	1	0	2	FFH
19.2k (19,200)	11.0592 MHz	1	0	2	FDH
9.6k (9,600)	11.0592 MHz	0	0	2	FDH
4.8k (4,800)	11.0592 MHz	0	0	2	FAH
2.4k (2,400)	11.0592 MHz	0	0	2	F4H
1.2k (1,200)	11.0592 MHz	0	0	2	E8H
137.5	11.0592 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 5.1 การเลือกอัตราบอดของวงจรถ่ายทอดอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51

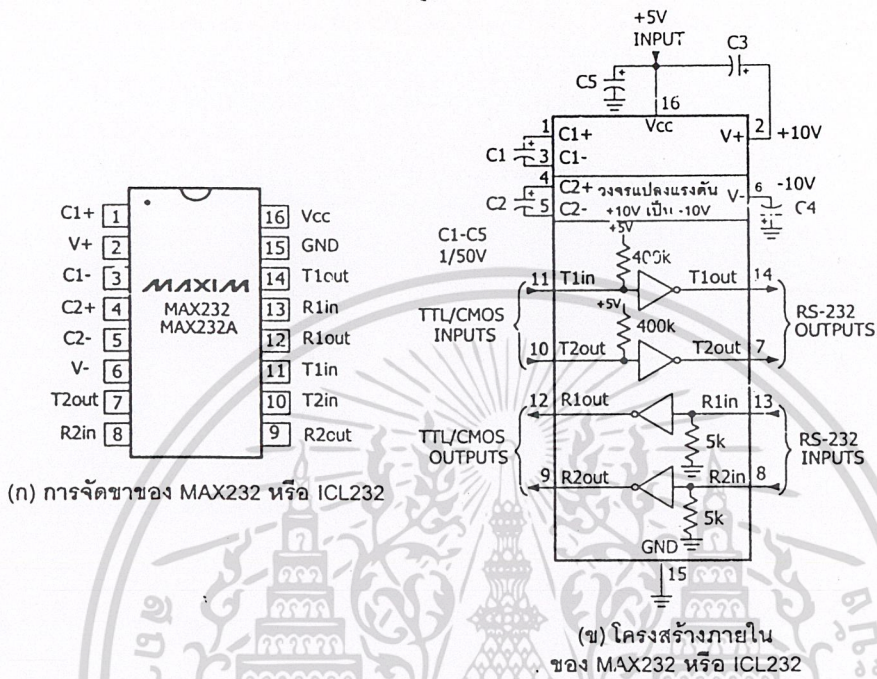
5.5 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ 3 ถึง 12 V ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับที่ทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ

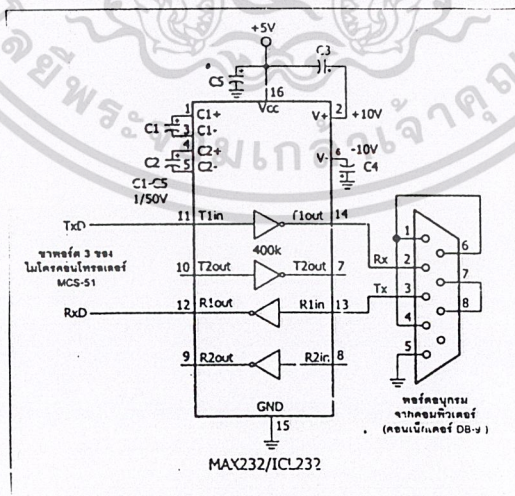
ไอซีพิเศษทำหน้าที่ในการแปลงระดับสัญญาณนี้ต้องทำการแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับที่ทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับที่ทีแอลเพื่อให้สามารถถ่ายทอดไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ MCS-51 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น ในรูปที่ 5.2 แสดงการจัดขาของไอซี ICL232 ซึ่งใช้ในการแปลงสัญญาณ RS-232 ส่วนวงจรของการต่อกับไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 5.3



รูปที่ 5.2 รายละเอียดเบื้องต้นของ ไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์



รูปที่ 5.3 วงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรมของคอมพิวเตอร์และการส่งข้อมูล ภาพและเสียงรวมทั้งคำสั่งควบคุมผ่านทางอินเทอร์เน็ต

ในการเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมนี้ มีหลายโปรแกรมที่สามารถนำมาใช้งานได้ ในโครงการนี้ได้เลือกใช้โปรแกรม Visual Basic เพราะเป็น โปรแกรมที่ออกแบบหน้าจอและใช้งานง่าย อีกทั้งยังมี Control MSComm เพื่อใช้ในการติดต่อผ่านพอร์ตอนุกรมอีกด้วย

6.1 คอนโทรล MSComm

สำหรับการใช้งาน Visual Basic ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual Basic จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยใน Visual Basic เวอร์ชัน 2 และ 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับระบบปฏิบัติการ 32 บิต สำหรับใน Visual Basic เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรม ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm หนึ่งตัวสามารถควบคุมการทำงานพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ตจะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขที่ Control Panel

6.2 คุณสมบัติของ MSComm

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (Property) มากมาย แต่สามารถทำความเข้าใจได้ไม่ยาก โดยคุณสมบัติที่สำคัญมีดังนี้

1. CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อยู่ (COM1,COM2,COM3,COM4)

รูปแบบการใช้งานคำสั่ง

```
object.CommPort[ = value ]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ต โดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียน โปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่จะใช้คำสั่ง Openport

2. Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด,พาริตี,จำนวนของบิตข้อมูล,จำนวนของบิตปิดท้าย
รูปแบบการใช้งานคำสั่ง

```
object.Settings[ = value ]
```

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น "BBBB,P,D,S" โดย BBBB เป็นค่าอัตราบอด ,P เป็นค่าพาริตี ,D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น "9600,N,8,1"

ตัวอย่างการใช้งานคำสั่ง Settings โดยจะกำหนดค่าอัตราบอดเท่ากับ 9600 ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียน โปรแกรมได้ดังนี้
MSComm1.Settings= "9600,N,8,1"

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด "" เนื่องจากค่าที่กำหนดนี้อยู่ในรูปแบบแปร String

3. PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรมเพื่อเปิดและปิดพอร์ตอนุกรม
รูปแบบการใช้งานคำสั่ง

```
object.PortOpen [ = value ]
```

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ portOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้ โปรแกรมก็จะแสดงข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัตินี้ของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิด โปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9600 บิตต่อวินาที ไม่มีพาริตีบิต จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต ดังนี้

```
MSComm.Settings = "9600,n,8,1"
```

```
MSComm.CommPort = 1
```

```
MSComm.Portopen = True
```

3. Input

ใช้ในการอ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

```
object.Input
```

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น ComInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น ComInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบไบนารีและชนิดข้อมูลเป็นแบบ Variant

ตัวอย่าง โปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command1_Click()
```

```
Dim InString as String
```

```
MSComm.InputLen = 0 ; Retrieve all available data
```

```
If MSComm1.InBufferCount then ; Check for data
```

```
InString = MSComm1.Input ; Read data
```

```
End If
```

```
End Sub
```

4. InBufferCount

ใช้ในการส่งค่าจำนวนตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้คำสั่ง

```
object.InBufferCount [ = value ]
```

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

เอกสาร 5. InBufferSize ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการกำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

```
object.InBufferSize[ = value ]
```

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

6. InputLen

ใช้ในการกำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

```
object.InputLen [= value]
```

ค่าเริ่มต้นของคุณสมบัติ InputLen มีค่าเท่ากับ "0" การกำหนดค่าเท่ากับ "0" จะทำให้คำสั่ง Input ของ MSComm อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่าง ("") กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ InBufferCount โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

7. InputMode

ใช้ในการกำหนดค่าและคืนค่าชนิดของข้อมูลที่ได้รับ โดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

```
object.InputMode [= value ]
```

คุณสมบัตินี้ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

comInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับข้อมูลก็จะเป็นค่านี้

comInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปแบบไบนารีรวมกันอยู่เป็นไบต์ข้อมูล

8. Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งานคำสั่ง

```
object.output [= value ]
```

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัพเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารีจะต้องกำหนดชนิดของข้อมูลเป็นแบบ variant และมีข้อมูลภายในเป็นแบบ Byte

ตัวอย่างโปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ด ไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ Output Private Sub Form_KeyPress (KeyAscii As Integer)

Dim Buffer as Variant

MSComm1.Commport = 1 ; Use COM1

MSComm1.PortOpen = True ; Open port

Buffer = Chr\$(KeyAscii)

MSComm1.output = Buffer ; Send Data

End Sub

9. OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัพเฟอร์ภาคส่ง และสามารถใช่คำสั่งนี้เพื่อเคลียร์บัพเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

object.OutBufferCount [= value]

ผู้ใช้งานสามารถเคลียร์บัพเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ "0"

10. OutBufferSize

ใช้ในการกำหนดค่าและคืนค่าขนาดของบัพเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

object.OutBufferSize [= value]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดของบัพเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

6.3 การเขียนโปรแกรมเพื่อควบคุมการเชื่อมต่อผ่านอินเทอร์เน็ต

ในการเขียนโปรแกรมผ่านอินเทอร์เน็ตเราใช้ control สำเร็จรูป ที่ได้ดาวน์โหลดมาจากอินเทอร์เน็ตมีชื่อว่า LEADTOOLS ซึ่งเป็นคอนโทรลที่สามารถทำการส่งภาพและเสียงผ่านอินเทอร์เน็ตได้ ซึ่งคอนโทรลที่มีอยู่ใน Visual Basic (winsock control) ไม่สามารถทำได้โดยตรง

ในที่นี้จะกล่าวถึง property , method , และ event ที่สำคัญที่ในการโปรแกรมเพียงส่วนหนึ่งเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.1 คอนโทรล Leadcap

เป็นคอนโทรลที่ใช้ในการจับไคร์เวอร์ของภาพและเสียงเพื่อแสดงออกทางจอคอมพิวเตอร์ รวมทั้งเป็นที่พักข้อมูลที่ส่งผ่านทางอินเทอร์เน็ตเพื่อแสดงออกทางด้าน client ด้วย

CapAudioFormat Property

ใช้สำหรับเซตหรือได้รับค่า รูปแบบของ Audio ที่จะทำการบันทึกเพื่อส่งผ่านอินเทอร์เน็ต ในที่นี้เราจะกำหนดรูปแบบเป็น FILE_WAVE_1M08 ซึ่งมี sampling rate 11.025 kHz, Mono sound, 8 bit ซึ่งรูปแบบนี้เป็นรูปแบบมาตรฐานอย่างหนึ่งของ PCM format ซึ่งได้รับการสนับสนุนจาก sound card ส่วนมาก

CapCloseRecord Method

ใช้สำหรับปิดการติดต่อกับอินพุต ดีไวซ์ เช่น sound card

CapConnect Method

ใช้สำหรับการเชื่อมต่อกับอินพุต ดีไวซ์ เช่น การติดต่อกับกล้อง เพื่อแสดงออกทางจอภาพ

CapDisconnect Method

ใช้สำหรับการปิดการเชื่อมต่อกับ อินพุตดีไวซ์ เช่น กล้อง Method นี้ไม่สามารถใช้ขณะมีการจับ single frame หรือ streaming ได้

CapDisplayMode Property

ใช้สำหรับกำหนดรูปแบบการแสดงผลภาพออกทางคอนโทรลในที่นี้มีอยู่ด้วยกัน 2 โหมด DISPLAY_PREVIEW ใช้สำหรับแสดงผลภาพออกทางคอนโทรลตลอดเวลาโดย windowsGDI จะเป็นผู้สร้างเฟรม โหมดนี้จะช้าแต่สามารถทำให้เกิด CapVideoData event ได้

DISPLAY_OVERLAY ตัว capture hardware จะเป็นผู้วาดเฟรมเอง เร็วกว่าแต่บางทีก็ไม่สนับสนุนโดยทุกๆดีไวซ์

CapDriver Event

จะเกิดขึ้นในแต่ละครั้งของ capture device ที่ install ลงในระบบเมื่อ CapEnumDrivers method ถูกเรียก ใช้ event นี้ในการทำ list รายชื่อของตัวอุปกรณ์ capture ที่เราสามารถใช้งานได้

CapEnumDrivers method

เป็น method ในการค้นหาชื่อของ capture devices ที่ install ในระบบ เป็น method ที่ใช้ร่วมกับ CapDriver Event

CapFeedSound method

ใช้สำหรับ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CapFeedSound method

ใช้สำหรับ

- feed sound data ไปยัง file เพื่อ save
- feed sound data เพื่อ play ของ sound device
- หรือสามารถทำได้ทั้ง 2 อย่างข้างต้น

ในโปรเจกต์นี้ใช้สำหรับรับข้อมูลเสียงจากserver แล้วทำการ feed sound data เพื่อ play

CapStartRecord method

ใช้เพื่อเริ่มต้นบันทึกเสียง CapOpenRecord ควรจะถูกเรียกก่อน เพื่อเตรียมกระบวนการพื้นฐานก่อน

CapStopCapture method

ใช้เพื่อหยุดการจับภาพ method นี้จะถูกต้องก็ต่อเมื่อมีการจับ streaming data เท่านั้น data capture จะหยุดก็ต่อเมื่อมีเหตุการณ์ต่อไปนี้นั้น

1. เปอร์เซ็นต์ของการ Drop frame มีค่าเกินกว่าที่ตั้งไว้ใน CapPercentDrop property.
2. ช่วงเวลาที่ระบุใน CapTimeLimit property ได้มาถึง
3. CapStopCapture method ถูกเรียก
4. ปุ่ม ESC ถูกกด
5. CapAbortLeftMouse ถูกเซตให้เป็นจริง และมีการกดคลิกเมาส์ซ้าย
6. CapAbortRightMouse ถูกเซตให้เป็นจริง และมีการกดคลิกเมาส์ขวา

CapStopFeedSound method

หยุด feed sound อาจจะหยุดหรือไม่หยุดก็ได้ขึ้นอยู่กับการเซตค่าว่าเป็น จริง หรือ เท็จ ถ้ามีการเซตให้เป็น จริง queue sound data ในbuffer ทั้งหมดจะ play แล้วจึงค่อยปิด device ถ้าเซตให้เป็น เท็จ จะทำการปิด device ก่อน โดยไม่สนใจ sound data buffer ที่มาถึงอีกต่อไป

CapStopRecord method

หยุดการบันทึกจาก input device วิธีนี้ไม่ได้เป็นการทำให้ข้อมูลเป็นอิสระจากกระบวนการบันทึก ดังนั้น CapStartRecord สามารถถูกเรียกได้อีกครั้งเพื่อบันทึกเสียงจาก input device ได้ต่อไป

CapVideoData event

เกิดขึ้นแบบต่อเนื่องถ้ามีการจับ streaming data อยู่ เหตุการณ์นี้มักจะเกิดในแต่ละเฟรมของ CapDisplayMode property ที่ถูกเซตค่าไว้เป็นแบบ โหมด DISPLAY_PREVIEW

เหตุการณ์นี้เกิดก็ต่อเมื่อถ้ามีการเซต EnableCaptureEvents property เป็น จริง เท่านั้น

EnableCaptureEvents property

อนุญาต หรือไม่อนุญาต ในการกำเนิด เหตุการณ์ capture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EnableMethodErrors property

ระบุว่าจะให้มีการเกิด LEADTOOLS runtimes exception หรือไม่

6.3.2 คอนโทรล Inet

ConnectList property

รายชื่อคอมพิวเตอร์ที่ได้รับการติดต่อเข้ามา การติดต่อนี้เป็นผลลัพธ์มาจาก InetConnect หรือ InetAcceptConnect.

ใช้propertyนี้ในการเก็บรายชื่อการติดต่อ ตัวจัดการ computer จะเพิ่มหรือลบอัตโนมัติจาก array ขณะที่การติดต่อได้สร้างขึ้นหรือปิดลง

ConnectListNum property

ใช้สำหรับรับจำนวน items ใน ConnectList array.

InetAccept event

เกิดขึ้นเมื่อมีคอมพิวเตอร์อย่างน้อย 1 เครื่องติดต่อเข้ามาที่ local server

เหตุการณ์นี้เกิดเป็นผลตอบสนองจาก remote computer ที่มาrequest การติดต่อ เพื่อติดต่อกับ remote computer การใช้ control inet ให้ใช้ InetConnect method.

InetAcceptConnect method ควรจะถูกเรียกตอบสนองของเหตุการณ์นี้เป็นลำดับขั้นของการติดต่อ

InetAcceptConnect method

ยอมรับการติดต่อที่ร้องขอเข้ามาจาก remote computer บน server ที่ระบุ เราควรเรียก method นี้ใน InetAccept event. เท่านั้น

InetClearQueue method

เราสามารถใช้ฟังก์ชันนี้ในการออกจากการรับ-ส่งข้อมูล ฟังก์ชันนี้ควรจะใช้ด้วยความระมัดระวังเพราะที่ท้ายสุดของการติดต่อจะไม่มีทางรู้ว่าqueue นั้นถูกเคลียร์

InetConnect method

ใช้สำหรับติดต่อกับ remote computer บน port ที่ระบุ วิธีนี้จะทำให้เกิด InetAccept event บน Host computer ถ้า server นั้นได้ทำการเปิดไว้

InetConnected event

เหตุการณ์นี้จะเกิดขึ้นเมื่อ local computer ประสบความสำเร็จกับ remote computer หนึ่งการติดต่อสามารถทำได้ 2 วิธี

1. ตั้งทางlocal computer เสมือน เป็น Server และรอฟังการติดต่อ โดยใช้ InetServerInit method เมื่อมีการร้องขอเข้ามา ทำการเปิดการติดต่อโดยใช้ InetAcceptConnect

เอกสารนี้เป็นmethodที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ติดต่อไปยัง server ที่มีอยู่แล้ว โดยใช้ InetConnect method

InetDataSent event

เหตุการณ์นี้จะเกิดเมื่อข้อมูลทั้งหมดสำหรับ object ที่ระบุ เช่น bitmap ได้รับการส่งไปยัง remote computer ที่ได้ออกขอเข้ามา

InetDisconnect method

ปิดการติดต่อ ใช้ฟังก์ชันนี้ในการ ยกเลิกการติดต่อไปยัง remote computer

InetDisconnected event

จะเกิดขึ้นเมื่อ local computer ได้ยกเลิกการติดต่อ จาก remote computer

InetGetHostName method

ฟังก์ชันนี้จะทำการรับชื่อของ host name ของ computer ที่ถูกติดต่อที่ปลายอีกด้านหนึ่ง ถ้าชื่อมีการผิดพลาด วิธีจะทำให้เกิด Error

InetQueueSize property

คืนค่าขนาดของ queue size ภายในที่ใช้โดย computer handle ถ้า property นี้คืนค่า 0 สำหรับนั้นหมายถึงไม่มี ข้อมูล อยู่ใน queue ถ้าคืนค่า ระหว่าง 1 และ 0xFFFFFFFF แสดงว่านั่นคือจำนวน byte ที่อยู่ใน queue

InetReceiveBitmap event

เกิดเมื่อมีไฟล์ภาพเข้ามาในการติดต่อ เหตุการณ์นี้จะเกิดเป็นผลตอบสนองจาก remote computer ที่ได้ทำการเรียก InetSendBitmap method หรือ InetSendDib method

InetReceiveData event

เกิดเมื่อข้อมูลได้รับครบถ้วน โดยสมบูรณ์แล้วขณะที่มีการติดต่อ เหตุการณ์นี้จะเกิดเป็นผลตอบรับจาก remote computer ที่ได้ทำการเรียก InetSendData method

เหตุการณ์นี้จะเกิดหลังจากข้อมูลตัวสุดท้ายได้รับและ InetReceiving event เกิดไปเรียบร้อยแล้ว หลังจากข้อมูลตัวสุดท้ายที่ถูกส่ง

InetReceiveSound event

เหตุการณ์นี้จะ เกิดขึ้นมา เมื่อ มี wave data เข้ามาที่ตัวคอมพิวเตอร์ มันเกิดเป็นผลตอบสนองจาก remote computer ที่ได้ทำการเรียก InetSendSound method พารามิเตอร์ต่อไปนี้จะถูก update สำหรับ sound format iFormatTag, iChannels, iSamplesPerSec, AvgBytesPerSec, iBlockAlign, iBitsPerSample, iExtraDataSize, pExtraData

InetSendBitmap method

เป็นวิธีการส่ง ไฟล์ภาพไปยัง remote computer ที่อยู่ใน sendlist และจะทำให้เกิดเหตุการณ์ InetReceiveBitmap event ที่ remote computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bitmap จะถูก เซฟ เสมือน ไฟล์ ในหน่วยความจำในรูปแบบที่กำหนด ถ้าเราต้องการ additional option เมื่อมีการเซฟไฟล์ (เช่น Progressive JPEG , Interlaced GIF , หรือ อื่น ๆ) ต่อไป เราควรจะใช้ วิธี SaveMemory method ใน Main Control แล้วทำการส่ง โดยใช้ InetSendData ด้วย iType ที่เป็น IDATA_IMAGE สิ่งนี้หมายถึงเราสามารถส่งไฟล์ในรูปแบบของ supported compressed หรือ uncompressed formats ใดๆก็ได้

InetSendData method

เป็นวิธีการส่งจำนวนbyte ของข้อมูลไปยัง remote computer ที่อยู่ใน SendList และเกิด InetReceiveData event บน remote computer ด้วย

InetSendSound method

ส่ง wave data ไปยัง remote computer ที่อยู่ใน SendList property วิธีนี้จะทำให้เกิด InetReceiveSound event บน remote computer

InetServerClose method

ทำการปิด server บน local computer ใช้วิธีนี้ในการปิด server ที่ได้เริ่มไว้ด้วยวิธี

InetServerInit method

ในแต่ละครั้งที่วิธีนี้เสร็จสมบูรณ์ remote computer จะไม่สามารถทำการติดต่อมาได้จนกระทั่ง InetServerInit จะถูกเรียกเสร็จสิ้นเพื่อที่จะเริ่มต้น server ใหม่

วิธีนี้ไม่ได้ทำการปิดการติดต่อสื่อสารแบบเปิด แต่มันเพียงแค่หยุดการร้องขอติดต่อครั้งใหม่จากที่ได้รับ

InetServerInit method

เป็นการเริ่มต้นให้ computer เป็น server ใช้วิธีนี้เพื่ออนุญาตให้computer ทำการconnect และรับส่งข้อมูล

SendList property

รายชื่อของ ตัวจัดการ computer ที่จะถูกส่ง ใช้propertyนี้เพื่อสร้างรายชื่อของ remote computer ที่เราหวังจะทำการส่งข้อมูล

การลบรายชื่อ computer ออกจาก Sendlist ให้ทำการเซตค่า index เป็น 0 วิธีนี้จะเป็นการ update รายชื่อและ SendListNum property

เมื่อการติดต่อได้ถูกลบออกจาก ConnectList property ในการตอบสนอง จะทำการเรียก InetDisconnected event , ซึ่งสอดคล้องกับตัวจัดการ computer ที่จะถูก ลบ ออกแบบอัตโนมัติจาก SendList array

SendListNum property

ได้รับค่าของ items ที่อยู่ใน SendList array

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกำไรค่า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังมีคอนโทรลที่สำคัญอีกตัวหนึ่งที่ใช้ในการรับส่งภาพที่อยู่ทางด้าน Client คือ control LEADMAIN นั่นคือเป็น คอนโทรลที่ใช้ในการจัดการกับไฟล์รูปภาพ เช่น ใช้สำหรับรับภาพที่ได้มาจากทางฝั่ง Server มาแสดงออกทางจอ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

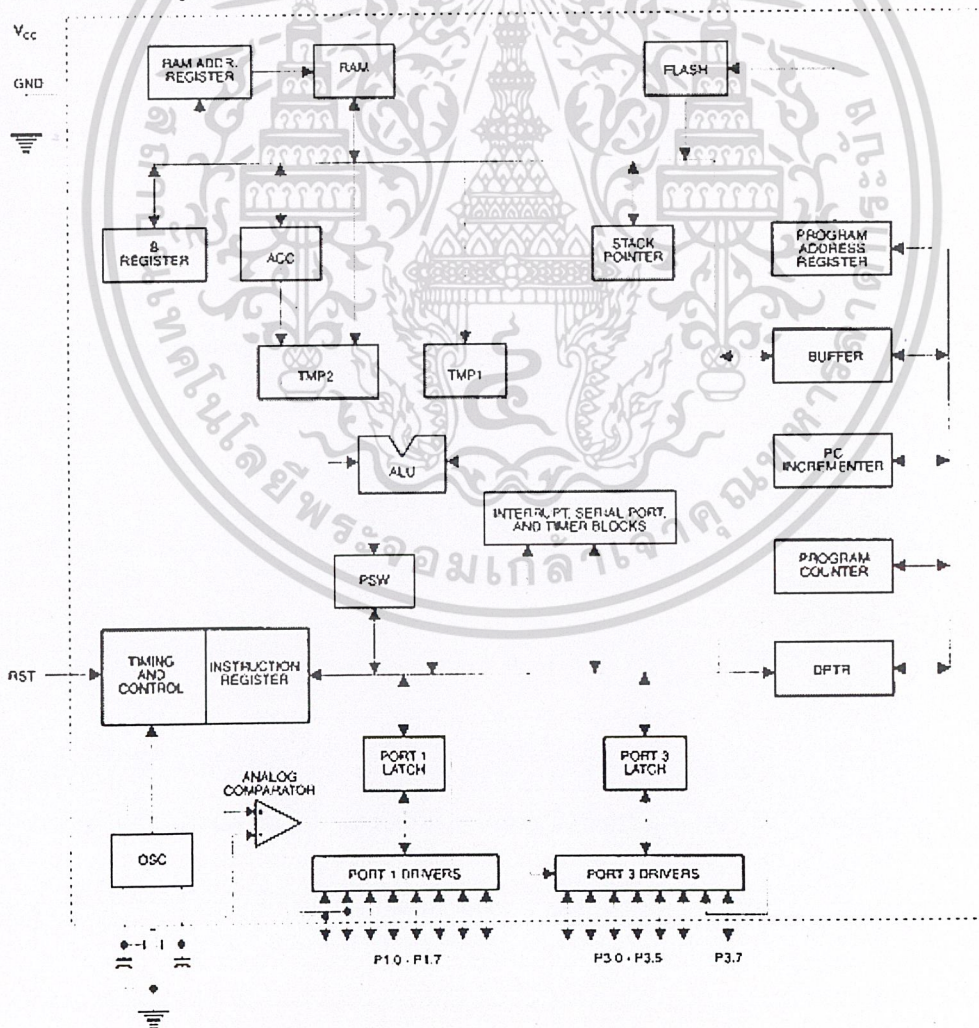
หลักการออกแบบและการสร้าง

การออกแบบและการสร้างจะประกอบด้วย ชุดควบคุมกล้อง และส่วนของโปรแกรม Visual Basic สำหรับใช้ในส่วนควบคุมกล้องและส่งภาพและเสียง จะได้อธิบายดังต่อไปนี้

7.1 ชุดควบคุมกล้อง (Camera Control) โดยจะแบ่งเป็น 2 ส่วนคือ

7.1.1 ส่วนฐานจับกล้อง

7.1.2 ส่วนไมโครคอนโทรลเลอร์และวงจรขั้วมอเตอร์ โดยจะมีอยู่ 2 ส่วนด้วยกันคือ ส่วนที่เป็นไมโครคอนโทรลเลอร์ซึ่งโครงงานนี้ได้เลือกใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 เบอร์ AT89C2051 ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาดเล็ก ใช้งานง่าย โครงสร้างภายในของ AT89C2051 จะคล้ายกับโครงสร้างหลักของ MCS-51 แต่จะลดจำนวนพอร์ตลง ทำให้เป็นไมโครคอนโทรลเลอร์ขนาดเล็กขนาด 20 ขาดังรูปที่ 7.1

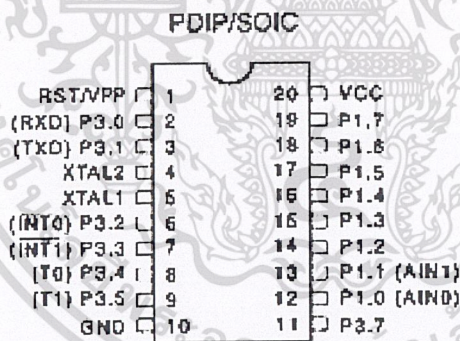


รูปที่ 7.1 แสดงโครงสร้างภายในของ AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของ AT89C2051

- มีโครงสร้างและชุดคำสั่งเหมือนกับไมโครคอนโทรลเลอร์ในตระกูล MCS-51
- มีหน่วยความจำโปรแกรมชนิด Flash Memory ขนาด 2kByte สามารถโปรแกรมซ้ำได้ 1000 ครั้ง
- มีหน่วยความจำแบบ RAM 8 bit ขนาด 128 Byte (Internal Ram)
- ทำงานที่แรงดัน 2.7-6 โวลต์
- Run ที่ความเร็วนาฬิกา 0Hz-24MHz
- มีพอร์ต I/O 15 บิต (พอร์ต 1 และ 3)
- พอร์ตสามารถ Sink กระแสได้ 20 mA และใช้เป็นเอาต์พุตขับ LED ได้โดยตรง
- มีพอร์ตสื่อสารแบบอนุกรม 1 Channel (UART)
- สามารถโปรแกรมข้อมูลเพื่อป้องกันการอ่าน-เขียนได้ 2 ระดับ
- มีวงจรถ่ายและวงจรมับขนาด 16 บิต จำนวน 2 Channel
- มีสัญญาณอินเทอร์รัปต์ 5 แหล่ง แบ่งระดับความสำคัญได้ 2 ระดับ
- มีวงจรถ่ายเทียบสัญญาณอะนาล็อก 1 Channel
- มีระบบประหยัดพลังงาน



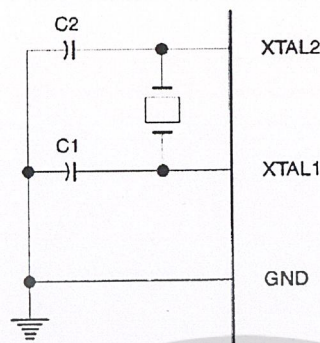
รูปที่ 7.2 แสดงตำแหน่งขาต่างๆ ของ AT89C2051

การออกแบบวงจรส่วนของ AT89C2051

AT89C2051 มี I/O พอร์ตทั้งหมด 15 บิต คือ พอร์ต 1 (P1.0-P1.7) และพอร์ต 3 (P3.0-P3.5 และ P3.7) แต่ในโครงงานนี้จะใช้เพียง 10 บิต คือ ใช้พอร์ต 1 (P1.0-P1.7) เป็นเอาต์พุตพอร์ตสำหรับการควบคุมมอเตอร์ 2 ตัว และ P3.2 กับ P3.3 สำหรับต่อกับตัวเซนเซอร์ 2 ตัว จากรูปที่ 5.1 จะเห็นว่า P1.0 และ P1.1 จะใช้เป็นขาาร่วมของอินพุตของวงจรถ่ายคอมพาราเตอร์ (Comparator) ด้วยจึงไม่มี Internal Pull-Up ทำให้เมื่อเรานำ P1.0, P1.1 มาใช้เป็นเอาต์พุตจะต้องทำการเพิ่ม Pull-Up ภายนอกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการต่อ XTAL ให้กับ AT89C2051 เราจะใช้ XTAL ค่า 11.059 MHz เป็นตัวกำหนดสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์



รูปที่ 7.3 แสดงการต่อคริสตัลให้กับ AT89C2051

C1 และ C2 ในรูปที่ 7.3 จะใช้ค่าในช่วง 20 pF-40pF สำหรับการใส่คริสตัล และ 30 pF – 50pF สำหรับการใส่ Ceramic Resonators

ส่วนพอร์ต 3 นอกจากจะเป็น I/O พอร์ตแล้วยังทำงานในหน้าที่อื่นด้วย ดังตารางที่ 7.1

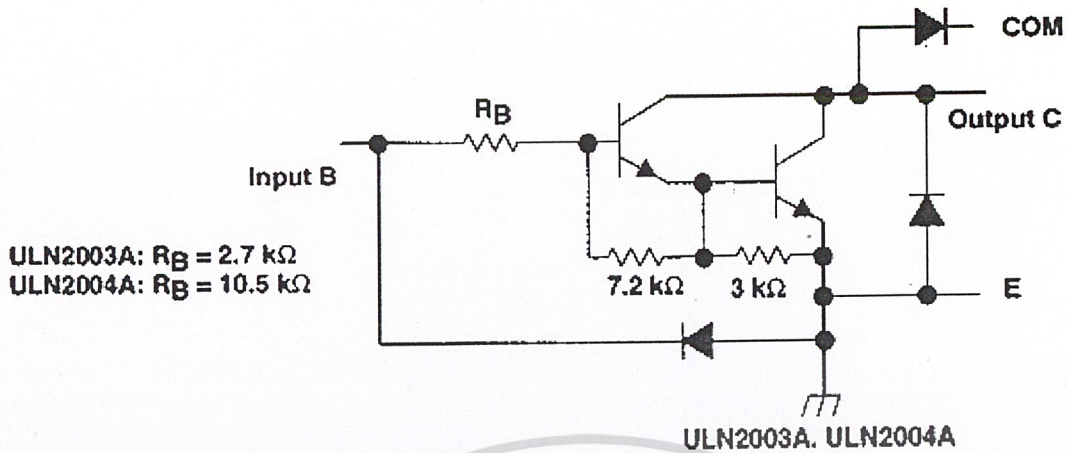
Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

ตารางที่ 7.1 แสดงหน้าที่พิเศษของพอร์ต 3

ซึ่งในโครงการนี้จะใช้หน้าที่พิเศษของพอร์ต 3 นี้ด้วยคือ P3.0 และ P3.1 โดยจะนำมาเป็นส่วนของการส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ในลักษณะการสื่อสารแบบอนุกรมตามแบบ RS-232 ซึ่งจะต้องเพิ่มส่วนของ Line Driver เพื่อให้แรงดันในการสื่อสารเป็นไปตาม RS-232 โดยได้เลือก IC เบอร์ MAX232 ในการทำหน้าที่นี้ ส่วน P3.2 และ P3.3 ใช้ต่อกับตัวเซนเซอร์

ส่วนที่สองคือ ส่วนของวงจรถับมอเตอร์ เนื่องจากมอเตอร์ที่ใช้เป็นแบบสตีปเปอร์มอเตอร์สามารถควบคุมได้โดยการป้อนพัลส์ (Pulse) ให้แต่ละเฟสของมอเตอร์ แต่เอาต์พุตของพอร์ตไม่สามารถที่จะไปจับมอเตอร์ได้โดยตรงและเพื่อเป็นการป้องกันความเสียหายของตัวไมโครคอนโทรลเลอร์จึงจำเป็นต้องมีชุดขับกระแสต่อระหว่างพอร์ตของไมโครคอนโทรลเลอร์กับมอเตอร์

ในโครงการนี้ได้ใช้ IC ULN2003A เป็นตัวขับกระแสและแรงดันให้กับมอเตอร์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



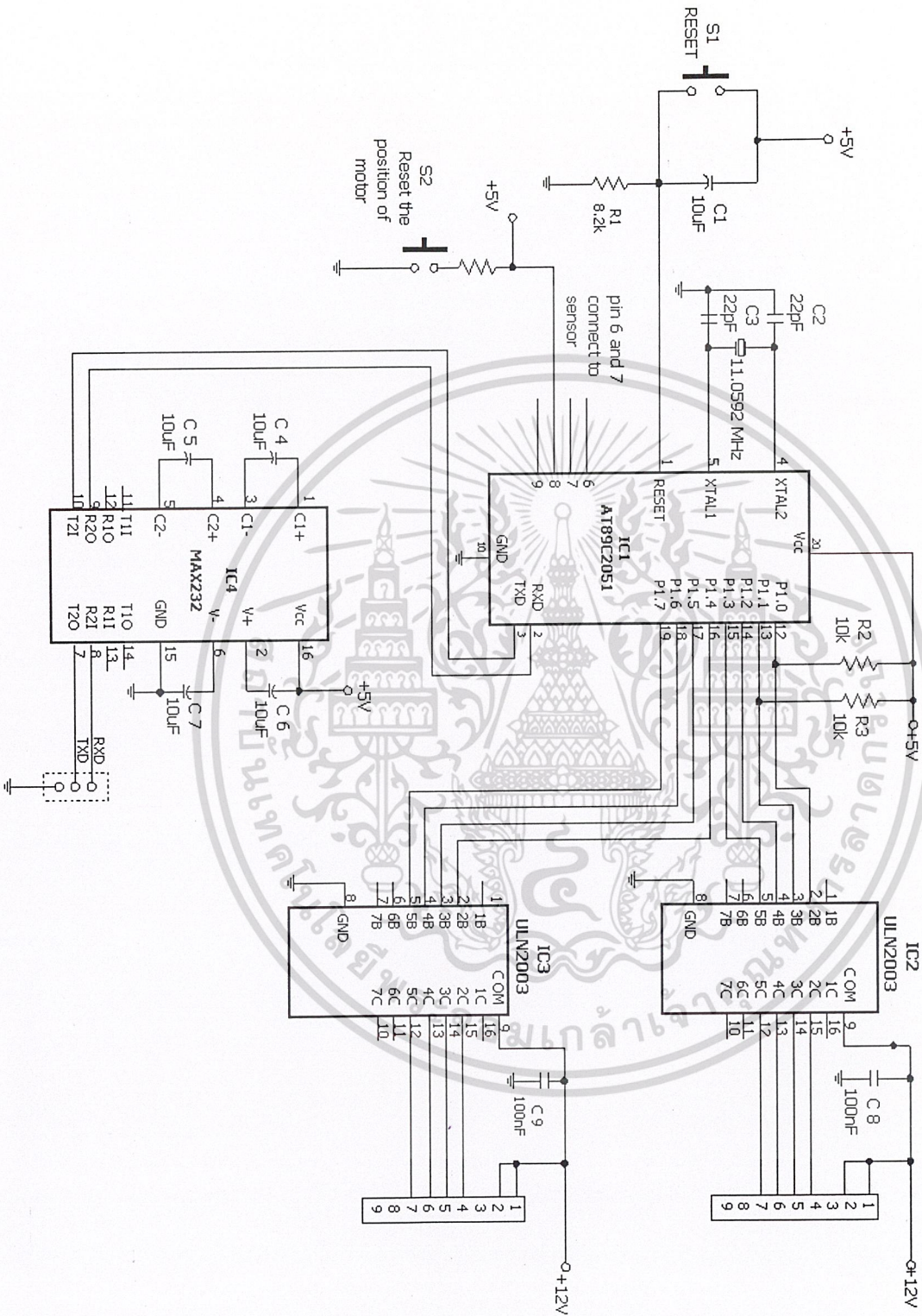
รูปที่ 7.4 แสดงรายละเอียดของ ULN2003A

ULN2003A หนึ่งตัวสามารถใช้เป็นวงจรขับได้ 7 ตัว ทำให้เราต้องใช้ 2 ตัวในการขับวงจร 8 บิต โดยแบ่งเป็นตัวละ 4 บิตบน-ล่าง ทำให้แยกขับออกเป็นมอเตอร์แต่ละตัวได้โดยไม่จำเป็นต้องต่ออุปกรณ์เพิ่มจากภายนอกเลย

คุณสมบัติที่สำคัญของ ULN2003A คือ

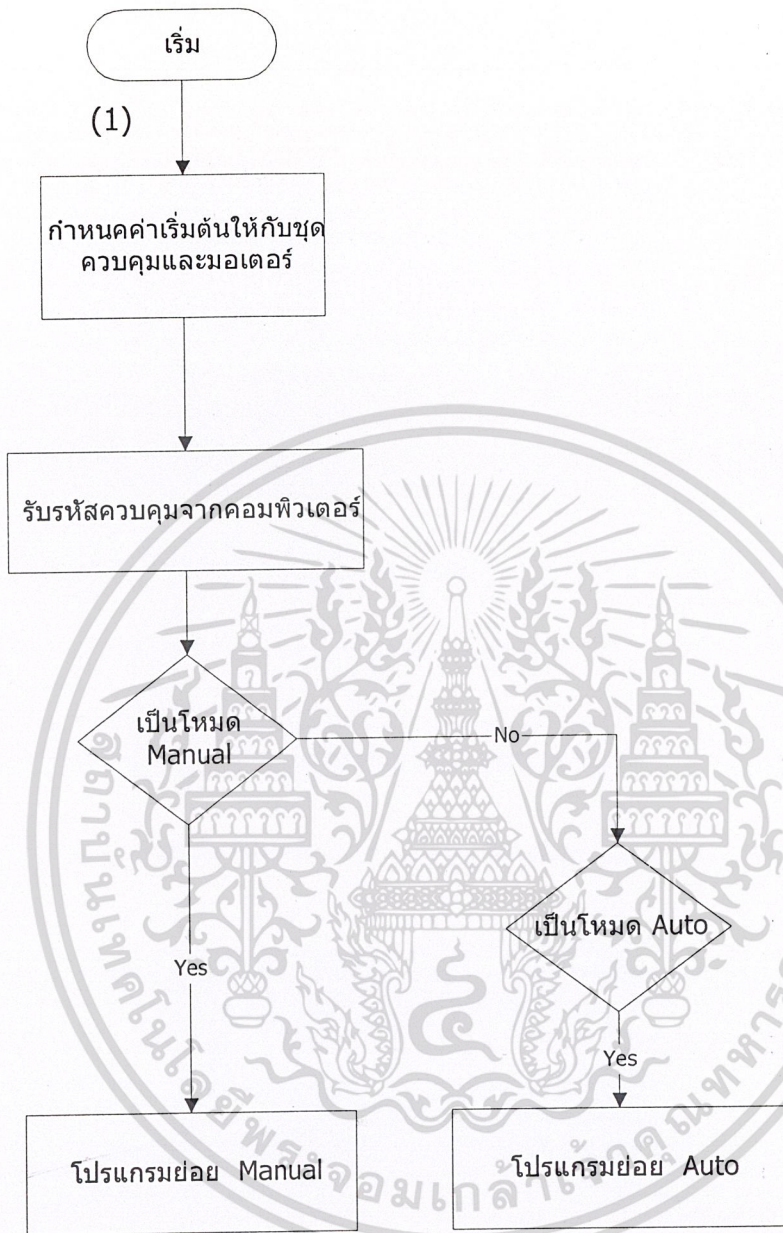
- มี Output Voltage สูงสุดได้ถึง 50 V
- จ่ายกระแส Output สูงสุดได้ถึง 500 mA

ในรูปที่ 7.5 ได้แสดงวงจรของ AT89C2051 และส่วนวงจรขับมอเตอร์ที่ใช้ในโครงการนี้ สำหรับโปรแกรมสำหรับควบคุมการทำงานของสเต็ปเปอร์มอเตอร์ทั้งสองตัวนี้ ใช้ภาษา Assembly เก็บไว้ใน Flash Memory ของ AT89C2051 ที่มีรายละเอียดการทำงานตาม Flowchart ดังที่แสดงไว้ในรูปที่ 7.6-7.9 โดยให้การจ่ายไฟเพื่อขับมอเตอร์เป็นแบบครึ่งสเต็ป



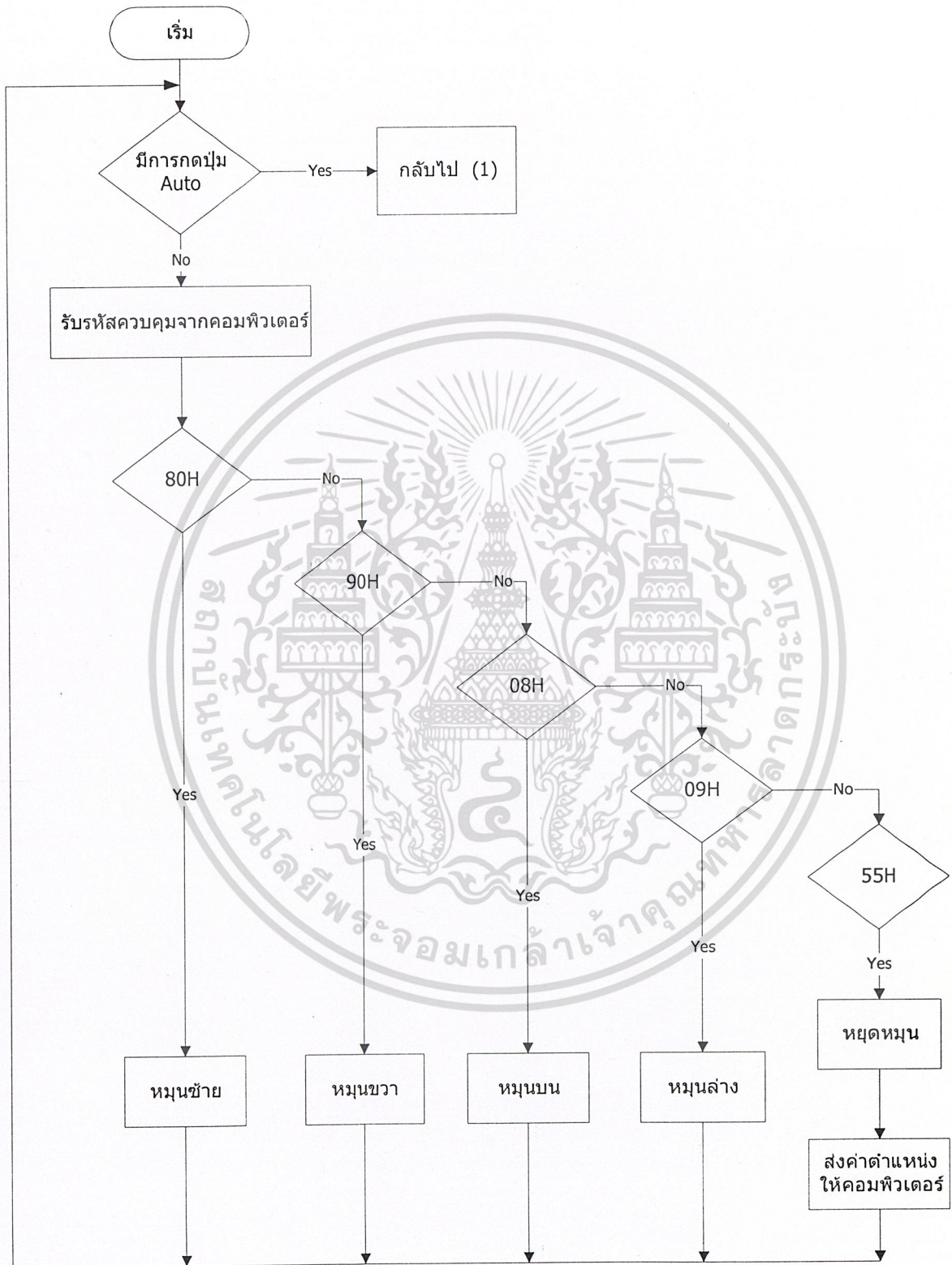
รูปที่ 7.5 แสดงวงจรการเชื่อมต่อของ MCS-51 และวงจรการขยับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



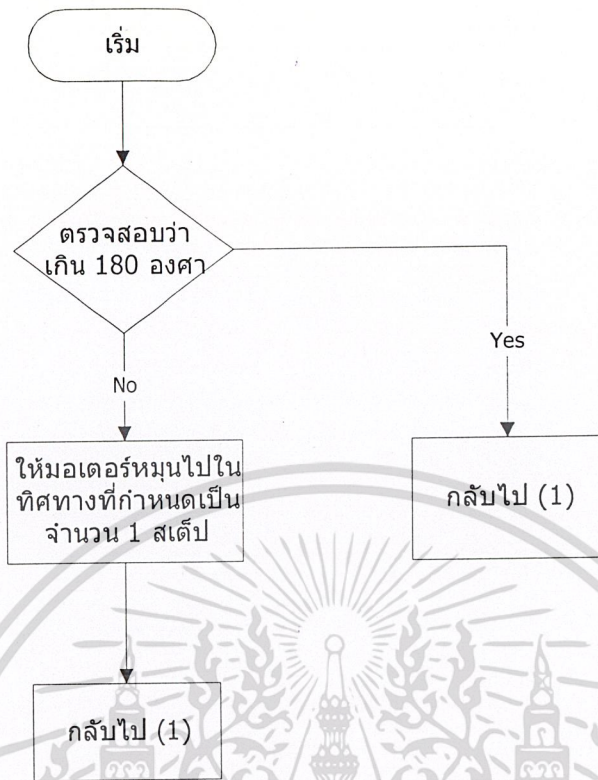
รูปที่ 7.6 แสดง Flowchart โปรแกรมหลักที่อยู่ในชุดควบคุมกึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



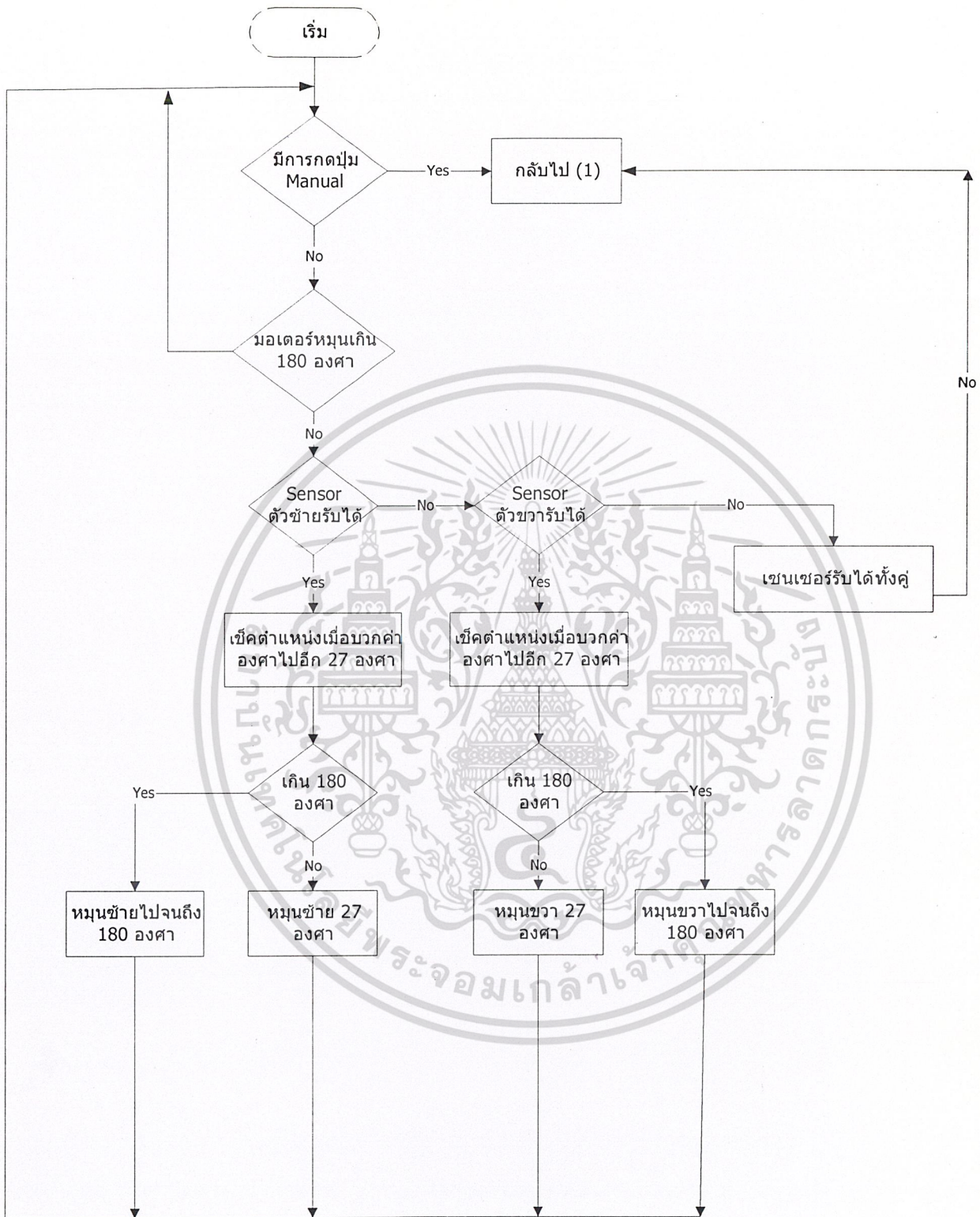
รูปที่ 7.7 แสดง Flowchart การทำงานของโปรแกรมโหมด Manual ที่อยู่ในชุดควบคุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.8 แสดง Flowchart โปรแกรมย่อยของโปรแกรมหมุนมอเตอร์โหมด Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.9 แสดง Flowchart โปรแกรมย่อยของ โปรแกรมหมุนมอเตอร์โหมค Auto

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 การโปรแกรมด้วย Visual Basic

ในที่นี้จะแบ่งออกเป็น 2 ส่วน คือ การโปรแกรมส่งคำสั่งควบคุมผ่านพอร์ตอนุกรม และการโปรแกรมส่งภาพและเสียงผ่านอินเทอร์เน็ต สำหรับสาเหตุที่ใช้ Visual Basic ในการโปรแกรมนั้น เพราะเป็นโปรแกรมที่สามารถใช้งานได้ง่าย มีรูปแบบ interface ที่สามารถทำความเข้าใจได้รวดเร็ว ซึ่งตรงกับความต้องการในการเขียนโปรแกรม

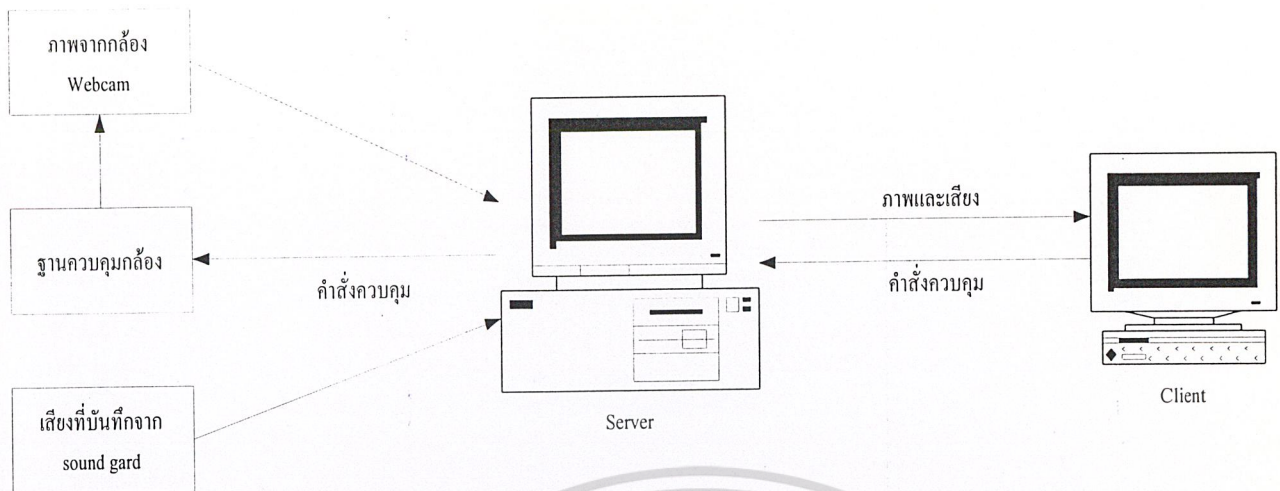
7.2.1 การโปรแกรมส่งคำสั่งควบคุมผ่านพอร์ตอนุกรม

ในการโปรแกรมส่วนนี้เราได้ใช้ control ที่มีอยู่แล้วใน VB นั่นคือ คอนโทรล Microsoft Mscomm 6.0 ซึ่งเป็นคอนโทรลที่ใช้สำหรับทำการเชื่อมต่อคอมพิวเตอร์กับพอร์ตอนุกรม RS-232 ซึ่งรายละเอียดการใช้งาน property , method , event ได้กล่าวถึงในส่วนของทฤษฎีไปบ้างแล้ว การโปรแกรมในส่วนนี้จะอยู่ในส่วนของ server เพื่อทำการควบคุมกล้อง

7.2.2 การโปรแกรมส่งภาพและเสียงผ่านอินเทอร์เน็ต

ในการโปรแกรมส่วนนี้ได้ใช้ Active X control ที่ได้ทำการดาวน์โหลดมาจากอินเทอร์เน็ต มีชื่อว่า LEADTOOLS ซึ่งเป็น component ที่เราใช้ในการส่ง-รับภาพและเสียงได้แทนการใช้ component ที่มีใน VB คือ Microsoft winsock control 6.0 ทั้งนี้เพราะ ตัว component ของ VB เองนั้นไม่สามารถทำการส่งภาพและเสียงได้โดยตรง เพราะข้อมูลภาพและเสียงประกอบไปด้วยตัวแปรที่ winsock ไม่สามารถทำการส่งได้โดยตรง ซึ่งถ้าจะใช้ตัว winsock ทำการส่งภาพและเสียงนั้นเราจำเป็นต้องรู้ตัวแปรต่าง ๆ ในการจัดการกับไฟล์รูปภาพและไฟล์เสียง ซึ่งถือว่าเป็นเรื่องที่ยากลำบากเพราะต้องเกี่ยวข้องกับการใช้ API ของระบบปฏิบัติการของ windows และยังคงจำเป็นต้องใช้ความรู้ในเรื่องของ ภาษา C ในการติดต่อกับไดร์เวอร์อีกด้วย ดังนั้นเพื่อความสะดวก เราจึงได้ทำการ ใช้ Active x control ดังกล่าว

การโปรแกรมในส่วนนี้จะกระทำที่ทั้งทางด้าน Client และ Server โดยที่ด้าน Server นั้นเราจะใช้ Active x control ในการจับภาพ และเสียง เพื่อแสดงภาพและบันทึกเสียงเพื่อทำการส่งผ่านเครือข่าย อินเทอร์เน็ต ส่วนที่ทางด้าน Client ก็จะใช้ control เดียวกันในการรับข้อมูลภาพ และ เสียง มาเพื่อแสดงและ เล่นเสียงที่รับได้จากทางด้าน Server โดยมีการทำงานโดยรวมของระบบเป็นดังนี้

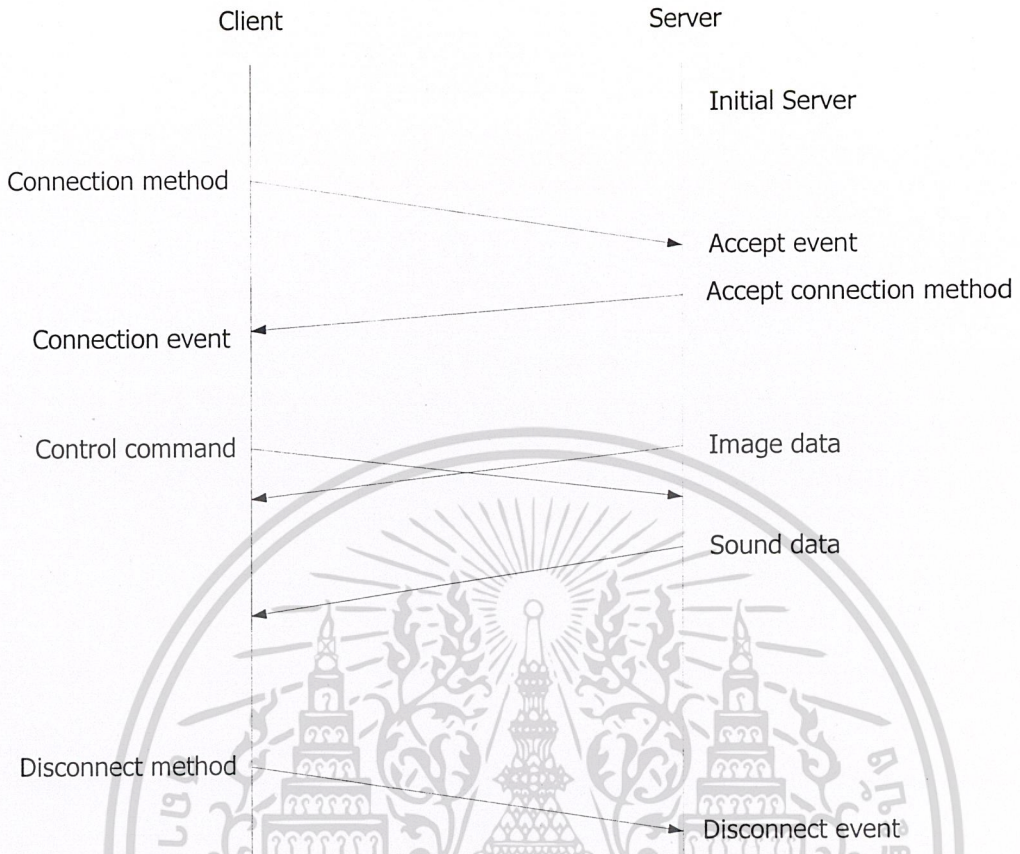


รูปที่ 7.10 แสดงโครงสร้างการทำงานของระบบ

สำหรับขั้นตอนการทำงานในการส่งภาพและเสียงผ่านอินเทอร์เน็ตนั้น ก่อนการเชื่อมต่อจะต้องมีการกำหนดสถานะให้ฝั่งทางด้าน Server พร้อมรับในการติดต่อก่อนนั้นคือการกำหนด Init Server ของคอนโทรล Inet เมื่อกำหนดแล้วทางด้าน Client ก็สามารถติดต่อเข้ามาได้ซึ่งในการติดต่อเราใช้คำสั่ง connection method เมื่อทำการติดต่อเข้ามาแล้วทางด้าน Server จะเกิดเหตุการณ์ที่ชื่อว่า Accept event ขึ้นเพื่อเป็นการรับรู้ว่าได้รับการติดต่อมาแล้วถ้า Server ยอมรับการติดต่อจากทางด้าน Client ก็จะส่งคำสั่ง Acception connection method กลับไปยังทางด้าน Client ที่ได้ทำการติดต่อเข้ามา ซึ่งจะทำให้เกิดเหตุการณ์ connetction event ซึ่งแสดงว่าได้ทำการติดต่อเสร็จสมบูรณ์แล้ว

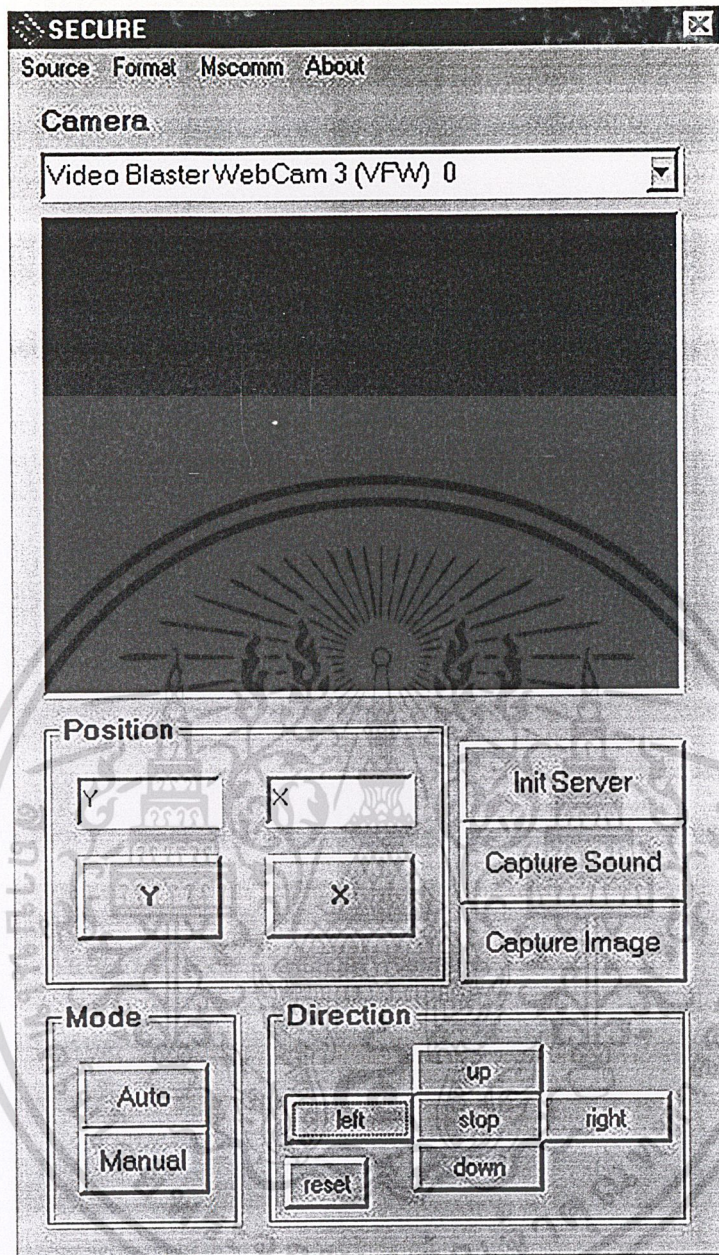
เมื่อทำการติดต่อเสร็จสมบูรณ์แล้ว เราก็สามารถทำการส่งหรือรับข้อมูลได้ นั่นคือ ทางด้าน Server เมื่อเราได้จับภาพจากกล้อง Webcam เราก็ส่งภาพผ่านเครือข่าย เช่นเดียวกับทางด้าน เสียง ถ้าเราได้ทำการบันทึกเสียงจาก microphone เราก็สามารถส่งผ่านเครือข่ายได้ ส่วนคำสั่งควบคุม ทางด้าน Client ก็สามารถส่งไปยัง Server ได้เช่นกัน เนื่องจากการส่งแบบ full duplex ของ โพรโทคอล TCP/IP

เมื่อต้องการยกเลิกการติดต่อทางด้าน Client ก็จะทำการส่งคำสั่ง Disconnect method มายังทางด้าน Server ซึ่งจะทำให้ทาง Server รับรู้ว่าได้ยกเลิกการติดต่อแล้วซึ่งเป็นอันสิ้นสุดการสื่อสาร รูปแสดงการทำงานสามารถสรุปได้ดังนี้



รูปที่ 7.11 กลไกการเชื่อมต่อผ่านทางอินเทอร์เน็ตของคอนโทรล Active X LEADTOOL ที่ชื่อว่า Inet

รูปต่อไปจะแสดงหน้าจอควบคุมการทำงานทางด้านฝั่ง Client และ Server



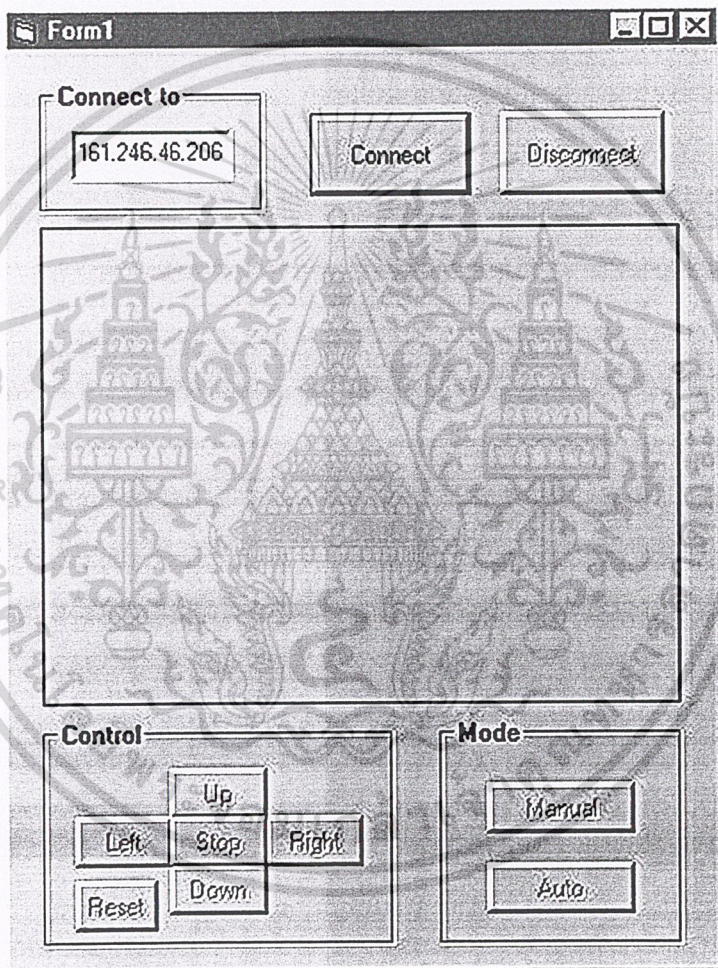
รูปที่ 7.12 แสดงหน้าจอทางด้านฝั่ง Server

สำหรับปุ่มแต่ปุ่มทำหน้าที่ต่างการกันดังนี้

- Position X,Y ใช้ทำหน้าที่บอกตำแหน่งของกล้องว่าอยู่ที่ตำแหน่งใด
- Init Server ไว้สำหรับทำหน้าที่เพื่อเตรียมให้ Computer นั้นมีสถานะเป็น Server เพื่อรอรับการติดต่อจากทางด้าน Client
- Capture Sound ไว้สำหรับบันทึกเสียงเพื่อทำการส่งผ่านเครือข่าย
- Capture Image ไว้สำหรับบันทึกภาพเพื่อทำการส่งผ่านเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Direction ไว้สำหรับควบคุมทิศทางของกล้อง โดยคำสั่ง Reset ใช้เพื่อให้กล้องกลับมาอยู่ที่ตำแหน่งเริ่มต้น คือ จุดพิกัด (0,0)
- Mode เป็นปุ่มที่ใช้สำหรับเลือกโหมดการทำงานของตัวควบคุมกล้องว่าจะทำงานในโหมดใด โดย โหมด Auto จะเป็นการทำงานของการควบคุมกล้อง โดยการใช้ sensor ultrasonic ในการควบคุมแบบอัตโนมัติในการติดตามวัตถุ ส่วนโหมด Manual เป็นโหมดที่ใช้ปุ่มควบคุมผ่านทางหน้าจอกอมพิวเตอร์



รูปที่ 7.13 แสดงหน้าจอควบคุมการติดต่อทางด้านฝั่ง ไคลเอนต์

โดยมีปุ่มควบคุมการทำงานคล้ายเดียวกับทางด้าน Server จะมีส่วนที่แตกต่างคือ

- Text box Connect to ใช้สำหรับใส่ชื่อ Server ที่จะทำการติดต่อ
- ปุ่ม Connect /Disconnect สำหรับใช้ในการสั่งให้เชื่อมต่อหรือไม่เชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 หลักการออกแบบเซนเซอร์อุลตราโซนิกเพื่อใช้ตรวจจับสิ่งกีดขวาง

อาศัยคุณสมบัติการทำงานของสัญญาณอุลตราโซนิก โดยอาศัยหลักการสะท้อนของสัญญาณ และเวลาที่สัญญาณอุลตราโซนิกใช้เดินทางในอากาศ

7.3.1 หลักการทำงานของวงจรโดยรวม

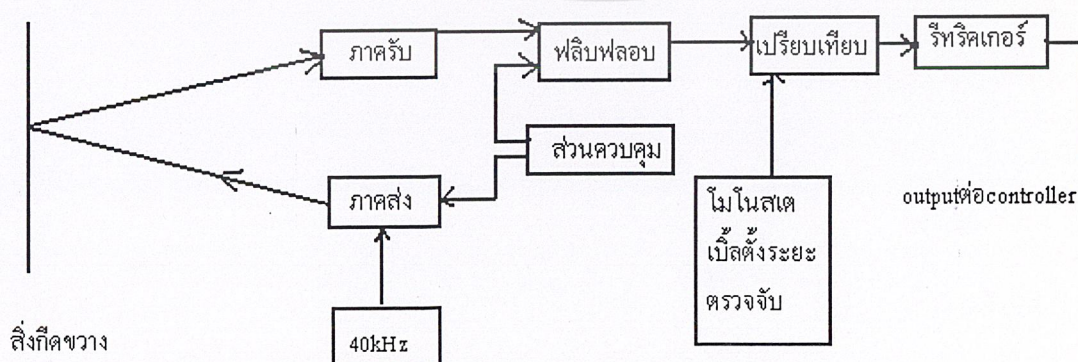
ทำการส่งสัญญาณผ่านตัวอุลตราโซนิกทรานสดิวเซอร์ภาคส่ง ด้วยสัญญาณ โมโนสเตเบิล ซึ่งถูกมอดคูเลทด้วยสัญญาณ 40 kHz

สัญญาณภาครับที่ถูกส่งมาผ่านอุลตราโซนิกทรานสดิวเซอร์ภาครับ จะนำไปต่อเป็นสัญญาณนาฬิกาของฟลิปฟล็อปซึ่งถูกทำการรีเซทในเวลาเดียวกับเวลาที่ใช้ในการส่งสัญญาณภาคส่ง

จากหลักการข้างต้นจะทำให้ได้ว่าตัวฟลิปฟล็อปจะนับที่ค่าเวลาที่สัญญาณอุลตราโซนิกใช้เดินทางจากภาคส่งสะท้อนสิ่งกีดขวางกลับเข้าไปภาครับในรูปของคาบสัญญาณทางไฟ ซึ่งจะได้ว่าคาบสัญญาณดังกล่าวแปรผันตรงกับระยะทางที่เกิดการสะท้อนซึ่งแปรผันตรงกับเวลาที่สัญญาณอุลตราโซนิกใช้เดินทางในอากาศ

เราทำการกำหนดครีสมิการตรวจจับของเซนเซอร์เพื่อนำมาใช้ในส่วนการทำงานของโครงการในโหมดออโตเมติกได้โดย สร้างสัญญาณ โมโนสเตเบิลขึ้นมาเพื่อใช้เปรียบเทียบกับสัญญาณคาบทางด้านเอาต์พุตของฟลิปฟล็อป โดยเมื่อสัญญาณคาบของฟลิปฟล็อปมีค่าน้อยกว่าสัญญาณ โมโนสเตเบิลที่ตั้งไว้ ซึ่งแสดงว่ามีสิ่งกีดขวางมาทำให้เกิดการสะท้อนที่ระยะภายในรัศมีการทำงานของเซนเซอร์ จะทำให้เกิดการทริกสัญญาณซึ่งสัญญาณดังกล่าวจะถูกนำไปต่อกับอุปกรณ์ ไมโครคอนโทรลเลอร์เพื่อตั้งการให้มอเตอร์ควบคุมล้อวงล้อโอมายังตำแหน่งที่เซนเซอร์ตรวจจับสัญญาณได้

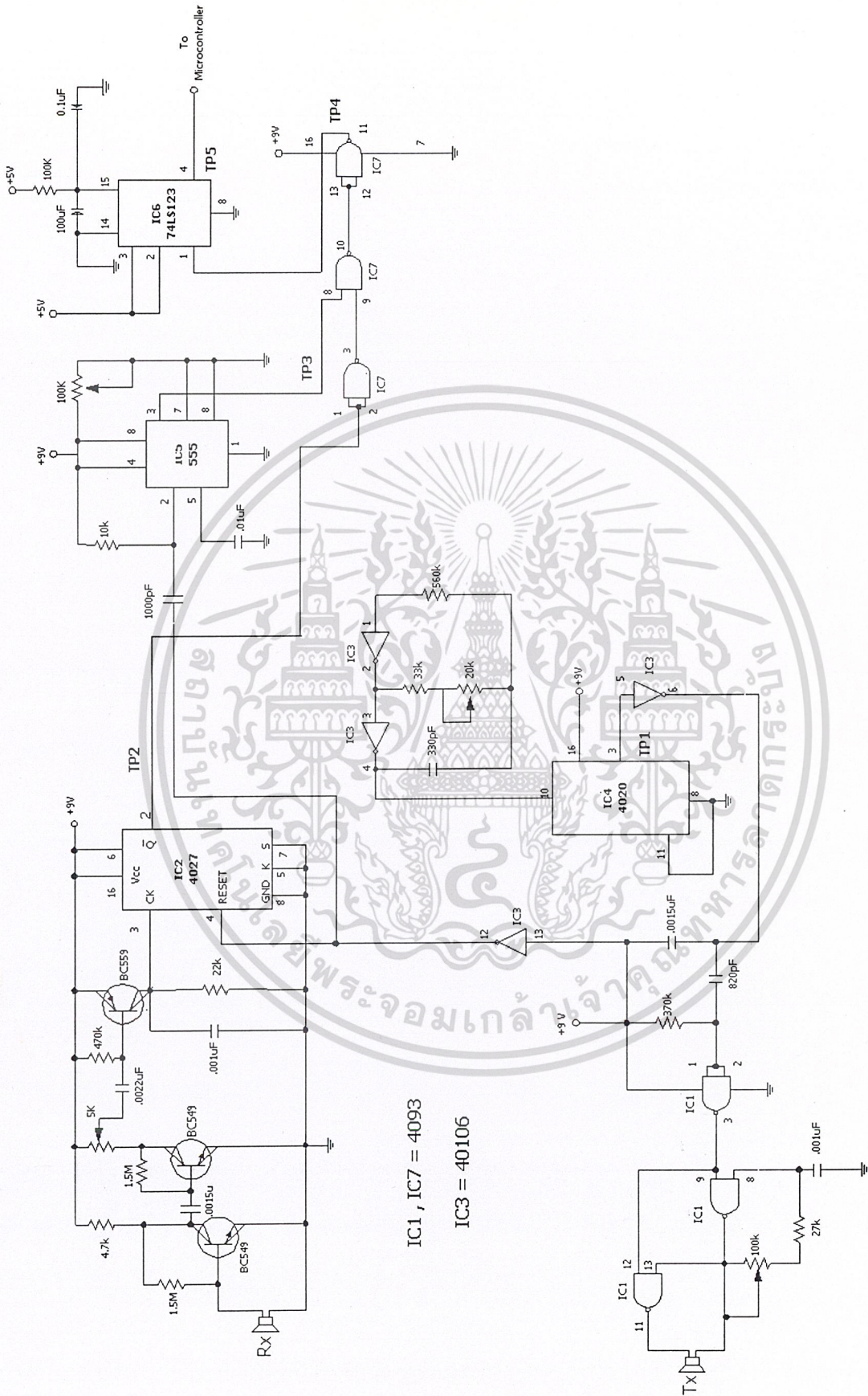
จากหลักการข้างต้นเราสามารถออกแบบบล็อกไดอะแกรมของวงจรได้เป็นดังรูปที่ 7.14



รูปที่ 7.14 บล็อกไดอะแกรมการทำงานของวงจรส่วนเซนเซอร์

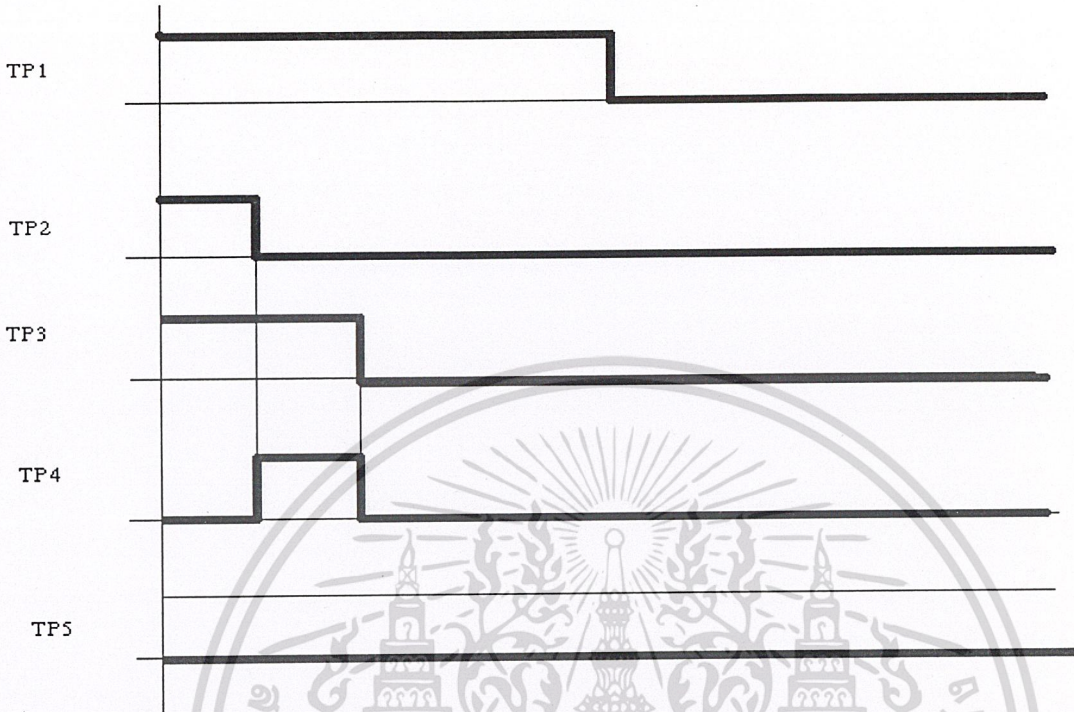
ซึ่งจากบล็อกไดอะแกรมตามรูปนำไปออกแบบวงจรได้เป็น ดังรูปที่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารรูปที่ 7.15 รูปวงจรทั้งหมดของวงจรส่วนเซนเซอร์ที่จะนำไปต่อกับไมโครคอนโทรลเลอร์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรข้างต้นจะมีการทำงานโดยการควบคุมสัญญาณที่จุดทดสอบ(TP)ต่างๆให้เป็นไปตามรูปที่ 7.16 เพื่อให้การทำงานของวงจรเป็นไปตามบล็อกไดอะแกรมที่ออกแบบไว้

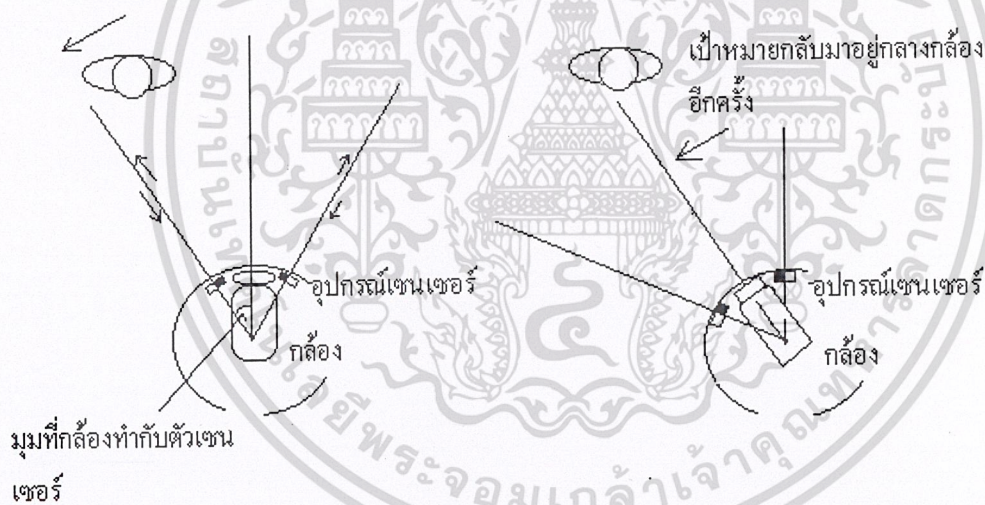


รูปที่ 7.16 สัญญาณที่จุดทดสอบต่างๆ แสดงการทำงานของวงจร

จากรูปที่ 7.15 และ 7.16 จะเห็นว่าสัญญาณ TP1 คือสัญญาณจากส่วนควบคุมมีความถี่ 1 Hz ซึ่งเป็นตัวควบคุมส่วนต่างๆให้ทำงานความสัมพันธ์กันโดยจะทำหน้าที่เสมือนว่าเป็นสวิทช์อัตโนมัติที่คอยเปิดปิดทุกๆหนึ่งวินาที สัญญาณ TP2 คือสัญญาณที่มีคาบเวลาสัมพันธ์กับระยะเวลาที่คลื่นอุลตราโซนิกใช้เดินทางในอากาศซึ่งสัมพันธ์กับระยะทางที่ใช้ในการสะท้อนของคลื่นอุลตราโซนิกด้วย ซึ่งทำให้ได้ระยะห่างระหว่างสิ่งกีดขวางที่ตัวเซนเซอร์ตรวจจับได้จะแปรผันตรงกับสัญญาณคาบเวลาของสัญญาณ TP2 สัญญาณ TP3 คือสัญญาณคาบเวลาที่ถูกร่างขึ้นมาโดยวงจรโมโนเสเตเบิลเพื่อนำมาเปรียบเทียบกับสัญญาณ TP2 โดยถ้าสัญญาณ TP2 มีค่าน้อยกว่าสัญญาณ TP3 ที่ตั้งเอาไว้หมายความว่าระยะห่างของสิ่งกีดขวางกับตัวเซนเซอร์มีค่าอยู่ในช่วงที่ตั้งเอาไว้ซึ่งถูกกำหนดให้เป็นรัศมีขอบเขตที่เซนเซอร์จะสามารถตรวจจับสิ่งกีดขวางได้ จากการดำเนินงานในทางลอจิกโดยการกลับค่า TP2 โดยน็ทเกต(Not Gate)ก่อนนำมาทำการแอน(And)กับสัญญาณ TP3 จะทำให้ได้ว่าเมื่อ TP2มีค่าน้อยกว่า TP3ผลลัพธ์จากการดำเนินงานดังกล่าวออกมาเป็นสัญญาณ TP4ซึ่งถ้าTP2มากกว่าTP3สัญญาณTP4จะไม่เกิดเป็นสัญญาณดังรูปแต่จะมีค่าเป็นศูนย์ ส่วนสัญญาณ TP5 คือการนำสัญญาณ TP4 มารีทริกด้วยคาบเวลาประมาณ 1 วินาที เพื่อให้สัญญาณพัลส์ความถี่ 1 Hz ที่ TP4 เมื่อมีสิ่งกีดขวางค้างอยู่ในขอบเขตตรวจจับกลายเป็นสัญญาณไฟตรงก่อนสัญญาณไฟตรงถูกกลับให้เป็น ศูนย์(TP5)ซึ่งสัญญาณนี้เองที่จะถูกนำไปต่อเข้าคอนโทรลเลอร์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.2 การติดตั้งส่วนเซนเซอร์เข้ากับขาตั้งเพื่อใช้ในการควบคุมสำหรับการทำงานในโหมดAuto

การออกแบบ โดยกำหนดให้ การทำงานของเซนเซอร์มีลักษณะการตรวจสอบสิ่งกีดขวาง โดยอาศัยการสะท้อนแบบเป็นเส้นตรง จึงทำการติดตั้งตัวเซนเซอร์ดังรูปที่ เพื่อให้การทำงานในโหมดAutoเป็นการทำงานในลักษณะติดตามการเคลื่อนไหว โดยเราให้รัศมีการทำงานของเซนเซอร์ที่มีลักษณะเป็นเส้นตรงดังกล่าว ขนาบข้างมุมกล้องทั้งสอง โดยเมื่อเป้าหมายมีการเคลื่อนที่ออกจากมุมกล้องทางใดทางหนึ่ง(ซ้ายหรือขวา)ก่อนที่เป้าหมายจะพ้นขอบเขตของมุมกล้องออกไป เป้าหมายจะเคลื่อนที่ไปยังขอบเขตของตัวเซนเซอร์และถูกตรวจสอบโดยตัวเซนเซอร์ด้านเดียวกับที่เป้าหมายเคลื่อนที่ออก หลังจากตรวจสอบได้ ตัวเซนเซอร์จะไปสั่งการควบคุมขาตั้งกล้องให้หมุนไปด้วยมุมมองเดียวกับมุมระหว่างตำแหน่งขอบเขตเซนเซอร์กับตำแหน่งกึ่งกลางกล้อง ดังนั้นจะได้ว่าตำแหน่งกึ่งกลางกล้องจะย้ายไปอยู่ที่ตำแหน่งของเซนเซอร์เดิม ได้ว่าเป้าหมายกลับมาอยู่ตรงกลางกล้องอีกครั้ง ในที่นี้ทำการประมาณค่ามุมกล้องได้ว่ามีค่าอย่างน้อย 60 องศา โดยเป็นทางซ้ายและขวาจากจุดกึ่งกลางตำแหน่งของกล้อง ข้างละ 30 องศา เราทำการติดตั้งตัวเซนเซอร์ด้วยมุมดังกล่าว และทำการโปรแกรมตัวคอนโทรลเลอร์ให้หมุนเมื่อพบการตรวจจับด้วยมุมตัวกล้องเช่นกัน



รูปที่ 7.17 ก.เมื่อเป้าหมายเคลื่อนที่ออกจากกึ่งกลางกล้อง

ข. เมื่อเซนเซอร์ตรวจจับเป้าหมายได้ และควบคุมให้กึ่งกลางกล้องย้ายตำแหน่งเดิมของเซนเซอร์

บทที่ 8

สรุปและวิเคราะห์ผลการทดลอง

8.1 การทดลองที่ 1 การรับ-ส่งรหัสควบคุมมอเตอร์และการหมุนของมอเตอร์

จากการทดลองรับ-ส่งรหัสควบคุมมอเตอร์โดยใช้โปรแกรม Visual Basic พบว่าไมโครคอนโทรลเลอร์มีการตอบสนองต่อคำสั่งทันทีที่ส่งรหัสควบคุมไปที่ไมโครคอนโทรลเลอร์และทำงานได้ถูกต้องตามโค้ดการทำงานที่คอมพิวเตอร์ส่งมาให้ สำหรับตำแหน่งการหมุนของมอเตอร์ที่แสดงไว้ในหน้าจอของโปรแกรม Visual Basic และตำแหน่งจริงๆที่มอเตอร์หมุนนั้นได้ทำการทดลองโดยผลการทดลองแสดงในตารางที่ 8.1 –8.2

ตำแหน่งที่มอเตอร์แกน X หมุน (องศา)	ตำแหน่งมอเตอร์แกน X ที่แสดงบนหน้าจอ vb (องศา)	ค่าความคลาดเคลื่อน (เปอร์เซ็นต์)
17	19.8	-16.47
48	50.4	-5
70	73.8	-5.43
90	90	0
-20	-18	10
-50	-46.8	6.4
-72	-68.4	5
-92	-90	2.17

ตารางที่ 8.1 ผลการทดลองวัดตำแหน่งมอเตอร์แกน X ที่หมุนจริงกับค่าตำแหน่งที่แสดงบนหน้าจอโปรแกรม Visual Basic

โดยที่ค่าบวกขององศาคือค่าองศาเมื่อเทียบกับจุดเริ่มต้นไปทางขวา ค่าลบคือค่าองศาเมื่อเทียบกับจุดเริ่มต้นไปทางซ้าย

ตำแหน่งที่มอเตอร์แกน Y หมุน (องศา)	ตำแหน่งมอเตอร์แกน Y ที่แสดงบนหน้าจอ vb (องศา)	ค่าความคลาดเคลื่อน (เปอร์เซ็นต์)
11	10.8	1.81
24	23.4	2.5
33	32.4	1.81
-12	-12.6	-4.76
-23	-23.4	-1.74
-32	-32.4	-1.25

ตารางที่ 8.2 ผลการทดลองวัดตำแหน่งมอเตอร์แกน Y ที่หมุนจริงกับค่าตำแหน่งที่แสดงบนหน้าจอ โปรแกรม Visual Basic

โดยที่ค่าบวกขององศาเป็นค่าองศาเมื่อเทียบกับจุดเริ่มต้นไปทางด้านบนและค่าลบคือค่าองศาเมื่อเทียบกับจุดเริ่มต้นไปทางด้านล่าง

สรุปผลการทดลองที่ 1

จากผลการทดลองในตารางที่ 8.1 ซึ่งเป็นการทดลองหาค่าความคลาดเคลื่อนของค่าองศาที่มอเตอร์แกน X หมุนจริงกับค่าองศาที่แสดงในโปรแกรม Visual Basic พบว่าค่าความคลาดเคลื่อนจะมีค่าอยู่ในช่วง 0-16 % ซึ่งค่าความคลาดเคลื่อนในตารางที่ 8.1 มีค่าสูงกว่าค่าความคลาดเคลื่อนในตารางที่ 6.2 ทั้งนี้เนื่องจากมอเตอร์ในแกน X ต้องรับน้ำหนักมอเตอร์และกลิ้ง ซึ่งค่าโหลดนี้ทำให้การหมุนของมอเตอร์คลาดเคลื่อนไปเล็กน้อย การแก้ไขควรทำโดยต้องออกแบบฐานกลิ้งให้มีความเหมาะสมกว่านี้ ส่วนผลการทดลองในตารางที่ 8.2 พบว่า ค่าความคลาดเคลื่อนมีน้อยมากเนื่องจากมอเตอร์ในแกน Y มีการรับน้ำหนักเพียงตัวกลิ้งซึ่งต่างกับมอเตอร์แกนอื่นที่ต้องรับน้ำหนักของมอเตอร์และกลิ้ง และปัญหาที่เกิดขึ้นเมื่อเริ่มใช้งานคือ จุดเริ่มต้นของมอเตอร์ไม่แน่นอน ดังนั้นในส่วนนี้จึงต้องแก้ปัญหาโดยได้ติดตั้งไมโครสวิทช์ที่บริเวณจุดตำแหน่งอ้างอิงโดยเมื่อเราเริ่มต้นใช้งานในส่วนควบคุมนี้ จะให้มอเตอร์หมุนไปเรื่อยๆ จนเจอจุดอ้างอิงหรือก็คือจุดที่มีไมโครสวิทช์อยู่นั่นเอง เมื่อมอเตอร์สัมผัสกับไมโครสวิทช์ก็จะทำให้ทราบตำแหน่งขณะนั้นได้

8.2 การทดลองในส่วนของการส่งภาพและเสียง

เราจะแบ่งการทดลองออกเป็น 2 ส่วน คือ การส่งภาพ และ การส่งเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2.1 การทดลองส่วนการส่งภาพ

การทดลองในส่วนนี้เราจะทำการใช้ความละเอียดของภาพและความละเอียดของสีเป็นตัวแปรในการทดลองเพื่อดูผล คือ ความต่อเนื่องของภาพที่ทางด้านClient โดยขนาดของไฟล์เป็นค่าที่อ่านได้จากตัวกล้องซึ่งได้ผ่านการ compression แล้วได้ผลดังตาราง

ความละเอียดของภาพ	จำนวนสี	ขนาดไฟล์(byte)	คุณภาพของภาพที่ได้และความต่อเนื่อง
160*120	1420	28800	ภาพเบลอมากที่สุดแต่ความต่อเนื่องดีที่สุด
	RGB24	57600	ภาพเบลอมากที่สุดแต่ความต่อเนื่องพอใช้
	RGB555(16 bit)	38400	ภาพเบลอมากที่สุดแต่ความต่อเนื่องดี
320*240	1420	115200	ภาพเบลอพอใช้แต่ความต่อเนื่องดี
	RGB24	230400	ภาพเบลอพอใช้แต่ความต่อเนื่องแย่มาก
	RGB555(16 bit)	153600	ภาพเบลอพอใช้ความต่อเนื่องพอใช้
640*480	1420	460800	ภาพชัดแต่ความต่อเนื่องแย่มาก
	RGB24	921600	ภาพชัดที่สุดแต่ความต่อเนื่องแย่มากที่สุด
	RGB555(16 bit)	614400	ภาพชัดแต่ความต่อเนื่องแย่มาก

ตารางที่ 8.3แสดงผลการทดลองในการส่งภาพออกทางด้าน client

สรุปผลการทดลองที่ 8.2.1

จากตารางการทดลองจะเห็นได้ว่าขนาดไฟล์ของภาพมีผลต่อความต่อเนื่องของภาพและคุณภาพของภาพ โดยถ้าภาพที่มีความละเอียดสูงและจำนวนสีมากจะทำให้ขนาดไฟล์มีขนาดใหญ่ด้วย ซึ่งทำให้การส่งภาพผ่านเครือข่ายเป็นไปอย่างช้า ๆ นั่นคือผลจะทำให้เกิดความต่อเนื่องที่แย่ในทางกลับกันก็เช่นเดียวกัน ถ้าขนาดไฟล์ของภาพเล็กนั่นคือมีความละเอียดและจำนวนสีที่น้อยกว่าทำให้การส่งภาพเป็นไปได้อย่างรวดเร็ว ทำให้ความต่อเนื่องที่ออกมาดีขึ้น นอกจากนี้แล้วคุณภาพของภาพที่ได้ยังขึ้นอยู่กับแสงสว่างภายในห้องอีกด้วย

แต่ในการใช้งานด้านรักษาความปลอดภัยนั้นเราจำเป็นต้องรู้ว่ามิสิ่งใดเกิดขึ้นบ้างภายในจอภาพ นั่นคือ ภาพที่รับได้ต้องมีความละเอียดและจำนวนสีที่พอสมควร และความต่อเนื่องก็ต้องพอใช้ได้ ดังนั้น เราจึงใช้ความละเอียดของภาพที่ 320*240 และสีที่ 16 บิต เป็นค่าปกติในการใช้งาน

หมายเหตุ : การทดลองนี้ใช้การบีบอัดข้อมูลภาพแบบ LEAD Proprietary Compression Format

Bits per pixel: 8 for grayscale, 24 for colorในการส่งผ่านเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2.2 การทดลองในส่วนของการส่งเสียง

การทดลองในส่วนนี้เราใช้รูปแบบของเสียงเป็นตัวแปรในการหาค่า Delay ที่ออกมาทางด้าน client ผลการทดลองมีดังตาราง

รูปแบบเสียง	คุณภาพเสียง	Delay (วินาที)	รูปแบบเสียง	คุณภาพเสียง	Delay (วินาที)
11.025 kHz, Mono, 8-bit	พอฟังได้	1	22.05 kHz, Mono, 16-bit	พอฟังได้	1.2
11.025 kHz, Stereo, 8-bit	พอฟังได้	1	22.05 kHz, Stereo, 16-bit	พอฟังได้	1.2
11.025 kHz, Mono, 16-bit	พอฟังได้	1	44.1 kHz, Mono, 8-bit	พอฟังได้	1.2
11.025 kHz, Stereo, 16-bit	พอฟังได้	1	44.1 kHz, Stereo, 8-bit	พอฟังได้	1.2
22.05 kHz, Mono, 8-bit	พอฟังได้	1	44.1 kHz, Mono, 16-bit	พอฟังได้	1.5
22.05 kHz, Stereo, 8-bit	พอฟังได้	1	44.1 kHz, Stereo, 16-bit	พอฟังได้	1.5

ตารางที่ 8.4 แสดงผลการทดลองการส่งเสียงผ่านเครือข่าย

สรุปผลการทดลอง 8.2.2

จากผลการทดลองจะเห็นว่าคุณภาพเสียงที่รับได้ต่างก็เท่า ๆ กัน แต่ Delay มีขนาดที่แตกต่างกันเพียงเล็กน้อย ดังนั้นเราจึงสามารถใช้รูปแบบเสียงแบบไหนก็ได้ ทั้งนี้เนื่องมาจากหุคที่เราไม่สามารถแยกแยะรูปแบบเสียงที่มีความแตกต่างตามการทดลองได้ แต่สำหรับค่า Delay นั้นจะมีค่าแปรผันตามตัวแปรต่าง ๆ อีก เช่น ระยะทางการในการสื่อสาร ตัวกลางในการสื่อสาร เป็นต้น

แนวทางในการพัฒนาต่อไป

สำหรับในส่วนของการสื่อสารผ่านอินเทอร์เน็ตนั้นเราสามารถทำการพัฒนาต่อไปได้โดย

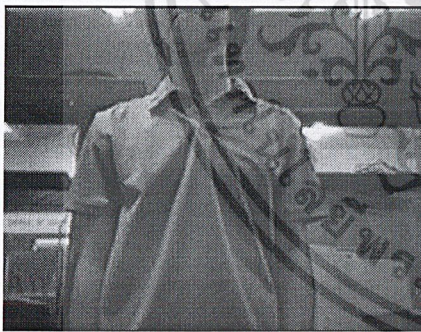
- พัฒนาให้เป็นระบบ multi-user คือสามารถส่งข้อมูลไปได้หลาย เครื่อง
- พัฒนาให้สามารถรับและส่งภาพและเสียงได้ภายใน User เดียวกัน นั่นคือ สามารถรับภาพและเสียง พร้อมทั้ง สามารถส่งภาพและเสียงได้ภายในเวลาเดียวกัน
- พัฒนาการบีบอัดไฟล์ข้อมูลให้มีขนาดเล็กมากขึ้นเพื่อการสื่อสารที่รวดเร็ว
- เนื่องจากว่า control active X ที่ใช้เป็น shareware สามารถใช้งานได้แค่ 2 เดือน เท่านั้น ถ้าเราจะทำการสร้าง control ขึ้นมาใช้เอง ขอแนะนำว่าให้ใช้ Direct X ในการพัฒนาซึ่งเป็นทางเลือกที่ง่ายกว่า เนื่องจาก API Direct X นั้น เป็น API ที่สามารถสามารถเข้าได้ถึงการควบคุมการทำงานของทางด้าน ฮาร์ดแวร์ของเครื่องได้ง่ายกว่าการใช้ภาษา C โดยตรง โดยอาจจะมีการใช้

เอกสารนี้ Direct show, Direct music เป็นส่วนในการแสดงภาพและเสียงได้ เป็นต้น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

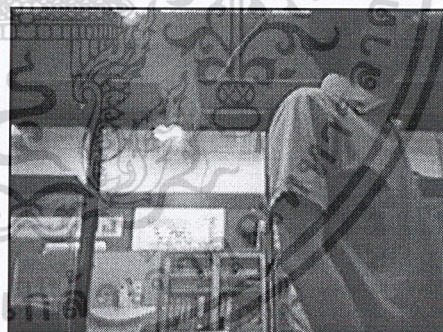
8.3 การทดลองในส่วนของ เซนเซอร์

ทำการทดสอบในส่วนของเซนเซอร์ซึ่งใช้ตัวอุลตราโซนิกทรานสดิวเซอร์ ซึ่งมีการส่งคลื่นอุลโซนิกเป็นลักษณะใกล้เคียงกับเส้นตรง โดยมีรัศมีการตรวจจับที่ด้านข้างจากจุดกึ่งกลางของตัวรับและตัวส่งเล็กน้อย ซึ่งในกรณีที่เป้าหมายที่ใช้ในการตรวจจับเป็นคน ซึ่งมีขนาดความกว้างของตัวประมาณอย่างน้อย 30 เซนติเมตร การใช้งานเซนเซอร์ให้เป็นไปตามลักษณะที่ออกแบบไว้จึงพบปัญหาการคลาดเคลื่อนในการควบคุมเป้าหมายให้อยู่ตรงกลางกล้องเพียงเล็กน้อย โดยจากการทดสอบระยะห่างของเป้าหมายกับตัวสะท้อนเพื่อหาระยะที่วงจรมอเตอร์จะตรวจจับสัญญาณที่สะท้อนได้อย่างแม่นยำที่สุดโดยไม่เกิดการรบกวนจากสัญญาณรบกวนอื่นคือที่ระยะตั้งกึ่งกลางห่างออกไปไม่เกิน 70 เซนติเมตร ซึ่งจากการทดลองนำค่า 70 เซนติเมตรที่ได้ไปตั้งคาบเวลาของสัญญาณ TP3 ในรูปที่ 7.16 โดยทำการทดสอบจากระยะจริงของสิ่งกีดขวางที่เกิดขึ้น ทำการปรับตั้งค่าของวงจรมอเตอร์ ไมโครคอนโทรลเลอร์ IC555 ทดสอบค่าที่จุด TP5 ให้มีค่าเป็นศูนย์ที่ระยะต่ำกว่า 70 เซนติเมตรซึ่งหมายถึงเซนเซอร์สามารถตรวจจับสิ่งกีดขวางได้และเมื่อสิ่งกีดขวางอยู่ห่างเกินออกไปจากระยะ 70 เซนติเมตรจากตัวเซนเซอร์ แต่ยังคงอยู่ในแนวเส้นตรงเดิม ให้สัญญาณมีค่าเป็น 5 โวลท์ซึ่งหมายถึงสิ่งกีดขวางเกินระยะตรวจจับของเซนเซอร์และเซนเซอร์ไม่สามารถตรวจจับสิ่งกีดขวางได้แล้ว

หลังจากทำการทดสอบและติดตั้งดังกล่าว ทำการทดสอบการติดตามของเซนเซอร์โดยดูจากภาพที่ปรากฏบนหน้าจอ ดังรูปที่ 8.1 ก. ข. ค.



รูปที่ 8.1 ก. เมื่อเป้าหมายอยู่ตรงกลาง

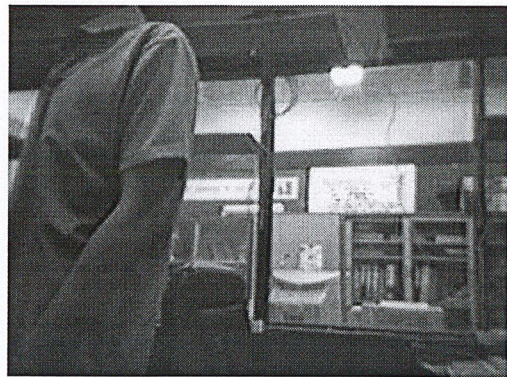
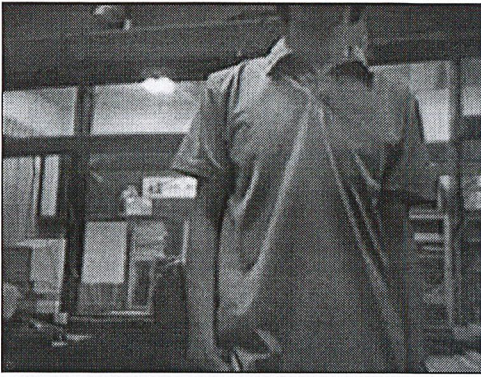


รูปที่ 8.1 ข. เมื่อเป้าหมายเคลื่อนไปทาง ซ้าย



รูปที่ 8.1 ค. เมื่อเซนเซอร์ควบคุมกล้องให้เคลื่อนมาหาเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 ก เมื่อเป้าหมายอยู่กลางกล้อง เมื่อเริ่มแรก

รูปที่ 8.2 ข เมื่อเป้าหมายเคลื่อนที่ออกไปทาง

ขวา



รูปที่ 8.2 ค เมื่อเซนเซอร์ควบคุมกล้องให้เคลื่อนมาหาเป้าหมาย

จากการทดลองเมื่อให้บุคคลที่เป็นเป้าหมายการติดตามของเซนเซอร์เคลื่อนที่ออกจากจุดกลางของกล้องตามรูป 8.1 ก และ 8.2 ก ไปทางซ้ายและขวา จากรูป 8.1 ข และ 8.2 ข จะเห็นว่าเมื่อเป้าหมายเคลื่อนไปที่บริเวณขอบของรูป จะเข้าไปอยู่ภายในรัศมีของเซนเซอร์ ทำให้เซนเซอร์สั่งการให้กล้องหมุนไปยังตำแหน่งของเซนเซอร์ซึ่งเป็นตำแหน่งเดียวกับที่เป้าหมายยืนอยู่ เป้าหมายจึงกลับมาอยู่ที่กลางของภาพอีกครั้งดังรูปที่ 8.1 ค และ 8.2 ค

จากรูปที่ 8.1 ค และ 8.2 ค ทำให้เราเห็นว่าเมื่อเป้าหมายหยุดเคลื่อนที่เมื่อพบการตรวจจับเพื่อรอให้กล้องติดตาม หลังจากการเคลื่อนที่ติดตามตามมุมที่ตั้งไว้เป็นที่เรียบร้อยแล้วเป้าหมายมีความคลาดเคลื่อนจากกึ่งกลางภาพเล็กน้อย เนื่องจากตัวเซนเซอร์มีรัศมีการตรวจสอบที่ไม่ใช่เส้นตรงทีเดียว แต่มีความกว้างเล็กน้อยซึ่งความคลาดเคลื่อนนี้สามารถทดสอบและปรับแต่งตำแหน่งมุมที่วางสำหรับตัวเซนเซอร์ได้ภายหลังให้ได้ความแม่นยำในลักษณะที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. วรพจน์ กรแก้ววัฒนกุล, "เรียนและปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51", บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
2. กฤษดา ใจเย็น, "เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม", บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, หน้า 43-65
3. Steve Brown, "Visual Basic 6 In Record Time", SYBEX, 600p, 1998
4. สุวัฒน์ ปุณณะชัยยะ, "เปิดโลกของ TCP/IP และ โพรโทคอลของอินเทอร์เน็ต", โปรวิชั่น, 312 หน้า, 2543
4. www.microsoft.com/msdn
5. www.allapi.net
6. www.leadtools.com



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้