สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ป้ายโฆษณาไร้สายส่งข้อมูลผ่านอินเทอร์เน็ต

WIRELESS DISPLAY BOARD VIA INTERNET

T 0 5 0 3 1 4

โดย

นาย จักรพงศ์ วีระพงษ์

นาย อาคม ทองลอย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50314
วัน,เดือน,ปี 29 มิ.ย. 2547

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้ายโฆษณาไร้สายส่งข้อมูลผ่านอินเทอร์เน็ต

WIRELESS DISPLAY BOARD VIA INTERNET

โดย

นาย จักรพงศ์ วีระพงษ์ รหัส 43015203

นาย อาคม ทองลอย รหัส 43015244

อาจารย์ที่ปรึกษา

อาจารย์ อิทธิภูมิ บุญพิคำ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ป้ายโฆษณาไร้สายส่งข้อมูลผ่านอินเทอร์เน็ต

ผู้จัดทำ

1. นาย จักรพงศ์ วีระพงษ์ 43015203
2. นาย อาคม ทองฉอย 43015244

.....

(อาจารย์ อิทธิภูมิ บุญพิคำ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้ายโฆษณาไร้สายส่งข้อมูลผ่านอินเทอร์เน็ต

นาย จักรพงศ์ วีระพงษ์
นาย อาคม ทองลอย
อ. อิทธิภูมิ บุญพิคำ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

โครงงานฉบับนี้นำเสนอป้ายโฆษณาและประชาสัมพันธ์ โดยใช้ภาคแสดงผล LED DOTMATRIX ขนาด 16*64 ซึ่งใช้อินเทอร์เน็ตเชื่อมต่อไปยังอีกเครือข่าย เมื่อทำการติดต่อกันได้เรียบร้อยแล้วก็สามารถส่งข้อมูลที่ต้องการแสดงผ่านทางแป้น Keyboard ของ PC ซึ่งเป็น Client ผ่านเครือข่ายอินเทอร์เน็ตไปยัง Sever ซึ่งต่ออยู่กับป้ายแสดงผล โดยส่วนของการส่งข้อมูลระหว่างป้ายแสดงผลกับ PC ที่ Sever นั้นประยุกต์ใช้งานเป็น Wireless communication สู่ภาครับซึ่งอยู่ส่วนเดียวกับป้ายแสดงผลควบคุมด้วย Microcontroller โดย MCS-51 จะรับข้อมูลที่เข้ามาแล้วส่งออกไปยังภาคแสดงผล LED DOTMATRIX เพื่อแสดงผลตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIRELESS DISPLAY BOARD VIA INTERNET

Mr. Jakkrapong Veerapong

Mr. Arkom Tongloi

Mr. Ittibhoom Boonpikum (Advisor)

Education Year 2002

Abstract

This project is presented on Promotion and Advertisement by using a 16*64 LED DOT MATRIX display unit via internet connected toward another network. When the system can be connected, then the presenting information also can be transmitted via PC's keyboard which is the CLIENT station via internet network to SERVER that is connected with display board.

The transmitter unit between the Display board and SERVER will use Wireless Communication to receiver which is in the same part as the display board controlled by MICROCONTROLLER and MCS-51 will receive the entered data. Then transmit to the display unit LED DOT MATRIX in order to display as required.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 การทำงานโดยรวมของระบบ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 คุณสมบัติของหลอดไดโอดนำแสง (LED)	3
2.2 โครงสร้างไมโครคอนโทรลเลอร์ MCS-51	5
2.3 การสแกน (SCANNING)	8
2.3.1 การสแกนทางคอลัมน์ (Column Scan)	9
2.3.2 การสแกนทางโรว์ (Row Scan)	9
2.4 วงจรขับกระแส (Current Driver)	9
2.5 ทฤษฎีระบบเครือข่าย และ วินโดวส์ซ็อกเก็ต	12
2.5.1 ระบบเครือข่าย(Network System)	12
2.5.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์	12
2.5.1.2 สถาปัตยกรรมของอินเตอร์เน็ต	12
2.5.1.3 ข้อกำหนดรูปแบบของเกตเวย์	13
2.5.2 โมเดลโอเอสไอ	14
2.5.3 ทีซีพี/ไอพี	17
2.5.3.1 ทีซีพี/ไอพี คืออะไร	17
2.5.3.2 ส่วนประกอบของ ทีซีพี/ไอพี	18
2.5.3.3 โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี	20
2.5.3.4 ข้อแตกต่างระหว่างชุดโพรโตคอล ทีซีพี/ไอพี	21
2.5.3.5 ลักษณะของการติดต่อ	22
2.5.3.6 ความสัมพันธ์ของ TCP และ UDP กับอินเตอร์เน็ต	22
2.5.3.7 พอร์ตและซ็อกเก็ต	23
2.5.3.8 ข้อกำหนดรูปแบบไอพี	24
2.5.3.9 ข้อกำหนดรูปแบบของ ICMP	25
2.5.4 ไอพีแอดเดรส	25
2.5.5 การติดต่ออินเตอร์เน็ตของวิชวลเบสิกโดยใช้ MS Winsock Control	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.6 โปรแกรมวิชวลเบสิก	33
2.6.1 การเปิดโปรแกรมขึ้นมาใช้งาน	33
2.6.2 คุณสมบัติและข้อดีของ Visual Basic	35
2.6.3 ส่วนประกอบต่างๆที่ใช้ในการทำงาน	36
2.6.4 คอนโทรลของ VB/Win (Custom Control)	41
2.6.5 ภาษาโปรแกรมของ Visual Basic	42
บทที่ 3 หลักการออกแบบ	50
3.1 วงจรถอดรหัส	50
3.2 การใช้ทรานซิสเตอร์ขับกระแส	54
3.3 ภาคแสดงผล	54
3.4 การออกแบบโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็น Client/Server	56
3.4.1 การออกแบบโปรแกรม Client	57
3.4.2 การออกแบบโปรแกรม Server	58
3.4.3 การทำงานในส่วนการส่งข้อมูลไร้สาย (Wireless)	59
บทที่ 4 ผลการทดลอง	61
4.1 การทำงานของโปรแกรม Client/Server	61
4.2 ผลการทำงานของไมโครคอนโทรลเลอร์	63
4.2.1 สัญญาณที่ได้จากไอซีเบอร์ 74HC154	63
4.2.2 สัญญาณนาฬิกาและข้อมูลที่ป้อนให้กับไอซีเบอร์ 74HC164	65
4.2.3 การคิดค่าความต้านทานที่ต่อกับทรานซิสเตอร์	65
บทที่ 5 สรุปผลการทดลอง	68
5.1 ปัญหาในการเชื่อมต่อกันระหว่าง Client กับ Server	68
5.2 ปัญหาของการส่งข้อมูลระหว่าง Computer (Server) กับ Microcontroller	69
5.3 ปัญหาในส่วนการรับและส่งข้อมูลระหว่างเครื่องรับและส่ง	70
5.4 ปัญหาในส่วนของแหล่งจ่ายไฟ (Supply)	70
ภาคผนวก	71
ภาคผนวก ก	72
Flowchart การทำงานของ Client	73
Flowchart การทำงานของ Server	74
Flowchart การทำงานของโปรแกรม (Microcontroller)	75
Flowchart ในส่วนของโปรแกรมอินเตอร์รัพท์	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
ภาคผนวก ข	77
วงจรแหล่งจ่ายไฟและขาสัญญาณที่ต่อเข้ากับภาคส่ง	78
วงจรควบคุมการทำงานของไมโครคอนโทรลเลอร์	79
วงจรส่วนของชุดควบคุมการสแกนของภาคแสดงผล	80
โปรแกรมภาษาวิชวลเบสิก	81
กิตติกรรมประกาศ	87
หนังสืออ้างอิง	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

เรื่อง	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมแสดงการทำงานโดยรวมของระบบ	2
รูปที่ 2.1 แสดงสัญลักษณ์ของ LED ชนิด 2 สี	4
รูปที่ 2.2 แสดงภาพของ dot matrix ขนาด 8×8 dot	4
รูปที่ 2.3 แสดงขาต่างๆ ของ MCS-51 เบอร์ AT89C52	6
รูปที่ 2.4 วงจรในส่วนขับกระแสของบอร์ดแสดงผล	11
รูปที่ 2.5 แสดงเครือข่ายอินเทอร์เน็ต	13
รูปที่ 2.6 แสดงคลาสของไอพีแอดเดรส	26
รูปที่ 2.7 ขั้นตอนการเปิดโปรแกรม Microsoft Visual Basic 6	34
รูปที่ 2.8 เมื่อเริ่มเปิดโปรแกรม Microsoft Visual Basic 6.0	35
รูปที่ 2.9 ส่วนประกอบต่าง ๆ ของโปรแกรม Visual Basic 6.0	36
รูปที่ 2.10 แสดงขอบเขตของการใช้ตัวแปรและค่าคงที่ในระดับต่าง ๆ	45
รูปที่ 3.1 แสดงลักษณะขาสัญญาณของไอซีเบอร์ 74HC154	50
รูปที่ 3.2 แสดงการต่อใช้งานของ 74HC154 กับ 89C52	51
รูปที่ 3.3 แสดงลักษณะขาสัญญาณของไอซีเบอร์ 74HC164	52
รูปที่ 3.4 แสดงการต่อใช้งานของ 74HC164 2 ตัว กับ 89C52	53
รูปที่ 3.5 วงจรภายในของ TR BD 678	54
รูปที่ 3.6 แสดงการต่อ DOT MATRIX ขนาด 16×64 dot	55
รูปที่ 3.7 แสดงการต่อ LED ภายใน 1 MODULE	56
รูปที่ 3.8 แสดงหน้าจอโปรแกรม Client	57
รูปที่ 3.9 แสดงหน้าจอโปรแกรมที่ตรวจดูหมายเลข IP Address	58
รูปที่ 3.10 แสดงหน้าจอโปรแกรม Server	58
รูปที่ 4.1 แสดงการทำงานเมื่อรันโปรแกรม	61
รูปที่ 4.2 แสดงการเชื่อมต่อระหว่าง Client และ Server	62
รูปที่ 4.3 แสดงการส่งข้อความให้กับ Server	62
รูปที่ 4.4 สัญญาณที่ได้จากขาเอาท์พุทไอซีเบอร์ 74HC154	63
รูปที่ 4.5 สัญญาณในการสแกนแต่ละแถว	64
รูปที่ 4.6 สัญญาณนาฬิกาและสัญญาณข้อมูลที่ป้อนให้กับไอซีเบอร์ 74HC164	65
รูปที่ 4.7 แสดงค่าตัวแปรต่างๆ ในวงจร	66
รูปที่ 5.1 แสดงการเชื่อมต่อระหว่าง Client / Server กรณีป้อน IP Address ไม่ถูกต้อง	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เรื่อง		หน้า
ตารางที่ 2.1	แสดงไมโครโปรเซสเซอร์ตระกูล MCS-51	6
ตารางที่ 2.2	แสดงบิท และหน้าที่ต่างๆของพอร์ท 3	7
ตารางที่ 2.5.1	โอ เอส ไอ โมเดล	15
ตารางที่ 2.5.2	โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี เปรียบเทียบกับโอ เอส ไอ	20
ตารางที่ 2.5.3	หมายเลขพอร์ตที่ให้บริการต่างๆ	24
ตารางที่ 2.5.4	คุณสมบัติวินซ็อกคอนโทรล	29
ตารางที่ 2.5.5	คุณลักษณะสถานะของวินซ็อกคอนโทรล	30
ตารางที่ 2.5.6	Methods ของวินซ็อกคอนโทรล	31
ตารางที่ 2.5.7	Event ของวินซ็อก	32
ตารางที่ 2.6.1	ตัวแปรและค่าคงที่ใน โปรแกรม Visual Basic	43
ตารางที่ 2.6.2	แสดงกลุ่มของโอเปอร์เรเตอร์	46
ตารางที่ 3.1	การต่อใช้งานของ IC เบอร์ 74HC154 กับ ไมโครคอนโทรลเลอร์	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันเทคโนโลยีต่างๆได้เข้ามามีบทบาทเป็นอย่างมากในชีวิตประจำวัน ซึ่งจะเห็นว่าเทคโนโลยีในด้านของข่าวสารข้อมูล ได้ก้าวหน้าไปอย่างมาก และได้มีการนำเทคโนโลยีนี้ไปใช้ในด้านการโฆษณากันมากขึ้น ดังนั้นสื่อที่สามารถแสดงข้อมูลข่าวสารได้ จึงมีความจำเป็นมากขึ้น ซึ่งเป็นการสื่อสารที่สามารถเข้าใจได้ง่ายในข้อมูลที่แสดงออกมา

การสื่อสารด้วยแผงแสดงผลโดย LED จึงเป็นอีกสื่อหนึ่งที่ใช้ในการประชาสัมพันธ์และ ให้ข่าวสารที่ได้รับความนิยมอย่างมาก เพราะสามารถเข้าใจได้ง่าย ซึ่งสามารถพบเห็นได้ตามสถานที่ต่างๆ ไม่ว่าจะเป็นแหล่งชุมชน ย่านธุรกิจ หรือแม้แต่ตามสถานที่ราชการ เช่น โรงพยาบาล หรือ สถานที่ชำระค่าไฟฟ้า หรือ ค่าโทรศัพท์

โครงงานนี้ จัดทำขึ้นโดยได้นำความรู้ที่ได้เรียนมา นำมาศึกษาและออกแบบ เพื่อสร้างแผงแสดงผลโดย LED โดยโครงงานแผงแสดงผลโดย LED นี้เป็นการประยุกต์ใช้งานไมโครคอนโทรลเลอร์เพื่อนำข้อมูลออกแสดงผล โดยใช้ dot matrix ขนาด 8×8 dot นำมาใช้ เป็นตัวแสดงผล โดยจะแสดงผลได้ 1 ระดับสี สามารถแสดงตัวอักษรภาษาอังกฤษได้ ซึ่งการทำงานทั้งหมดอาศัยการใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 89C52 เป็นตัวควบคุมการทำงานโดยรวมของระบบทั้งหมด

โครงงานนี้จะเป็นแผงแสดงผลโดย LED ซึ่งจะใช้ dot matrix ขนาด 8×8 dot เป็นตัวแสดงผล โดยใช้ทั้งหมด 16 ตัว นำมาต่อกัน ซึ่งรวมแล้วจะเป็นแผงแสดงผลขนาด 16×64 dot

ส่วนเนื้อหาของโครงงานจะกล่าวถึง บล็อกไดอะแกรมการทำงานโดยรวมของระบบ ทฤษฎีที่เกี่ยวข้อง การทำงานในส่วนต่างๆ การออกแบบฮาร์ดแวร์ และซอฟต์แวร์ รวมถึงการทดลองในส่วนต่างๆของโครงงานนี้

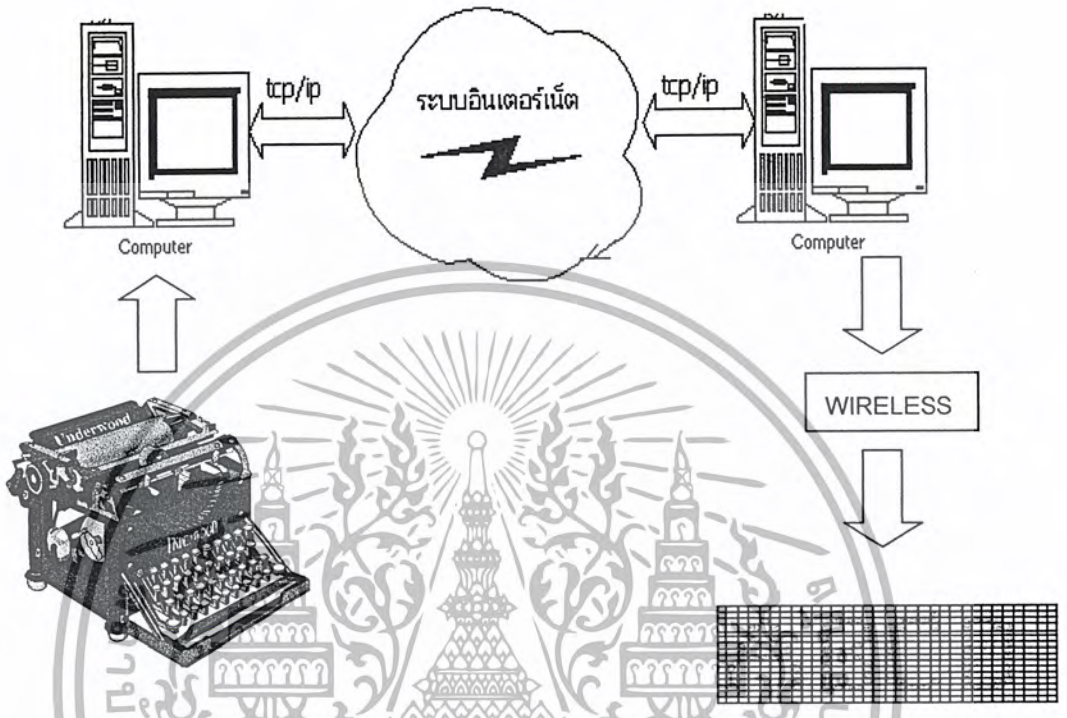
1.1 วัตถุประสงค์

โครงงานนี้จัดทำขึ้น เพื่อศึกษาการทำงานของ **Client/Server** ผ่านเครือข่ายอินเทอร์เน็ต การสื่อสารไร้สาย (**Wireless**) ส่วนของการควบคุมไมโครคอนโทรลเลอร์ , วงจรสแกนทางด้านแถว , วงจรส่งข้อมูลทางด้านหลัก และ อุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการทำแผงแสดงผล โดยนำ dot matrix ขนาด 8×8 dot มาทำแผงแสดงผล และเพื่อเป็นแนวทางในการพัฒนา Application บนระบบเครือข่าย Internet สู่เครือข่ายอื่นๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 การทำงานโดยรวมของระบบ

การทำงานของระบบแสดงได้ดังรูปที่ 1.1



รูปที่ 1.1 แสดงการทำงานโดยรวมของระบบ

จากบล็อกไดอะแกรม สามารถอธิบายการทำงานได้ดังนี้

1. **ส่วนควบคุม (Client)** จะประกอบไปด้วยคอมพิวเตอร์ที่เชื่อมต่อเข้ากับเครือข่ายอินเทอร์เน็ตและรวมเอาโปรแกรมควบคุมการทำงานในส่วนของโครงงานนี้เอาไว้

2. **ส่วนประมวลผล (Server)** จะเป็นคอมพิวเตอร์ที่ทำหน้าที่เป็น server สำหรับติดต่อกับส่วนควบคุม โดยผ่านเครือข่ายอินเทอร์เน็ต และนำผลของการประมวลผลส่งไปยังส่วนควบคุมภาคแสดงผล

3. **ควบคุมภาคแสดงผล (Microcontroller)** ทำหน้าที่ควบคุมการทำงานทั้งหมด โดยจะควบคุมการทำงานของ , วงจรสแกนทางด้านแถว ,วงจรส่งข้อมูลทางด้านหลัก และควบคุมการทำงานของแผงแสดงผล LED รวมไปถึง RAM ที่ใช้ในการเก็บรูปแบบของตัวอักษร

4. **ภาคแสดงผล (Display Board)** ทำหน้าที่แสดงผลการติดดับของ LED และการแสดงแบบตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

โครงงานนี้จะประกอบด้วยส่วนสำคัญ คือ ชุดควบคุมไมโครคอนโทรลเลอร์ , วงจรสแกนทางด้านแถว ,วงจรส่งข้อมูลทางด้านหลัก และ ภาคแสดงผล ดังนั้นจึงต้องทำความเข้าใจเกี่ยวกับคุณสมบัติและการทำงานของส่วนประกอบต่างๆ ที่ได้กล่าวมาแล้วข้างต้น

การแสดงผลแบบเมตริกเพื่อให้เป็นตัวอักษรนั้น เกิดขึ้นจากการกำหนดจุดแสดงผลหรือที่เรียกว่า " พิกเซล " ซึ่งสามารถควบคุมได้โดยการเขียนโปรแกรม

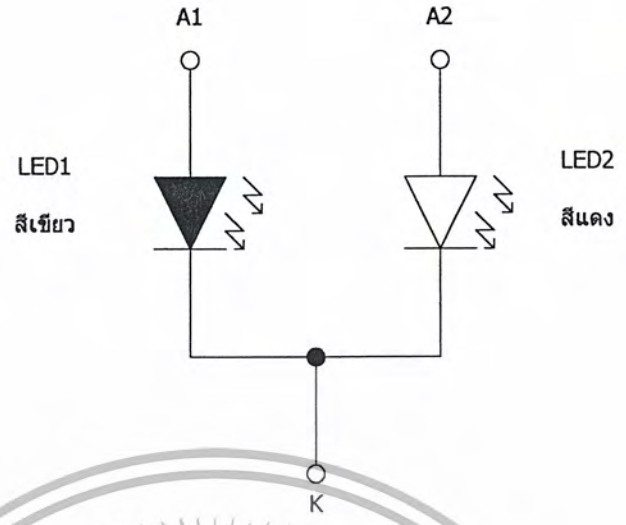
2.1 คุณสมบัติของหลอดไดโอดนำแสง (LED)

LED เป็นไดโอดชนิดหนึ่งซึ่งสามารถเปล่งแสงได้ โดยโครงสร้างของ LED มีลักษณะเป็นสารกึ่งตัวนำชนิดพี (P-Type) และชนิดเอ็น (N-Type) ต่อกันเป็นรอยต่อ พี-เอ็น (P-N Junction) LED จะเปล่งแสงออกมาก็ต่อเมื่อมีการจ่ายกระแสไบอัสตรง (Forward Bias) ให้กับมัน กระแสไบอัสตรงนี้จะไปกระตุ้นอิเล็กตรอน (Electron) และโฮล (Hole) ข้ามรอยต่อ พี-เอ็น เพื่อมารวมตัวกัน ในการรวมตัวกันจะมีการปลดปล่อยพลังงานออกมาในรูปของโฟตอน (Photon) ซึ่งเป็นอนุภาคของแสง ซึ่งต่างจากอุปกรณ์อย่างอื่นๆที่ปล่อยพลังงานออกมาในรูปของความร้อน สำหรับสารกึ่งตัวนำที่นิยมนำมาสร้างเป็น LED จะใช้แกลเลี่ยมอาร์เซไนด์ฟอสไฟต์ (Galliumarsenidephosphide : GaAsp) ซึ่งสารชนิดนี้จะใช้กระแสไฟฟ้าไม่มากในการไบอัส เพื่อให้เกิดการปล่อยโฟตอนออกมา การให้แสงของ LED โดยการจ่ายกระแสไฟฟ้า เรียกว่า อิเล็กโตรลูมีเนเซนส์ (Electroluminesence)

คุณสมบัติที่สำคัญของ LED คือ

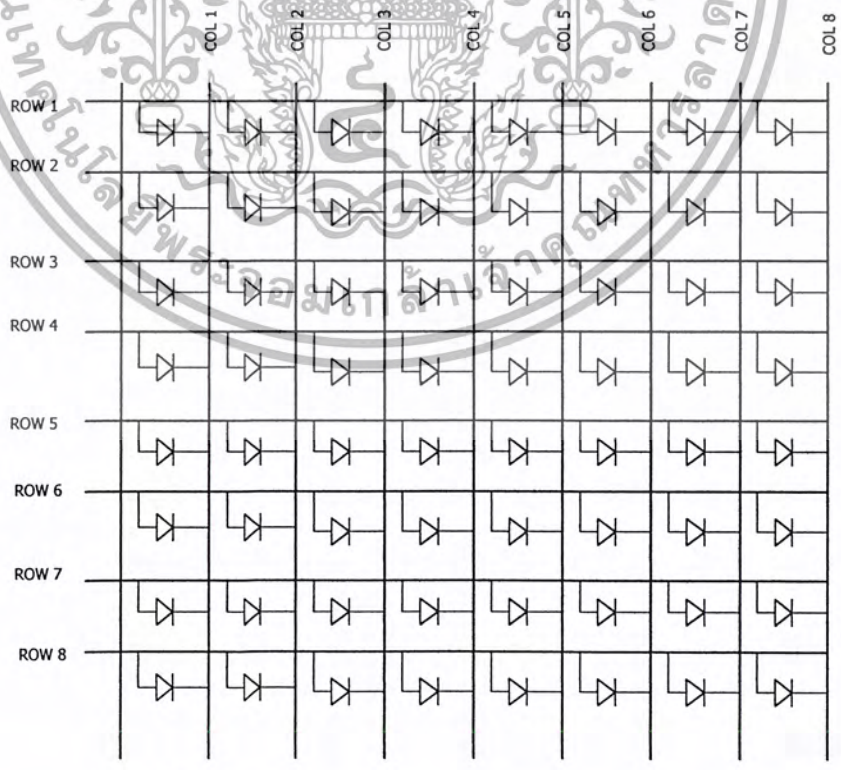
- เป็นแหล่งกำเนิดแสงขนาดเล็กแต่มีความทนทานสูง
- ใช้แรงดันที่มาไบอัส LED ต่ำ
- มีกำลังส่องสว่างแปรผันตามกระแสที่ไหลผ่าน ดังนั้นจึงสามารถส่งข้อมูลในรูปของกระแสจากจุดหนึ่งที่ห่างไกลออกไปโดยใช้แสงได้
- ทนทานต่อการกระทบกระเทือน
- แสงที่เปล่งออกไปมีช่วงความยาวคลื่นแคบ
- สามารถผลิตให้มีขนาดเล็กได้
- มีหลายสี
- ราคาถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงสัญลักษณ์ของ LED ชนิด 2 สี

จากที่ได้กล่าวไปแล้ว แผงแสดงผลที่ใช้ในโครงงานนี้จะใช้ dot matrix ขนาด 8×8 dot นำมาต่อกันทั้งหมด 16 ตัว ซึ่งก็จะได้ขนาดแผงแสดงผลเป็น 16×64 dot สาเหตุที่ใช้ dot matrix เพราะว่า dot matrix มีความทนทานกว่า LED และมีประสิทธิภาพมากกว่า



รูปที่ 2.2 แสดงภาพของ dot matrix ขนาด 8×8 dot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายใน บางเบอร์มีหน่วยความจำภายในแบบ ROM บางเบอร์เป็นแบบ EPROM บางเบอร์มี RAM ภายใน 128 ไบต์ เป็นต้น ซึ่งรายละเอียดจะศึกษาได้จากคู่มือการใช้โดยตรง และลักษณะของขาต่างๆ จะเหมือนกันคุณลักษณะพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ที่จัดอยู่ภายในตระกูล MCS-51 นี้ ประกอบด้วย

- หน่วยประมวลผลกลางขนาด 8 บิท
- หน่วยประมวลผลสำหรับข้อมูลแบบบิท (Boolean Processor)
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ EPROM (เบอร์ 8751) หรือแบบ ROM (เบอร์ 8051)
- หน่วยความจำแบบ RAM ภายในจำนวน 128 ไบต์
- พอร์ทอินพุต/เอาท์พุต แบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิท จำนวน 2 วงจร
- วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)
- วงจรควบคุมการอินเตอร์รัพท์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้ 2 ระดับ
- วงจรออสซิลเลเตอร์ภายใน
- หนึ่งวัฏจักรคำสั่งกินเวลาประมาณ 1 μsec ขณะทำงานด้วย clock 12 MHz

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซี เป็นแบบ DIP ขนาด 40 ขา ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่า 1 อย่าง ซึ่งจะไม่สามารถใช้ งานในเวลาเดียวกันได้ ตัวอย่างเช่น ขาสัญญาณบิท 0 ของพอร์ท 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณ เอาท์พุต/อินพุต ตามปกติหรืออาจจะทำหน้าที่เป็นขาสัญญาณอินพุตของข้อมูลสื่อสารแบบอนุกรม (RXD) ให้กับวงจรสื่อสารแบบอนุกรมของ 8051 ได้ ซึ่งการกำหนดว่าจะใช้งานในลักษณะใด ก็ขึ้นอยู่กับการเชื่อมต่อวงจรเข้ากับสัญญาณและโปรแกรมควบคุมของระบบนั้น

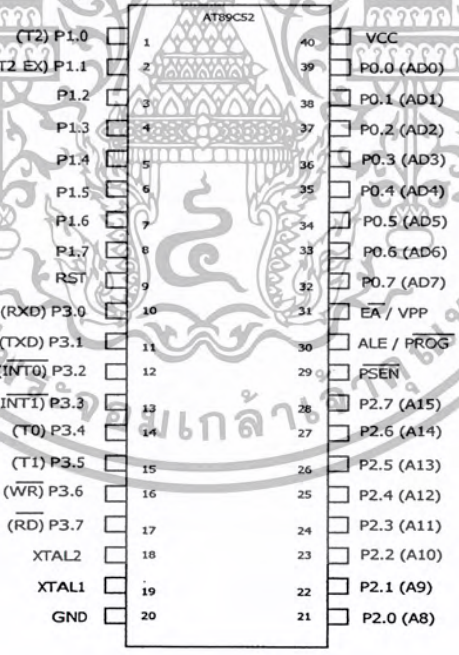
ตัวอย่างไมโครคอนโทรลเลอร์ตระกูล MCS-51 และลักษณะต่างๆ สามารถแสดงได้ในตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงไมโครโปรเซสเซอร์ตระกูล MCS-51

เบอร์	หน่วยความจำโปรแกรมบนชิพ	หน่วยความจำข้อมูลบนชิพ	TIMERS
8051	4 K ROM	128 byte	2
8031	-	128 byte	2
8751	4 K EPROM	128 byte	2
8052	8 K ROM	256 byte	3
8032	-	256 byte	3
8752	8 K EPROM	256 byte	3

การจัดขาต่างๆ ของ MCS-51 ใน ไอซีไมโครคอนโทรลเลอร์ 8051 โครงสร้าง IC เป็นแบบ DIP มีขาทั้งหมด 40 ขา โดยขาต่างๆ จะใช้เป็นขาอินพุต , เอาท์พุต , ขาสัญญาณควบคุม , ขาตำแหน่งหน่วยความจำและขาข้อมูลดังรูปที่ 2.3



รูปที่ 2.3 แสดงขาต่างๆ ของ MCS-51 เบอร์ AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานและหน้าที่ของขาต่างๆเป็นดังต่อไปนี้

พอร์ต 0 (ขา 39 ~ 32) P0.0 ~ P0.7 เป็นพอร์ตที่ใช้งาน 2 หน้าที่ หน้าที่แรกใช้เป็นอินพุต/เอาท์พุตพอร์ต ส่วนอีกหน้าที่หนึ่งใช้ควบคุมหน่วยความจำภายนอก หมายถึง เมื่อต้องการขยายระบบให้ใหญ่ขึ้น โดยต่อหน่วยความจำภายนอกใช้งาน ในกรณีนี้ MCS-51 จะให้สัญญาณพอร์ต 0 ออกมาเป็นสัญญาณมัลติเพลกซ์ ระหว่างบัสตำแหน่งกับบัสข้อมูลบิท 0 ถึงบิท 7 (AD0 ~ AD7)

พอร์ต 1 (ขา1 ~ 8) ใช้เป็นอินพุต/เอาท์พุตพอร์ตอย่างเดียว ใช้สัญลักษณ์เรียงกันเป็น P1.0 , P1.1 จนถึง P1.7 พอร์ตนี้ใช้เชื่อมต่อกับอุปกรณ์ภายนอกหน้าที่เดียวเท่านั้น ยกเว้น 8032/8052 ที่ใช้ P1.0 และ P1.1 เป็นอินพุต/เอาต์พุต หรืออินพุทที่รับจากภายนอกของไทเมอร์ชุดที่สาม

พอร์ต 2 (ขา21 ~ 28) เป็นพอร์ตที่ใช้งาน 2 หน้าที่เช่นกันหน้าที่แรกเป็นอินพุต/เอาต์พุตพอร์ต หน้าที่ที่ 2 ใช้ควบคุมหน่วยความจำภายนอก โดยใช้สัญญาณแอดเดรสด้านไบต์สูง (A8 ~ A15) ออกมา

พอร์ต 3 (ขา10 ~ 17) เป็นพอร์ตที่ใช้งาน 2 หน้าที่เช่นกันหน้าที่แรกเป็นอินพุต/เอาต์พุตพอร์ต หน้าที่ที่ 2 แยกออกได้ หลายฟังก์ชัน ตามตารางที่ 2.2

ตารางที่ 2.2 แสดงบิท และหน้าที่ต่างๆของพอร์ท 3

บิท	ชื่อ	ทำหน้าที่
P3.0	RXD	รับข้อมูลสำหรับพอร์ตอนุกรม
P3.1	TXD	ส่งข้อมูลสำหรับพอร์ตอนุกรม
P3.2	INT0	อินเตอร์รัพต์ภายนอกหมายเลข 0
P3.3	INT1	อินเตอร์รัพต์ภายนอกหมายเลข 1
P3.4	T0	เคาน์เตอร์ 0 (รับอินพุต จากภายนอก)
P3.5	T1	เคาน์เตอร์ 1 (รับอินพุต จากภายนอก)
P3.6	WR	สัญญาณเขียนใช้ต่อกับหน่วยความจำภายนอก
P3.7	RD	สัญญาณอ่านใช้ต่อกับหน่วยความจำภายนอก

PSEN (program store enable) (ขา 29) เป็นสัญญาณเอาท์พุต ใช้เป็นสัญญาณควบคุมการอีนาเบิล (enable) หน่วยความจำที่ใช้เก็บรหัส หรือ โปรแกรมที่ต่ออยู่ภายนอก (external program memory) โดยทั่วไปจะต่อกับขาเอาท์พุตอีนาเบิล (output enable : OE) ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชิพรอมหรืออีพรอมขา PSEN จะให้ลอจิก 0ในขณะที่อยู่ในขั้นตอนเฟตซ์คำสั่งจากรอมหรืออีพรอมที่ต่ออยู่ภายนอก ข้อมูลจะถูกวางในบัสข้อมูล และถูกเก็บไว้ในรีจิสเตอร์คำสั่งภายในตัว MCS-51 เพื่อรอรับการถอดรหัสต่อไป เมื่อทำการรันโปรแกรมจากรอมภายในตัว (8051/8052) สัญญาณ PSEN จะมีระดับสัญญาณเป็นลอจิก 1 ตลอด

ALE (address latch enable) (ขา 30) เป็นสัญญาณเอาท์พุต ใช้เป็นสัญญาณควบคุมการแลตช์ (latch) สัญญาณตำแหน่งที่ออกมาจากพอร์ต 0 เมื่อพอร์ต 0 ถูกใช้งานในหน้าที่ที่ สอง (เป็นสัญญาณตำแหน่งและข้อมูลมัลติเพลกซ์กันออกมา) ความถี่ของสัญญาณ ALE จะมีค่าเป็น 1/6 เท่าของความถี่ที่เกิดขึ้นภายในชิพ เช่น ถ้าใช้คริสตอลความถี่ 12 MHz สัญญาณ ALE จะมีความถี่เป็น 2 MHz นั่นคือ ALE จะเป็น HIGH 2 ครั้งใน 1 แมชชีนไซเคิล ยกเว้นการทำคำสั่ง MOVX ซึ่งใช้เวลา 2 แมชชีนไซเคิล สัญญาณ ALE จะหายไป 1 ลูก ในแมชชีนไซเคิลที่ 2 ของคำสั่งนี้

EA (external access) (ขา 31) ขา EA เป็นขาอินพุต เพื่อใช้ต่อกับ Vcc หรือ กราวด์ ถ้าต่อกับ Vcc 8051/8052 จะรันโปรแกรมจากรอมภายในก่อน ในช่วงหน่วยความจำ 4K/8K แรก จากนั้นจึงออกมารันที่หน่วยความจำภายนอก ถ้าต่อกับกราวด์ 8051/8052 จะรันโปรแกรมจากหน่วยความจำภายนอกทั้งหมด เริ่มจากตำแหน่งแรก ถ้าใช้ 8031/8032 ขา EA จะต้องต่อลงกราวด์ เนื่องจากตัวมันไม่มีหน่วยความจำรอมภายใน

RST (reset) (ขา 9) เป็นขาอินพุตใช้รีเซ็ต 8051 เมื่อสัญญาณที่ขา RST เป็นลอจิก 1 (อย่างน้อย 2 วงรอบแมชชีน) ทำให้ MCS-51 ทำการรีเซ็ตระบบภายในของมันทั้งหมด

ออสซิลเลเตอร์อินพุต (oscillator input , XTAL1 ~ XTAL2) (ขา18 ~ 19) MCS-51 มีวงจรออสซิลเลเตอร์อยู่ภายใน โดยเชื่อมต่อกับขาคริสตอลผ่านทางขา 18 และ 19 ในตระกูล MCS-51 สามารถใช้คริสตอลที่มีความถี่สูงถึง 12 MHz บางเบอร์ เช่น 8031 BH-1 สามารถใช้คริสตอลที่มีความถี่สูงถึง 16 MHz

แหล่งจ่ายไฟ (ขา 40 และ 20) 8051ใช้แหล่งจ่ายไฟแหล่งเดี่ยวขนาด +5 โวลท์ โดย Vcc ต่อกับขา 40 และกราวด์ต่อกับขา 20

2.3 การสแกน (Scanning)

ตัวอักษรต่างๆที่เห็นในแผงแสดงผลนั้น จะประกอบไปด้วยจุดเล็กๆจำนวนหนึ่ง ที่ทำให้เกิดขึ้นโดยเส้นแนวนอนและแนวตั้งโดยจะเป็นส่วนๆ ส่วนละเท่าๆกันจุดเหล่านี้ก็คือ LED ดวงหนึ่งนั่นเอง การที่เราจะมองเห็นว่าเป็นตัวอักษรได้ ก็คือ การควบคุมให้ LED สว่างติดตามความต้องการ แต่ไม่สามารถทำให้ LED สว่างติดตลอดเวลาได้ ดังนั้นจึงต้องใช้เทคนิคในการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาอย่างหนึ่ง คือเราจำเป็นต้องควบคุมการติดดับของ LED ให้สามารถมองเห็นเป็นตัวอักษรนิ่งหรือตัวอักษรที่เคลื่อนไหวได้ โดยปกติแล้ว LED ที่ใช้กันอยู่จะมีความเข้มของการส่องสว่างที่ระดับ มิลลิดาเดลลา (mcd) ซึ่งเป็นหน่วยการวัดความเข้มของการส่องสว่าง การมองเห็นตัวอักษรซึ่งเป็นแสงต่อเนื่องได้นั้น จะต้องป้อนกระแสพัลส์ที่มีความถี่มากกว่า 30 Hz ตาคนเราจึงจะมองเห็นแสงเอาท์พุต LED เป็นแสงต่อเนื่องแต่ถ้าเราต้องการให้เห็นเป็นตัวอักษรวิ่งที่ไม่สามารถสังเกตเห็นการกระพริบได้เลย เราควรป้อนพัลส์ที่มีความถี่ประมาณ 50 – 60 Hz จึงจะไม่เห็นตัวอักษรที่เกิดการกระพริบ และหน้าที่ของการสแกนก็คือการเลือก LED ที่ติดสว่างให้ได้ตัวอักษรที่ชัดเจนตามต้องการ

การสแกน หมายถึง จำนวนเส้นการสแกนต่อหนึ่งภาพ และจำนวนภาพที่ส่งออกไปต่อวินาที ถ้าส่งจำนวนต่อวินาทีมากเท่าไร การกระพริบของภาพก็จะลดลงเท่านั้น ซึ่งในการทำบอร์ดแสดงผลมีหลักการอยู่ที่การสแกน ซึ่งการสแกนนี้สามารถทำได้ 2 วิธี คือ การสแกนทางคอลัมน์ (Column) และการสแกนทางโรว์ (Row)

2.3.1 การสแกนทางคอลัมน์ (Column Scan)

การสแกนทางคอลัมน์จะทำการส่งข้อมูลออกไปทางโรว์ โดยส่งข้อมูลตัวที่ 1 ออกไปแล้วให้คอลัมน์ที่ 1 แอคทีฟ (Active) จากนั้นก็ทำการส่งข้อมูลตัวที่ 2 ออกไปแล้วให้คอลัมน์ที่ 2 แอคทีฟ ทำเช่นนี้ไปจนกระทั่งข้อมูลถูกส่งออกไปจนครบหมดทุกคอลัมน์ก็จะเป็นการสแกนครบ 1 รอบ ดังนั้นถ้าจำนวนหลักที่จะแสดงผลออกมาเป็นตัวอักษรมีจำนวนหลายหลัก วิธีนี้จึงไม่เหมาะสมที่จะนำมาใช้งาน เพราะว่าเมื่อให้ LED ในคอลัมน์ที่ 1 ติด กว่าที่ LED ที่คอลัมน์สุดท้ายจะติดต้องใช้เวลานาน

2.3.2 การสแกนทางโรว์ (Row Scan)

การสแกนทางโรว์ จะทำการส่งข้อมูลออกไปจนครบทุกหลักก่อน แล้วให้ โรว์ที่ 1 แอคทีฟ จากนั้นก็ทำการส่งข้อมูลชุดถัดไปออกไปจนครบหมดทุกหลัก แล้วให้ โรว์ที่ 2 แอคทีฟทำเช่นนี้ จนกระทั่งข้อมูลถูกส่งออกไปจนครบหมดทุกโรว์ ก็จะเป็นการสแกนครบ 1 รอบ วิธีนี้สามารถแสดงผลเป็นตัวอักษรพร้อมกันได้หลายหลัก และถ้าจัดเวลาให้เหมาะสมแล้ว เมื่อเวลาทำการสแกนจะไม่เกิดการพริ้ว

2.4 วงจรขับกระแส (Current Driver)

หน้าที่ของวงจรขับกระแส คือ ช่วยในการจ่ายกระแสให้กับ LED ซึ่ง LED จะมีการทำงานคล้ายไดโอดทั่วๆไปโดยสามารถนำกระแสได้ถ้ามีการป้อนแรงดันในลักษณะ Forward Bias แต่แตกต่างจากไดโอดทั่วๆไป คือ LED จะสามารถให้แสงสว่างในขณะที่นำกระแสอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_{B1} = \frac{V_{CC} - V_{EB1} - V_{IN1}}{I_{B1}} \quad (2.1)$$

$$R_{B2} = \frac{V_{IN2} - V_{BE2}}{I_{B2}} \quad (2.2)$$

$$R_C = \frac{V_{CC} - V_{EC1} - V_{CE2} - V_D}{I} \quad (2.3)$$

วงจรขับกระแสโดยที่ใช้ทรานซิสเตอร์นั้นจะทำหน้าที่เป็นเหมือนสวิทช์ คือ ขณะที่ทรานซิสเตอร์ทำงาน กระแสคอลเลกเตอร์จะสามารถไหลลงกราวด์ได้ ซึ่งจะทำให้ LED สว่าง และขณะที่ทรานซิสเตอร์ไม่ทำงาน จะไม่มีกระแสคอลเลกเตอร์ไหลผ่าน LED ได้ ทำให้ LED ไม่สว่าง ส่วนค่าความต้านทาน (Resister) ที่ขาเบสมีไว้เพื่อป้องกันแรงดันที่ตกคร่อม V_{BE} มากเกิน ไปอาจทำให้เกิดความเสียหายต่อทรานซิสเตอร์ โดยค่าความต้านทานนี้จะเป็นเท่าไรขึ้นอยู่กับ กระแสเบสขณะที่ทรานซิสเตอร์กำลังทำงาน ส่วนตัวต้านทานอีกตัวที่ต่ออนุกรมกับ LED ไว้ สำหรับจำกัดกระแสที่ไหลผ่าน LED โดยคำนวณหาค่าความต้านทานได้จากสูตรที่ได้กล่าวไว้ข้างต้น

หมายเหตุ :

V_{CC} คือ แรงดันของแหล่งจ่าย

V_{EC1} คือ แรงดันตกคร่อมระหว่างขาอิมิตเตอร์ กับ ขาคอลเลกเตอร์ใน ขณะที่ ทรานซิสเตอร์แบบคาร์ลิงตันทำงาน (มีค่าประมาณ 1 โวลต์)

V_{EB1} คือ แรงดันตกคร่อมระหว่างขาอิมิตเตอร์ กับขาเบส ของ ทรานซิสเตอร์แบบดาร์ลิงตัน (มีค่าประมาณ 1.4 โวลต์)

V_{CE2} คือ แรงดันตกคร่อมระหว่างขาคอลเลกเตอร์ กับขาอิมิตเตอร์ใน ขณะที่ทรานซิสเตอร์ทำงาน (มีค่าประมาณ 0.2 โวลต์)

V_{BE2} คือ แรงดันตกคร่อมระหว่างขาเบส กับขาอิมิตเตอร์ (มีค่าประมาณ 0.7 โวลต์)

V_D คือ แรงดันตกคร่อม LED ขณะนำกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V_{IN1} ,V_{IN2} คือ แรงดันอินพุต.

I_{B1} ,I_{B2} คือ กระแสที่ไหลผ่านขาเบสของทรานซิสเตอร์

I คือ กระแสที่ไหลผ่าน LED

รูปที่ 2.4 วงจรในส่วนขับกระแสของบอร์ดแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนซอฟต์แวร์ (Software part)

2.5 ทฤษฎีระบบเครือข่าย และ วินโดวส์ซ็อกเก็ต

2.5.1 ระบบเครือข่าย (Network System)

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควร ในหัวข้อนี้จะอธิบายหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายซึ่งจะกล่าวดังต่อไป

2.5.1.1 ความหมายของระบบเครือข่ายคอมพิวเตอร์และอินเทอร์เน็ตเวิร์คกิ้ง (Internetworking)

ระบบเครือข่ายคอมพิวเตอร์(Networking) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End System) ซึ่งระบบปลายทางเป็นระบบที่อิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นไปได้ตั้งแต่ ไมโครคอมพิวเตอร์ (Microcomputer) ไปจนกระทั่ง ซูเปอร์คอมพิวเตอร์ (Super computer) ขนาดใหญ่ เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งข้อมูลในการแบ่งปันทรัพยากรของระบบ เช่น ไฟล์ข้อมูล, เครื่องพิมพ์ (Printer) โมเด็ม (Modem) ตลอดจนการให้บริการฐานข้อมูลร่วม (Sharing Database)

อินเทอร์เน็ตเวิร์คกิ้ง หรือ อินเทอร์เน็ต (Internet) คือ การเชื่อมต่อระบบเครือข่าย 2 เครือข่ายขึ้นไป ดังนั้น คอมพิวเตอร์บนระบบเครือข่ายหนึ่งก็สามารถติดต่อกับคอมพิวเตอร์บนระบบเครือข่ายอื่นๆ ได้

2.5.1.2 สถาปัตยกรรมของอินเทอร์เน็ต

คำกล่าวถึงอินเทอร์เน็ตว่าเป็นเครือข่ายที่ใหญ่ที่สุดในโลก เครือข่ายที่มีการต่อเสมือนกับใยแมงมุมครอบคลุมโลก อินเทอร์เน็ตเชื่อมโยงโลกเข้าด้วยกันอย่างไร้มิติ จากคำกล่าวเหล่านี้ทำให้เราควรรู้ว่า อินเทอร์เน็ตจัดการส่วนประกอบต่างๆ อย่างไร จึงทำให้แต่ละส่วนสามารถรับส่งข้อมูลและทำงานสัมพันธ์กันได้เป็นอย่างดี เราจึงจำเป็นต้องศึกษาสถาปัตยกรรมของอินเทอร์เน็ตดังต่อไปนี้

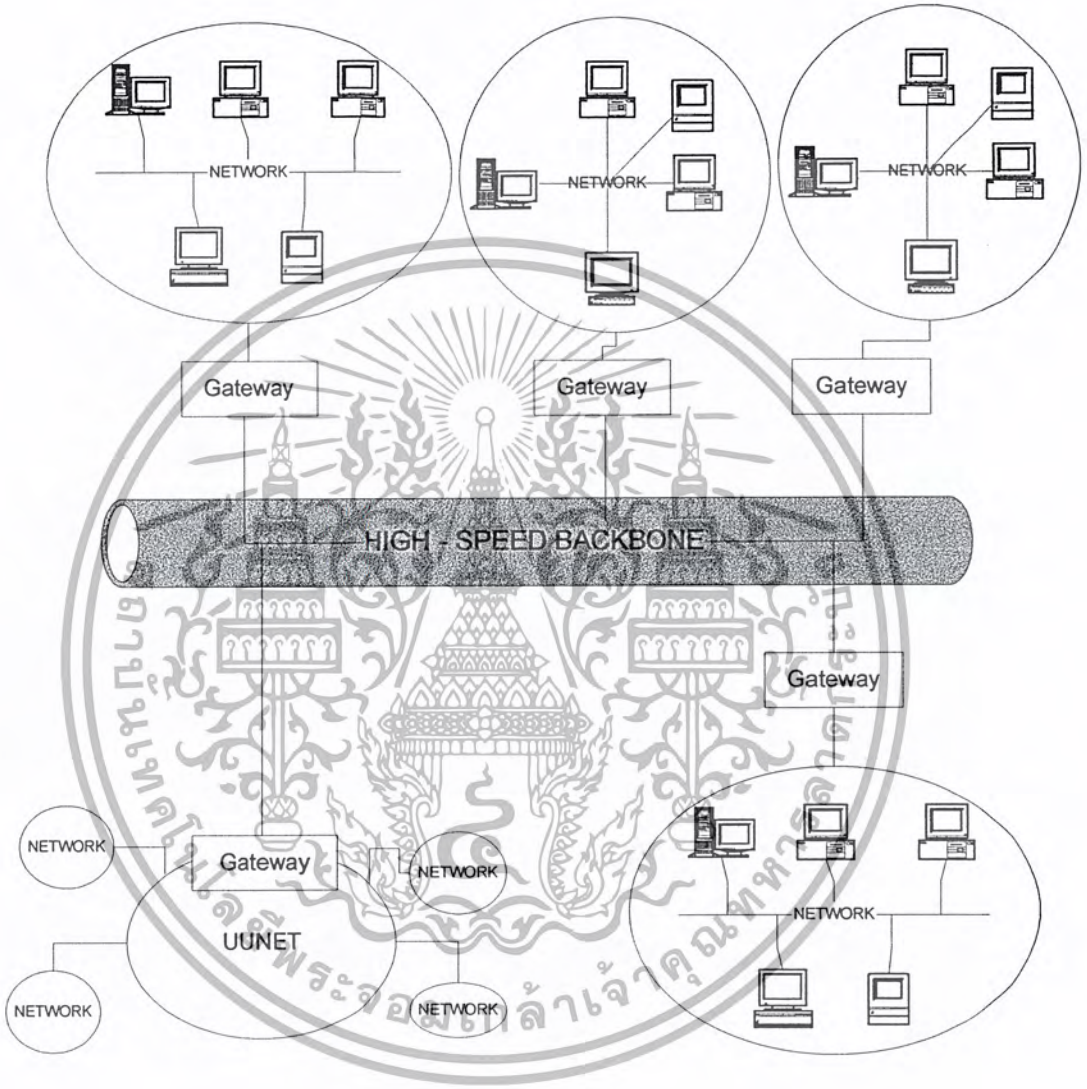
อินเทอร์เน็ตประกอบด้วยสายสื่อสารความเร็วสูงเป็นแบคโบน (Backbone) เป็นสายโทรศัพท์ตามมาตรฐาน TS สามารถวิ่งด้วยความเร็ว 44.736 เมกะบิทต่อวินาที

เครือข่ายที่ต้องการเชื่อมโยงโดยตรงกับอินเทอร์เน็ตจะต่อกับแบคโบน ด้วยอุปกรณ์ที่เรียกว่า "เกตเวย์" (Gateway) ซึ่งเป็นเครื่องคอมพิวเตอร์ที่ทำหน้าที่ควบคุมการผ่านเข้า-ออกของข่าวสารระหว่างเครือข่ายกับแบคโบน เกตเวย์ทุกตัวสามารถกำหนดการติดต่อกับเกตเวย์ตัวอื่น หรือเครือข่ายอื่นได้โดยใช้ไอพีแอดเดรส (IP Address) ของเครือข่ายอ้างอิงถึงกัน เช่น อินเทอร์เน็ตประเทศไทยเป็นเกตเวย์ของเครือข่ายในกลุ่มติดต่อกับเกตเวย์ของ UUNET ที่รัฐเวอร์จิเนียร์ สหรัฐอเมริกา

ข่าวสารจากเครือข่ายถูกส่งออกไปผ่านเกตเวย์เข้าสู่อินเทอร์เน็ต โดยที่เกตเวย์เป็นตัวเลือกทิศทางการเดินทางเพื่อไปยังปลายทางที่ต้องการ แต่ตามเส้นทางอาจจะต้องผ่านเกตเวย์อีกหลายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อรับช่วงส่งข่าวสารจนถึงที่หมาย ถึงแม้ว่าจะต้องเดินทางระยะไกลก็ตาม แต่ด้วยสายสื่อสารความเร็วสูงทำให้การส่งข่าวสารทำได้อย่างรวดเร็ว



รูปที่ 2.5 แสดงเครือข่ายอินเทอร์เน็ต

2.5.1.3 ข้อกำหนดรูปแบบของเกตเวย์ (Gateway Protocols)

เกตเวย์ต้องมีข้อมูลของเกตเวย์ตัวอื่นและรู้จักเครือข่ายปลายทาง เพื่อนำมาใช้ในการกำหนดเส้นทางที่ข่าวสารสามารถเดินทางไปถึงได้เร็วที่สุด เกตเวย์จะมีการแลกเปลี่ยนข้อมูลระหว่างกันเกี่ยวกับเส้นทาง รายละเอียด สถานะเครือข่ายและคุณสมบัติเครือข่ายย่อยที่ติดต่อเข้าสู่เครือข่ายใหญ่ตามลำดับขั้นจึงต้องมีการกำหนดรูปแบบพิเศษสำหรับเกตเวย์ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดรูปแบบเกตเวย์แบ่งออกตามการใช้งานได้ดังนี้

1. IGP(Interior Gateway Protocol) ถูกนำใช้กับเกตเวย์ที่อยู่ในเครือข่ายลูกติดต่อกับเครือข่ายลูกที่อยู่ภายในเครือข่ายแม่เดียวกัน หรือติดต่อกันระหว่างเครือข่ายแม่กับครือข่ายลูก การเชื่อมโยงเกตเวย์ประเภทนี้มักจะไม่ค่อยมีการเปลี่ยนแปลงจึงเรียกว่าระบบอิสระจากกัน (Autonomous) หรือ Selfcomplete

2. EGP (External Gateway Protocol) ในเครือข่ายใหญ่ๆ การติดต่อกับเครือข่ายจะมีการเปลี่ยนแปลงอยู่เสมอ ตามการเปลี่ยนแปลงของเครือข่ายย่อยที่เชื่อมโยงอยู่เป็นจำนวนมากจึงมีข้อกำหนดรูปแบบที่ใช้กับการสื่อสารระหว่างเกตเวย์ของเครือข่ายด้วย EGP

3. GGP (Gateway-to-Gateway Protocol_) การเดินทางของข่าวสารระยะไกลบนแบคโบน อาจจะต้องผ่านเกตเวย์หลายตัวกว่าจะถึงปลายทาง GGP เป็นข้อกำหนดรูปแบบของการสื่อสารระหว่างเกตเวย์บนแบคโบน เพื่อทำให้การจราจรบนแบคโบนไม่ติดขัด ข่าวสารเคลื่อนที่ไปได้รวดเร็ว

2.5.2 โมเดลโอเอสไอ (OSI Model)

เพื่อลดปัญหาในความยุ่งยากสับสนในการติดต่อสื่อสารข้อมูล โครงสร้างของการสื่อสารข้อมูลภายในอุปกรณ์คอมพิวเตอร์ส่วนใหญ่จะถูกแบ่งเป็นชั้นๆ โดยแต่ละชั้นมีอิสระไม่ขึ้นต่อกันทำให้การแปลงบริการชั้นใดๆ ไม่ก่อปัญหากับบริการชั้นอื่นการเพิ่มเติมการบริการใหม่ทำได้โดยไม่ต้องเปลี่ยนแปลงโปรแกรมระบบเดิมและสิ่งที่สำคัญก็คือ การทำระดับชั้นนั้นทำให้ตัวโปรแกรมมีขนาดเล็กสามารถระบุส่วนที่จะต้องปรับปรุงได้แน่นอน ไม่ต้องวิตกกังวลถึงโปรแกรมส่วนอื่น ทำให้การพัฒนาประสิทธิภาพของระบบทำได้ง่าย และดียิ่งขึ้นISO (International Standardization for Organization) ซึ่งเป็นองค์กรที่จัดขึ้นมาเพื่อดูแลและส่งเสริมตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ได้พัฒนาสถาปัตยกรรม ข้อกำหนดรูปแบบ สำหรับเครือข่ายซึ่งเป็นลักษณะระบบเปิดที่เรียกว่า "Open System Interconnection Model " (OSI) โดยโมเดล OSI นี้มีลักษณะเป็นสถาปัตยกรรมแบบระบบเปิด (Open System) เพราะมุ่งหมายที่จะให้ระบบคอมพิวเตอร์ในหลายๆ รูปแบบที่แตกต่างกันสามารถเชื่อมต่อกันได้ OSI โมเดลได้แบ่งโพรโตคอล (Protocol) ในการสื่อสารออกเป็น 7 เลเยอร์(Layer) ซึ่งโพรโตคอล คือ ชุดหรือข้อตกลงในการติดต่อ ข้อสังเกตโมเดล OSI เป็นเพียงข้อเสนอแนะ มิใช่ข้อกำหนด และควรรู้ว่ายังไงก็ไม่มีระบบการเชื่อมต่อที่สร้างเหมือนกับโมเดล OSI จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.5.1 โอ เอส ไอโมเดล

ระดับชั้น	ชื่อระดับชั้น	ระดับหน้าที่
7	Application	Upper Layers
6	Presentation	
5	Session	
4	Transport	Lower Layers
3	Network	
2	Data Link	
1	Physical	

ข้อกำหนดรูปแบบแต่ละระดับชั้นของ โอ เอส ไอ มีการจัดหน้าที่ต้องปฏิบัติแน่นอน โดยจะต้องเป็นตามกลุ่มการบริการดังนี้

1. ระดับชั้น7 แอปพลิเคชั่น(Application Layer) เป็นเลเยอร์ชั้นบนสุดของรูปแบบ OSI ซึ่งเป็นขั้นที่ติดต่อระหว่างผู้ใช้โดยตรงซึ่งได้แก่โฮสท์คอมพิวเตอร์เทอร์มินัลหรือคอมพิวเตอร์ส่วนบุคคล(PC)เป็นต้น แอปพลิเคชั่นใน เลเยอร์นี้ สามารถนำเข้าหรือออกจากระบบเครือข่ายได้โดยไม่ต้องสนใจจะมีขั้นตอนการ ทำงานอย่างไรเพราะจะมีชั้นพรีเซนเตชั่นเป็นผู้รับผิดชอบอยู่แล้ว ในรูปแบบ 0SI นี้ แอปพลิเคชั่นจะการติดต่อกับชั้นพรีเซนเตชั่นโดยตรง

2. ระดับชั้น6 พรีเซนเตชั่นเลเยอร์(Presentation Layer) คอยรวบรวมข้อความและแปลงรหัสหรือแปลงในแบบของข้อมูลที่รับมาจากระดับชั้น 7 เพื่อให้เป็นแบบการสื่อสารเดียวกันเพื่อช่วยลดปัญหาต่างๆที่อาจจะเกิดขึ้นกับผู้ใช้งานในระบบ

3. ระดับชั้น 5 เซสชั่นเลเยอร์(Session Layer) ทำหน้าที่เชื่อมโยงระหว่างผู้ใช้คอมพิวเตอร์เครื่องอื่นๆ โดยผู้ใช้จะใช้คำสั่ง หรือข้อความที่กำหนดไว้ป้อนเข้าไปในระบบ ในการสร้างการเชื่อมโยงนี้ผู้ใช้จะต้องกำหนดรหัสตำแหน่งของจุดหมายปลายทางที่ต้องการติดต่อด้วยใน บางครั้งเครือข่ายอาจมีชั้นเซสชั่นและชั้นทรานสปอร์ตรวมเป็นชั้นเลเยอร์เดียวกัน

4. ระดับชั้น 4 ทรานสเปอร์ตเลเยอร์ (Transport Layer) บางครั้งเรียกว่า ขั้นเครื่องต่อเครื่อง (Host to Host) และจากเลเยอร์ชั้นที่ 4 ถึง ชั้นที่ 7 นี้เรียกรวมกันว่า เลเยอร์เอนทูเอน (End-to-End) ในเลเยอร์ชั้นทรานสปอร์ต (Transport) นี้จะเป็นการสื่อสารระหว่างต้นทางและปลายทางกันจริงๆ สามารถแบ่งการบริการเป็น 2 กรณีคือ

กรณีที่หนึ่ง ส่งข้อมูลของระดับชั้น 5 โดยนำมาใส่ข้อมูลของปลายทาง เช่นที่อยู่แล้วส่งออกไป โดยแบ่งช่วงข่าวสารยาวๆให้เป็นหน่วยที่เล็กลง เรียกว่าแพคเก็ต (Packet) หรือเฟรม (Frame)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่สอง เป็นบริการที่ทำการถอดรหัสข้อมูลที่รับเข้ามาพิจารณาตัดสินว่ามีความสัมพันธ์เกี่ยวข้องกับข้อมูลที่ได้ส่งออกไปหรือเปล่า โดยเลเยอร์ชั้น ทรานสปอร์ต จะทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งมาจากชั้นเลเยอร์ที่ 5 นั้นไปถึงปลายทางจริงหรือไม่ ดังนั้นการกำหนดตำแหน่งของข้อมูลจึงสำคัญในชั้นนี้

5. ระดับชั้น 3 เน็ตเวอร์คเลเยอร์(Network Layer) เป็นชั้นที่ออกแบบหรือกำหนดเส้นทางการเดินทางของข้อมูลที่รับ-ส่งในการส่งผ่านข้อมูลระหว่างต้นทางและปลายทางซึ่งเป็นที่แน่นอนว่าในการสื่อสารข้อมูลผ่านเครือข่ายการสื่อสารจะต้องมีเส้นทางการรับ - ส่ง ข้อมูลมากกว่า 1 เส้นทางดังนั้นในชั้นนี้จะมีหน้าที่เลือกเส้นทางที่ใช้เวลาในการสื่อสารน้อยที่สุดและระยะทางสั้นที่สุดด้วย

6. ระดับชั้น 2 ดาต้าลิ้งค์เลเยอร์(Data Link Layer) เป็นเสมือนผู้ควบคุมการรวมบิทเข้าเป็นตัวอักษรและจัดข้อมูลให้เป็นแพ็คเกตรวมทั้งตรวจสอบความผิดพลาดในการส่งข้อมูล ถ้าผู้รับได้รับข้อมูลแล้วถูกต้องก็จะส่งสัญญาณยืนยันกลับมาว่าได้รับข้อมูลแล้วเรียกว่าสัญญาณ ACK (Acknowledge)ให้กับผู้ส่งแต่ถ้าผู้ส่งไม่ได้รับสัญญาณ ACK หรือรับสัญญาณ NAK (Negative Acknowledge) กลับมาผู้ส่งอาจจะทำการส่งข้อมูลไปใหม่ อีกหน้าที่ของชั้นนี้ คือ ป้องกันไม่ให้เครื่องส่งทำการส่งข้อมูลเร็วจนเกินขีดความสามารถของเครื่องผู้รับข้อมูลจะรับได้

7. ระดับชั้น 1 ฟิสิกส์คอลเลเยอร์(Physical Layer) เป็นระดับชั้นการติดต่อระหว่างอุปกรณ์กับสื่อกลางของเครือข่ายจริงๆ การส่งข้อมูลระดับชั้นนี้ข้อมูลนี้มีลักษณะเป็นบิท สื่อกลางที่ใช้อาจจะเป็นสายไฟเบอร์ออปติค ไมโครเวฟ หรือ ดาวเทียม ตามระบบที่ใช้ โดยมาตรฐานสำหรับเลเยอร์ชั้นนี้จะกำหนดว่าแต่ละคอนเนคเตอร์ เช่น RS-232-C มีกี่ พิน แต่ละพินทำหน้าที่อะไรบ้าง ใช้สัญญาณไฟกี่โวลต์ เทคนิคการมัลติเพลกซ์แบบต่างๆ ก็จะถูกกำหนดอยู่ในชั้นเลเยอร์นี้

ในการสื่อสารกันระหว่างระบบหรือระหว่างคอมพิวเตอร์กับคอมพิวเตอร์โพรโตคอลแบบหนึ่งก็จะถูกใช้สำหรับการสื่อสารระหว่างชั้น N ของคอมพิวเตอร์เครื่องหนึ่งกับชั้น N เดียวกันของคอมพิวเตอร์อีกเครื่องหนึ่งซึ่งในระหว่างการสื่อสารข้อมูลกันจริงๆนั้น โพรโตคอลชั้น N ของอุปกรณ์ทั้งสองเครื่องจะสื่อสารกันผ่านชั้นล่างซึ่งเป็นกายภาพหรือวัตถุ (สื่อกลางการสื่อสาร) แต่โดยแนวความคิดของการสื่อสารข้อมูลแล้วถือว่าชั้น ที่ N ของเครื่องหนึ่งกำลังติดต่อโดยตรงกับชั้น N ของอีกเครื่องหนึ่งเรียกการสื่อสารแบบนี้ว่าการสื่อสารแบบเสมือนจริง (Virtual Communication)

โพรโตคอลของในเลเยอร์แต่ละชั้นจะแตกต่างกันออกไป แต่อย่างไรก็ตามการที่เครื่องคอมพิวเตอร์หลายๆเครื่องจะติดต่อสื่อสารกันได้ในแต่ละเลเยอร์ของแต่ละเครื่องจะต้องใช้โพรโตคอลแบบเดียวกันหรือถ้าใช้โพรโตคอลต่างชนิดกันต้องมีอุปกรณ์ หรือซอฟต์แวร์ที่สามารถแปลงโพรโตคอลที่ต่างกันนั้นให้มีรูปแบบอย่างเดียวกันเพื่อเชื่อมโยงให้คอมพิวเตอร์ทั้ง 2 เครื่องสามารถติดต่อกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ในหนึ่งชั้นของเลเยอร์ไม่ได้ มีการกำหนดว่าจะต้องมีเพียงหนึ่งโพรโตคอลเท่านั้นที่อยู่ระดับเลเยอร์เดียวกันและในทางตรงข้าม ชุดของโพรโตคอลใดๆ อาจจะมีมากกว่าหนึ่งเลเยอร์ประกอบกันเป็นข้อกำหนดของระบบเครือข่าย เรียกว่า ชุดโพรโตคอล (Protocol Suite) เป็นชุดโพรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นต้น ประโยชน์ในการแบ่งเลเยอร์คือ กำหนดการติดต่อระหว่างเลเยอร์ทำโดยไม่ต้องคำนึงถึงการเปลี่ยนในเลเยอร์ใด ๆ ที่ติดกัน

2.5.3 ทีซีพี/ไอพี [TCP/IP(Transmission Control Protocol/Internet Protocol)]

อินเทอร์เน็ตนับได้ว่าเป็นเครือข่ายบนเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นเชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่นๆ แต่สิ่งที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันก็คือ แต่ละเครือข่ายใช้คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อและระบบปฏิบัติการที่ต่างกัน มาตรฐานของ ทีซีพี(TCP) จึงถูกใช้เป็นกุญแจสำคัญในการแก้ปัญหาเหล่านี้ โดยกลายเป็นระบบเปิดที่สมบูรณ์แบบมีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่เครื่องพีซีจนถึงเครื่องเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ ทีซีพี จึงเป็นมาตรฐานที่ทั่วโลกยอมรับมีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน ทีซีพี/ไอพี มากมาย ดังนั้นจึงนับได้ว่า ทีซีพี/ไอพี เป็นหัวใจของอินเทอร์เน็ตเลยทีเดียว

2.5.3.1 ทีซีพี/ไอพี คืออะไร

ทีซีพี/ไอพี เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Network Protocol) จัดทำเพื่อ ใช้เป็นกฎเกณฑ์ให้เครื่องคอมพิวเตอร์ ใช้งานร่วมกันในลักษณะของ ระบบเปิด (Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตาม จะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันใช้ได้

ทีซีพี/ไอพี เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่างๆ ที่กำหนดให้ทำเมื่อเกิดความผิดพลาด หรือต้องทำเพื่อป้องกันเพื่อไม่ให้เกิดความผิดพลาด

ทีซีพี/ไอพี เกิดจากการนำข้อกำหนดของรูปแบบต่างๆ กันมาใช้ร่วมกัน ทีซีพี(TCP) และไอพี(IP) ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ (แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า ทีซีพี/ไอพี) ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบ เครือข่ายจะใช้ ทีซีพี/ไอพี เป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ (Local Area Network) และเครือข่ายบริเวณกว้าง (Wide Area Network) ไม่ได้ใช้งานเฉพาะกับอินเทอร์เน็ตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50314

2.5.3.2 ส่วนประกอบของ ทีซีพี/ไอพี

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่างๆซึ่งแบ่งเป็นกลุ่มได้ ดังนี้

2.5.3.2.1 กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocols) ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูล ระหว่างคอมพิวเตอร์สองเครื่อง แบ่งย่อยออกได้เป็นสองชนิด คือ

1. **TCP** (Transmission Control Protocol) เป็นการบริการแบบ คอนเน็คชั่นเบสเซอร์วิส (Connection Based Service) ซึ่งคอมพิวเตอร์ด้านผู้รับและส่งต้องส่งต่อถึงกันอยู่ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบก็คล้ายกับระบบโทรศัพท์ที่ต้องติดต่อกันให้ได้ก่อนจะพูดคุยกันได้

2. **UDP** (User Datagram Protocol) เป็นการให้บริการแบบ คอนเน็คชั่นเซอร์วิส (Connection Service) คอมพิวเตอร์ด้านผู้ส่งไม่จำเป็นต้องติดต่อกับด้านผู้รับก่อน เพียงรู้ที่อยู่ของด้านผู้รับแล้วใส่ที่อยู่นั้นไปกับข้อมูลที่ส่งออก ข้อมูลจะเดินทางตามเส้นทางต่างๆ เพื่อไปถึงปลายทางตามที่อยู่ คล้ายกับการส่งจดหมายที่ไปรษณีย์จะส่งให้ตามที่อยู่ที่จ่าหน้าซองจดหมายโดยผู้ส่งและผู้รับไม่ต้องติดต่อถึงกัน

2.5.3.2.2 กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol) ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่ กลุ่มข้อมูลรูปแบบนี้ก็จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสมแล้วส่งออกไป เมื่อถึงผู้รับปลายทาง กลุ่มข้อมูลนี้จะทำหน้าที่ตรงข้าม คือ รวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผล กลุ่มข้อกำหนดรูปแบบกลุ่มนี้ประกอบด้วย

1. **IP** (Internet Protocol) เป็นการกำหนดรูปแบบการส่งข้อมูล
2. **ICMP** (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับสถานะของ IP เช่น ข่าวสารความผิดพลาดและผลกระทบต่อเส้นทาง เมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ใน เครือข่าย
3. **RIP** (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับทำการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุดเหมาะสมกับข้อมูลมากที่สุด
4. **OSPF** (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากทางที่สั้นที่สุดก่อน

2.5.3.2.3 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address) ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยที่ไม่ว่าเครือข่ายจะใหญ่โตสักเพียงใดหรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตาม ที่อยู่ต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบนี้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. **ARP** (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน

2. **DNS** (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะการใช้งานจริงนั้นจะใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบมีชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้

3. **RARP** (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เดียวกับ ARP แต่จะทำตรงข้ามกับ ARP

2.5.3.2.4 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocols) และ สนับสนุนข้อมูลสถานะเพื่อนำไปใช้เลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

1. **EGP** (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะหาการถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์กับเครือข่ายภายนอกเพื่อทำการสื่อสาร

2. **GGP** (Gateway to Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่างเกตเวย์ กับ เกตเวย์

3. **IGP** (Interior Gateway Protocol) ข้อกำหนดในแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

2.5.3.2.5 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services) ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดรูปแบบนี้ประกอบด้วย

1. **BOOTP** (BOOT Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายให้เริ่มทำงาน ข้อกำหนดรูปแบบนี้จะอ่านโปรแกรมควบคุมการทำงานจากคอมพิวเตอร์ให้บริการ(Server Computer) มาให้

2. **FTP** (File Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนไฟล์ข้อมูลระหว่างคอมพิวเตอร์ ซึ่งอาจจะอยู่บนเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้

3. **TELNET** เป็นข้อกำหนดรูปแบบที่ทำให้บริการเกี่ยวกับการควบคุมการติดต่อระยะทางไกล (Remote Login)

กลุ่มข้อกำหนดรูปแบบอื่นที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญ ๆ จัดทำไว้ให้บนเครือข่ายที่สนใจมีดังนี้

NFS (Network File System) เป็นข้อกำหนดรูปแบบที่ทำให้ผู้ใช้คอมพิวเตอร์เครื่องหนึ่งสามารถเข้าใช้งานไฟล์ข้อมูลและดูไฟล์ข้อมูลซึ่งอยู่ในคอมพิวเตอร์เครื่องอื่นได้

NIS (Network Information System) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Logins และ Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล

SMTP (Simple Procedure Call) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนจดหมายอิเล็กทรอนิกส์ (Electronic Mail) ระหว่างคอมพิวเตอร์

SNMP (Simple Network Management Protocol) ข้อกำหนดรูปแบบที่ให้บริการข่าวสารต่างๆที่แสดงสถานะของเครือข่ายและอุปกรณ์ที่ต่ออยู่บนเครือข่าย

2.5.3.3 โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี

ตาราง 2.5.2 โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี เปรียบเทียบกับแบบจำลอง โอ เอส ไอ

OSI Model	TCP/IP(Internet)
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	Physical

1. Application Layer ในชั้นนี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้ในเครือข่ายเช่น โปรแกรมส่งถ่ายข้อมูล (File-Transfer Program) และอาจกล่าวได้ว่าโพรโตคอล TCP/IP ก็คือโพรโตคอลในชั้นแอปพลิเคชั่นรวมกับชั้นพรีเซนเตชั่นของ OSI โมเดลนั้นเอง

2. Transport Layer ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายเปรียบเทียบได้กับชั้นรวมกับทรานสปอร์ตเลเยอร์นั่นเอง โดยโพรโตคอล TCP/IP มีซ็อกเก็ต (Socket) เป็นจุดปลาย (End-Point) ในการสื่อสารซึ่งซ็อกเก็ตนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์และหมายเลขเพอร์ต (Port) ของเครื่องที่ต้องการส่งข้อมูลไปถึงในชั้นนี้มีการรับรองให้ถึงที่หมายและลำดับข้อมูลที่ส่งโดยปราศจากข้อมูลซ้ำซ้อน โดยในชั้นนี้มีโพรโตคอลหลัก 2 ตัว คือ TCP และ UDP

3. Internet Layer ในชั้นนี้มีการกำหนดดาต้าแกรม (Datagram) และทำการหาเส้นทางการส่ง การทำงานในชั้นนี้จะเป็นแบบ Connectionless เนื่องจากไม่มีการเชื่อมต่อระหว่างต้นทางกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปลายทางก่อนดาต้าแกรมแต่ละตัวสามารถเลือกเส้นทางไปโดยอิสระและไม่มีการรับประกันความถูกต้องข้อมูลหรือลำดับการส่ง

4. Network Interface Physical Layer ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็นส่วน ๆ เรียกว่า เฟรม หรือแพคเก็ตและส่งข้อมูลที่ได้ไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

2.5.3.4 ข้อแตกต่างระหว่างชุดโพรโตคอล TCP/IP

1. ลำดับการติดต่อสื่อสารของชั้นเลเยอร์ ในรูปแบบ OSI นั้นจะกำหนดลำดับชั้นการสื่อสารที่เป็นลำดับขั้นตอนการติดต่อที่แน่นอน โดยเฉพาะการอินเตอร์เฟสระหว่างชั้นเลเยอร์ซึ่งทำให้รูปแบบ OSI สามารถเป็นระบบเปิดสำหรับระบบเครือข่ายคอมพิวเตอร์ทั่วไป เพราะไม่ว่าจะมีการเปลี่ยนแปลงโพรโตคอลในเลเยอร์ชั้นใดก็ตามจะไม่มีผลกระทบต่อการสื่อสารเลเยอร์ชั้นถัดไปในขณะที่ชุดโพรโตคอล TCP/IP จะไม่มีการกำหนดรูปแบบการติดต่อที่ตายตัวเพื่อให้ผู้ออกแบบเครือข่ายมีอิสระสามารถเปลี่ยนแปลงโครงสร้างของเครือข่ายได้ง่ายขึ้น

2. การติดต่อสื่อสารระหว่างเครือข่ายหรือการติดต่อผ่านอินเทอร์เน็ต คือการติดต่อสื่อสารข้อมูลระหว่างระบบคอมพิวเตอร์ 2 ระบบที่ไม่สามารถติดต่อสื่อสารกันได้โดยผ่านทางเครือข่ายการสื่อสารข้อมูลเพียงเครือข่ายเดียวได้ต้องอาศัยเครือข่ายตั้งแต่ 2 เครือข่ายขึ้นไปในการติดต่อสื่อสารกันและเครือข่ายเหล่านี้อาจจะมีลักษณะของเครือข่ายที่ต่างกันก็ได้

ความแตกต่างในเรื่องของอินเทอร์เน็ตระหว่างชุดโพรโตคอล TCP/IP กับรูปแบบ OSI ก็คือในชุดโพรโตคอล TCP/IP จะใช้ โพรโตคอลสำหรับอินเทอร์เน็ต ที่เรียกว่า โพรโตคอล IP (Internet Protocol) ซึ่งในรูปแบบ OSI จะเรียกว่าโพรโตคอลสำหรับการอินเทอร์เน็ตว่า โพรโตคอลเน็ตเวิร์ค

3. การบริการการเชื่อมต่อการสื่อสาร (Connection Service) ในชุดโพรโตคอล TCP/IP นั้นจะมีการบริการการเชื่อมต่อการสื่อสารระหว่างต้นทางและปลายทาง 2 แบบ คือการบริการแบบ Connectionless และแบบ Connection-Oriented ส่วนในรูปแบบ OSI จะให้ความสำคัญเฉพาะกับการบริการแบบ Connection-Oriented เท่านั้น

4. โพรโตคอลควบคุมการจัดการสื่อสาร ในชุดโพรโตคอล TCP/IP จะใช้โพรโตคอล TCP เป็นโพรโตคอลสำหรับควบคุมการสื่อสาร กำหนดตำแหน่งต้นทางและปลายทาง และอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับข้อมูลซึ่งในรูปแบบ OSI นั้นจะแบ่งแยกการควบคุมการสื่อสารออกจากกันโดยใช้โปรโตคอลเซสชั่นและโปรโตคอลทรานสปอร์ตตามลำดับ

2.5.3.5 ลักษณะของการติดต่อ

แบ่งออกเป็น 2 ชนิดคือ

Connection-Oriented คือการติดต่อที่ต้องมีการเชื่อมโพรเซส (Process) ที่จะมีการส่งหรือรับข้อมูลซึ่งใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่างโพรเซส ถึงแม้ว่าข้อมูลนี้อาจจะจ่าย Pocket-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งขั้นตอนการทำงานออกเป็น

- ขั้นการสร้างการติดต่อ (Connection Establishment)
- ขั้นการส่งผ่านข้อมูล (Data Transfer)
- ขั้นยกเลิกการติดต่อ (Connection Termination)

Connectionless หรือดาต้าแกรม คือจะไม่มีขั้นการสร้างการติดต่อ และขั้นการยกเลิกการติดต่อ แต่จะมีขั้นการส่งผ่านข้อมูลเพียงอย่างเดียว โดยข้อมูลซึ่งเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

2.5.3.6 ความสัมพันธ์ของ TCP และ UDP กับอินเตอร์เน็ต

ตามสถาปัตยกรรมของชุดข้อกำหนดรูปแบบ TCP/IP ระดับชั้น ทรานสปอร์ต จะมีบริการส่งข้อมูลอยู่ 2ประเภท คือ TCP และ UDP

ความแตกต่างของ TCP กับ UDP อยู่ที่วิธีการสื่อสารระหว่างเครื่องคอมพิวเตอร์ สองเครื่อง โดย **TCP** ต้องสร้างการติดต่อให้ได้ก่อนแล้วจึงส่งข้อมูลและยกเลิกเส้นทางที่ต่อกันเมื่อจบการส่งข้อมูล ส่วน **UDP** จะไม่มีการสร้างการติดต่อ แต่จะใช้ไอพีแอดเดรสของคอมพิวเตอร์ปลายทางใส่รวมกับข้อมูลแล้วส่งออกไป ข้อมูลจะเดินตามไอพีแอดเดรสด้วยเส้นทางที่ไม่แน่นอน ขึ้นอยู่กับความเหมาะสมจนกว่าจะถึงคอมพิวเตอร์ปลายทาง

TCP เป็นวิธีการสื่อสารที่มีความเชื่อมั่นได้มากกว่า UDP ทั้งการส่งและการตอบรับข่าวสาร วิธีของ UDP จะไม่มีการรับประกันว่าข่าวสารจะไปถึงผู้รับแต่ UDP มีวิธีการที่จะตรวจสอบโดยให้เครื่องคอมพิวเตอร์ปลายทางส่งข่าวสารการตอบรับกลับมาเมื่อได้รับข่าวสารที่ส่งไปให้ ทางด้านผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งจะตั้งเวลาว่านานเท่าใดควรมีการตอบรับกลับมา ถ้าเกินเวลา (time-out) แล้วยังไม่มีการตอบรับกลับมาก็สันนิฐานได้ว่าข่าวสารสูญหาย UDP ก็จะทำการส่งข่าวสารเดิมให้ใหม่

เครือข่ายขนาดใหญ่เช่นอินเทอร์เน็ต ซึ่งมีเครื่องคอมพิวเตอร์บนเครือข่ายเป็นจำนวนมาก การใช้ TCP ที่ต้องสร้างการติดต่อและยกเลิกอยู่บ่อย ๆ จึงเป็นไปไม่ได้ในทางปฏิบัติ ดังนั้นอินเทอร์เน็ต จึงใช้การส่งข่าวสารด้วย UDP แทน

การบริการที่ดำเนินการด้วยชุดข้อกำหนดรูปแบบ TCP/IP ต้องออกแบบการให้บริการเป็น TCP หรือ UDP อย่างใดอย่างหนึ่งเท่านั้น การบริการอย่างเดียวจะให้บริการทั้งสองประเภทไม่ได้ เช่น การ Login จากระยะไกล (Telnet) และการโอนย้ายไฟล์ข้อมูล FTP ใช้การให้บริการแบบ TCP ส่วน TFTP (Trivial File Transfer Protocol) ใช้การให้บริการแบบ UDP

2.5.3.7 พอร์ตและซ็อกเก็ต (TCP Port and Socket)

การใช้บริการของ TCP ทำโดยผ่านทางพอร์ต โดยมีการกำหนดหมายเลขให้กับการบริการต่าง ๆ การบริการที่ต่างกันจะมีหมายเลขพอร์ตไม่เหมือนกัน

พอร์ตถูกกำหนดตามชนิดของการบริการ เช่น ต้องการกำหนดหมายเลขให้กับการบริการต่าง การบริการที่ต่างกันจะมีหมายเลขพอร์ตไม่เหมือนกัน

พอร์ตถูกกำหนดตามชนิดของการบริการ เช่น ต้องการสื่อสารกันด้วย Telnet เครื่องคอมพิวเตอร์ที่ต้องการจะส่งความต้องการออกทางพอร์ตหมายเลข 23 ที่จัดไว้สำหรับบริการ Telnet ไปยังเครื่องคอมพิวเตอร์ที่ต้องการติดต่อด้วย ถ้าเครื่องคอมพิวเตอร์ปลายทางยอมรับการติดต่อ การสื่อสาร Telnet ก็จะเริ่มขึ้น ในช่วงเวลานี้ถ้ามีความต้องการการสื่อสารด้วย Telnet จากเครื่องคอมพิวเตอร์อื่นเข้ามา จะไม่สามารถทำการติดต่อได้เนื่องจากพอร์ตหมายเลข 23 สำหรับบริการ Telnet ถูกใช้งานอยู่ TCP พอร์ตเป็นพอร์ตทาง Logical ไม่ใช่พอร์ตที่อยู่ด้านหลังของเครื่องคอมพิวเตอร์ที่ใช้คอนเน็กเตอร์(Connector)บนแบคโบนเพื่อให้สามารถสื่อสารบริการหลายประเภทได้พร้อม ๆ กัน

ซ็อกเก็ตเป็นหมายเลขกำหนดวงจรซึ่ง ทางเข้า-ออก ของระดับชั้น TCP ของเครื่องคอมพิวเตอร์บนเครือข่าย หมายเลขซ็อกเก็ตประกอบด้วยเลขไอพีแอดเดรสกับหมายเลขพอร์ตรวมกัน หมายเลขซ็อกเก็ตที่ใช้กับการสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่อยู่ต่างเครือข่ายได้เนื่องจากไอพีแอดเดรสของเครือข่ายจะไม่ซ้ำกัน ถึงแม้ว่าหมายเลขพอร์ตจะซ้ำกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคอมพิวเตอร์บนเครือข่ายจะสร้างตารางพอร์ตขึ้นมาเก็บรายการของพอร์ตที่มีอยู่เพื่อใช้ตรวจสอบว่าจะใช้บริการพอร์ตที่ต้องการได้หรือไม่และพอร์ตมีสถานะเป็นอย่างไรการที่พอร์ตที่อยู่ในรายการตารางพอร์ตของแต่ละเครื่อง ร้องขอการสื่อสารซึ่งกันและกัน เรียกว่า 'Binding' ของพอร์ตและวิธีที่ทำให้พอร์ตที่ถูกใช้สามารถใช้สื่อสารได้อีกจากการขอใช้บริการเดียวกันเรียกว่า 'Multiplexing'

ตารางที่ 2.5.3 หมายเลขพอร์ตที่ให้บริการต่างๆ

TCP พอร์ต	บริการ
20	FTP Data
21	FTP Control
23	Telnet
25	Simple Mail Transfer Protocol
53	Domain Name Server
69	Trivial File Transfer Protocol
79	Finger

2.5.3.8 ข้อกำหนดรูปแบบ IP (Internet Protocol)

IP เป็นข้อกำหนดรูปแบบหนึ่งในชุดข้อกำหนดรูปแบบ TCP/IP ลักษณะการให้บริการเป็นแบบ Connectionless เช่น เดียวกับ UDP มีหน้าที่หลักเกี่ยวกับการจัดการดาต้าแกรมระหว่างเครื่องคอมพิวเตอร์ เช่น ที่อยู่ปลายทาง วิธีการหาเส้นทางที่ดีที่สุดที่จะไปถึงปลายทาง และจัดเตรียมวิธีการแก้ปัญหาที่อาจจะเกิดขึ้นในกรณีต่าง ๆ เป็นต้น นอกจากนี้ยังมีหน้าที่แบ่งย่อยดาต้าแกรมที่มีขนาดใหญ่เกินกว่าที่ IP Message กำหนดไว้(ประมาณ 64 กิโลไบต์) และรวบรวมดาต้าแกรม ย่อย ๆ กลับมาเป็น Message ให้เหมือนเดิมที่เครื่องปลายทาง การแบ่งย่อยและรวบรวมดาต้าแกรม มีอยู่ 4 วิธีคือ

1. Segmentation วิธีแบ่งดาต้าแกรมขนาดใหญ่ให้เป็นดาต้าแกรมย่อย
2. Reassemble วิธีรวบรวมดาต้าแกรมย่อย ๆ ให้กลับอยู่ในสภาพเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Concatenation วิธีรวมดาต้าแกรมให้เป็นบล็อก

4. Separation วิธีแบ่งบล็อกให้เป็นดาต้าแกรม

2.5.3.9 ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol)

ปัญหาหนึ่งที่เกิดขึ้นได้ง่ายกับดาต้าแกรม ก็คือ ดาต้าแกรมเดินทางผิดเส้นทาง ทำให้สูญหายหรือบางส่วนข่าวสารเสียหาย เครื่องคอมพิวเตอร์ด้านส่งต้องรู้เงื่อนไขที่จะทำให้เกิดความผิดพลาดต่างๆ บนเครือข่ายเพื่อให้ดาต้าแกรมถูกส่งออกไปอย่างถูกต้อง

ICMP จะรายงานความผิดพลาดในระบบให้กับไอพีพิเศษ ICMP มี Header เหมือนกับข่าวสารไอพีอื่น ๆ โดยจะถูกส่งรวมไปกับข่าวสารอื่นสู่เครื่องปลายทาง ซึ่งเครื่องปลายทางจะส่งรายงานของความผิดพลาดกลับมาให้เครื่องด้านส่งได้รู้ว่าเกิดอะไรขึ้นบ้างในการเดินทาง ซึ่งข้อมูลในรายงานจะถูกนำมาใช้แก้ปัญหาต่อไป

2.5.4 ไอพีแอดเดรส (IP Address)

เครื่องคอมพิวเตอร์ที่ต่อเชื่อมกับอินเทอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP มีอยู่เป็นจำนวนมาก เครื่องคอมพิวเตอร์แต่ละเครื่องต้องสามารถระบุหรือ อ้างอิงได้โดยไม่เกิดความซ้ำซ้อนกับเครื่องคอมพิวเตอร์อื่น มิฉะนั้นแล้วข่าวสารที่เครือข่ายรับมาจะไม่สามารถส่งไปให้กับเครื่องคอมพิวเตอร์ที่ต้องการข่าวสารนั้นได้ จึงต้องมีการจัดระบบที่ดี เครือข่ายอินเทอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP ได้ออกแบบการจัดการระบบตรงส่วนนี้ไว้แล้ว เครือข่ายอินเทอร์เน็ตใช้รหัสหมายเลขมากำหนดให้แต่ละเครือข่ายและเครื่องคอมพิวเตอร์ภายในเครือข่ายที่เชื่อมโยงเรียกรหัสหมายเลขเหล่านี้ว่า **อินเทอร์เน็ตแอดเดรส (Internet Address)**

ไอพีแอดเดรส ประกอบด้วยเลขฐานสองจำนวน 32 บิต แบ่งออกเป็น 4 ส่วน แต่ละส่วนมี 8 บิตเมื่อดูเฉพาะแต่ละส่วนเป็นเลขฐานสิบจะได้เลขจำนวน 256 ค่าไม่ซ้ำกัน (0-255)ไอพีแอดเดรสจะนำเอาหมายเลขทั้ง 4 ส่วนมารวมกัน โดยแยกแต่ละส่วนด้วยจุดดังนั้นหมายเลขทั้งหมดที่เป็นไปได้ โดยไม่ซ้ำกัน คือ 256^4 หรือ 4,294,967,296 จำนวน มีค่าหมายเลขจาก 000.000.000.000 จนถึง 255.255.255.255 หมายเลขเหล่านี้เองที่อินเทอร์เน็ตใช้กำหนดให้กับเครือข่ายและเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิงถึง

ไอพีแอดเดรสบางหมายเลขสงวนไว้ใช้ ด้วยจุดหมายกรณีพิเศษ ทำให้ไอพีแอดเดรสที่ใช้งานทั่วไปลดลงจากจำนวนที่เป็นไปได้ความหมายของไอพีแอดเดรสจะแบ่งได้เป็น 2 กลุ่มดังนี้

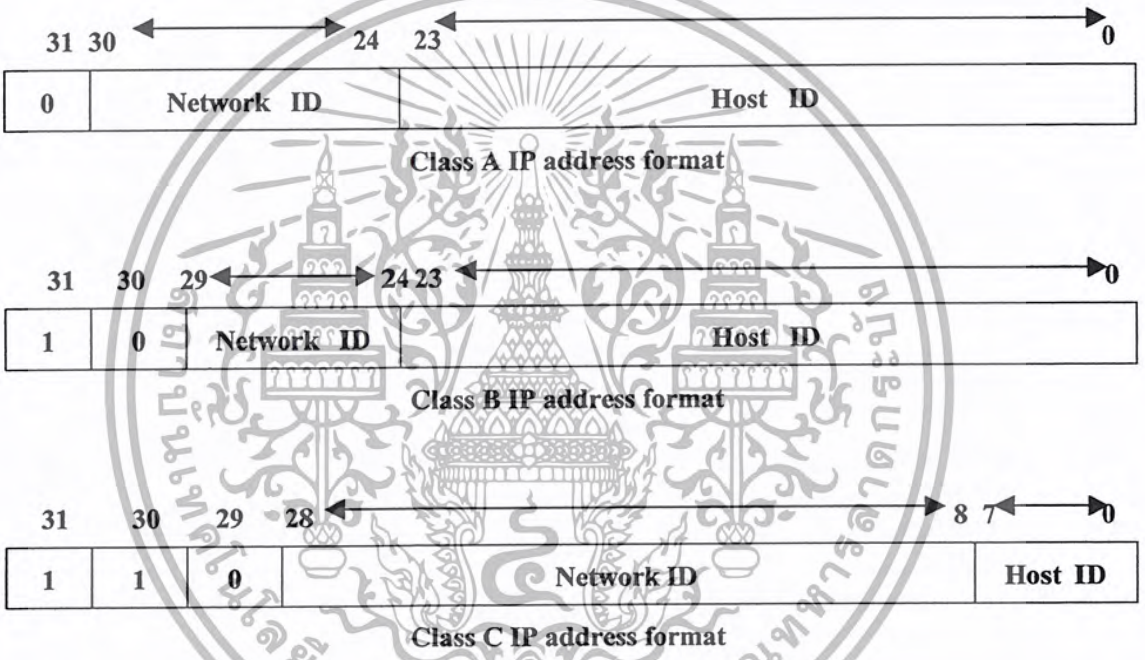
1. กลุ่มที่ใช้เป็นรหัสประจำเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย (Host Computers)

ไอพีแอดเดรสในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์สามารถซ้ำกันได้ แต่กลุ่มรหัสประจำเครือข่ายจะซ้ำกันไม่ได้ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิงถึง

นอกจากนี้เพื่อความเหมาะสมในการกำหนดไอพีแอดเดรสให้กับผู้ขอ ทางผู้บริหารอินเทอร์เน็ตแบ่งคลาสของผู้ขอไอพีแอดเดรสตามขนาดของเครือข่ายเพื่อให้ทรัพยากรส่วนนี้ถูกใช้อย่างคุ้มค่าที่สุด องค์กรขนาดใหญ่ก็จะจัดให้อยู่ในคลาสที่สามารถกำหนด ไอพีแอดเดรสให้กับเครื่องคอมพิวเตอร์ในเครือข่ายได้มาก การแบ่งคลาสจะแบ่งได้ดังนี้



รูปที่ 2.6 แสดงคลาสของไอพีแอดเดรส

การกำหนดหมายเลขของเครือข่ายและเครื่องคอมพิวเตอร์นอกจากจะแบ่งไอพีแอดเดรสเป็นคลาสทั้ง 5 ประเภทแล้ว ยังมีข้อกำหนดปลีกย่อยอีกหลายประการที่ผู้วางระบบต้องรู้ไว้ เพื่อที่จะกำหนดแอดเดรสใช้งานได้อย่างถูกต้องไม่ผิดพลาด ได้แก่ ไอพีแอดเดรสที่เป็น '0' ทุกบิต จะไม่ใช้งานทั่วไป แต่นำไปใช้กับอุปกรณ์หาเส้นทาง (Router) เพื่อกำหนด ' Default Route '

1. ไอพีแอดเดรสในแต่ละส่วนคือ ส่วนหมายเลขเครือข่ายและส่วนหมายเลขประจำเครื่องคอมพิวเตอร์จะเป็น '0' หรือ '1' ทุกบิตไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าส่วนของหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '0' ทุกบิต หมายถึง หมายเลขเครือข่ายนั้นใช้งานร่วมกับอุปกรณ์หาเส้นทางหรือเพื่อบอกให้เครื่องคอมพิวเตอร์รู้ว่าตัวเองอยู่ในเครือข่ายใด

- ส่วนของหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '1' บิต หมายถึง เลขเครือข่ายนั้นใช้สำหรับการกระจายข่าวภายในเครือข่าย (Broadcast Address)

- ถ้าไอพีแอดเดรสทั้งสองส่วนเป็น '1' ทุกบิต หมายถึง แอดเดรสที่ใช้กระจายข่าวหรืออีกนัยหนึ่งเครื่องคอมพิวเตอร์ที่ส่งแอดเดรสนี้เข้าสู่ระบบไม่ทราบว่าตนเองอยู่ในเครือข่ายใด

2. ไอพีแอดเดรสของคลาส A หมายเลข **127.0.0.0** จะสงวนไว้ใช้งานเฉพาะอย่าง เช่น IPC (Inter-Process Communication)

จากรูปที่ **2.6** จะแสดงประเภทของไอพีแอดเดรสและข้อกำหนดปลีกย่อยต่างๆเราสามารถทราบได้ว่าเครือข่ายขององค์กรถูกจัดอยู่ใน Class ใด โดยดูจากค่าหมายเลขของ 8 บิทแรกซ้ายมือสุดดังนี้

Class A: ไอพีแอดเดรสอยู่ในช่วง 1 ถึง 126

Class B: ไอพีแอดเดรสอยู่ในช่วง 128 ถึง 191

Class C: ไอพีแอดเดรสอยู่ในช่วง 192 ถึง 233

2.5.5 การติดต่ออินเทอร์เน็ตของ วิชวลเบสิค โดยใช้ MS Winsock Control

ในการที่เราจะทำการเขียน Internet Application โดยใช้ Visual Basic จะมี Control (ในที่นี้คือ ActiveX Control) อยู่หลายตัวที่ทำหน้าที่ช่วยเรา ซึ่ง Control เหล่านี้บางตัวก็จะถูกออกแบบมาสำหรับบริการบางประเภทเท่านั้นซึ่งต้องเลือกเอาเอง ตามความเหมาะสม ซึ่งบริการเหล่านั้นจะเป็นบริการพื้นฐานเท่านั้น ถ้าเราต้องการจะสร้าง Application เพื่อติดต่อกับบริการบางประเภทที่ไม่เป็นที่นิยม หรือเป็นรูปแบบบริการที่เราสร้างขึ้นมาเอง ก็คงจำเป็นที่จะต้องใช้ Control ที่เป็น Control ด้าน **TCP** หรือ **UDP** ซึ่งเป็น Control ที่ค่อนข้างจะเป็น Control ที่ทำหน้าที่จัดการเกี่ยวกับการติดต่อสื่อสารโดยใช้ Protocol ที่อยู่บน IP (Internet-Protocol) โดยสามารถสร้างการติดต่อสื่อสารที่เป็น TCP (Transfer Control Protocol) ซึ่งเป็น Protocol ที่เป็น Connection-Oriented กับอีก Protocol หนึ่งก็คือ UDP (User Datagram Protocol) ซึ่งเป็น Protocol แบบ Connectionless ซึ่งก็เพียงพอที่เราจะสร้าง Application ใดๆก็ได้สำหรับ Internet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MS Winsock Control เป็น Control ที่จะทำให้การใช้งาน TCP และ UDP ได้ง่ายขึ้น ซึ่ง Control ตัวนี้สามารถที่จะใช้ได้ทั้ง MS Visual Basic, MS Visual C++, MS Visual FoxPro หรือแม้แต่ MS Access ซึ่งทำให้เราสามารถที่จะสร้าง Client หรือ Server Application ได้โดยที่ไม่จำเป็นต้องทราบรายละเอียดเกี่ยวกับ TCP หรือวิธีที่จะทำการเรียกใช้ Windows API ระดับล่างเลยเราเพียงแค่กำหนด Properties ต่างๆ แล้วเรียกใช้ Method เหล่านั้นเราก็จะสามารถที่จะสร้างการติดต่อสื่อสารกับเครื่องอื่นๆแล้วแลกเปลี่ยนข้อมูลข่าวสารซึ่งกันและกัน

โพรโตคอล TCP เบื้องต้น

TCP เป็นการติดต่อสื่อสารที่เป็น Connection- Oriented ที่จะมีการสร้าง Connection กันไว้ ซึ่งก็จะทำให้ทั้งสองฝั่งสามารถที่จะส่งข้อมูลที่เป็นลักษณะ stream ไปมาระหว่างสองเครื่องนั้นได้ การสร้าง Application ในลักษณะนี้จะมีเครื่องอยู่สองฝั่งโดยฝั่งหนึ่งจะทำงานเป็น Server ที่จะรอให้เครื่อง Client ทำการขอการติดต่อสื่อสารเข้ามา

ในกรณีที่เราจะสร้าง Application ให้เป็น Client เราจำเป็นที่จะต้องทราบที่อยู่ของเครื่องปลายทางซึ่งอาจจะเป็น Hostname หรือหมายเลข IP เพื่อที่จะใช้ในการระบุใน RemoteHost property และต้องทราบหมายเลข Port เพื่อจะใช้ระบุใน RemotePort property ที่ทำการ Listen (รอการ connect) อยู่จากนั้นก็ทำการเรียก Connect method เพื่อสร้าง Connection

สำหรับถ้าต้องการสร้าง Application ที่เป็น Server ในส่วน Server นี้เราต้องกำหนด LocalPort property แล้วทำการเรียก Listen method เพื่อรอการร้องขอเข้ามาจาก Client เมื่อมีการร้องขอเข้ามา ก็จะเกิดเหตุการณ์ ConnectionRequest event เข้ามาเพื่อที่จะสร้าง Connection เราจำเป็นต้องเรียก Accept method ในขณะที่เกิดเหตุการณ์ ConnectionRequest event

เมื่อสร้าง Connection แล้วเครื่อง Computer ทั้งสองฝั่งก็จะสามารถที่จะรับและส่ง Data ไปมาระหว่างกันได้ ถ้าต้องการจะส่ง Data ก็ทำการเรียก SendData method และเมื่อมีข้อมูลเข้ามาก็จะเกิดเหตุการณ์ DataArrival event ขึ้น ถ้าจะอ่านข้อมูลก็ทำการเรียก GetData method ในขณะที่เกิดเหตุการณ์ DataArrival event

โพรโตคอล UDP เบื้องต้น

UDP (User Datagram Protocol) เป็น Protocol ที่ทำงานในลักษณะ Connection-less ซึ่งจะต่างจาก TCP ตรงที่ UDP ไม่มีการการสร้าง Connection และเครื่องที่มีการส่งการติดต่อสื่อสารกันนั้นเครื่องจะเป็น Client หรือ Server ก็ได้

ในการที่จะส่งข้อมูลหากันนั้น เครื่องที่เป็น Client จะเพียงแค่กำหนดหมายเลข Port ใน LocalPort property และกำหนดหมายเลข Port ใน RemotePort property ซึ่งจะต้องเป็นหมายเลขเดียวกันที่กำหนดไว้ที่เครื่อง Client จากนั้นก็ทำการเรียก SendData method เพื่อส่งข้อมูล และเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่อง Client ต้องการจะอ่านข้อมูลก็เพียงเรียก GetData method ในขณะที่มีเหตุการณ์ DataArrival event

คุณสมบัติของ Winsock Control (Properties)

Property นี้จะใช้สำหรับการกำหนดคุณสมบัติต่างๆให้กับวินซ็อก เช่น เลขหมายเครื่องปลายทางเลขหมายพอร์ต ที่เราต้องการจะติดต่อ หรือจะเป็นการเรียกดูคุณสมบัติของวินซ็อกในขณะนั้นเช่นดูว่าขณะนั้น วินซ็อกมีสถานะเป็นอย่างไร อ่านค่าหมายเลขที่อยู่ IP หรือ ค่าพอร์ต ของเราเอง เป็นต้น

ตาราง 2.5.4 คุณสมบัติวินซ็อกคอนโทรล

ชื่อ Property	*Return Value*	อ่านได้อย่างเดียว?	*คำอธิบาย*
BytesReceived	*Long*	ใช่	เป็นการคืนค่ากลับจำนวนของ ไบต์ ที่รอในการรับจาก buffer ซึ่งใช้ *GetData* method ในการรับข้อมูล
LocalHostName	*String*	ใช่	เป็นคืนค่า ชื่อที่อยู่ของเรา เพื่อให้เรารู้ว่าเป็นอะไร
LocalIP	*String*	ใช่	คืนค่าของ Local host ที่อยู่ IP ในรูปแบบของ String เช่น 198.164.0.94 เป็นต้น
LocalPort	*Long*	ไม่ใช่	กำหนด port ที่รับข้อมูลของเครื่องเรา เช่น 1000 ถ้ามีข้อมูลส่งมาที่ IP เราและตรงกับ LocalPort ในที่นี้ คือ 1000 เราก็จะได้รับ ข้อมูลนี้จากวินซ็อกตัวที่มี LocalPort อยู่ที่ 1000
Protocol	*Long*	ไม่ใช่	เป็นการ ตั้งค่าให้กับ Winsock ว่า จะใช้ Protocol อะไรซึ่งมีให้เลือก คือ *sckTCPProtocol* และ *sckUDPProtocol* (TCP และ UDP) ซึ่งการคืนค่าคือ 0 และ 1 ตามลำดับ
RemoteHost	*String*	ไม่ใช่	คืนค่าหรือตั้ง เครื่องปลายทางที่ติดต่อ ซึ่งสามารถที่จะตั้งให้เป็นแบบ String หรือ เลขหมายที่อยู่ IP ก็ได้แบบ String ก็เช่น **www.kmitl.ac.th** และ แบบเลขหมายที่อยู่ ก็เช่น **161.246.10.21**
RemoteHostIP	*String*	ใช่	คืนค่า ที่อยู่IP ของเครื่องปลายทาง สำหรับ TCP นี้จะได้ค่าเมื่อการติดต่อเสร็จแล้วเท่านั้น ส่วน UDP จะคืนค่านี้ได้เมื่อเกิดเหตุการณ์ *DataArrival* , ซึ่งตอนนั้นก็จะได้รับค่า ที่อยู่ IP ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			เครื่องที่ส่งข้อมูลมาด้วย
RemotePort	*Long*	ไม่ใช่	จะคืนค่าหรือ ติดตั้ง ค่าพอร์ต เพื่อที่เราจะส่งข้อมูลไป
SocketHandle	*Long*	ใช่	คืนค่าที่ตรงกับที่การใช้ SocketHandle
State	*Integer*	ใช่	คืนค่าสถานะของตัวควบคุม, ซึ่งจะมีเป็นแต่ละชนิดไป

คุณลักษณะสถานะ (State Property)

คุณลักษณะสถานะนี้ จะส่งค่าสถานะของวินซ็อกเป็นค่าๆออกมาว่าขณะนั้นวินซ็อกตัวนั้นอยู่ในสถานะใด อาจจะเอาสถานะตัวนี้ไปเป็นเงื่อนไขในการเขียนโปรแกรมเพื่อป้องกันการเกิดการผิดพลาดที่เกี่ยวกับข้อบังคับของวินซ็อกได้ หรือจะใช้เพื่อเช็คดูว่าขณะนี้กำลังทำอะไรอยู่เพื่อที่เราจะได้เขียนโปรแกรมได้ว่าถ้าเป็นอย่างนี้อยู่จะทำอะไรได้บ้าง ซึ่งโดยที่รายละเอียดทั้งหมดจะระบุอยู่ที่ตารางนี้

ตาราง 2.5.5 คุณลักษณะสถานะ

ค่า Constant	*ค่า*	*ความหมาย*
SckClosed	0	ปิด เป็นค่าปกติ
SckOpen	1	เปิด ซ็อกเกต
SckListening	2	การรอรับรู้สำหรับการเชื่อมต่อ
SckConnectionPending	3	การร้องขอการเชื่อมต่อมาถึงแล้วแต่ยังไม่เสร็จ
SckResolvingHost	4	Host ทำตามความต้องการ
SckHostResolved	5	ความต้องการของ Host เป็นจริง
SckConnecting	6	การร้องขอการติดต่อเริ่มทำ แต่ ยังไม่เสร็จ
SckConnected	7	การเชื่อมต่อสำเร็จ
SckClosing	8	เริ่มต้นการปิด
SckError	9	มีการผิดพลาดเกิดขึ้น

Methods ของ Winsock Control

Methods นี้จะเป็นวิธีการที่จะให้วินซ็อกทำตามความต้องการของเราในการเขียนโปรแกรม การกำหนดก็จะมีลักษณะที่จะขึ้นต้นด้วยชื่อของวินซ็อกแล้วตามด้วยวิธีที่จะให้วินซ็อกทำ เช่น ต้องการให้วินซ็อกที่ชื่อ winsock1 สร้างการเชื่อมต่อ ก็จะใช้คำสั่งดังนี้ Winsock1.connect

การที่จะใช้ Methods นี้ส่วนใหญ่จะต้องมีการกำหนด คุณสมบัติ ให้กับวินซ็อกก่อนจึงต้องระวังด้วยว่าเราได้กำหนดให้มันหรือยัง ตัวอย่างก็เช่นการที่เราจะใช้ Winsock1.connect ได้นั้นต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการกำหนดหมายเลข IP และ หมายเลข Port ปลายทางให้กับ วินซ็อกก่อน เป็นต้น ซึ่งรายละเอียดต่างๆของ Methods ก็อยู่ที่ตารางข้างล่าง

ตาราง 2.5.6 Methods ของวินซ็อกคอนโทรล

Methods	**พารามิเตอร์**	***Return Value***	**คำอธิบาย**
Accept	*RequestID*	Void	สำหรับการติดต่อแบบ TCP เท่านั้นใช้ Methods นี้ ในการยอมรับการติดต่อเมื่อเกิดเหตุการณ์ *ConnectionRequest*
Bind	*LocalPort* *LocalIP*	Void	ทำการเชื่อมต่อ ซ็อกเก็ตให้กับ *LocalPort* และ *LocalIP* ใช้ *Bind* ถ้ามี Network หลายตัว *Bind* จะต้องถูกเรียกก่อน *Listen*
Close	None	Void	ปิดการติดต่อ หรือ ปิดการรอรับการติดต่อ
Connect	*RemoteHost* *RemotePort*	Void	สร้างการเชื่อมต่อของ TCP ไปยังหมายเลข *RemotePort* ที่อยู่บน *RemoteHost*
GetData	*Data* *Type* *MaxLen*	Void	รับเอาข้อมูลที่ค้างอยู่เข้ามา มี *Type* และ *MaxLen* เป็นพารามิเตอร์ควบคุม พารามิเตอร์ *Type* จะใช้กำหนดชนิดของข้อมูลสำหรับอ่าน พารามิเตอร์ *MaxLen* จะใช้กำจำนวนไบต์ หรือตัวอักษรที่รับเข้ามา *GetData* จะข้ามพารามิเตอร์ *MaxLen* สำหรับชนิดอื่นๆที่มากกว่า ไบต์อะเรย์ และ string.
Listen	None	Void	สร้าง ซ็อกเก็ตและวางมันไว้ในโหมดการรอรับการติดต่อ *Listen* จะใช้ สำหรับการติดต่อแบบ TCP เท่านั้น
PeekData	*Data* *Type* *MaxLen*	Void	คุณสมบัติเหมือนกับ *GetData* ยกเว้นแต่ว่าข้อมูลไม่ได้ถูกย้ายจาก Buffer ของระบบ
SendData	*Data*	Void	ส่งข้อมูลไปยังเครื่องปลายทาง ถ้า UNICODE string ผ่าน มันก็จะถูกแปลงเป็น ANSI string ก่อน จำไว้ว่าต้องใช้ ไบต์อะเรย์สำหรับข้อมูลไบนารี่เสมอ

Events ของวินซ็อก

Event คือเหตุการณ์ที่เกิดขึ้นเมื่อมีลักษณะของซ็อกเก็ตจากภายในและภายนอกเปลี่ยนไปจากเดิม เช่นการเข้ามาของข้อมูล การทำการติดต่อสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะใช้ประโยชน์ของ Events ในการที่เรากำหนดให้กับโปรแกรมว่าเราต้องการให้โปรแกรมทำอะไรเมื่อเกิดเหตุการณ์นี้ขึ้น เช่น สมมุติว่า เมื่อมีข้อมูลเข้ามาถึงซ็อกเก็ต (ในที่นี้คือ DataArrival) ของเราให้เรารับข้อมูลนี้ไว้ (โดยใช้ GetData Method) ซึ่งแต่ละ Event ก็มีดังตารางข้างล่างนี้

ตาราง 2.5.7 Event ของวินซ็อก

Event	*Arguments*	*คำอธิบาย*
Close	ไม่มี	เกิดขึ้นเมื่อขาดการติดต่อกันกับเครื่องปลายทางที่เราติดต่อด้วย
Connect	ไม่มี	เกิดขึ้นหลังจาก *Connect* method ติดต่อได้สำเร็จแล้ว
ConnectionRequest	*RequestID*	เกิดขึ้นเมื่อเครื่องปลายทางต้องการจะติดต่อด้วย
DataArrival	*BytesTotal*	เกิดขึ้นเมื่อมีข้อมูลใหม่เข้ามา
Error	*Number* *Description* *Scode* *Source* *HelpFile* *HelpContext* *CancelDisplay*	เมื่อ Winsock มีข้อผิดพลาด Event นี้ก็จะถูกสร้างขึ้น
SendComplete	ไม่มี	เกิดขึ้นเมื่อคำสั่งการส่งทำได้สำเร็จเรียบร้อยแล้ว
SendProgress	*BytesSent* *bytesRemaining*	เกิดขึ้นขณะที่ข้อมูลทำการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 โปรแกรมวิชวลเบสิก

เป็นที่ยอมรับในวงการกันดีว่าตั้งแต่วินโดวส์ 95 และวินโดว์ NT เป็นต้นมา วินโดวส์นั้นได้ช่วยให้คอมพิวเตอร์มีประสิทธิภาพในการทำงานสูงกว่าดอส ดังนั้นในคอมพิวเตอร์ที่ใช้วินโดวส์ ถ้าเขียนโปรแกรมสำหรับดอสเพื่อรันในดอสของวินโดวส์ก็เท่ากับไม่ได้ใช้วินโดวส์ที่มีอยู่เลย เพราะโปรแกรมจะลดความสามารถของวินโดวส์ให้เหลือเท่ากับคอมพิวเตอร์ที่ใช้ดอสเท่านั้นเพราะฉะนั้นสำหรับคอมพิวเตอร์ที่ใช้วินโดวส์ การเขียนโปรแกรมเพื่อรันในวินโดวส์จึงเป็นทางเดียวที่จะทำให้เกิดประโยชน์สูงสุด 2 ประการคือ

1. โปรแกรมสำหรับวินโดวส์จะนำความสามารถของวินโดวส์มาใช้ได้อย่างเต็มที่
2. การเขียนโปรแกรมสำหรับวินโดวส์ทำได้ง่ายกว่าการเขียนโปรแกรมสำหรับดอส

โปรแกรมหรือแอปพลิเคชั่นในวินโดวส์นั้นได้มีวิธีการเขียน 2 แบบ คือการเขียนโปรแกรมแบบวิชวลหรือวิชวลโปรแกรมมิ่ง (Visual Programming) และการเขียนโปรแกรมแบบนอนวิชวลหรือนอนวิชวลโปรแกรมมิ่ง (None Visual Programming) "วิชวล หมายถึง มองเห็นการเขียนโปรแกรม หมายความว่า การเขียนแบบนี้เราจะมองเห็นส่วนประกอบต่าง ๆ ของโปรแกรมนั้นทันที" ส่วนการเขียนโปรแกรมแบบนอนวิชวลเป็นการเขียนโปรแกรมด้วยอักษรและเครื่องหมายซึ่งจะยังไม่เห็นส่วนประกอบของโปรแกรมในทันทีแต่สามารถจะมองเห็นส่วนประกอบต่าง ๆ ของโปรแกรมได้ในภายหลังคือตอนรันโปรแกรม ดังนั้นการเขียนโปรแกรมแบบวิชวลจึงช่วยให้ผู้เขียนโปรแกรมสามารถเขียนได้สะดวกและรวดเร็วกว่าในการทำงานด้านโปรแกรมมิ่งนั้นนับตั้งแต่เริ่มต้น ในการพัฒนาโปรแกรมต่าง ๆ นั้นจะมีโปรแกรมภาษาต่าง ๆ มากมายที่ใช้ในการพัฒนางานหรือสร้างเป็นโปรแกรมประยุกต์ขึ้นมาใช้งานสำหรับการทำงานในองค์กรหรือพัฒนาขึ้นมาใช้งานส่วนตัว แต่โดยส่วนใหญ่แล้วโปรแกรมเหล่านี้จะเป็นโปรแกรมที่ต้องใช้ความจำเป็นเลิศเกือบทั้งสิ้น เนื่องจากโปรแกรมเหล่านั้นโดยส่วนใหญ่จะเป็นโปรแกรมประเภทการเขียนโดยการขึ้นคำสั่งเฉพาะต่าง ๆ ในการสร้างหรือควบคุมการทำงานของส่วนต่าง ๆ ที่ต้องการโดยเรียกการใช้งานโปรแกรมเหล่านี้ว่า "การโค๊ดดิ้ง" แต่สำหรับโปรแกรม Visual Basic 6.0 นั้นโปรแกรมพัฒนาที่การผสมผสานกันระหว่างการโค๊ดดิ้งและจับวาง (แดรกแอนด์ดรอป) นั่นก็คือในการทำงานนั้นสามารถที่จะกำหนดหรือสร้าง Object ต่าง ๆ โดยการใช้เครื่องมือต่าง ๆ ที่ตัวโปรแกรมมีมาให้โดยไม่ต้องเขียนคำสั่งเพื่อสร้าง Object ต่าง ๆ เหล่านั้นขึ้นมาใช้งานและยังสามารถที่จะเขียนคำสั่งเพื่อใช้ในการสร้างเงื่อนไขพิเศษอื่น ๆ ที่ใช้ในการทำงานได้อีกด้วย

2.6.1 การเปิดโปรแกรมขึ้นมาใช้งาน

ในการเปิดโปรแกรมขึ้นมาใช้งานนั้นสามารถที่จะทำได้โดยวิธีการดังต่อไปนี้

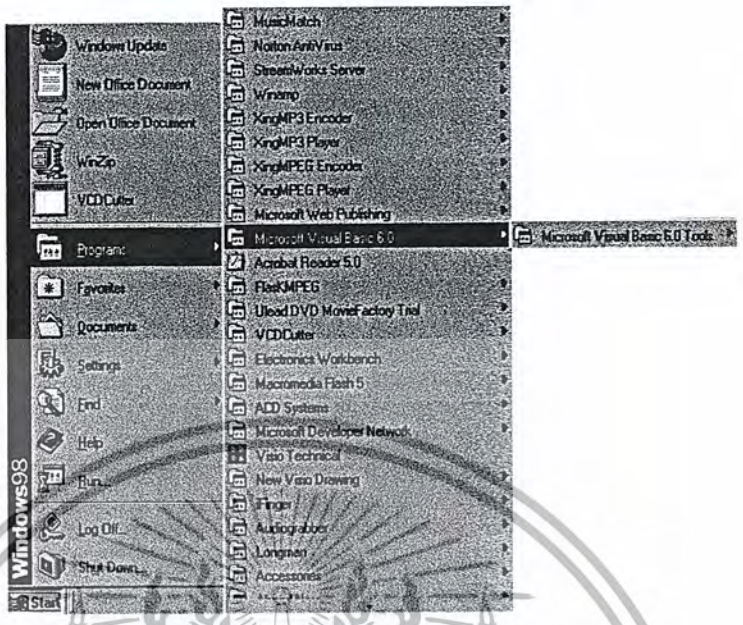
1.1 คลิกปุ่ม

1.2 เลือกเมนู Programs

1.3 เลือกเมนูย่อย Microsoft Visual Studio 6.0

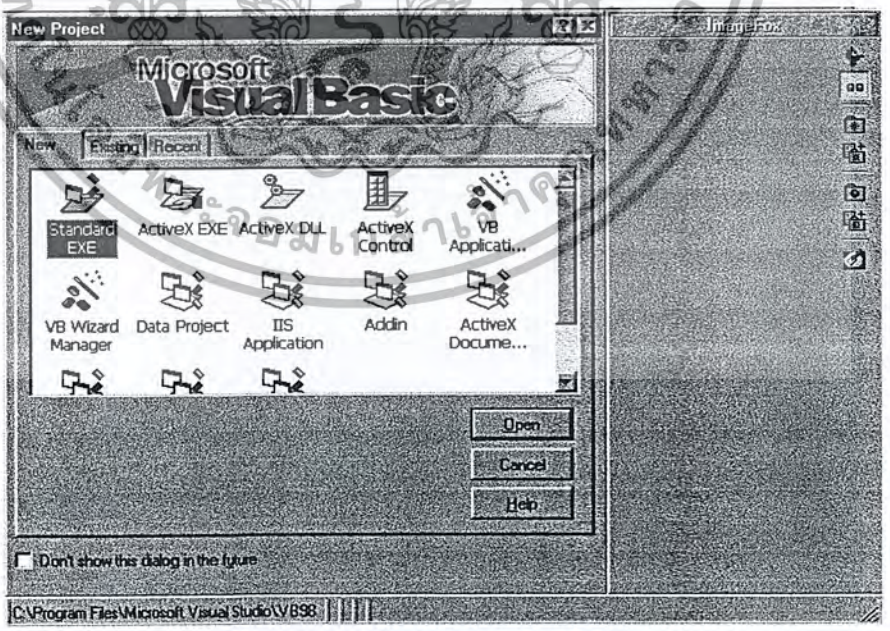
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 คลิกไอคอน Microsoft Visual Basic 6.0



รูปที่ 2.7 ขั้นตอนการเปิดโปรแกรม Microsoft Visual Basic 6.0

เมื่อได้เปิดโปรแกรม Microsoft Visual Basic 6.0 ขึ้นมาแล้วนั้นจะปรากฏดังรูป และสำหรับส่วนประกอบในการทำงานต่าง ๆ ที่ใช้ในการทำงานนั้นจะได้



รูปที่ 2.8 เมื่อเริ่มเปิดโปรแกรม Microsoft Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปนั้นจะเห็นว่าจะปรากฏไดอะล็อกบ๊อกซ์ New Project ขึ้นมาก่อนที่จะเข้าไปใช้งานโปรแกรมซึ่งในไดอะล็อกบ๊อกซ์ New Project นั้นจะประกอบด้วย 3 ส่วนด้วยกันคือ

New

ส่วนนี้เป็นส่วนที่ใช้ในการเลือกสร้างงานใหม่ขึ้นมา ซึ่งจะมีรูปแบบของการสร้างงานให้เลือกใช้หลากหลายรูปแบบด้วยกัน

Existing

ส่วนนี้จะเป็นส่วนที่ใช้ในการเปิดงานที่ได้สร้างขึ้นจากโปรแกรม Visual Basicในเวอร์ชันต่าง ๆ ที่มีอยู่แล้วซึ่งเก็บไว้ในที่ต่าง ๆ ที่ต้องการขึ้นมาใช้งาน ได้ตามต้องการ

Recent

ในส่วนนี้จะเป็นการเลือกเปิดไฟล์ที่ได้เคยเปิดขึ้นมาใช้งานแล้วปิดลงไป ซึ่งในส่วนนี้จะปรากฏชื่อไฟล์ต่าง ๆ ที่ได้เคยเปิดขึ้นมาใช้งานแล้ว

2.6.2 คุณสมบัติและข้อดีของ Visual Basic

ในVisual Basic มีคุณสมบัติที่ดีหลายประการกล่าวคือ ในการเขียนโปรแกรมแบบเดิมนั้นจะต้องมานั่งออกแบบหน้าจอระบุตำแหน่งการแสดงผลคิดหาขั้นตอนการทำงานและอื่น ๆ จากนั้นจึงทำการเขียนโปรแกรม แต่ใน Visual Basic จะใช้หลักการของภาพและการมองเห็น โดยเริ่มจากออกแบบวินโดวส์ย่อยหรือที่ใน Visual Basic เรียกว่า ฟอร์ม ในฟอร์มจะประกอบด้วยสิ่งต่าง ๆ ที่จะทำงานด้วยหรือเรียกว่าเป็น Object เช่น ข้อความ, ช่องรับข้อความ, Scroll Bar,หรือปุ่ม(Button) เมื่อกำหนดสิ่งต่าง ๆ เหล่านี้ครบตามต้องการแล้วจึงระบุว่าองค์ประกอบแต่ละอย่างทำงานอย่างไร โดยเขียนโปรแกรมย่อย ปะเข้าไปกับ Object เหล่านี้ ที่ต้องทำแบบนี้ก็เพราะว่าการทำงานใน Windows เป็นแบบที่เรียกว่า Event-Driven คือขึ้นกับเหตุการณ์ (Even) การเขียนโปรแกรมแบบเดิมคือ สั่งงานตามลำดับยุ่งยากมาก

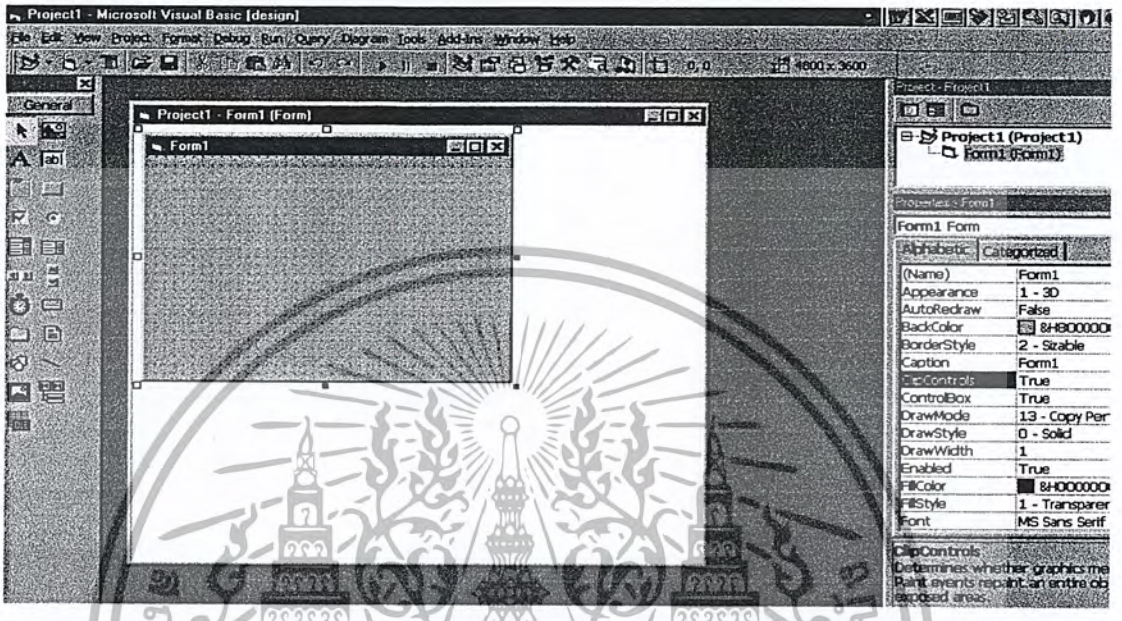
Visual Basicนี้ยังคงไว้ด้วยข้อดีต่าง ๆ ของ Microsoft Quick Basic และยังมีการเพิ่มเติมคุณสมบัติหลากหลายที่จะสนับสนุนให้ตัวมันเองเป็นโปรแกรมพื้นฐาน สำหรับการพัฒนาโปรแกรมบน Microsoft Windows อีกด้วย ตัวอย่างเช่น กราฟฟิกเอาต์พุตที่สามารถถูกส่งออกไปยังส่วนต่าง ๆ ของวินโดวส์หรือแม้กระทั่งส่งไปยังเครื่องพิมพ์ สามารถเลือกสีสำหรับงานกราฟฟิกได้มากกว่า 16 ล้านเฉดสี (โดยวินโดวส์จะจัดการแสดงผลกราฟฟิกนั้นตามที่ต้องการ หรือจะลดลงมาได้เท่าที่ฮาร์ดแวร์ของเครื่องนั้น ๆ จะสนับสนุนในการแสดงผลได้) โดยที่ไม่ต้องกังวลในส่วนนี้ว่าจะมีกระบวนการในการจัดการอย่างไรไม่ว่าในตอนนี้หรือต่อไปในอนาคต

ข้อดีเหนือ Quick Basic อีกอย่างหนึ่งก็คือ การจัดการตัวแปรใน Visual Basic มีกฎเกณฑ์ซึ่งง่ายในการเข้าใจและจดจำ เพราะว่าถูกพัฒนาให้ง่ายและมีประสิทธิภาพโดยที่โปรแกรมของ Visual Basic จะประกอบไปด้วยไฟล์ 2 แบบ คือ From และ Module ยกเว้นเมื่อ Declare globally ที่อื่น ตัวแปรและค่าคงที่ในโปรแกรมย่อยและฟังก์ชั่นนั้นจะเป็น Local สำหรับกระบวนการที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 ส่วนประกอบต่าง ๆ ที่ใช้ในการทำงาน

เมื่อทำการเปิดโปรแกรมขึ้นมาใช้งานแล้วนั้นจะปรากฏส่วนประกอบต่าง ๆ ของโปรแกรมที่ควรทราบก่อนที่จะเริ่มใช้งานโปรแกรมในการสร้างงานต่าง ๆ เพราะจะทำให้เลือกใช้เครื่องมือหรืออุปกรณ์ในส่วนต่าง ๆ ได้สะดวกและรวดเร็วมากยิ่งขึ้น



รูปที่ 2.9 ส่วนประกอบต่าง ๆ ของโปรแกรม Visual Basic 6.0

Title Bar

จะเป็นส่วนที่แสดงชื่อของ Application ที่กำลังเปิดใช้งานอยู่ขณะนั้น

Minimize Button

เป็นปุ่มที่ใช้สำหรับคลิกเพื่อหดหรือซ่อนส่วนของ Window หรือหน้าต่างทำการไว้ โดยจะปรากฏหรือแสดงที่หน้าจอเฉพาะส่วนของ Title Bar ของหน้าต่างดังกล่าวไว้และหากต้องการแสดงหรือ Show หน้าต่างอีกครั้งให้ใช้เมาส์คลิกซ้ำที่ปุ่มเดิมซึ่งปุ่มดังกล่าวจะอยู่ทางด้านขวาของ Title Barของหน้าต่างนั้น ๆ

Maximize Button or Restore Button

จะเป็นปุ่มที่ใช้สำหรับคลิกเพื่อขยายขนาดใหญ่ขึ้นจนเต็มหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Close Button

เป็นปุ่มสำหรับคลิดเพื่อปิดหรือจบการทำงานของหน้าต่างนั้น ๆ

Menu bar

เป็นแถบของเมนูหลักที่บรรจุคำสั่งย่อยหลายคำที่ใช้ในการทำงานไว้ สามารถเขียนใช้งานเมนูได้โดยการใช้เมาส์คลิกที่ชื่อของเมนูที่ต้องการบน Menu Bar หรืออาจเรียกเมนูดังกล่าวได้โดยการกดปุ่มที่คีย์บอร์ดดังนี้ กดปุ่ม Alt ค้างไว้พร้อมกับกดตัวอักษรที่ขีดเส้นใต้ของชื่อเมนูที่ต้องการเปิดตังตัวอย่างต่อไปนี้ หากต้องการใช้คำสั่งย่อยที่อยู่ในเมนู และที่ชื่อของเมนูบาร์จะมีเส้นขีดไว้ที่ใต้ตัวอักษร ดังนั้นในการเรียกใช้คำสั่งย่อยในเมนูดังกล่าวจำเป็นต้องเปิดเมนูก่อน โดยกดปุ่มที่คีย์บอร์ดดังนี้ Alt+F

Tool Bar

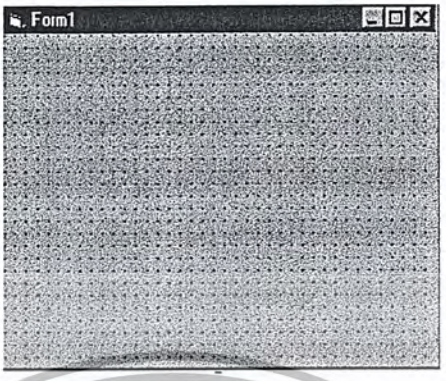
เป็นส่วนที่ปรากฏอยู่ด้านใต้ของเมนูบาร์ ซึ่งเป็นส่วนที่รวบเอาไอคอนเล็ก ๆ มากมายเอาไว้ ซึ่งแต่ไอคอนเหล่านี้ก็เปรียบเสมือนคำสั่งหนึ่ง ๆ ของเมนู ดังนั้นไอคอนในส่วนนี้จึงถูกออกแบบเพื่อให้การเลือกใช้คำสั่งของเมนูรวดเร็วและมีลักษณะที่สื่อความหมายกับผู้ใช้มากขึ้น

Tool Box

เป็นหน้าต่างที่รวบรวมเอาเครื่องมือต่าง ๆ สำหรับใช้ในขณะออกแบบฟอร์ม ซึ่งการรวมเอาเครื่องมือตัวใดให้กับฟอร์มผู้ใช้สามารถทำได้โดยการคลิ๊กที่คอนโทรลนั้น เพื่อเลือกแล้วนำมา

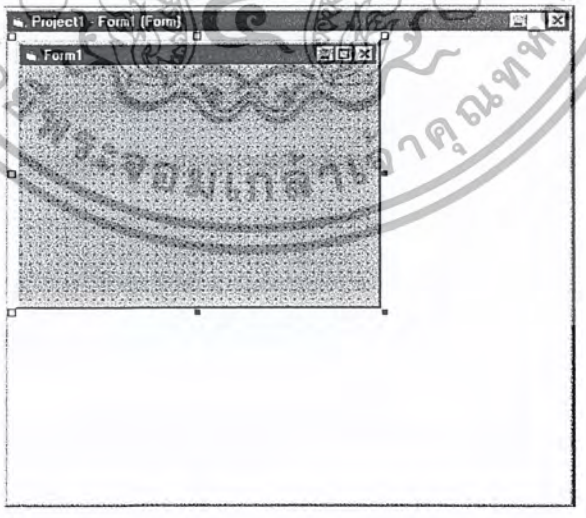
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วางลงในฟอร์มโดยวิธีการลากแล้ววาง หรือโดยวิธีการดับเบิลคลิ๊กที่ไอคอนคอนโทรลในทูลบ๊อกซ์ ก็ได้ ซึ่ง Visual Basic ก็จะทำการวางคอนโทรลลงในฟอร์มปัจจุบันโดยอัตโนมัติ



Form

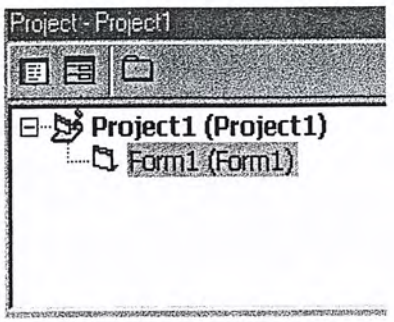
ในการแสดงหน้าต่างเพื่อที่จะสื่อการทำงานกับผู้ใช้ ทั้งนี้เนื่องจาก Visual Basic ได้รับการออกแบบให้โปรแกรมเมอร์สามารถนำมาใช้ออกแบบแอพพลิเคชั่นในลักษณะของการสื่อด้วยรูปดังนั้น การสื่อการทำงานต่าง ๆ ระหว่างแอพพลิเคชั่นกับผู้ใช้จะต้องกระทำผ่านฟอร์ม โดยในโปรเจคหนึ่ง ๆ สามารถมีได้หลายฟอร์มและภายในฟอร์มต่างๆ ก็จะถูกใช้ในการบรรจุคอนโทรลต่างๆ เช่น text box, picture box หรือ image เอาไว้ ดังนั้นฟอร์มจึงทำหน้าที่เป็นตัวบรรจุ (container) และฟอร์มก็ยังเป็นออบเจ็กต์ตัวหนึ่งของ Visual Basic ที่อนุญาตให้ผู้ใช้แก้ไขคุณสมบัติได้ในขณะออกแบบจากหน้าต่าง ๆ คุณสมบัติและสามารถควบคุมพฤติกรรมต่าง ๆ ของฟอร์มได้โดยผ่านทางวิธีเช่นเดียวกับออบเจคอื่น ๆ



Project Container Window

คือบริเวณทั้งหมดของฉากด้วยหลังที่เป็นสีขาว ของ From ใช้สำหรับเป็นพื้นที่หรือหน้าต่างๆ ที่ใช้บรรจุ Form หรือ สร้าง Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Project Explore Window

เป็นหน้าต่างที่รวบรวมรายชื่อของฟอร์มโมดูลไฟล์ CustomControl(Class Module) หรือไฟล์ทรัพยากร (resource file) สำหรับการสร้างแอพพลิเคชันหนึ่ง ๆ ซึ่งการรวมเอาไฟล์เหล่านี้เข้ามาด้วยกันเพื่อสร้างแอพพลิเคชั่นภายใต้ Visual Basic/win เรียกว่า โปรเจค (project) หรือโครงการ ซึ่งโดยปกติ Visual Basic จะจัดเก็บไฟล์โปรเจคโดยจะมีนามสกุล .MAK โดยจะมีการจัดเก็บแยกออกจากไฟล์อื่น ๆ โดยเด็ดขาด

Form Layout Windows

เป็นส่วนที่ใช้กำหนดตำแหน่งของ Form ที่ต้องการแสดงบนหน้าจอและบนหน้าต่างที่ทำงานนอกจากนี้ในการกำหนดตำแหน่งของ Form จะสามารถเลือกกำหนดได้จากการกำหนดค่าตัวแปรต่างๆ หรือคุณสมบัติต่าง ๆที่เกี่ยวกับการจัดวางตำแหน่งได้ในส่วนของ Properties ได้เช่นกัน

Pop Up Menu or Shortcut Menu

เป็นส่วนของเมนูคำสั่งต่าง ๆ ที่จะปรากฏหรือแสดงเมื่อมีการกดปุ่มด้านขวาของเมาส์หรือที่เรียกว่า คลิกขวา

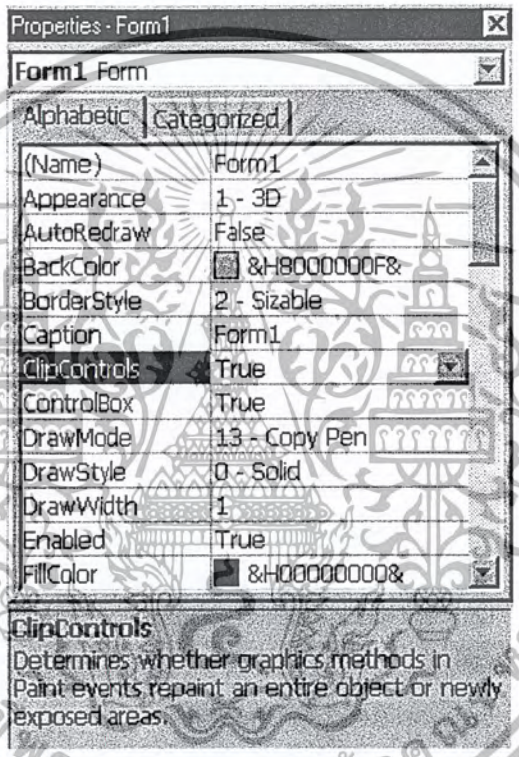
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tool Tips

เป็นส่วนที่ใช้บอกหรือแสดงชื่อของเครื่องมือหรือปุ่มต่าง ๆ ที่ตำแหน่งของ Mouse Pointer ชี้ อยู่ในขณะนั้น

Pointer

เป็นเครื่องมือที่ใช้ในการเลือกเมนูคำสั่งและเครื่องมือต่าง ๆ



Properties Windows

จะเป็นหน้าต่างที่รวบรวมคุณสมบัติทั้งหมดของฟอร์มหรือคอนโทรลเอาไว้ ซึ่งคุณสมบัติทั้งหมดที่ปรากฏในหน้าต่างนี้จะเป็นคุณสมบัติที่ผู้ใช้สามารถกำหนดค่าได้ในขณะออกแบบ เช่น Caption, Text, Left, Top หรือ Back Color เป็นต้น เมื่อผู้ใช้ทำการแก้ไขค่าคุณสมบัติต่าง ๆ ในหน้าต่างคุณสมบัตินี้ก็จะส่งผลต่อคอนโทรลตัวนั้นทันที ซึ่งบางคุณสมบัติสามารถแสดงให้เห็นการเปลี่ยนแปลงคุณสมบัติได้ทันที เช่น Left, Top หรือ Fore Color เป็นต้น ส่วนคุณสมบัติบางอย่างจะแสดงผลให้เห็นก็ต่อเมื่อผู้อ่านมีการรับแอพพลิเคชั่นเท่านั้น เช่น Enabled หรือ Visible เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 คอนโทรลของ VB/Win (custom control)

คอนโทรลเป็นเครื่องมืออย่างหนึ่งที่ช่วยให้การสร้างแอพพลิเคชั่นด้วย Visual Basic สามารถกระทำได้ง่ายและรวดเร็วยิ่งขึ้น ซึ่งในความเป็นจริงคอนโทรลก็คือไฟล์ไดนามิกลิงค์ไลบรารีทั่ว ๆ ไปของวินโดวส์โดยทั่วไปมักจะถูกเขียนด้วยภาษาซี หรือ Visual C++ เพียงแต่ไฟล์เหล่านี้จะมีนามสกุล .VBX หรือ .OCX ซึ่งเรียกว่า custom control เนื่องจากคอนโทรลเหล่านี้ได้รับการออกแบบเป็นพิเศษเพื่อให้สามารถใช้งานร่วมกับ Visual Basic ได้อย่างไม่มีปัญหาแล้ว ดังนั้นไฟล์เหล่านี้จะถูกย้ายไปจัดเก็บไว้ในไดเรคทอรีย่อย system ของวินโดวส์และไม่คอมแพตทิเบิลกับเวอร์ชันที่กำลังติดตั้งใหม่ ไฟล์ custom control ที่มีอยู่ในไดเรคทอรีย่อย system ของวินโดวส์นั้น ก็จะถูกย้ายไปจัดเก็บไว้ในไดเรคทอรีของ Visual Basic เวอร์ชันนั้น ๆ แทนผู้ใช้สามารถโหลดไฟล์คอนโทรล มารวมกับโปรเจคได้ด้วยการใช้คำสั่ง Add File ในเมนู File แต่ถ้าหากโปรเจคไม่มีการใช้คอนโทรลใด ๆ ก็ให้ผู้ใช้ลบคอนโทรลตัวนั้น ๆ ออกจากโปรกเจคซึ่งสามารถทำได้ด้วยการใช้คำสั่ง Remove Fileในเมนู File ทั้งนี้เพื่อจะให้การโหลดโปรเจคของ Visual Basicสามารถกระทำได้รวดเร็วยิ่งขึ้น เมื่อผู้ใช้นั้นได้มีการสร้างแอพพลิเคชั่นลงในไดเรคทอรีย่อย system ของวินโดสว์ของเครื่องผู้ใช้ด้วย (วิธีการสร้างแผ่นดิสก์สำหรับติดตั้งแอพพลิเคชั่นสามารถทำได้โดยใช้ Setup Wizard หรือชุดโค้ดที่มีในไดเรคทอรีย่อย System ในไดเรคทอรีของ Visual Basic)

คอนโทรลแต่ละตัวในความเป็นจริงก็คือโค้ดชุดหนึ่ง (Library) ที่สามารถทำงานได้เฉพาะด้านตามที่กำหนดและตัวคอนโทรลก็มีคุณสมบัติเฉพาะตัวที่ผู้อ่านสามารถแก้ไขได้ นอกจากนี้เมื่อมีการกระทำใด ๆ กับคอนโทรล เช่น การคลิ๊ก การเปลี่ยนค่า เป็นต้น เหตุการณ์เกี่ยวข้องของคอนโทรลก็จะถูกเรียก (Event-Driven) ซึ่งผู้ใช้สามารถที่จะเขียนโค้ดเพื่อควบคุมการทำงานเมื่อเกิดเหตุการณ์ได้โดยการเขียนโค้ดลงในส่วนโพรซีเดอร์เหตุการณ์ (Event Procedure) ซึ่งจะมีรายละเอียดคร่าว ๆ ของคอนโทรลแต่ละตัวดังนี้

Pointer เป็นเคอร์เซอร์ ที่ใช้สำหรับเลือก เลื่อนตำแหน่ง หรือขยายขนาดของฟอร์มหรือคอนโทรล ซึ่งคอน โทรลตัวนี้เป็นเครื่องมือของเอนไวรอนเมนท์ดังนั้นผู้ใช้จึงไม่สามารถนำไปใช้ในแอพพลิเคชั่น

Picture box ตัวคอนโทรลนี้ใช้สำหรับแสดงบิตแม็พ ไอคอน ไฟล์เมต้า วิธีการใช้ในแสดงผลกราฟฟิกเช่น Pset หรือ Line เป็นต้น หรือแสดงข้อความใดก็ได้ นอกจากนี้สามารถใช้บรรจุคอนโทรลอื่น ๆ ได้อีกด้วย (container)

Label ใช้แสดงข้อความที่ไม่สามารถแก้ไขได้โดยผู้ใช้ ยกเว้นแก้ไขโดยการเขียนโค้ดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command button ปุ่มคำสั่งซึ่งผู้ใช้สามารถเลือกได้โดยการคลิ๊กที่ปุ่ม

Frame Frame ใช้เป็นตัวบรรจุ (Container) คอนโทรลอื่น ๆ นอกจากนี้ยังใช้เป็นตัวรวมกลุ่ม (Group)คอนโทรลอื่น ๆ เช่น Option button เป็นต้น

Text box กรอบสี่เหลี่ยมสำหรับกรอกข้อความต่าง ๆ ซึ่งจะถูกใช้เป็นกล่องสำหรับตัวอักษรที่ถูกคีย์โดยผู้ใช้

Check box ปุ่มสำหรับเลือกสถานะถูกหรือผิดหรือสถานะใช่หรือไม่ใช่

Option button ปุ่มที่ถูกใช้รวมกันเป็นกลุ่มเพื่อให้ผู้ใช้ได้ใช้เป็นตัวเลือก (อย่างใดอย่างหนึ่ง)

Combo box คอนโทรลที่รวมเอาความสามารถของ text box และ list box เอาไว้ด้วยกัน

List box คอนโทรลที่ใช้ในการแสดงผลรายการของข้อความเพื่อให้ผู้ใช้สามารถเลือกได้

Horizontal scroll bar แถบเลื่อนในแนวนอนซึ่งสามารถกำหนดค่าในช่วงที่ต้องการเลือกได้

Vertical scroll bar แถบเลื่อนในแนวตั้งซึ่งผู้ใช้สามารถกำหนดค่าในช่วงที่ต้องการเลือกได้

Timer นาฬิกาที่นี้ถูกใช้ในการสร้างเหตุการณ์โดยที่อ้างอิงกับช่วงเวลา ซึ่งสามารถเทียบเคียงได้กับประโยค In Timer ของ Quick BASIC

2.6.5 ภาษาโปรแกรมของ Visual Basic

2.6.5.1 ตัวแปรและค่าคงที่

ตัวแปรใน Visual Basic มีความยาวได้ถึง 40 ตัวอักษร สามารถตั้งชื่อตัวแปรโดยผสมจากตัวอักษร ตัวเลขและ underscore (_) แต่ตัวแรกต้องเป็นอักษรเท่านั้นและห้ามตั้งชื่อตรงกับคำที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สงวนไว้ (Reserved word) ซึ่งได้แก่ฟังชันก์และคำสั่งต่าง ๆ ตัวแปรแบ่งออกเป็น 7 ชนิด ดังตารางที่ 2.6.1

ตารางที่ 2.6.1 ตัวแปรและค่าคงที่ในโปรแกรม Visual Basic

ชนิด	คำอธิบาย	ตัวอักษรบอกชนิด	ขอบเขต
Integer	2- byte integer	%	-32,768 จนถึง 32,767
Long	4- byte integer	&	-2,147,678 จนถึง 2,147,483,647
Single	4- byte floating-point Number	!	-3.402823E38 จนถึง -1.401298E-45
Double	8- byte floating-point Number	#	-1.797693134862232222D308 จนถึง –4.94065645841247D-324 (ค่าลบ) 4.94065645841247D-324 จนถึง 1.797693134862232222D308
Currency	8- byte Number with Fixed decimal point	@	-922337203668547705808 จนถึง 922337203668547705808
String	String of characters	$	0 จนถึงประมาณ 65,000 ตัวอักษร
Variant	Date/time} floating-Pint number, or string		Date valued : วันที่ 1 มากราคม ปี ค.ศ. 0000 ถึงวันที่ 31 ธันวาคม ปี ค.ศ. 9999

Integer

ไว้สำหรับเก็บค่าจำนวนเต็มในช่วง –32,768 ถึง 32,767 ตัวแปรชนิดนี้สามารถคำนวณได้เร็วมาก แต่มีข้อจำกัดด้านค่าที่เก็บได้ซึ่งอยู่ในขอบเขตที่แคบกว่าแบบอื่น ตัวแปรชนิดนี้ระบุด้วย % เช่น A%

Long

เป็นค่าจำนวนเต็มเช่นเดียวกับชนิด Integer แต่ขยายช่วงไปเป็น –2,147,483,648 ถึง 2,147,483,647 ใช้เนื้อที่ 4 ไบต์และใช้ตัวอักษร & เป็นตัวระบุชนิด เช่น A&

Single

ใช้เนื้อที่ในการเก็บ 4 ไบต์เก็บค่าตัวเลขที่เป็นทศนิยมซึ่งไม่ต้องการความละเอียดมากนัก (-3.102823E38 ถึง –1.401298E-45) ใช้ ! ในการระบุชนิด เช่น A!

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Double

คล้ายกับชนิด Single แต่สามารถเก็บค่าได้มากว่าและใช้เนื้อที่เก็บเป็น 2 เท่า ของ Single (8 ไบต์) อย่างไรก็ตามก็ใช้เวลาการคำนวณมากกว่าด้วย ระบุชนิดด้วย # เช่น A#

Currency

ใช้เก็บค่าทางการเงิน โดยเฉพาะเป็นแบบทศนิยมคงที่ 4 ตำแหน่ง ใช้เนื้อที่ 8 ไบต์ การระบุถึงชนิดใช้ตัวอักษร @ เช่น A@

String

ใช้เก็บข้อความหรือตัวอักษรซึ่งไม่สามารถนำไปคำนวณได้ ความยาวของข้อความคือ 65,5000 ตัวอักษร เนื้อที่ที่ใช้เก็บขึ้นอยู่กับขนาดของข้อความที่เก็บ ระบุชนิดด้วย $ เช่น a$

Variant

ตัวแปรชนิดนี้เพิ่งเริ่มมีใช้ใน Visual Basic 2.0 ถูกออกแบบมาใช้เก็บค่าต่าง ๆ กัน โดยสามารถเก็บค่าชนิดใดก็ได้ 6 ชนิดข้างต้น ไม่ว่าจะเป็นตัวเลข ตัวอักษร หรือวันที่ ข้อมูลชนิดนี้มีประโยชน์ในกรณีที่ต้องการประกาศตัวแปร ซึ่งอาจเก็บค่าได้หลายแบบโดยที่ไม่สามารถระบุได้แน่นอนว่าเป็นแบบใด ปัญหาคือจะทำงานช้ากว่าตัวแปรแบบอื่นเพราะ Visual Basic ต้องมีการตรวจสอบชนิดของตัวแปรประเภทนี้ทุกครั้งก่อนจะใช้งาน

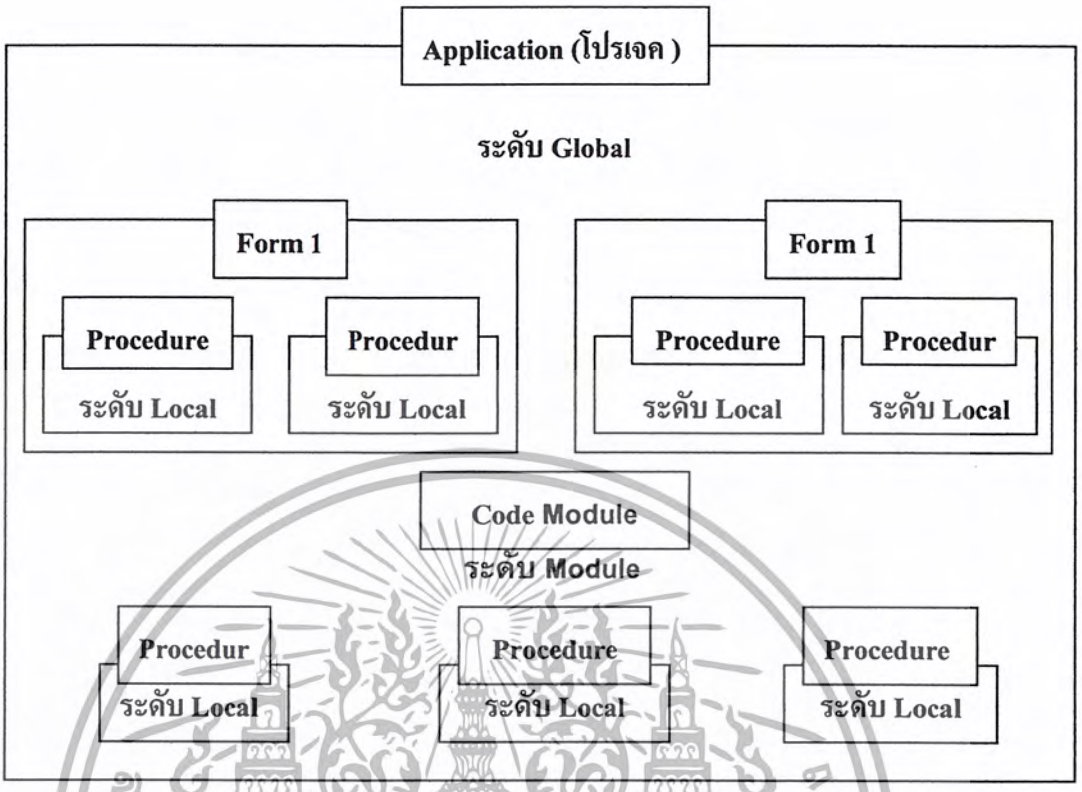
2.6.5.2 ขอบเขตของตัวแปร

ทั้งตัวแปรและค่าคงที่ที่ได้กล่าวถึงตอนต้น โดยปกติจะมีขอบเขตของการใช้งานหรือ ใช้งานได้ในแต่ละส่วนของโปรแกรมเท่านั้น ซึ่งขอบเขตของตัวแปร (Scope) นี้ แบ่งออกได้เป็น 3 ระดับ คือ

1. Local

ตัวแปรชนิดนี้จะเป็นที่รู้จักกันมาก ใช้ได้ในระดับโพรซีเยอร์หรือฟังก์ชันที่ประกาศใช้ตัวแปรนั่นเอง ส่วนมากมักจะใช้กับการประกาศตัวแปรสำหรับการคำนวณหรือจะใช้เก็บค่าชั่วคราว ตัวแปรในระดับ Local จะถูกยกเลิกไปเมื่อการทำงานของโพรซีเยอร์หรือฟังก์ชันนั้นสิ้นสุดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงขอบเขตของตัวแปรและค่าคงที่ในระดับต่างๆ

2. Module

ตัวแปรในระดับโมดูล สามารถอ้างอิงถึง สามารถใช้ร่วมกันได้หลายฟังก์ชันหรือหลายโพรซีเยอร์ซึ่งอยู่ในโมดูลนั้น ไม่ว่าจะเป็นในโมดูลของโปรแกรมที่ประกอบด้วยรูทีนย่อย ๆ หลาย ๆ รูทีน หรือในโมดูลของฟอร์มซึ่งอยู่ในรูทีนสำหรับทำงานกับ Object ในฟอร์ม

3. Global

ในตัวแปรระดับ Global นี้สามารถจะอ้างอิงได้จากจุดใด ๆ ก็ตามใน Application ทุกรูทีนหรือทุกส่วนจะสามารถนำไปใช้ได้การประกาศใช้จะเหมือนกับระดับโมดูล คือประกาศค่าคงที่ในส่วน Declaration เพียงแต่ว่าจะต้องมีคำ "Global" นำหน้าชื่อตัวแปร สำหรับในการประกาศค่าคงที่ในระดับ Global ก็ใช้คำว่า "Global"นำหน้าเช่นกัน

```
Global A As Integer
Global Const BAT_Rate = 70/100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5.3 โอเปอร์เรเตอร์

โอเปอเรเตอร์ คือสิงที่จะก่อให้การกระทำเพื่อให้ได้ผลลัพธ์ออกมาพบโอเปอร์เรเตอร์ได้ในนิพจน์ทั่ว ๆ ไป ซึ่งแบ่งออกเป็นกลุ่มโอเปอร์เรเตอร์ทางคณิตศาสตร์ การเปรียบเทียบ และทางตรรกะ ดังตารางที่ 2.6.2 โดยสัญลักษณ์ในวงเล็บคือ โอเปอร์เรเตอร์ที่ใช้ในนิพจน์

ตารางที่ 2.6.2 แสดงกลุ่มของโอเปอร์เรเตอร์

ทางคณิตศาสตร์	การเปรียบเทียบ	ทางตรรกะ
ยกกำลัง ( )	เท่ากัน (=)	Not
ค่าลบ (-)	ไม่เท่ากัน (≠)	And
คูณ,หาร (*, /)	น้อยกว่า (<)	Or
หารจำนวนเต็ม (\ )	มากกว่า (>)	Xor
Module (Mod)	น้อยกว่าหรือเท่ากับ (<=)	Eqv
บวก,ลบ (+ , -)	มากกว่าหรือเท่ากับ(>=)	Imp
เชื่อมข้อความ (String)(& หรือ +)	เหมือน (Like)	Is

2.6.5.4 ประโยคคำสั่ง

เป็นประโยคที่ส่งให้เกิดการทำงานตามที่ผู้เขียนโปรแกรมต้องการ ประโยคคำสั่งหลัก ๆ ได้แก่ ประโยคสำหรับกำหนดค่า ประโยคหมายเหตุ ประโยคประกาศ และประโยคเพื่อควบคุมลำดับการทำงานของโปรแกรมนอกเหนือจากนี้จะเป็นประโยคเพื่อการทำงานอื่น ๆ เช่น ทำงานทางกราฟฟิก เป็นต้น

1. ประโยคสำหรับกำหนดค่า

ใช้สัญลักษณ์ " =" เพื่อเป็นการกำหนดค่าทางขวามือของเครื่องหมายให้กับตัวแปรหรือคุณสมบัติ (Property) ทางซ้ายมือ เช่น

```
A = 5
Text1.Text = "Hi!"
```

2. ประโยคหมายเหตุ

สำหรับการใส่หมายเหตุต่อคำอธิบายการทำงานของโปรแกรม ซึ่งการใส่หมายเหตุในโปรแกรมนี้ถือว่าเป็นเรื่องสำคัญเนื่องจากสิ่งนี้จะช่วยป้องกันการหลงลืมของคนเขียนโปรแกรมเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรืออธิบายให้ผู้อ่านซึ่งต้องมาใส่การทำงานของโปรแกรมได้ง่ายขึ้น มิฉะนั้นจะพบว่าวันหนึ่งมีผู้ไม่เข้าใจโปรแกรมส่วนนี้ทำงานอย่างไรมีไว้เพื่ออะไร ซึ่งอาจเกิดกับตัวผู้เขียนโปรแกรมเองก็ได้ ประโยคหมายเหตุนี้อาจขึ้นต้นประโยคด้วยคำว่า "Rem" ที่ย่อมาจาก คำว่า "Remark" หรือใช้สัญลักษณ์

" ' " นำหน้าประโยคก็ได้ ดังตัวอย่าง

```
Rem This function is use to open database files
'This function is use to close database files
```

3. ประโยคประกาศ

ใช้ประกาศถึงตัวแปร ค่าคงที่ และฟังก์ชันหรือรูทีนย่อย สำหรับการประกาศตัวแปรและค่าคงที่ได้กล่าวไว้แล้วในข้างต้นว่าแบ่งออกเป็น Global, Module หรือ Local ส่วนการประกาศฟังก์ชันหรือรูทีนย่อยใช้คำสั่ง "declare"ซึ่งมีรูปแบบดังนี้

```
Declare Sub globalname Lib "libname" [Alias aliasname"]
[([[ByVal] variable [As type][,ByVal] variable [As type]]...])]
Declare Function globalname Lib "libname" [As type][,ByVal]
variable [As type]]...])] [As type]
```

2.6.5.5 ประโยคควบคุมลำดับการทำงาน

แบ่งออกเป็นหลายกลุ่มย่อยมีทั้งประโยคตัดสินใจ ประโยคการทำงานวนรอบ (loop) การกระโดดแบบมีเงื่อนไข รวมถึงคำสั่งจบการทำงานของโปรแกรมทั้งแบบชั่วคราวและถาวร ซึ่งจะขอกล่าวเรียงตามลำดับดังนี้

1. IF...Then...Else

เป็นคำสั่งตัดสินใจว่าจะทำงานหรือไม่ตามเงื่อนไขที่กำหนด สามารถกำหนดเพิ่มได้ว่าหากไม่เป็นไปตามเงื่อนไขให้ทำงานใด ซึ่งอยู่หลังคำว่า "Else" อีกทั้งยังกำหนดให้ตรวจสอบหลายเงื่อนไขได้ด้วย "Else"

```
If condition1 Then
    [Statementblock-1]
[Else If condition2 Then
    [Statementblock-2]]...
[Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Statementblock-n]]...

End If

2. Select Case

ในบางครั้งการเขียนประโยคตัดสินใจด้วย IF...Then...Else จะทำให้โปรแกรมเยิ่นเย้อและอ่านยาก ประโยค Select Case จะทำหน้าที่คล้ายกับ IF...Then...Else แต่ให้ความหมายกระชับกว่าโดยใช้กับค่าต่าง ๆ ของนิพจน์เดียวเท่านั้น

```
Select Case testexpression
[Case exp1
        [Statementblock-1]
[Case exp2
        [Statementblock-2]]...
[Case Else
        [Statementblock-n]
End Select
```

3. For ...Next

เพื่อกำหนดให้มีการทำงานซ้ำ ๆ กันเป็นจำนวนครั้งที่ต้องการ รูปแบบคำสั่งได้แก่

```
For counter = start to end [Step increment]
[Statements]
Next [counter]
```

ทำการกำหนดค่าเริ่มต้น (Start) ให้กับ Counter เพื่อเป็นตัวนับให้มีการทำงานในส่วน Statements ไปเรื่อย ๆ จนกว่าค่า end โดยในการทำงานแต่ละรอบจะเพิ่มค่า Counter ขึ้นตามค่าใน Increment

4. Do...Loop

ใช้เมื่อต้องการให้ทำงานวนรอบแบบมีเงื่อนไข ประโยค Do...Loop นี้สามารถกำหนดให้ตรวจสอบเงื่อนไขก่อนหรือหลังการทำงานในแต่ละรอบก็ได้โดยใส่ค่า "While" หรือ " Until" ข้างหลัง Do หรือ Loop ตามลำดับการใช้ while จะหมายถึงให้ทำงานวนรอบเมื่อนิพจน์ที่ตามมาให้ค่าจริง (True) คำทำงานจนกว่านิพจน์จะกลายเป็นเท็จ (False) หรือพูดอีกนัยหนึ่งคือทำงานไปจนกว่านิพจน์เป็นจริงนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Do [{While/Unit} condition]

[Statementblock]

Loop

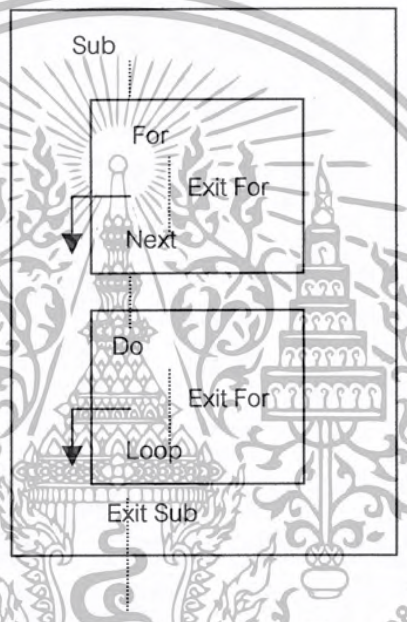
หรือ

Do

[Statementblock]

Loop [{While/Unit} condition]

5. หยุดการทำงานของโปรแกรม



จากรูปแสดงการทำงานจากส่วนของโปรแกรมแบบต่าง ๆ ด้วยคำสั่ง **Exit**

ใน Visual Basic มีทั้งคำสั่งในการหยุดการทำงานแบบชั่วคราวและถาวรรวมทั้งจบการทำงานในวนรอบ (Loop) มีฟังก์ชันหรือโพรซีเยอร์ด้วย การหยุดทำงานชั่วคราวใช้คำสั่ง Stop ซึ่งจะเป็นคำสั่งโดดไม่ต้องมีพารามิเตอร์ประกอบส่วนการหยุดโปรแกรมอย่างถาวรใช้คำสั่ง End ข้อแตกต่างของ 2 คำสั่งนี้คือ Stop จะไม่ยกเลิกตัวแปรและจะยังไม่ปิดไฟล์ที่เปิดค้างอยู่ให้ อีกทั้งยังสามารถสั่งให้ทำงานต่อไปภายหลังได้โดยเลือกหัวข้อ Continue จากเมนู Run หรือหัวข้อ Single Step ในเมนู Debug ส่วน END จะยกเลิกตัวแปรและปิดไฟล์ทั้งหมดที่เปิดค้างอยู่ในโปรแกรมนั้น

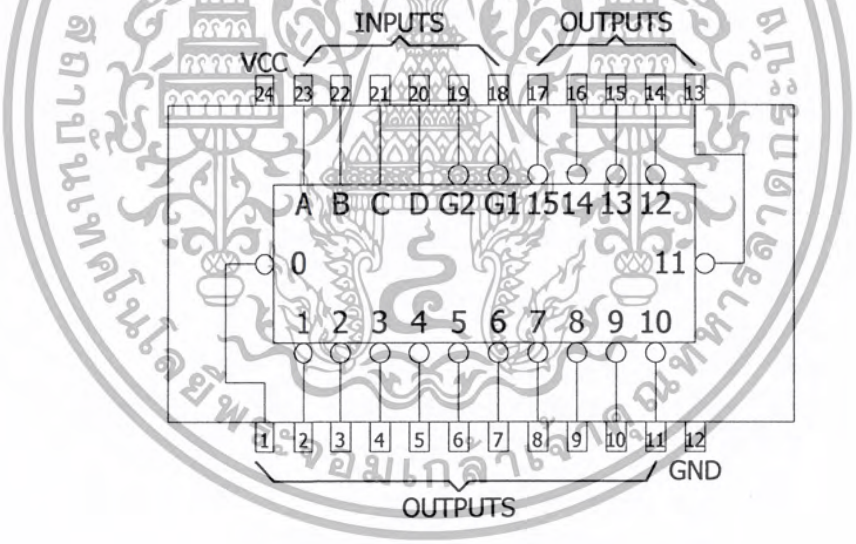
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการออกแบบ

แนวคิดเริ่มต้นในการออกแบบโครงงานวงจรการทำงานแผงแสดงผล LED เริ่มจากการออกแบบส่วนควบคุมการแสดงผล และภาคแสดงผล (Display Board) โดยจากหลักการที่กล่าวมาแล้ว ในภาคทฤษฎีนำมาออกแบบวงจรรวมทั้งหมด โดยการออกแบบทั้งหมดจะแยกอธิบายออกเป็นส่วนๆ ดังนี้

3.1 วงจรถอดรหัส

วงจรถอดรหัสนี้จะแบ่งออกเป็น 2 ส่วน คือ ส่วนที่ใช้ส่งข้อมูลในด้านหลัก และส่วนที่ใช้ควบคุมการสแกนทางด้านแถว โดยเลือกใช้ IC เบอร์ 74HC154 (4-Line-to-16-Line Decoder/Demultiplexer) ซึ่งเป็น IC ถอดรหัส เข้า 4 ออก 16 มีลักษณะของขาสัญญาณดังรูปที่ 3.1



รูปที่ 3.1 แสดงลักษณะขาสัญญาณของไอซีเบอร์ 74HC154

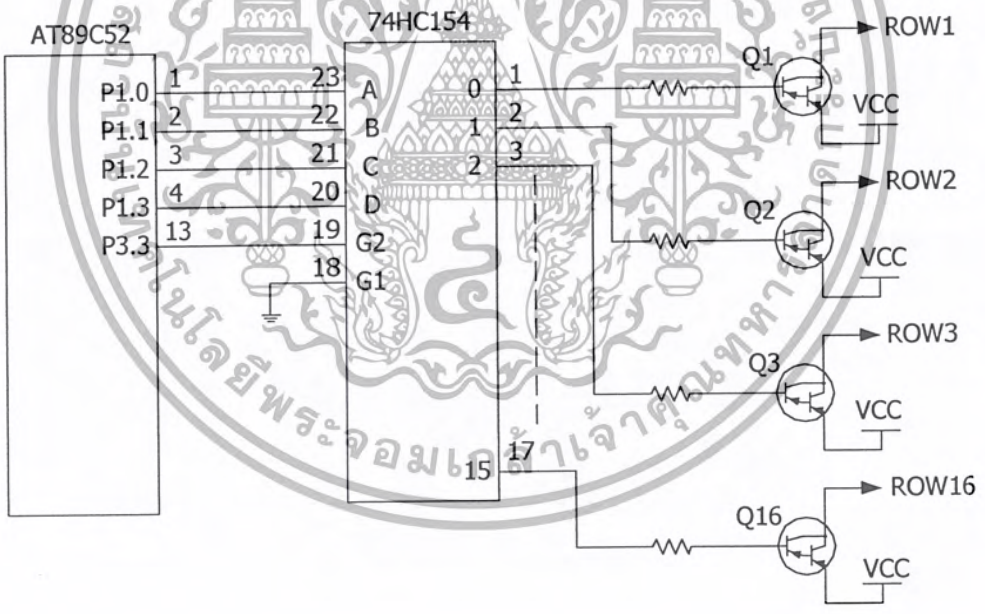
IC ตัวนี้ จะถูกควบคุมการทำงานโดย MCS-51 เบอร์ AT89C52 โดยการต่อใช้งานดูได้จากตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 การต่อใช้งานของ IC เบอร์ 74HC154 กับ ไมโครคอนโทรลเลอร์

ขาอินพุต ของ 74HC154	ขาสัญญาณของ AT89C52
อินพุต A (ขา 23)	P1.0 (ขา 1)
อินพุต B (ขา 22)	P1.1 (ขา 2)
อินพุต C (ขา 21)	P1.2 (ขา 3)
อินพุต D (ขา 20)	P1.3 (ขา 4)
ขาควบคุม G2 (ขา 19)	P3.3 (ขา 13)

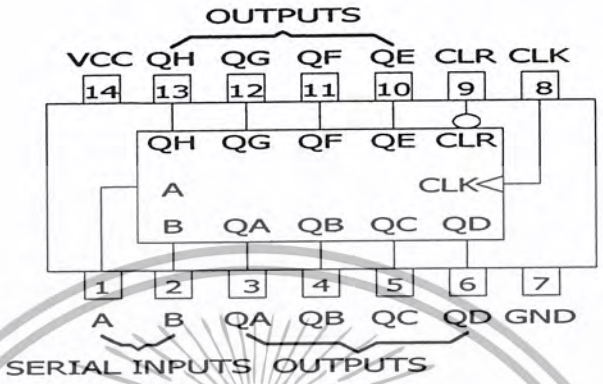
ส่วนขาควบคุม G1 ของ 74HC154 จะต่อลงกราวด์ และ ในส่วนของขาเอาท์พุต นั้น จะนำไปใช้ในการไบอัสทรานซิสเตอร์ เบอร์ BD679 ซึ่งเป็นทรานซิสเตอร์ที่ใช้ควบคุมการสแกนทางด้านแถว ซึ่งการต่อใช้งานแสดงได้ดังรูปที่ 3.2



รูปที่3.2 แสดงการต่อใช้งานของ 74HC154 กับ 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับส่วนที่ใช้ควบคุมการส่งข้อมูลแบบอนุกรมทางด้านหลัก จะเลือกใช้ IC เบอร์ 74HC164 (8-Parallel-Out Serial Shift Register) มีลักษณะของขาสัญญาณดังรูปที่ 3.3

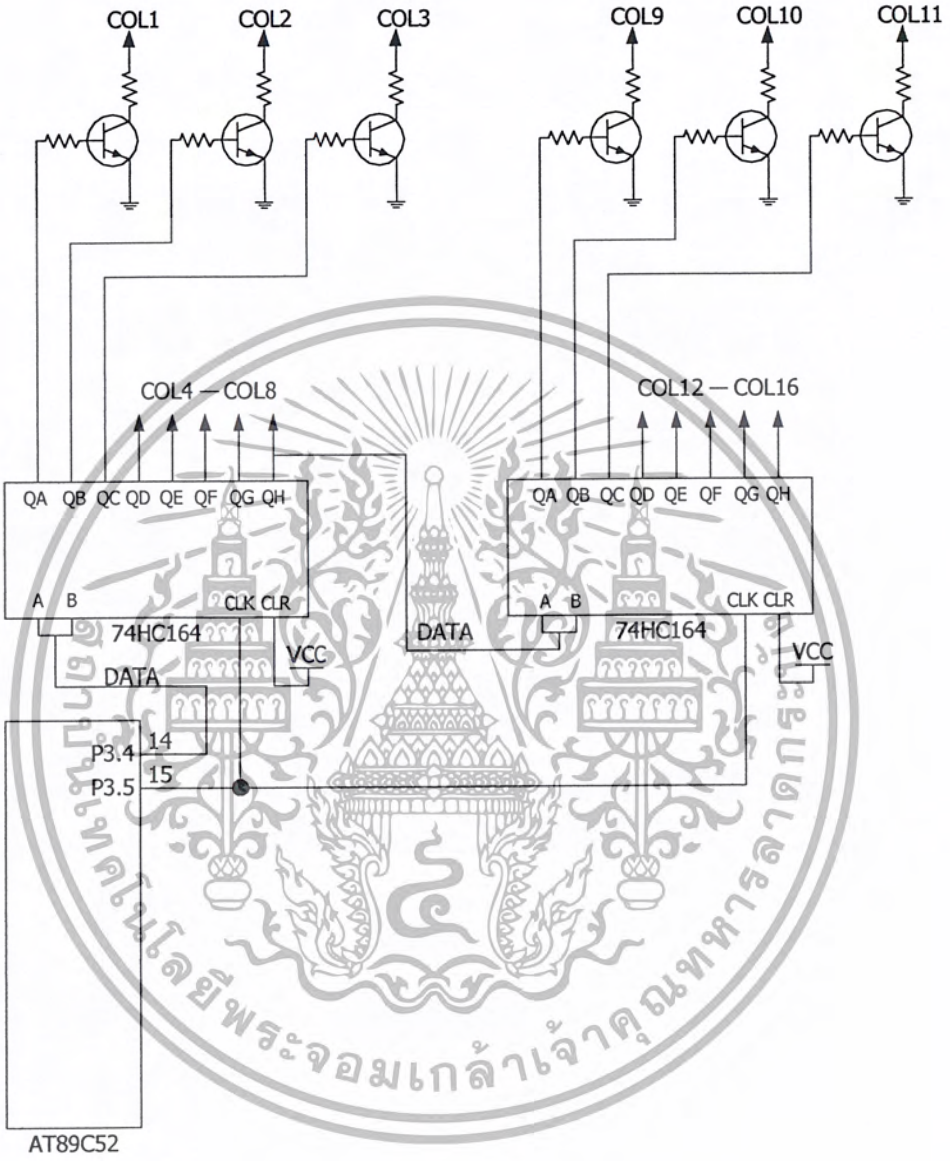


รูปที่ 3.3 แสดงลักษณะขาสัญญาณของไอซีเบอร์ 74HC164

การต่อใช้งานของ 74HC164 นั้น 74HC164 จะมีขาอินพุต 2 ขาคือ อินพุต A (ขา1) และอินพุต B (ขา 2) ซึ่งขาอินพุต 2 ขานี้จะต่อถึงกันแล้วนำไปต่อเข้ากับ ขาสัญญาณ P3.4 (ขา14) ของ ไมโครคอนโทรลเลอร์ ส่วนขา CLK (ขา 8) ต่อเข้ากับ ขาสัญญาณ P3.5 (ขา 15) และขา CLR (ขา 9) จะต่อกับ Vcc

การต่อใช้งานนั้น จะใช้ 74HC164 ทั้งหมด 8 ตัว ซึ่งแต่ละตัวจะมีขา เอาท์พุต 8 ขา ซึ่งขาเอาท์พุทจะใช้ในการไบอัสทรานซิสเตอร์ เบอร์ C1815 ซึ่งเป็นทรานซิสเตอร์ที่ใช้ควบคุมการสแกนทางคอลัมน์ โดยที่ ขาสัญญาณ P3.4 ของ ไมโครคอนโทรลเลอร์จะต่อเข้ากับขาอินพุต A และขาอินพุต B ของ 74HC164 เพียงตัวเดียวเท่านั้น คือ ตัวแรก ส่วน 74HC164 ตัวที่เหลือนั้น จะต่อใช้งานโดยที่ ขาเอาท์พุต QH (ขา 13) ของตัวแรก จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 2 ส่วนขาเอาท์พุต QH ของตัวที่ 2 จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 3 ส่วนขาเอาท์พุต QH ของตัวที่ 3 จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 4 ส่วนขาเอาท์พุต QH ของตัวที่ 4 จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 5 ส่วนขาเอาท์พุต QH ของตัวที่ 5 จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 6 ส่วนขาเอาท์พุต QH ของตัวที่ 6 จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 7 ส่วนขาเอาท์พุต QH ของตัวที่ 7 จะต่อกับขาอินพุต A และ ขาอินพุต B ของตัวที่ 8 ส่วนขา CLK นั้นจะต่อร่วมกันหมด โดยต่อกับขา สัญญาณ P3.5 ของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

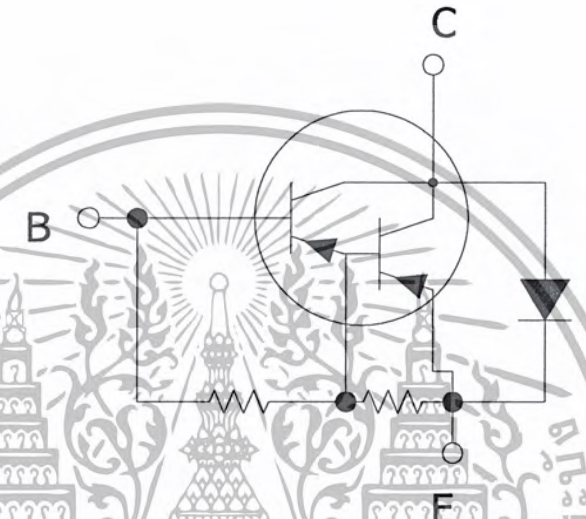


รูปที่ 3.4 แสดงการต่อใช้งานของ 74HC164 2 ตัว กับ 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การใช้ทรานซิสเตอร์ขับกระแส

การใช้ทรานซิสเตอร์ขับกระแสนั้นจะใช้ทรานซิสเตอร์เบอร์ BD 678 เป็นตัวจ่ายกระแสให้กับ LED ทางด้านโรว์ ซึ่งเป็นทรานซิสเตอร์แบบดาร์ลิงตัน (Darlington) สาเหตุที่เลือกใช้ทรานซิสเตอร์แบบดาร์ลิงตัน เพราะว่าสามารถจ่ายกระแสได้สูง เพราะว่าในด้านโรว์นั้นจะต้องใช้กระแสที่สูง โดยเลือกใช้ทรานซิสเตอร์เบอร์ BD 678 แบบ PNP



รูปที่ 3.5 วงจรภายในของ TR BD 678

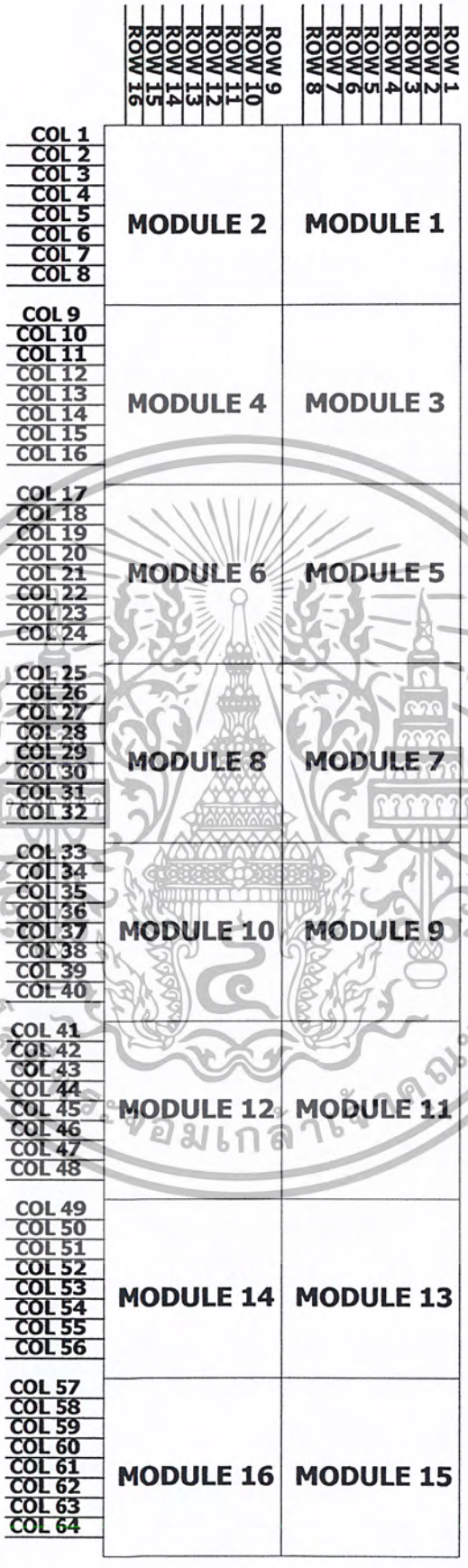
ส่วนทางด้านคอลัมน์นั้นจะใช้ทรานซิสเตอร์ เบอร์ C1815 แบบ NPN

ในการออกแบบนั้น สาเหตุที่ใช้ทรานซิสเตอร์ แบบ PNP ในด้านโรว์นั้น เนื่องจากว่า IC เบอร์ 74HC154 นั้น จะให้เอาท์พุตเป็นลอจิก 0 ออกมา ดังนั้นจึงต้องใช้ทรานซิสเตอร์แบบ PNP ส่วนทางด้านคอลัมน์ IC เบอร์ 74HC164 เป็น shift register ซึ่งจะให้เอาท์พุตเป็นลอจิก 1 จึงใช้ทรานซิสเตอร์แบบ NPN

3.3 ภาคแสดงผล

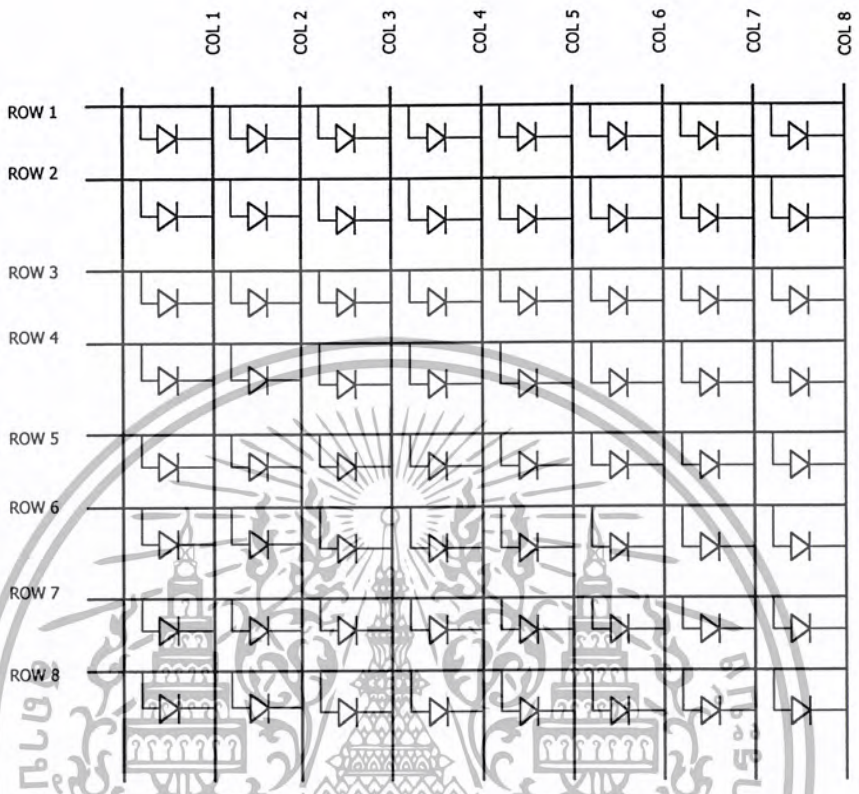
ในส่วนของภาคแสดงผลนี้ แผงแสดงผลจะมีขนาด 16×64 dot โดยได้เลือกใช้ LED แบบ DOT MATRIX ขนาด 8×8 dot จำนวน16 ตัวมาต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการต่อ DOT MATRIX ขนาด 16×64 dot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงการต่อ LED ภายใน 1 MODULE

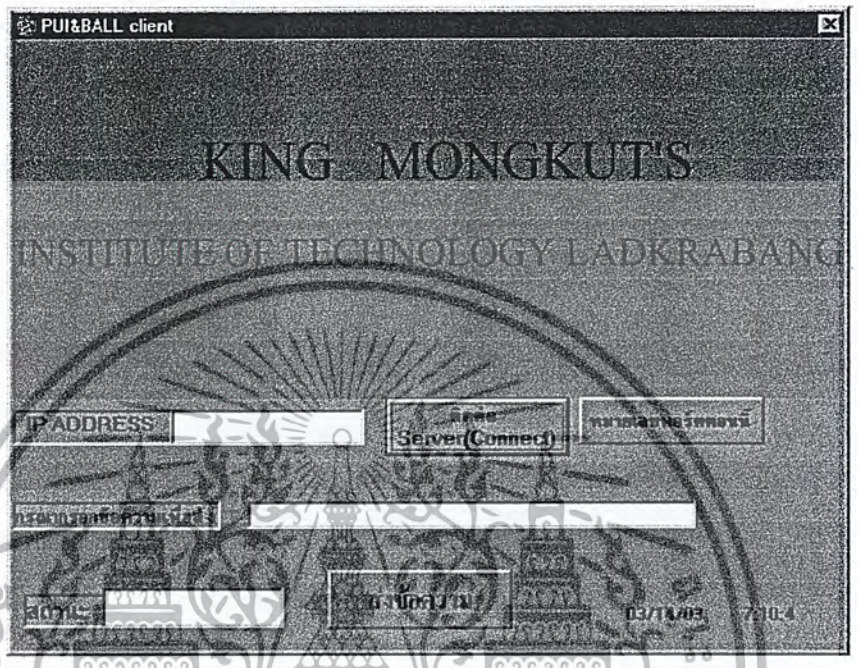
3.4 การออกแบบโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็น Client / Server

โปรแกรมที่ออกแบบในโครงงานนี้ สร้างขึ้นโดยใช้ VISUAL BASIC 6 เป็นโปรแกรมในการสร้างซึ่งแต่ละโปรแกรมซึ่งสร้างขึ้นโดย VISUAL BASIC 6 จะเรียกว่าโปรเจค (Project) โดยในหนึ่งโปรเจคจะประกอบด้วยไฟล์สกุล .frm ที่ทำหน้าที่เก็บข้อมูลส่วนที่เป็นหน้าต่างของโปรเจคนั้น และไฟล์สกุล .VBP ทำหน้าที่เป็นไฟล์หลักในการทำงานของโปรเจค นอกจากนี้ในหนึ่งโปรเจคอาจจะประกอบด้วยไฟล์สกุล .bas ที่เป็นโมดุล และไฟล์อื่นๆ ที่ช่วยการทำงานของโปรเจค

การออกแบบโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่เป็น Client / Server นี้จะใช้โปรโตคอล TCP/IP โดยใช้ช่องการสื่อสาร (Port) **5000** เป็นช่องการสื่อสารที่ใช้ควบคุมการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

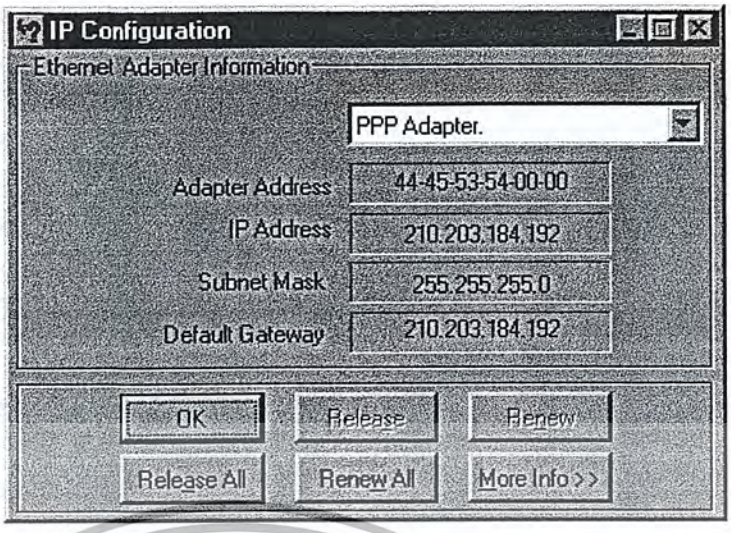
3.4.1 การออกแบบโปรแกรม Client



รูปที่ 3.8 แสดงหน้าจอโปรแกรม **Client**

โปรแกรม **Client** นี้จะทำหน้าที่ส่งการร้องขอการติดต่อเพื่อสร้างเส้นทางการส่งข้อมูลตามที่ได้กำหนดช่องทางการสื่อสาร (Port) ไว้ โดย Client นี้จะต้องทราบที่อยู่ (IP Address) ของ Server ก่อน โดยหมายเลข **IP Address** สามารถตรวจดูได้ใน Windows มีโปรแกรมที่ชื่อว่า **IP Configuration** ไฟล์ **exe** จะอยู่ที่ **Window/Winipcfg.exe** โดยมีหน้าตาดังรูปด้านล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

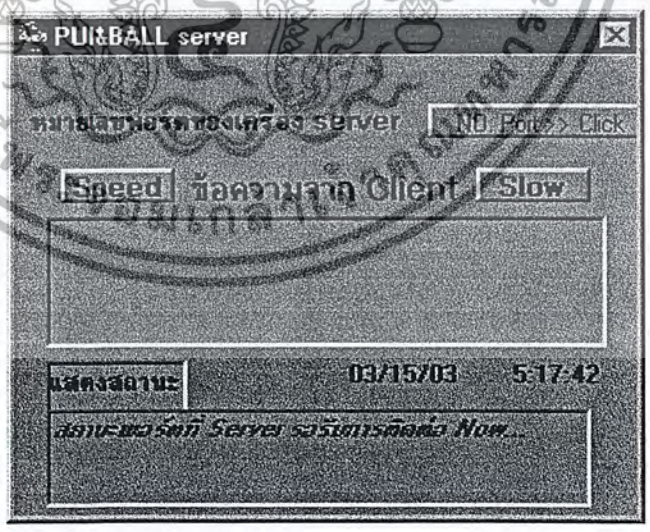


รูปที่ 3.9 แสดงหน้าจอโปรแกรมที่ตรวจดูหมายเลข **IP Address**

ถ้าต้องการดูรายละเอียดอื่นๆนอกเหนือจากนี้ก็กดปุ่ม **More Info**

Server และ Client ต่างก็มีตำแหน่งที่อยู่ (IP Address) , ช่องทางการติดต่อ (Port) โดยทั้งสองฝ่าย จะสามารถติดต่อถึงกันได้จะต้องอยู่ในช่องทางเดียวกัน ซึ่งเราสามารถกำหนดหมายเลขของ Port ได้ ทั้งนี้โปรแกรมที่ติดต่อนั้นจะต้องอ้างอิงหมายเลขของ Port ทุกครั้ง เนื่องจากในระบบมีโปรแกรมมากมายที่กำลังติดต่อกันอยู่

3.4.2 การออกแบบโปรแกรม Server



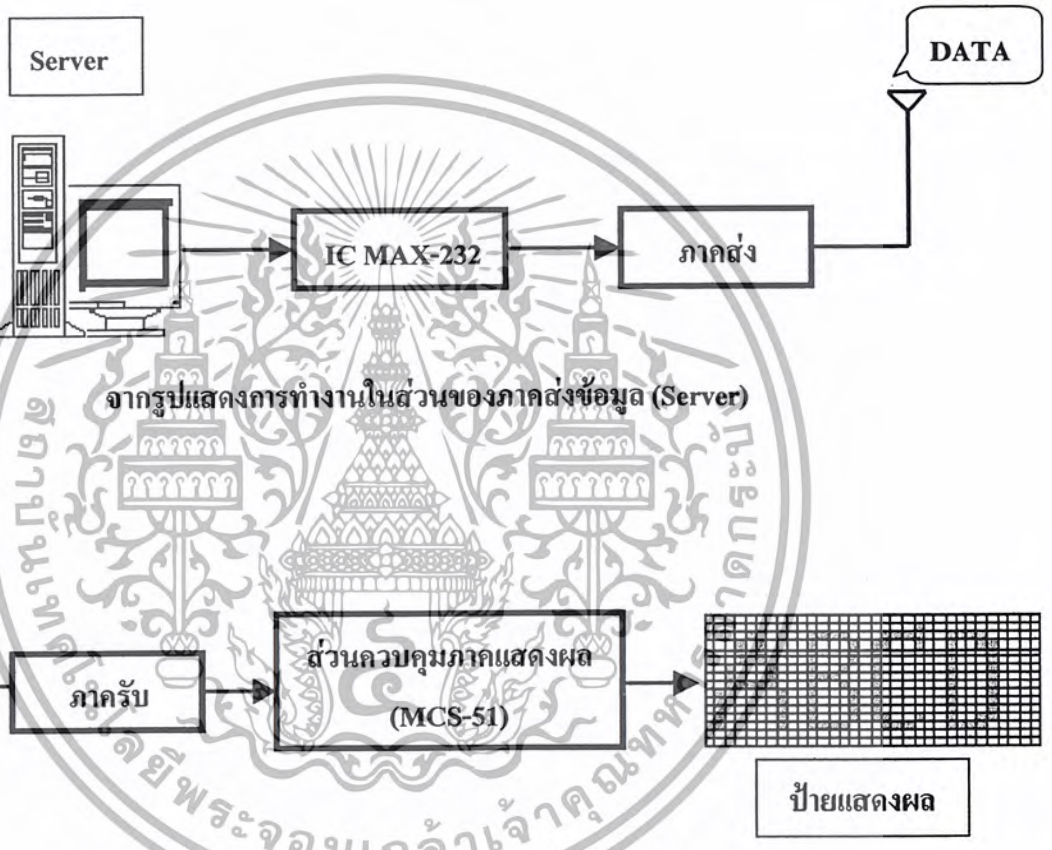
รูปที่ 3.10 แสดงหน้าจอโปรแกรม **Server**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม **Server** จะเป็นส่วนที่ทำหน้าที่เหมือนกองอำนวยการ , ประชาสัมพันธ์ , เมสเซทเจอร์ รวมถึงผู้จัด ให้กับระบบ ลูกข่าย โดย Server ในที่นี้จะทำหน้าที่รับข้อมูลจาก Client แล้วทำการส่งข้อมูลผ่านทางพอร์ตอนุกรม (Serial Port) ให้กับป้ายแสดงผล (Display Board)

3.4.3 การทำงานในส่วนการส่งข้อมูลไร้สาย (Wireless)

ในส่วนของการรับและส่งข้อมูลในส่วนของโครงงานนี้ได้ประยุกต์ใช้วงจรออดไร้สายซึ่งใช้งานเป็นเครื่องรับและส่งข้อมูลระหว่าง Computer (Server) และส่วนของ Microcontroller โดยมีรูปแสดงการทำงานดังนี้



จากรูปแสดงการทำงานในส่วนของภาคส่งข้อมูล (Server)

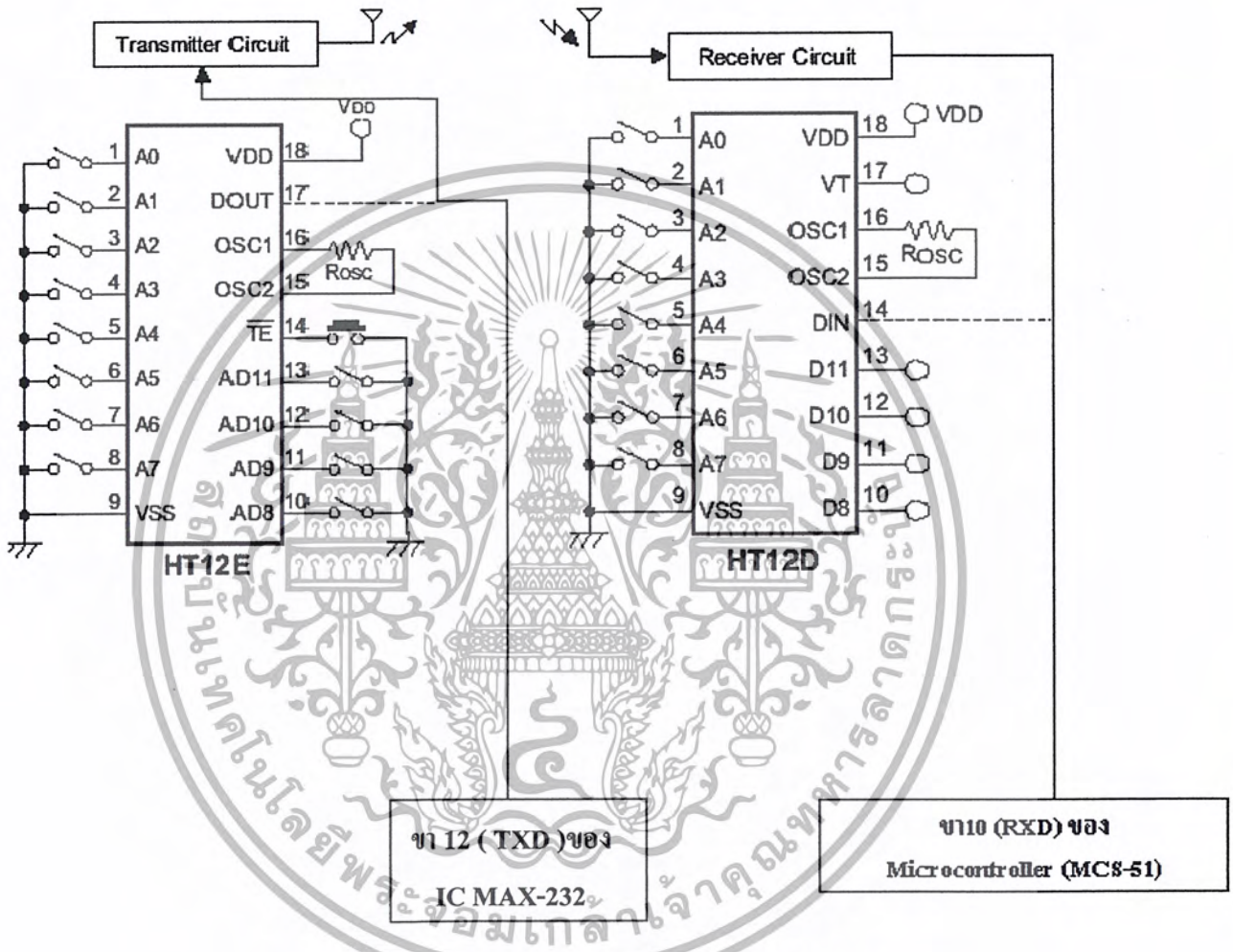
จากรูปแสดงการทำงานของภาครับข้อมูลและส่วนควบคุมการแสดงผล

การทำงานในส่วนภาคส่งของชุด Wireless ได้ประยุกต์ใช้ โดยนำสัญญาณที่ ขา TXD (ขา 12) ของ IC MAX-232 ซึ่งเป็นขา DATA มาเข้าที่ขา Dout (ขา 17) ของ HT 12E ให้เป็นขา Input แทนก็จะได้วงจรภาคส่ง

การทำงานในส่วนของภาครับ สัญญาญที่รับเข้ามาจะถูกแปลงเป็นสัญญาณ Digital เข้ามาที่ขา Din (Data In) ขา14 ของ HT 12D ที่ขา Din นี้เป็นสัญญาณข้อมูลทั้งหมดที่ Client ส่งมา ซึ่งจะต้องนำไปเข้าที่ขา RXD (ขารับข้อมูล) ของ Microcontroller (MCS-51) เพื่อประมวลผลการทำงานให้กับป้ายแสดงผล (Display Board)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application circuit



จากรูปแสดงการต่อขาใช้งานของภาครับและส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

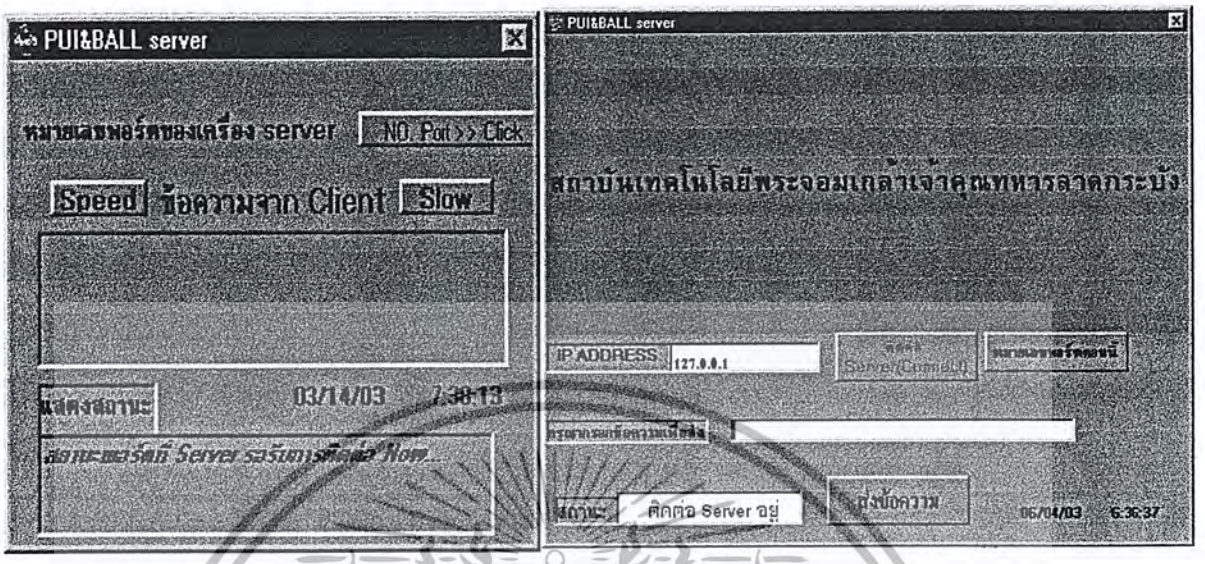
4.1 การทำงานของโปรแกรม Client / Server

เมื่อทำการเปิดโปรแกรม จะปรากฏหน้าจอเริ่มต้นของโปรแกรม ดังรูปที่ 4.1 หน้าที่ของปุ่มคำสั่งต่างๆสามารถอธิบายได้ดังนี้

รูปที่ 4.1 แสดงการทำงานเมื่อรันโปรแกรม

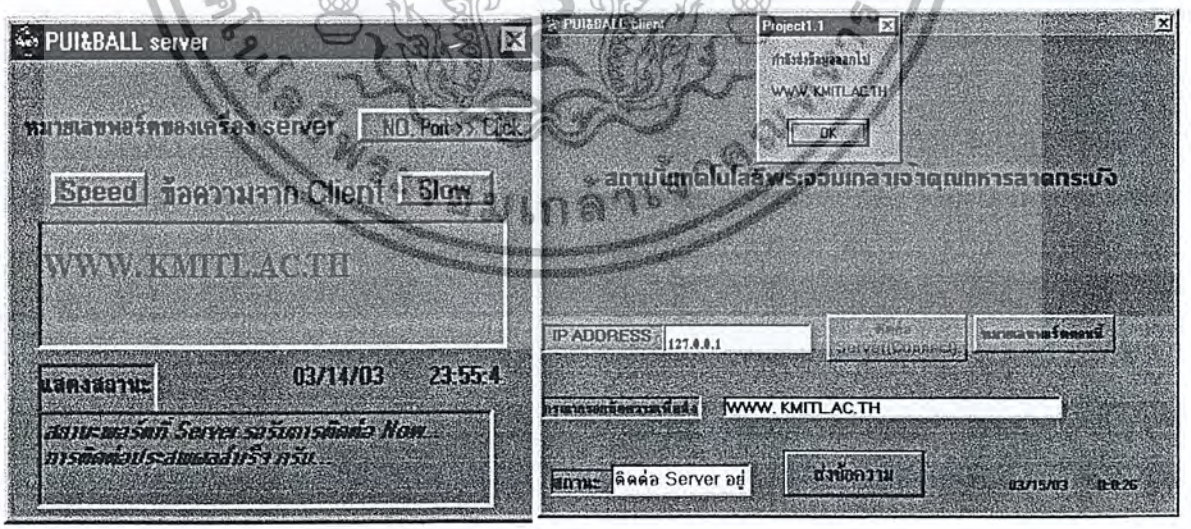
เมื่อเครื่องต้นทาง (Client) ทำการใส่หมายเลข IP Address แล้วคลิ๊กที่ปุ่ม **ติดต่อ Server** ถ้าติดต่อเครื่องปลายทาง (Server) ได้ เครื่องต้นทาง (Client) จะแสดงสถานะการติดต่อ สามารถเชื่อมต่อกับ Server ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงการเชื่อมต่อระหว่าง Client และ Server แต่ละฝั่งจะแสดงสถานะการเชื่อมต่อไว้

เมื่อทำการเชื่อมต่อเรียบร้อยแล้วก็สามารถส่งข้อความได้ โดยข้อความที่ส่งสามารถส่งได้เฉพาะภาษาอังกฤษเท่านั้น ซึ่งสามารถส่งได้ทั้งตัวพิมพ์เล็กและพิมพ์ใหญ่ ตัวเลขและตัวอักษรพิเศษ ไม่สามารถส่งเป็นภาษาไทยได้



รูปที่ 4.3 แสดงการส่งข้อความให้กับ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงการส่งข้อความ **WWW.KMITL.AC.TH** ให้กับ Server โดยข้อความที่ส่งจะแสดงใน **Text Box** ที่ Server ที่ปุ่ม **No. Port >> Click** จะเป็นปุ่มแสดง Port ที่ใช้ในการติดต่อระหว่าง **Client / Server**

4.2 ผลการทำงานของไมโครคอนโทรลเลอร์

ผลการทดลองที่ได้จากการวัดสัญญาณพัลส์ ที่ออกจากไอซีเบอร์ 74HC154 ซึ่งมี หน้าที่กำเนิดสัญญาณในการสแกนทางด้านแนวนอน โดยต้องการให้ มีค่าในการสแกนจำนวนภาพใน 1 รอบ หรือ 16 แถวเท่ากับ 45 ครั้งต่อวินาที หรือ 45 Hz

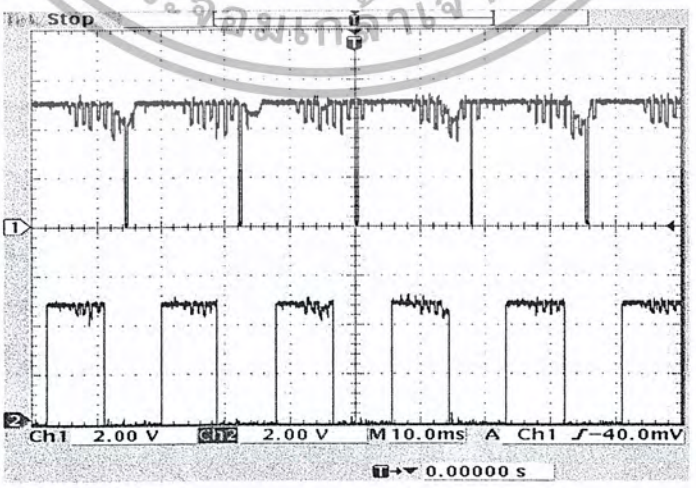
4.2.1. สัญญาณที่ได้จากไอซีเบอร์ 74HC154

เมื่อกำหนดให้มีการสแกนจำนวนภาพเท่ากับ 45 ครั้งต่อวินาทีใน 1 รอบ ดังนั้นค่าสัญญาณการสแกน 1 รอบจะหาได้จากสมการที่ 4.1

$$f = \frac{1}{T} \tag{4.1}$$

ดังนั้นสามารถหาค่าเวลาในการสแกนใน 1 รอบได้ เท่ากับ 22.2 msec

$$T = \frac{1}{45} = 22.2\ msec$$



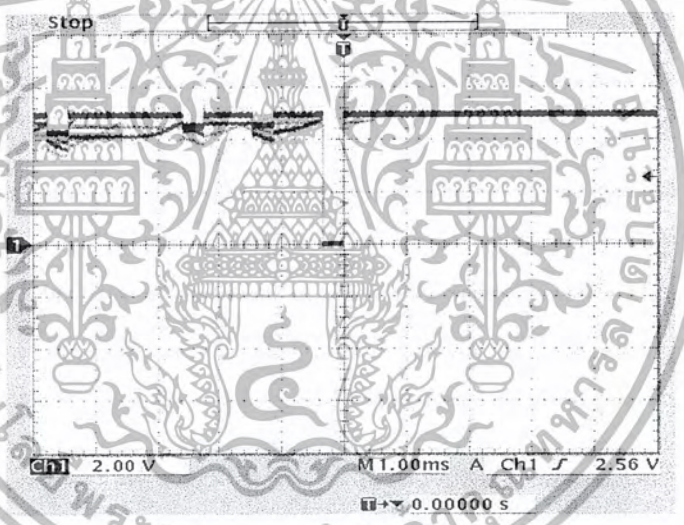
รูปที่ 4.4 สัญญาณที่ได้จากขาเอาท์พุทไอซีเบอร์ 74HC154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเป็นการใช้ออสซิลโลสโคป ทำการวัดสัญญาณที่ได้จากขาเอาท์พุทของไอซีเบอร์ 74HC154 ซึ่งเป็นสัญญาณในการสแกนทางด้านแนวนอน(CH 1)ใน 1 รอบโดยทำการวัดสัญญาณจากขาเอาท์พุท เปรียบเทียบกับสัญญาณอินพุท (CH 2) โดยที่ค่าความถี่ ที่วัดได้เท่ากับ 43 Hz ซึ่งใกล้เคียงกับค่าที่ได้กำหนดไว้

เมื่อรู้ค่าเวลา ที่ใช้ในการสแกนครบ 1 รอบแล้ว ก็จะสามารถหาค่าเวลาที่ใช้ในการสแกนแต่ละแถวได้ ซึ่งในการสแกนครบ 1 รอบ คือ การสแกนทั้งหมด 16 แถว ดังนั้นสามารถหาค่าเวลาในการสแกนแต่ละแถวได้ดังนี้

$$\frac{22.2m\,\text{sec}}{16} = 1.3m\,\text{sec}$$

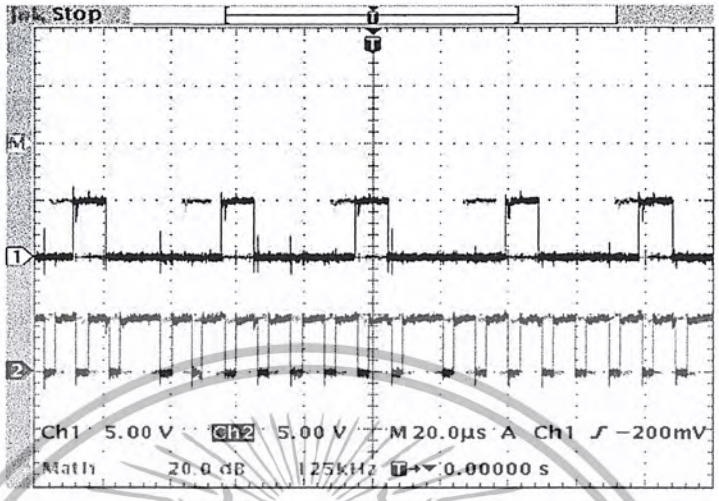


รูปที่ 4.5 สัญญาณในการสแกนแต่ละแถว

จากรูปเป็นการใช้ออสซิลโลสโคป ทำการวัดสัญญาณที่ได้จากขาเอาท์พุทของไอซีเบอร์ 74HC154 ซึ่งเป็นสัญญาณในการสแกนทางด้านแนวนอนใน 1แถว โดยทำการวัดสัญญาณจากขาเอาท์พุท 2 ขา โดยค่าเวลาที่ใช้ในการสแกน 1 รอบ จะใช้เวลา 1 msec ซึ่งใกล้เคียงกับค่าที่ได้กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 สัญญาณนาฬิกาและสัญญาณข้อมูลที่ป้อนให้กับไอซีเบอร์ 74HC164



รูปที่ 4.6 สัญญาณนาฬิกาและสัญญาณข้อมูลที่ป้อนให้กับ ไอซีเบอร์ 74HC164

จากรูปเป็นการวัดสัญญาณนาฬิกา และสัญญาณข้อมูล ที่มาจากชุดควบคุมไมโครคอนโทรลเลอร์ ที่นำมาป้อนให้กับไอซีเบอร์ 74HC164 โดยที่สัญญาณที่วัดได้จาก CH1 ของออสซิลโลสโคป คือ สัญญาณข้อมูล ส่วนสัญญาณที่วัดได้จาก CH2 ของออสซิลโลสโคป คือ สัญญาณนาฬิกา ซึ่งจากรูป เป็นการส่งข้อมูล AA ออกไป โดยจะวัดสัญญาณที่ ขอบขาขึ้นของสัญญาณนาฬิกา

4.2.3 การคิดค่าความต้านทานที่ต่อกับทรานซิสเตอร์

ในส่วนของวงจรนั้นจะใช้ตัวต้านทานที่ต่อกับทรานซิสเตอร์ทั้งหมด 3 ค่า โดยจะมี R_{B2} และ R_C ของ ทรานซิสเตอร์เบอร์ C1815 ที่ใช้ในด้านของคอลัมน์ และ R_{B1} ของ ทรานซิสเตอร์เบอร์ BD 678 ที่ใช้ในด้านของโรว์

4.2.3.1 ค่าความต้านทาน R_{B2} และ R_C ของทรานซิสเตอร์เบอร์ C1815

ผลการทดลองในส่วนของ LED นั้นได้ทำการวัดกระแสที่ไหลผ่าน LED ในแต่ละตัว ซึ่งวัดได้ประมาณ 60 mA ซึ่งจะเป็นกระแสพีค ซึ่งกระแสนี้ จะนำไปใช้ในการคำนวณหาค่าของตัวต้านทาน ที่ขาคอลเลกเตอร์ของ ทรานซิสเตอร์เบอร์ C1815 ที่ใช้ในด้านของคอลัมน์ ซึ่งจะสามารถคำนวณหาค่าความต้านทาน R_C ได้ตามสมการที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_C = \frac{V_{CC} - V_{EC1} - V_{CE2} - V_D}{I} \quad (4.2)$$

$$R_C = \frac{5 - 1 - 0.2 - 2}{60mA} = 30\Omega$$

รูปที่ 4.7 แสดงค่าตัวแปรต่างๆ ในวงจร

ในการใช้งานจริงได้เลือกใช้ความต้านทาน R_C ที่มีค่า เท่ากับ 15 Ω เนื่องจากเมื่อต่อใช้งานจริง LED ไม่ค่อยสว่าง จึงลดค่าความต้านทานลง

ในส่วนของการคิดค่าความต้านทาน R_{B2} ของ ทรานซิสเตอร์เบอร์ C1815 จะคำนวณได้ตามสมการที่ 4.3

$$R_{B2} = \frac{V_{IN2} - V_{BE2}}{I_{B2}} \quad (4.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_{B2} = \frac{5 - 0.7}{857\mu A} = 5K\Omega$$

โดยที่ค่า I_{B2} หาได้จากกระแสคอลเลกเตอร์ I_C (60 mA) หารกับค่า β (70) ซึ่งดูได้จาก Data Sheetของ C1815 มีค่า β =120 ซึ่งเป็นค่า β ที่ทรานซิสเตอร์ทำงานในย่านแอคทีฟ สาเหตุที่ใช้ค่า β น้อยกว่า 120 เพราะว่าต้องการให้ทรานซิสเตอร์ทำงานที่ย่านอิ่มตัว ในการใช้ งานจริงได้เลือกใช้ความต้านทาน R_B ที่มีค่า เท่ากับ 4.7 KΩ

4.2.3.2 ค่าความต้านทาน R_{B1} ของทรานซิสเตอร์เบอร์ BD 678

ในส่วนของค่าความต้านทาน R_{B1} ของทรานซิสเตอร์เบอร์ BD 678 ที่ใช้ในด้านโรว์ นั้นจะคำนวณได้ตามสมการที่ 4.4

$$R_{B1} = \frac{V_{CC} - V_{EB1} - V_{IN1}}{I_{B1}} \qquad (4.4)$$

$$R_{B1} = \frac{5 - 1.4 - 0}{3.8mA} = 947.3\Omega$$

โดยที่ค่า I_{B1} หาได้จาก กระแสคอลเลกเตอร์ I_C (3.8 A) หารกับค่า β (1000) ซึ่งดูได้จาก Data Sheet ของ BD 678 โดย BD 678 จะจ่ายกระแส I_C ได้ 4 A และมีค่า β = 2000 ซึ่งเป็นค่า β ที่ทรานซิสเตอร์ทำงานในย่านแอคทีฟ สาเหตุที่ใช้ค่า β น้อยๆ เพราะว่าต้องการให้ทรานซิสเตอร์ทำงานที่ย่านอิ่มตัว ค่ากระแสคอลเลกเตอร์ I_C (3.8 A) นั้น มาจากกระแสที่ ไหลผ่าน LED มีแต่ละตัวมีค่าเท่ากับ 60 mA ซึ่งต้องคิดทั้งหมด 64 ตัว เพราะมี 64 หลัก ในการใช้งานจริงได้เลือกใช้ความต้านทาน R_{B1} ที่มีค่า เท่ากับ 1 KΩ

เมื่อนำค่าที่ได้ทั้งหมด มาทดลองต่อใช้งานจริงดูก็ปรากฏว่าสามารถทำงานได้โดยไม่มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

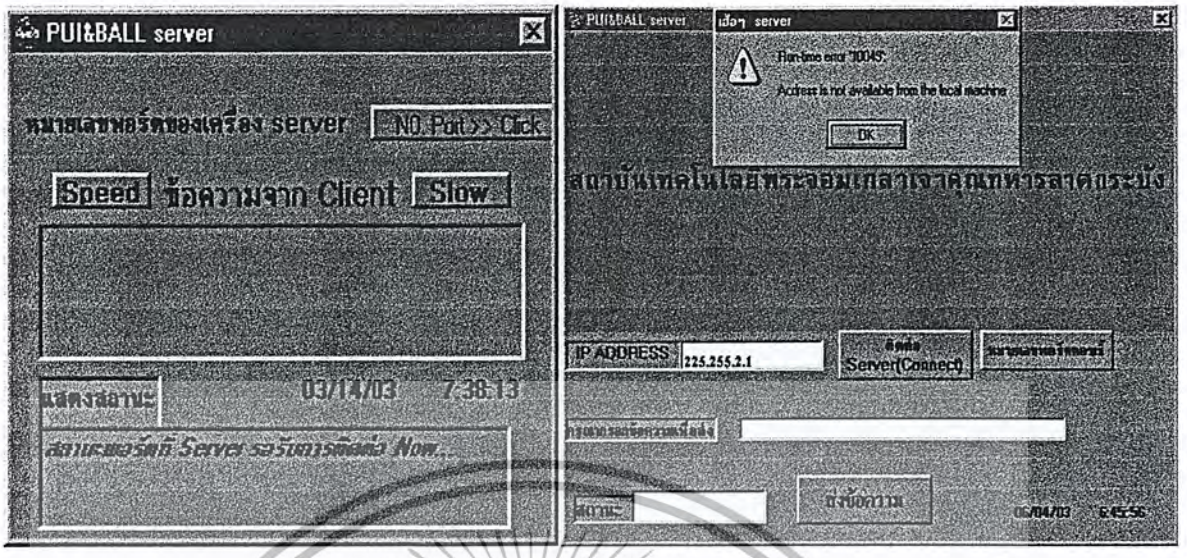
บทสรุป

จากกการทำงานของโปรแกรม **Client/Server** พบว่าการเชื่อมต่อของเครื่องคอมพิวเตอร์ซึ่งอยู่คนละเครือข่ายและแต่ละเครื่องมีการเชื่อมต่อกับอินเทอร์เน็ตไว้ และกำหนดการติดต่อกัน โดยที่ให้เครื่องหนึ่งเป็นเครื่องควบคุมและติดต่อเข้ามายังอีกเครื่องหนึ่ง อีกเครื่องหนึ่งเป็นเครื่องรอรับการติดต่อและรอรับคำสั่งจากเครื่องควบคุม และทำตามคำสั่งที่ส่งมานั้น เครื่องทั้งสองเครื่องนี้สามารถที่จะติดต่อกันได้ โดยโครงงานนี้ได้เขียนโปรแกรมควบคุมการติดต่อระหว่างเครื่องทั้งสองผ่านทางอินเทอร์เน็ตและมีการส่งคำสั่ง (ข้อความ) ไปควบคุมภาคแสดงผลที่อยู่ที่เครื่องปลายทางหรือเครื่องเป้าหมายอีกทีโดยใช้ภาษาวิชวลเบสิก ซึ่งมีประสิทธิภาพสูงสามารถเขียนโปรแกรมติดต่อผ่านระบบเครือข่ายได้โดยจะต้องทราบหมายเลขประจำเครื่อง (**IP Address**) ของเครื่องปลายทาง เพราะถ้าไม่ทราบหมายเลขประจำเครื่องก็ไม่สามารถที่จะทำการติดต่อกันได้

5.1 ปัญหาในการเชื่อมต่อกันระหว่าง Client กับ Server

สำหรับการเขียนโปรแกรมการเชื่อมต่อทางอินเทอร์เน็ตนั้นจะต้องมีการทดสอบการทำงานซึ่งเป็นไปไม่ได้ที่เราจะต้องเชื่อมต่อกับอินเทอร์เน็ตตลอดเวลา จึงต้องมีตัวทดสอบการทำงานส่วนนี้แทนโดยใช้ **Personal Web Server** โดยใน Window มีโปรแกรมนี้ให้อยู่แล้ว สามารถนำมาใช้โดยติดต่อผ่าน IP หมายเลข **127.0.0.1** ซึ่งเป็นหมายเลข IP Address ที่ไม่ได้มีการเชื่อมต่อกับอินเทอร์เน็ต เราสามารถสร้าง Application บนอินเทอร์เน็ต แล้วทดสอบจากหมายเลข IP Address นี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงการเชื่อมต่อระหว่าง Client / Server กรณีป้อน IP Address ไม่ถูกต้อง

จากรูปแสดงการเชื่อมต่อระหว่าง **Client** และ **Server** ที่ไม่สำเร็จเนื่องจากที่อยู่ของ **Server (IP Address)** ไม่ถูกต้อง เพราะฉะนั้นการเชื่อมต่อจะต้องทราบที่อยู่ของเครื่อง **Server** ที่ถูกต้องด้วย หรืออาจจะใช้ชื่อโดเมนเนม (DNS) แทนหมายเลข IP Address ก็ได้ เช่น **WWW.KMITL.AC.TH**

5.2 ปัญหาในส่วนของการส่งข้อมูลระหว่าง Computer (Server) กับ Microcontroller

การส่งข้อมูลระหว่าง **Serever** กับป้ายแสดงผลมีปัญหาเรื่องความเร็วในการรับส่งข้อมูล จากการที่เริ่มต้นได้เขียนโปรแกรมที่ **Server** ให้รับค่าเข้ามาแล้วส่งค่าที่เข้ามา (ข้อความ) ออกทางพอร์ตอนุกรมโดยทันที ทำให้ข้อมูล (ข้อความ) ที่ Microcontroller รับเข้ามาไม่ทัน ทำให้ข้อมูล (ข้อความ) ออกมาไม่ครบ จนในที่สุดพบว่าการที่เรารับข้อมูลจาก **Client** เข้ามาควรนำมาเก็บค่าก่อนไว้แล้ว แล้วนับจำนวนของข้อมูล หลังจากนั้นก็ทำการหน่วงเวลาในการส่งข้อมูล ก่อนส่งให้ Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ปัญหาในส่วนการรับและส่งข้อมูลระหว่างเครื่องรับและส่ง

ในการสื่อสารระหว่างภาครับและส่งข้อมูลจำเป็นต้องมีการใส่บิตลักษณะพิเศษของชุดตัวอักษรเพื่อทราบจุดเริ่มต้นและจุดสิ้นสุดของข้อมูล ในกรณีที่การส่งมีสัญญาณรบกวนมากๆ เราจำเป็นที่จะต้องใส่บิตลักษณะพิเศษนี้เป็นจำนวนมากเพื่อให้ภาครับทำการตรวจสอบบิตที่เป็นข้อมูลจริงๆเท่านั้น

5.4 ปัญหาในส่วนของแหล่งจ่ายไฟ (Supply)

สิ่งที่เป็นปัญหาที่เกิดขึ้นกับวงจร คือ ปัญหาของแหล่งจ่ายไฟ ที่ป้อนให้กับวงจร ซึ่งโครงงานนี้ ควรจะเลือกใช้แหล่งจ่ายไฟแบบสวิทช์ชิ่ง สาเหตุที่ใช้ แหล่งจ่ายไฟแบบสวิทช์ชิ่งเพราะมีน้ำหนักเบา และสามารถจ่ายกระแสได้สูง แต่ต้องระวังในเรื่องของสัญญาณรบกวน ซึ่งถ้าแหล่งจ่ายไฟแบบสวิทช์ชิ่ง ที่ใช้ไม่มีการป้องกันสัญญาณรบกวน จะทำให้การแสดงผลของ LED ผิดเพี้ยนไป

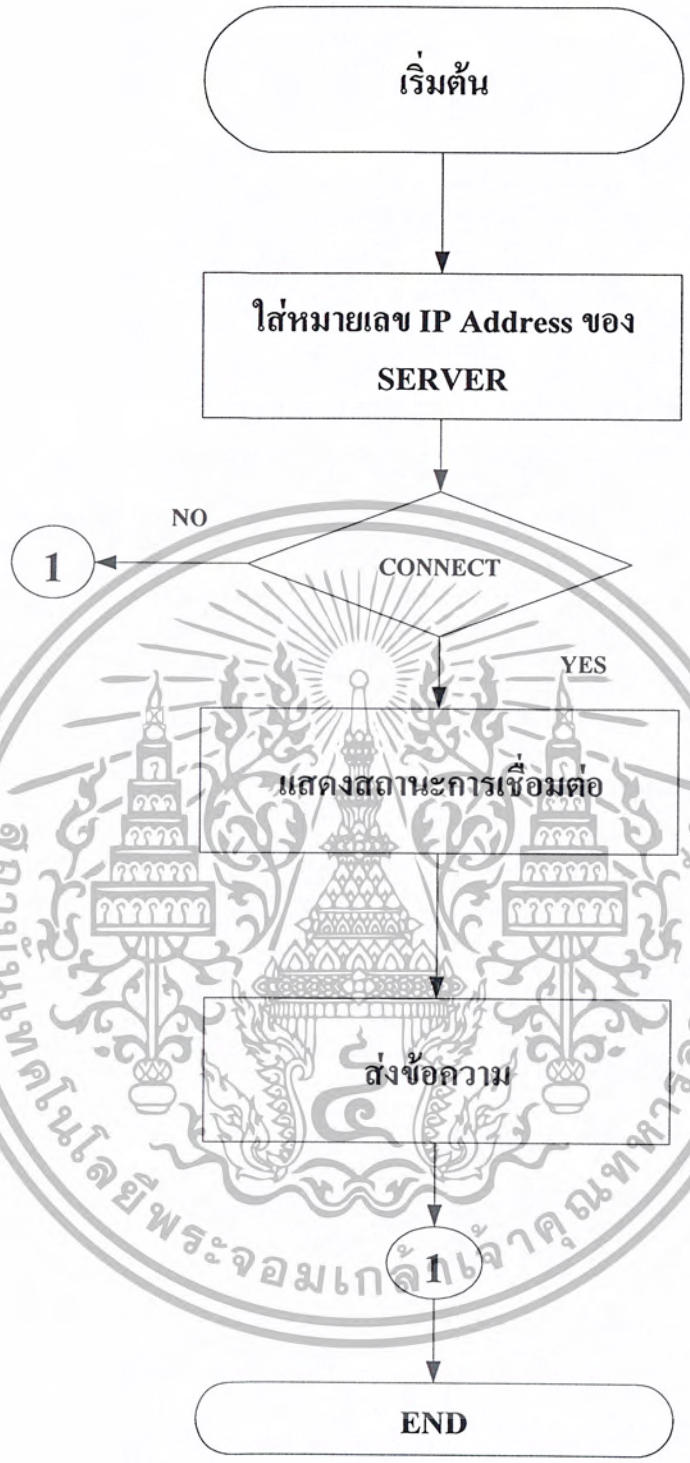
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

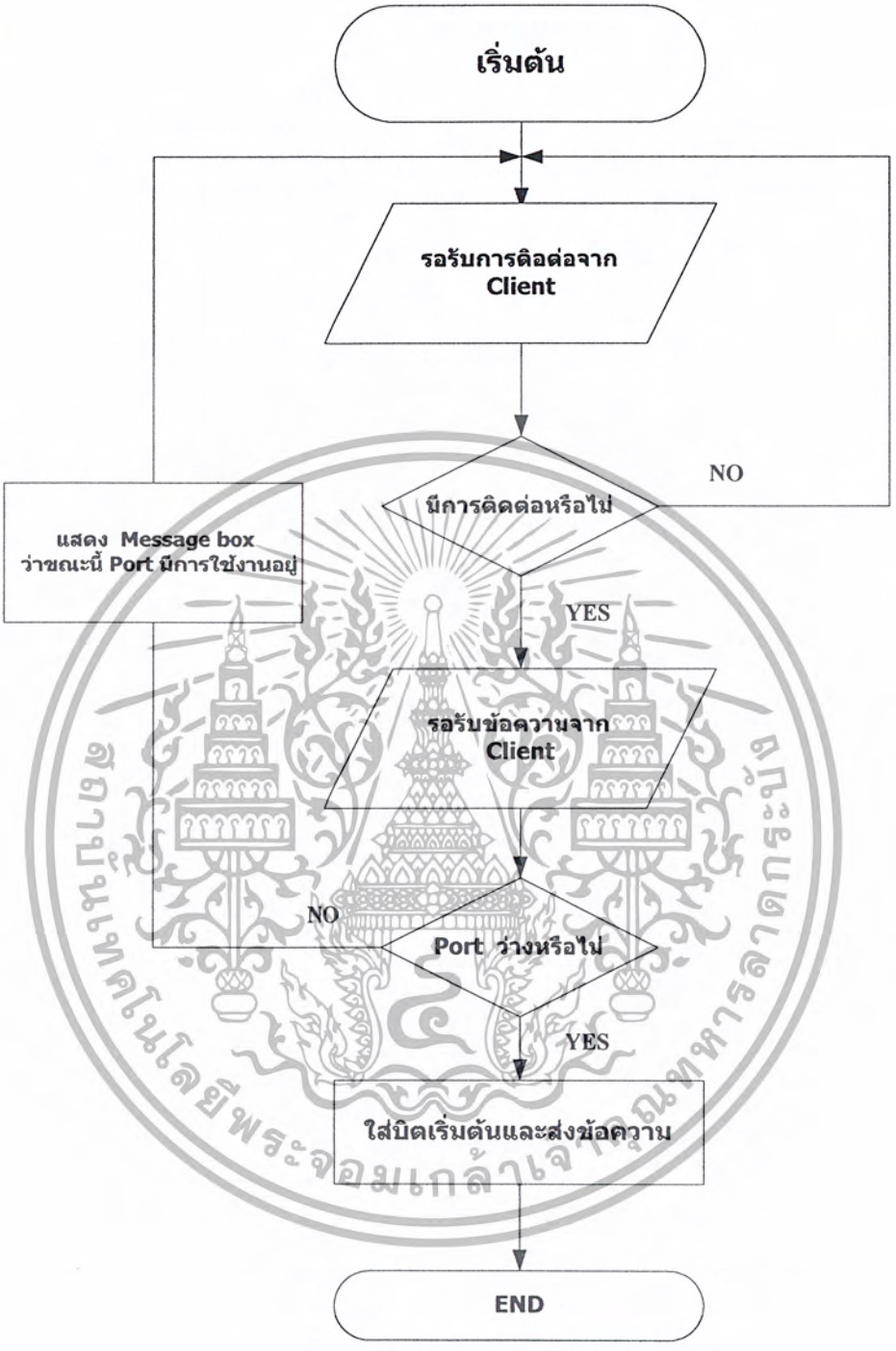
ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



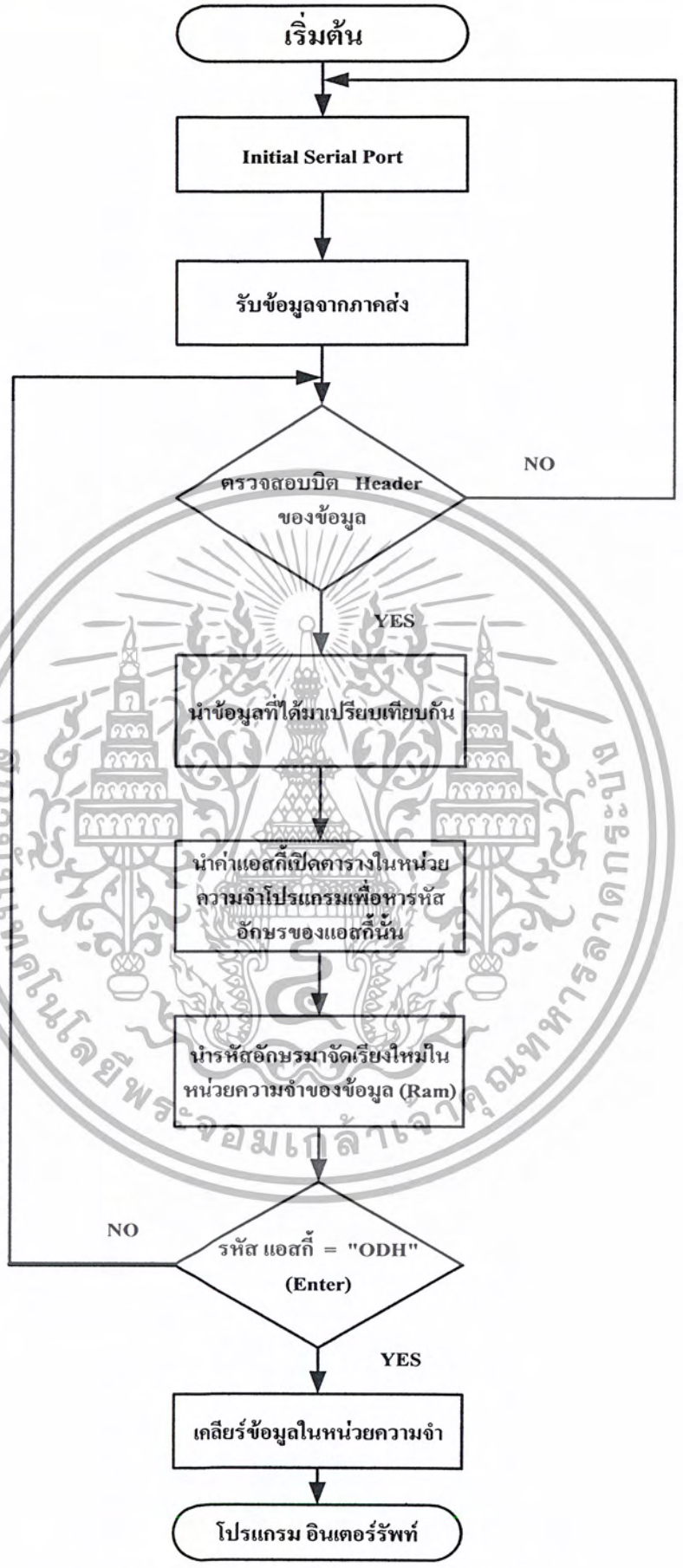
Flowchart แสดงการทำงานของ Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



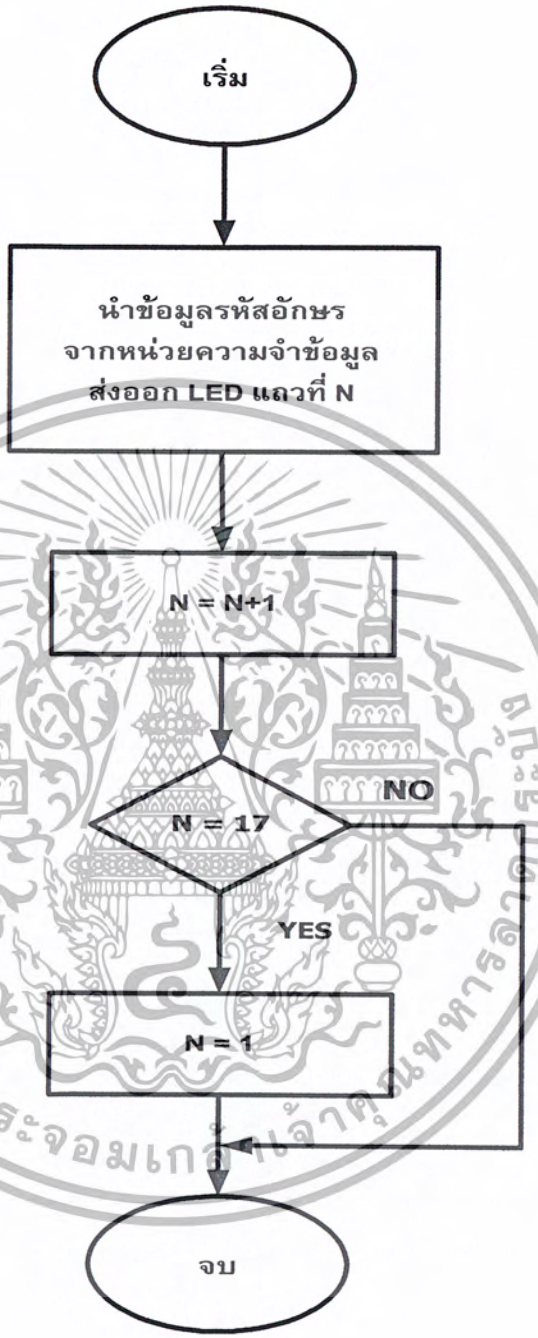
Flowchart แสดงการทำงานของ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Flowchart แสดงการทำงานของ Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

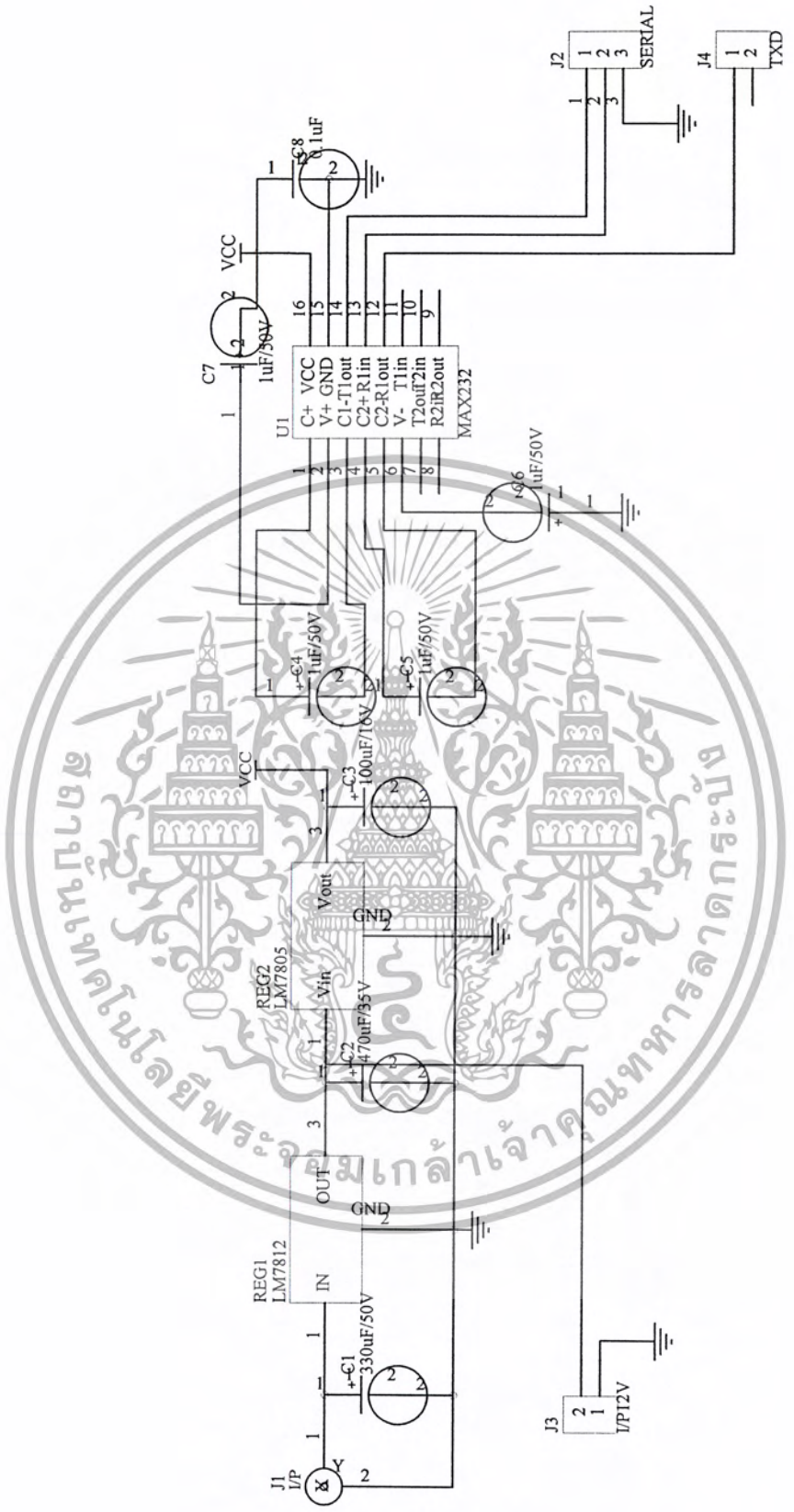


Flow Chart ในส่วนของโปรแกรมอินเตอร์รัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

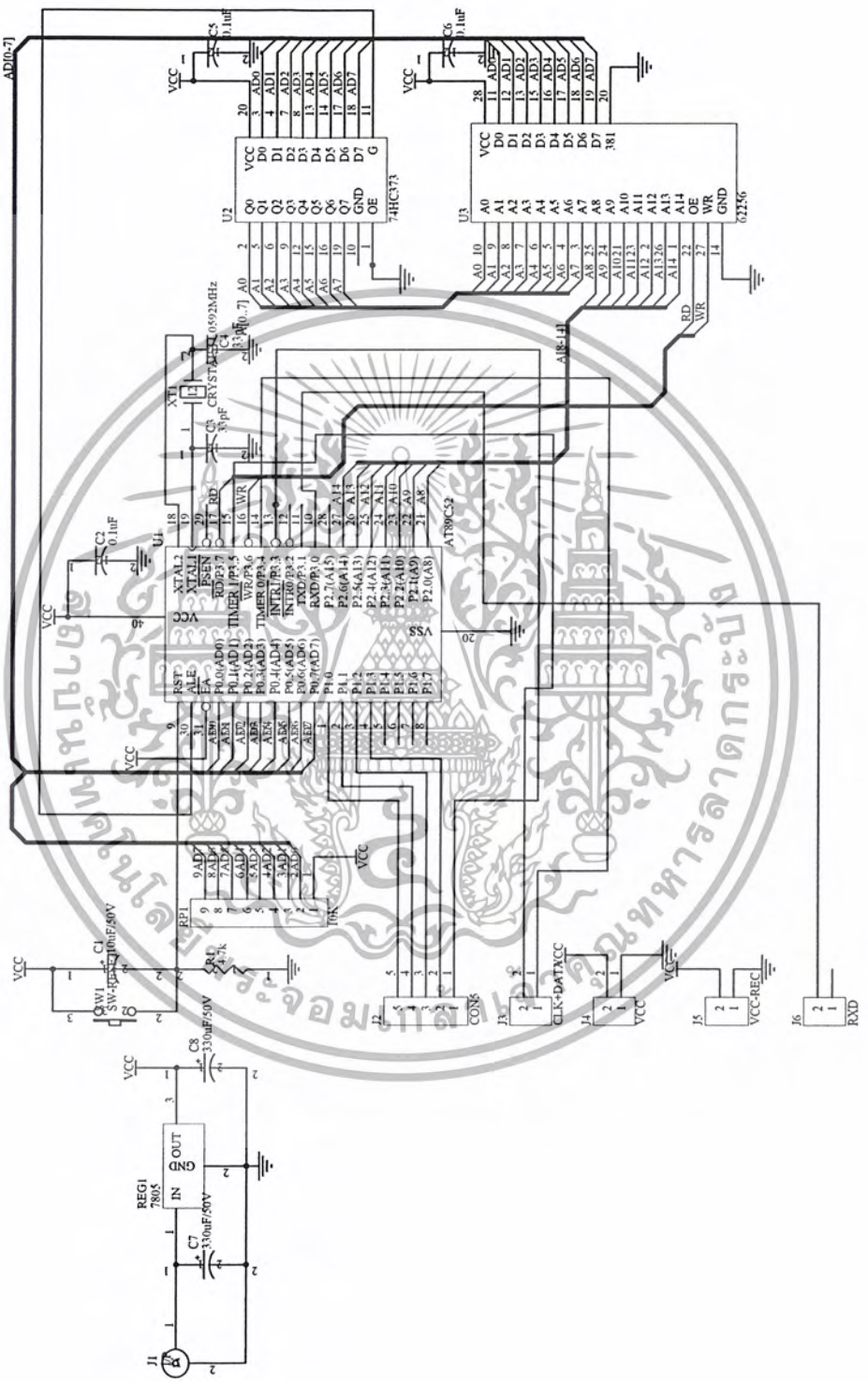
ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



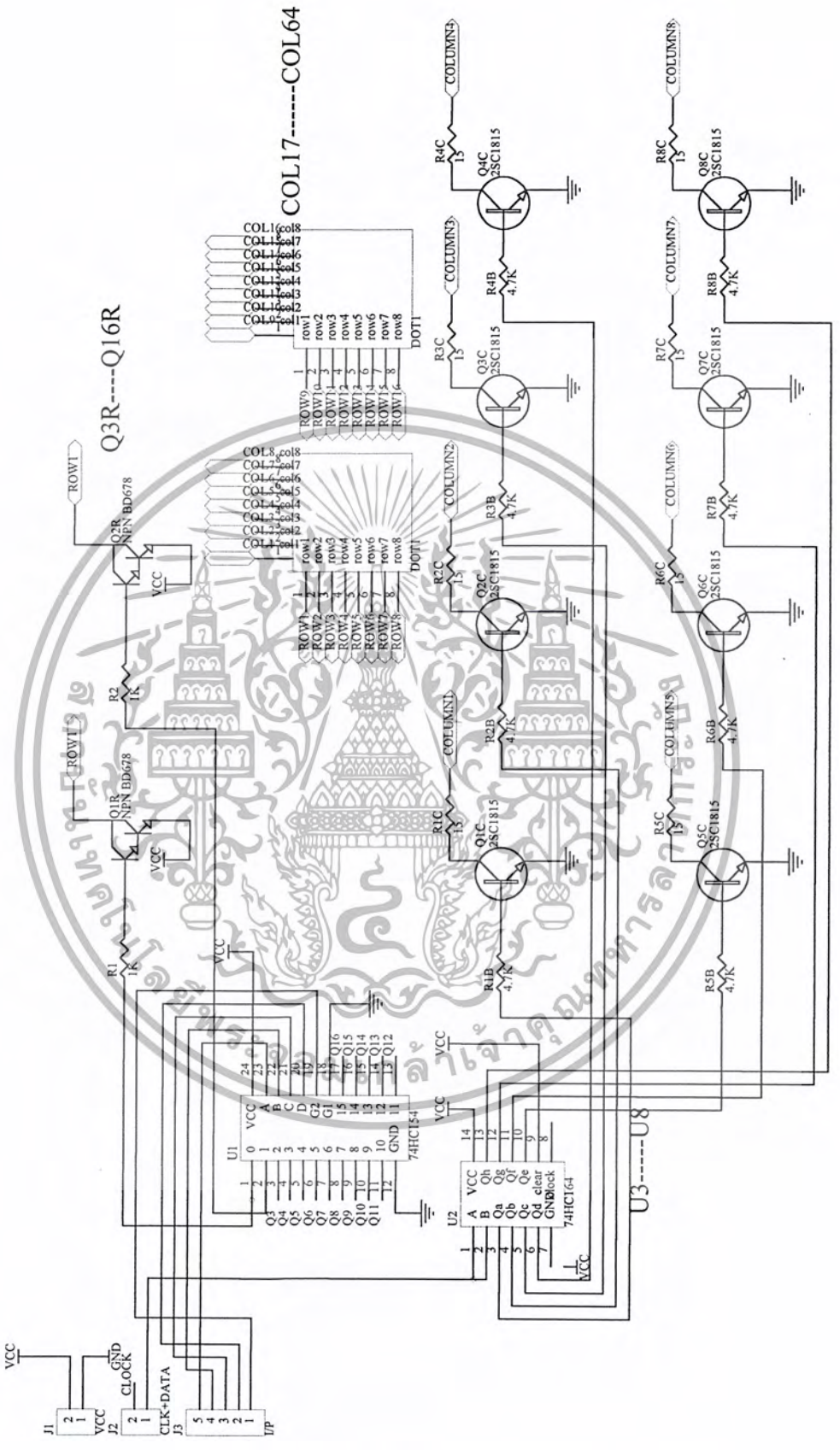
วงจรแหล่งจ่ายไฟและขาสัญญาณที่ต่อเข้ากับภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรควบคุมการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรส่วนของชุดควบคุมการสแกนของภาคแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมวิชวลเบสิก 6.0

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@@@@@@@@@@@@@@@@@@@@&CLIENT&@@@@@@@@@@@@@@@@@@@

```
Option Explicit
Private Sub Command1_Click()
Winsock1.RemoteHost = Text1.Text   ' IP address ที่ใช้ในการติดต่อ Server
Winsock1.RemotePort  = "5000"      ' ระบุ port ที่ใช้ในการติดต่อ เป็น port อะไรก็ได้แต่ต้องไม่
                                     ซ้ำกับ port ที่สงวนเอาไว้
Winsock1.Connect                   ' ใช้ winsock ในการติดต่อ
Command1.Enabled = False           ' ไม่ให้ใช้ปุ่มของการติดต่อ winsock เนื่องจากเกิด error ใน
                                     การทำงานซ้ำของการติดต่อ
Text2.SetFocus
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''แสดงหมายเลข PORT''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Command2_Click()
MsgBox "Num Com port : " & Winsock1.LocalPort, vbInformation, "local port"      'ให้ winsock
แสดง port  ที่กำลังติดต่อ
End Sub
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''ส่งข้อมูลใน Text box โดยการ Click''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''',,,,,,,
Private Sub Command4_Click()
Winsock1.SendData Text2.Text       ' ส่งข้อมูลที่อยู่ใน text box โดยใช้ winsock
MsgBox "กำลังส่งข้อมูลออกไป" & vbCrLf + vbCrLf + Text2.Text
Text2.Text = ""    'เคลียร์ Text box
'MSComm1.OutBufferCount = 0        ' ไม่แสดงข้อความี่ส่ง (ว่าง)
Text2.SetFocus                     'ให้ตำแหน่งของ curzer อยู่ที่ text box
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''ส่งข้อมูลใน Text box โดยการกดแป้นพิมพ์''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then              ' เมื่อกด Enter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If KeyAscii = 13 Then                   ' เมื่อกด Enter
Winsock1.RemoteHost = Text1.Text  ' IP address ที่ใช้ในการติดต่อ Server
Winsock1.RemotePort = "5000"          ' ระบุ port ที่ใช้ในการติดต่อ เป็น port อะไรก็ได้แต่ต้องไม่
                                          ซ้ำกับ port ที่สงวนเอาไว้
Winsock1.Connect                        ' ใช้ winsock ในการติดต่อ
Command1.Enabled = False              ' ไม่ให้ใช้ปุ่มของการติดต่อ winsock เนื่องจากเกิด error ใน
                                          การทำงานซ้ำของการติดต่อ
Text2.SetFocus
End If
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Text2_KeyPress(KeyAscii As Integer)
Text3.Text = "    ติดต่อ Server อยู่"   ' แสดงชื่อของปุ่มในการทำงาน
If KeyAscii = 13 Then
Winsock1.SendData Text2.Text          ' ส่งข้อมูลที่อยู่ใน text box โดยใช้ winsock
MsgBox "กำลังส่งข้อมูลออก ไป" & vbCrLf + vbCrLf + Text2.Text
Text2.Text = ""                          ' ไม่แสดงข้อความที่ส่ง (ว่าง)
Text2.SetFocus
End If
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal Scode As
Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
Winsock1.Close                          ' ปิดการทำงานของ winsock เมื่อเกิด error ในการทำงาน
MsgBox "Have a Ploblem", vbExclamation, "error" ' ให้ Massage box โชว์ข้อความเมื่อเกิด error
Command1.Enabled = True               'ให้ปุ่มที่ใช้ในการติดต่อ Server ใช้ได้ปกติ
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub winsock1_Connect()
MsgBox " Server ตอบรับการติดต่อเรียบร้อยแล้ว", vbInformation, " Wireless Display Board Via
Internet"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Timer1_Timer()
        Interval.FontBold = True
        Interval.Caption = Format(Date$, "mm/dd/yy") & "      " & Format(Time, "H:M:S")
End Sub
Private Sub winsock1_Sendcomplete()
Text3.Text = "SEND  COMPLETE  " & vbCrLf
End Sub

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@&SERVER&@@@@@@@@@@@@@@@@@@@@@
Option Explicit
Option Base 0
Dim value(100) As String
Dim i, j, k As Integer
Private Sub Command3_Click()
Winsock1.SendData "Cancel Connecting"
Winsock1.Close
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''แสดงหมายเลข PORT''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Private Sub Command4_Click()
MsgBox "No. port >> " & Winsock1.LocalPort, vbInformation, "Local Port No."  ' แสดง port ที่
ใช้ในการติดต่อ
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''ขณะรันโปรแกรม''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Form_Load()
server1.Show
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Winsock1.LocalPort = 5000          ' แสดง Port ที่ client ใช้ในการเชื่อมต่อ ต้องกำหนดให้ตรงกัน
                                     ทั้ง 2 ฝ่าย
Winsock1.Listen                    ' Winsock รอการเรียกใช้งานของ Client
Text3.Text = Text3.Text & "สถานะพอร์ตที่ Server รอรับการติดต่อ Now...  " & vbCrLf 'บอกว่า
เมื่อติดต่อได้แล้ว กำลังรอข้อมูลที่ Client จะส่งแล้วให้ขึ้นบรรทัดใหม่
MSComm1.Settings = "1200,N,8,1"
MSComm1.CommPort = 1
MSComm1.PortOpen = True
'MSComm1.OutBufferCount = 0
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Form_Unload(Cancel As Integer)
If Winsock1.State <> sckClosed Then
Winsock1.Close
Unload server1
End If
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''การส่งข้อมูล''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Timer3_Timer()
Dim dat As String
Dim aa As String
Dim Count As Integer
Dim i As Integer
  For i = 0 To 4
          Do While MSComm1.OutBufferCount > 0
          Loop
          MSComm1.Output = Chr$(&HF4) & Chr$(&HF1) & Chr$(&HFA)
  Next
          Do While MSComm1.OutBufferCount > 0
          Loop
          dat = Text1.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For i = 0 To Len(Text1.Text) - 1          ' รับค่าจากtext box ทั้งหมด
    aa = Left$(dat, 1)                    ' เอาตัวแปร aa เก็บค่าตัวอักษรทางซ้ายจำนวน 1 ตัว
    Debug.Print aa
    dat = Right$(dat, Len(dat) - 1)       ' เอาตัวแปร dat เก็บค่าตัวอักษรที่เหลือจากการตัดทาง
                                          ซ้ายไป 1 ตัวอักษร
    Debug.Print dat                       ' แสดงข้อมูลที่ส่งออกไป เพื่อทดสอบการทำงาน
  For Count = 0 To 1
    aa = aa & aa & aa                     ' ส่งข้อมูลออกทีละ 6 ไบต์
  Next                                    ' Loop ของข้อมูล 6 ไบต์
    Debug.Print aa
    Do While MSComm1.OutBufferCount > 0
    Loop
    MSComm1.Output = Chr$(&HF5) & aa
Next                                      ' Loop ของตัวอักษรถัดไป
    Do While MSComm1.OutBufferCount > 0
    Loop
    MSComm1.Output = Chr$(&HF5) & Chr$(13) & Chr$(13) & Chr$(13) & Chr$(13) &
Chr$(13) & Chr$(13)                       ' บิตปิดท้ายของข้อมูลทั้งหมด
    Timer3.Enabled = False
End Sub
```

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
On Error GoTo error:
MsgBox "server ได้รับการติดต่อจาก client แล้ว", vbInformation, " Connection Complete" 'เมื่อ
                                          การติดต่อเกิดขึ้นให้แสดงข้อความนี้
Text3.Text = "การติดต่อประสพผลสำเร็จ ครับ..." & vbCrLf + " หมายเลข IP Address ของ Client
คือ" & "  " + Winsock1.LocalHostName     ' แสดงข้อความที่ SERVER เมื่อติดต่อได้เรียบร้อยแล้ว
'MSComm1.PortOpen = True                  ' สั่งให้ Server ทำการเปิด port ที่ใช้ส่งข้อมูลทาง serial
                                          port ในที่นี้ใช้ port COM 1
MSComm1.OutBufferCount = 0
If Winsock1.State <> sckClosed Then
Winsock1.Close
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Winsock1.Accept requestID
End If
error:
'MsgBox "Connecting it as a problem.."
End Sub
```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Winsock1.GetData value(i)
Text1.Text = value(i)
i = i + 1
MSComm1.OutBufferCount = 0
Timer3.Enabled = True
End Sub
```

```
Private Sub Timer2_Timer()
        Interval1.FontBold = True
        Interval1.Caption = Format(Date$, "mm/dd/yy") & "      " & Format(Time, "H:M:S")
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้จะสำเร็จลุล่วงไปไม่ได้เลยถ้าขาดซึ่งความพยายามและเงินทุนที่ในการทำงานการค้นคว้าหาข้อมูล ขอบคุณหลายๆนามแฝงที่ทุกๆท่านใช้ ทำให้พวกเราได้มีโอกาสซักถามปัญหาใน Web Board ในหลายๆ Web ซึ่งเป็นประโยชน์อย่างมาก อีกทั้งรุ่นพี่โดยเฉพาะพี่ปูและเพื่อนๆที่คอยให้ความช่วยเหลือที่ดีเสมอมา ให้พวกเรามีแนวทางแก้ไขปัญหาที่เราคิดว่ายาก กลับกลายเป็นเรื่องที่ไม่ยากอย่างที่คิด และสุดท้ายขอขอบคุณทุกๆคนในครอบครัว (แม้ว่าพ่อจะไม่ได้อยู่เคียงข้างดูความสำเร็จครั้งนี้ด้วย) ที่ให้กำลังใจเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. **พ.อ. เจนวิทย์ เหลืองอร่าม, ปิยวิทย์ เหลืองอร่าม , " การเขียนโปรแกรมสำหรับ Applications ด้วย Visual Basic 6 "** บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2543
2. **สุวัฒน์ ปุณณะชัยยะ , ตัน ตัณฑ์สุทธิวงศ์ , สุพจน์ ปุณณะชัยยะ , " เปิดโลก TCP/IP และโปรโตคอลของอินเตอร์เน็ต "** บริษัท โปรวิชั่น จำกัด 2545
3. **สุเจตน์ จันทรังษ์ , " ไมโครคอนโทรลเลอร์ 8051 " กรุงเทพมหานคร** โรงพิมพ์ มหาวิทยาลัยเทคโนโลยีมหานคร 2535
4. **HTTP://WWW.thaiio.com**
5. **HTTP://WWW.konthainetwork.com**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้