สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอุปกรณ์ไฟฟ้าระยะไกลในอาคารสูงผ่านเครือข่ายทีซีพีไอพี

REMOTE EQUIPMENT CONTROL SYSTEM IN BUILDING VIA TCP/IP PROTOCOL

T050187

โดย

นาย พงศกร งามเสงี่ยม รหัส 43015309

นาย ศรายุทธ อ่อนทรง รหัส 43015328

อาจารย์ที่ปรึกษา

อาจารย์ เกียรติวรรณ ทรงสัตย์

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50187
วัน,เดือน,ปี 27 เม.ย. 2547
.b.....
.i.....

ห้องสมุดกลางไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอุปกรณ์ไฟฟ้าระยะไกลในอาคารสูงผ่านเครือข่ายทีซีพีไอพี

REMOTE EQUIPMENT CONTROL SYSTEM IN BUILDING

VIA TCP/IP PROTOCOL

ผู้จัดทำ

นาย พงศกร งามเสงี่ยม รหัส 43015309

นาย ศรายุทธ อ่อนทรง รหัส 43015328

---------- อาจารย์ที่ปรึกษา

(อาจารย์ เกียรติวรรณ ทรงสัตย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์ไฟฟ้าระยะไกลในอาคารสูงผ่านเครือข่ายทีซีพีไอพี

REMOTE EQUIPMENT CONTROL SYSTEM IN BUILDING

VIA TCP/IP PROTOCOL

โดย

นาย พงศกร งามเสงี่ยม รหัส 43015309

นาย ศรายุทธ อ่อนทรง รหัส 43015328

อาจารย์ที่ปรึกษา

อาจารย์ เกียรติวรรณ ทรงสัตย์

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอระบบควบคุมอุปกรณ์ไฟฟ้าในอาคารสูงผ่านเครือข่าย TCP/IP ซึ่งโครงงานนี้ได้พัฒนาขึ้น โดยใช้คอมพิวเตอร์ส่วนบุคคลในการสั่งงานผ่านเครือข่าย TCP/IP ส่งผ่านมายังเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ และได้ใช้การรับส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 และใช้มาตรฐาน RS-485 ซึ่งสามารถรองรับวงจรควบคุมได้จำนวนสูงสุด 32 อุปกรณ์ และแสดงผลที่หน้าจอคอมพิวเตอร์ในลักษณะของกราฟฟิก ที่ถูกออกแบบด้วยโปรแกรม Microsoft Visual Basic 6.0

ABSTRACT

The Project of us presents an electrical equipment control system in building via TCP/IP protocol. This project uses a personal computer control circuit. And can use RS-232 to RS-485 Adapter for tranform +12 to TTL logic signal. Also computer can control and displays status of equipment in graphic mode what is designed with Microsoft Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

สารบัญ i

สารบัญตาราง ii

สารบัญรูป iii

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 ขอบเขตของโครงงาน 2

บทที่ 2 ทฤษฎีและการออกแบบ 3

2.1 การสื่อสารผ่านพอร์ตอนุกรม 3

2.1.1 การรับส่งข้อมูลแบบอนุกรมตามมาตรฐาน RS232 4

2.1.2 การรับส่งข้อมูลแบบอนุกรมตามมาตรฐาน RS485 5

2.2 ไมโครคอนโทรลเลอร์ตระกูล 8051 7

2.2.1 ไมโครคอนโทรลเลอร์ 8051 7

2.2.2 การสื่อสารระหว่างไมโครคอนโทรลเลอร์ 29

2.3 ระบบเครือข่าย 31

2.3.1 ระบบเครือข่าย 31

2.3.2 สถาปัตยกรรมของอินเตอร์เน็ต 31

2.3.3 โมเดลโอเอสไอ 33

2.3.4 ทีซีพี/ไอพี 33

2.3.5 ไอพีแอดเดรส 44

2.4 การออกแบบทางซอฟท์แวร์และฮาร์ดแวร์ 46

2.4.1 การออกแบบทางซอฟท์แวร์ 47

2.4.1.1 โปรแกรมฝั่งไคลเอนท์และการตรวจสอบการ login 47

2.4.1.2 โปรแกรมฝั่งเซิร์ฟเวอร์ 51

2.4.1.3 โปรแกรมควบคุมสำหรับไมโครคอนโทรลเลอร์ 52

2.4.2 การออกแบบทางฮาร์ดแวร์ 54

2.4.2.1 วงจรแปลงสัญญาณ 54

2.4.2.2 วงจรควบคุม 55

2.4.2.3 วงจรขับรีเลย์ 56

2.4.2.4 วงจรตรวจสอบสถานะ 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การทดลอง 59
บทที่ 4 สรุปผลการทดลองและวิจารณ์ 65
เอกสารอ้างอิง 66
ภาคผนวก ก. วงจรรวม ก-1
ภาคผนวก ข. Microsoft Visual Basic เบื้องต้น ข-1
ภาคผนวก ค. ดาต้าชีท ค-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงการเปรียบเทียบมาตฐานการสื่อสาร RS-232 กับ RS485 6

ตารางที่ 2.2 เบอร์ไมโครคอนโทรลเลอร์พื้นฐาน 7

ตารางที่ 2.3 หน้าที่พิเศษของขาต่าง ๆ ของ PORT 3 8

ตารางที่ 2.4 รีจีสเตอร์ที่ใช้เป็น Timer 15

ตารางที่ 2.5 รีจีสเตอร์ TMOD (Timer Mode) 16

ตารางที่ 2.6 การใช้ Timer โหมดต่าง ๆ 16

ตารางที่ 2.7 แสดงความหมายแต่ละบิตของรีจีสเตอร์ TCON (Timer Control) 17

ตารางที่ 2.8 ค่าสูงสุดของการใช้ Timer โหมดต่าง ๆ 22

ตารางที่ 2.9 บิตต่าง ๆ ของรีจีสเตอร์ IE 24

ตารางที่ 2.10 รีจิสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการอินเตอร์รัพท์ 25

ตารางที่ 2.11 บิตและหน้าที่ต่าง ๆ ของรีจีสเตอร์ IP 25

ตารางที่ 2.12 แฟลกที่จะทำงานเมื่อถูกอินเตอร์รัพท์ 26

ตารางที่ 2.13 อินเตอร์รัพท์เวกเตอร์ของอินเตอร์รัพท์ต่างๆ 27

ตารางที่ 2.14 โอ เอส ไอ โมเดล 33

ตารางที่ 2.15 โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี เปรียบเทียบกับแบบจำลอง โอ เอส ไอ 39

ตารางที่ 2.16 หมายเลขพอร์ตที่ให้บริการต่างๆ 43

ตารางที่ 3.1 แสดงผลการทดลองของชั้นที่ 1 61

ตารางที่ 3.2 แสดงผลการทดลองของชั้นที่ 2 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 แสดงแนวความคิดของระบบการควบคุมอุปกรณ์ไฟฟ้า 1
รูปที่ 1.2 ระบบการควบคุมอุปกรณ์ไฟฟ้าโดยการติดผ่านอินเตอร์เน็ต 2
รูปที่ 2.1 แสดงรายละเอียดพอร์ตอนุกรม 5
รูปที่ 2.2 แสดงตัวอย่างการรับส่งข้อมูล RS-485 6
รูปที่ 2.3 การจัดขาของ 8051 9
รูปที่ 2.4 แสดงหน่วยความจำโปรแกรมของ 8051 10
รูปที่ 2.5 แสดงหน่วยความจำข้อมูลของ 8051 10
รูปที่ 2.6 แสดงหน่วยความจำข้อมูลภายใน 11
รูปที่ 2.7 แสดงรายละเอียดของ Special Function Register 12
รูปที่ 2.8 แสดงตำแหน่งการอ้างอิงระดับบิตของรีจีสเตอร์ SFR 13
รูปที่ 2.9 รีจีสเตอร์ที่ใช้เป็น Timer 14
รูปที่ 2.10 การทำงานของ Timer ในโหมดต่าง 17
รูปที่ 2.11 ความถี่ของสัญญาณนาฬิกาที่เข้าขา Timer 19
รูปที่ 2.12 การใช้บิตควบคุม TR 20
รูปที่ 2.13 ระบบทั้งหมดของ Timer 1 20
รูปที่ 2.14 ขั้นตอนการทำงานของโปรแกรมเมื่อถูกอินเตอร์รัพท์ 23
รูปที่ 2.15 แสดงการอินเตอร์รัพท์ 26
รูปที่2.16 แสดงสัญญาณของขา RXD , TXD 28
รูปที่ 2.17 แสดง Baudrate ใน Mode 1 28
รูปที่2.18 แสดง Baudrate ใน Mode 2 29
รูปที่ 2.19 การสื่อสารระบบ Multiprocessor Mode 29
รูปที่ 2.20 สัญญาณที่ทำงานแบบ Single Processor และ MultiProcessor 30
รูปที่ 2.21 แสดงเครือข่ายอินเตอร์เน็ต 32
รูปที่ 2.22 แสดงคลาสของไอพีแอดเดรส 45
รูปที่ 2.23 แสดงกราฟฟิกยูเซอร์อินเตอร์เฟสของโปรแกรมฝั่งไคลเอนต์ 47
รูปที่ 2.24 แสดงเมนูควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า 48
รูปที่ 2.25 แสดงหน้าจอของส่วนแก้ไขรายการ 48
รูปที่ 2.26 แสดงหน้าจอของ Login 49
รูปที่ 2.27 แสดง Help File 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 2.28 แสดงโฟลว์ชาร์ตการทำงานในส่วนการ Login และเชื่อมต่อกับเซิร์ฟเวอร์ 50

รูปที่ 2.29 แสดง โฟลว์ชาร์ต แสดงการใช้งานโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า 50

รูปที่ 2.30 แสดงโปรแกรมฝั่งเซิร์ฟเวอร์ 51

รูปที่ 2.31 แสดงเมนูการเซตค่าพอร์ตอนุกรมของฝั่งเซิร์ฟเวอร์ 51

รูปที่ 2.32 แสดงการทำงานของโปรแกรมฝั่งเซิร์ฟเวอร์ 52

รูปที่ 2.33 แสดงการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์ 53

รูปที่ 2.34 แสดงบล็อกไดอะแกรมและวงจรแปลงสัญญาณ 54

รูปที่ 2.35 แสดงวงจรควบคุมและส่วนขยายพอร์ต 55

รูปที่ 2.36 แสดงวงจรขับรีเลย์ 57

รูปที่ 2.37 แสดงวงจรตรวจสอบสถานะ 58

รูปที่ 3.1 โปรแกรมฝั่งไคลเอนท์ที่ใช้ในการทดลอง 56

รูปที่ 3.2 โปรแกรมทางฝั่งเซิร์ฟเวอร์ 60

รูปที่ 3.3 แสดงเมนูของส่วนการควบคุม 60

รูปที่ 3.4 แสดงชุดที่ประกอบสำเร็จของวงจรขับโหลด และตรวจสอบสภาวะการทำงาน 62

รูปที่ 3.5 แสดงชุดประกอบสำเร็จรูปของวงจรที่ในการควบคุมทั้งหมด 62

รูปที่ 3.6 แสดงการทำงานชุดจำลองอาคารสำนักงาน 2 ชั้น 63

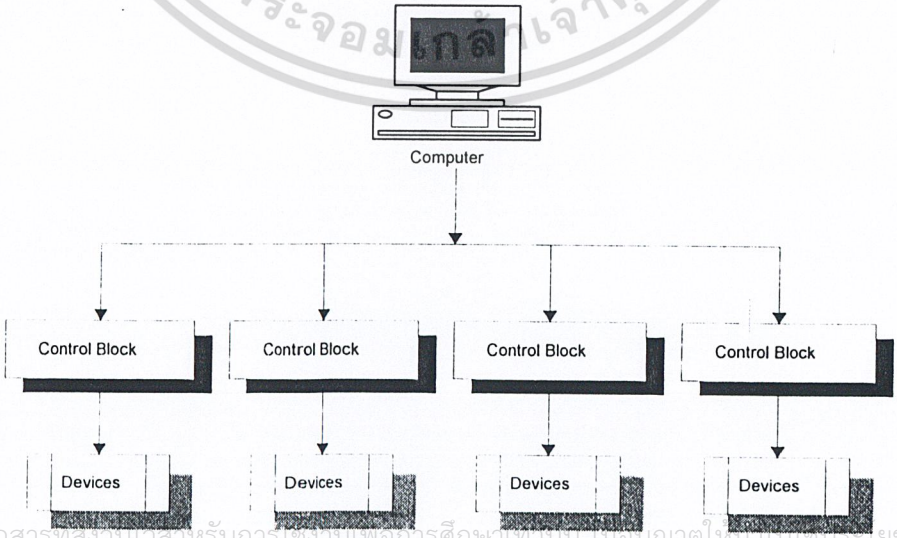
รูปที่ 3.7 แสดงการทำงานอุปกรณ์ไฟฟ้าชั้นที่ 2 ของชุดจำลอง 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

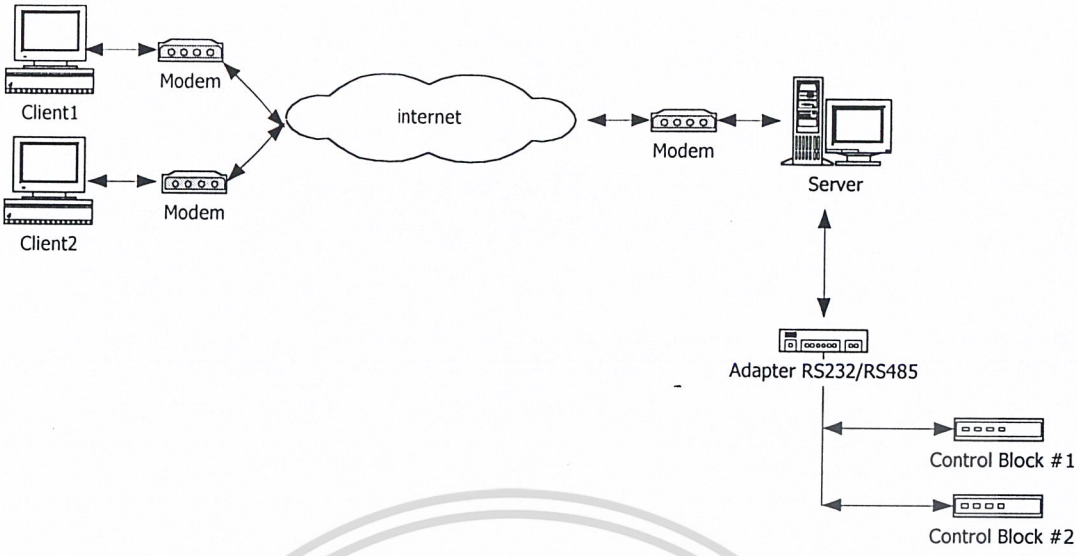
1.1 ความเป็นมา

ปัจจุบันเทคโนโลยีการพัฒนาทางเทคโนโลยีและการติดต่อสื่อสารได้พัฒนาไปอย่างรวดเร็วทำให้มีการพัฒนาอุปกรณ์ต่าง ๆ เพื่อที่จะอำนวยความสะดวกและช่วยในการทำงานต่างๆแทนมนุษย์ ซึ่งอุปกรณ์ส่วนใหญ่เหล่านั้นจำเป็นที่จะต้องใช้พลังงานไฟฟ้า อีกทั้งจำนวนอุปกรณ์ไฟฟ้าที่มากขึ้นทุกวัน ความต้องการในการใช้งานที่เพิ่มขึ้น ความสะดวกสบาย หรือแม้แต่การทำงานที่เร่งรีบ เพื่อประหยัดเวลาในการทำงาน และเพื่อเป็นการนำเอาพลังงานไฟฟ้ามาใช้ให้เกิดประโยชน์อย่างมีประสิทธิภาพและคุ้มค่าสูงสุดนั้น แนวทางหนึ่งที่จะตอบสนองความต้องการดังกล่าว ก็คือการนำเอาคอมพิวเตอร์ส่วนบุคคล และเทคโนโลยีอินเตอร์เน็ต มาประยุกต์ใช้ร่วมกันในการสร้างระบบควบคุมอุปกรณ์ไฟฟ้าที่สามารถใช้งานง่าย มีความยืดหยุ่น สามารถตอบสนองความต้องการของผู้ใช้ได้ โดยใช้การสั่งงานระยะไกลจากเครื่องไคลเอนต์ไปยังเครื่องเซิร์ฟเวอร์ ที่ใช้โปรโตคอลมาตรฐานคือ TCP / IP เป็นตัวกลางในการติดต่อสื่อสารระหว่างเครื่องไคลเอนท์และเครื่องเซิร์ฟเวอร์ และเมื่อข้อมูลมาถึงเครื่องเซิร์ฟเวอร์ เครื่องเซิร์ฟเวอร์ก็ทำการส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 เพื่อทำการกระจายการควบคุมไปสู่ส่วนควบคุมย่อยต่อไป และเมื่ออุปกรณ์ที่จะควบคุมอยู่ไกลจากเครื่องคอมพิวเตอร์มาก จึงทำการแปลงข้อมูลจากมาตรฐาน RS-232 เป็นมาตรฐาน RS-485 เพื่อให้การควบคุมในระยะไกลเป็นไปอย่างสะดวก อีกทั้งยังสามารถแสดงผลในรูปแบบกราฟิกและสามารถที่จะตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าทางหน้าจอคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1.1 แสดงแนวความคิดของระบบการควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 1.2 ระบบการควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ โดยการติดผ่านอินเตอร์เน็ต

1.2 ขอบเขตการทำงานของโครงงาน

ขอบเขตของโครงงานนี้ จะประกอบไปด้วย 4 ส่วนหลัก คือ

1.ส่วนของเครื่องคอมพิวเตอร์ที่มีหน้าที่สั่งการควบคุมและแสดงสถานะของอุปกรณ์ได้ 256 จุด โดยผ่านทางหน้าจอคอมพิวเตอร์

2.ส่วนของวงจรแปลงสัญญาณ ที่มีหน้าที่ในการเป็นตัวกลางในการแปลงสัญญาณข้อมูลระหว่างคอมพิวเตอร์กับวงจรควบคุม โดยอาศัยการสื่อสารแบบ RS-485 ที่สามารถเชื่อมกับวงจรควบคุมได้จำนวน 32 ตัว

3.ส่วนของวงจรควบคุม ทำหน้าที่ ควบคุมการเปิดปิดตามคำสั่งที่รับมาจากคอมพิวเตอร์ และส่งสถานะให้คอมพิวเตอร์

4.ส่วนของวงจรตรวจสอบสภาวะ ที่มีหน้าที่ในการตรวจสอบสถาวะของอุปกรณ์ไฟฟ้าว่าได้ทำงานตามคำสั่งที่ได้ส่งมาหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและการออกแบบ

2.1 การสื่อสารผ่านพอร์ตอนุกรม

การรับส่งข้อมูลแบบอนุกรมนั้นเป็นการรับส่งข้อมูลครั้งละหนึ่งบิต แต่เราก็สามารถรับส่งข้อมูลได้คราวละหลาย ๆ บิตได้ ถ้าเราได้มีการตกลงกันระหว่างตัวส่งและตัวรับว่า จะส่งคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงจะทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน ซึ่งในด้านของจำนวนสายที่ใช้ในการรับส่งแบบอนุกรมนี้จะใช้สายน้อยกว่าแบบขนานมาก คือประมาณ 2-3 เส้นเท่านั้นเอง แต่ว่าอัตราการรับส่งข้อมูลนั้นต่ำกว่าแบบขนาน นอกจากนั้นการรับส่งแบบอนุกรมยังสามารถส่งได้ไกลกว่าแบบขนาน

การสื่อสารแบบอนุกรมนั้นสามารถแบ่งออกได้เป็น 2 แบบคือ

1. การสื่อสารอนุกรมแบบซิงโครนัส
2. การสื่อสารอนุกรมแบบอะซิงโครนัส

การสื่อสารแบบซิงโครนัส(synchronous interface)

การส่งข้อมูลจะไม่มีบิตเริ่มต้น และบิตสุดท้าย ซึ่งการส่งผ่านจะทำได้โดยการส่งผ่านข้อมูลโดยตรง จะมีสัญญาณนาฬิการ่วมอยู่กับการรับส่งสัญญาณด้วย อย่างเช่นคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา อีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อจะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิการ ข้อมูล และกราวด์

การสื่อสารข้อมูลแบบอะซิงโครนัส (Asynchronous interface)

จะตรงข้ามกับการสื่อสารแบบซิงโครนัส คือการส่งผ่านข้อมูลจะมีบิตเริ่มต้นและบิตสุดท้าย หมายถึงการรับส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัสแต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per secone :bps)และจะไม่คำนึงถึงช่วงเวลาการขอใช้ข้อมูลของระบบ ซึ่งจะเหมาะสมการใช้กับสองระบบที่มีความแตกต่างกันในด้านความเร็ว เช่น การสั่งงานเครื่องพิมพ์จากเครื่องคอมพิวเตอร์ ซึ่งเครื่องคอมพิวเตอร์จะมีความเร็วในการรับส่งข้อมูลที่เร็วกว่าเครื่องพิมพ์ เครื่องคอมพิวเตอร์ก็จะรอให้เครื่องพิมพ์ทำงานให้เสร็จแต่ละขั้นตอนแล้วจึงค่อยส่งข้อมูลต่อไปให้กับเครื่องพิมพ์อีก ทำให้ทราบข้อดีว่า อุปกรณ์ราคาถูกกว่าการส่งข้อมูลแบบซิงโครนัส และทราบข้อเสียว่า จะมีความผิดพลาดเนื่องจากบิตสิ้นสุด อาจเกิดจากสัญญาณรบกวนที่เหมือนกับบิตเริ่มต้น และวงจรจะไม่บอกถึงความผิดพลาดที่เกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนหลักคือ

1.บิตเริ่มต้น มีขนาด 1 บิต

2.บิตข้อมูลแบบอนุกรม จะมีขนาด 5,6,7 หรือ 8 บิต

3.บิตตรวจสอบพาริตี้ จะมีขนาด 1 บิต หรือ ไม่มีก็ได้

4.บิตปิดท้าย จะมีขนาด 1,1.5 หรือ 2 บิต

ค่าพารามิเตอร์แรกที่ควรรู้ คือ ความเร็วในของพอร์ตที่ใช้ในการสื่อสารรับส่งข้อมูลซึ่งจะมีหน่วยเรียกการส่งผ่านข้อมูลเป็น จำนวนบิตต่อวินาที ความเร็วที่พอร์ตอนุกรมสามารถรองรับได้คือตั้งแต่ 110 bps ถึง 19.2 Kbps

ค่าพารามิเตอร์สอง คือ ขนาดของข้อมูล หรือจำนวนบิตในข้อมูลแต่ละไบท์ ซึ่งมีสองทางเลือกได้แก่ 7 บิตหรือ 8 บิต ส่วนใหญ่จะใช้ 7 บิต

ค่าพารามิเตอร์ที่สาม คือ ค่าพารีรี้ ซึ่งมีความสำคัญในการตรวจสอบความผิดพลาดที่เกิดขึ้นได้ในการรับส่งข้อมูล ซึ่งตามมาตรฐาน RS-232 สามารถระบุค่าพาริตี้ได้จากทางเลือก 3 แบบ คือ แบบคู่ หรือ คี่ หรือ ไม่ต้องใช้พาริตี้

ค่าพารามิเตอร์สุดท้าย คือ บิตเริ่มต้นและบิตสุดท้ายซึ่งเป็นบิตหยุด ซึ่งข้อมูลแต่ละไบท์จะถูกล้อมรอบไปด้วยบิตเริ่มต้นและบิตสุดท้ายของการส่งข้อมูลในชุดนั้น ๆ และเราสามารถกำหนดจำนวนบิตหยุดเป็น 1 บิต หรือ 2 บิตก็ได้

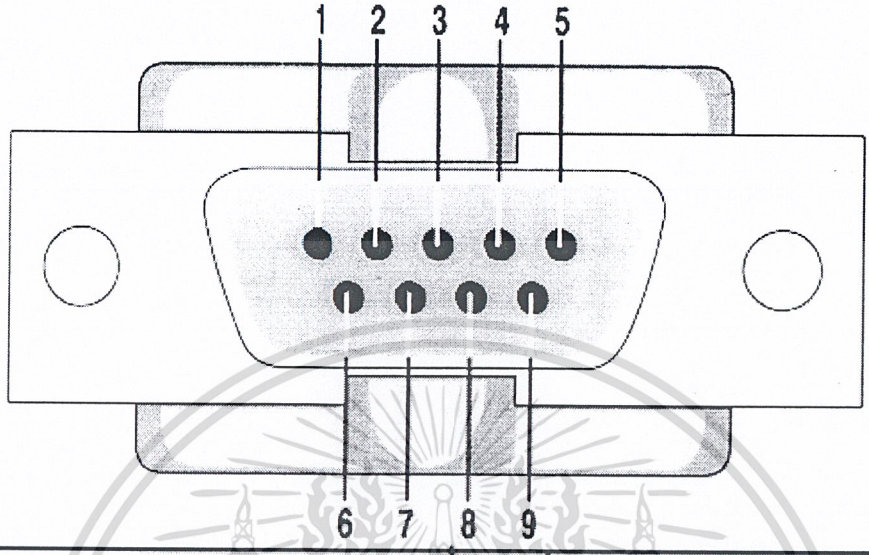
อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า UART (Universal Asynchronous Receiver / Transmitter) และอัตราความเร็วในการรับส่งแบบอะซิงโครนัส คือ ค่าบอดเรต ซึ่งคือค่าจำนวนบิตที่ใช้ในการับและส่งข้อมูล และบอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 คือ 110, 150, 300, 600, 1200, 4800, 9600, และ 19200 บิตต่อวินาที และจะมีค่าเพิ่มขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มจะสามารถกำหนดค่าบอดเรตได้สูงสุดถึง 115200 บิตต่อวินาที

2.1.1 การรับส่งข้อมูลแบบอนุกรมตามมาตรฐาน RS-232

การรับส่งข้อมูลแบบอนุกรมขั้นพื้นฐานที่มีใช้งานและอ้างอิงจนถึงปัจจุบันนี้มีคุณลักษณะการเชื่อมต่อที่ตรงตามมาตรฐาน ANSI (American National Standards Institute) คืออุปกรณ์และวงจรที่ใช้ในการรับส่งข้อมูลสามารถรับส่งข้อมูลที่มีสถานะ 0 และ 1 เท่านั้น ซึ่งการกำหนดมาตรฐานแบบนี้ทำให้อุปกรณ์ที่ผลิตจากผู้ผลิตรายอื่นสามารถสร้างอุปกรณ์ที่สามารถทำงานร่วมกันได้ และอุปกรณ์สื่อสารเหล่านี้ได้รับความนิยมเป็นอย่างมาก และยังเป็นมาตรฐานในคอมพิวเตอร์ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บุคคลอีกด้วย โดยได้ออกแบบให้มีโครงสร้างในการรับส่งข้อมูลแบบจุดต่อจุด ที่มีอัตราการรับส่งข้อมูลสูงสุดอยู่ที่ 20 kbps ในระยะทาง 15-20 เมตรเท่านั้น



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

รูปที่ 2.1 แสดงรายละเอียดพอร์ตอนุกรม

2.1.2 การรับส่งข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

เป็นการรับส่งข้อมูลที่ได้รับการพัฒนามาจากการรับส่งข้อมูลแบบอนุกรมมาตรฐานRs-422 ซึ่งยังคงใช้การเปรียบเทียบเทียบสัญญาณระหว่างสายตัวนำสองเส้น การลดทอนของขนาดสัญญาณของข้อมูลมีขนาดใกล้เคียงกัน การเปลี่ยนแปลงของขนาดของข้อมูลมีน้อย ทำให้ไม่เกิดการผิดพลาดในการรับส่งข้อมูล แต่จะเป็นการส่งข้อมูลแบบสองทิศทางในสายสัญญาณเพียงแค่สองเส้น โดยรับส่งได้ทิศทางเดียวต่อครั้ง ซึ่งสามารถต่อขนานร่วมกันได้หลายจุด และสามารถส่งผ่านข้อมูลได้ระยะทางไกลถึง 1200 เมตร โดยตัวส่งสามารถต่อตัวรับได้ประมาณ 32 ตัว

การส่งข้อมูลแบบ RS485 จะใช้การส่งแบบสมดุล (Balanced Transmission) เป็นการส่งข้อมูลที่ใช้ผลต่างของระดับแรงดันของสัญญาณ 2 เส้น ซึ่งข้อมูล 1 เส้นจะใช้สายส่งสัญญาณ 2 เส้น โดยมีช่วงการเปลี่ยนแปลงของระดับสัญญาณอยู่ในช่วง ±200 มิลลิโวลต์เท่านั้น ทำให้ทนต่อสัญญาณรบกวนได้ดี สามารถส่งข้อมูลได้ด้วยความเร็วสูงและสูงสุดได้ถึง 2.5 เมกะบิตต่อวินาที และจะสามารถส่งได้ไกลมากขึ้น คือประมาณ 100 เมตร

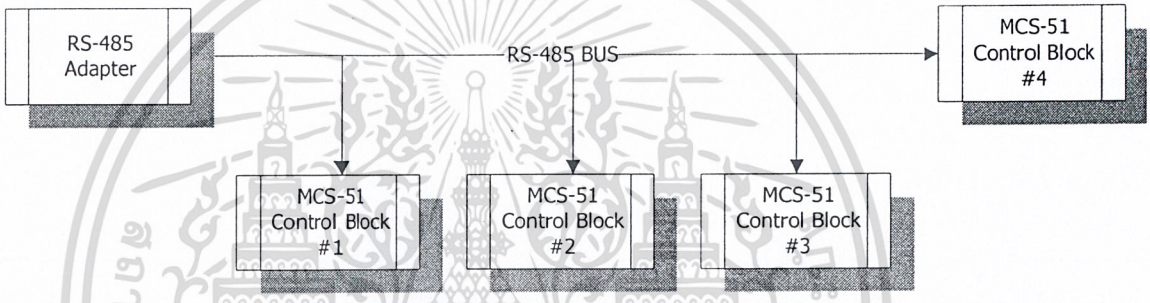
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลด้วย RS-485 มีอยู่ด้วยกัน 2 แบบคือ

- สายสัญญาณข้อมูล 4 เส้น
- ใช้สายสัญญาณข้อมูล 2 เส้น

แบบใช้สายสัญญาณ 4 เส้น สัญญาณข้อมูลที่รับและส่งนั้นจะใช้สายสัญญาณคนละคู่กัน คือส่ง 2 เส้นและรับ 2 เส้น สื่อสารแบบฟูลดูเพล็ก (Full duplex) คือสามารถรับส่งข้อมูลได้ในเวลาเดียวกัน

แบบใช้สายสัญญาณ 2 เส้น สัญญาณข้อมูลที่รับและส่งนั้นจะใช้สายสัญญาณร่วมกัน และใช้การสื่อสารแบบฮาร์ฟดูเพล็ก คือไม่สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน และในเครือข่ายหนึ่งจะมีผู้ส่งได้เพียงผู้เดียวเท่านั้นจะส่งพร้อมกันไม่ได้



รูปที่ 2.2 แสดงตัวอย่างการรับส่งข้อมูล RS-485

เบอร์ไอซีของตัวรับ-ส่งข้อมูลของ RS-485 มีหลายเบอร์ให้เลือกใช้ คือ SN75156, SN77177, SN75178, SN75179 เป็นต้น

และข้างล่างนี้ได้แสดงการเปรียบเทียบคุณสมบัติของ RS-232 และ RS-485

พารามิเตอร์	RS-232	RS-485
Mode	Single-ended	Differential
จำนวนตัวรับจำนวนตัวส่ง	1 ตัวรับ 1 ตัวส่ง	32 ตัวรับ 32 ตัวส่ง
ความยาวสายสัญญาณ	15 m	1200 m
อัตราการส่งข้อมูล	20 Kb/s	35 Mb/s
แรงดัน output ต่ำสุด	± 5 V	± 1.5 V
แรงดัน output สูงสุด	± 3 V	± 200 m V

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงการเปรียบเทียบมาตฐานการสื่อสาร RS-232 กับ RS485

2.2 ไมโครคอนโทรลเลอร์ 8051

2.2.1 ไมโครคอนโทรลเลอร์ในตระกูล 8051 นั้นมีอยู่หลายเบอร์ให้เลือกใช้ ซึ่งแต่ละเบอร์จะมีความสามารถพิเศษมากน้อยแตกต่างกันไป ดังจะแสดงเบอร์พื้นฐานในตระกูลนี้

ชื่อเบอร์	หน่วยความจำภายใน		จำนวนไทเมอร์ /เคาน์เตอร์	จำนวนอินเตอร์รัพต์
	เก็บโปรแกรม	เก็บข้อมูล		
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16 bit	6
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16 bit	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16 bit	5
8032AH	ไม่มี	256 x 8 RAM	3 x 16 bit	6
8031AH	ไม่มี	128 x 8 RAM	2 x 16 bit	5
8031	ไม่มี	128 x 8 RAM	2 x 16 bit	5
8751H	4K x 8 EROM	128 x 8 RAM	2 x 16 bit	5
8751H-12	4K x 8 EROM	128 x 8 RAM	2 x 16 bit	5

ตารางที่ 2.2 เบอร์ไมโครคอนโทรลเลอร์พื้นฐาน

หน้าที่การใช้งานของแต่ละขาของไมโครคอนโทรลเลอร์ 8051

V_{cc} : สำหรับแหล่งจ่ายไฟฟ้า (+5v.)

V_{ss} : สำหรับต่อกราวด์

P0 : เป็นขาพอร์ต 0 ของ 8051 ที่มีขนาด 8 บิตชนิดสองทิศทาง ซึ่งแต่ละบิตสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไปหากต้องการให้เป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังบิตนั้น โดยแต่ละบิตเมื่อเป็นเอาต์พุตจะสามารถต่อพ่วงกับอุปกรณ์ TTL แบบ LS ได้ 8 ตัว และยังเป็นขาให้สัญญาณ Multiplex ระหว่างสัญญาณข้อมูลกับสัญญาณ Address 8 บิตแรก ในกรณีที่ใช้หน่วยความจำภายนอก

P1 : เป็นขาพอร์ต 1 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasibi-directional ซึ่งแต่ละบิตสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไปหากต้องการให้เป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังบิตนั้น และสามารถต่อพ่วงกับอุปกรณ์ LS TTL ได้ 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P2 : เป็นขาพอร์ต 2 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasibi-directional เช่นเดียวกับพอร์ต 1 นอกจากนี้พอร์ต2นี้ยังทำหน้าที่ให้สัญญาณ Address 8 บิตบน ในกรณีที่ใช้หน่วยความจำภายนอก ในกรณีอ้าง Address หน่วยความจำขนาด 16 บิต ดังนั้นขณะที่

ใช้หน่วยความจำภายนอก จะต้องไม่มีการเขียนข้อมูลใด ๆ ไปที่พอร์ต 2 จะทำให้เกิดความผิดพลาดการทำงานได้

P3 : เป็นขาพอร์ต 3 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasibi-directional เช่นเดียวกับขาพอร์ต 1 และพอร์ต 2 แต่พอร์ต 3 นี้จะมีหน้าที่พิเศษดังตารางที่ 2.3

ขาพอร์ต	หน้าที่พิเศษ
P3.0	R x D (สำหรับรับข้อมูลแบบอนุกรม)
P3.1	T x D (สำหรับส่งข้อมูลแบบอนุกรม)
P3.2	INT0 (ขาอินเตอร์รัพท์ภายนอก 0)
P3.3	INT1 (ขาอินเตอร์รัพท์ภายนอก 1)
P3.4	T0 (ขาอินพุตของ Timer 0)
P3.5	T1 (ขาอินพุตของ Timer 1)
P3.6	WR (สำหรับสัญญาณเขียนหน่วยความจำข้อมูลภายนอก)
P3.7	RD (สำหรับสัญญาณอ่านหน่วยความจำข้อมูลภายนอก)

ตารางที่ 2.3 หน้าที่พิเศษของขาต่าง ๆ ของ PORT 3

ดังนั้น เมื่อมีการใช้สัญญาณดังกล่าว จึงไม่ควรเขียนข้อมูลไปที่พอร์ต 3 จะทำให้การทำงานของ 8051 ผิดพลาดได้

RST : เป็นขาสำหรับรีเซตการทำงานของ 8051 โดยการให้ลอจิกหนึ่งเป็นเวลาอย่างน้อย 2 ช่วง Machine Cycle

ALE : เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขา พอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก

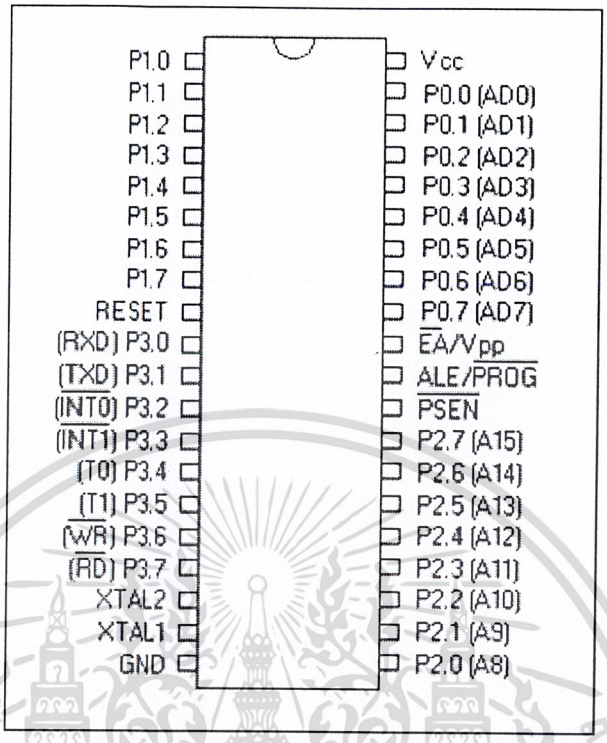
PSEN : เป็นขาสัญญาณเพื่อร้องขอติดต่อหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านหน่วยความจำโปรแกรมภายนอก

EA : เป็นขาใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมภายนอกหรือภายในไมโครคอนโทรลเลอร์ โดยที่ให้ลอจิก 0 จะอ่านหน่วยความจำโปรแกรมภายนอก และลอจิก 1 จะอ่านหน่วยความจำโปรแกรมภายใน

XTAL1 : ขาเข้าของวงจรกำเนิดความถี่อ้างอิงภายในของ 8051

XTAL2 : ขาออกของวงจรกำเนิดความถี่อ้างอิงภายในของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

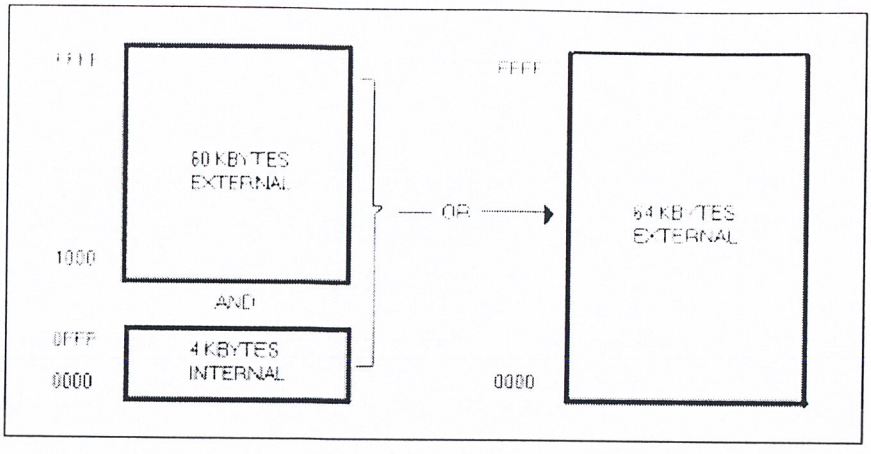


รูปที่ 2.3 การจัดขาของ 8051

โครงสร้างหน่วยความจำของ 8051

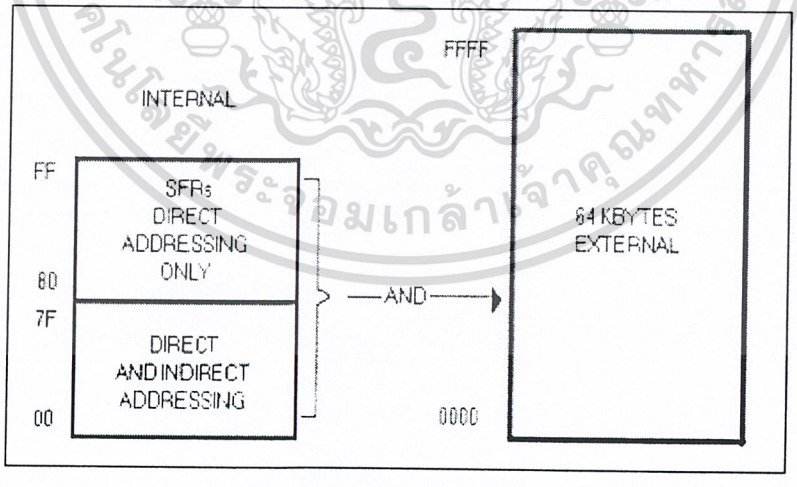
ดังที่กล่าวมาแล้ว 8051 จะแบ่งหน่วยความจำออกเป็นสองส่วน ได้แก่ หน่วยความจำสำหรับโปรแกรมและหน่วยความจำสำหรับเก็บข้อมูล โดยมีขนาดของแต่ละส่วนเท่ากับ 64 กิโลไบต์ ในส่วนของหน่วยความจำโปรแกรมจะเป็นส่วนหน่วยความจำสำหรับอ่านอย่างเดียว โดยที่ 8051 จะใช้สัญญาณ PSEN ในการอ่านเท่านั้น แต่หน่วยความจำข้อมูลของ 8051 จะสามารถอ่านและเขียนได้โดยใช้สัญญาณ RD และ WR ตามลำดับ แต่อย่างไรก็ตาม ผู้ใช้สามารถรวมหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลเข้าด้วยกันได้ โดยนำสัญญาณ RD และ PSEN มาต่อเข้าวงจรแอนเกต สำหรับสร้างสัญญาณในการอ่านหน่วยความจำ นอกจากนี้หน่วยความจำโปรแกรมยังแบ่งออกเป็นภายนอกและภายในของ 8051 ดังแสดงในรูปที่ 2.4 รูปที่ 2.5 โดยรูปที่ 2.4 แสดงหน่วยความจำโปรแกรมในกรณีที่เลือกใช้หน่วยความจำภายนอกและภายใน ในด้านซ้ายมือเป็นส่วนหนึ่งของหน่วยความจำโปรแกรมภายในที่มีขนาด 4 กิโลไบต์ของ 8051 ส่วนที่เหลือจะเป็นหน่วยความจำภายนอก ส่วนด้านขวามือแสดงหน่วยความจำโปรแกรมเมื่อเลือกให้ติดต่อหน่วยความจำภายนอกทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงหน่วยความจำโปรแกรมของ 8051

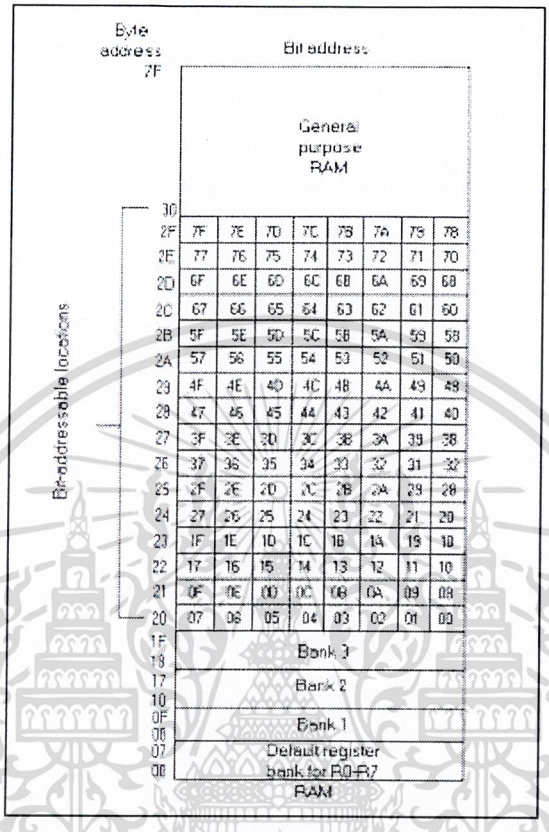
สำหรับหน่วยความจำข้อมูลของ 8051 สามารถแบ่งออกเป็นภายนอกและภายใน โดยหน่วยความจำภายนอกแสดงไว้ด้านขวามือของรูปที่ 2.4 ซึ่งมีขนาด 64 กิโลไบต์ ส่วนหน่วยความจำข้อมูลภายในแสดงไว้ด้านซ้ายของรูปที่ 2.4 โดยหน่วยความจำภายในของ 8051 แบ่งออกเป็นสองส่วน ได้แก่ ส่วนของหน่วยความจำข้อมูลที่สามารถอ้างอิงแบบ Direct และ Indirect ซึ่งมีขนาด 128 ไบต์ กับหน่วยความจำที่อ้างอิงได้เฉพาะแบบ Direct หรือในส่วนนี้จะเรียกอีกแบบหนึ่งว่า SFR (Special Function Register) โดยจะแบ่งกล่าวได้ดังนี้



รูปที่ 2.5 แสดงหน่วยความจำข้อมูลของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของหน่วยความจำข้อมูลภายในที่อ้างอิงแบบ direct และ Indirect นั้นจะสามารถแบ่งออกได้ 3 ส่วน ดังแสดงในรูปที่ 2.6 โดยมีรายละเอียดดังนี้



รูปที่ 2.6 แสดงหน่วยความจำข้อมูลภายใน

ส่วนที่ 1 เรียกว่า Register Banks 0-3 ซึ่งอยู่ที่ตำแหน่งความจำข้อมูลภายใน ตั้งแต่ 00H ถึง 1FH จำนวน 32 ไบต์ โดยจะแบ่งออกเป็นชุด ชุดละ 8 ไบต์จำนวน 4 ชุด ซึ่งแต่ละชุดจะมีชื่อเรียกเป็น R0 ถึง R7 จะเป็น Register ที่ใช้งาน โดยเมื่อ 8051 ถูกรีเซต Register Bank 0 จะถูกเลือกใช้

ส่วนที่ 2 เรียกว่า Bit Addressable Area ซึ่งมีขนาด 16 ไบต์ที่ตำแหน่งหน่วยความจำข้อมูล 20H ถึง 2FH ในส่วนนี้สามารถที่จะอ้างอิงข้อมูลได้เป็นระดับบิตถึง 128 บิต โดยการอ้างอิงตำแหน่งโดยตรงในลักษณะบิต ตั้งแต่ตำแหน่ง 00H ถึง 7FH

ส่วนที่ 3 เรียกว่า Scratch Pad Area จะอยู่ที่ตำแหน่งตั้งแต่ 30H ถึง 7FH ซึ่งเป็นบริเวณหน่วยความจำข้อมูลภายในเอนกประสงค์ที่ผู้ใช้สามารถใช้ได้โดยตรง นอกจากนี้ยังสามารถใช้หน่วยความจำข้อมูลบริเวณนี้สำหรับการเก็บข้อมูลแบบ Stack ได้ด้วย

ในส่วนของหน่วยความจำข้อมูลภายในที่ใช้อ้างอิงแบบ Direct เพียงอย่างเดียวหรือที่เรียกว่า SFR ซึ่งเป็นส่วนสำหรับเก็บหรือกำหนดการทำงานภายในของ 8051

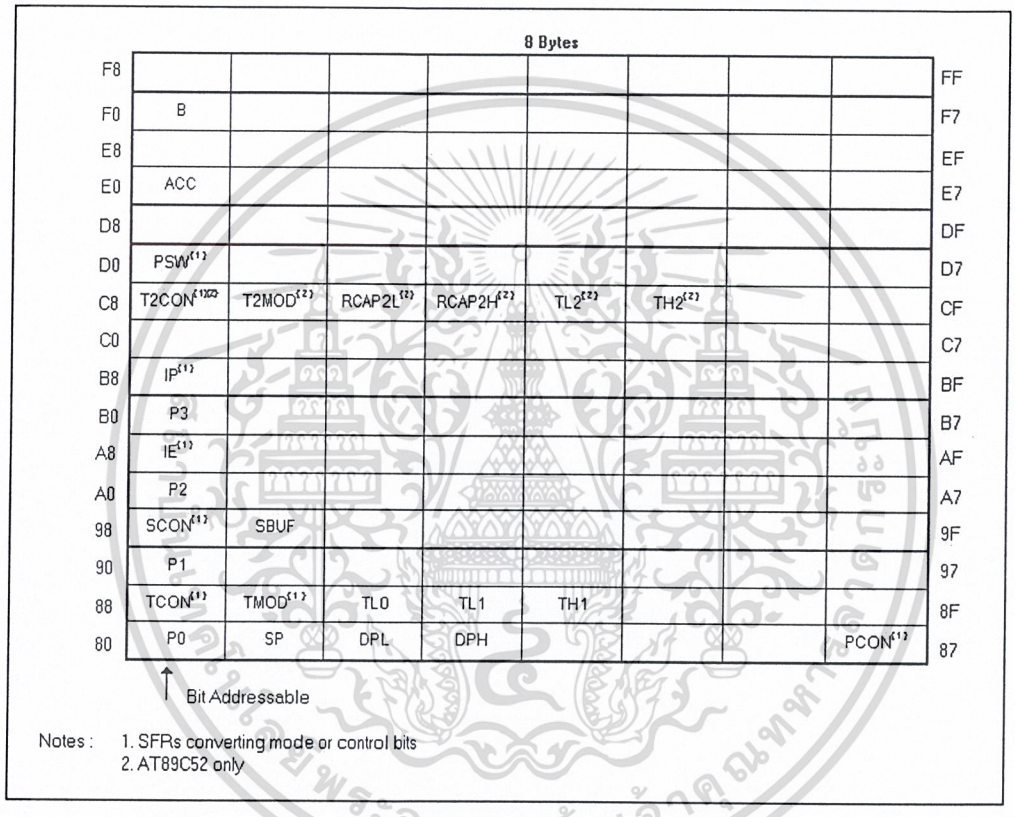
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของบริเวณนี้จะมีขนาด 128 ไบต์แต่ในการใช้งานนั้นใช้ได้เฉพาะตำแหน่ง หากผู้ใช้อ้างตำแหน่งนอกเหนือจากนี้นั้นจะได้ข้อมูลที่คาดเดาไม่ได้ โดยแต่ละตำแหน่งจะมีหน้าที่ดังนี้

ACC : เป็น Accumulator ซึ่งเป็นรีจีสเตอร์สำหรับการประมวลผลทางคณิตศาสตร์และลอจิก โดยผู้ใช้สามารถอ้างอิงได้ในรูปแบบของไบต์หรือระดับบิตได้

B : เป็นรีจีสเตอร์พิเศษสำหรับใช้กับคำสั่งในการคูณหรือหาร นอกจากนี้ยังใช้เป็นรีจีสเตอร์สำหรับเก็บพักข้อมูลได้

PSW : เป็นรีจีสเตอร์ Program Status Word หรือแฟลกจะแสดงสถานะการทำงานของ 8051 สำหรับการตรวจสอบซึ่งจะอธิบายรายละเอียดในภายหลัง



รูปที่ 2.7 แสดงรายละเอียดของ Special Function Register

SP : เป็นรีจีสเตอร์สำหรับชี้หน่วยความจำข้อมูลภายในสำหรับการเก็บแบบ Stack

DPTR : เป็นรีจีสเตอร์ขนาด 16 บิต โดยแบ่งเป็น 8 บิตบนและ 8 บิตล่าง ให้สำหรับชี้ตำแหน่งของหน่วยความจำข้อมูลภายนอกหรือสำหรับการอ่านตารางข้อมูลของหน่วยความจำโปรแกรม

P0 : เป็นรีจีสเตอร์สำหรับพอร์ต 0 ของ 8051

P1 : เป็นรีจีสเตอร์สำหรับพอร์ต 1 ของ 8051

P2 : เป็นรีจีสเตอร์สำหรับพอร์ต 2 ของ 8051

P3 : เป็นรีจีสเตอร์สำหรับพอร์ต 3 ของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP : เป็นรีจีสเตอร์สำหรับกำหนดลำดับความสำคัญของการอินเตอร์รัพท์ของ 8051

IE : เป็นรีจีสเตอร์สำหรับกำหนดการรับหรือไม่รับการอินเตอร์รัพท์ของ 8051

TMOD : เป็นรีจีสเตอร์สำหรับควบคุมหน้าที่ของ Timer/Counter ของ 8051

TCON : เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ Timer/Counter ของ 8051

T2CON : เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ Timer/Counter 2 ของ 8052

TH0 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 0 8บิตบน

TL0 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 0 8บิตล่าง

TH1 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 1 8บิตบน

TL1 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 1 8บิตล่าง

TH2 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 2 8บิตบนของ 8052

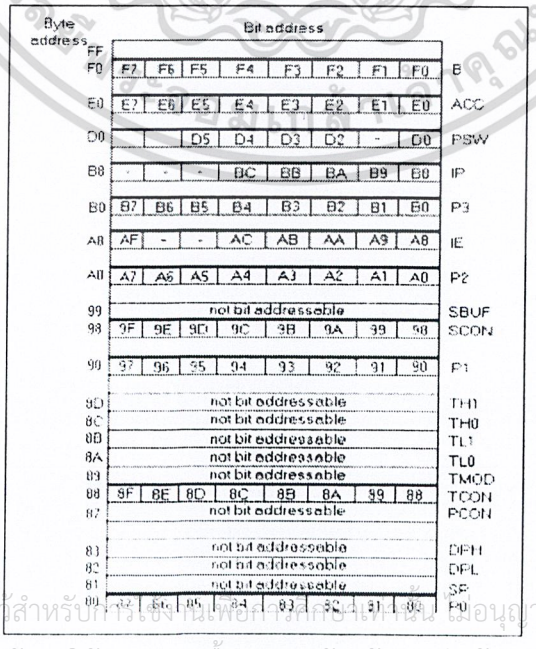
TL2 : เป็นรีจีสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 2 8บิตล่างของ 8052

RCAP2H : เป็น Capture Register ของ Timer/Counter 2 8บิตบนของ 8052SCON เป็นรีจีสเตอร์สำหรับควบคุมการรับส่งข้อมูลแบบอนุกรมของ 8051

SBUF : เป็นรีจีสเตอร์สำหรับเก็บพักข้อมูลที่ได้จากการรับส่งข้อมูลแบบอนุกรมของ 8051

PCON : เป็นรีจีสเตอร์สำหรับควบคุมการทำงานของ 8051 ด้านเกี่ยวกับการใช้กำลังไฟฟ้า

ในส่วนของรีจีสเตอร์ SFR นี้สามารถที่จะอ้างอิงในระดับบิตได้โดยตำแหน่งการอ้างอิงระดับบิตแสดงไว้ในตารางต่อไปนี้

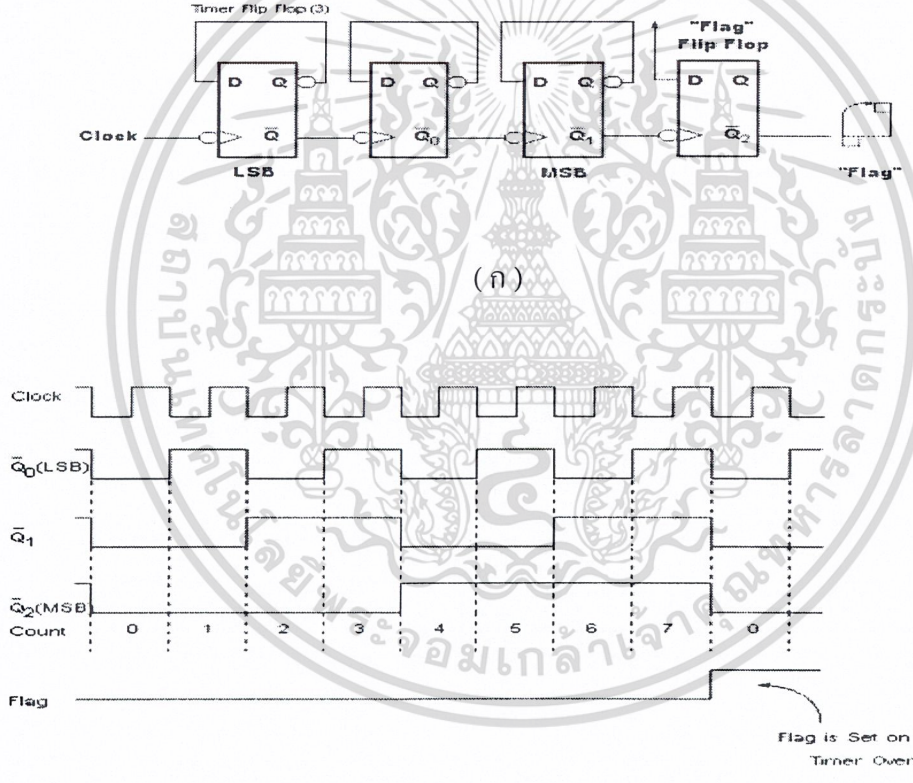


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 แสดงตำแหน่งการอ้างอิงระดับบิตของรีจีสเตอร์ SFR

TIMER

ตัว Timer อาจพิจารณาได้ง่าย ๆ ว่าเป็นตัวฟลิปฟลอปมาต่อเรียงกันโดยมี Clock เป็นอินพุต สำหรับเอาต์พุตที่ออกมาจากฟลิปฟลอปแต่ละตัวจะถูกหารด้วย 2 พิจารณาการต่อฟลิปฟลอปตามรูปที่ 2.9 ถ้าใส่ Clock เข้าไปในฟลิปฟลอปตัวแรก ความถี่ของ Clock ที่ออกจากเอาต์พุตตัวแรกจะถูกหารด้วย 2 และเอาต์พุตนี้จะต่อกับฟลิปฟลอปตัวที่สอง และสัญญาณที่ออกมาจะถูกหารด้วย 2 อีก ดังนั้น ถ้ามีฟลิปฟลอปต่ออยู่ n Stages จะหารสัญญาณนาฬิกาได้ 2^n ถ้าให้เอาต์พุต Stage สุดท้ายของ Timer เป็น Overflow Flip-Flop หรือ Flag และจะให้เอาต์พุตออกมาเมื่อการนับเป็น Overflow เช่น ถ้าเป็นตัวนับแบบ 16 บิต (มีฟลิปฟลอปต่ออยู่ 16 ตัว) วงจรจะนับตั้งแต่ 0000H ถึง FFFFH เมื่อฟลิปฟลอปเปลี่ยนจาก FFFFH เป็น 0000H จะให้บิต Overflow ออกมา



(ก)

(ข)

รูปที่ 2.9 รีจีสเตอร์ที่ใช้เป็น Timer

พิจารณารูป 2.9(ก) เป็น 3-bit Timer โดยฟลิปฟลอปแต่ละตัวจะนำขา Q มาต่อกับ D ซึ่งอาจเรียกว่าเป็นการใช้ฟลิปฟลอปแบบ Divide-by-two Mode โดยความถี่ของสัญญาณที่ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟลิปฟลอปแต่ละตัวจะมีค่าหารสองจากสัญญาณนาฬิกาที่เข้ามา เมื่อนับไปถึงค่า 111 (หรือ $Q_2 = 1$, $Q_1 = 1$, $Q_0 = 1$) และเปลี่ยนกลับมาเป็น 000 จะให้บิต Flag ออกมา ดังแสดงในรูปที่ 2.7(ข)

ใน MCS – 51 จะมีตัวจับเวลาอยู่ภายในชิพ ถ้าเป็นเบอร์ 8051 หรือ 8031 จะมี 2 ตัว คือ Timer 0 และ Timer 1 แต่ถ้าเป็นเบอร์ 8052 จะมีเพิ่มอีกหนึ่งตัวคือ Timer 2 รีจีสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการใช้ Timer แสดงได้ดังตารางที่ 2.4 ซึ่งจะเห็นว่ามีรีจีสเตอร์บางตัวสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย นอกจากนี้ตัว Timer สามารถใช้เป็นตัวนับ (Counter) ได้อีกด้วย โดยการโปรแกรมในรีจีสเตอร์ TMOD

รีจีสเตอร์	หน้าที่	ตำแหน่ง	สามารถอ้างอิงตำแหน่งบิต
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TL0	Timer 0 Low-byte	8AH	No
TL1	Timer 1 Low-byte	8BH	No
TH0	Timer 0 High-byte	8CH	No
TH1	Timer 1 High-byte	8DH	No
T2CON*	Timer 2 Control	C8H	Yes
RCAP2L*	Timer 2 Low-byte Capture	CAH	No
RCAP2H*	Timer 2 High-byte Capture	CBH	No
TL2*	Timer 2 Low-byte	CCH	No
TH2*	Timer 2 High-byte	CDH	No

มีในเบอร์ 8032 / 8052

ตารางที่ 2.4 รีจีสเตอร์ที่ใช้เป็น Timer

Timer Mode Register (TMOD)

ตัวรีจีสเตอร์ TMOD เป็นรีจีสเตอร์ควบคุม Timer จะแบ่งออกเป็น 2 กลุ่ม กลุ่มละ 4 บิต โดย 4 บิตบนจะเป็นการควบคุม Timer 1 ส่วน 4 บิตล่างจะเป็นการควบคุม Timer 0 ความหมายของแต่ละบิตดูในตารางที่ 2.3 ซึ่งตัวรีจีสเตอร์นี้เป็นตัวเลือกการทำงานว่าจะให้ตัว Time/Counter ทำงานในโหมดใด และเป็น Timer หรือ Counter รีจีสเตอร์ TCON ไม่สามารถจะโปรแกรมเข้าไปในระดับบิตได้ (Not Bit-Addressable) ซึ่งการใช้งานมักจะโปรแกรมเข้าไปครั้งเดียวในตำแหน่งเริ่มต้นของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต	ชื่อ	Timer	ความหมาย
7	GATE	1	Gate bit ถ้าบิตนี้เซตวงจรจะทำงาน เมื่อ INT1 เป็น High
A	C/T	1	เป็นบิตเลือก Counter / Timer 1 = ใช้เป็น Counter 0 = ใช้เป็น Timer
5	M1	1	Mode bit 1
4	M0	1	Mode bit 0
3	GATE	0	บิต Gate ของ Timer 0
2	C/T	0	บิตเลือก Counter / Timer ของ Timer 0
1	M1	0	Timer 0 M1 bit
0	M0	0	Timer 0 M0 bit

ตารางที่ 2.5 รีจีสเตอร์ TMOD (Timer Mode)

M1	M0	Mode	ความหมาย
0	0	0	ใช้เป็น Timer แบบ 13-bit (8048 Mode)
0	1	1	ใช้เป็น Timer แบบ 16-bit
1	0	2	ใช้เป็น Timer แบบ 8-bit Auto-reload Mode
1	1	3	Split Timer Mode : แยก Timer 0 ออกเป็น Timer 8 บิตสองตัวคือ TL0 และ TH0 โดยไม่ใช้ Timer 1

ตารางที่ 2.6 การใช้ Timer โหมดต่าง ๆ

Timer Control Register (TCON)

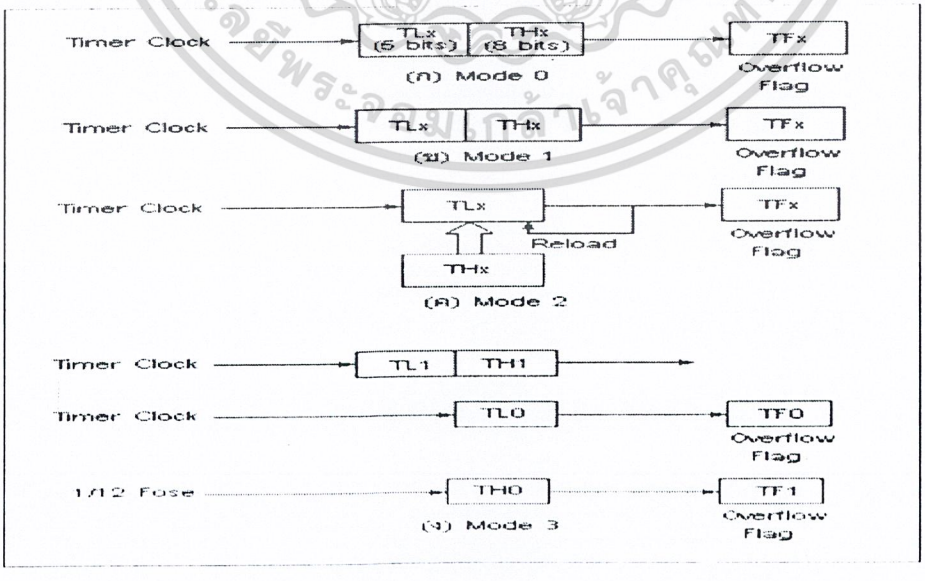
รีจีสเตอร์ TCON เป็นรีจีสเตอร์ที่บอกสถานะและควบคุมบิต Timer 0 และ Timer 1 ซึ่งดูได้จากตารางที่ 2.7 รีจีสเตอร์นี้สามารถเข้าถึงข้อมูลระดับบิตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บิต	ชื่อ	ตำแหน่งบิต	ความหมาย
TCON.7	TF1	8FH	บิตแฟลกแสดงการโอเวอร์โฟลว์ของ Timer 1 จะ Set โดย Hardware และ Clear โดย Software
TCON.6	TR1	8EH	บิตควบคุมการปิด-เปิด Timer 1 Set และ Clear โดย Software
TCON.5	TF0	8DH	แฟลกแสดงการโอเวอร์โฟลว์ของ Timer 0
TCON.4	TR0	8CH	บิตควบคุมการปิด-เปิด Timer 0
TCON.3	IE1	8BH	บิตแฟลกแสดงการอินเทอร์รัพท์จาก INT1 จะ Set โดย Hardware และสามารถ Clear ได้ด้วย Software
TCON.2	IT1	8AH	บิตเลือกชนิดของสัญญาณอินเทอร์รัพท์จากอินเทอร์รัพท์ภายนอก INT1 สามารถ Set และ Clear ได้ด้วย Software
TCON.1	IE0	89H	บิตแฟลกแสดงการอินเทอร์รัพท์จาก INT0
TCON.0	IT0	88H	บิตเลือกชนิดของสัญญาณอินเทอร์รัพท์จากอินเทอร์รัพท์ภายนอก INT0

ตารางที่ 2.7 แสดงความหมายแต่ละบิตของรีจีสเตอร์ TCON (Timer Control)



รูปที่ 2.10 การทำงานของ Timer ในโหมดต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50187

Timer Mode And Overflow Flag

เมื่อใช้ Timer 0 และ Timer 1 จะต้องใช้รีจีสเตอร์คู่ TLx และ THx โดยค่า x จะเป็นตัวบอกว่าเป็น Timer 0 หรือ Timer 1 การใช้ Timer สามารถใช้งานได้หลายโหมด ดังแสดงในรูปที่ 2.10 ซึ่งเราสามารถเซตค่าโหมดการทำงานได้ โดยการโปรแกรมในรีจีสเตอร์ TMOD

13-Bit Timer Mode (Mode 0)

การทำงานในโหมด 0 นี้จะเป็นการใช้ Timer แบบ 13 บิต ดังแสดงในรูป 2.9(ก) ซึ่งจะใช้ 5 บิตล่างของ TLx โดยไม่สนใจ 3 บิตที่เหลือ และ 8 บิต ของ THx การทำงานในโหมดนี้ เมื่อบิตของ TLx นับไปจนเป็น "1" ทุกบิตจะส่ง Clock 1 ลูกให้ หนึ่งลูกให้ THx นับต่อและเมื่อนับเป็น "1" ทุกบิต และเปลี่ยนกลับเป็น "0" จะเกิด Overflow Flag เกิดขึ้น

16-Bit Timer Mode (Mode 1)

การทำงานในโหมดนี้จะเหมือนกับการทำงานในโหมด 0 แต่เป็น Timer แบบ 16 บิต ซึ่งการนับจะเริ่มตั้งแต่ 0000H,0001H,0002H ไปเรื่อย ๆ และจะเกิด Overlflow ขึ้น เมื่อมีการเปลี่ยนจาก FFFFH เป็น 0000H ซึ่งเป็นการเซต Overflow Flag และค่านี้จะเกิดขึ้นในบิต TFx ของรีจีสเตอร์ TCON ซึ่งสามารถอ่านและเขียนด้วยโปรแกรม

การใช้ตัว Timer นี้ค่าของบิตสูงสุด (MSB) คือค่าบิต 7 ของ THx ส่วนบิตต่ำสุด (LSB) คือ บิต 0 ของ TLx บิต LSB จะเป็น Toggles เมื่อมีสัญญาณอินพุตเข้ามา ถูกหารด้วย 2 ดังนั้นจะพบว่าบิต MSB จะ Toggles ด้วยค่าความถี่ของสัญญาณอินพุตหารด้วย 65,536 (2^{16}) และค่า Timer รีจีสเตอร์นี้ (TLx/THx) สามารถอ่านและเขียนได้ด้วยการโปรแกรม ดังนั้นสามารถนำไปประยุกต์ใช้งานได้ตามต้องการ

8-Bit Auto – Reload Mode (Mode 2)

การทำงานในโหมด 2 เรียกอีกอย่างหนึ่งว่า 8-bit Auto – reload Mode โดยใช้ Timer ไบต์ต่ำ (TLx) เป็น Timer แบบ 8 บิต เมื่อไบต์ต่ำเกิด Overflows หรือเกิดการเปลี่ยนแปลงจาก FFH เป็น 00H จะมีการโหลดค่าที่เก็บไว้ในไบต์สูง (THx) ไปเก็บไว้ในไบต์ต่ำ (TLx) ซึ่งจะเป็นค่าเริ่มต้นของการนับครั้งต่อไป นิยมใช้สร้างเป็นฐานเวลาที่สามารถโปรแกรมได้

Split Timer Mode (Mode 3)

การทำงานในโหมด 3 นี้ ตัว Timer 1 จะไม่ทำงาน ตัว Timer 0 จะแยกเป็น 2 ตัว ตัวละ 8 บิต คือ TL0 และ TH0 เมื่อ Timer เกิด Overflows จะมีการเซตบิต TF0 และ TF1 การทำงานในโหมด 3 นี้ Timer 1 จะไม่ถูกใช้งานแต่เราสามารถสวิตช์ให้ Timer 1 ไปทำงานในโหมดอื่นได้ แต่การทำงานของ Timer 1 จะไม่มีการอินเทอร์รัพท์เกิดขึ้น เพราะบิต TF1 ถูกใช้ในการนับของ TH0 ในการทำงานของโหมด 3 ไปแล้ว เราอาจมองว่าถ้าให้ Timer ทำงานในโหมด 3 ทำให้เรามี Timer เพิ่มขึ้น คือ TH0 และ TL0 ใน Timer 0 โหมด 3 และโปรแกรมให้ Timer 1 ไปทำงานในโหมดอื่น ๆ

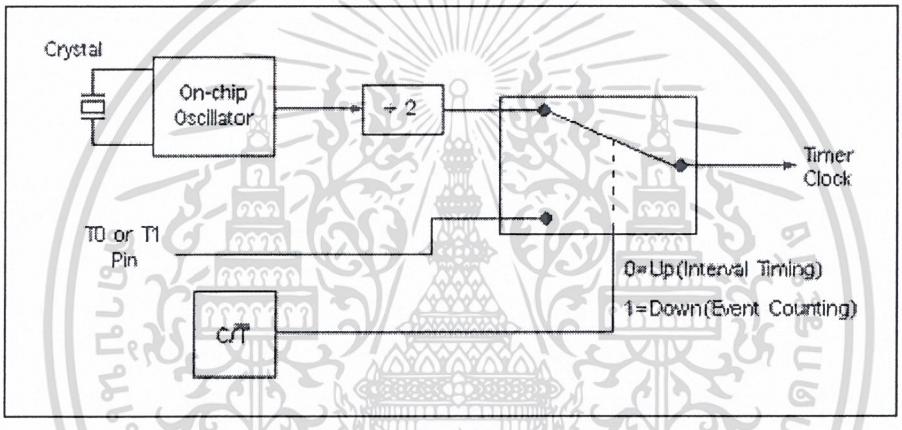
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Clocking Source

การใช้ Timer สามารถใช้ได้ 2 หน้าที่ คือเป็นตัวจับเวลา (Timer) และเป็นตัวนับ (Counter) ซึ่งสามารถโปรแกรมได้โดยการเซตหรือรีเซตบิต C / Tในรีจีสเตอร์ TMOD

การใช้เป็นตัวจับเวลา (Timer)

ถ้าบิต C / Tใน TMOD เป็นลอจิก "0" จะเป็นการเลือกให้ Timer นำ Clock มาจากวงจร Oscillator ในชิพ ซึ่งสัญญาณนาฬิกาจะเข้ามาทุก ๆ Machine Cycle หรืออาจกล่าวได้ว่าค่าใน THx และ TLx จะมีค่าเพิ่มขึ้นด้วยอัตราการนับแต่ละครั้งใช้เวลาเท่ากับ 1/12 ของความถี่ของสัญญาณนาฬิกาที่ใช้บนชิพ ถ้า MCS – 51 ใช้สัญญาณนาฬิกา 12 MHz การนับจะมีความถี่เท่ากับ 1 MHz



รูปที่ 2.11 ความถี่ของสัญญาณนาฬิกาที่เข้าขา Timer

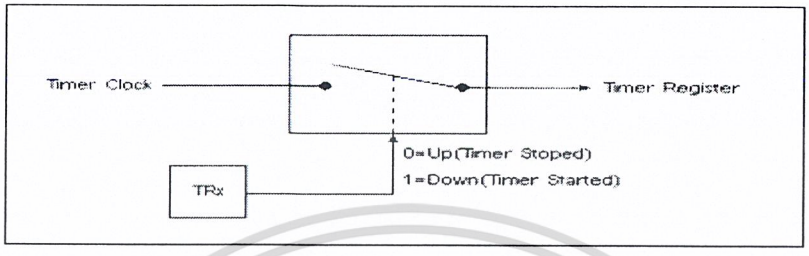
การใช้เป็นตัวนับ (Counter)

ถ้าบิต C / T เป็น "1" ตัว Timer จะนำ Clock มาจากภายนอกโดยใช้ขา P3.4 หรือ T0 เป็นขา Input Clock ให้กับ Timer 0 และใช้ขา P3.5 หรือ T1 เป็น Input Clock ให้กับ Timer 1 หรืออาจมองว่า ถ้าจะให้นับอะไรสัญญาณที่จะนับให้ต่อกับขา T0 และ T1 ในการใช้เป็น Counter สัญญาณที่เข้ามามีการเปลี่ยนแปลงจาก "1" เป็น "0" จะทำให้วงจรนับ TLx มีค่าเพิ่มขึ้น 1 ภายใน MCS – 51 นี้จะตรวจสอบขาอินพุด T0 และ T1 ในช่วงเวลาเฟส 2 ของ State 5 (S5P2) ถ้าพบว่ามีค่าเป็น "1" ต่อมาในอีกหนึ่ง Machine Cycle ที่เฟส 2 ของ State 5 (S5P2) ลอจิกอินพุดเปลี่ยนเป็น "0" จะทำให้ค่าใน Timer เพิ่มขึ้น 1 ดังนั้น จะเห็นได้ว่าการนับ 1 ครั้งจะต้องใช้เวลา 2 Machine Cycles ดังนั้นความถี่สูงสุดที่จะให้ Timer ทำงานเป็น Counter นับได้ จะมีค่ามากที่สุด 500 kHz ถ้า MCS – 51 ทำงานที่ความถี่สัญญาณนาฬิกา 12 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก ารเริ่ม , หยุด และการควบคุม Timer

วิธีเริ่มและหยุดตัว Timers สามารถควบคุมได้ที่บิต TRx ในรีจีสเตอร์ TCON โดยปกติแล้ว TRx จะเคลียร์หลังจากที่ระบบถูกรีเซต ซึ่งจะเป็นการให้ Timer ไม่นับและ TRx นี้จะเซตได้จากชุดคำสั่ง หรือการโปรแกรม พิจารณารูปที่ 2.12



รูปที่ 2.12 การใช้บิตควบคุม TR

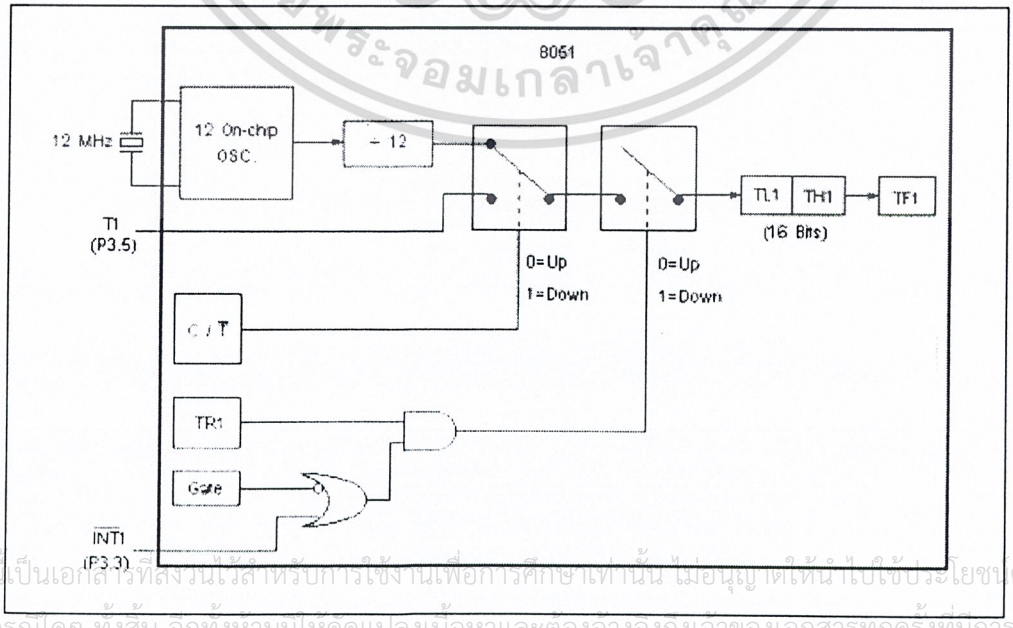
ตัวบิต TRx จะเป็นส่วนที่สามารถเข้าถึงข้อมูลในระดับบิตได้ (Bit Addressable)ในรีจีสเตอร์ TCON ถ้าจะให้ TIMER 0 เริ่มทำงานจะเขียนคำสั่งได้ดังนี้

SETB TR0

ถ้าจะหยุดทำงานเขียนคำสั่งได้ดังนี้

CLR TR0

ในการเขียนโปรแกรมภาษาแอสแซมบลี สามารถใช้สัญลักษณ์ TR0 ในคำสั่ง SETB TR0 เลยได้ เพราะตัวแอสแซมบลีจะตีความ TR0 เป็น Bit Address ตำแหน่ง 8CH



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.13 ระบบทั้งหมดของ Timer 1

วิธีควบคุม Timer สามารถควบคุมได้ที่บิต GATE ใน TMOD และขาอินเตอร์รัพท์จากภายนอก INTx ถ้า INT0 เป็นลอจิก "0" และโปรแกรมให้ Timer 0 ทำงานในโหมด 2 เมื่อ TL0/TH0 = 0000H, GATE = 1 และ TR0 = 1 เมื่อ INT0 ขึ้นเป็นลอจิก "1" ตัว Timer จะ "Gate On" และจะให้สัญญาณนาฬิกาความถี่ 1 MHz เมื่อ INT0 ลงเป็น "0" ตัว Timer " Gate Off " สัญญาณที่ได้จะมีความกว้างของสัญญาณนาฬิกา 1 μs ส่งเข้าไปใน TL0/TH0

รูปที่ 2.13 จะเป็นระบบที่สมบูรณ์ของ Timer 1 เมื่อทำงานในโหมด 1 ซึ่งเป็น 16-bit Timer โดยใช้รีจีสเตอร์ TL1 / TH1 และ Overflow Flag TF1 ในรูปจะเห็นถึงการควบคุมแหล่งกำเนิด Clock การเริ่มทำงาน และการหยุดทำงาน

Intializing And Accessing Timer Register

การใช้งาน Timer เริ่มแรกจะต้องโปรแกรมเพื่อเลือกโหมดการทำงานของ Timer ก่อนเมื่อเริ่มใช้งานก็โปรแกรมให้ เริ่มทำงาน, หยุดทำงาน, อ่าน และ เคลียร์ค่า Flag Bits อ่านค่า Timer Registers ตามลำดับ เพื่อนำไปประยุกต์การใช้งานต่อไป

TMOD คือ รีจีสเตอร์ที่ต้องโปรแกรม โดยเซตโหมดการทำงานก่อน ตัวอย่างเช่น ถ้าให้ Timer 1 เป็น 16-bits Timer (โหมด 1) นับสัญญาณนาฬิกาบนชิพ สามารถเขียนคำสั่งได้ดังนี้

```
MOV TMOD, #00010000B
```

ผลที่ได้จากคำสั่งข้างบนคือ เซตบิต M1 = 0 และ M0 = 1 ซึ่งเป็นการเลือกโหมด 1 และให้ C / T= 0 และ GATE = 0 ซึ่งเป็นการใช้สัญญาณนาฬิกาจากภายในหรือใช้เป็น Timer และตัว Timer นี้จะยังไม่ทำงาน ถ้าบิตควบคุม TR1 ยังไม่ได้เซต

ถ้าให้ Timer นี้นับขึ้นโดยใช้รีจีสเตอร์ TL1 / TH1 และจะเซตบิต Overflow Flag เมื่อรีจีสเตอร์เปลี่ยนจาก FFFFH เป็น 0000H โดยให้นับเวลาไป 100 μs หรือให้ TL1 / TH1 นับสัญญาณนาฬิกาได้ 100 ลูก ดังนั้นค่าเริ่มต้นของ TL1 / TH1 จะไม่เริ่มที่ 0000H จะต้องเริ่มที่ FFFFH ลบด้วย 100 ลูก หรือ FF9CH เพื่อให้นับไปถึง FFFFH และเปลี่ยนเป็น 0000H ได้สัญญาณนาฬิกา 100 ลูกพอดี สามารถเขียนคำสั่งได้ดังนี้

```
MOV TL1 , # 9CH
MOV TH1 , # 0FFH
```

ถ้าให้ Timer เริ่มทำงานก็ให้บิตควบคุมดังนี้

```
SETB TR1
```

จากนั้นบิต Overflow Flag จะส่งออกมาหลังเวลาผ่านไป 100 μs ซึ่งเราสามารถเขียนโปรแกรมเป็นโปรแกรมวนลูป 100 μs ได้ โดยตรวจสอบบิต TF1 ว่าถูกเซตหรือไม่ ถ้าไม่เซตก็ให้วนลูปต่อไปดังนี้

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLR TR1
CLR TF1
```

การใช้แบบ Readind a Timer "On the Fly"

การใช้งานแบบประยุกต์บางงานจะต้องอ่านค่าจาก Timer Register เนื่องจากตัว Timer Register มีขนาด 2 ไบต์ ถ้าหากไบต์ต่ำเกิด Overflow จะทดเข้าไบต์สูง ถ้าหากเขียนโปรแกรมให้อ่านค่าจากไบต์ต่ำก่อน แล้วจึงอ่านไบต์สูงข้อมูลที่ได้อาจเกิดข้อผิดพลาดได้เนื่องจากไบต์ต่ำมีการเปลี่ยนแปลงเร็วกว่าไบต์สูง การอ่านข้อมูลควรอ่านจากไบต์สูงก่อน แล้วจึงกลับมาอ่านไบต์ต่ำ จากนั้นอ่านข้อมูลไบต์สูงอีกครั้ง ถ้าค่าไบต์สูงที่อ่านได้ไม่มีการเปลี่ยนแปลงให้ใช้ค่านั้นได้เลย แต่ถ้ามีการเปลี่ยนแปลงให้อ่านอีกครั้ง ถ้าต้องการอ่านข้อมูลจาก TL1 / TH1 เข้าในรีจีสเตอร์ R6 / R7 อาจเขียนโปรแกรมได้ดังนี้

```
AGAIN :  MOV A, TH1
         MOV R6, TL1
         CJNE A, TH1, AGAIN
         MOV R7, A
```

Short Intervals And Long Intervals

ถ้า MCS – 51 ทำงานที่ความถี่สัญญาณนาฬิกา 12 MHz ถ้าให้ Timers ใช้วงจร Oscillator บนชิพ สัญญาณนาฬิกาจะถูกหารด้วย 12 และ Timer จะทำงานด้วยความถี่ 1 MHz ถ้าต้องการใช้โปรแกรมสร้างสัญญาณนาฬิกาออกมาอาจทำได้โดยง่าย ซึ่งพิจารณาจากการทำงานชุดคำสั่งต่าง ๆ ของ MCS – 51 ใน 1 Machine Cycle จะใช้เวลา 1μS ในตารางที่ 2.8 จะแสดงความกว้างของสัญญาณที่สร้างขึ้นจาก MCS – 51 ที่ทำงานด้วย Crystal ความถี่ 12 MHz

Maximum Interval in Microseconps	**Technique**
≈10	Software Tuing
256	8 – bit Timer with Auto-reload
65536	8 – bit Timer
No Limit	16 – bit Timer Plus Software Loops

ตารางที่ 2.8 ค่าสูงสุดของการใช้ Timer โหมดต่าง ๆ

การอินเตอร์รัพท์

การทำงานของระบบคอมพิวเตอร์ โดยทั่วไปมักมีอุปกรณ์ภายนอกต่อร่วมอยู่ ถ้าคอมพิวเตอร์ต้องการทำงานกับอุปกรณ์ภายนอกจะต้องคอยตรวจสอบอุปกรณ์เหล่านั้นเสมอ ตัวอย่างเช่น ถ้าหากให้คอมพิวเตอร์พอร์ทหนึ่งต่ออยู่กับหลอด LED 7 ส่วน อีกพอร์ทหนึ่งต่อกับสวิตช์ ถ้าระบบของเราทำงานเป็นนาฬิกาเดินไปให้คอยตรวจสอบสวิตช์ด้วยว่ามีการกดหรือยัง การ

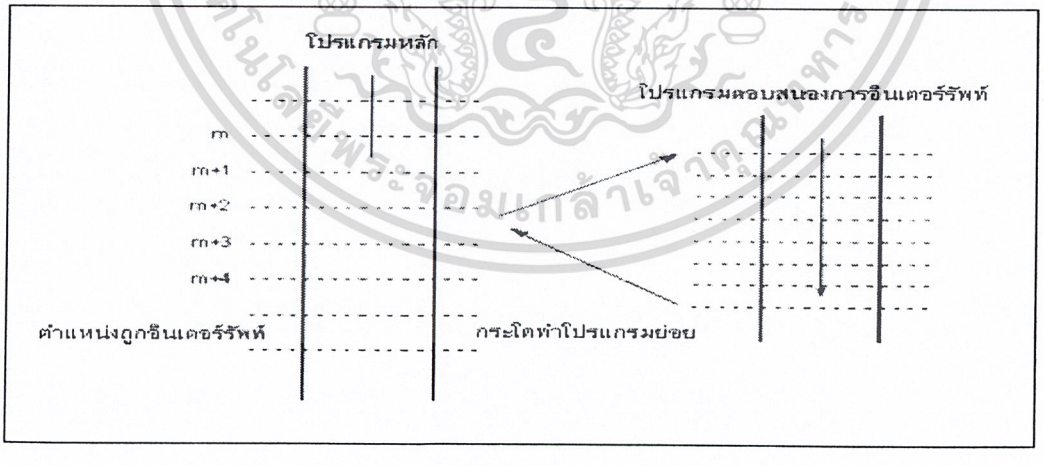
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆ ทั้งสิ้น

ทำงานแบบนี้เรียกว่า Polling Method คือตัวไมโครโปรเซสเซอร์จะต้องคอยตรวจสอบอุปกรณ์อินพุตตลอดเวลาว่ามีข้อมูลเข้ามาหรือยัง การทำงานแบบนี้ ถ้ามีอุปกรณ์ภายนอกหลายตัวระบบต้องตรวจสอบอุปกรณ์ภายนอกหลายตัว ทำให้เสียเวลาในการทำงานหลักไป การทำงานอีกแบบหนึ่งจะให้ CPU ทำงานหลัก ถ้ามีการกดสวิตช์เมื่อไรให้นาฬิกาหยุดเดินทันที การทำงานในลักษณะนี้ CPU ไม่ต้องเสียเวลาในการตรวจอุปกรณ์ภายนอก ถ้าอุปกรณ์ภายนอกต้องการติดต่อกับ CPU อุปกรณ์ภายนอกจะส่งสัญญาณมาบอก CPUเอง ระบบนี้เรียกว่า การอินเทอร์รัพท์ (Interrupt)

ขบวนการเกิดอินเตอร์รัพท์

ถ้าหากคอมพิวเตอร์กำลังทำงานโปรแกรมหลักอยู่เมื่อมีการอินเทอร์รัพท์เข้ามา คอมพิวเตอร์จะละทิ้งโปรแกรมหลัก แต่ไปทำงานโปรแกรมตอบสนองการอินเตอร์รัพท์ (Interrupt Service Routine) เมื่อทำโปรแกรมตอบสนองอินเตอร์รัพท์เสร็จ คอมพิวเตอร์จะกลับมาทำโปรแกรมเดิม

ถ้า CPU กำลังทำงานโปรแกรมหลักอยู่ เช่นกำลังทำคำสั่งในตำแหน่งของหน่วยความจำที่ m, m+1, m+2 ไปเรื่อย ๆ โดย PC จะชี้ที่ตำแหน่งที่จะอ่านค่าคำสั่งถัดมา เมื่อโปรแกรมทำงานมาถึงตำแหน่งที่ m+3 แล้วเกิดการอินเตอร์รัพท์ขึ้น (ขณะนั้น PC อยู่ที่ m+4) โปรแกรมจะต้องทำงานโปรแกรมตอบสนองการอินเตอร์รัพท์ โดยย้าย PC ไปที่ตำแหน่งที่เก็บโปรแกรมตอบสนองการอินเตอร์รัพท์ จากนั้นจะเก็บค่า PC เดิมลงในหน่วยความจำสแตค เมื่อคอมพิวเตอร์ทำงานโปรแกรมตอบสนองการอินเตอร์รัพท์เสร็จสิ้นลง จะคืนค่าใน สแตค (m+4) ให้กับ PC ทำโปรแกรมหลักต่อ



รูปที่ 2.14 ขั้นตอนการทำงานของโปรแกรมเมื่อถูกอินเตอร์รัพท์

สัญญาณอินเตอร์รัพท์

แหล่งกำเนิดสัญญาณอินเตอร์รัพท์ที่ใช้กับ MCS – 51 มีสองชนิดคือ อินเตอร์รัพท์ภายในและภายนอก โดยอินเทอร์รัพท์ภายในจะเกิดขึ้นจากภายในตัว MCS – 51 เอง ได้แก่สัญญาณจากไทเมอร์แฟลก 0 (TF0) ไทเมอร์แฟลก 1 (TF1) และพอร์ทอนุกรม สำหรับอินเตอร์รัพท์ภายนอกเกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสัญญาณที่กระตุ้นเข้ามาทางขา INT0 และ INT1 เมื่อมีสัญญาณอินเตอร์รัพท์จากแหล่งต่าง ๆ เข้ามา เราสามารถโปรแกรมได้ว่าจะให้ MCS – 51 ยอมให้มีการอินเตอร์รัพท์ได้หรือไม่ โดยการโปรแกรมไปที่ รีจีสเตอร์ IE (Interrupt Enable) และถ้ามีสัญญาณอินเตอร์รัพท์มากจากแหล่งต่าง ๆ หลายแหล่งพร้อมกันเราสามารถจัดลำดับได้ว่า จะให้อินเตอร์รัพท์ใดเกิดก่อน โดยการโปรแกรมไปที่ อินเตอร์รัพท์ไพออริตี้ IP (Interrupt Priority) รีจีสเตอร์ทั้งสองตัวมีรายละเอียดดังนี้

Interrupt Enables

เป็นรีจีสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ ใช้สำหรับกำหนดค่าว่าถ้าเกิดการอินเตอร์รัพท์จากแหล่งต่าง ๆ จะทำอินเตอร์รัพท์เหล่านั้นหรือไม่ โดยรายละเอียดของบิตต่าง ๆ มีดังตารางที่ 2.8

<u>บิต</u>	ชื่อบิต	ตำแหน่งบิต	รายละเอียด
IE.7	EA	AFH	ถ้าเซตยอมให้มีการอินเตอร์รัพท์
IE.6	-	AEH	ไม่ใช้งาน
IE.5	ET2	ADH	Enable อินเตอร์รัพท์จาก Timer 2 (ใช้กับ 8052)
IE.4	ES	ACH	Enable อินเตอร์รัพท์จากพอร์ทอนุกรม
IE.3	ET1	ABH	Enable อินเตอร์รัพท์จาก Timer 1
IE.2	EX1	AAH	Enable อินเตอร์รัพท์จาก INT1
IE.1	ET0	A9H	Enable อินเตอร์รัพท์จาก Timer 0
IE.0	EX0	A8H	Enable อินเตอร์รัพท์จาก INT0

ตารางที่ 2.9 บิตต่าง ๆ ของรีจีสเตอร์ IE

Interrupt Priority

เป็นรีจีสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ใช้ในการจัดลำดับความสำคัญของการอินเทอร์รัพท์ซึ่งสามารถจัดได้สองลำดับ ถ้าเป็น "1" หมายความว่ามีความสำคัญสูงสุด ถ้าเป็น "0" หมายความว่ามีลำดับความสำคัญต่ำสุด ถ้าหากกำหนดให้มีความสำคัญเป็น "1" เหมือนกันหมด MCS – 51 จะจัดลำดับความสำคัญใหม่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	อินเตอร์รัพท์
1 (สูงสุด)	IE0
2	TF0
3	IE1
4	TF1
5 (ต่ำสุด)	Serial Port

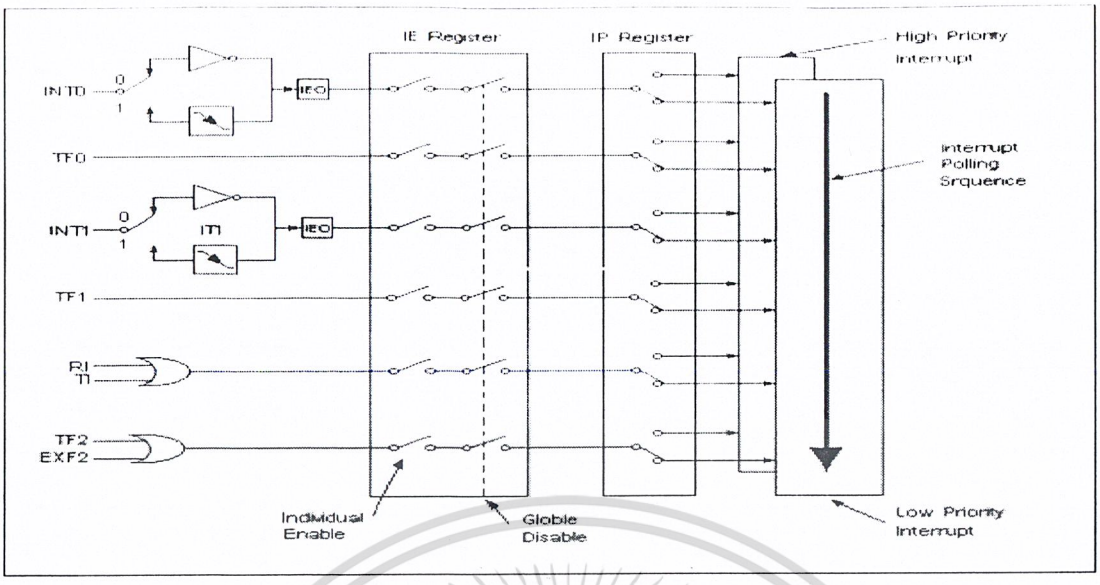
ตารางที่ 2.10 รีจิสเตอร์ต่าง ๆ ที่เกี่ยวข้องกับการอินเตอร์รัพท์

บิต	ชื่อบิต	ตำแหน่งบิต	รายละเอียด
IP.7	-	-	ไม่ใช้งาน
IP.6	-	-	ไม่ใช้งาน
IP.5	PT2	0BDH	ใช้กับ Timer 2 (8052)
IP.4	PS	0BCH	ใช้กับพอร์ทอนุกรม
IP.3	PT1	0BBH	ใช้กับ Timer 1
IP.2	PX1	0BAH	ใช้กับอินเทอร์รัพท์จาก INT1
IP.1	PT0	0B9H	ใช้กับ Timer 0
IP.0	PX0	0B8H	ใช้กับอินเทอร์รัพท์จาก INT0

ตารางที่ 2.11 บิตและหน้าที่ต่าง ๆ ของรีจีสเตอร์ IP

จากรูปที่ 2.15 แสดงการอินเตอร์รัพท์จากแหล่งต่าง ๆ ที่มีผลกับ MCS – 51 ถ้าเป็นเบอร์ 8051 8031 จะถูกอินเตอร์รัพท์ได้ 5 แหล่ง ถ้าเป็นเบอร์ 8052,8032 จะถูกอินเตอร์รัพท์ได้ 6 แหล่ง โดยเพิ่มอินเตอร์รัพท์จาก Timer 2 ในรูปที่ 2.13 จะแสดงให้เห็นว่า ถ้า MCS – 51 จะถูกอินเตอร์รัพท์ได้จะต้องเซตค่า Global Enable ในรีจีสเตอร์ IE นอกจากนี้ยังกำหนดได้ว่าจะให้อินเตอร์รัพท์ใดเกิดได้ โดยการเซตค่า Interrupt Enable ของอินเตอร์รัพท์จากแหล่งต่าง ๆ ในรีจีสเตอร์ IE จากรูปยังแสดงให้เห็นอีกว่าเมื่อมีการอินเทอร์รัพท์เข้ามาจะมีผลต่อแฟลกใด เช่นถ้า INT0 เป็น "1" บิต IE0 จะเป็น "1" หมายความว่าถูกอินเตอร์รัพท์ โดยแฟลกต่าง ๆ ที่มีผลจากการถูกอินเตอร์รัพท์แสดงได้ดังตารางที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงการอินเตอร์รัพท์

อินเตอร์รัพท์	แฟลก	ประกอบอยู่ในรีจีสเตอร์
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial port	T1	SCON.1
Serial port	RI	SCON.0
Timer 2	TF2	T2CON.7 (8052)
Timer 2	EXF2	T2CON.6 (8052)

ตารางที่ 2.12 แฟลกที่จะทำงานเมื่อถูกอินเตอร์รัพท์

จากตารางจะเห็นว่า ถ้ามีการอินเตอร์รัพท์จากภายนอกเข้ามา ตัวที่จะอินเตอร์รัพท์ MCS – 51 คือ บิตแฟลก IE0 ซึ่งอยู่ในรีจีสเตอร์ TCON ถ้ามีการสื่อสารแบบอนุกรม เมื่อข้อมูลถูกส่งไปหมดแล้วจะอินเตอร์รัพท์ MCS – 51 ทางบิตแฟลก TI ถ้ารับข้อมูลหมดแล้วจะอินเตอร์รัพท์ MCS – 51 ทางบิตแฟลก RI ซึ่งอยู่ในรีจีสเตอร์ SCON และถ้าใช้ Timer 0 ในการนับเมื่อเกิด Overflow สามารถอินเตอร์รัพท์ MCS – 51 ได้ทางบิต TF0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การทำงานของระบบหลังถูกอินเตอร์รัพท์

เมื่อ MCS 51 ถูกอินเทอร์รัพท์จะต้องกระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัพท์โดยตำแหน่งที่จะกระโดดไปเรียกว่า อินเตอร์รัพท์เวกเตอร์ (Interrupt Vectors) เมื่อทำโปรแกรมตอบสนองการอินเตอร์รัพท์เรียบร้อยแล้ว MCS – 51 จะกระโดดมาทำงานยังตำแหน่งเดิม โดยก่อนที่จะกระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัพท์จะต้องเก็บค่าตำแหน่งเดิมไว้ โดยเก็บค่า PC ลงหน่วยความจำสแตคซึ่งอยู่ที่หน่วยความจำที่ถูกชี้โดยรีจีสเตอร์ SP เมื่อทำโปรแกรมตอบสนองการอินเทอร์รัพท์เสร็จแล้วจะคืนค่าในหน่วยความจำสแตคให้ PC ตามเดิม ค่าอินเตอร์รัพท์เวกเตอร์ของ MCS – 51 แสดงได้ดังตารางที่ 2.13

อินเตอร์รัพท์	อินเตอร์รัพท์เวกเตอร์
System Reset	0000H
External 0	0003H
Timer 0	000BH
External 1	0013H
Timer 1	001BH
Serial Port	0023H
Timer 2	002BH

ตารางที่ 2.13 อินเตอร์รัพท์เวกเตอร์ของอินเตอร์รัพท์ต่าง ๆ

จากตารางจะเห็นว่าถ้าระบบถูกอินเตอร์รัพท์จากภายนอกทาง INT0 ตัว MCS – 51 จะกระโดดไปทำงานที่ตำแหน่ง 0003H ถ้าระบบถูกอินเตอร์รัพท์จาก Timer 0 จะกระโดดไปทำงานตำแหน่ง 000BH

2.2.3 การใช้งานพอร์ตสื่อสารข้อมูลอนุกรมของ 8051

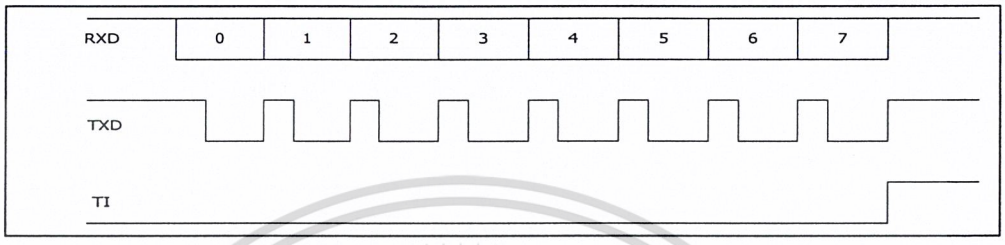
พอร์ทการสื่อสารมีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์(Full duplex)ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านส่งใช้ขา TxD (พอร์ท 3.1) ทางด้านรับใช้ขา RxD (พอร์ท 3.0) SBUF ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม

พอร์ทสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่บิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมด ของพอร์ทสื่อสารอนุกรม มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 0 : พอร์ทสื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อนทาง

ขา RxD และไม่มีการส่ง Start bit แต่จะส่ง Shift Clock ทางขา TxD (ความเร็ว1/12 เท่าของ CPU Clock)



รูป2.16 แสดงสัญญาณของขา RXD , TXD

โหมด 1: พอร์ทสื่อสารอนุกรม 10 บิตข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลง (ความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟร์ของ Timer 1)

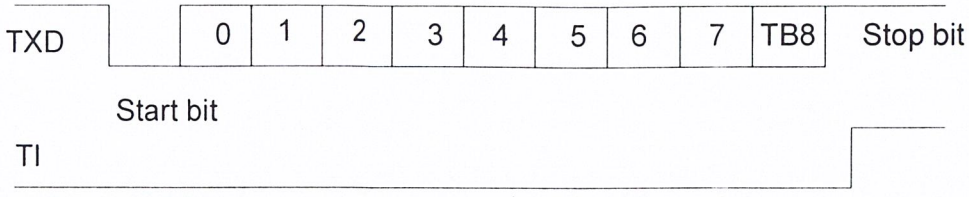
Baud Rate Mode 1,3 = $\frac{2^{SMOD} \times Oscillator..Frequrency}{32 \times 12 \times [256 - (TH1)]}$ โดยใช้ Timer 1

Baud Rate Mode 1,3 = $\frac{Oscillator..Frequrency}{32 \times [65536 - (RCAP2H, RCAP2L)]}$ โดยใช้ Timer 2

รูปที่ 2.17 แสดง Baudrate ใน Mode 1

โหมด 2 : พอร์ทสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 Start bit และ 1 Stop bit(TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU Clock โดยขึ้นกับบิต SMOD ใน PCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- Baud Rate Mode2 = (1/32)(Osc Freq) เมื่อ SMOD = 1
- Baud Rate Mode2 = (1/64)(Osc Freq.) เมื่อ SMOD = 0

รูปที่ 2.18 แสดง Baudrate ใน Mode 2

โหมด 3: พอร์ทสื่อสารแบบ 11 บิต URAT โดยส่งข้อมูล 9 บิต 1 Start bit และ 1 Stop bit เหมือนโหมด 2 ยกเว้นอัตราโอเวอร์โฟร์ของ Timer1,2 สำหรับ 8051 หรืออัตราโอเวอร์โฟร์ของ Timer2(สำหรับ 80C154D)

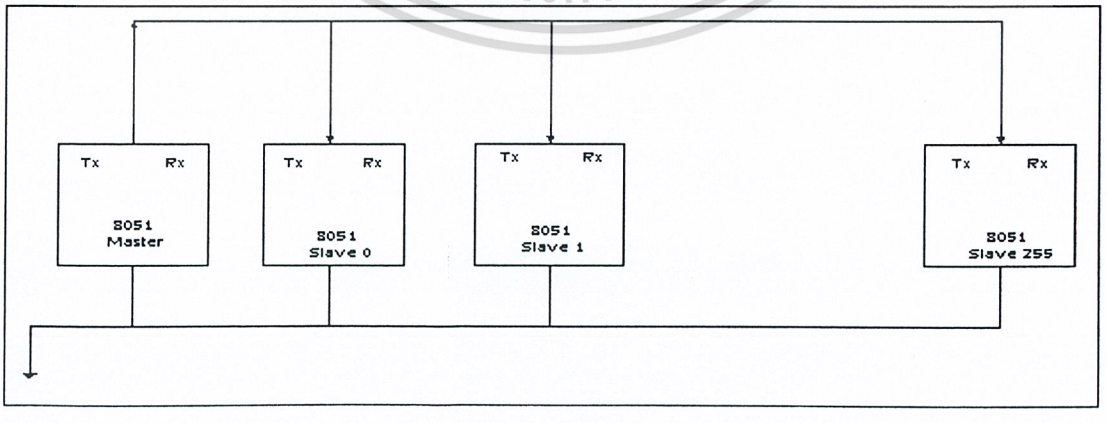
Baud Rate Mode 3 = $\frac{2^{SMOD} \times Oscillator.Freq.}{32 \times 12 \times [256 - (TH1)]}$ โดยใช้ Timer 1

Baud Rate Mode 3 = $\frac{Oscillator.Freq.}{32 \times [65536 - (RCAP2, RCAP2L)]}$ โดยใช้ Timer 2

2.2.4 การสื่อสารระหว่างไมโครคอนโทรลเลอร์

การสื่อสารข้อมูลแบบอนุกรมโดยใช้หลักการของมัลติโปรเซสเซอร์

ในการติดต่อสื่อสารข้อมูลอนุกรมโดยใช้หลักการไมโครโปรเซสเซอร์ จำเป็นที่จะกำหนดค่าแอดเดรสของตัว slave ออกไป(slave มีทั้งหมด 256) แล้วตามด้วย ดาต้าไบต์ เป็นไบต์ที่ 2ซึ่งมีข้อมูลส่งไปยัง slave การต่อสัญญาณจากตัว Master ไปยัง slave ทั้ง 256 ตัวดังแสดงในรูปที่ 2.19



รูปที่ 2.19 การสื่อสารระบบ Multiprocessor Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการส่งสัญญาณที่ส่งออกจากตัว Master

MCS-51 กำหนดรูปแบบการรับส่งข้อมูลได้ 2 แบบคือ single Processor Mode โดยเซ็ทSM2 = 0 และ Multiprocessor Mode โดยเซ็ท SM2 = 1 การส่งข้อมูลจะต้องเป็น (โหมด 2,3) และโหมด 3 เท่านั้นเพราะการส่งข้อมูลได้ 11 บิต โดยบิตแรกคือ Start bit อีก 8 บิต ข้อมูล ตามด้วยบิตที่ 9 ซึ่งใช้เป็นตัวบอกว่าไบต์ที่ส่งไปนี้เป็นแอดเดรสไบต์ หรือ ดาต้าไบต์ ถ้าโปรแกรม Multiprocessor Mode แล้ว

ทางด้านส่ง บิตที่ 9 (TB8) = 1 หมายถึงเลือกส่ง แอดเดรสไบต์

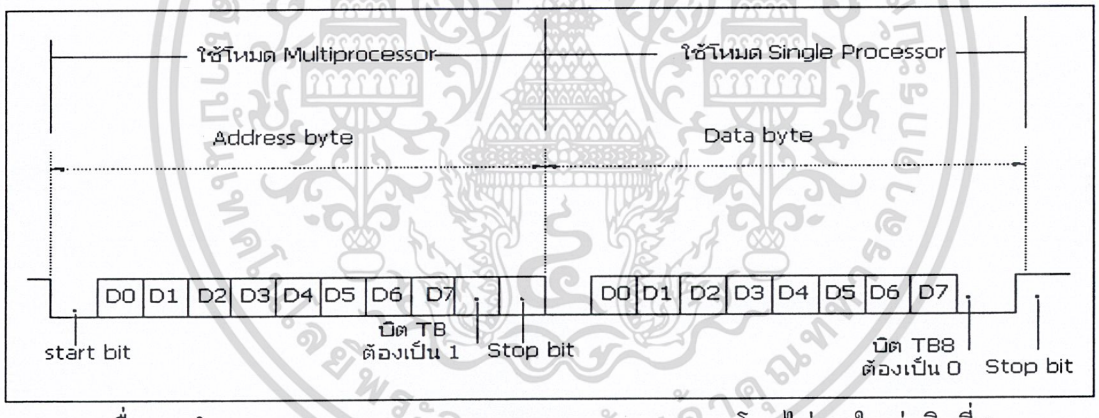
บิตที่ 9 (TB8) = 0 หมายถึงเลือกส่ง ดาต้าไบต์

ทางด้านรับ บิตที่ 9 (TB8) = 1 หมายถึงค่าที่รับมาคือ แอดเดรสไบต์

บิตที่ 9 (TB8) = 0 หมายถึงค่าที่รับมาคือ ดาต้าไบต์

การโปรแกรมบิตที่ 9 เป็น 1 หรือ 0 เราจะต้องโปรแกรมที่บิต TB8 ใน SCON

เมื่อการทำงานแบบ Multiprocessor Mode RI จะ SET เมื่อค่าบิตที่ 9 ที่รับมามีค่าเป็น 1



เมื่อการทำงานแบบ Single Processor Mode RI จะ SET โดยไม่สนใจ ค่าบิตที่ 9

รูปที่ 2.20 สัญญาณที่ทำงานแบบ Single Processor และ MultiProcessor

หมายเหตุ : บิต TB8 ทางด้านส่งจะเข้าไปเก็บใน RB8 ทางด้านรับ บิตนี้อยู่ใน SCON การส่งข้อมูล โหมดนี้จะต้องส่งแอดเดรสไบต์ ออกไปก่อน บิตที่ 9 (TB8) ต้องเป็น 1 หลังจากนั้นให้ เคลียร์ TB8 เพื่อส่งดาต้าไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบเครือข่าย(Network System)

2.3.1 ระบบเครือข่าย(Network System)

การพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายจำเป็นต้องมีความรู้พื้นฐานทางด้านระบบเครือข่ายพอสมควร ในหัวข้อนี้จะอธิบายหลักการของระบบเครือข่ายที่จำเป็นสำหรับการทำความเข้าใจในการพัฒนาโปรแกรมประยุกต์บนระบบเครือข่ายซึ่งจะกล่าวดังต่อไป

ความหมายของระบบเครือข่ายคอมพิวเตอร์และอินเตอร์เน็ตเวิร์คกิ้ง(Internetworking)

ระบบเครื่องข่ายคอมพิวเตอร์(Networking) คือระบบการเชื่อมต่อระหว่างระบบปลายทาง (End System) ซึ่งระบบปลายทางเป็นระบบที่อิสระจากกัน (Autonomous) ระบบปลายทางสามารถเป็นไปได้ตั้งแต่ไมโครคอมพิวเตอร์(Microcomputer) ไปจนกระทั่งซูเปอร์คอมพิวเตอร์(Super computer) ขนาดใหญ่ เพื่อจุดมุ่งหมายในการแลกเปลี่ยนข้อมูลและการแบ่งข้อมูลในการแบ่งปันทรัพยากรของระบบ เช่น ไฟล์ข้อมูล, เครื่องพิมพ์(Printer) โมเด็ม (Modem) ตลอดจนการให้บริการฐานข้อมูลร่วม(Sharing Database)

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

2.3.2 สถาปัตยกรรมของอินเตอร์เน็ต

คำกล่าวถึงอินเตอร์เน็ตว่าเป็นเครือข่ายที่ใหญ่ที่สุดในโลก เครือข่ายที่มีการต่อเสมือนกับใยแมงมุมครอบคลุมโลก อินเตอร์เน็ตเชื่อมโยงโลกเข้าด้วยกันอย่างไร้มิติ จากคำกล่าวเหล่านี้ทำให้เราควรรู้ว่า อินเทอร์เน็ตจัดการส่วนประกอบต่างๆ อย่างไร จึงทำให้แต่ละส่วนสามารถรับส่งข้อมูลและทำงานสัมพันธ์กันได้เป็นอย่างดี เราจึงจำเป็นต้องศึกษาสถาปัตยกรรมของอินเทอร์เน็ตดังต่อไปนี้

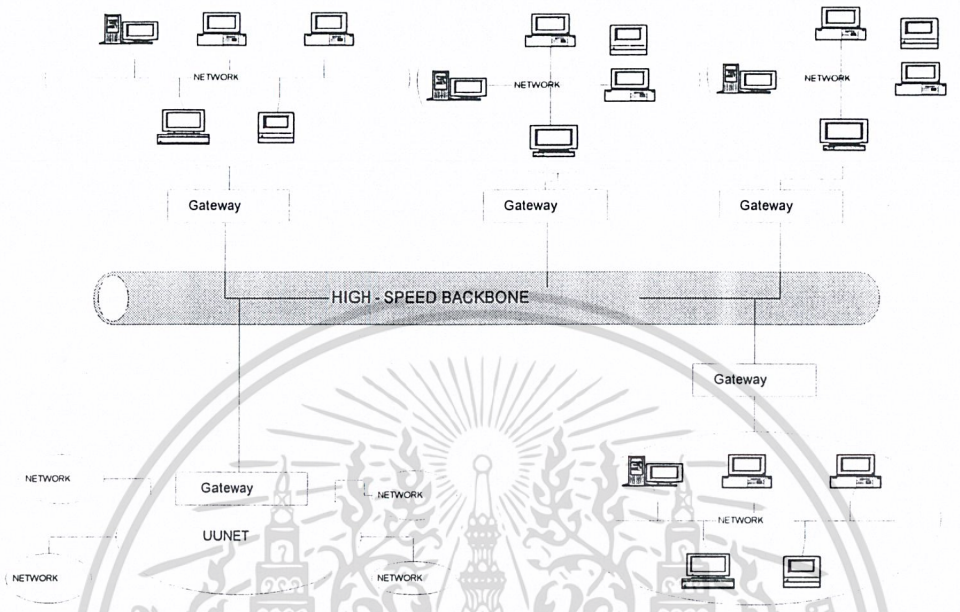
อินเทอร์เน็ตประกอบด้วยสายสื่อสารความเร็วสูงเป็นแบคโบน(Back bone) เป็นสายโทรศัพท์ตามมาตรฐาน TS สามารถวิ่งด้วยความเร็ว 44.736 เมกะบิทต่อวินาที

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

ข่าวสารจากเครือข่ายถูกส่งออกไปผ่านเกตเวย์เข้าสู่อินเทอร์เน็ต โดยที่เกตเวย์เป็นตัวเลือกทิศทางการเดินทางเพื่อไปยังปลายทางที่ต้องการ แต่ตามเส้นทางอาจจะต้องผ่านเกตเวย์อีกหลายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อรับช่วงส่งข่าวสารจนถึงที่หมาย ถึงแม้ว่าจะต้องเดินทางระยะไกลก็ตาม แต่ด้วยสายสื่อสารความเร็วสูงทำให้การส่งข่าวสารทำได้อย่างรวดเร็ว



รูปที่ 2.21 แสดงเครือข่ายอินเตอร์เน็ต

ข้อกำหนดรูปแบบของเกตเวย์ (Gateway Protocols)

เกตเวย์ต้องมีข้อมูลของเกตเวย์ตัวอื่นและรู้จักเครือข่ายปลายทาง เพื่อนำมาใช้ในการกำหนดเส้นทางที่ข่าวสารสามารถเดินทางไปถึงได้เร็วที่สุด เกตเวย์จะมีการแลกเปลี่ยนข้อมูลระหว่างกันเกี่ยวกับเส้นทาง รายละเอียด สถานะเครือข่ายและคุณสมบัติเครือข่ายย่อยที่ติดต่อเข้าสู่เครือข่ายใหญ่ตามลำดับขั้นจึงต้องมีการกำหนดรูปแบบพิเศษสำหรับเกตเวย์ขึ้นมา

ข้อกำหนดรูปแบบเกตเวย์แบ่งออกตามการใช้งานได้ดังนี้

1. IGP(Interior Gateway Protocol) ถูกนำใช้กับเกตเวย์ที่อยู่ในเครือข่ายลูกติดต่อกับเครือข่ายลูกที่อยู่ภายในเครือข่ายแม่เดียวกัน หรือติดต่อกันระหว่างเครือข่ายแม่กับครือข่ายลูก การเชื่อมโยงเกตเวย์ประเภทนี้มักจะไม่ค่อยมีการเปลี่ยนแปลง จึงเรียกว่าระบบอิสระจากกัน(Autonomous)หรือ Selfcomplete
2. GP (External Gateway Protocol) ในเครือข่ายใหญ่ๆ การติดต่อกับเครือข่ายจะมีการเปลี่ยนแปลงอยู่เสมอ ตามการเปลี่ยนแปลงของเครือข่ายย่อยที่เชื่อมโยงอยู่เป็นจำนวนมาก จึงมีข้อกำหนดรูปแบบที่ใช้กับการสื่อสารระหว่างเกตเวย์ของเครือข่ายด้วย EGP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2.3.3 โมเดลโอเอสไอ (OSI Model)

เพื่อลดปัญหาในความยุ่งยากสับสนในการติดต่อสื่อสารข้อมูล โครงสร้างของการสื่อสารข้อมูลภายในอุปกรณ์คอมพิวเตอร์ส่วนใหญ่จะถูกแบ่งเป็นชั้นๆ โดยแต่ละชั้นมีอิสระไม่ขึ้นต่อกันทำให้การแปลงบริการชั้นใดๆ ไม่ก่อปัญหากับบริการชั้นอื่นการเพิ่มเติมการบริการใหม่ทำได้โดยไม่ต้องเปลี่ยนแปลงโปรแกรมระบบเดิมและสิ่งที่สำคัญก็คือ การทำระดับชั้นนั้นทำให้ตัวโปรแกรมมีขนาดเล็กสามารถระบุส่วนที่จะต้องปรับปรุงได้แน่นอน ไม่ต้องวิตกกังวลถึงโปรแกรมส่วนอื่น ทำให้การพัฒนาประสิทธิภาพของระบบทำได้ง่าย และดียิ่งขึ้น

ISO (International Standardization for Organization) ซึ่งเป็นองค์กรที่จัดขึ้นมาเพื่อดูแลและส่งเสริมตลอดจนกำหนดมาตรฐานของการติดต่อสื่อสารของระบบเครือข่ายคอมพิวเตอร์ ได้พัฒนาสถาปัตยกรรม 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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.14 โอ เอส ไอ โมเดล

ข้อกำหนดรูปแบบแต่ละระดับชั้นของ โอ เอส ไอ มีการรุดหน้าที่ต้องปฏิบัติแน่นอน โดยจะต้องเป็นตามกลุ่มการบริการดังนี้

1. **ระดับชั้น7 แอปพลิเคชั่น(Application Layer)** เป็นเลเยอร์ชั้นบนสุดของรูปแบบ OSI ซึ่งเป็นขั้นที่ติดต่อระหว่างผู้ใช้โดยตรงซึ่งได้แก่โฮสท์คอมพิวเตอร์เทอร์มินัลหรือคอมพิวเตอร์ส่วนบุคคล(PC)เป็นต้น แอปพลิเคชั่นใน เลเยอร์นี้ สามารถนำเข้าหรือออกจากระบบเครือข่ายได้โดยไม่ต้องสนใจจะมีขั้นตอนการ ทำงานอย่างไรเพราะจะมีชั้นพรีเซนเตชั่นเป็นผู้รับผิดชอบอยู่แล้วในรูปแบบ 0SI นี้ แอปพลิเคชั่นจะการติดต่อกับชั้นพรีเซนเตชั่นโดยตรง
2. **ระดับชั้น6 พรีเซนเตชั่นเลเยอร์(Presentation Layer)** คอยรวบรวมข้อความและแปลงรหัสหรือแปลงในแบบของข้อมูลที่รับมาจากระดับชั้น 7 เพื่อให้เป็นไปแบบการสื่อสารเดียวกัน เพื่อช่วยลดปัญหาต่างๆที่อาจจะเกิดขึ้นกับผู้ใช้งานในระบบ
3. **ระดับชั้น5 เซสชั่นเลเยอร์(Session Layer)** ทำหน้าที่เชื่อมโยงระหว่างผู้ใช้คอมพิวเตอร์เครื่องอื่นๆ โดยผู้ใช้จะใช้คำสั่ง หรือข้อความที่กำหนดไว้ป้อนเข้าไปในระบบ ในการสร้างการเชื่อมโยงนี้ผู้ใช้จะต้องกำหนดรหัสตำแหน่งของจุดหมายปลายทางที่ต้องการติดต่อด้วยในบางครั้งเครือข่ายอาจมีชั้นเซสชั่นและชั้นทรานสปอร์ตรวมเป็นชั้นเลเยอร์เดียวกัน
4. **ระดับชั้น 4 ทรานสเปอร์ตเลเยอร์** (Transport Layer) บางครั้งเรียกว่าขั้นเครื่องต่อเครื่อง(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 (Feame)

   กรณีที่สอง เป็นบริการที่ทำการถอดรหัสข้อมูลที่รับเข้ามาพิจารณาตัดสินว่ามีความสัมพันธ์เกี่ยวข้องกับข้อมูลที่ได้ส่งออกไปหรือเปล่า เลเยอร์ชั้น ทรานสปอร์ต จะทำหน้าที่ตรวจสอบว่าข้อมูลที่ส่งมาจากชั้นเลเยอร์ที่ 5 นั้นไปถึงปลายทางจริงหรือไม่ ดังนั้นการกำหนดตำแหน่งของข้อมูลจึงสำคัญในชั้นนี้
5. **ระดับชั้น 3 เน็ตเวอร์คเลเยอร์(Network Layer)** เป็นชั้นที่ออกแบบหรือกำหนดเส้นทางการเดินทางของข้อมูลที่ส่ง-รับในการส่งผ่านข้อมูลระหว่างต้นทางและปลายทาง ซึ่งเป็นที่แน่นอนว่าในการสื่อสารข้อมูลผ่านเครือข่ายการสื่อสารจะต้องมีเส้นทางการรับ- ส่ง ข้อมูลมากกว่า 1 เส้นทางดังนั้นในชั้นนี้จะมีหน้าที่เลือกเส้นทางที่ใช้เวลาในการสื่อสารน้อยที่สุดและระยะทางสั้นที่สุดด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. **ระดับชั้น 2 ดาต้าลิ้งค์เลเยอร์(Data Link Layer)** เป็นเสมือนผู้ควบคุมการรวมบิทเข้าเป็นตัวอักษรและจัดข้อมูลให้เป็นแพ็คเกตรวมทั้งตรวจสอบความผิดพลาดในการส่งข้อมูล ถ้าผู้รับได้รับข้อมูลแล้วถูกต้องก็จะส่งสัญญาณยืนยันกลับมาว่าได้รับข้อมูลแล้ว เรียกว่าสัญญาณACK (Acknowledge)ให้กับผู้ส่งแต่ถ้าผู้ส่งไม่ได้รับสัญญาณ ACK หรือรับสัญญาณ NAK (Negative Acknowledge) กลับมาผู้ส่งอาจจะทำการส่งข้อมูลไปใหม่อีกหน้าที่ของชั้นนี้ คือ ป้องกันไม่ให้เครื่องส่งทำการส่งข้อมูลเร็วจนเกินขีดความสามารถของเครื่องผู้รับข้อมูลจะรับได้
7. **ระดับชั้น 1 ฟิสิกส์คอลเลเยอร์(Physical Layer)** เป็นระดับชั้นการติดต่อระหว่างอุปกรณ์กับสื่อกลางของเครือข่ายจริงๆ การส่งข้อมูลระดับชั้นนี้ข้อมูลนี้มีลักษณะเป็นบิท สื่อกลางทีใช้อาจจะเป็นสายไฟเบอร์ออปติค 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 ของอีกเครื่องหนึ่งเรียกการสื่อสารแบบนี้ว่าการสื่อสาร แบบเสมือนจริง(Virtual Communication)

โพรโตคอลของในเลเยอร์แต่และชั้นจะแตกต่างกันออกไป แต่อย่างไรก็ตามการที่เครื่องคอมพิวเตอร์หลายๆเครื่องจะติดต่อสื่อสารกันได้ในแต่ละเลเยอร์ของแต่ละเครื่องจะต้องใช้โพรโต-คอลแบบเดียวกันหรือถ้าใช้โพรโตคอลต่างชนิดกันต้องมีอุปกรณ์ หรือซอฟต์แวร์ที่สามารถแปลงโพรโตคอลที่ต่างกันนั้นให้มีรูปแบบอย่างเดียวกันเพื่อเชื่อมโยงให้คอมพิวเตอร์ทั้ง 2 เครื่องสามารถติดต่อกันได้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 ทีซีพี/ไอพี (TCP/IP : Transmission Control Protocol/Internet Protocol)

อินเทอร์เน็ตนับได้ว่าเป็นเครือข่ายบนเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่นเชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่นๆ แต่หาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันก็คือ แต่ละเครือข่ายใช้คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อ และระบบปฏิบัติการที่ต่างกัน มาตรฐานของ ทีซีพี(TCP) จึงถูกใช้เป็นกุญแจสำคัญในการแก้ปัญหาเหล่านี้ โดยกลายเป็นระบบเปิดที่สมบูรณ์แบบมีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่เครื่องพีซีจนถึงเครื่องเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ ทีซีพี จึงเป็นมาตรฐานที่ทั่วโลกยอมรับมีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน ทีซีพี/ไอพี มากมาย ดังนั้นจึงนับได้ว่า ทีซีพี/ไอพี เป็นหัวใจของอินเทอร์เน็ตเลยทีเดียว

ทีซีพี/ไอพี คืออะไร

- **ทีซีพี/ไอพี** เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Network Protocol) จัดทำเพื่อใช้ เป็นกฎเกณฑ์ให้เครื่องคอมพิวเตอร์ใช้งานร่วมกันในลักษณะของระบบเปิด(Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตาม จะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันใช้ได้
- **ทีซีพี/ไอพี** เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่างๆ ที่กำหนดให้ทำเมื่อเกิดความผิดพลาด หรือต้องทำเพื่อป้องกันเพื่อไม่ให้เกิดความผิดพลาด
- **พีซีพี/ไอพี** เกิดจากการนำข้อกำหนดของรูปแบบต่างๆ กันมาใช้ร่วมกัน ทีซีพี(TCP) และไอพี(IP) ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ (แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า ทีซีพี/ไอพี) ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกัน หรือต่างเครือข่ายกันก็ได้และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้นได้ภายในข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบ เครือข่ายจะใช้ ทีซีพี/ไอพีเป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ(Local Area Network)และเครือข่ายบริเวณกว้าง (Wide Area Network)ไม่ได้ใช้งานเฉพาะกับอินเตอร์เน็ตเท่านั้น

ส่วนประกอบของ ทีซีพี/ไอพี

จากที่กล่าวมาแล้วข้างต้นว่า TCP/IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่างๆซึ่งแบ่งเป็นกลุ่มได้ ดังนี้

กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocols) ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูล ระหว่างคอมพิวเตอร์สองเครื่อง แบ่งย่อยออกได้เป็นสองชนิด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา รวมทั้งห้ามมิให้นำเอกสารนี้ไปใช้

1. TCP (Transmission Control Protocol) เป็นการบริการแบบ คอนเน็คชั่นเบสเซอร์วิส(Connection Based Service) ซึ่งคอมพิวเตอร์ด้านผู้รับและส่งต้องส่งต่อถึงกันอยู่

ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบก็คล้ายกับระบบโทรศัพท์ที่ต้องติดต่อกันให้ได้ก่อนจะพูดคุยกันได้

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

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

1. IP (Internet Protocol) เป็นการกำหนดรูปแบบการส่งข้อมูล

2. ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับสถานะของ IP เช่นข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ใน เครือข่าย

3. RIP (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับทำการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุดเหมาะสมกับข้อมูลมากที่สุด

4. OSPF (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่งที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากทางที่สั้นที่สุดก่อน

กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address) ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยที่ไม่ว่าเครือข่ายจะใหญ่โตสักเพียงใดหรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตาม ที่อยู่ต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบนี้มีดังนี้

1. ARP (Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน

2. DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะการใช้งานจริงนั่นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล

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

SNMP (Simple Network Management Protocol) ข้อกำหนดรูปแบบที่ให้บริการข่าวสารต่างๆที่ แสดงสถานะของเครือข่ายและอุปกรณ์ที่ต่ออยู่บนเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี

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

ตารางที่ 2.15 โครงสร้างของชุดโพรโตคอล ทีซีพี/ไอพี เปรียบเทียบกับแบบจำลอง โอ เอส ไอ

1. Application Layerในชั้นนี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้ในเครือข่ายเช่นโปร-แกรมส่งถ่ายข้อมูล (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 เนื่องจากไม่มีการเชื่อมต่อระหว่างต้นทางกับปลายทางก่อนดาต้าแกรมแต่ละตัวสามารถเลือกเส้นทางไปโดยอิสระและไม่มีการรับประกันความถูกต้องข้อมูลหรือลำดับการส่ง

4.Network Interface Physical Layer ทำหน้าที่ควบคุมตัวกลางที่ใช้สื่อสารข้อมูลและรูปแบบการเชื่อมต่อในทางกายภาพ ชั้นนี้จะแบ่งข้อมูลออกเป็นส่วน ๆเรียกว่า เฟรม หรือแพคเก็ต และส่งข้อมูลที่ได้ไปยังปลายทางที่เชื่อมต่อกันอยู่บนเครือข่ายเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างระหว่างชุดโพรโตคอล TCP/IP

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

2. การติดต่อสื่อสารระหว่างเครือข่ายหรือการติดต่อผ่านอินเตอร์เน็ต คือการติดต่อสื่อสารข้อมูลระหว่างระบบคอมพิวเตอร์ 2 ระบบที่ไม่สามารถติดต่อสื่อสารกันได้โดยผ่านทางเครื่อข่าย การสื่อสารข้อมูลเพียงเครือข่ายเดียวได้ต้องอาศัยเครือข่ายตั้งแต่ 2 เครือข่ายขึ้นไปในการติดต่อสื่อสารกันและเครือข่ายเหล่านี้อาจจะมีลักษณะของเครือข่ายที่ต่างกันก็ได้

ความแตกต่างในเรื่องของอินเตอร์เน็ตระหว่างชุดโพรโตคอล TCP/IP กับรูปแบบ OSI ก็คือในชุดโพรโตคอล TCP/IP จะใช้โพรโตคอลสำหรับอินเตอร์เน็ตที่เรียกว่า โพรโตคอล IP(Internet Protocol)ซึ่งในรูปแบบOSI จะเรียกว่าโพรโตคอลสำหรับการอินเตอร์เน็ตว่า 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 กับข้อมูลซึ่งในรูปแบบ OSI นั้นจะแบ่งแยกการควบคุมการสื่อสารออกจากกันโดยใช้โปรโตคอลเซสชั่นและโปรโตคอลทรานสปอร์ตตามลำดับ

ลักษณะของการติดต่อ แบ่งออกเป็น 2 ชนิดคือ

Connection-Oriented คือการติดต่อที่ต้องมีการเชื่อมโพรเซส (Process) ที่จะมีการส่งหรือรับข้อมูลซึ่งใช้คำว่าวงจรเสมือน (Virtual Circuit) เพราะว่าจะทำงานเสมือนมีวงจรต่ออยู่ระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรเซส ถึงแม้ว่าข้อมูลนี้อาจจะจ่าย Pocket-Switching Network บริการชนิดนี้ส่วนมากจะใช้ในกรณีที่มีข่าวสารต้องการมากกว่าหนึ่งข่าวสาร ดังนั้นสามารถแบ่งขั้นตอนการทำงานออกเป็น

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

Connectionless หรือดาต้าแกรม คือจะไม่มีขั้นการสร้างการติดต่อ และขั้นการยกเลิกการติดต่อ แต่จะมีขั้นการส่งผ่านข้อมูลเพียงอย่างเดียว โดยข้อมูลซึ่งเรียกว่าดาต้าแกรมจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับดาต้าแกรมอื่น

ความสัมพันธ์ของ TCP และ UDP กับอินเตอร์เน็ต

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

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

TCP เป็นวิธีการสื่อสารที่มีความเชื่อมั่นได้มากกว่า UDP ทั้งการส่งและการตอบรับข่าวสาร วิธีของ UDP จะไม่มีการรับประกันว่าข่าวสารจะไปถึงผู้รับแต่ UDP มีวิธีการที่จะตรวจสอบโดยให้เครื่องคอมพิวเตอร์ปลายทางส่งข่าวสารการตอบรับกลับมาเมื่อได้รับข่าวสารที่ส่งไปให้ ทางด้านผู้ส่งจะตั้งเวลาว่านานเท่าใดควรมีการตอบรับกลับมา ถ้าเกินเวลา (time-out) แล้วยังไม่มีการตอบรับกลับมาก็สันนิฐานได้ว่าข่าวสารสูญหาย UDP ก็จะทำการส่งข่าวสารเดิมให้ใหม่

เครือข่ายขนาดใหญ่เช่นอินเตอร์เน็ต ซึ่งมีเครื่องคอมพิวเตอร์บนเครือข่ายเป็นจำนวนมาก การใช้ TCP ที่ต้องสร้างการติดต่อและยกเลิกอยู่บ่อย ๆ จึงเป็นไปไม่ได้ในทางปฏิบัติดังนั้นอินเตอร์เน็ต จึงใช้การส่งข่าวสารด้วย UDP แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบริการที่ดำเนินการด้วยชุดข้อกำหนดรูปแบบ TCP/IP ต้องออกแบบการให้บริการเป็น TCP หรือ UDP อย่างใดอย่างหนึ่งเท่านั้น การบริการอย่างเดียวจะให้บริการทั้งสองประเภทไม่ได้

เช่นการ Login จากระยะไกล (Telnet) และการโอนย้ายไฟล์ข้อมูล FTP ใช้การให้บริการแบบ TCP ส่วน TFTP (Trivial File Transfer Protocol) ใช้การให้บริการแบบ UDP

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 Telnet จากเครื่องคอมพิวเตอร์อื่นเข้ามา จะไม่สามารถทำการติดต่อได้เนื่องจากพอร์ตหมายเลข 23 สำหรับบริการ Telnet ถูกใช้งานอยู่ TCP พอร์ตเป็นพอร์ตทาง Logical ไม่ใช่พอร์ตที่อยู่ด้านหลังของเครื่องคอมพิวเตอร์ที่ใช้คอนเน็กเตอร์(Connector)บนแบคโบนเพื่อให้สามารถสื่อสารบริการหลายประเภทได้พร้อม ๆกัน

ซ็อกเก็ตเป็นหมายเลขกำหนดวงจรซึ่ง ทางเข้า-ออก ของระดับชั้น TCP ของเครื่องคอมพิวเตอร์บนเครือข่าย หมายเลขซ็อกเก็ตประกอบด้วยเลขไอพีแอดเดรสกับหมายเลขพอร์ตรวมกัน หมายเลขซ็อกเก็ตที่ใช้กับการสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่อยู่ต่างเครือข่ายได้เนื่องจากไอพีแอดเดรสของเครือข่ายจะไม่ซ้ำกัน ถึงแม้ว่าหมายเลขพอร์ตจะซ้ำกันได้

เครื่องคอมพิวเตอร์บนเครือข่ายจะสร้างตารางพอร์ตขึ้นมาเก็บรายการของพอร์ตที่มีอยู่เพื่อใช้ตรวจสอบว่าจะใช้บริการพอร์ตที่ต้องการได้หรือไม่และพอร์ตมีสถานะเป็นอย่างไรการที่พอร์ตที่อยู่ในรายการตารางพอร์ตของแต่ละเครื่อง ร้องขอการสื่อสารซึ่งกันและกัน เรียกว่า 'Binding' ของพอร์ตและวิธีที่ทำให้พอร์ตที่ถูกใช้สามารถใช้สื่อสารได้อีกจากการขอใช้บริการเดียวกันเรียกว่า 'Multiplexing'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ตารางที่ 2.16 หมายเลขพอร์ตที่ให้บริการต่างๆ

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
3. Concatenation วิธีรวมดาต้าแกรมให้เป็นบล็อก
4. Separation วิธีแบ่งบล็อกให้เป็นดาต้าแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดรูปแบบ ICMP (Internet Control Message Protocol)

ปัญหาหนึ่งทีเกิดขึ้นได้ง่ายกับดาต้าแกรม ก็คือ ดาต้าแกรมเดินทางผิดเส้นทาง ทำให้สูญหายหรือบางส่วนข่าวสารเสียหาย เครื่องคอมพิวเตอร์ด้านส่งต้องรู้เงื่อนไขที่จะทำให้เกิดความผิดพลาดต่างๆ บนเครือข่ายเพื่อให้ดาต้าแกรมถูกส่งออกไปอย่างถูกต้อง

ICMP จะรายงานความผิดพลาดในระบบให้กับไอพีพิเศษ ICMP มี Header เหมือนกับข่าวสารไอพีอื่น ๆ โดยจะถูกส่งรวมไปกับข่าวสารอื่นสู่เครื่องปลายทาง ซึ่งเครื่องปลายทางจะส่งรายงานของความผิดพลาดกลับมาให้เครื่องด้านส่งได้รู้ว่าเกิดอะไรขึ้นบ้านในการเดินทาง ซึ่งข้อมูลในรายงานจะถูกนำมาใช้แก้ปัญหาต่อไป

2.3.5 ไอพีแอดเดรส (IP Address)

เครื่องคอมพิวเตอร์ที่ต่อเชื่อมกับอินเตอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP มีอยู่เป็นจำนวนมาก เครื่องคอมพิวเตอร์แต่ละเครื่องต้องสามารถระบุหรือ อ้างอิงได้โดยไม่เกิดความซ้ำซ้อนกับเครื่องคอมพิวเตอร์อื่น มิฉะนั้นเล้วข่าวสารที่เครื่อข่ายรับมาจะไม่สามารถส่งไปให้กับเครื่องคอมพิวเตอร์ที่ต้องการข่าวสารนั้นได้ จึงต้องมีการจัดระบบที่ดี เครือข่ายอินเตอร์เน็ตหรือเครือข่ายที่ใช้ TCP/IP ได้ออกแบบการจัดการระบบตรงส่วนนี้ไว้แล้ว เครือข่ายอินเทอร์เน็ตใช้รหัสหมายเลขมากำหนดให้แต่ละเครือข่ายและเครื่องคอมพิวเตอร์ภายในเครือข่ายที่เชื่อมโยงเรียกรหัสหมายเลขเหล่านี้ว่า **อินเตอร์เน็ตแอดเดรส (Internet Address)**

ไอพีแอดเดรส ประกอบด้วยเลขฐานสองจำนวน 32 บิต แบ่งออกเป็น 4 ส่วน แต่ละส่วนมี 8 บิต เมื่อดูเฉพาะแต่ละส่วนเป็นเลขฐานสิบจะได้เลขจำนวน 256 ค่าไม่ซ้ำกัน (0-225)ไอพีแอดเดรสจะนำเอาหมายเลขทั้ง 4 ส่วนมารวมกัน โดยแยกแต่ละส่วนด้วยจุดดังนั้นหมายเลขทั้งหมดที่เป็นไปได้ โดยไม่ซ้ำกัน คือ 256^4 หรือ 4,294,967,296 จำนวน มีค่าหมายเลขจาก 000.000.000.000 จนถึง 255.255.255.255หมายเลขเหล่านี้เองที่อินเทอร์เน็ตใช้กำหนดให้กับเครือข่ายและเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิงถึง

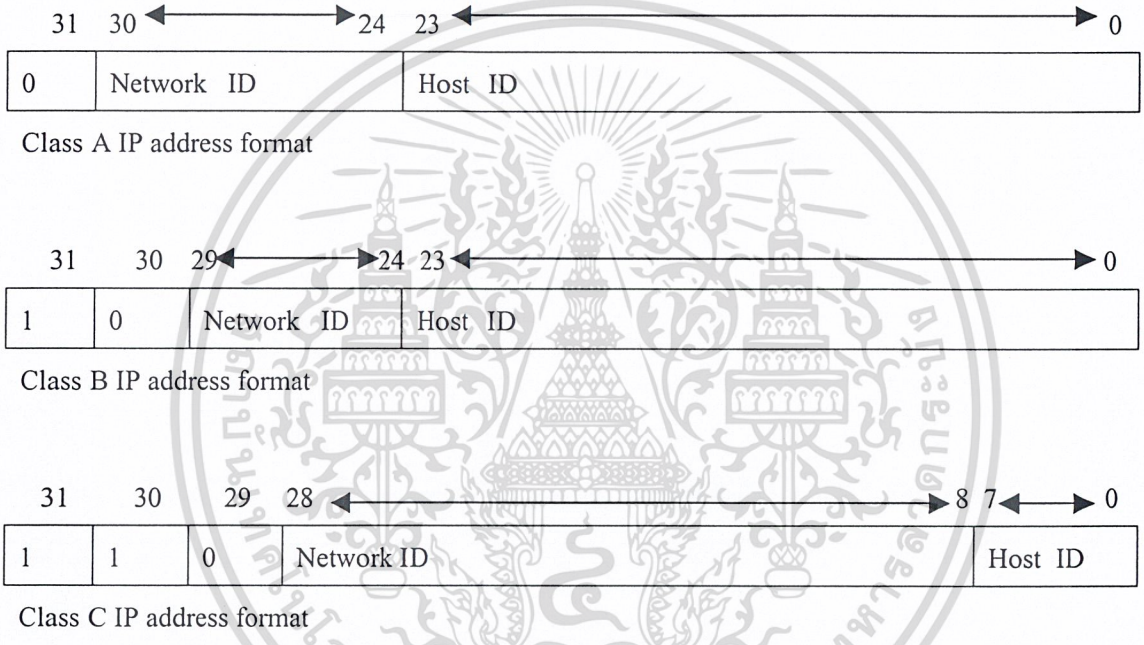
ไอพีแอดเดรสบางหมายเลขสงวนไว้ใช้ ด้วยจุดหมายกรณีพิเศษ ทำให้ไอพีแอดเดรสที่ให้งานทั่วไปลดลงจากจำนวนที่เป็นไปได้ความหมายของไอพีแอดเดรสจะแบ่งได้เป็น 2 กลุ่มดังนี้

1. กลุ่มที่ใช้เป็นระหัสประจำเครือข่าย
2. กลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย (Host Computers)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอพีแอดเดรสในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์สามารถซ้ำกันได้ แต่กลุ่มรหัสประจำเครือข่ายจะซ้ำกันไม่ได้ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิงถึง

นอกจากนี้เพื่อความเหมาะสมในการกำหนดไอพีแอดเดรสให้กับผู้ขอ ทางผู้บริหารอินเตอร์เน็ตแบ่งคลาสของผู้ขอไอพีแอดเดรสตามขนาดของเครือข่ายเพื่อให้ทรัพยากรส่วนนี้ถูกใช้อย่างคุ้มค่าที่สุด องค์กรขนาดใหญ่ก็จะจัดให้อยู่ในคลาสที่สามารถกำหนด ไอพีแอดเดรสให้กับเครื่องคอมพิวเตอร์ในเครือข่ายได้มาก การแบ่งคลาสจะแบ่งได้ดังนี้



รูปที่ 2.22 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าส่วนของหมายเลขประจำเครื่องคอมพิวเตอร์เป็น '0' ทุกบิต หมายถึง หมายเลขเครือข่ายนั้นใช้งานร่วมกับอุปกรณ์หาเส้นทางหรือเพื่อบอกให้เครื่องคอมพิวเตอร์รู้ว่าตัวเองอยู่ในเครือข่ายใด

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

- ถ้าไอพีแอดเดรสทั้งสองส่วนเป็น '1' ทุกบิต หมายถึง แอดเดรสที่ใช้กระจายข่าวหรืออีกนัยหนึ่งเครื่องคอมพิวเตอร์ที่ส่งแอดเดรสนี้เข้าสู่ระบบไม่ทราบว่าตนเองอยู่ในเครือข่ายใด

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

จากรูปที่ 2.22 จะแสดงประเภทของไอพีแอดเดรสและข้อกำหนดปลีกย่อยต่างๆเราสามารถทราบได้ว่าเครือข่ายขององค์กรถูกจัดอยู่ใน Class ใด โดยดูจากค่าหมายเลขของ 8 บิทแรกซ้ายมือสุดดังนี้

Class A: ไอพีแอดเดรสอยู่ในช่วง 1ถึง 126

Class B: ไอพีแอดเดรสอยู่ในช่วง 128 ถึง 191

Class C: ไอพีแอดเดรสอยู่ในช่วง 192 ถึง 233

2.4 การออกแบบซอฟท์แวร์และฮาร์ดแวร์

โครงสร้างของระบบควบคุมไฟฟ้า ผ่านระบบเครือข่าย ที่จะออกแบบ สามารถจำแนกออกเป็นส่วนประกอบได้สองส่วนหลัก คือ

1. **ส่วนซอฟท์แวร์** ซึ่งได้แก่
   - 1.1) โปรแกรมที่ติดตั้งที่เครื่องไคลเอนท์
   - 1.2) โปรแกรมที่ติดตั้งที่เครื่องเซิร์ฟเวอร์
   - 1.3) โปรแกรมที่ใช้ในการควบคุมเพื่อให้ไมโครคอนโทรลเลอร์ทำงาน
2. **ส่วนฮาร์ดแวร์** ได้แก่
   - 2.1) ส่วนการแปลงสัญญาณพอร์ตอนุกรม RS-232c เป็น RS-485
   - 2.2) ส่วนของวงจรควบคุมที่ประกอบด้วย MCS-51
   - 2.3) ส่วนของวงจรขับโหลด
   - 2.4) ส่วนของวงจรตรวจสอบสภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การออกแบบซอฟท์แวร์ แบ่งเป็นการออกแบบเป็น สองส่วนคือ

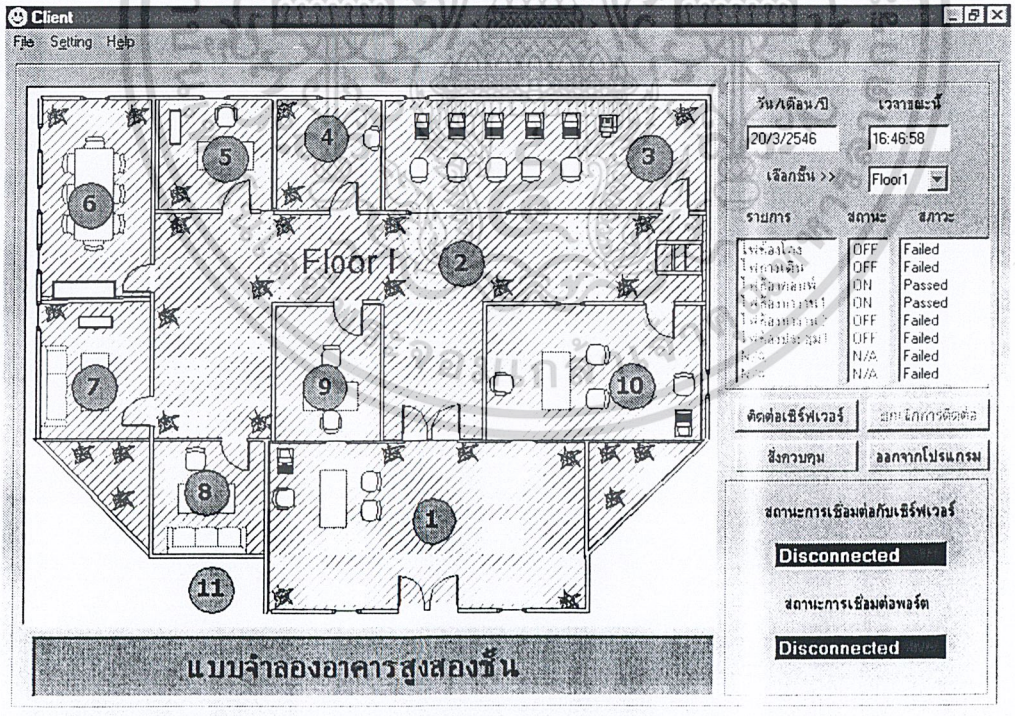
ส่วนที่ 1โปรแกรมบนเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) ประกอบด้วยหน้าจอโปรแกรมหลัก และsetting พอร์ต

1.โปรแกรมฝั่งไคลเอนด์ และ 2.โปรแกรมฝั่งเซิร์ฟเวอร์

ส่วนที่ 2โปรแกรมบนไมโครคอนโทรลเลอร์ ได้ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวประมวลผลหลักในการติดต่อกับฮาร์ดแวร์**ซึ่งมีรายละเอียดดังนี้**

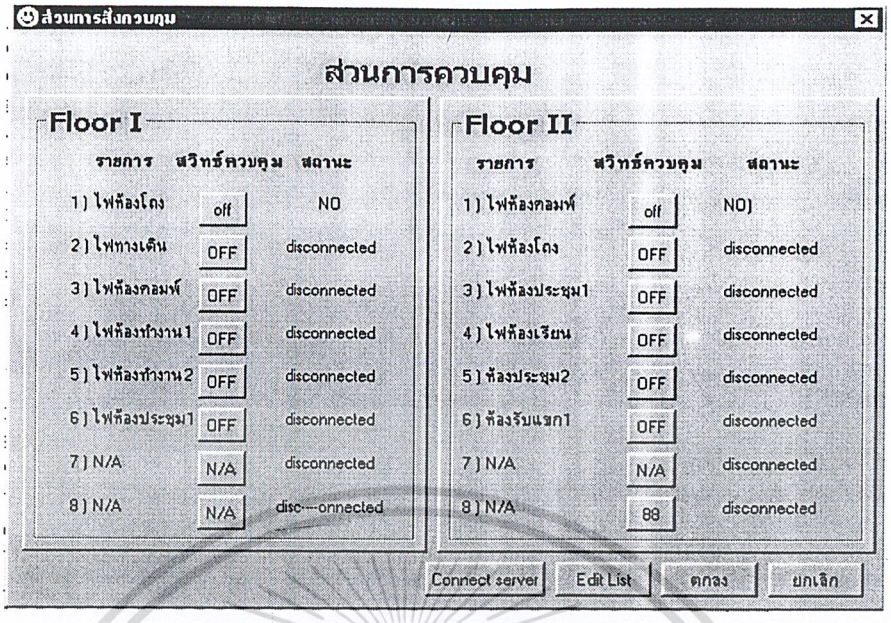
2.4.1.1 โปรแกรมฝั่งไคลเอนท์และการตรวจสอบการ login

1. การออกแบบโปรแกรมควบคุมอุปกรณ์ไฟฟ้าฝั่งไคลเอนด์ เริ่มจากการแสดงหน้าจอส่วน Loginเพื่อจำกัดสิทธิการใช้งานส่วนบุคคล และจะแสดงแผนผังของอาคารที่มองจากส่วน Topview ซึ่งเป็นจุดของอุปกรณ์ไฟฟ้าติดตั้งอยู่ และมีจุดควบคุมพร้อมกับแสดงสถานะของอุปกรณ์ไฟฟ้าทางโปรแกรม โดยสามารถเลือกจุดที่ต้องการควบคุมจากการคลิกเมาท์ที่จุดที่ต้องการจะสั่งงาน

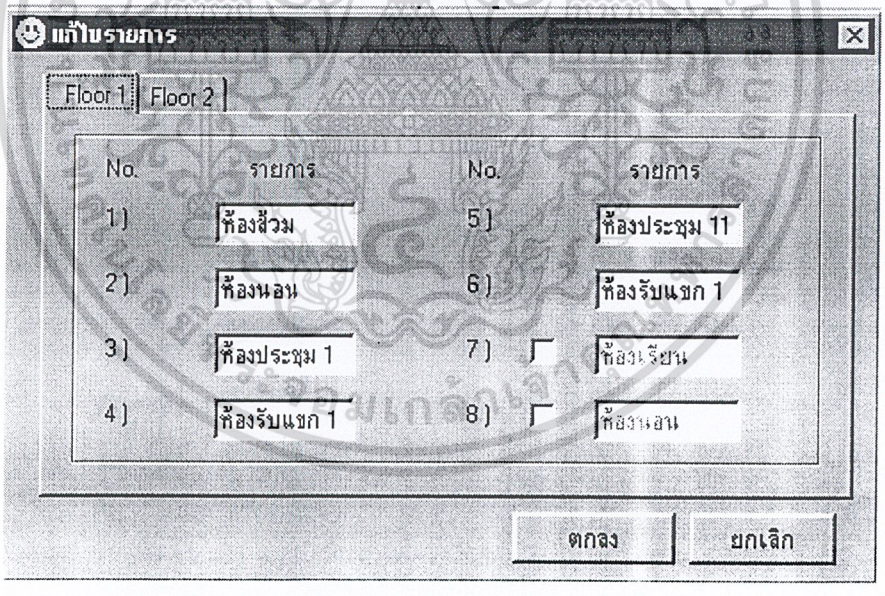


รูปที่ 2.23 แสดงกราฟฟิกยูเซอร์อินเตอร์เฟสของโปรแกรมฝั่งไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ

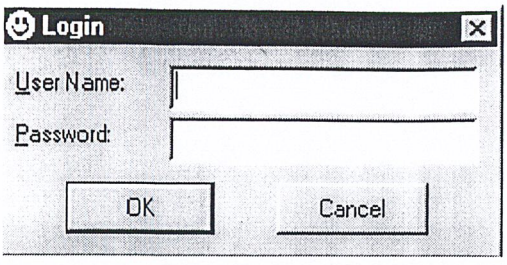


รูปที่ 2.24 แสดงเมนูควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า

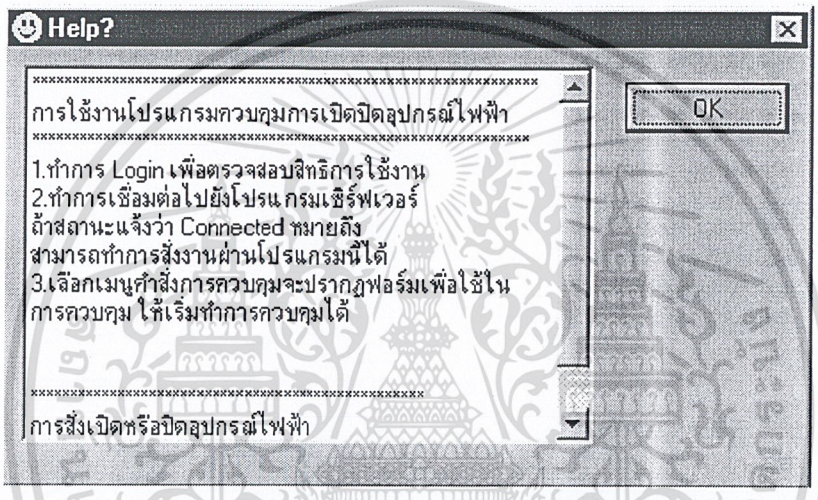


รูปที่ 2.25 แสดงหน้าจอของส่วนแก้ไขรายการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 แสดงหน้าจอของ Login



รูปที่ 2.27 แสดง Help File

2. การทำงานของโปรแกรมควบคุมอุปกรณ์ไฟฟ้าฝั่งไคลเอนต์ ซึ่งอธิบายได้ตามลำดับดังนี้

2.1 เริ่มต้นการทำงานของโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า

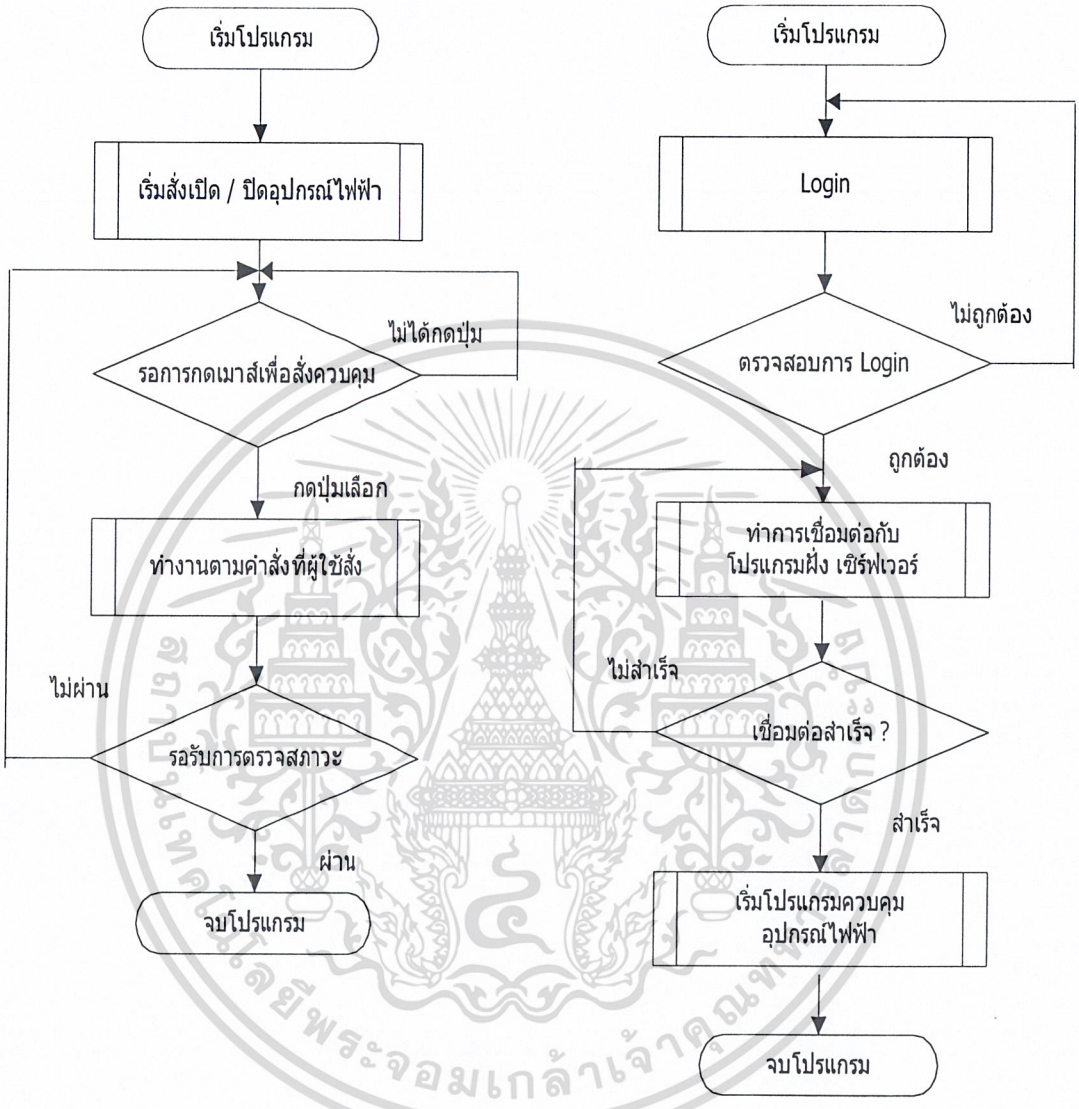
2.2 ตรวจสอบการ Login ว่ามีสิทธิในการใช้โปรแกรมหรือไม่ ถ้าไม่ใช่ให้ login อีกครั้ง แต่ถ้ามีสิทธิให้ทำการเชื่อมต่อ Server และไปยังขั้นตอนต่อไป

2.3 ทำการเชื่อมต่อโปรแกรมไปยังเซิร์ฟเวอร์

2.4. เริ่มใช้งานโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากขั้นตอนดังกล่าวเขียนเป็นโฟลว์ชาร์ตดังข้างล่างนี้



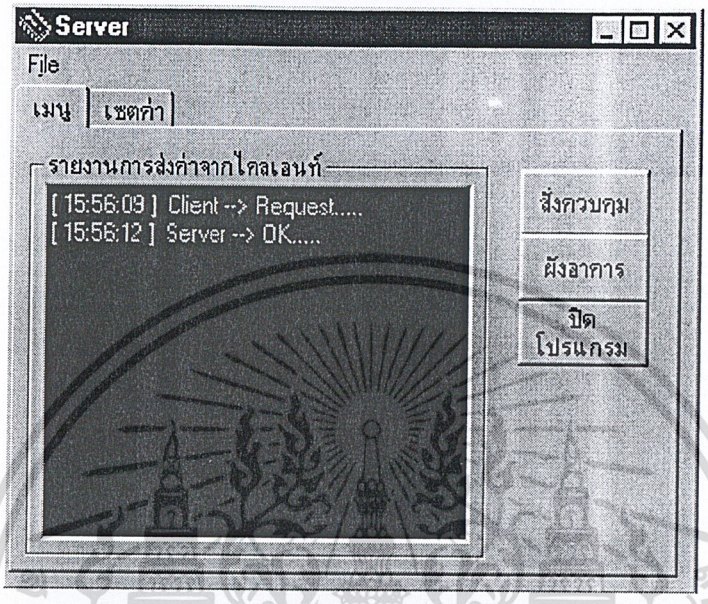
รูปที่ 2.28 แสดงโฟลว์ชาร์ตการทำงานของโปรแกรมในส่วนการ Login และเชื่อมต่อกับเซิร์ฟเวอร์

รูปที่ 2.29 แสดง โฟลว์ชาร์ต แสดงการใช้งานโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า

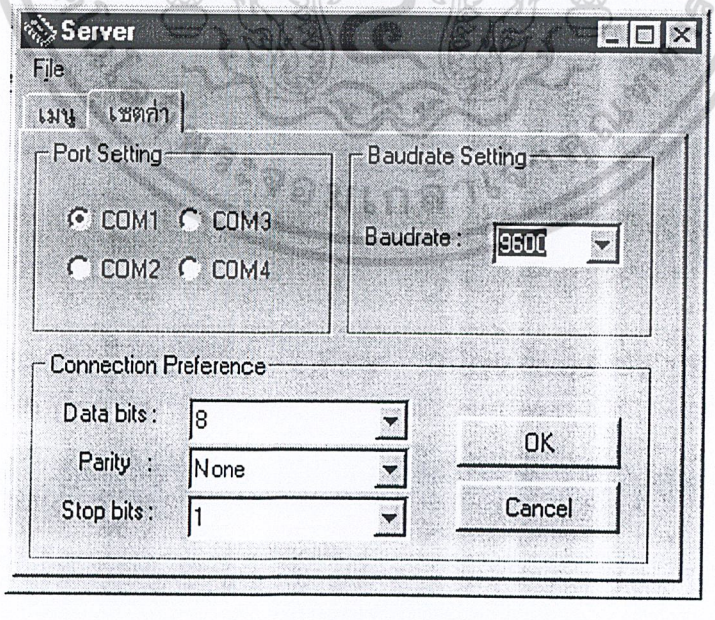
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.2โปรแกรมฝั่งเซิร์ฟเวอร์

จะแสดงสถานะการติดต่อระหว่างไคลเอนท์กับเซิร์ฟเวอร์ ได้แสดงดังรูปดังนี้



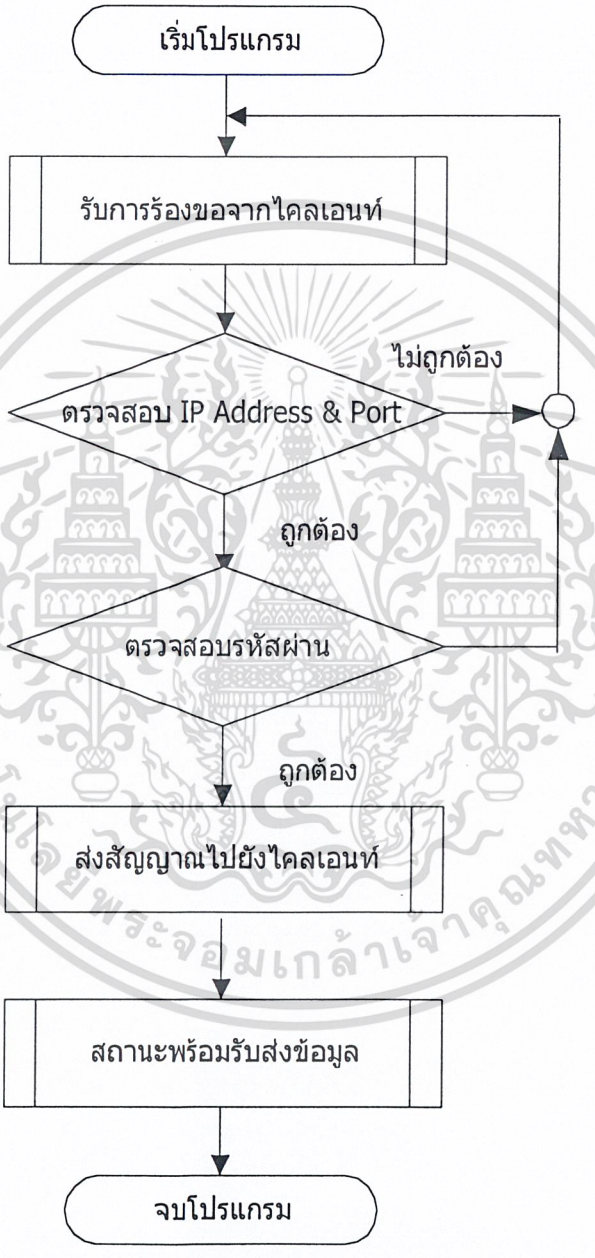
รูปที่ 2.30 แสดงโปรแกรมฝั่งเซิร์ฟเวอร์



รูปที่ 2.31 แสดงเมนูการเซตค่าพอร์ตอนุกรมของฝั่งเซิร์ฟเวอร์

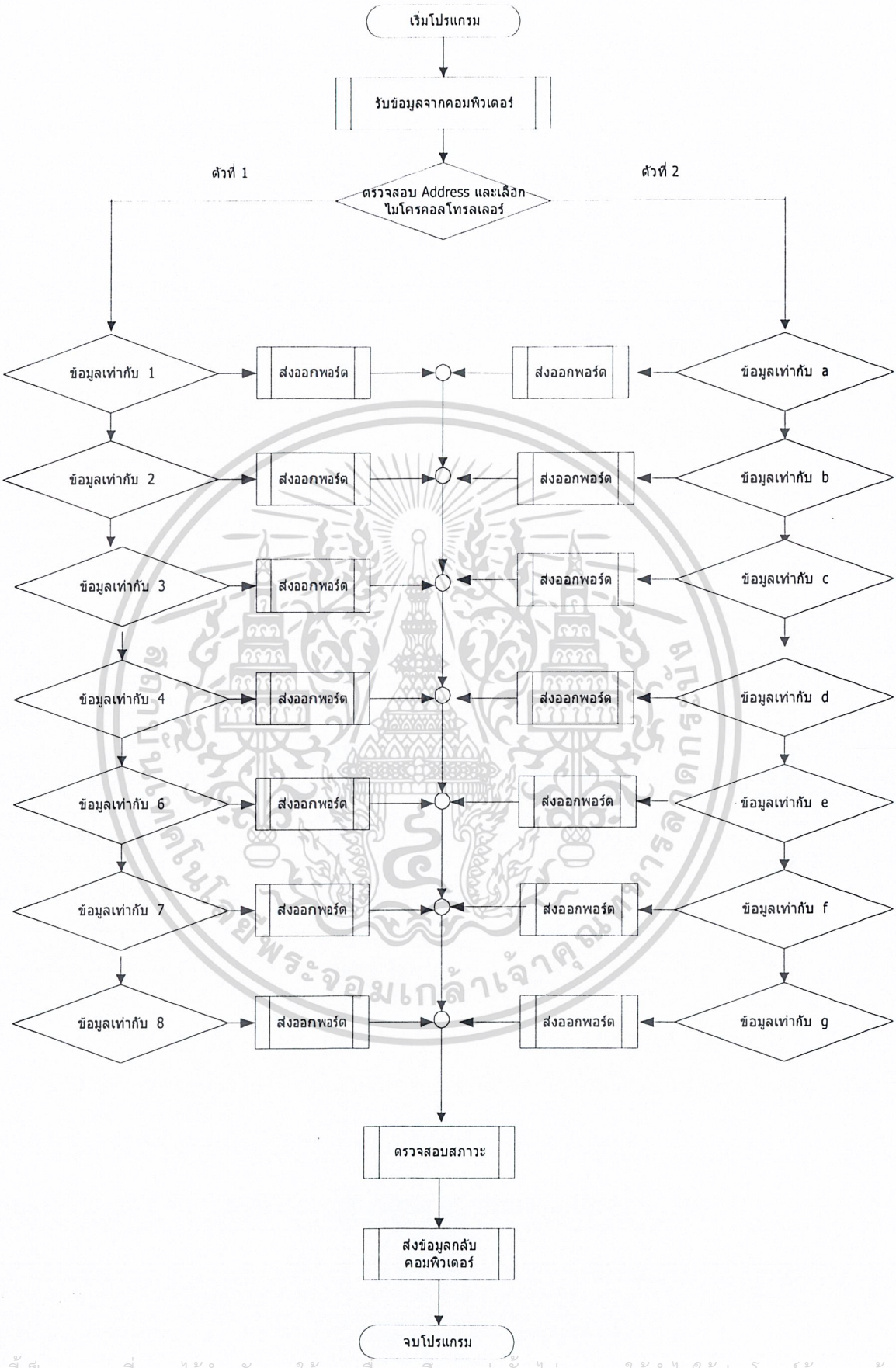
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมฝั่งเซิร์ฟเวอร์อธิบายได้ด้วยโฟล์วชาร์ตข้างล่างนี้



รูปที่ 2.32 แสดงการทำงานของโปรแกรมฝั่งเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.33 แสดงการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์

2.4.2 การออกแบบทางฮาร์ดแวร์

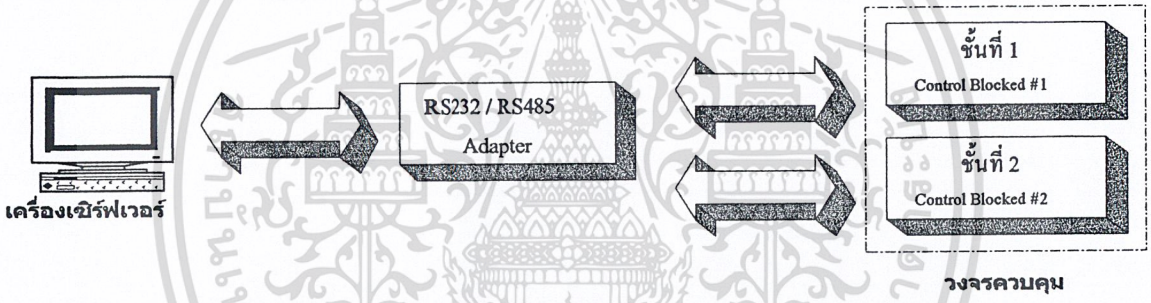
ในการออกแบบทางฮาร์ดแวร์นั้นจะประกอบไปด้วย ส่วนต่าง ๆ ดังต่อไปนี้

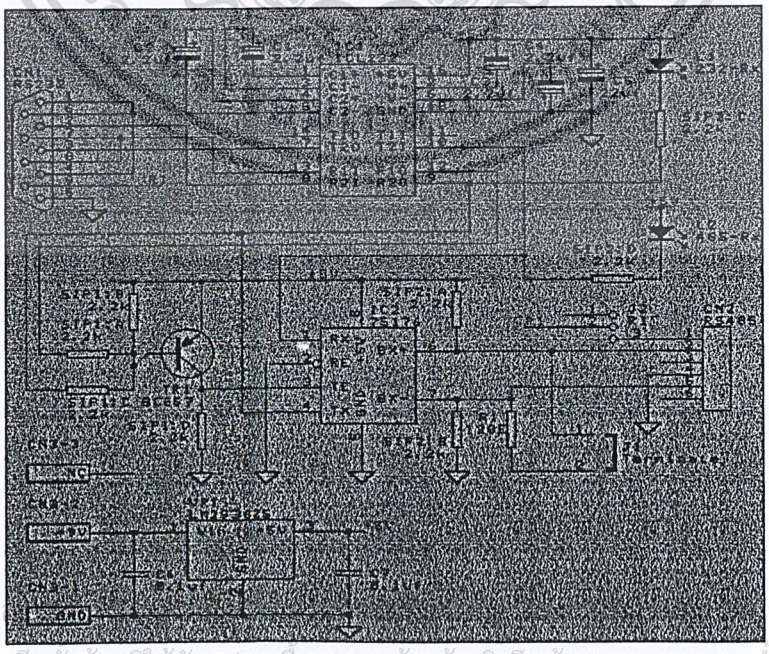
2.4.2.1 วงจรแปลงสัญญาณ

การออกแบบวงจรแปลงสัญญาณ

วงจรแปลงสัญญาณจะทำหน้าที่เป็นตัวกลางในการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับวงจรควบคุม มีอัตรารับส่งข้อมูลที่ 9600 บิตต่อวินาที ซึ่งประกอบด้วยช่องทางการสื่อสารแบบ RS-232 จำนวน 1 ช่องทาง ที่ใช้สำหรับติดต่อกับคอมพิวเตอร์ และช่องทางการสื่อสารแบบ RS-485 จำนวน 1 ช่องทางซึ่งใช้สำหรับติดต่อกับวงจรส่วนควบคุม และสามารถติดต่อกับวงจรควบคุมจำนวน 32 ชุด ซึ่งก็หมายถึงมีเอ้าท์พุทจำนวน 255 จุด

การส่งข้อมูล RS-485 นั้นที่ใช้ในวงจรนี้จะใช้แบบใช้สายสัญญาณ 2 เส้น เนื่องจากสามารถสร้างเป็นเครือข่ายข้อมูลที่แต่ละบอร์ดสามารถเป็นได้ทั้งตัวรับและตัวส่งเหมือนกันทุกบอร์ด การเชื่อมต่อสายก็จะเป็นดังรูป





เอกสารนี้เป็นเอกสารที่ ระโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.34 แสดงบล็อกไดอะแกรมและวงจรแปลงสัญญาณ

การทำงานของวงจรแปลงสัญญาณ

วงจรภาคแปลงสัญญาณ ทำหน้าที่แปลงสัญญาณ RS-232 ซึ่งเป็นการส่งสัญญาณแบบสมดุล (Unbalance Transmission) ให้เป็นสัญญาณ RS-485 ซึ่งเป็นการส่งสัญญาณแบบสมดุล (Balance Transmission) และสัญญาณที่ถูกแปลงมีดังนี้

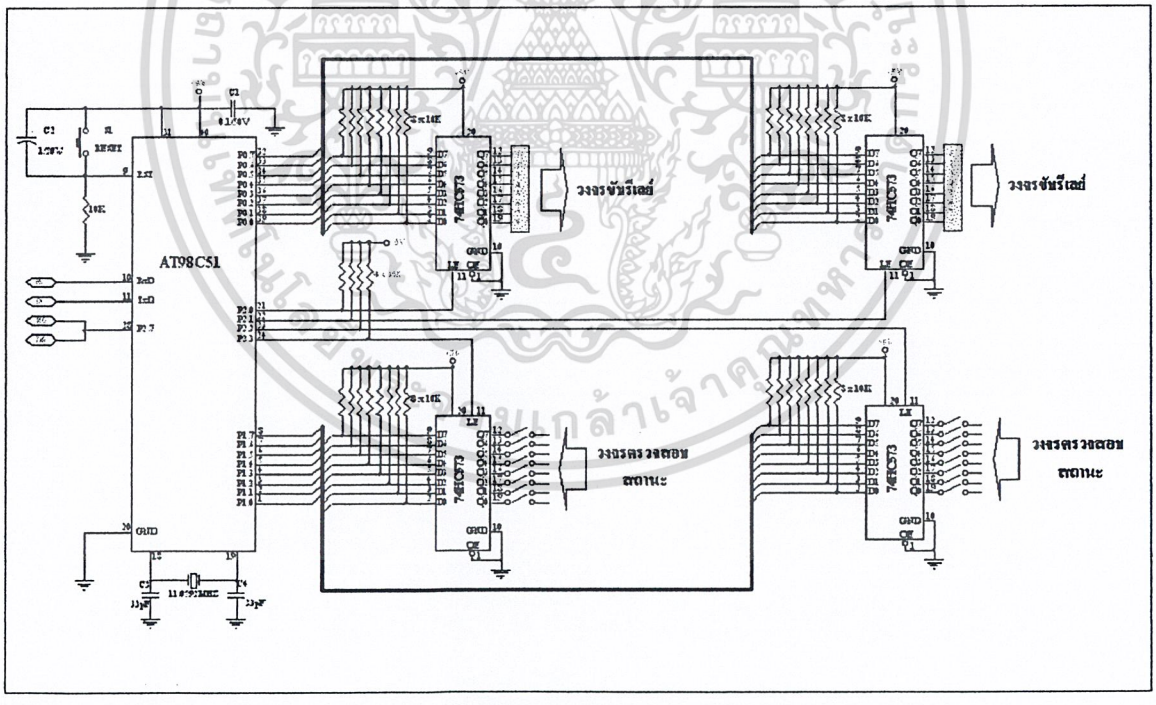
RXD เป็นสัญญาณข้อมูลอินพุตที่ PC ได้รับจากบอร์ดอื่นส่งมาให้

TXD เป็นสัญญาณข้อมูลเอาต์พุตที่ PC ต้องการส่งออกไป

RTS เป็นสัญญาณควบคุมทิศทางของสัญญาณข้อมูลว่าต้องการรับหรือส่งโดยถ้า RTS= 12 โวลต์ หรือมากกว่า 5 โวลต์ จะเป็นการส่งข้อมูล แต่หาก RTS = -12 โวลต์หรือน้อยกว่า 0 โวลต์ 0 จะหมายถึงการรับข้อมูลคือแปลงสัญญาณจาก RS485 เป็น RS232

นอกจากนี้แล้วได้นำสัญญาณ DTR ไปสร้างไฟเลี้ยงให้แก่บอร์ดด้วยกรณีที่อาศัยไฟเลี้ยงจากพอร์ต RS232

2.4.2.2 วงจรควบคุม



รูปที่ 2.35 แสดงวงจรควบคุมและส่วนขยายพอร์ตในแต่ละชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรควบคุม

วงจรควบคุมที่ใช้ในโครงงานนี้ประกอบด้วยไมโครคอนโทรลเลอร์ MCS-51 ที่ทำหน้าที่ในการประมวลผลเพื่อควบคุมอุปกรณ์ไฟฟ้า ซึ่งวงจรควบคุม 1 ชุด สามารถควบคุมอุปกรณ์ 8 จุด แต่ได้ทำการออกแบบวงจรให้สามารถควบคุมอุปกรณ์จำนวนมากกว่านั้น จึงจำเป็นที่จะต้องพึ่งพาอุปกรณ์อื่น นั่นคือ การขยายพอร์ต ซึ่งการขยายพอร์ตนั้นทำได้หลายวิธี อย่างเช่น การขยายพอร์ตด้วยไอซี 8255 หรือการใช้ระบบไอสแควซี ก็ทำให้เราสามารถขยายพอร์ตได้ตามต้องการ แต่เพื่อให้สอดคล้องกับงบประมาณในการพัฒนาและราคาของไอซี 8255 และไอซีจำพวกไอสแควซีสูงมากหลายเท่า เพราะฉะนั้นเราจึงใช้วิธีการที่ง่ายที่สุดในการประยุกต์ใช้ไอซีขยายพอร์ตโดยใช้ไอซี 74HC573 ซึ่งเป็นไอซีชิฟรีจีสเตอร์ เนื่องจากราคาถูกกว่าไอซีข้างต้นมาก อีกทั้งใช้งานง่ายความสามารถก็ไม่ด้อยไปกว่าไอซีข้างต้นอีกเช่นกันและสำหรับการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ MCS-51 ได้เซตอัตราการรับส่งข้อมูล 9600 บิตต่อวินาที ซึ่งพบว่าเป็นอัตราการรับส่งที่เหมาะสมต่อการสื่อสาร

การทำงานของวงจรควบคุม

การทำงานของวงจรนั้นเริ่มต้นด้วยการอ่านค่าสถานะเดิมจากหน่วยความจำที่ได้ทำการเซตไว้เริ่มแรกซึ่งจะให้เป็นศูนย์ และจากนั้นจะนำค่าสถานะมาเป็นสัญญาณส่งให้กับภาคต่อไป แล้วรอรับการอินเตอร์รัปต์จากพอร์ตสื่อสารแบบอนุกรม ซึ่งเราจะให้ไอซีเบอร์ SN75176 ทำงานเป็นตัวรับข้อมูล เพราะฉะนั้นเวลาที่เกิดการอินเตอร์รัปต์จากพอร์ตสื่อสารอนุกรม มันจะทำการรับข้อมูลขนาด 1 byte เข้ามาตรวจสอบว่าตรงกับตำแหน่งของวงจรที่กำหนดไว้หรือไม่ หากข้อมูลที่ส่งมาจากคอมพิวเตอร์นั้นตรงกับตำแหน่งของวงจรที่ได้กำหนด มันก็จะทำการส่งให้กับภาคต่อไป แต่ถ้าไม่ตรงมันก็จะกลับมารอรับอินเตอร์รัปต่อไป

2.4.2.3 วงจรขับรีเลย์

การออกแบบวงจรขับรีเลย์

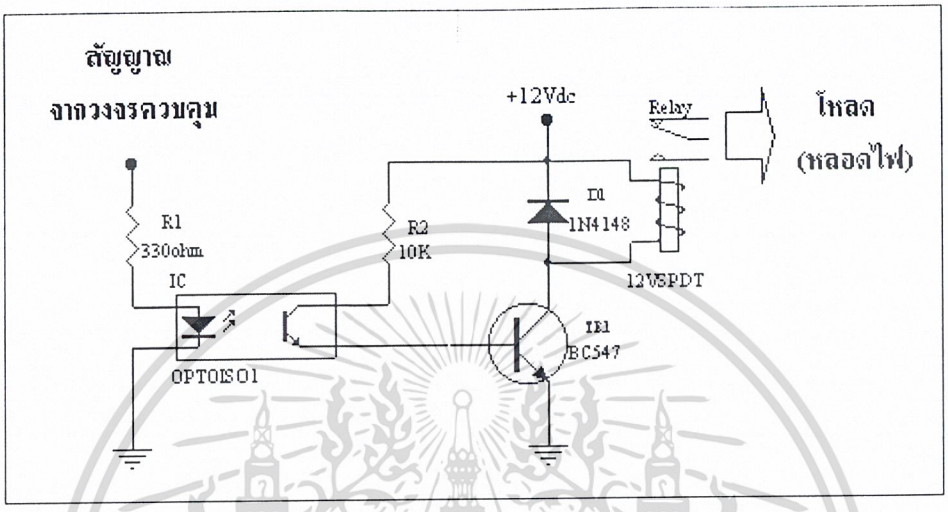
วงจรขับรีเลย์จะทำหน้าที่ควบคุมการทำงานของอุปกรณ์ไฟฟ้ากระแสสลับ โดยจะใช้สัญญาณเอาต์พุทที่ได้จากไมโครคอนโทรลเลอร์ เพื่อที่จะกำหนดการทำงานของวงจรขับรีเลย์ว่าต้องการให้อุปกรณ์ไฟฟ้าทำงานหรือหยุดการทำงาน โดยวงจรขับดังกล่าวในโครงงานนี้จะใช้กับโหลดที่กินกระแสไม่เกิน 1 mA โดยใช้ OptoIsolator เป็นสวิตช์ที่เชื่อมระหว่างภาคอินพุทกับภาคเอ้าท์พุทของวงจรได้ ซึ่งการเชื่อมต่อลักษณะนี้จะเป็นการเชื่อมโยงด้วยแสง และรูปข้างล่างจะแสดงส่วนของวงจรขับเลย์ที่ได้ออกแบบ

การทำงานของวงจรขับรีเลย์

การทำงานของวงจรขับเลย์นั้นเราจะใช้สัญญาณที่ได้จากพอร์ตของไมโครคอนโทรลเลอร์ในแต่ละพอร์ตที่กำหนดไว้ ป้อนเข้าที่ขาคาโทดของออปโต้ไอโซเลเตอร์ โดยจะต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความต้านทาน R1 ไว้จำกัดกระแสที่จะไหลผ่านออปโต้ไอโซเลเตอร์ เพื่อควบคุมการทำงานของทรานซิสเตอร์ในด้านเอ้าพุทของวงจรส่งผลให้กระแสไหลผ่านขดลวดภายในของรีเลย์ ทำให้เกิดการเหนี่ยวนำของไฟฟ้าในขดลวด และจากนั้นจะทำให้รีเลย์ทำงาน



รูปที่ 2.36 แสดงวงจรขับรีเลย์

2.4.2.4 วงจรตรวจสอบสถานะ

วงจรตรวจสอบสถานะ ทำหน้าที่ตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าแต่ละจุด ซึ่งเราจะใช้สัญญาณที่ได้จากวงจรตรวจสอบสถานะเป็นสัญญาณอินพุตให้ไมโครคอนโทรลเลอร์ โดยถ้าหากสัญญาณเป็นลอจิกต่ำ แสดงว่าอุปกรณ์ทำงาน แต่ถ้าหากสัญญาณเป็นลอจิกสูง จึงหมายถึง อุปกรณ์ไม่ทำงาน ซึ่งการตรวจสอบสถานะอุปกรณ์ไฟฟ้านั้นเราได้ออกแบบการตรวจสอบเป็นสองช่วง คือ

1.การตรวจสอบสถานะอุปกรณ์ที่รีเลย์

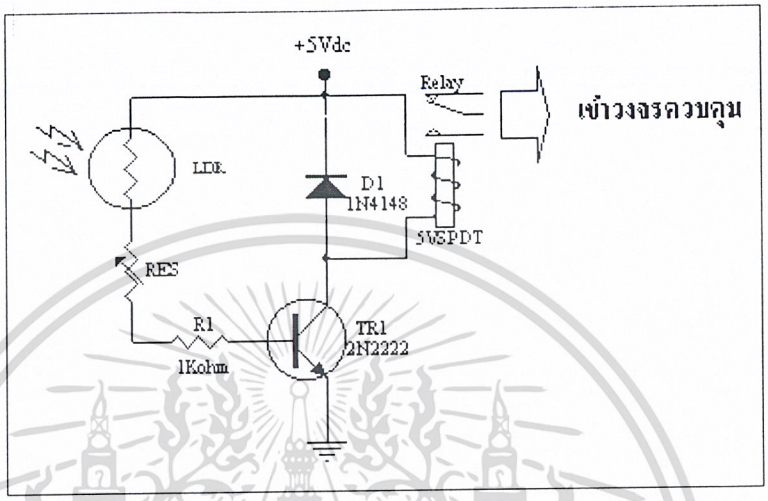
2.การตรวจสอบสถานะอุปกรณ์ทางกายภาพ โดยใช้อุปกรณ์จำพวกเซนเซอร์ ในโครงงานนี้จะใช้ LDR

การทำงานของวงจรตรวจสอบสถานะ

การทำงานของวงจรตรวจสอบสถานะเริ่มเมื่อรีเลย์ที่ทำหน้าที่เป็นสวิตช์ปิดวงจรส่งผลให้หลอดไฟฟ้าสว่างขึ้นทำให้เกิดการเปลี่ยนแปลงค่าความต้านทานในแอลดีอาร์ นั่นคือแอลดีอาร์จะมีความต้านทานน้อยเมื่อแสงที่ตกกระทบลงมา และเมื่อแสงที่ตกกระทบแหม่าะสมจะทำให้กระแสสามารถไหลผ่านวงจรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้เกิดการไบแอสที่ขาเบสของทรานซิสเตอร์ ทำให้ทรานซิสเตอร์ทำงาน และรีเลย์ทำงานตามลำดับรูปที่ 2.37

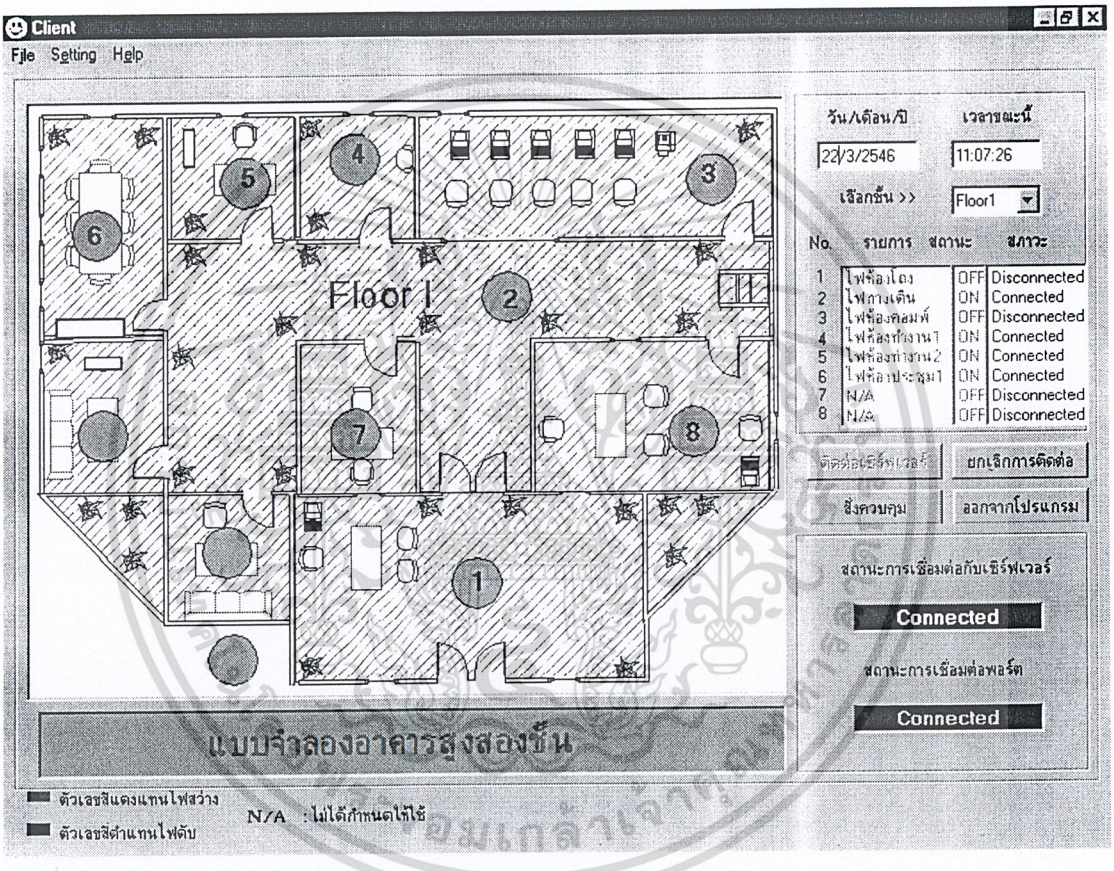


รูปที่ 2.37 แสดงวงจรตรวจสอบสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

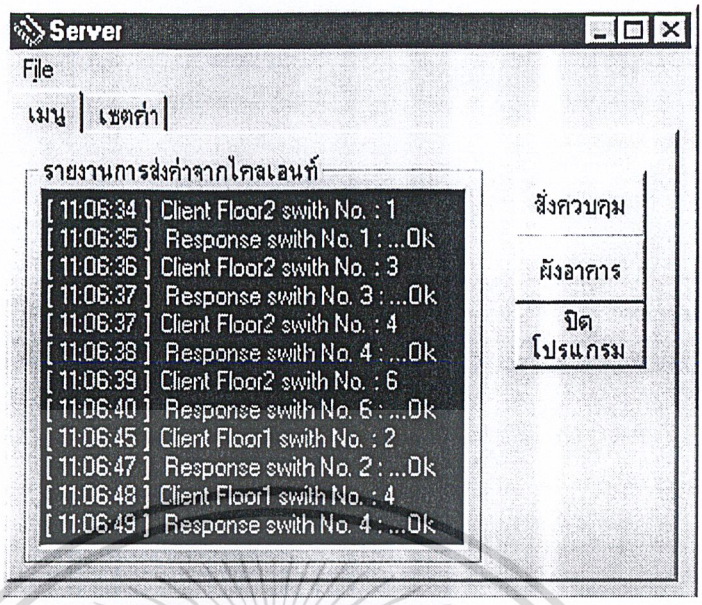
บทที่ 3
ผลการทดลอง

ในการทดลองได้ใช้โปรแกรมฝั่งไคลเอนท์เชื่อมต่อโปรแกรมฝั่งเซิร์ฟเวอร์ ซึ่งโปรแกรมฝั่งเซิร์ฟเวอร์นั้นได้จำลองเว็บเซิร์ฟเวอร์ในเครื่องเดียวกันนี้โดยใช้ไอพีแอดเดรส 127.0.0.1 เป็นแอดเดรสของเครื่องเซิร์ฟเวอร์เพื่อสะดวกในการทดสอบ ซึ่งโปรแกรมฝั่งไคลเอนท์และเซิร์ฟเวอร์ดังกล่าวมีหน้าตาดังรูป



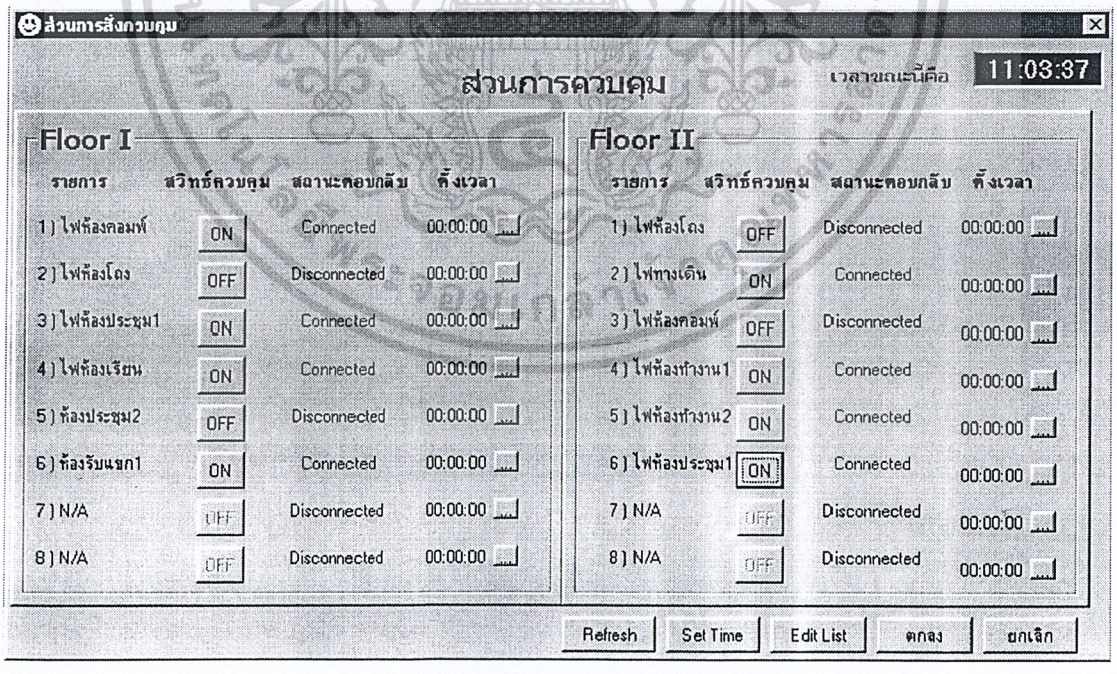
รูปที่ 3.1 โปรแกรมฝั่งไคลเอนท์ที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โปรแกรมทางฝั่งเซิร์ฟเวอร์

เมื่อได้ทำการเชื่อมต่อกับเซิร์ฟเวอร์สำเร็จแล้วขั้นต่อไปคือการสั่งควบคุมอุปกรณ์ไฟฟ้า โดยได้เลือกเมนูการสั่งควบคุม ขึ้นมาดังรูป



รูปที่ 3.3 แสดงเมนูของส่วนการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการทำงานของโปรแกรมทั้งหมดนั้นได้อธิบายไว้ในโฟล์วชาร์ตในบทที่ 2

หลังจากนั้นได้ทดลองการส่งค่าของแต่ละปุ่ม พบว่าสามารถส่งสัญญาณไปควบคุมหลอดไฟฟ้าได้ และมีการตอบสนองกับมาแสดงที่หน้าจอคอมพิวเตอร์ โดยมีผลลัพท์ดังตารางข้างล่างนี้

ชั้นที่ 1	**ผลการติดดับ**	**การตอบสนองกลับPC**	**หมายเหตุ**
1 ไฟห้องโถง	ติด	ตอบสนอง	
2 ไฟทางเดิน	ติด	ตอบสนอง	
3. ไฟห้องคอมพิวเตอร์	ติด	ตอบสนอง	
4. ไฟห้องทำงาน 1	ติด	ตอบสนอง	
5. ไฟห้องทำงาน 2	ติด	ตอบสนอง	
6.ไฟห้องประชุม	ติด	ตอบสนอง	
7. N/A	-	-	ไม่ได้ใช้
8. N/A	-	-	ไม่ได้ใช้

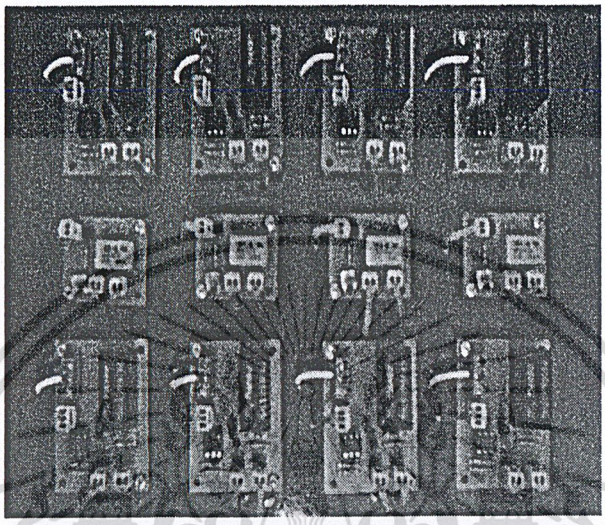
ตารางที่ 3.1 แสดงผลการทดลองการติดดับของหลอดไฟของชั้นที่ 1 และการตอบสนองกลับสู่คอมพิวเตอร์

ชั้นที่ 2	**ผลการติดดับ**	**การตอบสนองกลับPC**	**หมายเหตุ**
1 ไฟห้องคอมพิวเตอร์	ติด	ตอบสนอง	
2 ไฟห้องโถง	ติด	ตอบสนอง	
3. ไฟห้องประชุม 1	ติด	ตอบสนอง	
4. ไฟห้องเรียน	ติด	ตอบสนอง	
5. ไฟห้องประชุม 2	ติด	ตอบสนอง	
6.ไฟห้องรับแขก 1	ติด	ตอบสนอง	
7. N/A	-	-	ไม่ได้ใช้
8. N/A	-	-	ไม่ได้ใช้

ตารางที่ 3.2 แสดงผลการทดลองการติดดับของหลอดไฟของชั้นที่ 2 และการตอบสนองกลับสู่คอมพิวเตอร์

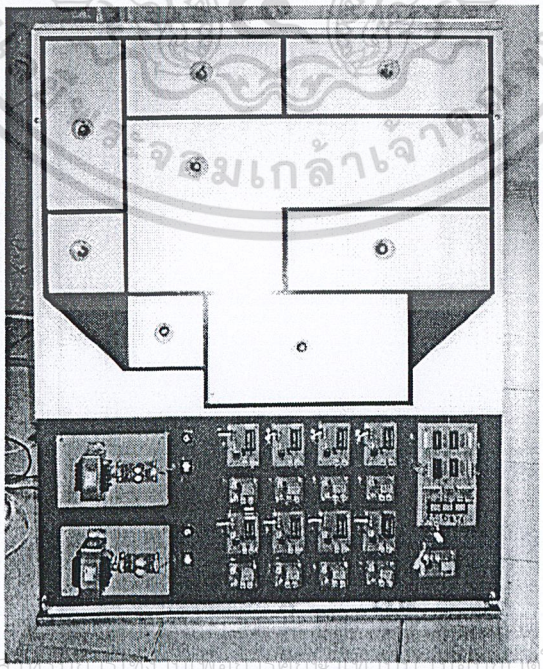
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบด้านไมโครคอนโทรลเลอร์ MCS-51 และวงจรควบคุมอุปกรณ์ รวมไปถึงวงจรตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้า(อุปกรณ์ไฟฟ้าชนิด ปิด-เปิด เราใช้อุปกรณ์ให้แสงสว่าง) จากวงจรในบทที่ 2 เป็นวงจรที่ใช้ในการควบคุมอุปกรณ์ไฟฟ้า เราสามารถทำรูปแบบสำเร็จตามรูปที่ 3.4 ด้านล่างนี้



รูปที่ 3.4 แสดงชุดที่ประกอบสำเร็จของวงจรขับโหลด และตรวจสอบสภาวะการทำงาน

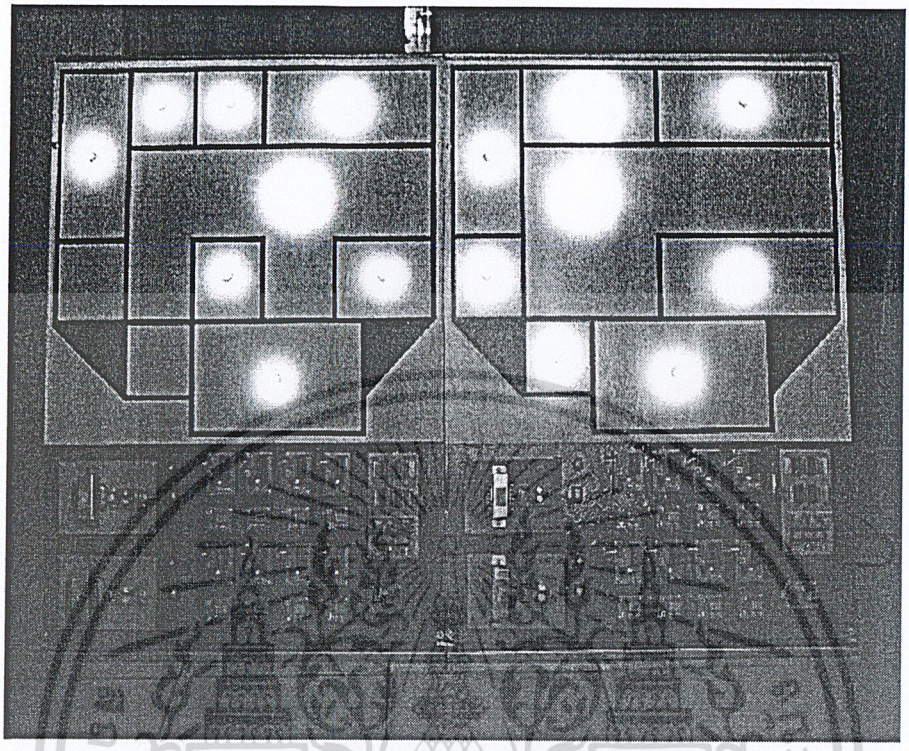
และโมเดลจำลองการควบคุมอุปกรณ์ไฟฟ้าที่ ได้แสดงดังรูปที่ 3.5



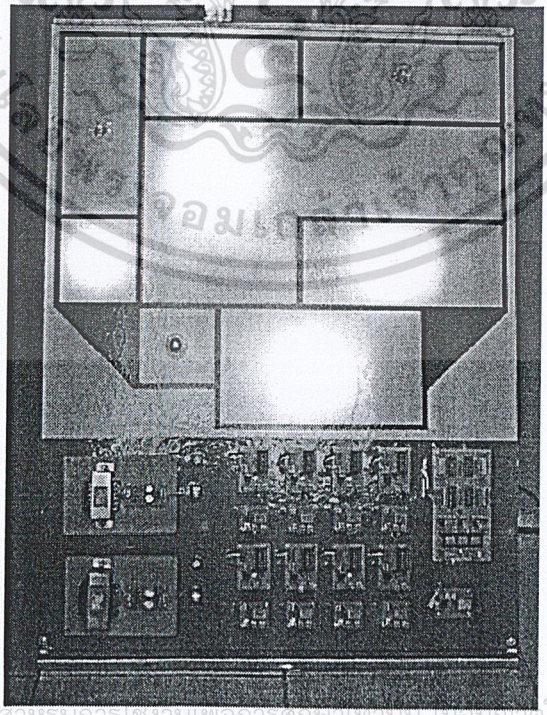
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 แสดงชุดประกอบสำเร็จรูปของวงจรที่ในการควบคุมทั้งหมด

เราได้ทำชุดจำลองอาคารสำนักงาน 2 ชั้น โดยแต่ละชั้นนั้นมีอุปกรณ์ที่ต้องการที่จะควบคุมทั้งหมด 8 ชุด (โดยแต่ละชุดเป็นโหลดให้แสงสว่าง) ซึ่งแสดงการทำงานทั้งหมดได้ดังรูปที่3.6



รูปที่ 3.6 แสดงการทำงานชุดจำลองอาคารสำนักงาน 2 ชั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 แสดงการทำงานอุปกรณ์ไฟฟ้าชั้นที่ 2 ของชุดจำลอง

จากรูปที่ 3.4 เราทำการควบคุมที่ละชั้นโดยเราจะใช้ชั้นที่ 2 ทดสอบในการปิด-เปิด อุปกรณ์ไฟฟ้าที่เราได้กำหนดไว้

เราทำการเปิดอุปการไฟฟ้า จุดที่ 1,2,3 และ5

ส่วนจุดที่ 4,6 และ 8 ทำการปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
บทสรุปและวิจารณ์ผลการทดลอง

จากผลการทดสอบอุปกรณ์ไฟฟ้าที่ได้ทำการเปิด-ปิด เราพบว่าอุปกรณ์ไฟฟ้าได้ทำงานถูกต้องตามเป้าหมายที่กำหนดไว้ซึ่งเราสังเกตุได้ตั้งแต่การทำงานของวงจรขับที่ทำหน้าที่เสมือนสวิทช์ที่สั่งให้อุปกรณ์ไฟฟ้านั้นทำงาน และในส่วนของชุดการตรวจการทำงานของอุปกรณ์ไฟฟ้านั้นได้ทำการส่งค่ากลับไปแสดงการทำงานที่หน้าจอมอนิเตอร์ต่อไป

การเดินสายและการป้อนแหล่งจ่ายไฟให้กับวงจรนั้นมีสำคัญมากในการที่จะทำให้วงจรนั้นเสถียร หากมีแหล่งจ่ายที่ให้สัญญาณ Output ที่ไม่ราบเรียบก็จะทำการส่งค่ามาควบคุมวงจรนั้นไม่เสถียร

ในส่วนของฮาร์ดแวร์ที่ได้ออกแบบไว้นั้นจะสามารถใช้กับอุปกรณ์ไฟฟ้าที่ต้องกำลังไฟฟ้าไม่เกิน 220 วัตต์ และหลอดที่ใช้ในวงจรการทดลองเป็นหลอดแรงเทียน แต่ถ้าใช้หลอดฟลูออเรสเซนต์จะต้องทำการปรับปรุงโปรแกรมใหม่ เพราะหลอดฟลูออเรสเซนต์ต้องใช้ช่วงเวลาในเริ่มทำงาน ทำให้การส่งข้อมูลมาตรวจสอบสถานะนั้นผิดพลาดได้

เนื่องจากการทดสอบที่ได้ลองกับชุดจำลองอาคารสำนักงาน 2 ชั้นนั้นได้ผลตามต้องการ แต่ในการใช้งานจริงการติดตั้งเราต้องทำการออกแบบการควบคุมไปพร้อมการก่อสร้างอาคาร ในกรณีที่เป็นอาคารเดิมที่ต้องการใช้ระบบการควบคุมแบบนี้อาจจะไม่เหมาะเนื่องจากค่าใช้จ่ายไม่คุ่มค่านั้นเอง

ในส่วนของแนวทางการพัฒนาต่อไป สามารถที่จะเพิ่มการตั้งเวลาเปิดปิด การจัดทำกราฟแสดงการทำงานในแต่ละวัน สัปดาห์ หรือเดือน ในส่วนของซอฟท์แวร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] กฤษดา ใจเย็น, อรรถพร บุญขะโภคาม , ชัยวัฒน์ ลิ้มพรจิตรวิไล, "เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม"

[2] ปรเมษฐ์ ประนยานันทน์, ปิยพงศ์ เผ่าวณิช , " ไมโครคอนโทรลเลอร์ MCS-51", กรุงเทพฯ : ซีเอ็ดยูเคชั่น , 2536

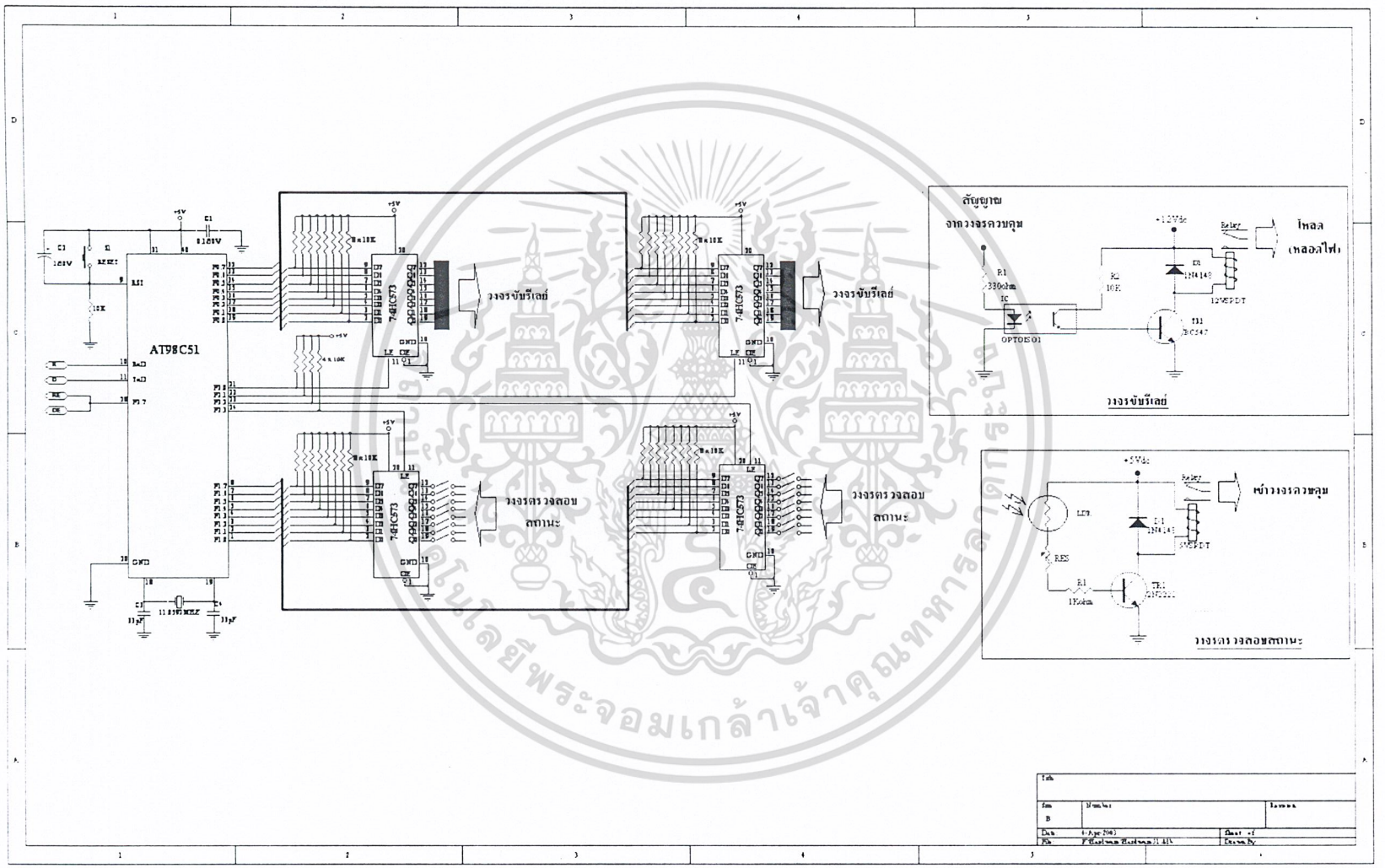
[3] สุนทร วิทูสุรพจน์, "การโปรแกรมภาษาแอสเซมบลี ของไมโครคอนโทรลเลอร์ ตระกูล 8051 ", บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

วงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


AT98C51
วงจรขับรีเลย์
วงจรขับรีเลย์
วงจรตรวจสอบ
สถานะ
วงจรตรวจสอบ
สถานะ
สัญญาณ
จากวงจรควบคุม
โหลด
(หลอดไฟ)
วงจรขับรีเลย์
เข้าวงจรควบคุม
วงจรตรวจสอบสถานะ

ภาคผนวก ข.

Microsoft Visual Basic เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

Data Sheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมวิชวลเบสิก

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

1. โปรแกรมสำหรับวินโดวส์จะนำความสามารถของวินโดวส์มาใช้ได้อย่างเต็มที่
2. การเขียนโปรแกรมสำหรับวินโดวส์ทำได้ง่ายกว่าการเขียนโปรแกรมสำหรับดอส

โปรแกรมหรือแอปพลิเคชั่นในวินโดวส์นั้นได้มีวิธีการเขียน 2 แบบ คือการเขียนโปรแกรมแบบวิชวลหรือวิชวลโปรแกรมมิ่ง (Visual Programming) และการเขียนโปรแกรมแบบนอนวิชวลหรือนอนวิชวลโปรแกรมมิ่ง (None Visual Programming) "วิชวล หมายถึง มองเห็นการเขียนโปรแกรม หมายความว่า การเขียนแบบนี้เราจะมองเห็นส่วนประกอบต่าง ๆ ของโปรแกรมนั้นทันที" ส่วนการเขียนโปรแกรมแบบนินวิชวลเป็นการเขียนโปรแกรมด้วยอักษรและเครื่องหมาย ซึ่งจะยังไม่เห็นส่วนประกอบของโปรแกรมในทันทีแต่สามารถจะมองเห็นส่วนประกอบต่าง ๆ ของโปรแกรมได้ในภายหลังคือตอนรันโปรแกรม ดังนั้นการเขียนโปรแกรมแบบวิชวลจึงช่วยให้ผู้เขียนโปรแกรมสามารถเขียนได้สะดวกและรวดเร็วกว่าในการทำงานด้านโปรแกรมมิ่งนั้นนับตั้งแต่เริ่มต้นในการพัฒนาโปรแกรมต่าง ๆ นั้นจะมีโปรแกรมภาษาต่าง ๆ มากมายที่ใช้ในการพัฒนางานหรือสร้างเป็นโปรแกรมประยุกต์ขึ้นมาใช้งานสำหรับการทำงานในองค์กรหรือพัฒนาขึ้นมาใช้งานส่วนตัว แต่โดยส่วนใหญ่แล้วโปรแกรมเหล่านี้จะเป็นโปรแกรมที่ต้องใช้ความจำเป็นเลิศเกือบทั้งสิ้น เนื่องจากโปรแกรมเหล่านั้นโดยส่วนใหญ่จะเป็นโปรแกรมประเภทการเขียนโดยการขี้คำสั่งเฉพาะต่าง ๆ ในการสร้างหรือควบคุมการทำงานของส่วนต่าง ๆ ที่ต้องการ โดยเรียกการใช้งานโปรแกรมเหล่านี้ว่า "การโค๊ดดิ้ง" แต่สำหรับโปรแกรม Visual Basic 6.0 นั้น โปรแกรมพัฒนาที่การผสมผสานกันระหว่างการโค๊ดดิ้งและจับวาง (แกรกแอนด์ดรอป) นั่นก็คือในการทำงานนั้นสามารถที่จะกำหนดหรือสร้าง Object ต่าง ๆ โดยการใช้เครื่องมือต่าง ๆ ที่ตัวโปรแกรมมีมาให้โดยไม่ต้องเขียนคำสั่งเพื่อสร้าง Object ต่าง ๆ เหล่านั้นขึ้นมาใช้งานและยังสามารถที่จะเขียนคำสั่งเพื่อใช้ในการสร้างเงื่อนไขพิเศษอื่น ๆ ที่ใช้ในการทำงานได้อีกด้วย

1.การเปิดโปรแกรมขึ้นมาใช้งาน

ในการเปิดโปรแกรมขึ้นมาใช้งานนั้นสามารถที่จะทำได้โดยวิธีการดังต่อไปนี้

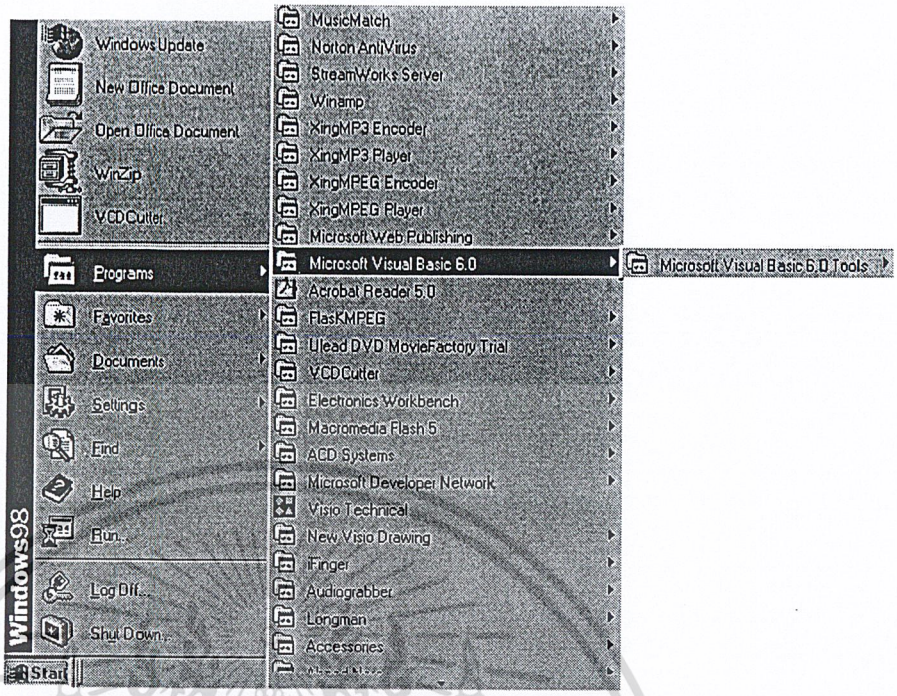
1.1 คลิกปุ่ม

1.2 เลือกเมนู Programs

1.3 เลือกเมนูย่อย Microsoft Visual Studio 6.0

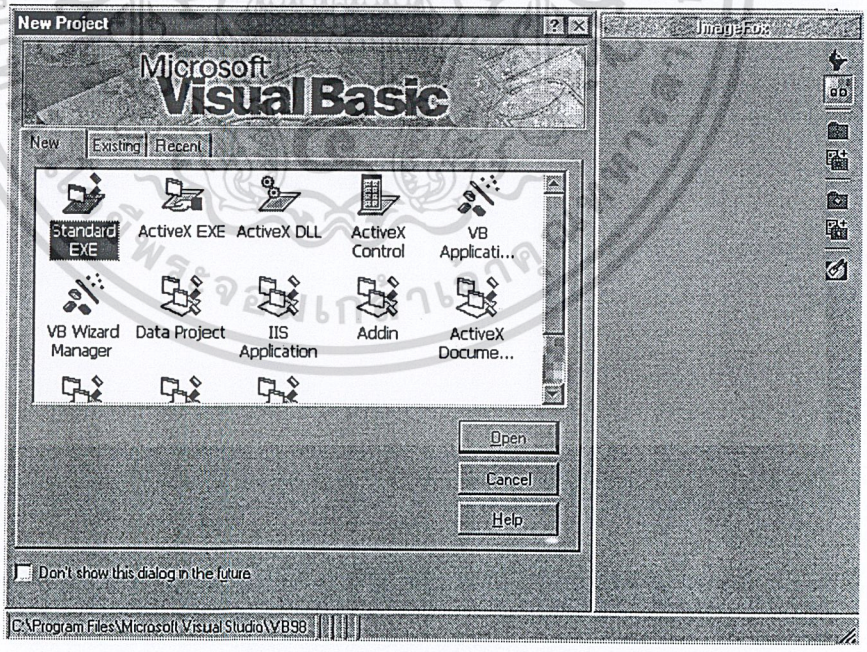
1.4 คลิกไอคอน Microsoft Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 ขั้นตอนเปิดโปรแกรม Microsoft Visual Basic 6.0

เมื่อได้เปิดโปรแกรม Microsoft Visual Basic 6.0 ขึ้นมาแล้วนั้นจะปรากฏดังรูป และสำหรับส่วนประกอบในการทำงานต่าง ๆ ที่ใช้ในการทำงานนั้นจะได้



รูปที่ 2 เมื่อเริ่มเปิดโปรแกรม Microsoft Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปนั้นจะเห็นว่าจะปรากฏไดอะล็อกบ๊อกซ์ New Project ขึ้นมาก่อนที่จะเข้าไปใช้งานโปรแกรมซึ่งในไดอะล็อกบ๊อกซ์ New Project นั้นจะประกอบด้วย 3 ส่วนด้วยกันคือ

New

ส่วนนี้เป็นส่วนที่ใช้ในการเลือกสร้างงานใหม่ขึ้นมา ซึ่งจะมีรูปแบบของการสร้างงานให้เลือกใช้หลากหลายรูปแบบด้วยกัน

Existing

ส่วนนี้จะเป็นส่วนที่ใช้ในการเปิดงานที่ได้สร้างขึ้นจากโปรแกรม Visual Basicในเวอร์ชันต่าง ๆ ที่มีอยู่แล้วsxzซึ่งเก็บไว้ในที่ต่าง ๆ ที่ต้องการขึ้นมาใช้งาน ได้ตามต้องการ

Recent

ในส่วนนี้จะเป็นการเลือกเปิดไฟล์ที่ได้เคยเปิดขึ้นมาใช้งานแล้วปิดลงไป ซึ่งในส่วนนี้จะปรากฏชื่อไฟล์ต่าง ๆ ที่ได้เคยเปิดขึ้นมาใช้งานแล้ว

2. คุณสมบัติและข้อดีของ Visual Basic

ในVisual Basic มีคุณสมบัติที่ดีหลายประการกล่าวคือ ในการเขียนโปรแกรมแบบเดิมนั้นจะต้องมานั่งออกแบบหน้าจอระบุตำแหน่งการแสดงผลคิดหาขั้นตอนการทำงานและอื่น ๆ จากนั้นจึงทำการเขียนโปรแกรม แต่ใน Visual Basic จะใช้หลักการของภาพและการมาองเห็น 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) การเขียนโปรแกรมแบบเดิมคือสั่งงานตามลำดับยุ่งยากมาก

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

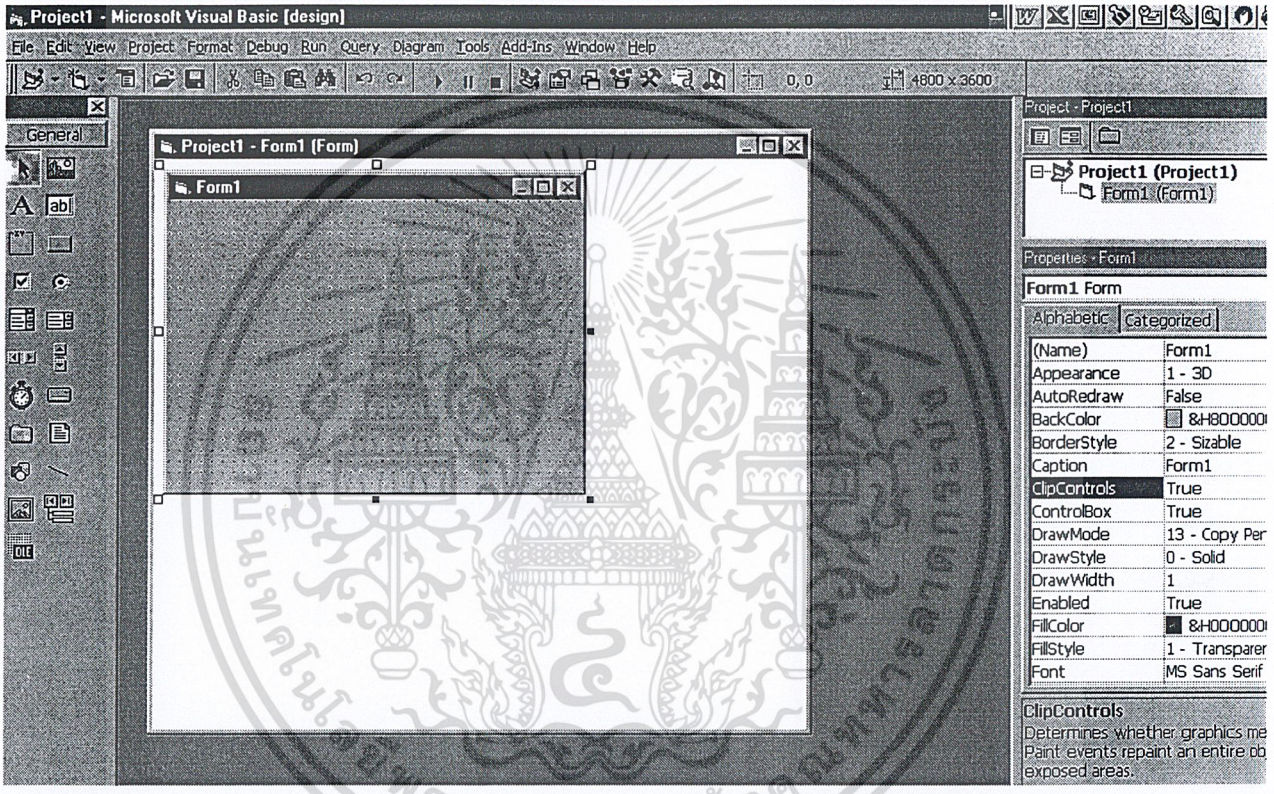
ข้อดีเหนือ Quick Basic อีกอย่างหนึ่งก็คือ การจัดการตัวแปรใน Visual Basic มีกฎเกณฑ์ซึ่งง่ายในการเข้าใจและจดจำ เพราะว่าถูกพัฒนาให้ง่ายและมีประสิทธิภาพโดยที่โปรแกรมของVisual Basicจะประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปด้วยไฟล์ 2 แบบ คือ From และ Module ยกเว้นเมื่อ Declare globally ที่อื่น ตัวแปรและค่าคงที่ใน โปรแกรมย่อยและฟังก์ชั่นนั้นจะเป็น Local สำหรับกระบวนการที่เกิดขึ้น

3. ส่วนประกอบต่าง ๆ ที่ในการทำงาน

เมื่อทำการเปิดโปรแกรมขึ้นมาใช้งานแล้วนั้นจะปรากฏส่วนประกอบต่าง ๆ ของโปรแกรมที่ควรทราบก่อนที่จะเริ่มใช้งานโปรแกรมในการสร้างงานต่าง ๆ เพราะจะทำให้เลือกใช้เครื่องมือหรืออุปกรณ์ในส่วนต่าง ๆ ได้สะดวกและรวดเร็วมากยิ่งขึ้น



รูปที่ 2.26 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

เอกสารนี้ได้รับการคุ้มครองลิขสิทธิ์ตามกฎหมาย ห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Maximize Button or Restore Button

จะเป็นปุ่มที่ใช้สำหรับคลิกเพื่อขยายขนาดใหญ่ขึ้นจนเต็มหน้าจอ

Close Button

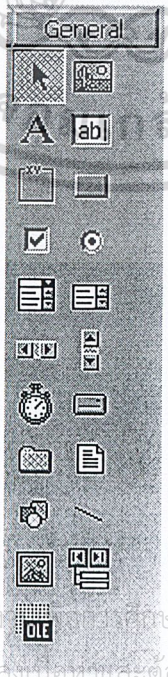
เป็นปุ่มสำหรับคลิดเพื่อปิดหรือจบการทำงานของหน้าต่างนั้น ๆ

Menu bar

เป็นแถบของเมนูหลักที่บรรจุคำสั่งย่อยหลายคำที่ใช้ในการทำงานไว้ สามารถเขียนใช้งานเมนูได้โดยการใช้เมาส์คลิกที่ชื่อของเมนูที่ต้องการบน Menu Bar หรืออาจเรียกเมนูดังกล่าวได้โดยการดกปุ่มที่คีย์บอร์ดดังนี้ กดปุ่ม Alt ค้างไว้พร้อมกับกดตัวอักษรที่ขีดเส้นใต้ของชื่อเมนูที่ต้องการเปิดตังตัวอย่างต่อไปนี้ หากต้องการใช้คำสั่งย่อยที่อยู่ในเมนู และที่ชื่อของเมนูบนจะมีเส้นขีดไว้ที่ใต้ตัวอักษร ดังนั้นในการเรียกใช้คำสั่งย่อยในเมนูดังกล่าวจำเป็นต้องเปิดเมนูก่อน โดยกดปุ่มที่คีย์บอร์ดดังนี้ Alt+F

Tool Bar

เป็นส่วนที่ปรากฏอยู่ด้านใต้ของเมนูบาร์ ซึ่งเป็นส่วนที่รวบเอาไอคอนเล็ก ๆ มากมายเอาไว้ ซึ่งแต่ไอคอนเหล่านี้ก็เปรียบเสมือนคำสั่งหนึ่ง ๆ ของเมนู ดังนั้นไอคอนในส่วนนี้จึงถูกออกแบบเพื่อให้การเลือกใช้คำสั่งของเมนูรวดเร็วและมีลักษณะที่สื่อความหมายกับผู้ใช้มากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tool Box

เป็นหน้าต่างที่รวบรวมเอาเครื่องมือต่าง ๆ สำหรับใช้ในขณะออกแบบฟอร์ม ซึ่งการรวมเอาเครื่องมือตัวใดให้กับฟอร์มผู้ใช้สามารถทำได้โดยการคลิ๊กที่คอนโทรลนั้น เพื่อเลือกแล้วนำมาวางลงในฟอร์มโดยวิธีการลากแล้ววาง หรือโดยวิธีการดับเบิลคลิ๊กที่ไอคอนคอนโทรลในทูลบ๊อกซ์ ก็ได้ ซึ่ง Visual Basicก็จะทำการวางคอนโทรลลงในฟอร์มปัจจุบันโดยอัตโนมัติ

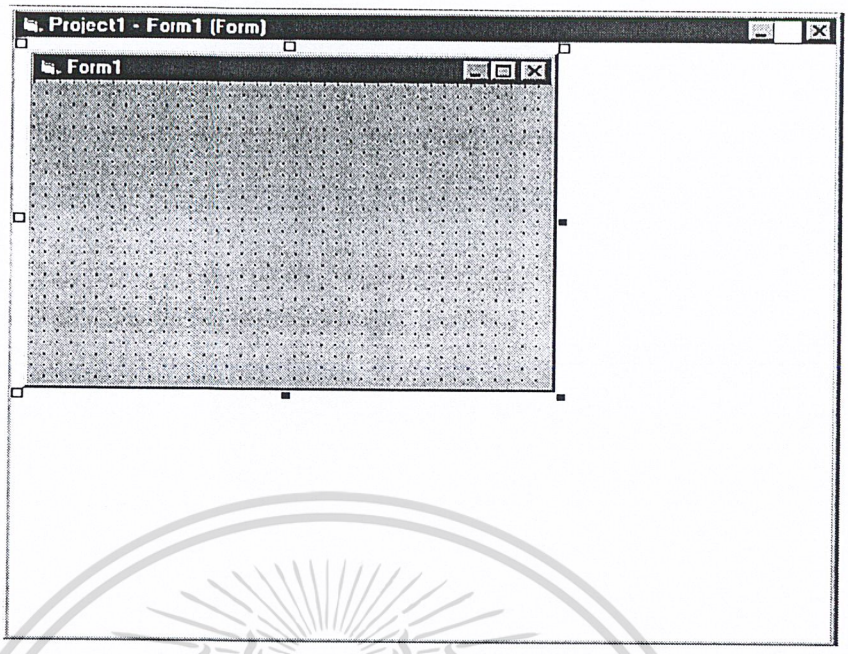
Form

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

Project Container Window

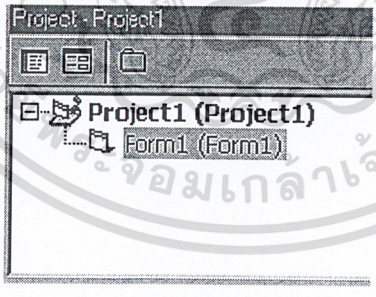
คือบริเวณทั้งหมดของฉากด้วยหลังที่เป็นสีขาว ของ From ใช้สำหรับเป็นพื้นที่หรือหน้าต่างๆ ที่ใช้บรรจุ Form หรือ สร้าง Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Project Explore Window

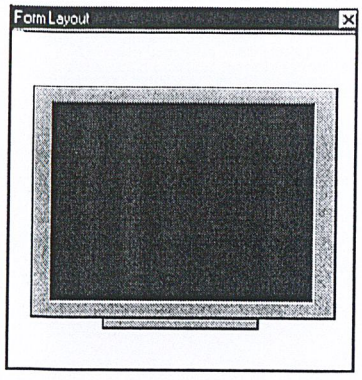
เป็นหน้าต่างที่รวบรวมรายชื่อของฟอร์ม โมดูลไฟล์ Custom Control (Class Module) หรือไฟล์ทรัพยากร (resource file) สำหรับการสร้างแอพพลิเคชันหนึ่ง ๆ ซึ่งการรวมเอาไฟล์เหล่านี้เข้ามาด้วยกันเพื่อสร้างแอพพลิเคชั่นภายใต้ Visual Basic/win เรียกว่า โปรเจค (project) หรือโครงการซึ่งโดยปกติ Visual Basic จะจัดเก็บไฟล์โปรเจคโดยจะมีนามสกุล .MAK โดยจะมีการจัดเก็บแยกออกจากไฟล์อื่น ๆ โดยเด็ดขาด



Form Layout Windows

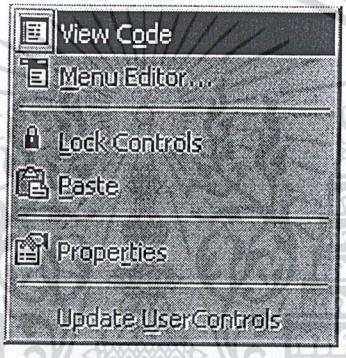
เป็นส่วนที่ใช้กำหนดตำแหน่งของ Form ที่ต้องการแสดงบนหน้าจอและบนหน้าต่างที่ทำงานนอกจากนี้ในการกำหนดตำแหน่งของ Form จะสามารถเลือกกำหนดได้จากการกำหนดค่าตัวแปรต่างๆ หรือคุณสมบัติต่าง ๆที่เกี่ยวกับการจัดวางตำแหน่งได้ในส่วนของ Properties ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pop Up Menu or Shortcut Menu

เป็นส่วนของเมนูคำสั่งต่าง ๆ ที่จะปรากฏหรือแสดงเมื่อมีการกดปุ่มด้านขวาของเมาส์หรือที่เรียกว่าคลิกขวา



Tool Tips

เป็นส่วนที่ใช้บอกหรือแสดงชื่อของเครื่องมือหรือปุ่มต่าง ๆ ที่ตำแหน่งของ Mouse Pointer ชี้ อยู่ในขณะนั้น

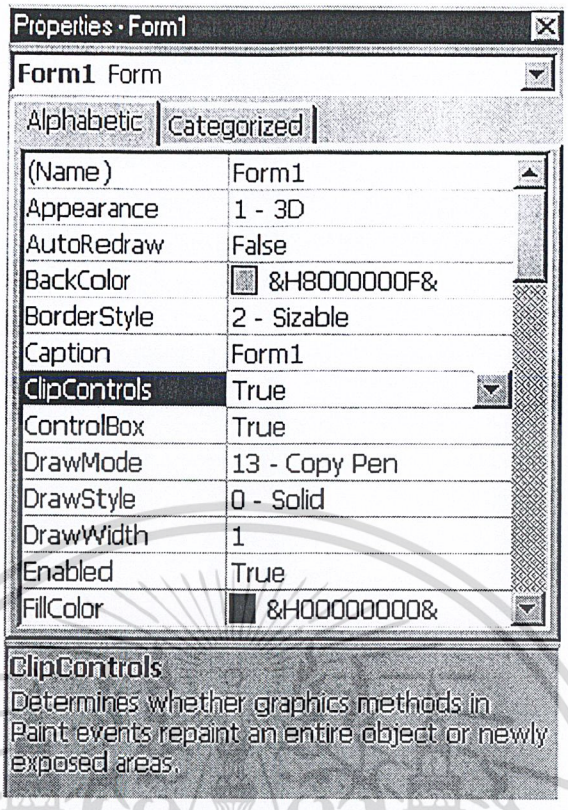
Pointer

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 ส่วนคุณสมบัติบางอย่างจะแสดงผลให้เห็นก็ต่อเมื่อผู้อ่านมีการรันแอพพลิเคชั่นเท่านั้น เช่น Enabled หรือ Visible เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4. คอนโทรลของVB/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) ที่สามารถทำงานได้เฉพาะด้านตามที่กำหนดและตัวคอนโทรดก็มีคุณสมบัติเฉพาะตัวที่ผู้อ่านสามารถแก้ไขได้นอกจากนี้เมื่อมีการกระทำใด ๆ กับคอนโทรล เช่น การคลิ๊ก การเปลี่ยนค่า เป็นต้น เหตุการณ์เกี่ยวข้องของคอนโทรลก็จะถูกเรียก (Event-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Driven) ซึ่งผู้ใช้สามารถที่จะเขียนโค้ดเพื่อควบคุมการทำงานเมื่อเกิดเหตุการณ์ได้โดยการเขียนโค้ดลงในส่วนโพรซีเดอร์เหตุการณ์ (Event Procedure) ซึ่งจะมีรายละเอียดคร่าว ๆ ของคอนโทรลแต่ละตัวดังนี้

Pointer เป็นเคอร์เซอร์ ที่ใช้สำหรับเลือก เลื่อนตำแหน่ง หรือขยายขนาดของฟอร์มหรือคอนโทรล ซึ่งคอนโทรลตัวนี้เป็นเครื่องมือของเอนไวรอนเมนท์ดังนั้นผู้ใช้จึงไม่สามารถนำไปใช้ในแอพพลิเคชั่น

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

Label ใช้แสดงข้อความที่ไม่สามารถแก้ไขได้โดยผู้ใช้ ยกเว้นแก้ไขโดยการเขียนโค้ดเท่านั้น

Command button ปุ่มคำสั่งซึ่งผู้ใช้สามารถเลือกได้โดยการคลิกที่ปุ่ม

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vertical scroll bar แถบเลื่อนในแนวตั้งซึ่งผู้ใช้สามารถกำหนดค่าในช่วงที่ต้องการเลือกได้

Time นาฬิกาที่นี้ถูกใช้ในการสร้างเหตุการณ์โดยที่อ้างอิงกับช่วงเวลา ซึ่งสามารถเทียบเคียงได้กับประโยค In Timer ของ Quick BASIC

5. ภาษาโปรแกรมของ Visual Basic

5.1 ตัวแปรและค่าคงที่

ตัวแปรใน Visual Basic มีความยาวได้ถึง 40 ตัวอักษร สามารถตั้งชื่อตัวแปรโดยผสมจากตัวอักษร ตัวเลข และ underscore (_) แต่ตัวแรกต้องเป็นอักษรเท่านั้นและห้ามตั้งชื่อตรงกับคำที่สงวนไว้ (Reserved word) ซึ่งได้แก่ ฟังชันก์และคำสั่งต่าง ๆ ตัวแปรแบ่งออกเป็น 7 ชนิด ดังตารางที่ 2.6

ชนิด	คำอธิบาย	ตัวอักษรบอกชนิด	ขอบเขต
Integer	2- byte integer	%	-32,768 จนถึง 32,767
Long	4- byte integer	&	-2,147,678จนถึง 2,147,483,647
Single	4- byte floating-point Number	!	-3.402823E38จนถึง-1.401298E-45
Double	8- byte floating-point Number	#	-1.797693134862232222D308จนถึง —4.94065645841247D-324 (ค่าลบ) 4.94065645841247D-324 จนถึง 1.797693134862232222D308
Currency	8- byte Number with Fixed decimal point	@	-922337203668547705808จนถึง 92233720366854770 5808
String	String of characters	$	0 จนถึงประมาณ65,000 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

ตารางที่ 1 ตัวแปรและค่าคงที่ในโปรแกรม Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Integer

ไว้สำหรับเก็บค่าจำนวนเต็มในช่วง –32,768 ถึง 32,767 ตัวแปรชนิดนี้สามารถคำนวณได้เร็วมาก แต่มีข้อจำกัดด้านค่าที่เก็บได้ซึ่งอยู่ในขอบเขตแคบกว่า แบบอื่นตัวแปรชนิดนี้ระบุด้วย % 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 –1.401298ฎ-45) ใช้ ! ในการระบุชนิด เช่น A!

Double

คล้ายกับชนิด Single แต่สามารถเก็บค่าได้มากกว่าและใช้เนื้อที่เก็บเป็น 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

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

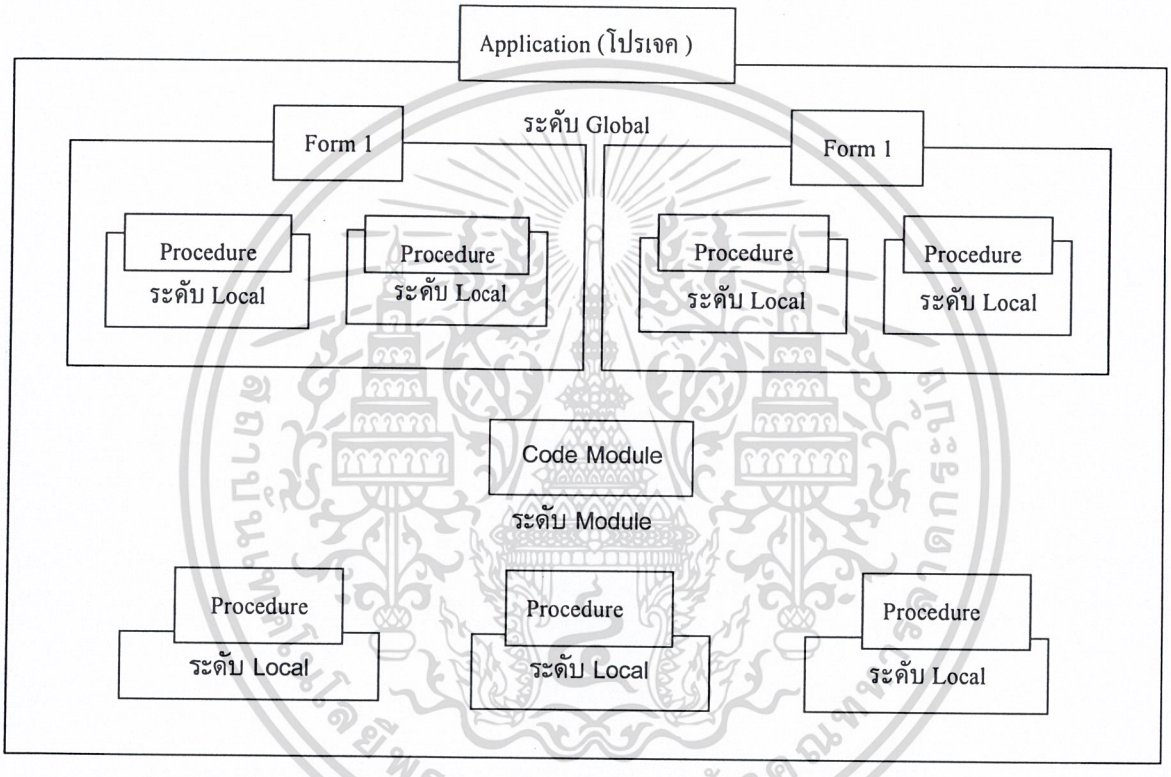
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2. ขอบเขตของตัวแปร

ทั้งตัวแปรและค่าคงที่ที่ได้กล่าวถึงตอนต้น โดยปกติจะมีขอบเขตของการใช้งานหรือ ใช้งานได้ในแต่ละส่วนของโปรแกรมเท่านั้น ซึ่งขอบเขตของตัวแปร (Scope) นี้แบ่งออกได้เป็น 3 ระดับ คือ

5.2.1 Local

ตัวแปรชนิดนี้จะเป็นที่รู้จักกันมาก ใช้ได้ในระดับโพรซีเดอร์หรือฟังก์ชันที่ประกาศใช้ตัวแปรนั่นเอง ส่วนมากมักจะใช้กับการประกาศตัวแปรสำหรับการคำนวณหร้อจะใช้เก็บค่าชั่วคราว ตัวแปรในระดับ Local จะถูกยกเลิกไปเมื่อการทำงานของโพรซีเดอร์หรือฟังก์ชันนั้นสิ้นสุดลง



รูปที่ 7 แสดงขอบเขตของการใช้ตัวแปรและค่าคงที่ในระดับต่าง ๆ

5.2.2 Module

ตัวแปรในระดับโมดูลสามารถอ้างอิงถึง สามารถใช้ร่วมกันได้หลายฟังก์ชันหรือหลายโพรซีเดอร์ซึ่งอยู่ในโมดูลนั้น ไม่ว่าจะเป็นในโมดูลของโปรแกรมที่ประกอบด้วยรูทีนย่อย ๆ หลาย ๆ รูทีน หรือในโมดูลของฟอร์มซึ่งอยู่ในรูทีนสำหรับทำงานกับ Object ในฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 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
```

5.3.โอเปอร์เรเตอร์

โอเปอเรเตอร์ คือสิงที่จะก่อให้การกระทำเพื่อให้ได้ผลลัพธ์ออกมาพบโอเปอร์เรเตอร์ได้ในนิพจน์ทั่ว ๆ ไป ซึ่งแบ่งออกเป็นกลุ่มโอเปอร์เรเตอร์ทางคณิตศาสตร์ การเปรียบเทียบ และทางตรรกะ ดังตารางที่ 2.7 โดยสัญลักษณ์ในวงเล็บคือ โอเปอร์เรเตอร์ที่ใช้ในนิพจน์

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

ตารางที่ 2.7 แสดงกลุ่มของโอเปอร์เรเตอร์

5.4. ประโยคคำสั่ง

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.1 ประโยคสำหรับกำหนดค่า

ใช้สัญลักษณ์ " =" เพื่อเป็นการกำหนดค่าทางขวามือของเครื่องหมายให้กับตัวแปรหรือคุณสมบัติ (Property) ทางซ้ายมือ เช่น

A = 5

Text1.Text = "Hi!"

5.4.2 ประโยคหมายเหตุ

สำหรับการใส่หมายเหตุต่อคำอธิบายการทำงานของโปรแกรม ซึ่งการใส่หมายเหตุในโปรแกรมนี้ถือว่าเป็นเรื่องสำคัญเนื่องจากสิ่งนี้จะช่วยป้องกันการหลงลืมของคนเขียนโปรแกรมเองหรืออธิบายให้ผู้อ่านซึ่งต้องมาใส่การทำงานของโปรแกรมได้ง่ายขึ้น 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

5.4.3 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]

5.5 ประโยคควบคุมลำดับการทำงาน

แบ่งออกเป็นหลายกลุ่มย่อยมีทั้งประโยคตัดสินใจ ประโยคการทำงานวนรอบ (loop) การกระโดดแบบมีเงื่อนไข รวมถึงคำสั่งจบการทำงานของโปรแกรมทั้งแบบชั่วคราวและถาวร ซึ่งจะขอกล่าวเรียงตามลำดับดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 IF...Then...Else

เป็นคำสั่งตัดสินใจว่าจะทำงานหรือไม่ตามเงื่อนไขที่กำหนด สามารถกำหนดเพิ่มได้ว่าหากไม่เป็นไปตามเงื่อนไขให้ทำงานใด ซึ่งอยู่หลังคำว่า "Else" อีกทั้งยังกำหนดให้ตรวจสอบหลายเงื่อนไขได้ด้วย "Else"

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

5.5.2 Select Case

ในบางครั้งการเขียนประโยคตัดสินใจด้วย IF...Then...Else จะทำให้โปรแกรมเยิ่นเย้อ และอ่านยาก 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
```

5.5.3 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

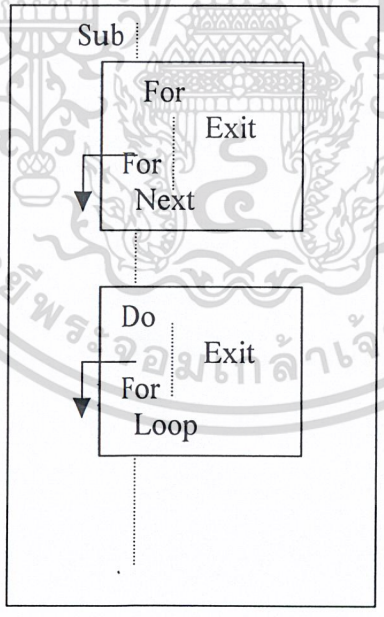
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.4 Do...Loop

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

```
Do [{While/Unit} condition]
        [Statementblock]
Loop
หรือ
Do
        [Statementblock]
Loop [{While/Unit} condition]
```

5.5.5 หยุดการทำงานของโปรแกรม



รูปที่ 2.28 แสดงการทำงานจากส่วนของโปรแกรมแบบต่าง ๆ ด้วยคำสั่ง Exit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2.1.5 การติดต่ออินเตอร์เน็ตโดยใช้ MS Winsock Control

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

MS Winsock Control เป็นControl ที่จะทำให้การใช้งาน TCP และ UDP ได้ง่ายขึ้น ซึ่งControl ตัวนี้สามารถที่จะใช้ได้ทั้ง MS Visual Basic, MS Visual C++, MS Visual FoxPro หรือแม้แต่ MS Access ซึ่งทำให้เราสามารถที่จะสร้าง Client หรือ Server Application ได้โดยที่*ไม่จำเป็นต้องทราบรายละเอียด*เกี่ยวกับ TCP หรือวิธีที่จะทำการเรียกใช้ Windows API ระดับล่างเลยเราเพียงแค่กำหนด Properties ต่างๆ แล้วเรียกใช้ Method เหล่านั้นเราก็จะสามารถที่จะสร้างการติดต่อสื่อสารกับเครื่องอื่นๆแล้วแลกเปลี่ยนข้อมูลข่าวสารซึ่งกันและกัน

คุณสมบัติของ Winsock Control (Properties)

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

ตาราง 2.4 คุณสมบัติวินซ็อกคอนโทรล

ชื่อ Property	*Return Value*	อ่านได้อย่างเดียว?	*คำอธิบาย*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

คุณลักษณะสถานะ(State Property)

คุณลักษณะสถานะนี้ จะส่งค่าสถานะของวินซ็อกเป็นค่าๆออกมาว่าขณะนั้นวินซ็อกตัวนั้นอยู่ในสถานะใด อาจจะเอาสถานะตัวนี้ไปเป็นเงื่อนไขในการเขียน โปรแกรมเพื่อป้องกันการเกิดการผิดพลาดที่เกี่ยวกับข้อบังคับของวินซ็อกได้ หรือจะใช้เพื่อเช็คดูว่าขณะนี้กำลังทำอะไรอยู่เพื่อที่เราจะได้เขียนโปรแกรมได้ว่าถ้าเป็นอย่างนี้อยู่จะทำอะไรได้บ้าง ซึ่งโดยที่รายละเอียดทั้งหมดจะระบุอยู่ที่ตารางนี้

ตาราง 2.5 คุณลักษณะสถานะ

ค่าConstant	***ค่า***	**ความหมาย**
SckClosed	0	ปิด เป็นค่าปกติ
SckOpen	1	เปิด ซ็อกเกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 นี้จะเป็นวิธีการที่จะให้วินซ็อกทำตามความต้องการของเราในการเขียนโปรแกรมการกำหนดก็จะมีลักษณะที่จะขึ้นต้นด้วยชื่อของวินซ็อกแล้วตามด้วยวิธีที่จะให้วินซ็อกทำ เช่น ต้องการให้วินซ็อกที่ชื่อ winsock1 สร้างการเชื่อมต่อ ก็จะใช้คำสั่งดังนี้ Winsock1.connect

การที่จะใช้ Methods นี้ส่วนใหญ่จะต้องมีการกำหนด คุณสมบัติ ให้กับวินซ็อกก่อนจึงต้องระวังด้วยว่าเราได้กำหนดให้มันหรือยัง ตัวอย่างก็เช่นการที่เราจะใช้ Winsock1.connect ได้นั้นต้องมีการกำหนดหมายเลขIP และ หมายเลข Port ปลายทางให้กับ วินซ็อกก่อน เป็นต้น ซึ่งรายละเอียดต่างๆของ Method ก็อยู่ที่ตารางข้างล่าง

Method	**พารามิเตอร์**	***Return Value***	***คำอธิบาย***
Accept	*RequestID*	Void	สำหรับการติดต่อแบบ TCP เท่านั้นใช้ Method 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 หรือ ตัวอักษรที่รับเข้ามา *GetData* จะข้ามพารามิเตอร์ *MaxLen* สำหรับชนิดอื่นๆที่มากกว่า ไบต์อะเรย์ และ string.
Listen	None	Void	สร้าง ซ็อกเก็ตและวางมันไว้ในโหมดการรอรับการติดต่อ *Listen* จะใช้ สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			การติดต่อแบบ 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

ตาราง 2.6 Method ของวินซ็อกคอนโทรล

Events ของวินซ็อก

Event คือเหตุการณ์ที่เกิดขึ้นเมื่อมีลักษณะของซ็อกเก็ตจากภายในและภายนอกเปลี่ยนไปจากเดิม เช่นการเข้ามาของข้อมูล การทำการติดต่อสำเร็จ

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

ตาราง 2.7 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 อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-1298; Rev 6; 12/02

MAXIM

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

General Description

The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E +3.0V-powered EIA/TIA-232 and V.28/V.24 communications interface devices feature low power consumption, high data-rate capabilities, and enhanced electrostatic-discharge (ESD) protection. The enhanced ESD structure protects all transmitter outputs and receiver inputs to ±15kV using IEC 1000-4-2 Air-Gap Discharge, ±8kV using IEC 1000-4-2 Contact Discharge (±9kV for MAX3246E), and ±15kV using the Human Body Model. The logic and receiver I/O pins of the MAX3237E are protected to the above standards, while the transmitter output pins are protected to ±15kV using the Human Body Model.

A proprietary low-dropout transmitter output stage delivers true RS-232 performance from a +3.0V to +5.5V power supply, using an internal dual charge pump. The charge pump requires only four small 0.1µF capacitors for operation from a +3.3V supply. Each device guarantees operation at data rates of 250kbps while maintaining RS-232 output levels. The MAX3237E guarantees operation at 250kbps in the normal operating mode and 1Mbps in the MegaBaud™ operating mode, while maintaining RS-232-compliant output levels.

The MAX3222E/MAX3232E have two receivers and two transmitters. The MAX3222E features a 1µA shutdown mode that reduces power consumption in battery-powered portable systems. The MAX3222E receivers remain active in shutdown mode, allowing monitoring of external devices while consuming only 1µA of supply current. The MAX3222E and MAX3232E are pin, package, and functionally compatible with the industry-standard MAX242 and MAX232, respectively.

The MAX3241E/MAX3246E are complete serial ports (three drivers/five receivers) designed for notebook and subnotebook computers. The MAX3237E (five drivers/three receivers) is ideal for peripheral applications that require fast data transfer. These devices feature a shutdown mode in which all receivers remain active, while consuming only 1µA (MAX3241E/MAX3246E) or 10nA (MAX3237E).

The MAX3222E, MAX3232E, and MAX3241E are available in space-saving SO, SSOP, and TSSOP packages. The MAX3237E is offered in an SSOP package. The MAX3246E is offered in the ultra-small 6 x 6 UCSP™ package.

Applications

Battery-Powered Equipment
Cell Phones
Cell-Phone Data Cables
Notebook, Subnotebook, and Palmtop Computers
Printers
Smart Phones
xDSL Modems

Features

- **ESD Protection for RS-232 I/O Pins (MAX3222E/MAX3232E/MAX3241E/MAX3246E)**
  - **±15kV—Human Body Model**
  - **±8kV—IEC 1000-4-2, Contact Discharge**
  - **±9kV (MAX3246E Only)—IEC 1000-4-2, Contact Discharge**
  - **±15kV—IEC 1000-4-2, Air-Gap Discharge**
- **ESD Protection for all Logic and Receiver I/O Pins (MAX3237E)**
  - **±15kV—Human Body Model**
  - **±8kV—IEC 1000-4-2, Contact Discharge**
  - **±15kV–IEC 1000-4-2, Air-Gap Discharge**
- **ESD Protection for Transmitter Output Pins (MAX3237E)**
  - **±15kV—Human Body Model**
- **Guaranteed Data Rate**
  - **250kbps (MAX3222E/MAX3232E/MAX3241E/MAX3246E/MAX3237E, Normal Operation)**
  - **1Mbps (MAX3237E, MegaBaud Operation)**
- **Latchup Free**
- **Low-Power Shutdown with Receivers Active**
  - **1µA (MAX3222E/MAX3241E/MAX3246E)**
  - **10nA (MAX3237E)**
- **Flow-Through Pinout (MAX3237E)**
- **Guaranteed Mouse Drivability (MAX3241E)**
- **Meets EIA/TIA-232 Specifications Down to +3.0V**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX3222ECUP	0°C to +70°C	20 TSSOP
MAX3222ECAP	0°C to +70°C	20 SSOP

Ordering Information continued at end of data sheet.

Pin Configurations appear at end of data sheet.

Selector Guide appears at end of data sheet.

Typical Operating Circuits appear at end of data sheet.

MegaBaud is a trademark of Maxim Integrated Products, Inc.

UCSP is a trademark of Maxim Integrated Products, Inc.

†Covered by U.S. Patent numbers 4,636,930; 4,679,134; 4,777,577; 4,797,899; 4,809,152; 4,897,774; 4,999,761; and other patents pending.

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

ABSOLUTE MAXIMUM RATINGS

V_{CC} to GND .....-0.3V to +6V
V+ to GND (Note 1) .....-0.3V to +7V
V- to GND (Note 1) .....+0.3V to -7V
V+ + IV-I (Note 1) .....+13V
Input Voltages
T_IN, $\overline{EN}$, $\overline{SHDN}$, MBAUD to GND .....-0.3V to +6V
R_IN to GND .....±25V
Output Voltages
T_OUT to GND .....±13.2V
R_OUT, R_OUTB (MAX3241E) .....-0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_OUT to GND .....Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin SSOP (derate 7.14mW/°C above +70°C) 571mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
18-Pin PDIP (derate 11.11mW/°C above +70°C) 889mW
20-Pin TSSOP (derate 10.9mW/°C above +70°C) 879mW
20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
28-Pin SSOP (derate 9.52mW/°C above +70°C) 762mW
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
28-Pin TSSOP (derate 12.8mW/°C above +70°C) 1026mW
32-Lead QFN (derate 23.2mW/°C above +70°C) 1860mW
6 x 6 UCSP (derate 12.6mW/°C above +70°C) 1010mW
Operating Temperature Ranges
MAX32_ _EC_ _ .....0°C to +70°C
MAX32_ _EE_ _ .....-40°C to +85°C
Storage Temperature Range .....-65°C to +150°C
Lead Temperature (soldering, 10s) .....+300°C
Bump Reflow Temperature (Note 2)
Infrared, 15s .....+200°C
Vapor Phase, 20s .....+215°C

Note 1: V+ and V- can have maximum magnitudes of 7V, but their absolute difference cannot exceed 13V.

Note 2: This device is constructed using a unique set of packaging techniques that impose a limit on the thermal profile the device can be exposed to during board-level solder attach and rework. This limit permits only the use of the solder profiles recommended in the industry-standard specification, JEDEC 020A, paragraph 7.6, Table 3 for IR/VPR and convection reflow. Preheating is required. Hand or wave soldering is not allowed.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V_{CC} = +3V to +5.5V, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.) (Notes 3, 4)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
DC CHARACTERISTICS (V_{CC} = +3.3V or +5V, T_A = +25°C)						
Supply Current	$\overline{SHDN}$ = V_{CC}, no load	MAX3222E, MAX3232E, MAX3241E, MAX3246E		0.3	1	mA
		MAX3237E		0.5	2.0	
Shutdown Supply Current	$\overline{SHDN}$ = GND			1	10	µA
	$\overline{SHDN}$ = R_IN = GND, T_IN = GND or V_{CC} (MAX3237E)			10	300	nA
LOGIC INPUTS						
Input Logic Low	T_IN, $\overline{EN}$, $\overline{SHDN}$, MBAUD				0.8	V
Input Logic High	T_IN, $\overline{EN}$, $\overline{SHDN}$, MBAUD	V_{CC} = +3.3V	2.0			V
		V_{CC} = +5.0V	2.4			
Transmitter Input Hysteresis				0.5		V
Input Leakage Current	T_IN, $\overline{EN}$, $\overline{SHDN}$	MAX3222E, MAX3232E, MAX3241E, MAX3246E		±0.01	±1	µA
	T_IN, $\overline{SHDN}$, MBAUD	MAX3237E (Note 5)		9	18	
RECEIVER OUTPUTS						
Output Leakage Current	R_OUT (MAX3222E/MAX3237E/MAX3241E/ MAX3246E), $\overline{EN}$ = V_{CC}, receivers disabled			±0.05	±10	µA
Output Voltage Low	I_{OUT} = 1.6mA (MAX3222E/MAX3232E/MAX3241E/ MAX3246E), I_{OUT} = 1.0mA (MAX3237E)				0.4	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = +3V to +5.5V, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.) (Notes 3, 4)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage High	I_{OUT} = -1.0mA		V_{CC} - 0.6	V_{CC} - 0.1		V
RECEIVER INPUTS						
Input Voltage Range			-25		+25	V
Input Threshold Low	T_A = +25°C	V_{CC} = +3.3V	0.6	1.1		V
		V_{CC} = +5.0V	0.8	1.5		
Input Threshold High	T_A = +25°C	V_{CC} = +3.3V		1.5	2.4	V
		V_{CC} = +5.0V		2.0	2.4	
Input Hysteresis				0.5		V
Input Resistance	T_A = +25°C		3	5	7	kΩ
TRANSMITTER OUTPUTS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5	±5.4		V
Output Resistance	V_{CC} = 0, transmitter output = ±2V		300	50k		Ω
Output Short-Circuit Current					±60	mA
Output Leakage Current	V_{CC} = 0 or +3.0V to +5.5V, V_{OUT} = ±12V, transmitters disabled (MAX3222E/MAX3232E/MAX3241E/MAX3246E)				±25	µA
MOUSE DRIVABILITY (MAX3241E)						
Transmitter Output Voltage	T1IN = T2IN = GND, T3IN = V_{CC}, T3OUT loaded with 3kΩ to GND, T1OUT and T2OUT loaded with 2.5mA each		±5			V
ESD PROTECTION						
R_IN, T_OUT	Human Body Model			±15		kV
	IEC 1000-4-2 Air-Gap Discharge (except MAX3237E)			±15		
	IEC 1000-4-2 Contact Discharge (except MAX3237E)			±8		
	IEC 1000-4-2 Contact Discharge (MAX3246E only)			±9		
T_IN, R_IN, R_OUT, $\overline{EN}$, $\overline{SHDN}$, MBAUD	MAX3237E	Human Body Model		±15		kV
		IEC1000-4-2 Air-Gap Discharge		±15		
		IEC1000-4-2 Contact Discharge		±8		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

TIMING CHARACTERISTICS—MAX3222E/MAX3232E/MAX3241E/MAX3246E

(V_{CC} = +3V to +5.5V, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.) (Notes 3, 4)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Maximum Data Rate		R_L = 3kΩ, C_L = 1000pF, one transmitter switching	T_A = T_{MIN} to T_{MAX} (MAX3222E/MAX3232E/MAX3241E)	250			kbps
			T_A = +25°C (MAX3246E)	250			
Receiver Propagation Delay	t_{PHL}	Receiver input to receiver output, C_L = 150pF			0.15		µs
	t_{PLH}				0.15		
Receiver Output Enable Time		Normal operation (except MAX3232E)			200		ns
Receiver Output Disable Time		Normal operation (except MAX3232E)			200		ns
Transmitter Skew	$\lvert t_{PHL} - t_{PLH} \rvert$	(Note 6)			100		ns
Receiver Skew	$\lvert t_{PHL} - t_{PLH} \rvert$				50		ns
Transition-Region Slew Rate		V_{CC} = +3.3V, T_A = +25°C, R_L = 3kΩ to 7kΩ, measured from +3.0V to –3.0V or –3.0V to +3.0V, one transmitter switching	C_L = 150pF to 1000pF	6		30	V/µs

TIMING CHARACTERISTICS—MAX3237E

(V_{CC} = +3V to +5.5V, C1–C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.) (Note 3)

PARAMETER	CONDITIONS			MIN	TYP	MAX	UNITS
Maximum Data Rate	R_L = 3kΩ, C_L = 1000pF, one transmitter switching, MBAUD = GND			250			kbps
	V_{CC} = +3.0V to +4.5V, R_L = 3kΩ, C_L = 250pF, one transmitter switching, MBAUD = V_{CC}			1000			
	V_{CC} = +4.5V to +5.5V, R_L = 3kΩ, C_L = 1000pF, one transmitter switching, MBAUD = V_{CC}			1000			
Receiver Propagation Delay	R_IN to R_OUT, C_L = 150pF	t_{PHL}			0.15		µs
		t_{PLH}			0.15		
Receiver Output Enable Time	Normal operation				2.6		µs
Receiver Output Disable Time	Normal operation				2.4		
Transmitter Skew	$\lvert t_{PHL} - t_{PLH} \rvert$, MBAUD = GND (Note 6)				100		ns
	$\lvert t_{PHL} - t_{PLH} \rvert$, MBAUD = V_{CC} (Note 6)						
Receiver Skew	$\lvert t_{PHL} - t_{PLH} \rvert$				50		ns
Transition-Region Slew Rate	V_{CC} = +3.3V, R_L = 3kΩ to 7kΩ, +3.0V to –3.0V or -3.0V to +3.0V, T_A = +25°C	C_L = 150pF to 1000pF	MBAUD = GND	6		30	V/µs
			MBAUD = V_{CC}	24		150	
		C_L = 150pF to 2500pF, MBAUD = GND		4		30	

Note 3: MAX3222E/MAX3232E/MAX3241E: C1–C4 = 0.1µF tested at +3.3V ±10%; C1 = 0.047µF, C2, C3, C4 = 0.33µF tested at +5.0V ±10%. MAX3237E: C1–C4 = 0.1µF tested at +3.3V ±5%, C1–C4 = 0.22µF tested at +3.3V ±10%; C1 = 0.047µF, C2, C3, C4 = 0.33µF tested at +5.0V ±10%. MAX3246E; C1-C4 = 0.22µF tested at +3.3V ±10%; C1 = 0.22µF, C2, C3, C4 = 0.54µF tested at 5.0V ±10%.

Note 4: MAX3246E devices are production tested at +25°C. All limits are guaranteed by design over the operating temperature range.

Note 5: The MAX3237E logic inputs have an active positive feedback resistor. The input current goes to zero when the inputs are at the supply rails.

Note 6: Transmitter skew is measured at the transmitter zero crosspoints.

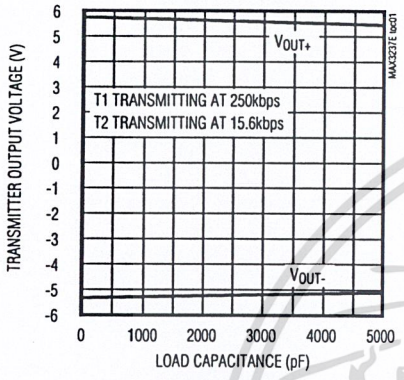
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

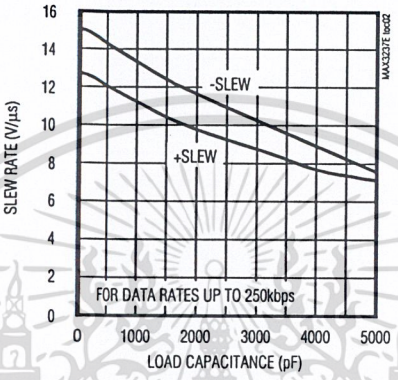
Typical Operating Characteristics

(V_{CC} = +3.3V, 250kbps data rate, 0.1µF capacitors, all transmitters loaded with 3kΩ and C_L, T_A = +25°C, unless otherwise noted.)

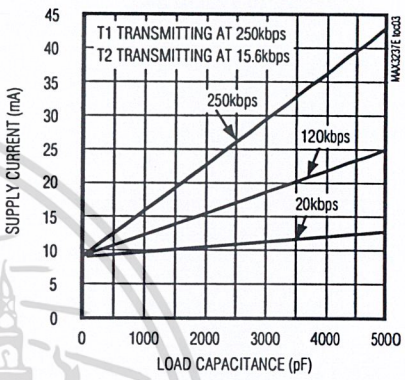
MAX3222E/MAX3232E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



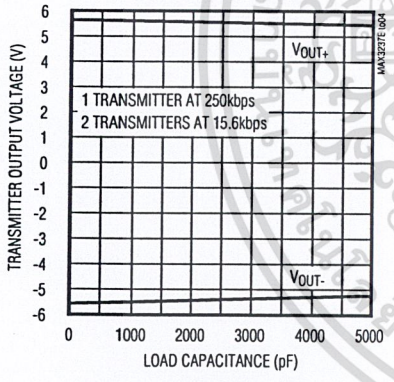
MAX3222E/MAX3232E
SLEW RATE vs. LOAD CAPACITANCE



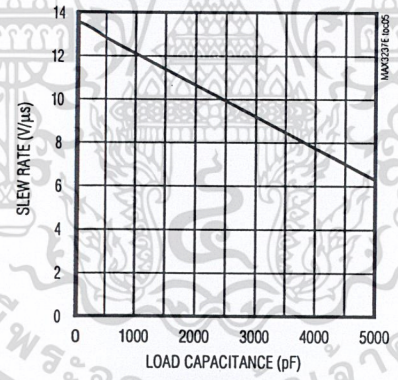
MAX3222E/MAX3232E
OPERATING SUPPLY CURRENT
vs. LOAD CAPACITANCE



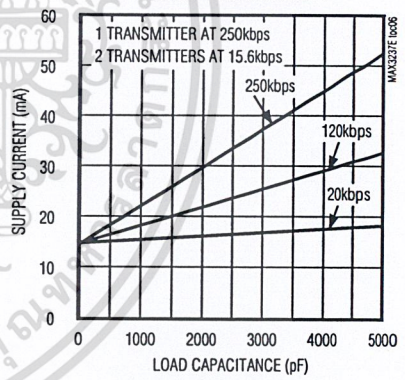
MAX3241E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



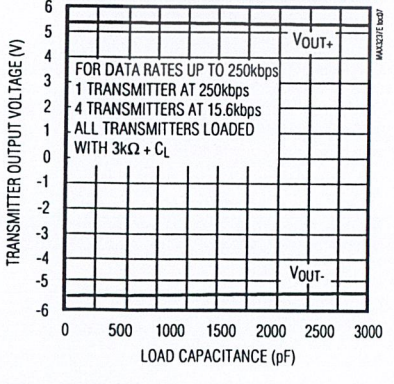
MAX3241E
SLEW RATE vs. LOAD CAPACITANCE



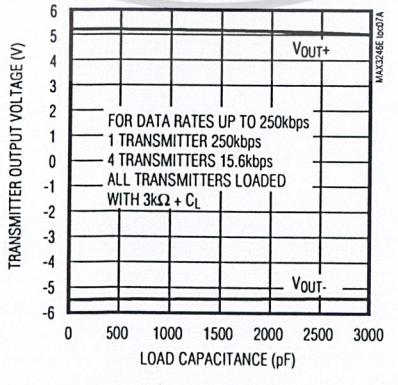
MAX3241E
OPERATING SUPPLY CURRENT
vs. LOAD CAPACITANCE



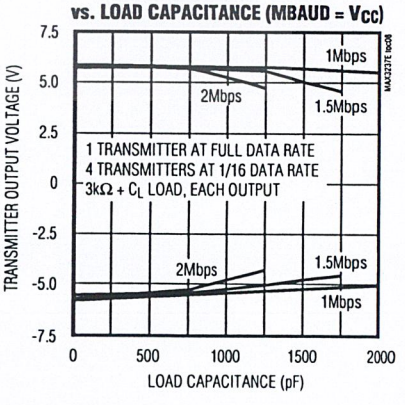
MAX3237E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE (MBAUD = GND)



MAX3237E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



MAX3237E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE (MBAUD = V_{CC})



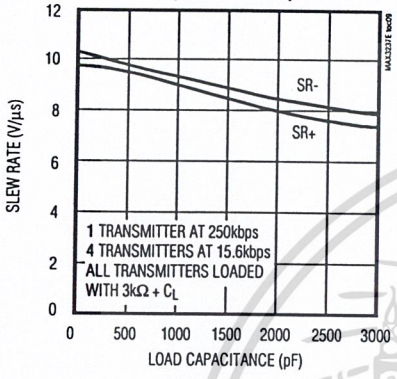
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

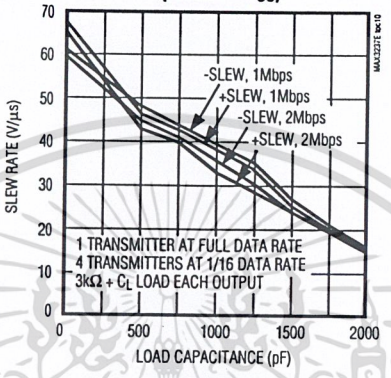
Typical Operating Characteristics (continued)

(VCC = +3.3V, 250kbps data rate, 0.1µF capacitors, all transmitters loaded with 3kΩ and CL, TA = +25°C, unless otherwise noted.)

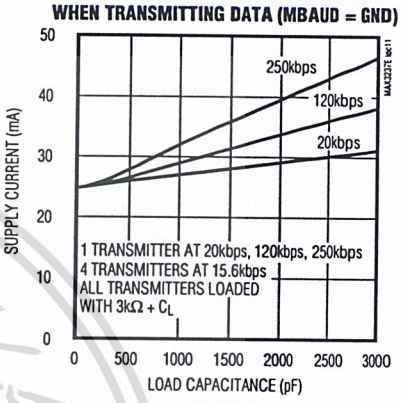
MAX3237E
SLEW RATE vs. LOAD CAPACITANCE (MBAUD = GND)



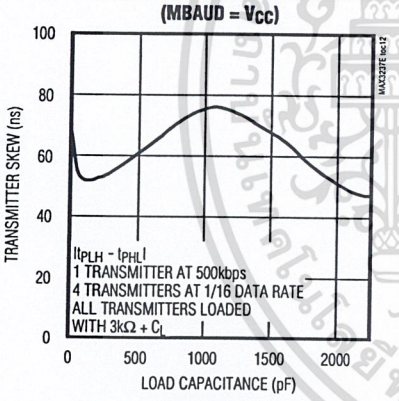
MAX3237E
SLEW RATE vs. LOAD CAPACITANCE (MBAUD = VCC)



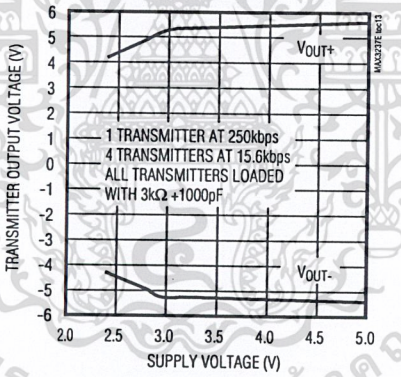
MAX3237E
SUPPLY CURRENT vs. LOAD CAPACITANCE WHEN TRANSMITTING DATA (MBAUD = GND)



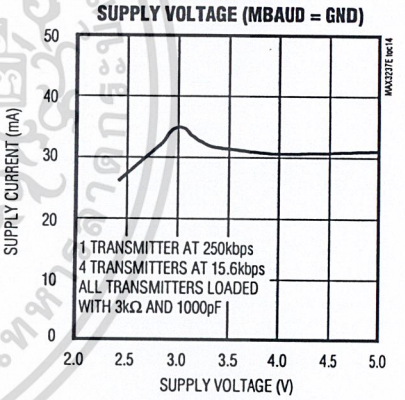
MAX3237E
TRANSMITTER SKEW vs. LOAD CAPACITANCE (MBAUD = VCC)



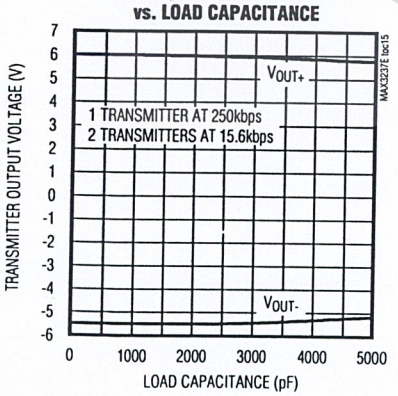
MAX3237E
TRANSMITTER OUTPUT VOLTAGE vs. SUPPLY VOLTAGE (MBAUD = GND)



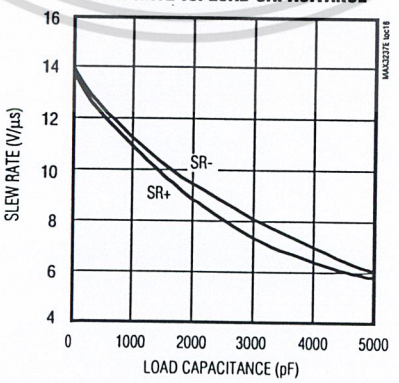
MAX3237E
SUPPLY CURRENT vs. SUPPLY VOLTAGE (MBAUD = GND)



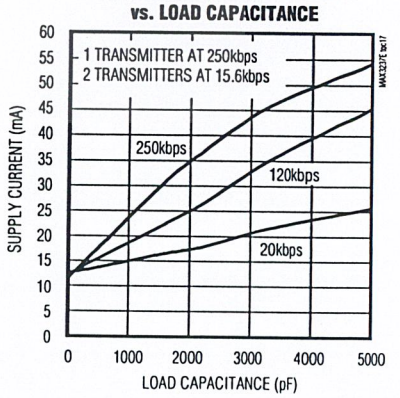
MAX3246E
TRANSMITTER OUTPUT VOLTAGE vs. LOAD CAPACITANCE



MAX3246E
SLEW RATE vs. LOAD CAPACITANCE



MAX3246E
OPERATING SUPPLY CURRENT vs. LOAD CAPACITANCE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

PIN								NAME	FUNCTION
MAX3222E		MAX3232E		MAX3237E	MAX3241E		MAX3246E		
SO/DIP	TSSOP/ SSOP	SO/DIP	TSSOP/ SSOP		SSOP/SO	QFN			
1	1	—	—	13*	23	22	B3	$\overline{EN}$	Receiver Enable. Active low.
2	2	1	2	28	28	28	F3	C1+	Positive Terminal of Voltage-Doubler Charge-Pump Capacitor
3	3	2	3	27	27	27	F1	V+	+5.5V Generated by the Charge Pump
4	4	3	4	25	24	23	F4	C1-	Negative Terminal of Voltage-Doubler Charge-Pump Capacitor
5	5	4	5	1	1	29	E1	C2+	Positive Terminal of Inverting Charge-Pump Capacitor
6	6	5	6	3	2	30	D1	C2-	Negative Terminal of Inverting Charge-Pump Capacitor
7	7	6	7	4	3	31	C1	V-	-5.5V Generated by the Charge Pump
8, 15	8, 17	7, 14	8, 17	5, 6, 7, 10, 12	9, 10, 11	6, 7, 8	F6, E6, D6	T_OUT	RS-232 Transmitter Outputs
9, 14	9, 16	8, 13	9, 16	8, 9, 11	4–8	1–5	A4, A5, A6, B6, C6	R_IN	RS-232 Receiver Inputs
10, 13	10, 15	9, 12	12, 15	18, 20, 21	15–19	13, 14, 15, 17, 18	C2, B1, A1, A2, A3	R_OUT	TTL/CMOS Receiver Outputs
11, 12	12, 13	10, 11	13, 14	17*, 19*, 22*, 23*, 24*	12, 13, 14	10, 11, 12	E3, E2, D2	T_IN	TTL/CMOS Transmitter Inputs
16	18	15	18	2	25	24	F5	GND	Ground
17	19	16	19	26	26	26	F2	V_{CC}	+3.0V to +5.5V Supply Voltage
18	20	—	—	14*	22	21	B2	$\overline{SHDN}$	Shutdown Control. Active low.
—	11, 14	—	1, 10, 11, 20	—	—	9, 16, 25, 32	C3, D3, B4, C4, D4, E4, B5, C5, D5, E5	N.C.	No Connection. For MAX3246E, these locations are not populated with solder bumps.
—	—	—	—	15*	—	—	—	MBAUD	MegaBaud Control Input. Connect to GND for normal operation; connect to V_{CC} for 1Mbps transmission rates.
—	—	—	—	16	20, 21	19, 20	—	R_OUTB	Noninverting Complementary Receiver Outputs. Always active.

**These pins have an active positive feedback resistor internal to the MAX3237E, allowing unused inputs to be left unconnected.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

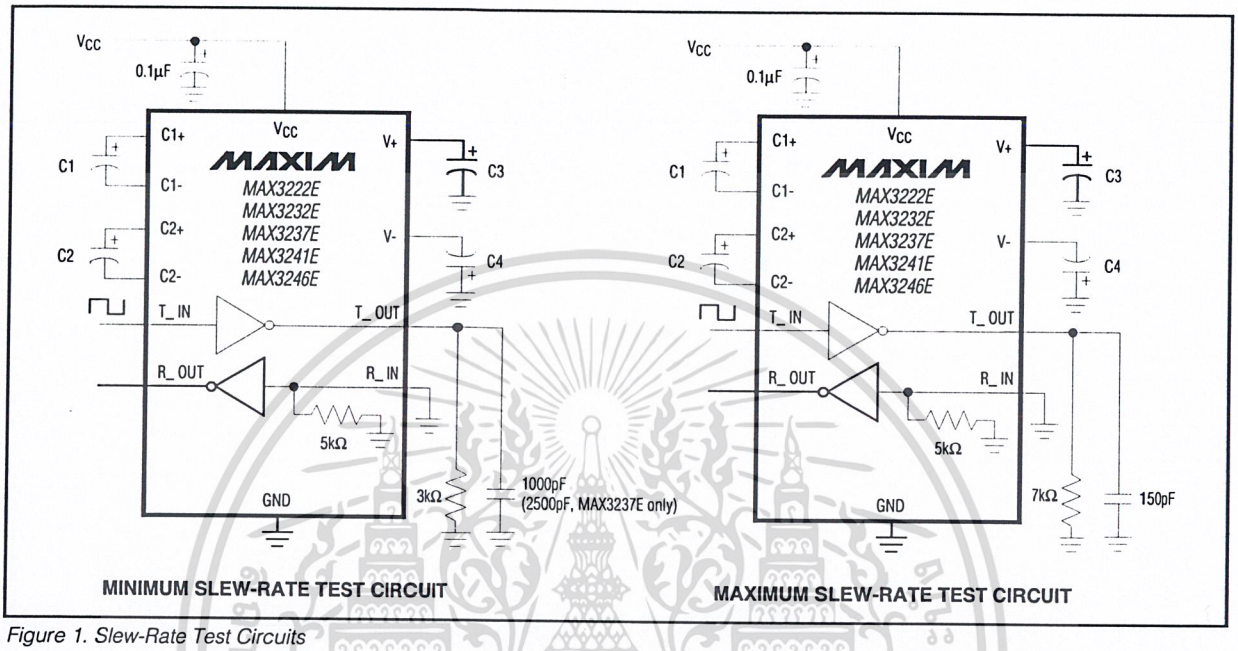


Figure 1. Slew-Rate Test Circuits

Detailed Description

Dual Charge-Pump Voltage Converter

The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246Es' internal power supply consists of a regulated dual charge pump that provides output voltages of +5.5V (doubling charge pump) and -5.5V (inverting charge pump) over the +3.0V to +5.5V V_{CC} range. The charge pump operates in discontinuous mode; if the output voltages are less than 5.5V, the charge pump is enabled, and if the output voltages exceed 5.5V, the charge pump is disabled. Each charge pump requires a flying capacitor (C1, C2) and a reservoir capacitor (C3, C4) to generate the V+ and V- supplies (Figure 1).

RS-232 Transmitters

The transmitters are inverting level translators that convert TTL/CMOS-logic levels to ±5V EIA/TIA-232-compliant levels.

The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E transmitters guarantee a 250kbps data rate with worst-case loads of 3kΩ in parallel with 1000pF, providing compatibility with PC-to-PC communication software (such as LapLink™). Transmitters can be paralleled to drive multiple receivers or mice.

The MAX3222E/MAX3237E/MAX3241E/MAX3246E transmitters are disabled and the outputs are forced into a high-impedance state when the device is in shutdown mode (SHDN = GND). The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E permit the outputs to be driven up to ±12V in shutdown.

The MAX3222E/MAX3232E/MAX3241E/MAX3246E transmitter inputs do not have pullup resistors. Connect unused inputs to GND or V_{CC}. The MAX3237E's transmitter inputs have a 400kΩ active positive-feedback resistor, allowing unused inputs to be left unconnected.

MAX3237E MegaBaud Operation

For higher-speed serial communications, the MAX3237E features MegaBaud operation. In MegaBaud operating mode (MBAUD = V_{CC}), the MAX3237E transmitters guarantee a 1Mbps data rate with worst-case loads of 3kΩ in parallel with 250pF for +3.0V < V_{CC} < +4.5V. For +5V ±10% operation, the MAX3237E transmitters guarantee a 1Mbps data rate into worst-case loads of 3kΩ in parallel with 1000pF.

RS-232 Receivers

The receivers convert RS-232 signals to CMOS-logic output levels. The MAX3222E/MAX3237E/MAX3241E/MAX3246E receivers have inverting three-state outputs. Drive $\overline{EN}$ high to place the receiver(s) into a high-impedance state. Receivers can be either active or inactive in shutdown (Table 1).

LapLink is a trademark of Traveling Software.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

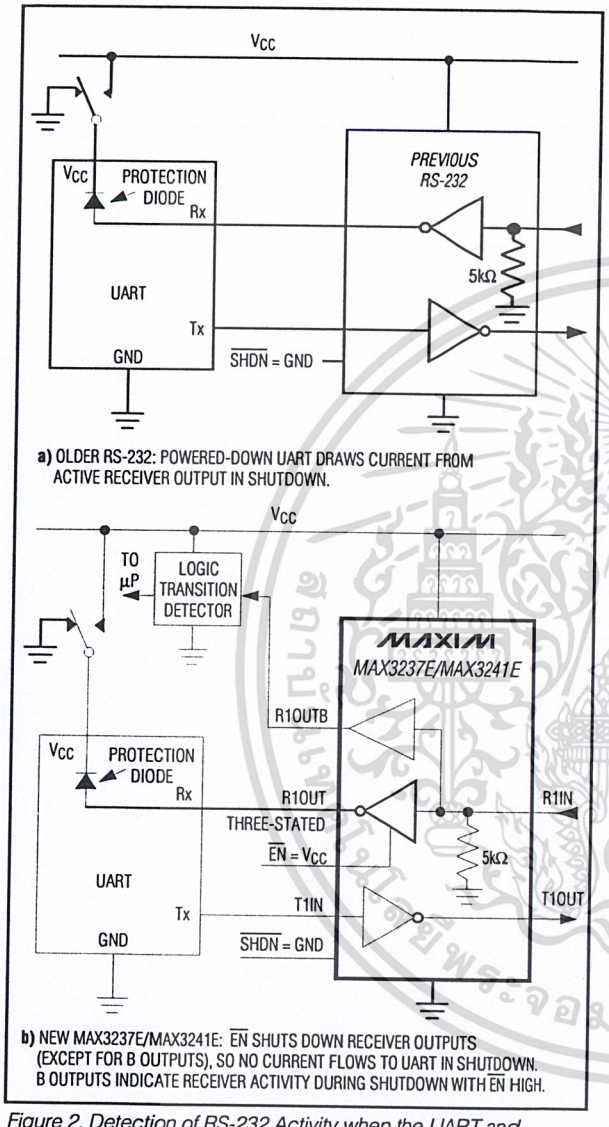


Figure 2. Detection of RS-232 Activity when the UART and Interface are Shut Down; Comparison of MAX3237E/MAX3241E (b) with Previous Transceivers (a)

The complementary outputs on the MAX3237E/MAX3241E (R_OUTB) are always active, regardless of the state of $\overline{EN}$ or $\overline{SHDN}$. This allows the device to be used for ring indicator applications without forward biasing other devices connected to the receiver outputs. This is ideal for systems where V_{CC} drops to zero in shutdown to accommodate peripherals such as UARTs (Figure 2).

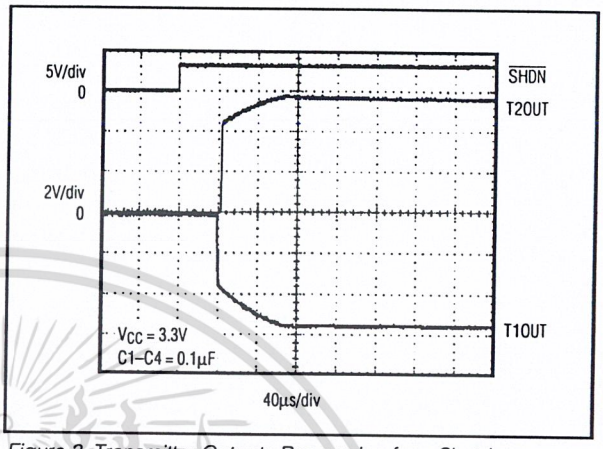


Figure 3. Transmitter Outputs Recovering from Shutdown or Powering Up

MAX3222E/MAX3237E/MAX3241E/ MAX3246E Shutdown Mode

Supply current falls to less than 1µA in shutdown mode ($\overline{SHDN}$ = low). The MAX3237E's supply current falls to10nA (typ) when all receiver inputs are in the invalid range (-0.3V < R_IN < +0.3). When shut down, the device's charge pumps are shut off, V+ is pulled down to V_{CC}, V- is pulled to ground, and the transmitter outputs are disabled (high impedance). The time required to recover from shutdown is typically 100µs, as shown in Figure 3. Connect $\overline{SHDN}$ to V_{CC} if shutdown mode is not used. $\overline{SHDN}$ has no effect on R_OUT or R_OUTB (MAX3237E/MAX3241E).

±15kV ESD Protection

As with all Maxim devices, ESD-protection structures are incorporated to protect against electrostatic discharges encountered during handling and assembly. The driver outputs and receiver inputs of the MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E have extra protection against static electricity. Maxim's engineers have developed state-of-the-art structures to protect these pins against ESD of ±15kV without damage. The ESD structures withstand high ESD in all states: normal operation, shutdown, and powered down. After an ESD event, Maxim's E versions keep working without latchup, whereas competing RS-232 products can latch and must be powered down to remove latchup.

Furthermore, the MAX3237E logic I/O pins also have ±15kV ESD protection. Protecting the logic I/O pins to ±15kV makes the MAX3237E ideal for data cable applications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

Table 1. MAX3222E/MAX3237E/MAX3241E/ MAX3246E Shutdown and Enable Control Truth Table

$\overline{SHDN}$	$\overline{EN}$	T_OUT	R_OUT	R_OUTB (MAX3237E/ MAX3241E)
0	0	High-Z	Active	Active
0	1	High-Z	High-Z	Active
1	0	Active	Active	Active
1	1	Active	High-Z	Active

ESD protection can be tested in various ways; the transmitter outputs and receiver inputs for the MAX3222E/MAX3232E/MAX3241E/MAX3246E are characterized for protection to the following limits:

- ±15kV using the Human Body Model
- ±8kV using the Contact Discharge method specified in IEC 1000-4-2
- ±9kV (MAX3246E only) using the Contact Discharge method specified in IEC 1000-4-2
- ±15kV using the Air-Gap Discharge method specified in IEC 1000-4-2

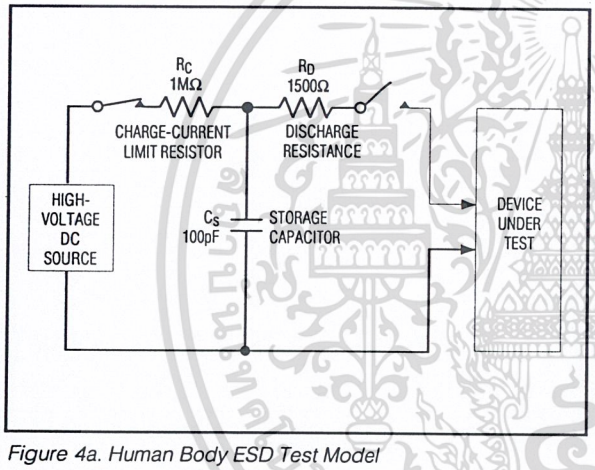


Figure 4a. Human Body ESD Test Model

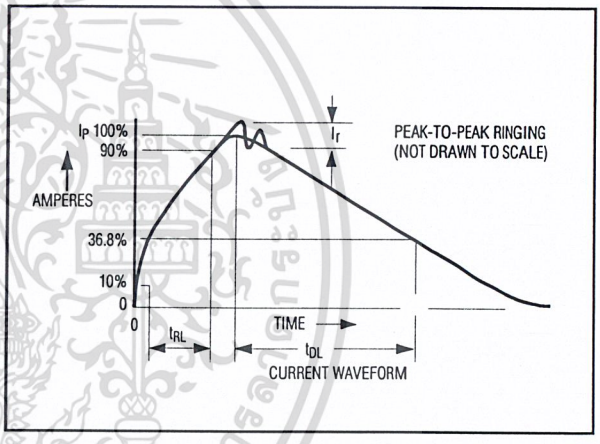


Figure 4b. Human Body Model Current Waveform

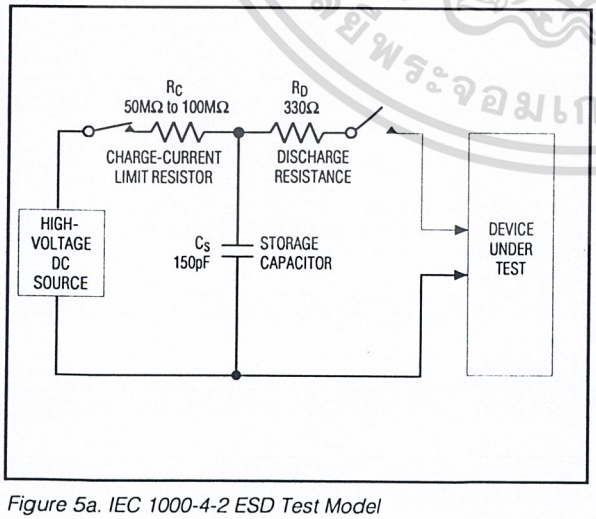


Figure 5a. IEC 1000-4-2 ESD Test Model

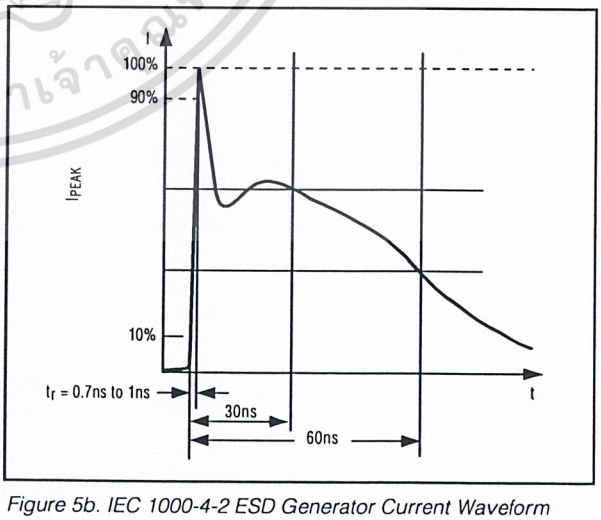


Figure 5b. IEC 1000-4-2 ESD Generator Current Waveform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Required Minimum Capacitor Values

VCC (V)	C1 (µF)	C2, C3, C4 (µF)
MAX3222E/MAX3232E/MAX3241E		
3.0 to 3.6	0.1	0.1
4.5 to 5.5	0.047	0.33
3.0 to 5.5	0.1	0.47
MAX3237E/MAX3246E		
3.0 to 3.6	0.22	0.22
3.15 to 3.6	0.1	0.1
4.5 to 5.5	0.047	0.33
3.0 to 5.5	0.22	1.0

Table 3. Logic-Family Compatibility with Various Supply Voltages

SYSTEM POWER-SUPPLY VOLTAGE (V)	VCC SUPPLY VOLTAGE (V)	COMPATIBILITY
3.3	3.3	Compatible with all CMOS families
5	5	Compatible with all TTL and CMOS families
5	3.3	Compatible with ACT and HCT CMOS, and with AC, HC, or CD4000 CMOS

For the MAX3237E, all logic and RS-232 I/O pins are characterized for protection to ±15kV per the Human Body Model.

ESD Test Conditions

ESD performance depends on a variety of conditions. Contact Maxim for a reliability report that documents test setup, test methodology, and test results.

Human Body Model

Figure 4a shows the Human Body Model, and Figure 4b shows the current waveform it generates when discharged into a low impedance. This model consists of a 100pF capacitor charged to the ESD voltage of interest, which is then discharged into the test device through a 1.5kΩ resistor.

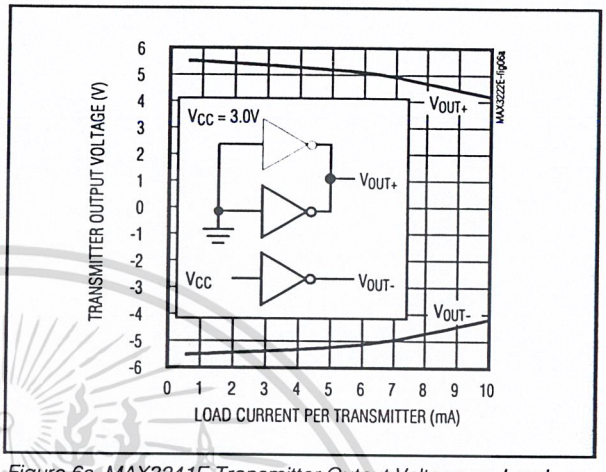


Figure 6a. MAX3241E Transmitter Output Voltage vs. Load Current Per Transmitter

IEC 1000-4-2

The IEC 1000-4-2 standard covers ESD testing and performance of finished equipment; it does not specifically refer to integrated circuits. The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E help you design equipment that meets level 4 (the highest level) of IEC 1000-4-2, without the need for additional ESD-protection components.

The major difference between tests done using the Human Body Model and IEC 1000-4-2 is higher peak current in IEC 1000-4-2, because series resistance is lower in the IEC 1000-4-2 model. Hence, the ESD withstand voltage measured to IEC 1000-4-2 is generally lower than that measured using the Human Body Model. Figure 5a shows the IEC 1000-4-2 model, and Figure 5b shows the current waveform for the ±8kV IEC 1000-4-2 level 4 ESD Contact Discharge test. The Air-Gap Discharge test involves approaching the device with a charged probe. The Contact Discharge method connects the probe to the device before the probe is energized.

Machine Model

The Machine Model for ESD tests all pins using a 200pF storage capacitor and zero discharge resistance. Its objective is to emulate the stress caused by contact that occurs with handling and assembly during manufacturing. All pins require this protection during manufacturing, not just RS-232 inputs and outputs. Therefore, after PC board assembly, the Machine Model is less relevant to I/O ports.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

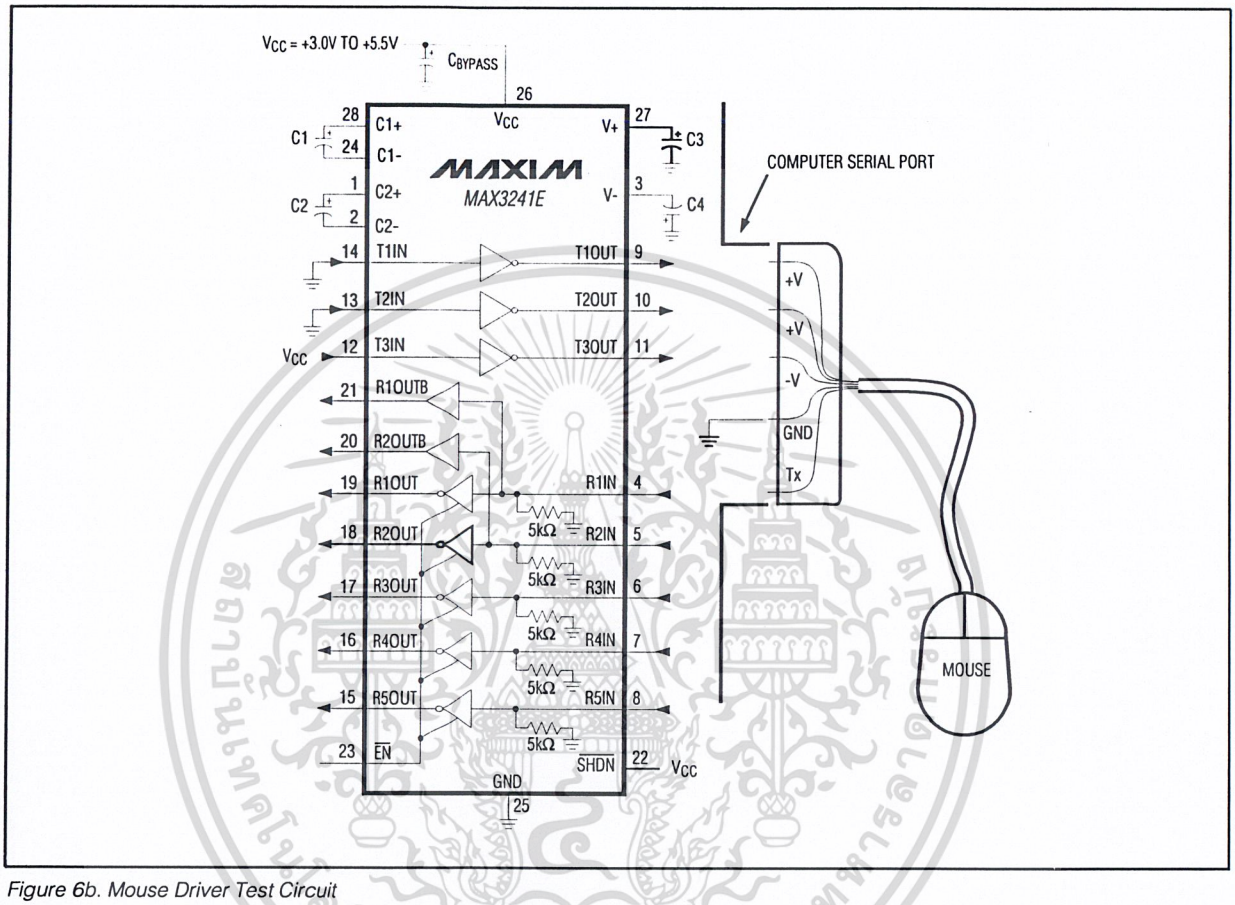


Figure 6b. Mouse Driver Test Circuit

Applications Information

Capacitor Selection

The capacitor type used for C1–C4 is not critical for proper operation; polarized or nonpolarized capacitors can be used. The charge pump requires 0.1µF capacitors for 3.3V operation. For other supply voltages, see Table 2 for required capacitor values. Do not use values smaller than those listed in Table 2. Increasing the capacitor values (e.g., by a factor of 2) reduces ripple on the transmitter outputs and slightly reduces power consumption. C2, C3, and C4 can be increased without changing C1's value. **However, do not increase C1 without also increasing the values of C2, C3, C4, and C_{BYPASS} to maintain the proper ratios (C1 to the other capacitors).**

When using the minimum required capacitor values, make sure the capacitor value does not degrade excessively with temperature. If in doubt, use capacitors with a larger nominal value. The capacitor's equivalent series resistance (ESR), which usually rises at low temperatures, influences the amount of ripple on V+ and V-.

Power-Supply Decoupling

In most circumstances, a 0.1µF V_{CC} bypass capacitor is adequate. In applications sensitive to power-supply noise, use a capacitor of the same value as charge-pump capacitor C1. Connect bypass capacitors as close to the IC as possible.

Operation Down to 2.7V

Transmitter outputs meet EIA/TIA-562 levels of ±3.7V with supply voltages as low as 2.7V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transmitter Outputs Recovering from Shutdown

Figure 3 shows two transmitter outputs recovering from shutdown mode. As they become active, the two transmitter outputs are shown going to opposite RS-232 levels (one transmitter input is high; the other is low). Each transmitter is loaded with 3kΩ in parallel with 2500pF. The transmitter outputs display no ringing or undesirable transients as they come out of shutdown. Note that the transmitters are enabled only when the magnitude of V- exceeds approximately -3.0V.

Mouse Drivability

The MAX3241E is designed to power serial mice while operating from low-voltage power supplies. It has been tested with leading mouse brands from manufacturers such as Microsoft and Logitech. The MAX3241E successfully drove all serial mice tested and met their current and voltage requirements.

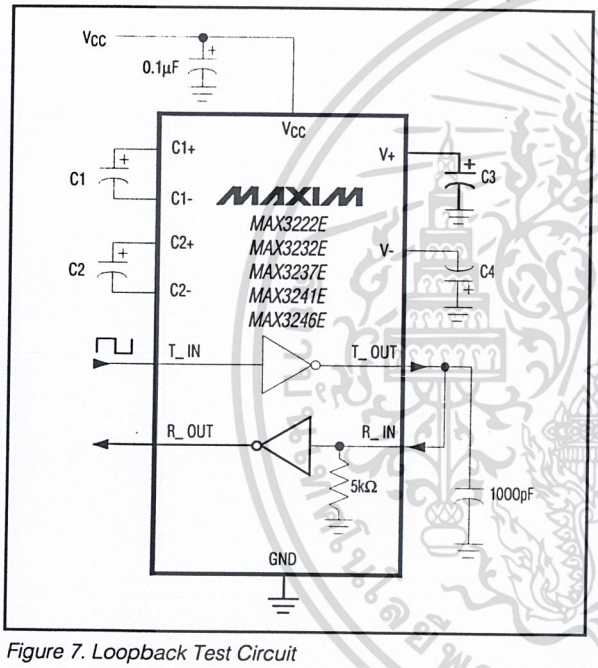


Figure 7. Loopback Test Circuit

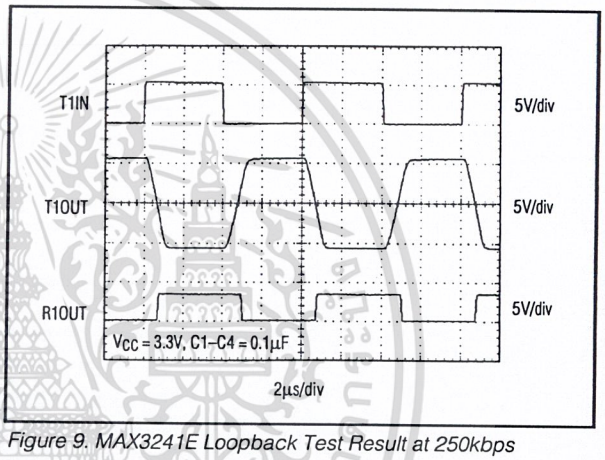


Figure 9. MAX3241E Loopback Test Result at 250kbps

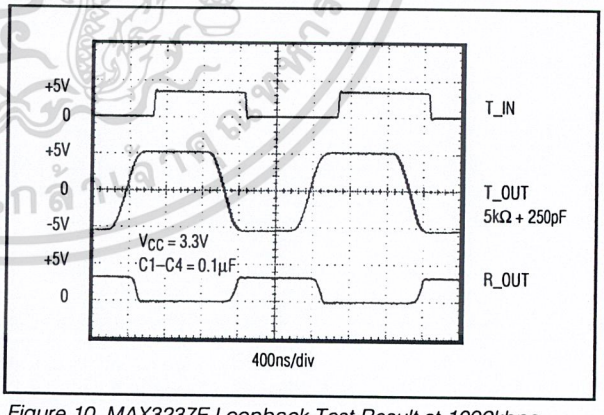


Figure 10. MAX3237E Loopback Test Result at 1000kbps (MBAUD = Vcc)

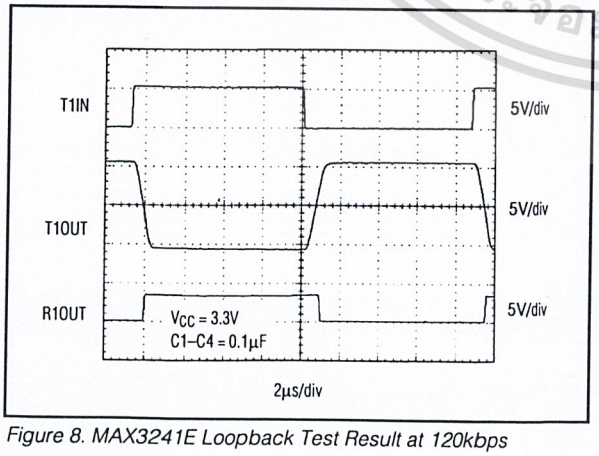


Figure 8. MAX3241E Loopback Test Result at 120kbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 6a shows the transmitter output voltages under increasing load current at +3.0V. Figure 6b shows a typical mouse connection using the MAX3241E.

High Data Rates

The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E maintain the RS-232 ±5V minimum transmitter output voltage even at high data rates. Figure 7 shows a transmitter loopback test circuit. Figure 8 shows a loopback test result at 120kbps, and Figure 9 shows the same test at 250kbps. For Figure 8, all transmitters were driven simultaneously at 120kbps into RS-232 loads in parallel with 1000pF. For Figure 9, a single transmitter was driven at 250kbps, and all transmitters were loaded with an RS-232 receiver in parallel with 1000pF.

The MAX3237E maintains the RS-232 ±5.0V minimum transmitter output voltage at data rates up to 1Mbps. Figure 10 shows a loopback test result at 1Mbps with MBAUD = V_{CC}. For Figure 10, all transmitters were loaded with an RS-232 receiver in parallel with 250pF.

Interconnection with 3V and 5V Logic

The MAX3222E/MAX3232E/MAX3237E/MAX3241E/MAX3246E can directly interface with various 5V logic families, including ACT and HCT CMOS. See Table 3 for more information on possible combinations of interconnections.

UCSP Reliability

The UCSP represents a unique packaging form factor that may not perform equally to a packaged product through traditional mechanical reliability tests. UCSP reliability is integrally linked to the user's assembly methods, circuit board material, and usage environment. The user should closely review these areas when considering use of a UCSP package. Performance through Operating Life Test and Moisture Resistance remain uncompromised as the wafer-fabrication process primarily determines it.

Mechanical stress performance is a greater consideration for a UCSP package. UCSPs are attached through direct solder contact to the user's PC board, foregoing the inherent stress relief of a packaged product lead frame. Solder joint contact integrity must be considered. Table 4 shows the testing done to characterize the UCSP reliability performance. In conclusion, the UCSP is capable of performing reliably through environmental stresses as indicated by the results in the table. Additional usage data and recommendations are detailed in the UCSP application note, which can be found on Maxim's website at www.maxim-ic.com.

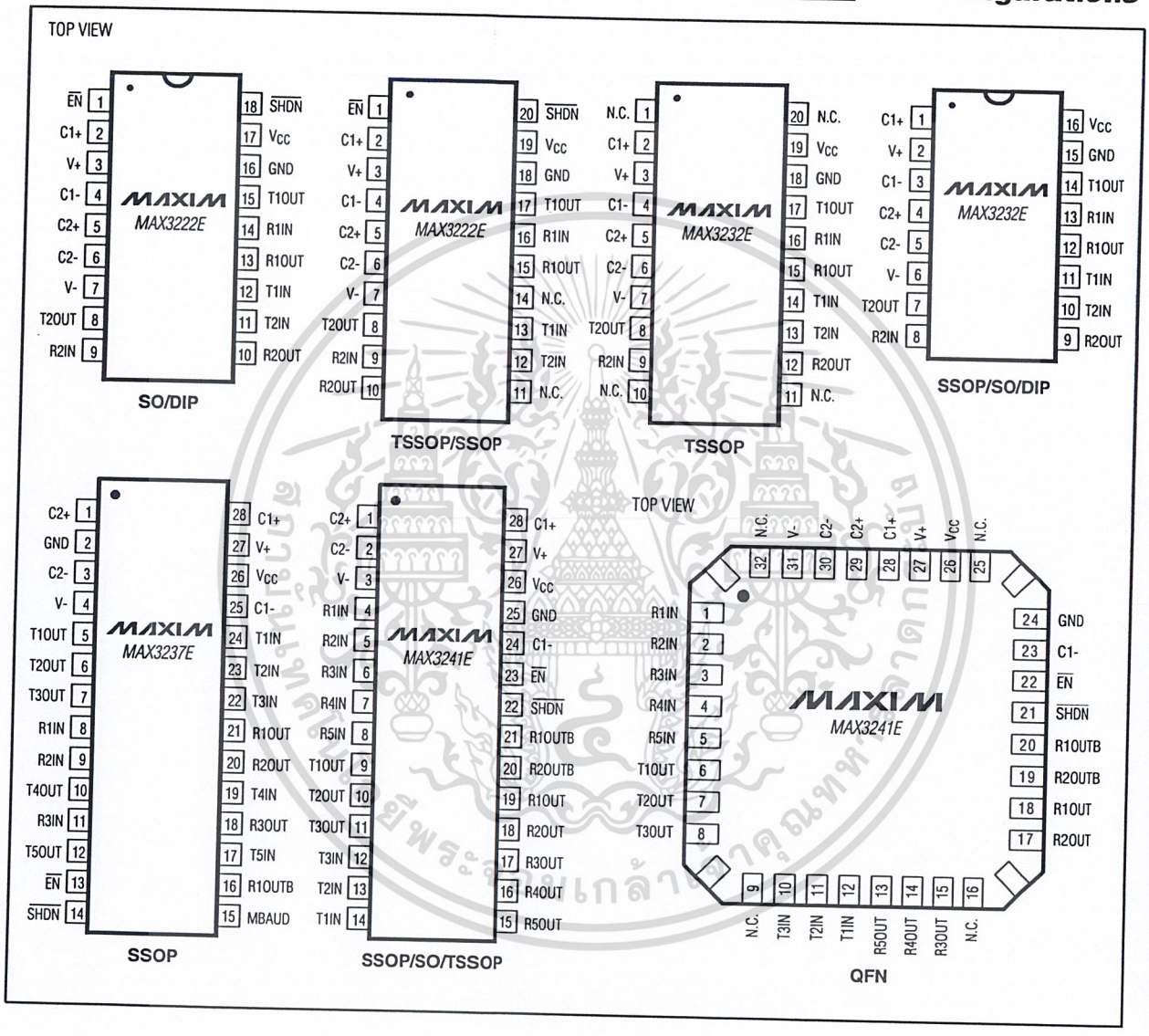
Table 4. Reliability Test Data

TEST	CONDITIONS	DURATION	FAILURES PER SAMPLE SIZE
Temperature Cycle	T_A = -35°C to +85°C, T_A = -40°C to +100°C	150 cycles, 900 cycles	0/10, 0/200
Operating Life	T_A = +70°C	240 hours	0/10
Moisture Resistance	T_A = +20°C to +60°C, 90% RH	240 hours	0/10
Low-Temperature Storage	T_A = -20°C	240 hours	0/10
Low-Temperature Operational	T_A = -10°C	24 hours	0/10
Solderability	8-hour steam age	—	0/15
ESD	±15kV, Human Body Model	—	0/5
High-Temperature Operating Life	T_J = +150°C	168 hours	0/45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

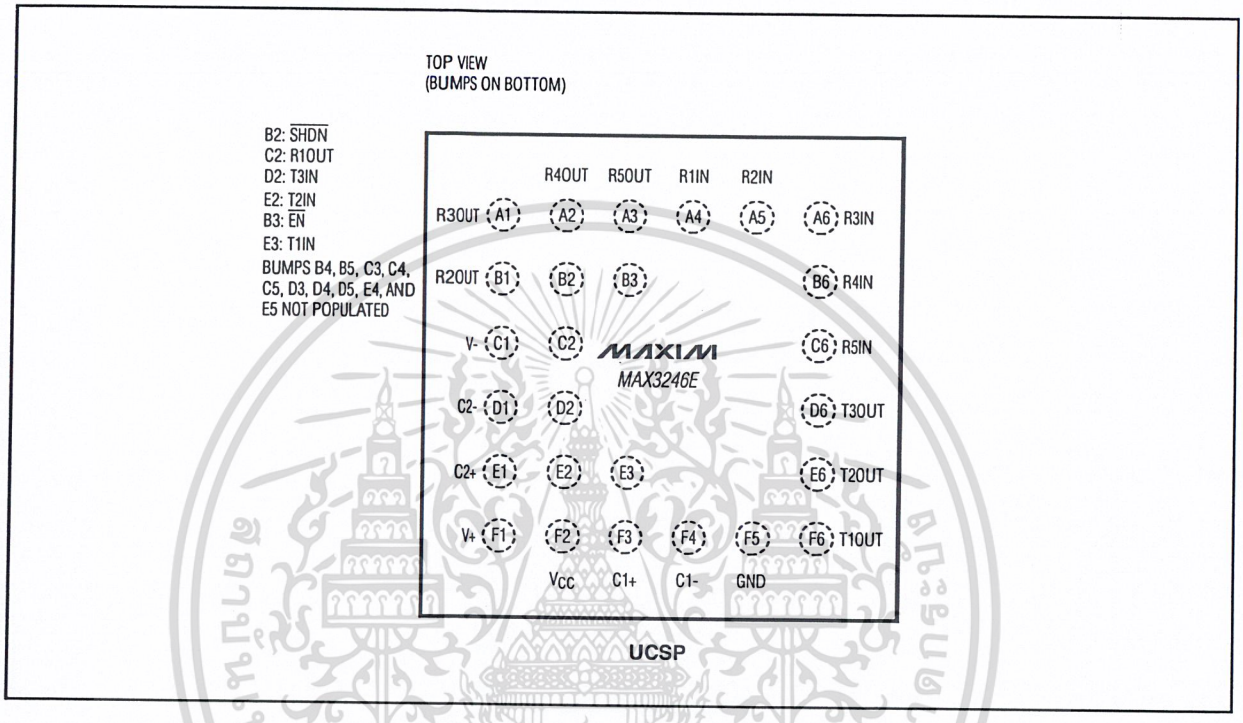
Pin Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

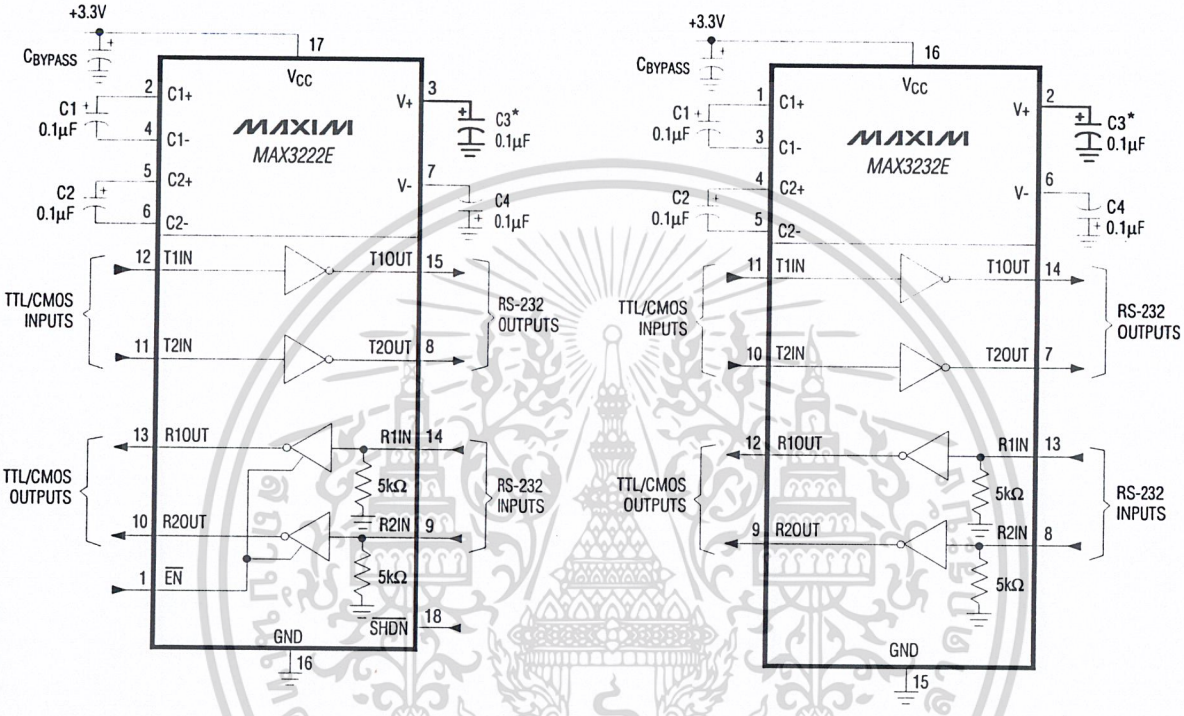
±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

Pin Configurations (continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Operating Circuits



*C3 CAN BE RETURNED TO EITHER V_{CC} OR GROUND.

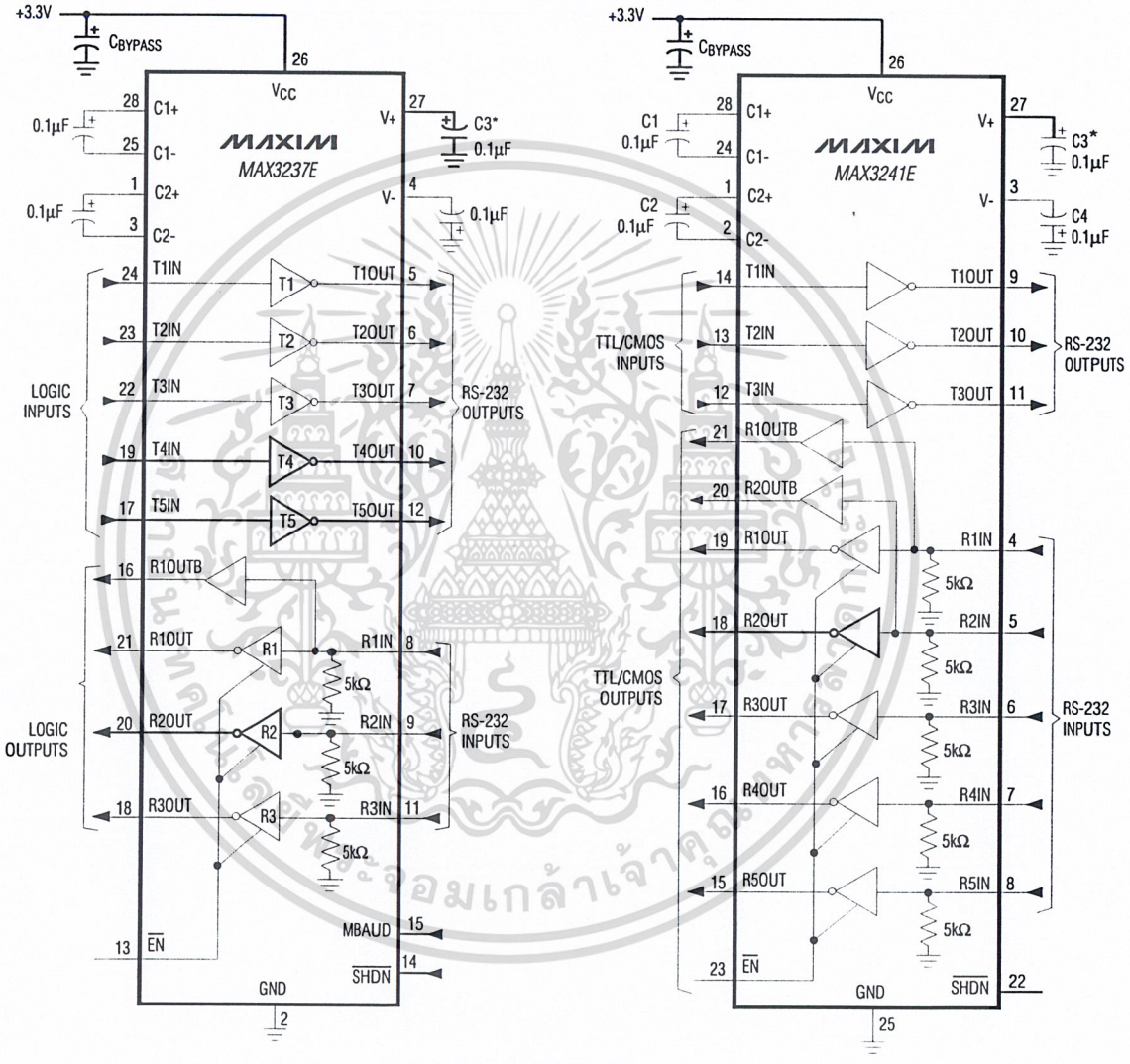
NOTE: PIN NUMBERS REFER TO SO/DIP PACKAGES.

SEE TABLE 2 FOR CAPACITOR SELECTION.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

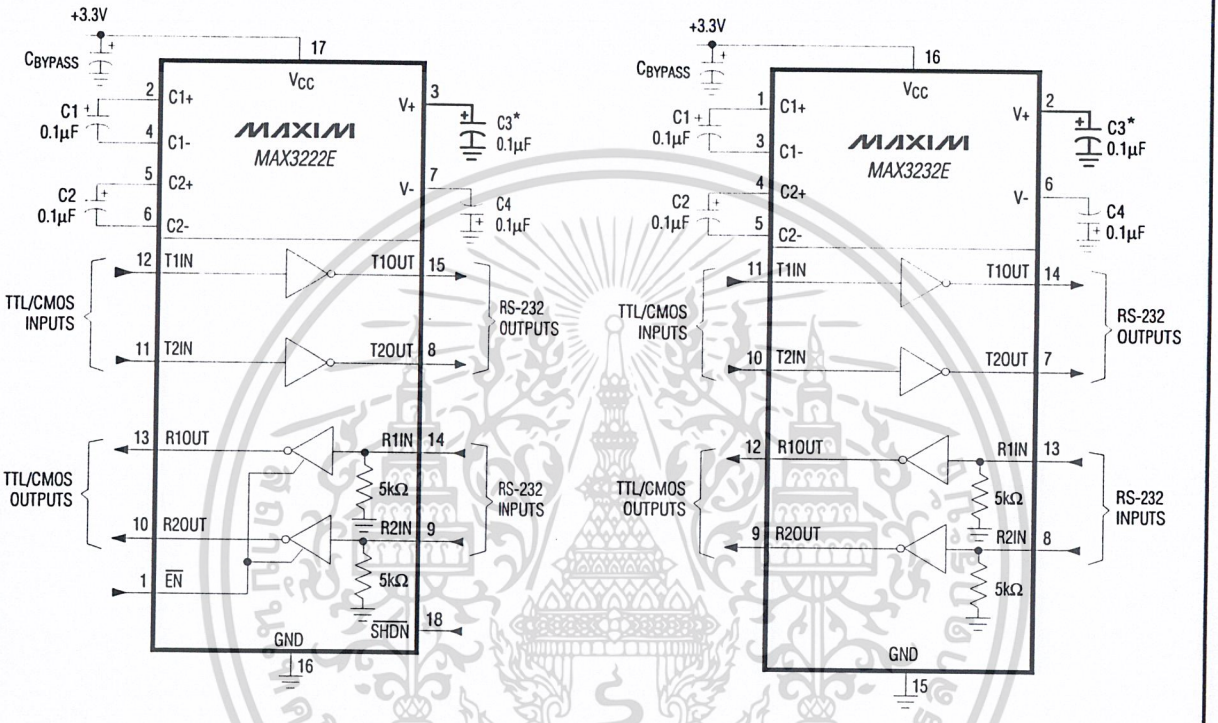
Typical Operating Circuits (continued)



*C3 CAN BE RETURNED TO EITHER V_{CC} OR GROUND.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Operating Circuits



*C3 CAN BE RETURNED TO EITHER V_{CC} OR GROUND.

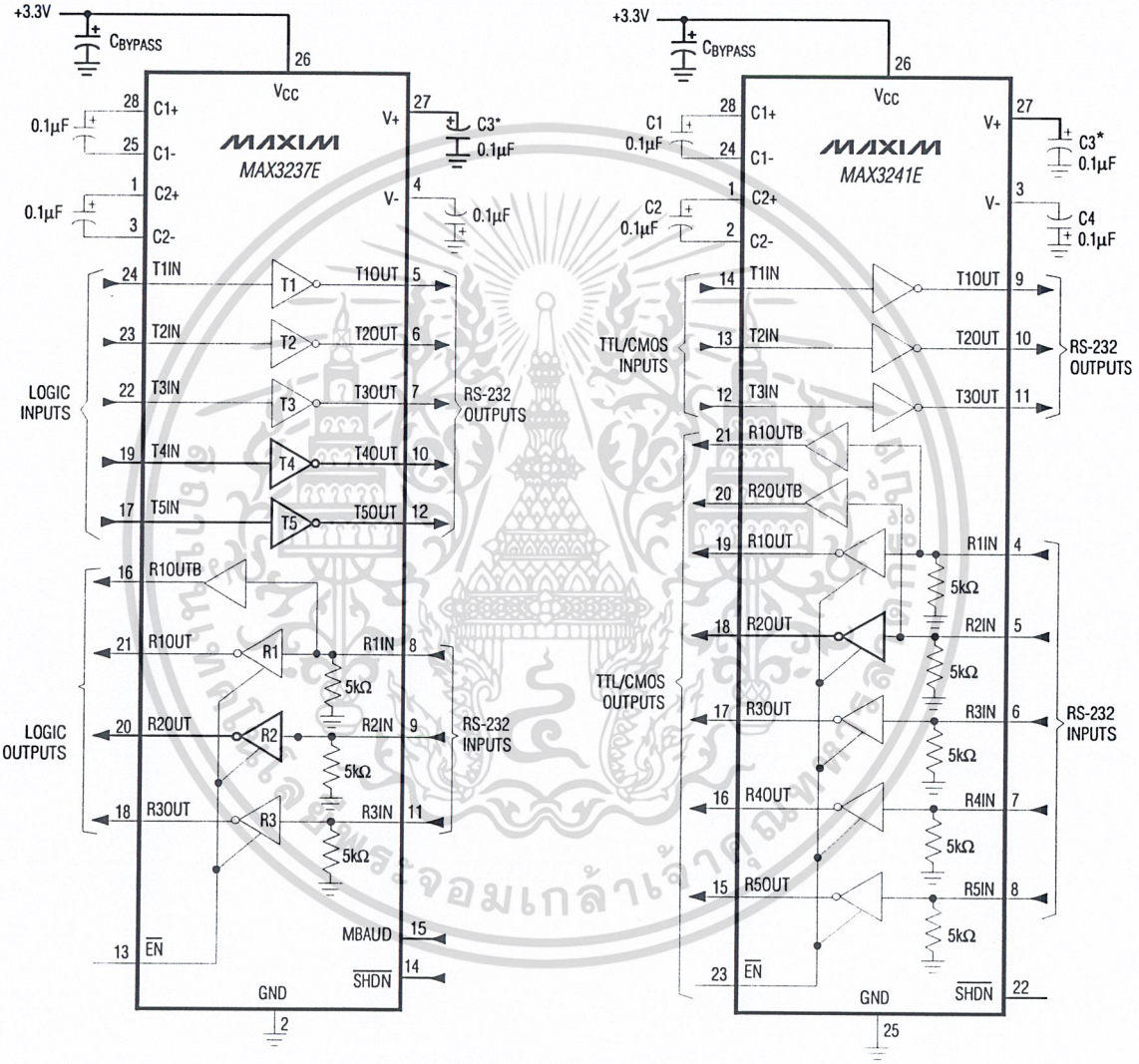
NOTE: PIN NUMBERS REFER TO SO/DIP PACKAGES.

SEE TABLE 2 FOR CAPACITOR SELECTION.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

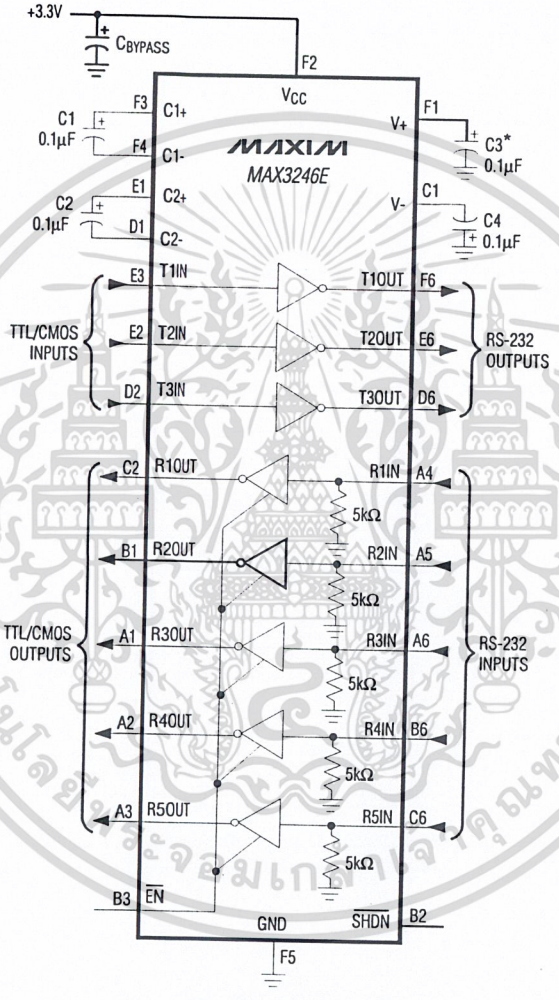
Typical Operating Circuits (continued)



*C3 CAN BE RETURNED TO EITHER V_{CC} OR GROUND.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Operating Circuits (continued)



*C3 CAN BE RETURNED TO EITHER V_{CC} OR GROUND.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

Selector Guide

PART	NO. OF DRIVERS/ RECEIVERS	LOW-POWER SHUTDOWN	GUARANTEED DATA RATE (bps)
MAX3222E	2/2	✔	250k
MAX3232E	2/2	—	250k
MAX3237E (Normal)	5/3	✔	250k
MAX3237E (MegaBaud)	5/3	✔	1M
MAX3241E	3/5	✔	250k
MAX3246E	3/5	✔	250k

Chip Information

TRANSISTOR COUNT:

MAX3222E/MAX3232E: 1129

MAX3237E: 2110

MAX3241E: 1335

MAX3246E: 842

Ordering Information (continued)

PART	TEMP RANGE	PIN-PACKAGE
MAX3222ECWN	0°C to +70°C	18 Wide SO
MAX3222ECPN	0°C to +70°C	18 Plastic DIP
MAX3222EC/D	0°C to +70°C	Dice*
MAX3222EEUP	-40°C to +85°C	20 TSSOP
MAX3222EEAP	-40°C to +85°C	20 SSOP
MAX3222EEWN	-40°C to +85°C	18 Wide SO
MAX3222EEPN	-40°C to +85°C	18 Plastic DIP
MAX3232ECAE	0°C to +70°C	16 SSOP
MAX3232ECWE	0°C to +70°C	16 Wide SO
MAX3232ECPE	0°C to +70°C	16 Plastic DIP
MAX3232ECUP	0°C to +70°C	20 TSSOP
MAX3232EEAE	-40°C to +85°C	16 SSOP
MAX3232EEWE	-40°C to +85°C	16 Wide SO
MAX3232EEPE	-40°C to +85°C	16 Plastic DIP
MAX3232EEUP	-40°C to +85°C	20 TSSOP
MAX3237ECAI	0°C to +70°C	28 SSOP
MAX3237EEAI	-40°C to +85°C	28 SSOP
MAX3241ECAI	0°C to +70°C	28 SSOP
MAX3241ECWI	0°C to +70°C	28 Wide SO
MAX3241ECUI	0°C to +70°C	28 TSSOP
MAX3241ECGJ	0°C to +70°C	32 QFN
MAX3241EEAI	-40°C to +85°C	28 SSOP
MAX3241EEWI	-40°C to +85°C	28 Wide SO
MAX3241EEUI	-40°C to +85°C	28 TSSOP
MAX3246ECBX-T	0°C to +70°C	6 x 6 UCSP*
MAX3246EEBX-T	-40°C to +85°C	6 x 6 UCSP*

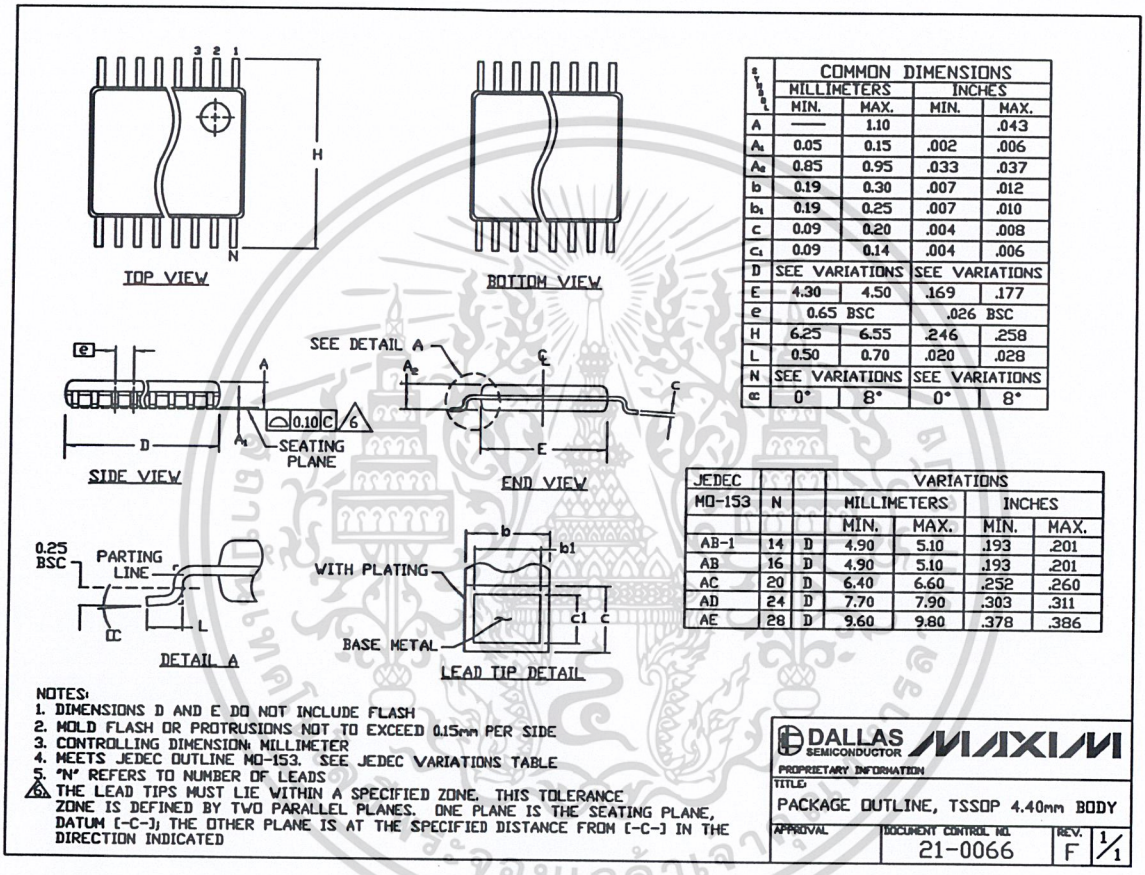
**Requires solder temperature profile described in the Absolute Maximum Ratings section. UCSP Reliability is integrally linked to the user's assembly methods, circuit board material, and environment. Refer to the UCSP Reliability Notice in the UCSP Reliability section of this datasheet for more information.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Package Information

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/packages**.)

TSSOP4.40mm.EPS



SYMBOL	COMMON DIMENSIONS			
	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	———	1.10		.043
A_1	0.05	0.15	.002	.006
A_2	0.85	0.95	.033	.037
b	0.19	0.30	.007	.012
b_1	0.19	0.25	.007	.010
c	0.09	0.20	.004	.008
c_1	0.09	0.14	.004	.006
D	SEE VARIATIONS		SEE VARIATIONS	
E	4.30	4.50	.169	.177
e	0.65 BSC		.026 BSC	
H	6.25	6.55	.246	.258
L	0.50	0.70	.020	.028
N	SEE VARIATIONS		SEE VARIATIONS	
α	0°	8°	0°	8°

JEDEC			VARIATIONS			
MO-153	N		MILLIMETERS		INCHES	
			MIN.	MAX.	MIN.	MAX.
AB-1	14	D	4.90	5.10	.193	.201
AB	16	D	4.90	5.10	.193	.201
AC	20	D	6.40	6.60	.252	.260
AD	24	D	7.70	7.90	.303	.311
AE	28	D	9.60	9.80	.378	.386

NOTES:
1. DIMENSIONS D AND E DO NOT INCLUDE FLASH
2. MOLD FLASH OR PROTRUSIONS NOT TO EXCEED 0.15mm PER SIDE
3. CONTROLLING DIMENSION: MILLIMETER
4. MEETS JEDEC OUTLINE MO-153. SEE JEDEC VARIATIONS TABLE
5. "N" REFERS TO NUMBER OF LEADS
6. THE LEAD TIPS MUST LIE WITHIN A SPECIFIED ZONE. THIS TOLERANCE ZONE IS DEFINED BY TWO PARALLEL PLANES. ONE PLANE IS THE SEATING PLANE, DATUM [-C-]; THE OTHER PLANE IS AT THE SPECIFIED DISTANCE FROM [-C-] IN THE DIRECTION INDICATED

DALLAS SEMICONDUCTOR MAXIM

PROPRIETARY INFORMATION

TITLE: PACKAGE OUTLINE, TSSOP 4.40mm BODY

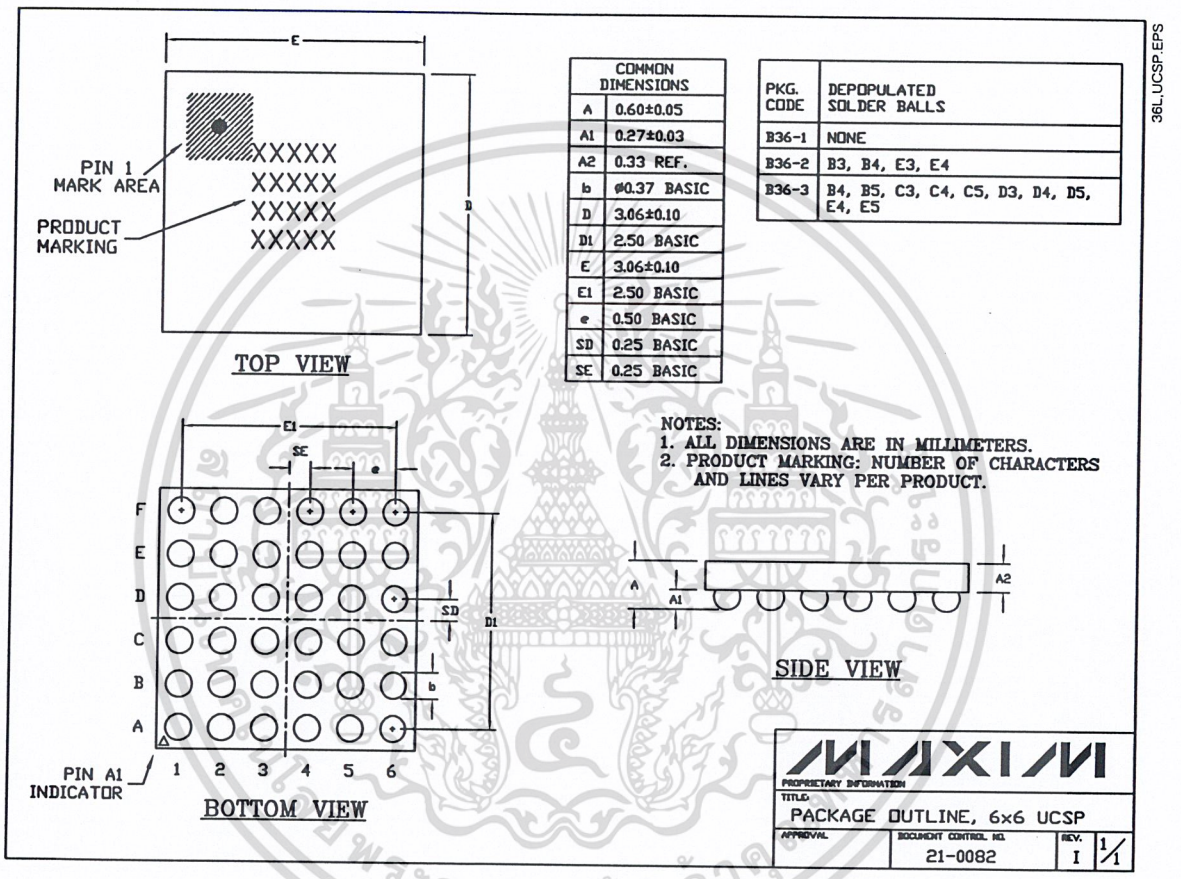
APPROVAL	DOCUMENT CONTROL NO.	REV.	
	21-0066	F	1/1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, Down to 10nA, 3.0V to 5.5V, Up to 1Mbps, True RS-232 Transceivers

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/packages**.)



Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

© 2002 Maxim Integrated Products Printed USA MAXIM is a registered trademark of Maxim Integrated Products.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

For a complete data sheet, please also download:

- The IC06 74HC/HCT/HCU/HCMOS Logic Family Specifications
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Information
- The IC06 74HC/HCT/HCU/HCMOS Logic Package Outlines

74HC/HCT573
Octal D-type transparent latch; 3-state

Product specification
File under Integrated Circuits, IC06

December 1990

Philips Semiconductors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Octal D-type transparent latch; 3-state | 74HC/HCT573

FEATURES

- Inputs and outputs on opposite sides of package allowing easy interface with microprocessors
- Useful as input or output port for microprocessors/microcomputers
- 3-state non-inverting outputs for bus oriented applications
- Common 3-state output enable input
- Functionally identical to the "563" and "373"
- Output capability: bus driver
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT573 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT573 are octal D-type transparent latches featuring separate D-type inputs for each latch and 3-state outputs for bus oriented applications.
A latch enable (LE) input and an output enable ($\overline{OE}$) input are common to all latches.

The "573" consists of eight D-type transparent latches with 3-state true outputs. When LE is HIGH, data at the D_n inputs enter the latches. In this condition the latches are transparent, i.e. a latch output will change state each time its corresponding D-input changes.

When LE is LOW the latches store the information that was present at the D-inputs a set-up time preceding the HIGH-to-LOW transition of LE.
When $\overline{OE}$ is LOW, the contents of the 8 latches are available at the outputs.
When $\overline{OE}$ is HIGH, the outputs go to the high impedance OFF-state.
Operation of the $\overline{OE}$ input does not affect the state of the latches.

The "573" is functionally identical to the "563" and "373", but the "563" has inverted outputs and the "373" has a different pin arrangement.

QUICK REFERENCE DATA

GND = 0 V; T_{amb} = 25 °C; $t_r = t_f$ = 6 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PHL}/ t_{PLH}	propagation delay	C_L = 15 pF; V_{CC} = 5 V			
	D_n to Q_n		14	17	ns
	LE to Q_n		15	15	ns
C_I	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per latch	notes 1 and 2	26	26	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in µW):
   $P_D = C_{PD} \times V_{CC}^2 \times f_i + \Sigma (C_L \times V_{CC}^2 \times f_o)$ where:
   f_i = input frequency in MHz; f_o = output frequency in MHz
   $\Sigma (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs
   C_L = output load capacitance in pF; V_{CC} = supply voltage in V
2. For HC the condition is V_I = GND to V_{CC}; for HCT the condition is V_I = GND to V_{CC} – 1.5 V

ORDERING INFORMATION

See *"74HC/HCT/HCU/HCMOS Logic Package Information"*.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
2, 3, 4, 5, 6, 7, 8, 9	D_0 to D_7	data inputs
11	LE	latch enable input (active HIGH)
1	$\overline{OE}$	3-state output enable input (active LOW)
10	GND	ground (0 V)
19, 18, 17, 16, 15, 14, 13, 12	Q_0 to Q_7	3-state latch outputs
20	V_{CC}	positive supply voltage

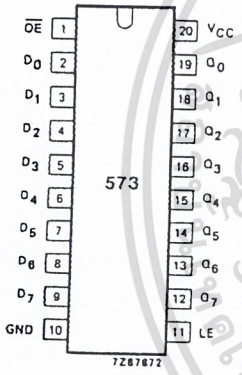


Fig.1 Pin configuration.

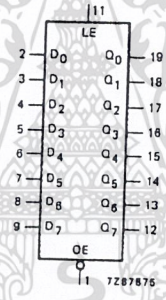


Fig.2 Logic symbol.

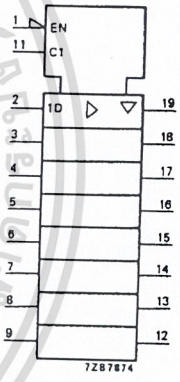


Fig.3 IEC logic symbol.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

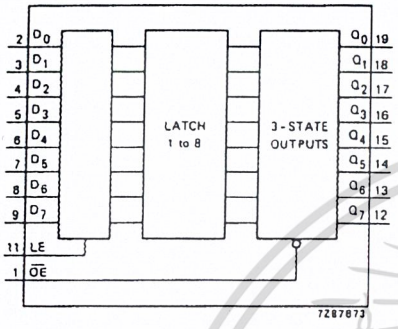


Fig.4 Functional diagram.

FUNCTION TABLE

OPERATING MODES	INPUTS			INTERNAL LATCHES	OUTPUTS
	$\overline{OE}$	LE	D_N		Q_0 to Q_7
enable and read register (transparent mode)	L	H	L	L	L
	L	H	H	H	H
latch and read register	L	L	l	L	L
	L	L	h	H	H
latch register and disable outputs	H	L	l	L	Z
	H	L	h	H	Z

Notes

1. H = HIGH voltage level
   h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition
   L = LOW voltage level
   l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition
   Z = high impedance OFF-state

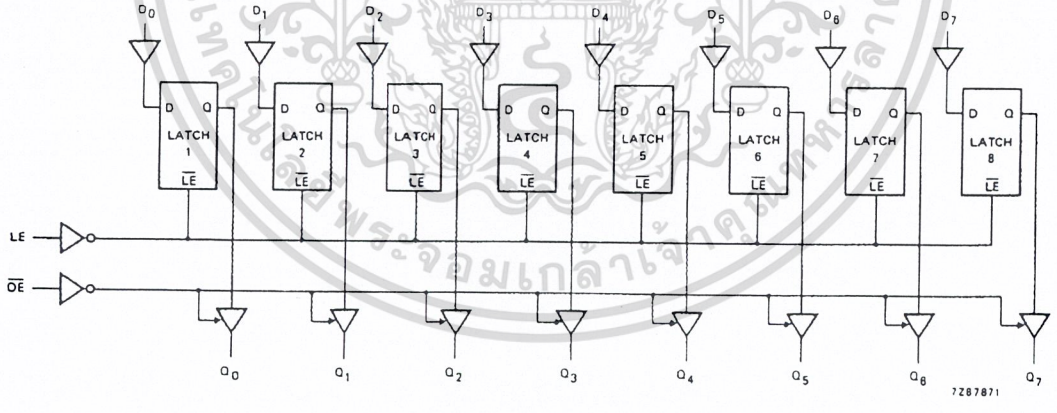


Fig.5 Logic diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see *"74HC/HCT/HCU/HCMOS Logic Family Specifications"*.

Output capability: bus driver
I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T_{amb} (°C)							UNIT	TEST CONDITIONS	
		74HC								V_{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125				
		min.	typ.	max.	min.	max.	min.	max.			
t_{PHL}/ t_{PLH}	propagation delay D_n to Q_n		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
t_{PHL}/ t_{PLH}	propagation delay LE to Q_n		50 18 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.7
t_{PZH}/ t_{PZL}	3-state output enable time $\overline{OE}$ to Q_n		44 16 13	140 28 24		175 35 30		210 42 36	ns	2.0 4.5 6.0	Fig.8
t_{PHZ}/ t_{PLZ}	3-state output disable time $\overline{OE}$ to Q_n		55 20 16	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.8
t_{THL}/ t_{TLH}	output transition time		14 5 4	60 12 10		75 15 13		90 18 15	ns	2.0 4.5 6.0	Fig.6
t_W	enable pulse width HIGH	80 16 14	14 5 4		100 20 17		120 24 20		ns	2.0 4.5 6.0	Fig.7
t_{su}	set-up time D_n to LE	50 10 9	11 4 3		65 13 11		75 15 13		ns	2.0 4.5 6.0	Fig.9
t_h	hold time D_n to LE	5 5 5	3 1 1		5 5 5		5 5 5		ns	2.0 4.5 6.0	Fig.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see *"74HC/HCT/HCU/HCMOS Logic Family Specifications"*.

Output capability: bus driver
I_{CC} category: MSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications. To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
D_n	0.35
LE	0.65
$\overline{OE}$	1.25

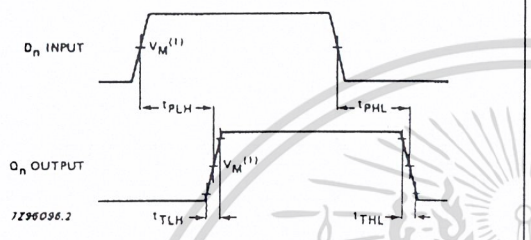
AC CHARACTERISTICS FOR 74HCT

GND = 0 V; $t_r = t_f$ = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T_{amb} (°C)							UNIT	TEST CONDITIONS	
		74HCT								V_{CC} (V)	WAVEFORMS
		+25			−40 to +85		−40 to +125				
		min.	typ.	max.	min.	max.	min.	max.			
t_{PHL}/ t_{PLH}	propagation delay D_n to Q_n		20	35		44		53	ns	4.5	Fig.6
t_{PHL}/ t_{PLH}	propagation delay LE to Q_n		18	35		44		53	ns	4.5	Fig.7
t_{PZH}/ t_{PZL}	3-state output enable time $\overline{OE}$ to Q_n		17	30		38		45	ns	4.5	Fig.8
t_{PHZ}/ t_{PLZ}	3-state output disable time $\overline{OE}$ to Q_n		18	30		38		45	ns	4.5	Fig.8
t_{THL}/ t_{TLH}	output transition time		5	12		15		18	ns	4.5	Fig.6
t_W	enable pulse width HIGH	16	5		20		24		ns	4.5	Fig.7
t_{su}	set-up time D_n to LE	13	7		16		20		ns	4.5	Fig.9
t_h	hold time D_n to LE	9	4		11		14		ns	4.5	Fig.9

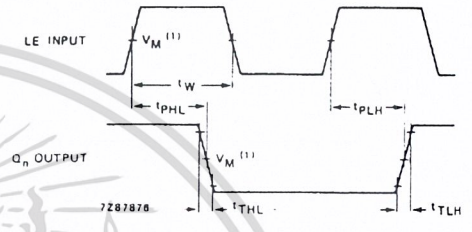
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC WAVEFORMS



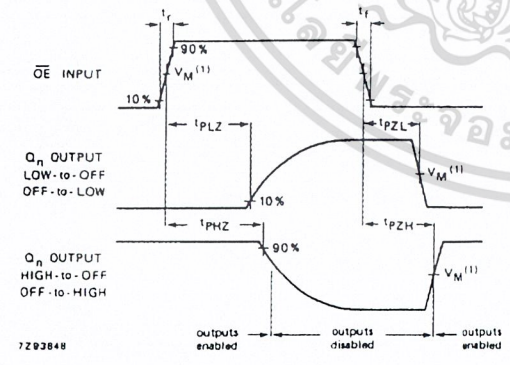
(1) HC : V_M = 50%; V_I = GND to V_{CC}.
HCT: V_M = 1.3 V; V_I = GND to 3 V.

Fig.6 Waveforms showing the data input (D_n) to output (Q_n) propagation delays and the output transition times.



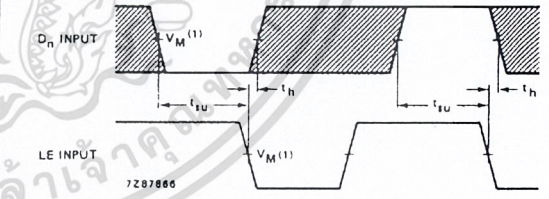
(1) HC : V_M = 50%; V_I = GND to V_{CC}.
HCT: V_M = 1.3 V; V_I = GND to 3 V.

Fig.7 Waveforms showing the latch enable input (LE) pulse width, the latch enable input to output (Q_n) propagation delays and the output transition times.



(1) HC : V_M = 50%; V_I = GND to V_{CC}.
HCT: V_M = 1.3 V; V_I = GND to 3 V.

Fig.8 Waveforms showing the 3-state enable and disable times.



The shaded areas indicate when the input is permitted to change for predictable output performance.

(1) HC : V_M = 50%; V_I = GND to V_{CC}.
HCT: V_M = 1.3 V; V_I = GND to 3 V.

Fig.9 Waveforms showing the data set-up and hold times for D_n input to LE input.

PACKAGE OUTLINES

See "74HC/HCT/HCU/HCMOS Logic Package Outlines".

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้