สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยภายในบ้านผ่านเครือข่ายคอมพิวเตอร์

Home Security System Via Computer Network

โดย

นางสาวโชติรส แผ้วศรี
นางสาวณัฐกานต์ สุวรรณวิศาล
นายณัฐพล ไชยบุญ

อาจารย์ที่ปรึกษา
รศ.ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เลขหมู่.....
เลขทะเบียน 46224
วัน, เดือน, ปี 21 มี.ค. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรักษาความปลอดภัยโดยผ่านระบบเครือข่ายคอมพิวเตอร์

ผู้จัดทำ

1. นางสาวโชติรส	แผ้วศรี	รหัส 41014114
2. นางสาวณัฐกานต์	สุวรรณวิศาล	รหัส 41014129
3. นายณัฐพล	ไชยบุญ	รหัส 41014135

.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานเรื่อง ระบบรักษาความปลอดภัยโดยผ่านระบบเครือข่ายคอมพิวเตอร์

Security System Via Computer Network

จัดทำโดย

1. นางสาวโชติรส	แผ้วศรี	รหัส 41014114
2. นางสาวณัฐกานต์	สุวรรณวิศาล	รหัส 41014129
3. นายณัฐพล	ไชยบุญ	รหัส 41014135

โครงสร้างนี้ได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยภายในบ้านผ่านเครือข่ายคอมพิวเตอร์

น.ส. โชติรส แผ้วศรี รหัส 41014114
น.ส. ณัฐกานต์ สุวรรณวิศาล รหัส 41014129
นาย ณัฐพล ไชยบุญ รหัส 41014135
รศ.ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา
ภาคเรียนที่ 2 ปีการศึกษา 2544

บทคัดย่อ

โครงงานนี้เป็นการนำไมโครคอมพิวเตอร์ ไมโครคอนโทรลเลอร์ และการรับส่งข้อมูลของเครือข่ายคอมพิวเตอร์มาประยุกต์เป็นระบบรักษาความปลอดภัยภายในบ้าน โดยแบ่งเป็น 3 ส่วน ส่วนแรกเป็นระบบเตือนภัยภายในบ้าน โดยใช้ไมโครคอนโทรลเลอร์ PIC16F84 เป็นส่วนประมวลผลกลางติดต่อกับอุปกรณ์ตรวจจับ (Sensor) อินฟราเรด, อุปกรณ์ตรวจวัดอุณหภูมิและควบคุมอุปกรณ์ไฟฟ้าได้ ส่วนที่สองเป็นส่วนที่ใช้คอมพิวเตอร์ควบคุมเพื่อจะทำการติดต่อกับไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรมและส่วนที่สามเป็นการทำงานผ่านเครือข่ายอินเตอร์เน็ต โดยระบบนี้สามารถที่จะตรวจสอบสถานะของอุปกรณ์ตรวจจับ (Sensor) ตรวจสอบอุณหภูมิหรือการเปิดปิดอุปกรณ์ไฟฟ้าผ่านระบบเครือข่ายอินเตอร์เน็ตโดยใช้โปรแกรมเว็บบราวเซอร์ (Web Browser) ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Home Security System Via Computer Network

Miss. Chotirot Paeosi 41014114
Miss. Nattakarn Suwanvisan 41014129
Mr. Nathapon Chaibun 41014135
Assoc.Prof. Manus Sangworasin Advisor
Academic Year 2001

Abstract

This project study about microcomputer, microcontroller and data transmission via computer network for application or security system. This system consists of three parts, the first part is a security system used PIC16F84 microcontroller is central processing unit for sensor device, sensor temperature and controlling electric device, the second part is computer control which interface with serial port. The last one concerns to internet system such as examining status of sensor and turn ON/OFF electrical equipment via internet by Web Browser program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 โครงสร้างของระบบเครือข่าย	2
2.1 ระบบเครือข่ายโดยรวม	2
2.2 ระบบเตือนภัยภายในบ้าน	3
2.3 ส่วนของคอมพิวเตอร์ที่บ้าน(Server)ที่ติดต่อกับพอร์ตอนุกรม	3
2.4 ส่วนของการรับและส่งข้อมูลผ่านหน้าเว็บ	3
บทที่ 3 ไมโครคอนโทรลเลอร์PIC16F84กับการรับส่งข้อมูลแบบอนุกรม	4
3.1 พื้นฐานของไมโครคอนโทรลเลอร์PIC16F84	4
3.2 สถาปัตยกรรมของPIC16F84	5
3.3 การสื่อสารข้อมูลแบบอนุกรม	9
3.4 การจัดสรรหน่วยความจำรีจีสเตอร์ควบคุมและสแต็ก	11
3.5 ชุดคำสั่งของPIC16F84	13
บทที่ 4 ระบบเตือนภัยภายในบ้าน	17
4.1 วงจรเชื่อมต่อเพื่อควบคุมอุปกรณ์ไฟฟ้า	17
4.2 วงจรตัวส่งและตัวรับสัญญาณอินฟราเรด	18
บทที่ 5 ส่วนของคอมพิวเตอร์ที่บ้าน(Server) ที่ติดต่อกับพอร์ตอนุกรม	21
5.1 การเขียนโปรแกรมวิชวลเบสิกเพื่อใช้งานพอร์ตอนุกรม	21
5.2 การแสดงผลของโปรแกรมและการเขียนโปรแกรมหน้าจอควบคุมหลัก	27
บทที่ 6 ส่วนของการรับและส่งข้อมูลผ่านหน้าเว็บ	30
6.1 การจำลองให้คอมพิวเตอร์(ที่ต่อกับระบบรักษาความปลอดภัย)ให้เป็น Web Server	30
6.2 ส่วนที่เป็นโปรแกรมที่ใช้รับและส่งข้อมูลผ่านพอร์ตและมาแสดงผลบนหน้าเว็บ	31
6.3 การใช้งานระบบรักษาความปลอดภัยผ่านทางหน้าเว็บ	36
บทที่ 7 ผลการทดลอง	37
บทที่ 8 สรุปและวิจารณ์โครงงาน	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 ภาพรวมของระบบ	2
รูปที่ 3.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC16F84	5
รูปที่ 3.2 แสดงการจัดขาของ PIC16F84	7
รูปที่ 3.3 รูปแบบการสื่อสารข้อมูลอนุกรมของ PIC16F84	9
รูปที่ 3.4 ตัวอย่างคาบเวลาของการสื่อสารข้อมูลอนุกรม	10
รูปที่ 3.5 การจัดสรรหน่วยความจำข้อมูลใน PIC16F84	11
รูปที่ 3.6 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F84	11
รูปที่ 4.1 แสดงวงจรควบคุมไฟ 220 โวลต์	17
รูปที่ 4.2 แสดงวงจรที่ใช้คำนวณหาความถี่ของไอซี555	18
รูปที่ 4.3 แสดงวงจรอินฟราเรดทางด้านส่ง	19
รูปที่ 4.4 แสดงวงจรอินฟราเรดทางด้านรับ	20
รูปที่ 5.1 แสดงหน้าจอหลักของโปรแกรม	24
รูปที่ 5.2 แสดง Flow Chart ของการควบคุมหลอดไฟจาก PC ผ่านพอร์ตอนุกรม	29
รูปที่ 5.3 แสดงการสั่งงานปิดหลอดไฟจากหน้าจอของโปรแกรม	28
รูปที่ 6.1 แสดงการติดต่อแบบ client-server ของเครือข่ายอินเตอร์เน็ต	30
รูปที่ 6.2 Flow Chart แสดงการรับส่งข้อมูลผ่านพอร์ตอนุกรมในโปรแกรมVisual Basic	33
รูปที่ 6.3 Flow Chart แสดงการทำงานของโปรแกรมที่ใช้แสดงสถานะของอุปกรณ์ไฟฟ้าของเครื่อง Server ผ่านทางหน้าเว็บ	34
รูปที่ 6.4 Flow Chart แสดงการทำงานของโปรแกรมที่ใช้ควบคุมอุปกรณ์ไฟฟ้าของเครื่อง Server ผ่านทางหน้าเว็บ	35
รูปที่ 6.5 แสดงสถานะของหลอดไฟและการเปิด-ปิดหลอดไฟผ่านทางหน้าเว็บ	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 3-1 แสดงรายละเอียดการใช้งานแต่ละขาของ PIC16F84	8
ตารางที่ 3-2 ชุดคำสั่งของ PIC16F84	15
ตารางที่ 7-1 แสดงผลการควบคุมอุปกรณ์ไฟฟ้า	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

เทคโนโลยีทางด้านอินเตอร์เน็ตได้เข้ามามีบทบาทอย่างมากในปัจจุบัน การรับรู้ข่าวสารสามารถเชื่อมโยงกันได้ทั่วโลกด้วยอินเตอร์เน็ต ซึ่งเป็นระบบเครือข่ายขนาดใหญ่ที่ใช้โปรโตคอล TCP/IP ในการเชื่อมต่อกัน ดังนั้นทางผู้จัดทำจึงได้นำเทคโนโลยีนี้มาประยุกต์โดยจัดทำระบบรักษาความปลอดภัยที่ทำงานผ่านทางเครือข่ายคอมพิวเตอร์ ซึ่งใช้งานผ่านอินเตอร์เน็ต ซึ่งจะเป็นประโยชน์กับเจ้าของบ้านในกรณีที่ไม่อยู่บ้านและไม่ว่าจะอยู่ที่ใดก็สามารถที่จะตรวจสอบความปลอดภัยของบ้านได้โดยอินเตอร์เน็ต

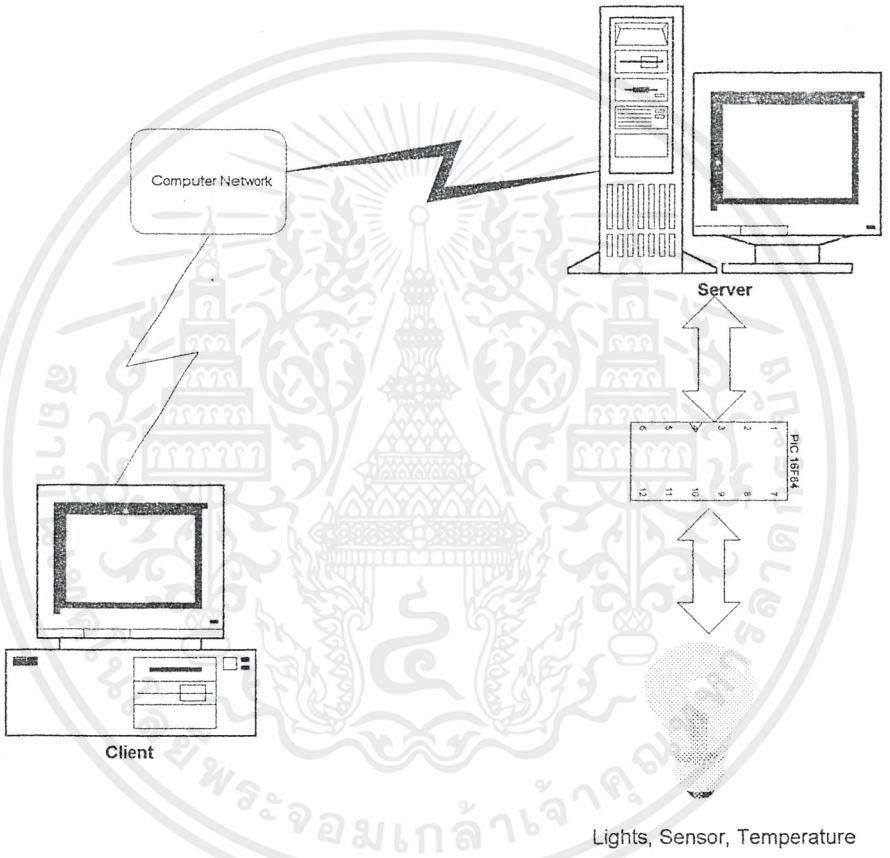
ระบบรักษาความปลอดภัยแบบพื้นฐานนี้ได้ออกแบบให้มีการติดตั้งเซนเซอร์ตรวจจับแบบอินฟราเรดตามประตูและหน้าต่างของบ้าน ซึ่งป้องกันการบุกรุกจากผู้ไม่พึงประสงค์ ในส่วนระบบป้องกันอัคคีภัยจะเป็นการตรวจวัดจากอุณหภูมิภายในบ้านว่ามีอุณหภูมิสูงมากจนผิดสังเกตุหรือไม่ และนอกจากนี้ผู้เข้าของบ้านยังสามารถทราบว่าอุปกรณ์ไฟฟ้าภายในบ้านนั้นเปิดหรือปิดอยู่ ถ้าลืมปิดก็สามารถที่จะสั่งให้อุปกรณ์นั้นๆปิดได้ที่เครื่องคอมพิวเตอร์เครื่องใดก็ได้ที่ออนไลน์อยู่บนเครือข่ายอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
โครงสร้างของระบบเครือข่าย

2.1 ระบบเครือข่ายโดยรวม

ระบบรักษาความปลอดภัยโดยผ่านระบบเครือข่ายคอมพิวเตอร์ สามารถแสดงภาพของระบบโดยรวมได้ดังรูปที่ 2.1



รูปที่ 2.1 ภาพรวมของระบบ

ระบบโดยรวมสามารถที่จะจำแนกได้ทั้งหมด 3 ส่วนใหญ่คือ

1. ระบบเตือนภัยภายในบ้าน
2. ส่วนของคอมพิวเตอร์ที่บ้าน(Server)ที่ติดต่อกับพอร์ตอนุกรม
3. ส่วนของการรับและส่งข้อมูลผ่านหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ระบบเตือนภัยภายในบ้าน

ระบบเตือนภัยภายในบ้านจะประกอบด้วย sensor ตรวจสอบที่จุดต่างๆ วงจรควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าต่างๆ และวงจรที่ใช้สำหรับตรวจสอบอุณหภูมิภายในบ้าน

สำหรับการควบคุมอุปกรณ์ต่างๆในระบบได้เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล PIC 16F84 เป็นตัวควบคุมเพราะมีความเร็วในการประมวลผลสูงและง่ายต่อการศึกษา รวมทั้งมีพอร์ตสำหรับรับส่งข้อมูลอยู่ภายในเลย ซึ่งช่วยให้การพัฒนาระบบเป็นไปด้วยความสะดวกและรวดเร็ว

2.3 ส่วนของคอมพิวเตอร์ที่บ้าน(Server)ที่ติดต่อกับพอร์ตอนุกรม

การควบคุมและตรวจสอบสถานะของวงจรต่างๆ ในระบบเตือนภัยภายในบ้าน สามารถดำเนินการผ่านเครื่อง Personal Computer (PC) โดยการรับส่งข้อมูลระหว่าง PC กับส่วนของไมโครคอนโทรลเลอร์ในระบบเตือนภัยจะผ่านทาง Serial Port

โดยในส่วนโปรแกรมที่ใช้รับคำสั่งและแสดงผลทาง PC ได้เลือกใช้โปรแกรม Visual Basic 6.0 ซึ่งจะเป็นการเขียนโปรแกรมในลักษณะ Visual Programming ทำให้เราสามารถสร้างโปรแกรมได้อย่างสะดวก อีกทั้งยังสามารถเขียนโปรแกรมติดต่อกับพอร์ตอนุกรมได้อย่างดีอีกด้วย

2.4 ส่วนของการรับและส่งข้อมูลผ่านทางหน้าเว็บ

ในการพัฒนาโปรแกรม นอกจากโปรแกรมในส่วนของ Visual Basic แล้วยังได้มีการเขียนโปรแกรมที่ใช้ติดต่อกับไมโครคอนโทรลเลอร์โดยใช้เท็กซ์ไฟล์เป็นสิ่งเชื่อมระหว่าง ASP กับ Visual Basic ที่สามารถ Run ได้ผ่านเครือข่ายต่างๆเช่น LAN, WAN และ Internet โดยมีเอกสารแบบ HTML เป็นตัวแสดงผลผ่านทาง Browser

ในส่วนการใช้งานโปรแกรมระบบเตือนภัยจากคอมพิวเตอร์เครื่องอื่นๆ ในเครือข่ายได้มีการจำลอง PC (ที่เชื่อมต่อกับไมโครคอนโทรลเลอร์) ให้เป็น Web Server และใช้การสร้าง Web ในรูปแบบของ ASP (ซึ่งเป็น Web แบบ Dynamic) ทำให้สามารถใช้โปรแกรมระบบเตือนภัยผ่านทางคอมพิวเตอร์เครื่องอื่นๆ ในเครือข่ายโดยใช้การรับส่งข้อมูลกับเครื่องที่เป็น Web Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ไมโครคอนโทรลเลอร์ PIC16F84 กับการรับส่งข้อมูลแบบอนุกรม

3.1 พื้นฐานของไมโครคอนโทรลเลอร์ PIC16F84

3.1.1 คุณสมบัติทางเทคนิคของ PIC16F84

สามารถแบ่งเป็น 3 ส่วนคือ หน่วยประมวลผลกลาง (Central Processing Unit : CPU),ส่วนของเพอริเฟอรัล (Peripheral) และคุณสมบัติพิเศษอื่นๆ

คุณสมบัติทางเทคนิคของหน่วยประมวลผลกลางภายใน PIC16F84

- หน่วยประมวลผลกลางเป็นแบบ RISC
- มีคำสั่งเพียง 35 คำสั่ง ขนาด 14 บิต
- ทุกคำสั่งใช้เวลาในการประมวลผลเพียง 1 ไซเกิลของสัญญาณนาฬิกา หรือประมาณ 250 นาโนวินาทีที่สัญญาณนาฬิกาความถี่ 4 MHz ยกเว้นชุดคำสั่งการกระโดดจะใช้เวลา 2 ไซเกิลของสัญญาณนาฬิกา
- ประมวลผลข้อมูลขนาด 8 บิต
- มีรีจิสเตอร์ฟังชั่นพิเศษ 15 ตัว
- มีสแต็ก 8 ระดับ
- มีโหมดการอ้างอิงแอดเดรส 3 โหมดคือ แบบโดยตรง(direct),แบบโดยอ้อม(indirect),และแบบสัมพัทธ์(relative)
- มีแหล่งกำเนิดอินเตอร์รัปต์ 4 แหล่งได้แก่
    1. รับสัญญาณจากภายนอก โดยป้อนสัญญาณอินเตอร์รัปต์เข้าที่ขา RB0/INT
    2. จากไทเมอร์ TMR0 เกิดโอเวอร์โฟลว
    3. เมื่อเกิดการเปลี่ยนแปลงที่พอร์ต B (ขา RB4-RB7)
    4. เมื่อการเขียนข้อมูลลงในหน่วยความจำอีอีพรอมเสร็จสิ้นสมบูรณ์
- หน่วยความจำข้อมูล (data memory) เป็นแบบอีอีพรอมสามารถลบและเขียนใหม่ได้ประมาณล้านครั้งและเก็บข้อมูลได้นาน 40 ปี
- ขนาดหน่วยความจำโปรแกรมซึ่งเป็นแบบแฟลชมีขนาด 1 กิโลเวิร์ด(1 เวิร์ดของ PIC16F84 มีขนาด 14 บิต),หน่วยความจำอีอีพรอมภายใน 64 ไบต์ และหน่วยความจำแรม 68 ไบต์ซึ่งใช้เป็นรีจิสเตอร์

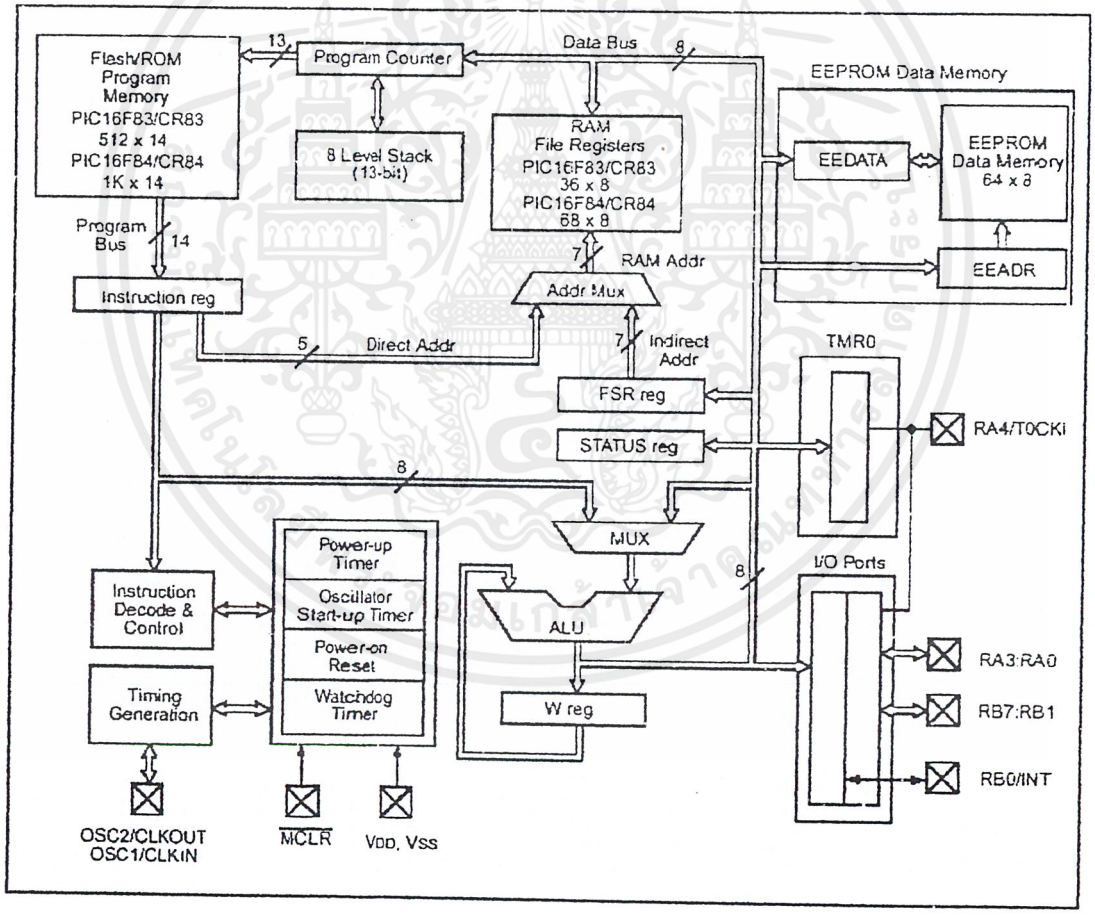
คุณสมบัติทางเทคนิคของเพอริเฟอรัลใน PIC16F84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีขาอินพุตเอาต์พุต 13 ขา สามารถกำหนดเป็นขาอินพุตหรือเอาต์พุตได้อย่างอิสระ
- กระแสซิงก์/ซอร์สของแต่ละขาอินพุตเอาต์พุตสูงพอที่จะขับ LED ได้โดยตรง
- กระแสซิงก์สูงสุด 25 mAต่อขา
- กระแสซอร์สสูงสุด 20 mAต่อขา
- มีไทเมอร์/เคาน์เตอร์ขนาด 8 บิตคือ TMR0 พร้อมกับปรีสเกลเลอร์ขนาด 8 บิตที่สามารถโปรแกรมได้

3.2 สถาปัตยกรรมของ PIC16F84

ไมโครคอนโทรลเลอร์ PIC16F84 ได้รับการบรรจุหน่วยประมวลผล,หน่วยความจำ,หน่วยคำนวณทางคณิตศาสตร์และหน่วยอินพุตเอาต์พุตไว้พร้อมสรรพ ทั้งยังมีไทเมอร์และวอตช์ด็อกครบถ้วนสมบูรณ์ ดังแสดงสถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC16F84 ในรูปที่3.1



รูปที่3.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC16F84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC16F84 มีการจัดสรรหน่วยความจำดังนี้

<u>หน่วยความจำโปรแกรม</u> มีโครงสร้างเป็นหน่วยความจำแบบแฟลช มีขนาด 1 กิโลเวิร์ด โดยใน 1 เวิร์ดของ PIC16F84 มีขนาด 14 บิต

<u>หน่วยความจำข้อมูลอีอีพรอม</u> ขนาด 64 ไบต์

<u>หน่วยความจำข้อมูลแรม</u> ได้รับการกำหนดให้ทำงานเป็นรีจีสเตอร์กำหนดแฟ้มข้อมูลหรือรีจีสเตอร์ไฟล์ขนาด 68 ไบต์

การเข้าถึงหน่วยความจำทั้งหมดของหน่วยประมวลผลกลางหรือซีพียูภายในไมโครคอนโทรลเลอร์นี้สามารถทำได้ทั้งในลักษณะโดยตรง,โดยอ้อม และแบบสัมพัทธ์ โดยมีรีจีสเตอร์ FSR (File Select Register) ทำหน้าที่ในการควบคุมการเข้าถึงหน่วยความจำแบบโดยอ้อม

เมื่อไมโครคอนโทรลเลอร์ทำงานตามคำสั่งที่กำหนดให้ข้อมูลของชุดคำสั่งจะถูกนำไปเก็บไว้ในรีจีสเตอร์คำสั่ง(Instruction register) จากนั้นจะถูกส่งต่อไปวงจรถอดรหัส เพื่อทำการควบคุมไทเมอร์ทั้งหมดภายในตัวไมโครคอนโทรลเลอร์ นอกจากนั้นยังส่งไปควบคุมหน่วยคำนวณทางคณิตศาสตร์โดยผ่านวงจรมัลติเพล็กด้วย

ใน PIC16F84 มีไทเมอร์/เคาน์เตอร์ขนาด 8 บิต 1 ตัวคือ TMR0 ภายในไทเมอร์/เคาน์เตอร์ตัวนี้มีปรีสเกลเลอร์ขนาด 8 บิตที่สามารถโปรแกรมได้ โดยสามารถรับสัญญาณอินพุตจากภายนอกผ่านทางขา RA4/T0CKI หรือเลือกรับสัญญาณนาฬิกาจากภายในก็ได้

หน่วยคำนวณทางคณิตศาสตร์ (Arithmatic Logic Unit : ALU)มีขนาด 8 บิต สามารถทำการบวก, ลบ,เลื่อนข้อมูล และประมวลผลทางลอจิก โดยใช้ฟังก์ชั่นบูลีน ในการทำงาน ALU จะต้องมีรีจีสเตอร์ W ช่วยเสมอ ซึ่งซีพียูจะไม่สามารถเข้าถึงรีจีสเตอร์ W นี้ได้โดยตรง เมื่อ ALU ทำงานมีผลล่อบิตทด(carry bit),บิตหลักทด(digit carry) และบิตศูนย์(zero bit)ในรีจีสเตอร์ STATUS

ในส่วนของพอร์ตอินพุตใน PIC16F84 มีด้วยกัน 2 พอร์ตคือ พอร์ต A และ B โดยในพอร์ตA มี 5 บิตคือ RA0-RA4 สำหรับขา RA4 ยังใช้เป็นขาอินพุตสำหรับรับสัญญาณนาฬิกาจากภายนอกให้แก่ TMR0 ด้วย ส่วนพอร์ต B มี 8 บิตคือ RB0-RB7 สำหรับขา RB0 ยังใช้เป็นขาอินพุตสำหรับสัญญาณอินเตอร์รัปต์ด้วย

การกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์ PIC16F84 เป็นหน้าที่ของส่วนกำเนิดจังหวะการทำงาน (timing generation) ซึ่งต้องทำงานสัมพันธ์กับไทเมอร์ทั้ง 3 ตัวคือ เพาเวอร์อัปไทเมอร์, ออสซิลเลเตอร์สตาร์ตอัปไทเมอร์ และวอชด็อกไทเมอร์ สำหรับ PIC16F84 สามารถใช้สัญญาณนาฬิกาจากภายนอกโดยต่อสัญญาณเข้าที่ขา OSC1 และOSC2 หรือจะใช้สัญญาณนาฬิกาจากภายในไมโครคอนโทรลเลอร์ก็ได้ด้วยการต่อตัวต้านทานและตัวเก็บประจุเพิ่มเติม

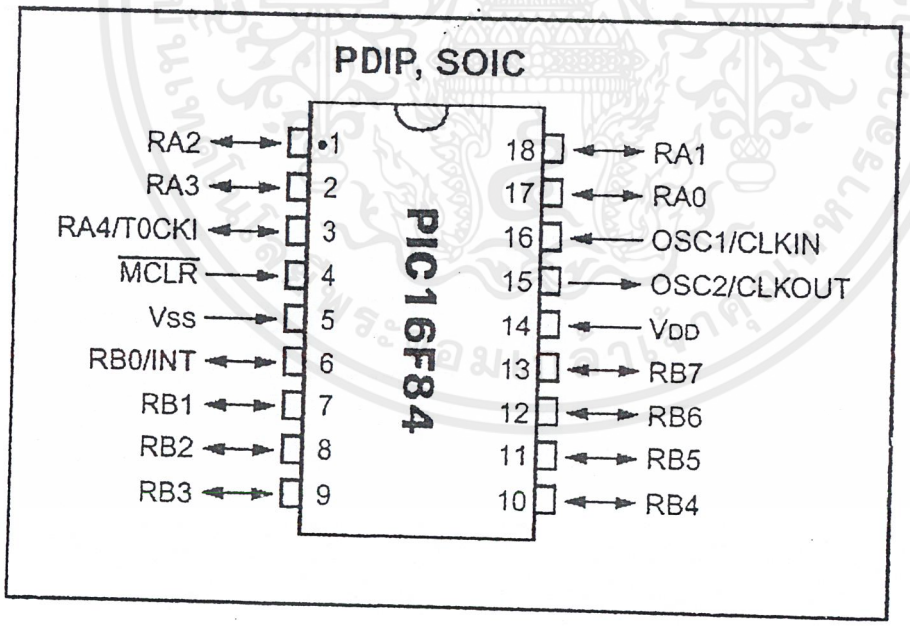
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การจัดขาของ PIC16F84

ไมโครคอนโทรลเลอร์ PIC16F84 บรรจุอยู่ในตัวถัง 2 แบบคือ PDIP (Plastic Dual-In Line package) ซึ่งมีลักษณะเดียวกับไอซีทั่วไป และแบบ SOIC อันเป็นตัวถังแบบที่ใช้ติดตั้งบนผิวหน้าของแผ่นวงจรพิมพ์ ตัวถังทั้งสองแบบมีขาต่อใช้งาน 18 ขาดังแสดงในรูปที่ 3.2 ซึ่งสามารถจัดขาต่อใช้งานของ PIC16F84 เป็น 4 กลุ่มคือ

1. กลุ่มสัญญาณนาฬิกา มี 2 ขา คือ OSC1/CLKIN(ขา 16) และOSC2/CLKOUT(ขา 15)
2. กลุ่มขาควบคุม มี 1 ขาคือ $\overline{\text{MCLR}}$(ขา 4)
3. กลุ่มขาพอร์ตอินพุตเอาต์พุต มี 13 ขา แบ่งเป็นขาพอร์ต A 5ขา ได้แก่ RA0-RA4(ขา 17,18,1,2 และ3)และขาพอร์ต B ได้แก่ ขาRB0-RB7(ขา 6 ถึง ขา 13)
4. กลุ่มขาไฟเลี้ยง มี 2 ขาคือ ขา Vss (ขา 5)หรือขาต่อกราวด์ และขา Vdd(ขา 14)หรือขาต่อไฟเลี้ยง ปกติใช้ไฟ +5V

สำหรับรายละเอียดโดยสรุปของขาต่อใช้งานทั้งหมดแสดงในรูปที่ 3.2 และตารางที่ 3.1



รูปที่ 3.2 แสดงการจัดขาของ PIC16F84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3-1 แสดงรายละเอียดการใช้งานแต่ละขาของ PIC16F84

ชื่อขา	ขาที่	ชนิดของขา	ชนิดของบัพเฟอร์	รายละเอียด
OSC1/CLKIN	16	อินพุต	ชมิตต์ทริกเกอร์ /CMOS	เป็นขาสำหรับรับสัญญาณนาฬิกาจากแหล่งกำเนิดภายนอก
OSC2 /CLKOUT	15	เอาท์พุท	-	-เป็นขาสำหรับส่งสัญญาณนาฬิกาออก -สัญญานที่ออกจากขานี้จะมีความถี่เท่ากับ 1/4 ของความถี่ที่ขา OSC1/CLKIN -หากป้อนสัญญาณนาฬิกาโดยใช้คริสตอลให้ต่อกับกับขา 1 ของคริสตอลหรือเซรามิกเรโซเนเตอร์ -หากป้อนสัญญาณโดยใช้ RC ขานี้ให้ปล่อยลอยไว้
MCLR	4	อินพุต	ชมิตทริกเกอร์	-เป็นขาสำหรับรับสัญญาณรีเซ็ตโดยทำงานที่ลอจิก "0" -เป็นขารับแรงดันสำหรับการโปรแกรมหรือเขียนข้อมูลลงในไมโครคอนโทรเลอร์
ขาสัญญาณพอร์ต A				
RA0	17	อินพุต/เอาท์พุต	TTL	-เป็นขาอินพุต -เอาท์พุต 2ทิศทางทุกขา
RA1	18	อินพุต/เอาท์พุต	TTL	
RA2	1	อินพุต/เอาท์พุต	TTL	
RA3	2	อินพุต/เอาท์พุต	TTL	
RA4/TOCKI	3	อินพุต/เอาท์พุต	ชมิตทริกเกอร์	-เฉพาะขานี้ใช้เป็นขารับสัญญาณนาฬิกาให้แก่ TMR0 ด้วย
ขาสัญญาณพอร์ต B				
RB0/INT	6	อินพุต/เอาท์พุต	TTL/ชมิตทริกเกอร์	-เป็นขาอินพุต -เอาท์พุต 2 ทิศทางทุกขา -ขา RB0/INT ใช้เป็นขารับสัญญาณ
RB1	7	อินพุต/เอาท์พุต	TTL	
RB2	8	อินพุต/เอาท์พุต	TTL	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RB3	9	อินพุต/เอาท์พุต	TTL	อินเตอร์รัปด้วย
RB4	10	อินพุต/เอาท์พุต	TTL	
RB5	11	อินพุต/เอาท์พุต	TTL	-ขา RB4-RB7 ใช้เป็นขาที่ทำให้เกิดสัญญาณอินเตอร์รัปได้โดยทำการเปลี่ยนแปลงลอจิกที่ขาเหล่านี้
RB6	12	อินพุต/เอาท์พุต	TTL/ชมิตทริกเกอร์	-ขา RB6 ยังใช้เป็นขารับสัญญาณนาฬิกาของการโปรแกรมแบบอนุกรมด้วย ในขณะที่ขา RB7 ใช้เป็นขารับข้อมูลของการโปรแกรมแบบอนุกรม
RB7	13	อินพุต/เอาท์พุต	TTL/ชมิตทริกเกอร์	
ขาไฟเลี้ยง				
Vss	5	ขาต่อไฟเลี้ยง	-	ต่อกับกราวด์
Vdd	14	ขาต่อไฟเลี้ยง	-	ต่อกับไฟเลี้ยงบวกตั้งแต่ 2 - 6 V

3.3 การสื่อสารข้อมูลแบบอนุกรม

มีรูปแบบการส่งข้อมูลตามรูปที่ 3.3 ซึ่งประกอบด้วย บิตเริ่มต้น(start bit),บิตข้อมูล(data bit)และบิตหยุด(stop bit)

รูปที่ 3.3 รูปแบบการสื่อสารข้อมูลอนุกรมของ PIC16F84

ซึ่งจังหวะในการรับและส่งข้อมูลจะต้องสอดคล้อง จึงต้องมีการกำหนดอัตราการรับส่งข้อมูลซึ่งเรียกว่า อัตราบอด หรือ บอดเรต (baud rate) มีหน่วยเป็นบิตต่อวินาที (bit per second : bps) จากการกำหนดรูปแบบข้อมูลในรูปที่ 3.3 ข้อมูลในแต่ละบิตมีคาบเวลา 104 ไซเกิลของสัญญาณนาฬิกาภายในของไมโครคอนโทรลเลอร์เมื่อมีสัญญาณนาฬิกาภายในมีความถี่เท่ากับ ¼ เท่าของความถี่จากแหล่งกำเนิดภายนอก ซึ่งในที่นี้คือคริสตอลค่า 4 MHz ดังนั้นความถี่ของสัญญาณนาฬิกาภายใน PIC16F84 จึง

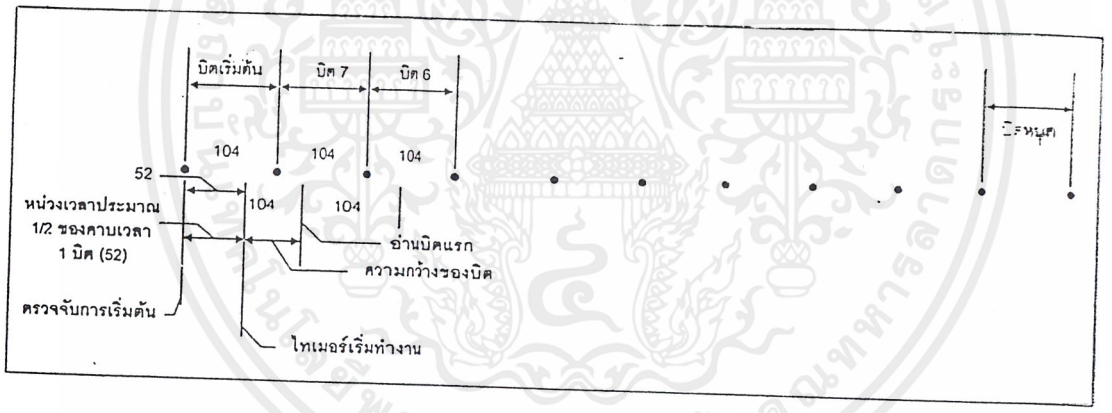
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ 1 MHz คาบเวลา 1 ไซเกิลจึงเทากับ 1 ไมโครวินาที ดังนั้นคาบเวลาของข้อมูล 1 บิตจึงเท่ากับ 104 ไมโครวินาที สามารถคำนวณอัตราบอดได้เท่ากับ 9,615 บิตต่อวินาทีโดยประมาณ

3.3.1 กระบวนการรับส่งข้อมูล

พิจารณารูปที่ 3.3 ในภาวะที่ยังไม่มีการรับส่งข้อมูล สถานะลอจิกจะเป็น "1" ค้างอยู่ตลอดเวลา ซึ่งในที่นี้จะเรียกว่าสถานะนี้ว่า "มาร์ก" (mark) เมื่อขาที่ใช้ส่งข้อมูล (transmit data line : TD) เป็น "0" ต่อจากสถานะมาร์กยาวนานถึง 104 ไซเกิลของสัญญาณนาฬิกาภายใน จะเรียกสิ่งที่เกิดขึ้นนี้ว่าบิตเริ่มต้น อันเป็นการแจ้งให้ตัวรับเตรียมพร้อมในการรับข้อมูล

ทางด้านตัวรับเมื่อเริ่มทำงานซีพียูจะวนการทำงานเพื่อรอบิตเริ่มต้นจากตัวส่ง เมื่อพบหรือตรวจจับบิตเริ่มต้นได้ โปรแกรม มอนิเตอร์ของตัวรับจะควบคุมให้ซีพียูรอหรือหน่วงเวลาอีกครึ่งหนึ่งของคาบเวลาข้อมูล 1 บิต ซึ่งในที่นี้เท่ากับ 52 ไซเกิล ถ้าหากที่ขารับข้อมูล (receive data line : RD) ยังคงเป็นลอจิก "0" ซีพียูจะเริ่มทำการอ่านข้อมูลบิตแรกเข้าสู่หน่วยความจำต่อไป แต่ถ้าหากหลังจากหน่วงเวลา 52 ไซเกิลแล้ว ที่ขารับข้อมูลเกิดเป็นลอจิก "1" ขึ้นมา แสดงว่า การส่งบิตเริ่มต้นจากตัวส่งเกิดความผิดพลาด ซีพียูจะยกเลิกการรับข้อมูล แล้วกลับไปรอรับบิตเริ่มต้นใหม่อีกครั้ง ในรูปที่ 3.4 แสดงรายละเอียดของคาบเวลาการรับข้อมูลของตัวรับ



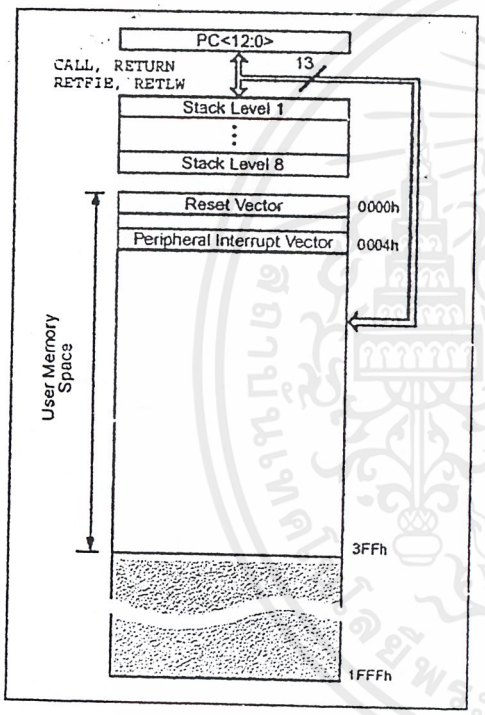
รูปที่ 3.4 ตัวอย่างคาบเวลาของการสื่อสารข้อมูลอนุกรม

หลังจากที่ตัวรับสามารถรับบิตเริ่มต้นได้แล้ว ก็จะเตรียมรับข้อมูลที่จะทยอยเข้ามา โดยบิต MSB ของข้อมูลจะถูกส่งเข้ามาก่อน จนครบ 8 บิต ตัวส่งก็จะส่งบิตหยุดไปยังตัวรับ บิตหยุดเกิดจากการทำให้ขาส่งข้อมูลเป็น "1" หรืออยู่ในสถานะมาร์กยาวนานกว่า 104 ไซเกิลของสัญญาณนาฬิกาภายในของไมโครคอนโทรลเลอร์หรือยาวนานกว่า 104 ไมโครวินาที เมื่อทำการรับส่งข้อมูลในแต่ละไบต์หรือเวิร์ดแรกเรียบร้อยแล้ว ก็จะเข้าสู่การรับส่งข้อมูลในไบต์หรือเวิร์ดต่อไป

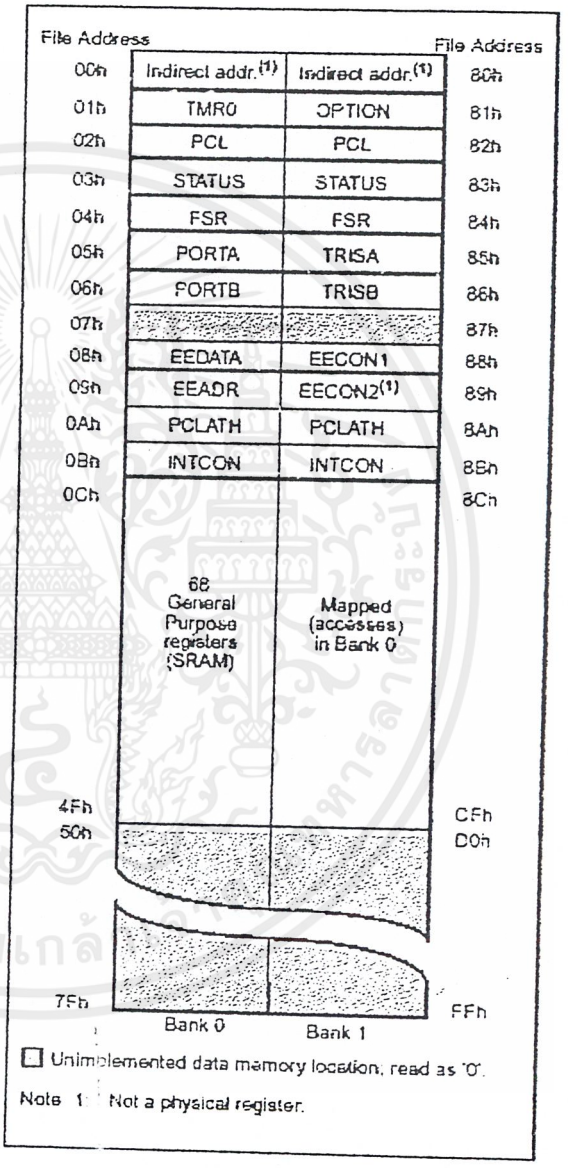
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การจัดสรรหน่วยความจำรีจีสเตอร์ควบคุมและสแต็ก

การจัดสรรหน่วยความจำภายใน PIC16F84 แบ่งเป็น 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล มีรีจีสเตอร์สำคัญ ๆ 6 ตัวและมีการใช้งานสแต็ก 8 ระดับซึ่งรายละเอียดมีดังนี้



รูปที่ 3.5 การจัดสรรหน่วยความจำข้อมูลใน PIC16F84



รูปที่ 3.6 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 หน่วยความจำโปรแกรม

เป็นหน่วยความจำที่ใช้ในการโปรแกรมควบคุมการทำงานของระบบ หรือใช้เป็นที่เก็บโปรแกรมนั่นเอง หน่วยความจำโปรแกรมใน PIC16F84 เป็นหน่วยความจำแบบแฟลช ในขณะที่ PIC16F84 ทำงานปกติหน่วยความจำนี้จะอ่านได้เพียงอย่างเดียวเท่านั้น และแก้ไขได้ก็ต่อเมื่ออยู่ในโหมดการโปรแกรมเท่านั้น

หน่วยความจำโปรแกรมมีขนาด 1 กิโลเวิร์ด (จำนวน 1 เวิร์ดของ PIC16F84 เท่ากับ 14 บิต) ได้รับการจัดสรรอยู่ในตำแหน่ง 0x000-0x3FFสามารถเขียนข้อมูลเก็บลงในพื้นที่ของหน่วยความจำนี้ได้ทุกแอดเดรสดังในรูปที่ 2.3 ซึ่งแสดงการจัดสรรแอดเดรสหน่วยความจำ

3.4.2 หน่วยความจำข้อมูล

พื้นที่หน่วยความจำข้อมูลได้รับการจัดสรรเป็นสองส่วน คือ พื้นที่ของรีจีสเตอร์ฟังก์ชั่นพิเศษ (Special Function registers : SFR) และพื้นที่ของรีจีสเตอร์ใช้งานทั่วไป (General Purpose Register : GPR) การจัดสรรหน่วยความจำทั้ง SFR และ GPR จะจัดแบ่งเป็นแบงก์ (Bank) ในแต่ละแบงก์ของ SFR จะเป็นรีจีสเตอร์ที่ทำหน้าที่แตกต่างกันออกไป การติดต่อกับหน่วยความจำข้อมูลในแต่ละแบงก์ จะต้องกำหนดค่าของบิตควบคุม RP0 และ RP1 ในรีจีสเตอร์ STATUS ในรูปที่ เป็นการจัดสรรหน่วยความจำข้อมูลของ PIC16F84

ในการใช้งานปกติ PIC16F84 จะเริ่มติดต่อหน่วยความจำข้อมูลที่แบงก์ศูนย์ก่อน ซึ่งในแบงก์นี้จะใช้เก็บค่าของรีจีสเตอร์ SFR พื้นฐานที่มีการเรียกใช้งานเสมอ และค่าของรีจีสเตอร์ GPR สำหรับการเข้าสู่รีจีสเตอร์แบงก์ 1 นั้นจะต้องทำการเซ็ตบิต RP0 ในรีจีสเตอร์ STATUS ก่อนเสมอ โดยในแบงก์ 1 นี้เป็นพื้นที่ของ SFR เฉพาะงานซึ่งมีการเรียกใช้ไม่บ่อย

3.4.3 รีจีสเตอร์ควบคุมของ PIC16F84

ใน PIC16F84 มีรีจีสเตอร์ควบคุมที่มีบทบาทสำคัญอยู่ 6 ตัวคือ STATUS, OPTION, INTCON, PCL, PCLATH และ W ซึ่งมีรายละเอียดดังนี้

- รีจีสเตอร์ STATUS

เป็นรีจีสเตอร์ที่ใช้แสดงสถานะทางคณิตศาสตร์ของ ALU, สถานะการทำงานของ PIC16F84 และใช้เป็นตัวกำหนดเลือกแบงก์ของหน่วยความจำข้อมูล การเข้าถึงรีจีสเตอร์ STATUS เพื่ออ่านหรือเขียนข้อมูลสามารถกระทำได้ด้วยวิธีการเดียวกับการอ่านและเขียนรีจีสเตอร์อื่น ๆ และถ้าหากมีการกระทำคำสั่งเกี่ยวกับคณิตศาสตร์และลอจิก บิต Z, DC และ C จะเกิดการเปลี่ยนแปลงค่าตามผลลัพธ์ที่เกิดขึ้น โดยไม่คำนึงถึงค่าเดิมที่เก็บอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจีสเตอร์ OPTION

เป็นรีจีสเตอร์ที่เก็บบิตควบคุมการอินเตอร์รัปต์จากสัญญาณภายนอก, ปรีสเกลเลอร์ของ TMR0/WDT, การพูลอัปที่ขาของพอร์ต B และควบคุมการทำงานของ TMR0 มีแอดเดรสอยู่ที่ 0x81

- รีจีสเตอร์ INTCON

เป็นรีจีสเตอร์ที่เก็บค่าการอีนาเบิลสัญญาณอินเทอร์รัปต์ มีแอดเดรสอยู่ที่ 0x0B มีบทบาทสำคัญมากในการอินเทอร์รัปต์

- รีจีสเตอร์โปรแกรมเคาท์เทอร์ PCL และ PCLATH

โปรแกรมเคาท์เทอร์ (Program Counter: PC) เป็นรีจีสเตอร์ที่มีหน้าที่ชี้ตำแหน่งแอดเดรสต่อไปของหน่วยความจำโปรแกรมที่ซีพียูจะต้องไปทำงาน

โปรแกรมเคาท์เทอร์หรือ PC ใน PIC16F84 มีขนาด 13 บิต แบ่งเป็นสองส่วนคือ รีจีสเตอร์โปรแกรมเคาน์เตอร์ไบต์ต่ำ หรือ Program Counter Low Bite (PCL) ซึ่งในส่วนนี้มีขนาด 8 บิต สามารถอ่านและเขียนค่าได้โดยตรง ในขณะที่อีกส่วนหนึ่งมีขนาด 5 บิต ไม่สามารถอ่านหรือเขียนข้อมูลได้โดยตรง ต้องอาศัยการอ่านและเขียนค่าผ่านรีจีสเตอร์ PCLATH โดยรีจีสเตอร์นี้จะทำการเก็บค่า 5 บิตบนของโปรแกรมเคาน์เทอร์แล้วผ่านค่านี้ให้แก่โปรแกรมเคาท์เทอร์ไบต์สูง (PCH) ก็ต่อเมื่อโปรแกรมเคาน์เทอร์มีการโหลดค่าใหม่เข้ามา ซึ่งจะเกิดขึ้นเมื่อกระทำคำสั่ง CALL หรือ GOTO

- รีจีสเตอร์ W

รีจีสเตอร์ W นี้เป็นรีจีสเตอร์ที่สำคัญมากของ PIC16F84 หากเปรียบกับไมโครคอนโทรลเลอร์เบอร์อื่น ๆ รีจีสเตอร์ W เทียบได้กับแอคคิวมูเลเตอร์นั่นเอง การกระทำทางคณิตศาสตร์ การโอนย้ายข้อมูล หรือการตรวจสอบข้อมูลล้วนเกี่ยวข้องกับรีจีสเตอร์ W นี้เป็นหลัก

3.4.4 สแต็ก (Stack)

ในไมโครคอนโทรลเลอร์ PIC16F84 ได้จัดสรร Stack หรือพื้นที่ในหน่วยความจำเพื่อใช้ในการเก็บค่าของโปรแกรมเคาน์เตอร์ชั่วคราวไว้ 8 ระดับในแต่ละระดับมีขนาด 13 บิต ถ้ามีการพุช (Push) 1 ครั้ง สแต็กจะทำการเก็บค่าโปรแกรมเคาน์เตอร์ (PC) ไว้ 1 ค่าสแต็กของ PIC16F84 จะสามารถรองรับได้ 8 ครั้งต่อเนื่องกัน หากมีการ Push เป็นครั้งที่ 9 ซีพียูจะนำค่าของ PC ในครั้งที่ 9 นี้เก็บลงในสแต็กที่เก็บค่า PC ครั้งที่ 1

3.5 ชุดคำสั่งของ PIC16F84

ไมโครชิปเทคโนโลยีผู้ผลิตไมโครคอนโทรลเลอร์ตระกูล PIC ได้จัดแบ่งชุดคำสั่งของ ไมโครคอนโทรลเลอร์ PIC ออกเป็น 3 กลุ่ม ดังตารางที่ 3-2 ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กลุ่มคำสั่งจัดการข้อมูลระดับไบต์กับรีจิสเตอร์ไฟล์ (Byte–oriented file register operations) เป็นกลุ่มคำสั่งที่ต้องเกี่ยวข้องหรือกระทำกับรีจิสเตอร์ไฟล์โดยตรงหรือโดยอ้อม โดยขนาดของข้อมูลที่ต้องประมวลผลอยู่ในระดับไบต์หรือ 8 บิต มีทั้งสิ้น 18 คำสั่ง
2. กลุ่มคำสั่งจัดการข้อมูลระดับบิตกับรีจิสเตอร์ไฟล์ (Bit–oriented file register operations) เป็นกลุ่มคำสั่งที่ต้องเกี่ยวข้องหรือกระทำกับรีจิสเตอร์ไฟล์โดยตรงหรือโดยอ้อม โดยขนาดของข้อมูลที่ต้องประมวลผลอยู่ในระดับบิต มีทั้งสิ้น 4 คำสั่ง
3. กลุ่มคำสั่งจัดการกับค่าคงที่และควบคุมการทำงาน (Literal and control operations) เป็นกลุ่มคำสั่งที่ผสมกันระหว่างการประมวลผลกับค่าคงที่ คำสั่งเกี่ยวกับการกระโดดไปยังโปรแกรมย่อยและคำสั่งกำหนดโหมดการทำงาน มีทั้งสิ้น 13 คำสั่ง

3.5.1 ความหมายของตัวแปรที่ควรทราบ

ในแต่ละคำสั่งของ PIC16F84 จะมีตัวแปรที่เกี่ยวข้องอยู่หลายตัว โดยจะเขียนต่อจากคำสั่งเพื่อเป็นการกำหนดหรือระบุว่า คำสั่งนั้น ๆ กระทำ หรือมีผลต่อรีจิสเตอร์หรือข้อมูลใดบ้าง สามารถสรุปรายละเอียดของความมายตัวแปรได้ดังนี้

f เป็นค่าแอดเดรสของรีจิสเตอร์ไฟล์ปกติมีค่าตั่งแต่ 0 –127 ในการเขียนโปรแกรมต้องทำการเลือกแบงก์ให้ถูกต้องเสียก่อน โดยรีจิสเตอร์ไฟล์ที่ใช้งานทั่วไป มักจะอยู่ในแบงก์ 0 ของหน่วยความจำข้อมูล

d ใช้ระบุปลายทางที่เก็บผลลัพธ์ของการกระทำคำสั่งนั้น ๆ มีค่าได้ 2 ค่าคือ “0” และ “1”

d = “ 0 ” หลังจากกระทำคำสั่งแล้วผลลัพธ์ให้เก็บไว้ในรีจิสเตอร์ W

d = “ 1 ” หลังจากกระทำคำสั่งแล้วผลลัพธ์ให้เก็บไว้ในรีจิสเตอร์ไฟล์คำสั่งนั้น ๆ

k หมายถึงค่าคงที่ใดๆในกรณีที่ใช้กับคำสั่งประมวลผลทางคณิตศาสตร์และลอจิก และยังหมายถึงตำแหน่งแอดเดรสที่ต้องการให้กระโดดไปในกรณีที่ใช้กับคำสั่งการกระโดด เช่น CALL หรือ GOTO

b หมายถึง ข้อมูลระดับบิตหรือบิตภายในรีจิสเตอร์ไฟล์ขนาด 8 บิต

W หมายถึง รีจิสเตอร์ w (แอกคิวมูเลเตอร์)

สำหรับค่าคงที่ของข้อมูลเลขฐานสิบหกที่เกี่ยวข้องในการเขียนโปรแกรมและอธิบายการทำงานของชุดคำสั่งของไมโครคอนโทรลเลอร์ตระกูล PIC จะเขียนเป็น 0x00 ตัวเลข 2 ตัวหลักท้ายจะเป็นค่าของเลขฐานสิบหก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3-2 ชุดคำสั่งของ PIC16F84

กลุ่มคำสั่งจัดการข้อมูลระดับไบต์ของรีจีสเตอร์ไฟล์			
นีโมนิก โอเปอร์แรนด์	รายละเอียด	ไซเคิล	บิตที่เกี่ยวข้อง
addwf f,d	w+f	1	C,DC,Z
andwf w,f	w and f	1	Z
clrf f	เคลียร์ค่าของ f	1	Z
clrw -	เคลียร์ค่าของ w	1	Z
comf f	คอมพลีเมนต์ f	1	Z
decf f,d	ลดค่า f	1	Z
decfsz f,d	ลดค่า f ,ข้ามไปหนึ่งแอดเดรสถ้าค่าเป็น 0	1(2)	ไม่มี
incf f,d	เพิ่มค่า f	1	Z
incfsz f,d	เพิ่มค่า f ,ข้ามไปหนึ่งแอดเดรสถ้าค่าเป็น 0	1(2)	ไม่มี
iorwf f,d	w OR f	1	Z
movf f	โอนย้ายข้อมูลไปยัง f	1	Z
movwf f,d	โอนย้ายข้อมูลจาก w ไปยัง f	1	ไม่มี
nop -	ไม่มีการทำงาน	1	ไม่มี
rlf f,d	เลื่อนข้อมูลไปทางซ้ายผ่าน C	1	C
rrf f,d	เลื่อนข้อมูลไปทางขวาผ่าน C	1	C
subwf f,d	f-w	1	C,DC.Z
swapf f,d	สลับค่า 4 บิตบนและ 4 บิตล่างของ f	1	ไม่มี
xorwf f,d	w XOR f	1	Z
กลุ่มคำสั่งจัดการข้อมูลระดับบิตของรีจีสเตอร์ไฟล์			
bcf f,b	เคลียร์บิตของ f	1	ไม่มี
bsf f,b	เซ็ตบิตของ f	1	ไม่มี
btfsc f,b	ตรวจสอบบิต ข้ามหนึ่งแอดเดรสถ้าเป็น 0	1(2)	ไม่มี
btfss f,b	ตรวจสอบบิต ข้ามหนึ่งแอดเดรสถ้าเป็น 1	1(2)	ไม่มี
กลุ่มคำสั่งจัดการกับข้อมูลค่าคงที่และควบคุมการทำงาน			
addlw k	ค่าคงที่+w	1	C,DC,Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

andlw k	ค่าคงที่ AND w	1	Z
call k	เรียกโปรแกรมย่อยที่มีแอดเดรสเท่ากับ k	2	ไม่มี
clrwdt -	เคลียร์ค่าของ WDT	1	TO,PD
goto k	กระโดดไปยังแอดเดรสที่กำหนด	2	ไม่มี
iorlw k	ค่าคงที่ OR w	1	Z
movlw k	นำค่าคงที่เก็บใน w	1	ไม่มี
retfie -	ออกจากโปรแกรมย่อยบริการอินเทอร์รัปต์	2	ไม่มี
retlw k	กลับสู่โปรแกรมด้วยค่าคงที่ใน w	2	ไม่มี
return -	ออกจากโปรแกรมย่อย	2	ไม่มี
sleep -	เข้าสู่โหมด sleep	1	TO,PD
sublw k	ค่าคงที่ – w	1	C,DC,Z
xorlw k	ค่าคงที่ XOR w	1	Z

หมายเหตุ ในคอลัมน์ของไซเคิลที่มีวงเล็บ หมายถึงว่าเมื่อทำการตรวจสอบเงื่อนไขถ้าถูกต้องจะใช้เวลา 2 ไซเคิล โดยในไซเคิลที่ 2 จะกระทำคำสั่งในลักษณะ nop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

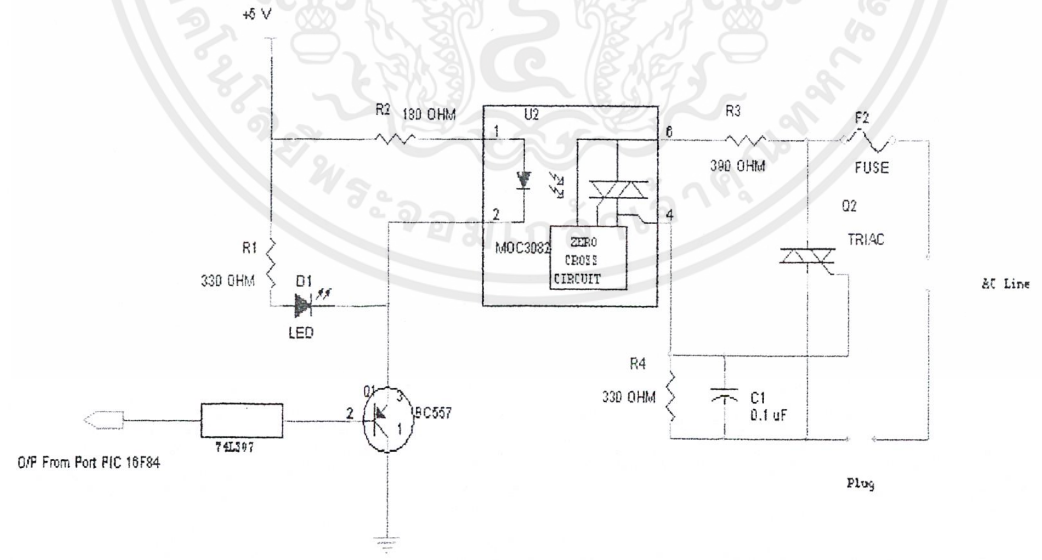
บทที่ 4
ระบบเตือนภัยภายในบ้าน

ในส่วนของระบบเตือนภัยภายในบ้านซึ่งถูกควบคุมการทำงานโดย PIC 16F84 จะถูกแบ่งออกเป็น 3 วงจรดังต่อไปนี้

1) วงจรเชื่อมต่อเพื่อควบคุมอุปกรณ์ไฟฟ้า
2) วงจรตัวส่งและตัวรับสัญญาณอินฟราเรด(วงจรเซนเซอร์)

4.1 วงจรเชื่อมต่อเพื่อควบคุมอุปกรณ์ไฟฟ้า

ในการควบคุมอุปกรณ์ไฟฟ้าซึ่งใช้แรงดันไฟฟ้ากระแสสลับ 220 โวลท์ ไมโครคอนโทรลเลอร์ไม่สามารถควบคุมได้โดยตรง จึงต้องใช้อุปกรณ์ โฟโต้ไตรแอค และ ไตรแอค (Triac) เพื่อทำการควบคุมอุปกรณ์ไฟฟ้า โดยจะส่งสัญญาณควบคุมมาทางพอร์ตของไมโครคอนโทรลเลอร์ แต่เนื่องจากพอร์ตของไมโครคอนโทรลเลอร์มีกระแสไม่เพียงพอ จึงต้องทำการเพิ่มกระแสโดยต่อบัฟเฟอร์(Buffer) และ Pull-up resistors ซึ่งในการควบคุมอุปกรณ์ไฟฟ้าได้กำหนดไว้จำนวน 2 ชุดด้วยกัน ส่วนควบคุมอุปกรณ์ไฟฟ้าหรือที่เรียกว่า โซลิดสเตทรีเลย์ (Solid State Relay) ซึ่งมีโครงสร้างประกอบด้วยส่วนต่าง ๆ คือ ส่วนควบคุมตัวเชื่อมโยงทางแสง ส่วนป้องกันทรานเซี้ยน และ ไตรแอค ดังรูปที่ 4.1



รูปแสดงวงจรควบคุมไฟ 220 โวลท์

รูปที่ 4.1 แสดงวงจรควบคุมไฟ 220 โวลต์

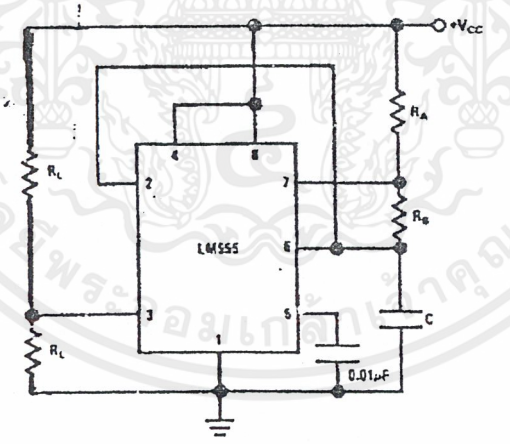
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46224

4.2 วงจรตัวส่งและตัวรับสัญญาณอินฟราเรด

วงจรอินฟราเรดนี้ใช้เป็นอุปกรณ์ตรวจสอบสถานะบริเวณประตูหรือหน้าต่าง โดยจะติดตั้งตัวส่งสัญญาณอินฟราเรดและตัวรับสัญญาณระหว่างประตูและหน้าต่าง แล้วติดตั้งตัวรับสัญญาณอินฟราเรดอีกด้านของประตูหรือหน้าต่าง เพื่อคอยตรวจสอบสัญญาณอินฟราเรดที่ส่งมาจากตัวส่ง ซึ่งในสภาวะปกติไม่มีวัตถุอะไรมาบัง ระหว่างตัวส่ง กับตัวรับ แสดงว่าไม่มีผู้ผ่านเข้าออก แต่ถ้าตัวรับไม่สามารถรับสัญญาณที่ส่งมาจากตัวส่งได้ แสดงว่ามีผู้ผ่านเข้าออก เราสามารถใช้ไมโครคอนโทรลเลอร์ตรวจสอบสถานะเอาท์พุทของตัวรับ

ในส่วนการออกแบบวงจรนี้จะใช้ไอซี 555 สร้างสัญญาณพัลส์โดยการเก็บประจุเข้าไปในตัวประจุ Ca ผ่าน Ra และ Rb เมื่อแรงดันคร่อม Ca มีค่า 2/3 ของ Vcc จะทำให้วงจรเปรียบเทียบแรงดันชุดบนทำงานซึ่งจะไปกระตุ้นให้วงจรควบคุมฟลิบฟลอบทำงานอีกต่อหนึ่ง เป็นผลให้ Ca คายประจุผ่าน Rb และขา 7 ลงกราวด์ ช่วงนี้เอาท์พุทขา 3 จะมีแรงดันต่ำและเมื่อแรงดันที่ Ca ลดลงมาถึง 1/3 ของ Vcc ก็จะไปกระตุ้นให้วงจรเปรียบเทียบชุดล่างทำงานกระตุ้นวงจรฟลิบฟลอป ทำให้แรงดันเอาท์พุทสูงขึ้น และ Ca ไม่สามารถคายประจุได้อีก เพราะทรานซิสเตอร์ภายในซึ่งทำหน้าที่เป็นสวิตช์จะอยู่ในสภาวะไม่ทำงาน ดังนั้น Ca จะเริ่มการเก็บประจุอีกจนแรงดันเท่ากับ 2/3 ของ Vcc ก็จะเริ่มทำงานแบบเดิมอีกครั้ง ซึ่งจะมีลักษณะดังรูปที่ 4.2



รูปที่ 4.2 แสดงวงจรที่ใช้คำนวณหาความถี่ของ ไอซี 555

จากลักษณะการทำงานของวงจรจะเห็นว่าช่วงความกว้างของพัลส์ที่ได้ทางเอาท์พุทจะขึ้นอยู่กับการเก็บประจุ และการคายประจุของ Ca ซึ่งจาก Data Sheet จะได้ว่า

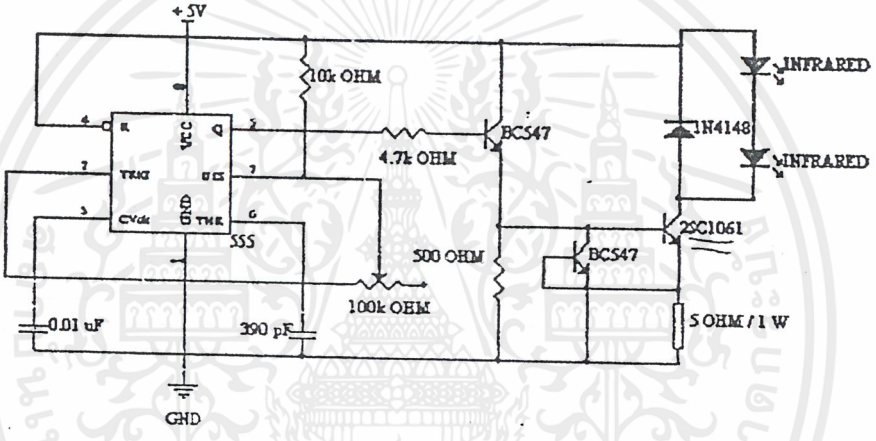
ช่วงการเก็บประจุ ; t1 = 0.693*(Ra+Rb)*Ca

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงการคายประจุ ; t2 = 0.693*Ra*Ca

ความถี่ของสัญญาณเอาท์พุท ; f = 1/(t1+t2)

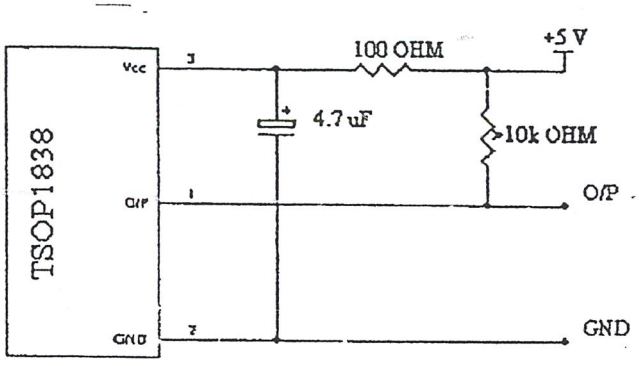
วงจรอินฟราเรดทางด้านส่งนี้จะสร้างพัลส์โดยใช้ไอซี 555 ซึ่งความถี่ที่สร้างขึ้นสามารถกำหนดได้จาก Ra, Rb และ Ca จากสูตรที่กล่าวมาแล้ว ในวงจรนี้เราจะสร้างสัญญาณพัลส์ความถี่ประมาณ 38 kHz เพื่อให้ตรงกับความถี่ของวงจรทางด้านตัวรับอินฟราเรด สัญญาณพัลส์ที่สร้างได้นี้จะถูกส่งไปยังส่วนไดรว์ โดยใช้เพาเวอร์ทรานซิสเตอร์เบอร์ 2SC 1061 ส่วน BC 547 ทำหน้าที่เป็นสวิทช์ และทำหน้าที่จำกัดกระแสที่ไหลผ่านตัวส่งอินฟราเรด ซึ่งลักษณะวงจรทางด้านตัวส่งจะเป็นดังรูปที่ 4.3



รูปที่ 4.3 แสดงวงจรอินฟราเรดทางด้านส่ง

ส่วนทางด้านของตัวรับอินฟราเรดจะใช้เป็น Photo Modules เบอร์ TSOP 1838 ที่สามารถรับสัญญาณอินฟราเรดที่ความถี่ 38 kHz ซึ่งภายในตัว Photo Modules TSOP 1838 จะประกอบด้วยส่วนรับแสงอินฟราเรด เฟสล็อคลูปที่ความถี่ 38 kHz และภาคขยายสัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



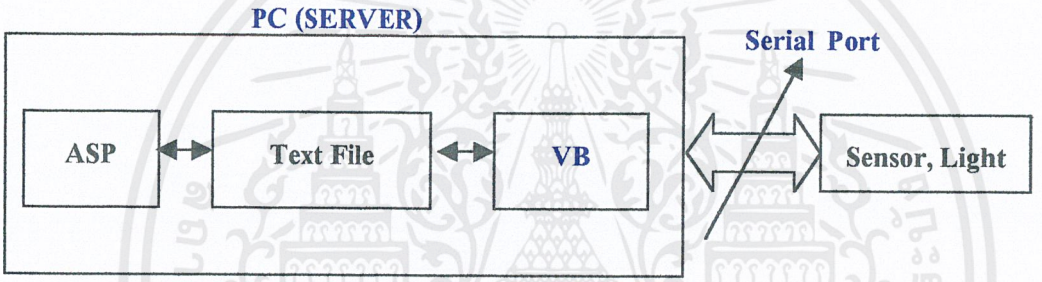
รูปที่ 4.4 แสดงวงจรอินฟราเรดทางด้านรับ

ในส่วนของตัวรับนี้จะ Active Low คือในสภาวะปกติที่สามารถรับสัญญาณอินฟราเรดจากตัวส่งได้ตามปกติ เอาท์พุทที่ได้จากตัวรับนี้จะเป็น LOW (0) แต่ถ้าตัวรับไม่สามารถรับสัญญาณอินฟราเรดจากตัวส่งได้ เอาท์พุทที่ได้จะเป็น High (1) วงจรทางตัวรับสัญญาณจะมีลักษณะดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ส่วนของคอมพิวเตอร์ที่บ้าน (Server) ที่ติดต่อกับพอร์ตอนุกรม

ในการใช้งานระบบรักษาความปลอดภัยทางคอมพิวเตอร์นั้นจะใช้งานผ่านโปรแกรมวิชวลเบสิก โดยใช้พอร์ตอนุกรม ในการเชื่อมต่อไมโครคอนโทรลเลอร์ PIC 16F84 กับคอมพิวเตอร์ เนื่องจากการเขียนโปรแกรมวิชวลเบสิกเป็นการเขียนโปรแกรมแบบ Event-Driven ซึ่งเป็นการเขียนโปรแกรมที่ต้องตอบสนองต่อเหตุการณ์ที่เกิดขึ้นมาเป็นเหตุการณ์ ๆ ไป และใช้แนวความคิดในการเขียนโปรแกรมแบบ Object Oriented programming (OOP) ทำให้สามารถสร้างแอพพลิเคชันที่มีความสลับซับซ้อนได้อย่างมีประสิทธิภาพ



Block Diagram แสดงการรับส่งข้อมูลผ่านพอร์ตอนุกรม

5.1 การเขียนโปรแกรมวิชวลเบสิกเพื่อใช้งานพอร์ตอนุกรม

คอลโทรลMSComm

สำหรับการใช้งานวิชวลเบสิกตั้งแต่เวอร์ชัน 2 เป็นต้นมา ในวิชวลเบสิกจะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยในวิชวลเบสิก เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับในวิชวลเบสิก เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือการสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send (RST) เหตุการณ์ Oncomm ของ MSComm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสามารถตรวจจับสัญญาณได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่าง ๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ตจะต้องใช้คอลโทรล MSComm มากกว่า 1ตัว เพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต โดยแอสแดรสของพอร์ตอนุกรมและแอสแดรสของการเกิดอินเทอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (property) มากมาย แต่สามารถทำความเข้าใจได้ไม่ยากดังนี้

1 . CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

object.CommPort[= value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่จะใช้คำสั่ง OpenPort

2 . Setting

ในการกำหนดและอ่านค่าอัตราบอด,พาริตี้,จำนวนบิตข้อมูล,จำนวนบิตปิดท้าย

รูปแบบการทำงาน

object.Setting [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBB เป็นค่าอัตราบอด, P เป็นค่าพาริตี้, D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนข้อมูลของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600,N,8,1”

ค่าบอดเรตมารตฐานที่ใช้กับ MSComm มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที

600 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1200 บิตต่อวินาที

2400 บิตต่อวินาที

9600 บิตต่อวินาที (ค่าปกติ)

14400 บิตต่อวินาที

19200 บิตต่อวินาที

28800 บิตต่อวินาที

38400 บิตต่อวินาที (สงวน)

56000 บิตต่อวินาที (สงวน)

12800 บิตต่อวินาที (สงวน)

256000 บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตี้มีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตี้คู่(Even)
M	ลอจิก "1" (Mark)
N	ไม่ใช้ (ค่าปกติ)
O	พาริตี้คี่ (Odd)
S	ลอจิก "0" (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4,5,6,7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ) ,1,5 และ 2

ตัวอย่างการใช้งานคำสั่ง Setting โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี้ จำนวนข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

MSComm.Setting = "9600, N, 8, 1"

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด "" เนื่องจาก ค่าที่กำหนดนี้อยู่ในรูปตัวแปร String

3. PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

object.PortOpen [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีน คือ True และ False โดย True หมายถึงการเปิดพอร์ตอนุกรม และ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากเปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากเปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9600 บิตต่อวินาที ไม่มีพาริตี้ จำนวนข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1.Settings = "9600, N, 8, 1"
MSComm1.Commport = 1
MSComm1.PortOpen = True
```

4. Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

object.Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปของไบนารี่และชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีในการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command1_Click()
Dim Instring as String
    MSComm1.InputLen = 0            'Retrieve all available data.
    If MSComm.InBufferCount Then        'Check for data.
    InString = MSComm1.Input            'Read data.
    End If
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

object.InBufferCount [=Value]

6. InBufferSize

การกำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

Object.InBufferSize [=value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1024 ไบต์

7. InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์

รูปแบบการใช้งานคำสั่ง

object.InputLen [=value]

8. InputMode

การกำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input

object.InputMode [=value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

comInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ASCI โดยจะต้องกำหนดค่าเป็น “0” และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

comInputModeBinary สำหรับข้อมูลอื่น ๆ ซึ่งจะเก็บในรูปไบนารี่รวมกันอยู่เป็นไบต์ข้อมูล

9. Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

object.Output [=value]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติเอาท์พุทสามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารี่ก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี่จะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบไบต์

10. OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

object.OutBufferCount [=value]

11. OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

object.outBufferSize [= object]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

12. ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี้

รูปแบบการใช้งานคำสั่ง

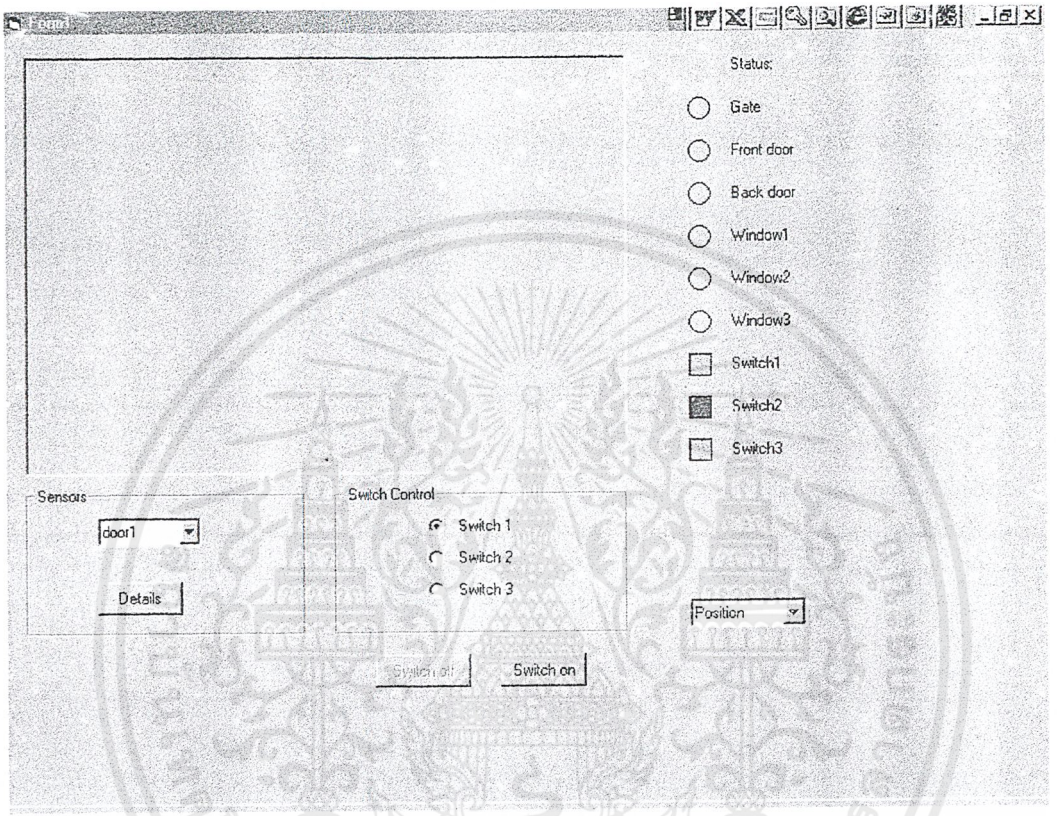
object.ParityReplace [= value]

บิตพาริตี้ เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี้ คอนโทรล MSComm จะทำการบวกบิตทุกบิตที่มีค่าลอจิก"1"ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก"1" เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตี้ไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การแสดงผลของโปรแกรมและการเขียนโปรแกรมหน้าจอควบคุมหลัก

โปรแกรมหน้าจอควบคุมหลักที่เขียนโดย Visual Basic มีรูปแบบการทำงานที่ส่วนต่าง ๆ ดังนี้



รูปที่ 5.1 แสดงหน้าจอหลักของโปรแกรม

การทำงานของโปรแกรมในส่วนหลัก ๆ มีดังนี้

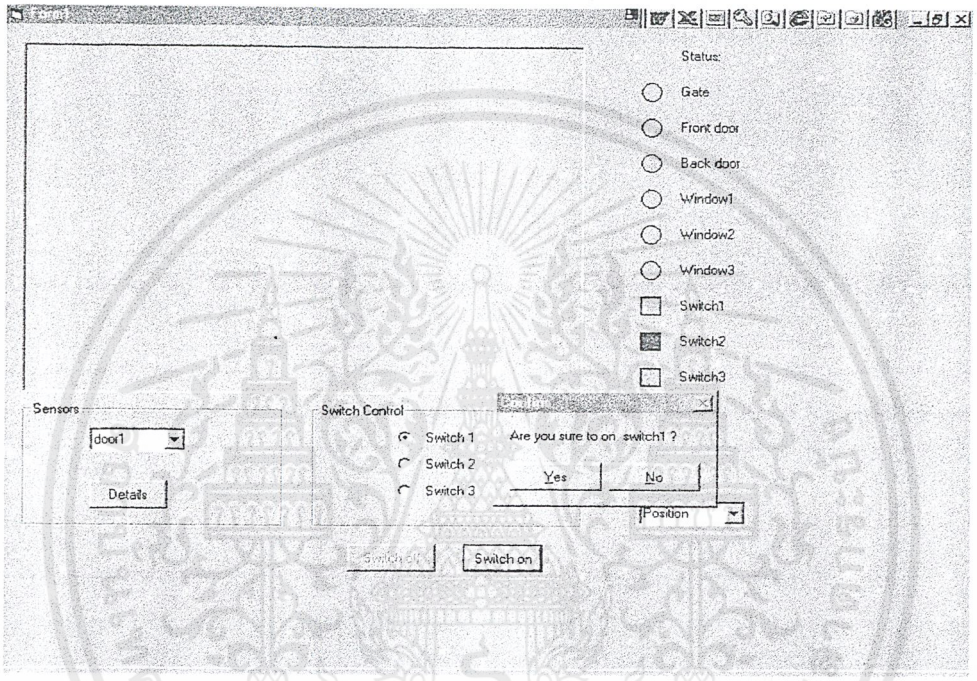
1. การรับข้อมูลจากพอร์ตอนุกรมเพื่อนำมาแสดงผล

- สำหรับสถานะของเซนเซอร์ที่ตำแหน่งต่าง ๆ จะแสดงผลโดย LED รูปวงกลม ถ้าเซนเซอร์ตัวใดทำงาน LED ที่แทนตำแหน่งนั้นจะเป็นสีเขียว ถ้าไม่ทำงานก็จะเป็นสีแดง
- สถานะของหลอดไฟ จะแทนด้วย LED รูปสี่เหลี่ยม ถ้าหลอดไฟเปิดอยู่ LED จะเป็นสีเขียว ถ้าปิดอยู่จะเป็นสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

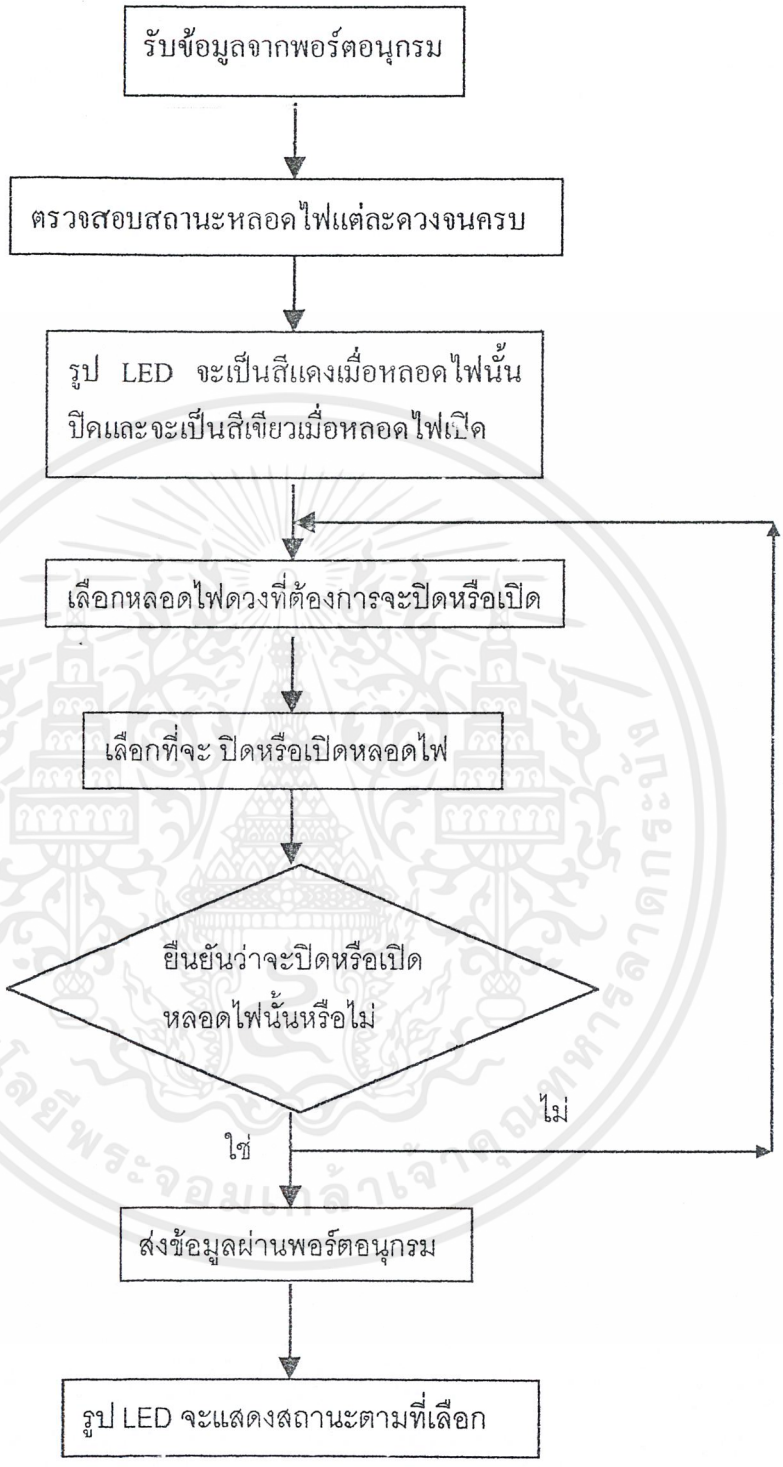
2. การสั่งงานจากหน้าจอของโปรแกรม

จะเป็นการส่งข้อมูลไปให้พอร์ตอนุกรมเพื่อสั่งให้ไมโครคอนโทรลเลอร์ ทำงานตามคำสั่งที่ส่งไปอีกทีหนึ่ง ซึ่งคำสั่งที่ส่งไปจะเป็นการสั่งเปิดหรือปิดหลอดไฟตำแหน่งต่าง ๆ (โดยไมโครคอนโทรลเลอร์จะนำข้อมูลที่ได้รับจากพอร์ตอนุกรมมาเทียบในตารางเพื่อเปลี่ยนเป็นคำสั่งต่าง ๆ)



รูปที่ 5.3 แสดงการสั่งงานปิดหลอดไฟจากหน้าจอของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดง Flow Chart การควบคุมหลอดไฟจาก PC ผ่านพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

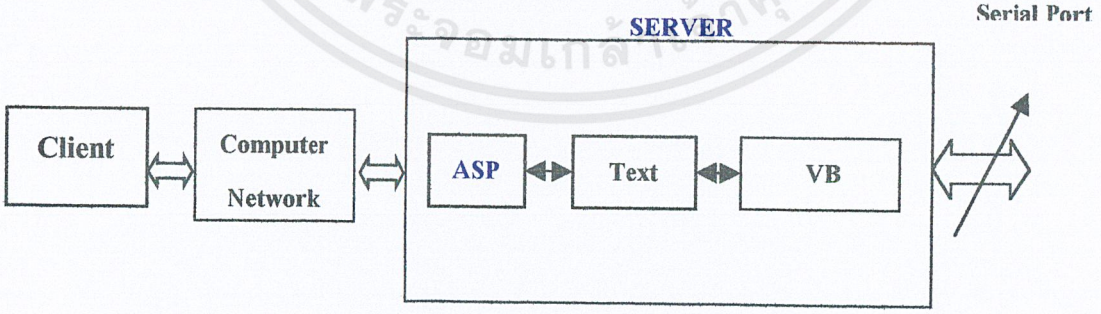
บทที่ 6
ส่วนของการรับและส่งข้อมูลผ่านหน้าเว็บ

ในการใช้งานระบบรักษาความปลอดภัยผ่านทางเครือข่ายคอมพิวเตอร์จะมีประโยชน์ในการใช้งานเป็นอย่างมาก ทั้งนี้ไม่ว่าเราจะอยู่ที่ใดที่มีเครือข่ายเชื่อมโยงอยู่ เราก็สามารถตรวจสอบและควบคุมอุปกรณ์ต่าง ๆ ในระบบรักษาความปลอดภัยได้ และถ้าเป็นเครือข่ายอินเตอร์เน็ตที่มีเครือข่ายขนาดใหญ่ครอบคลุมไปทั่วโลก จะทำให้การใช้งานระบบรักษาความปลอดภัยสามารถใช้งานได้กว้างขวางมากขึ้น

การออกแบบระบบรักษาความปลอดภัยให้สามารถใช้ในเครือข่ายคอมพิวเตอร์ได้นั้นจะต้องมีความรู้พื้นฐานดังนี้

6.1 การจำลองให้คอมพิวเตอร์(ที่ต่อกับระบบรักษาความปลอดภัย) ให้เป็น web server

ในระบบเครือข่ายอินเตอร์เน็ตจะใช้รูปแบบการติดต่อแบบ Client-Server ทั้งหมด คือ เครื่องลูกข่าย (Client) จะร้องขอการบริการใด ๆ จากเครื่องแม่ข่าย (Server) และเมื่อเครื่องแม่ข่ายรับการร้องขอจากเครื่องลูกข่ายแล้ว ก็จะทำการให้บริการแก่เครื่องลูกข่าย ตามที่เครื่องลูกข่ายร้องขอมาดังรูป



รูปที่ 6.1 แสดงการติดต่อแบบ client-server ของเครือข่ายอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web server เป็นโปรแกรมที่อยู่ในเครื่องฝั่ง Server ทำหน้าที่ในการรับคำร้องขอ และทำการประมวลผล แล้วส่งข้อมูลเว็บไปให้ฝั่ง Client นั่นคือโปรแกรม Web server จะทำหน้าที่ให้บริการเว็บนั่นเอง

ในโครงงานนี้ได้ใช้โปรแกรม IIS (Internet Information Service) ในการทำให้เครื่องคอมพิวเตอร์ที่ต่อกับระบบรักษาความปลอดภัยเป็นเครื่องที่ให้บริการหรือ Web Server เพื่อให้เครื่องในเครือข่ายสามารถใช้งานระบบรักษาความปลอดภัยได้

6.2 ส่วนที่เป็นโปรแกรมที่ใช้รับและส่งข้อมูลผ่านพอร์ตและมาแสดงผลบนหน้าเว็บ

การรับส่งข้อมูลระหว่าง PC กับส่วนของไมโคร คอนโทรลเลอร์ในระบบเตือนภัยจะผ่านทางพอร์ตอนุกรม (Serial Port) โดยใช้โปรแกรม Visual Basic 6.0

การรับคำสั่งและแสดงผลทางหน้าWeb รวมทั้งรับคำสั่งจากหน้าWeb จะใช้โปรแกรม ASP

พื้นฐานของ ASP

ASP ย่อมาจาก Active Server Pages เป็นรูปแบบการสร้างเว็บรูปแบบหนึ่งที่นิยมใช้กันมากในการสื่อสารทางอินเตอร์เน็ต ซึ่งความสามารถของ ASP ไม่เป็นเพียงหน้าเอกสารให้เราอ่านเท่านั้น แต่ยังสามารถเพิ่มลูกเล่นต่างๆมากมาย เช่น การsearchข้อมูล, การคำนวณค่าใช้จ่ายในการซื้อขายผ่านทางอินเตอร์เน็ต, การสร้าง โปรแกรมchat และอื่นๆอีกมากมายไม่จำกัด

หลักการทำงานของ ASP

ASP จะทำงานอยู่บน Server โดย Server จะประมวลผลข้อมูลที่รับมาจากเครื่อง Client แล้วส่งผลลัพธ์การประมวลผลกลับไปให้เครื่อง Client แสดงผลทาง Browser

องค์ประกอบของ ASP จะประกอบด้วย 3 ส่วนคือ

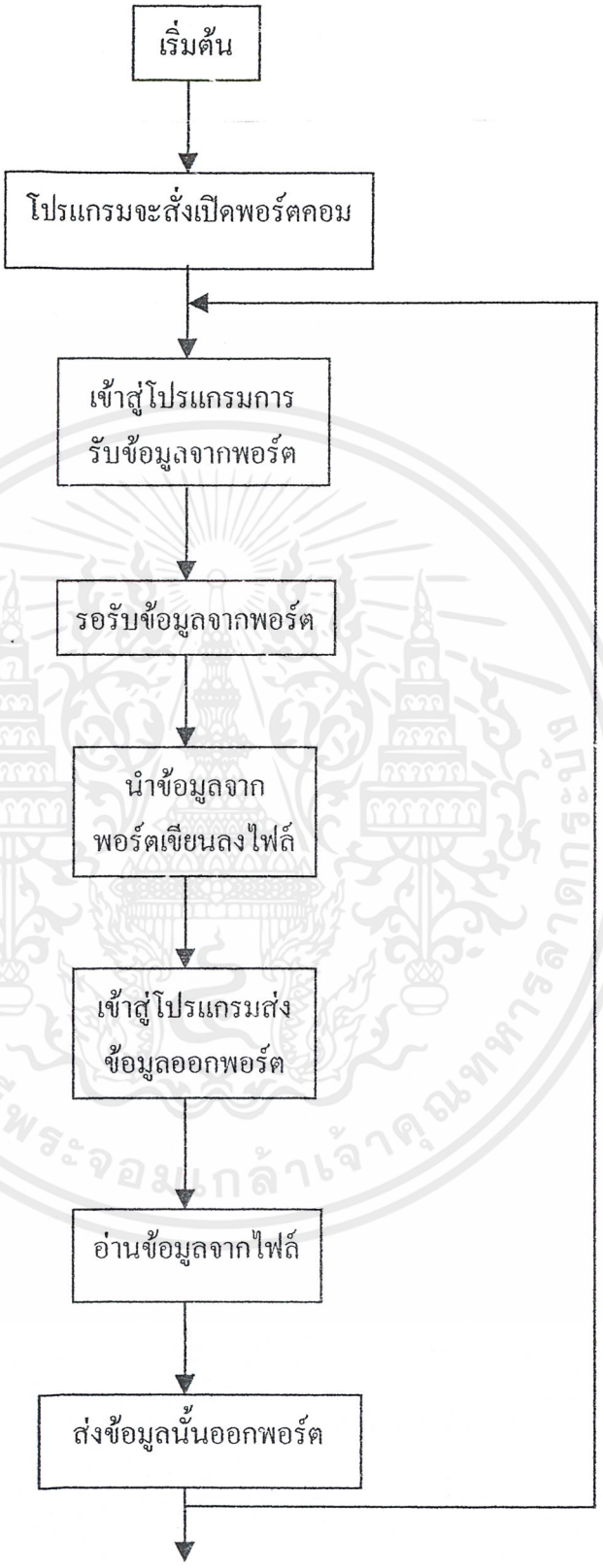
- แท็ก HTML ซึ่งเป็นส่วนที่รูปแบบคงที่ไม่เปลี่ยนแปลง (static) ส่วนนี้จะใช้ภาษา HTML ในการเขียนโปรแกรม
- ส่วนที่เปลี่ยนแปลงตามการคำนวณ (dynamic) ซึ่งส่วนนี้จะใช้ภาษา Script ชนิด Server-Side Scripting (เป็นโปรแกรมที่สามารถทำงานบน Web Server เช่น ASP, CGI ซึ่งเราสามารถเขียนด้วยภาษาต่างๆ เช่น JavaScript, VBScript, Perl เป็นต้น)
- ส่วนของ Active X และ Object ที่สร้างขึ้นมาทำงานเฉพาะอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการใช้ ASP

1. ASP ทำให้เว็บของคุณเป็นแบบไดนามิค (Dynamic) นั่นคือรูปแบบที่เว็บแสดงผลออกมานั้นสามารถเปลี่ยนแปลงได้ตามข้อมูลที่ ASP ได้รับ เช่น ตัวอย่างการsearchข้อมูลในเว็บ ผลลัพธ์จะเปลี่ยนไปตามคำที่เราใช้search
2. เพิ่มความเร็วในการดูเว็บ เนื่องจากการดูเว็บนี้นมักจะเสียเวลาการรอข้อมูลที่มาจากอินเตอร์เน็ต ยิ่งข้อมูลยิ่งมากยิ่งรอนาน ซึ่งASPสามารถช่วยในจุดนี้ได้ กล่าวคือ ASP จะทำการคำนวณต่างๆจนเสร็จแล้วส่งผลลัพธ์เฉพาะที่ต้องการเท่านั้น ทำให้ปริมาณการส่งข้อมูลน้อยลง เราก็จะเสียเวลารอข้อมูลน้อยลงและสามารถดูเว็บได้เร็วขึ้น
3. เพิ่มความปลอดภัยให้ระบบของเรา ในการเขียนโปรแกรมต่างๆ บางครั้งเราต้องอ้างถึงไดเร็กทอรี่ที่เก็บฐานข้อมูล ซึ่งการใช้ ASP ไดเร็กทอรี่ต่างๆจะไม่ถูกแสดงที่ฝั่งผู้ดูเว็บ จะแสดงเฉพาะผลลัพธ์ที่เอามาจากฐานข้อมูลเท่านั้น ทำให้ผู้ดูเว็บไม่สามารถรู้ถึงโครงสร้างของเว็บเราได้
4. ลดปัญหาความสามารถของเครื่องที่ใช้ดูเว็บ เนื่องจาก ASP จะส่งผลลัพธ์สุดท้ายมาแสดงผลเท่านั้น ดังนั้นไม่ว่าเครื่องของคุณจะทันสมัยหรือล้าสมัยเพียงใด ก็ไม่ทำให้เวลาที่ใช้เปิดดูเว็บต่างกันมากนัก เพราะว่าการประมวลผลทั้งหมดจะเสร็จสิ้นที่ฝั่ง Server แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 Flow Chart แสดงการรับส่งข้อมูลผ่านพอร์ตอนุกรมในโปรแกรม Visual Basic

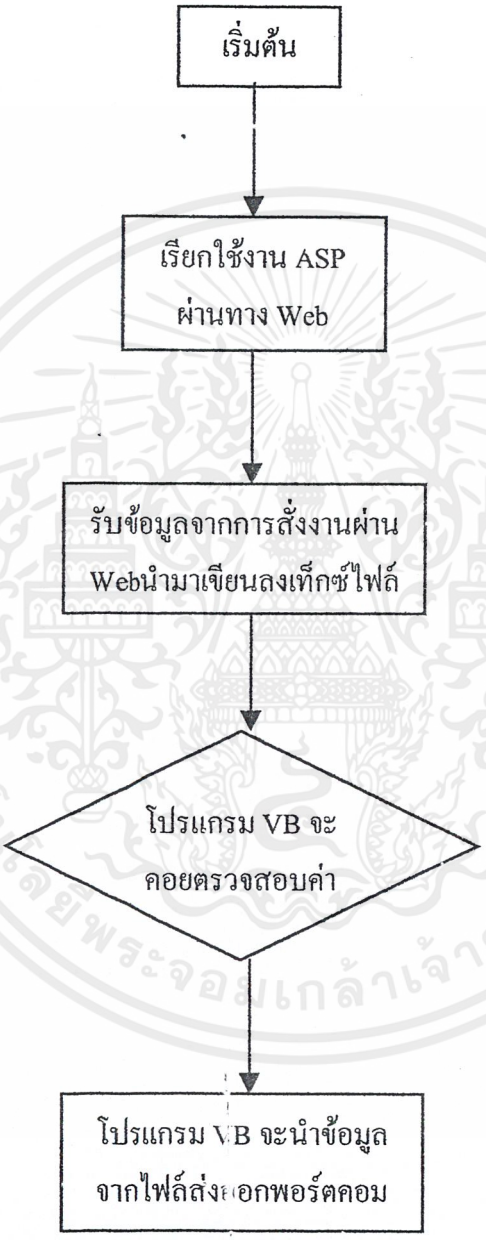
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>การแสดงสถานะอุปกรณ์ไฟฟ้าผ่านหน้าเว็บ</u>

รูปที่ 6.3 Flow Chart แสดงการทำงานของโปรแกรมที่ใช้แสดงสถานะอุปกรณ์ไฟฟ้าของเครื่อง Server ผ่านทางหน้า Web

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

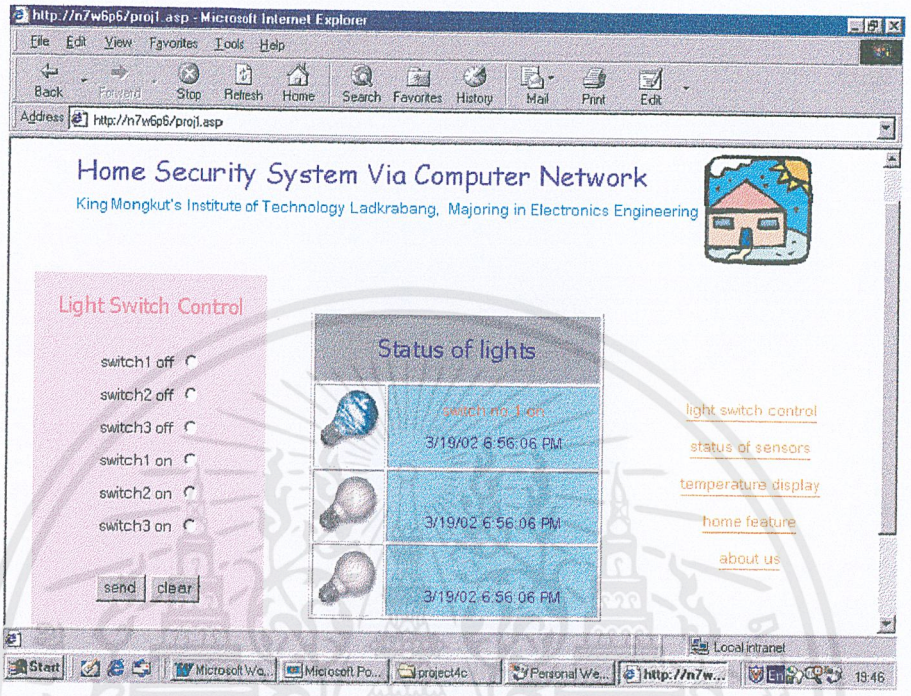
การควบคุมอุปกรณ์ไฟฟ้าผ่านหน้า Web



รูปที่ 6.4 Flow Chart แสดงการทำงานของโปรแกรมที่ใช้ควบคุมอุปกรณ์ไฟฟ้าของเครื่อง Server ผ่านทางหน้า Web

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การใช้งานระบบรักษาความปลอดภัยผ่านทางหน้าเว็บ



รูปที่ 6.5 แสดงสถานะของหลอดไฟและการเปิด-ปิดหลอดไฟผ่านทางหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

การทดลองแบ่งเป็น 2 ส่วน

1) การทดลองโปรแกรมที่ตัว Server
2) การทดลองโปรแกรมจากตัว Client ผ่านเครือข่ายคอมพิวเตอร์

7.1 การทดลองที่ Server

แบ่งการทดลองเป็นส่วนต่าง ๆ ดังนี้

ทดลองการเปิดปิดหลอดไฟ

จะทำการเปิดปิดหลอดไฟโดยสั่งงานจากหน้าจอคอมพิวเตอร์ แล้วดูผลลัพธ์ที่หลอดไฟบนวงจรการทดลองว่าทำงานตามที่สั่งงานหรือไม่ และดูสถานะ LED ของหลอดไฟบนหน้าจอโปรแกรมก่อนการทดลองและหลังการทดลองว่าเป็นอย่างไร(สถานะเปิด LED มีสีแดง สถานะปิด LED มีสถานะเป็นสีเขียว)

ตารางที่ 7.1 แสดงผลการควบคุมอุปกรณ์ไฟฟ้า

การทดลอง	สถานะของ LED บนหน้าจอก่อนการทดลอง	สถานะของ LED บนหน้าจอหลังการทดลอง	ผลของวงจรที่ใช้ทดลอง
ปิดหลอดไฟของSW1	ติด	ดับ	ดับ
เปิดหลอดไฟของSW1	ดับ	ติด	ติด
ปิดหลอดไฟของSW2	ติด	ดับ	ดับ
เปิดหลอดไฟของSW2	ดับ	ติด	ติด
ปิดหลอดไฟของSW3	ติด	ดับ	ดับ
เปิดหลอดไฟของSW3	ดับ	ติด	ติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองการทำงานของ SENSOR

โดยใช้อุปกรณ์ SENSOR 4 ตัว แบ่งไว้จุดละ2ตัว จำลองก่อนและหลังเข้าประตู จากนั้นให้วัตถุผ่าน SENSOR หากมีสถานะเปลี่ยนแปลง จะแสดงสถานะทางหน้าจอ เป็นจุดสีแดง(หากปรกติจะเป็นสีเขียว) และแยกบอกจุด ที่ติดอุปกรณ์ SENSOR แยกเป็นเข้าหรือออกพร้อมทั้งนับจำนวนครั้งที่มีการเข้าหรือออก โดยจะการทดลองติดต่อกันพร้อมทั้งดูผลทางหน้าจอควบคู่กันไป

ตารางที่7.2 ตารางผลการทำงานอุปกรณ์ SENSOR

การทดลองครั้งที่	SENSORที่ผ่านครั้งแรก	SENSORที่ผ่านครั้งที่ 2	ผลที่แสดง
ครั้งที่ 1	SENSOR ก่อนเข้าประตู	SENSOR หลังเข้าประตู	ประตู เข้า 1
ครั้งที่ 2	SENSOR ก่อนเข้าประตู	SENSOR หลังเข้าประตู	ประตู เข้า 2
ครั้งที่ 3	SENSOR หลังเข้าประตู	SENSOR ก่อนเข้าประตู	ประตู ออก 1
ครั้งที่ 4	SENSOR หลังเข้าประตู	SENSOR ก่อนเข้าประตู	ประตู ออก 2
ครั้งที่ 5	SENSOR ก่อนเข้าหน้าต่าง	SENSOR หลังเข้าหน้าต่าง	หน้าต่าง เข้า 1
ครั้งที่ 6	SENSOR ก่อนเข้าหน้าต่าง	SENSOR หลังเข้าหน้าต่าง	หน้าต่าง เข้า 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
สรุปและวิจารณ์โครงงาน

การออกแบบระบบรักษาความปลอดภัยภายในบ้านนี้เป็นแค่แนวความคิดพื้นฐานโดยเราสามารถที่จะพัฒนาต่อไปได้อีก โดยมีแนวคิดดังนี้

1. พัฒนาโปรแกรมการรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ให้มีระยะเวลาการรับส่งข้อมูลให้น้อยที่สุด
2. อาจมีการเพิ่มเติมหรือพัฒนาประสิทธิภาพของอุปกรณ์ที่ใช้ในการรักษาความปลอดภัยต่างๆ ที่ถูกควบคุมโดยไมโครคอนโทรลเลอร์
3. อาจจะพัฒนาโดยนำมาเชื่อมต่อกันกับบ้านหลายๆหลัง เช่นหมู่บ้าน แล้วใช้ Server เพียงเครื่องเดียวควบคุมและเชื่อมต่อกับอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

----------ASP PROJECT----------

```
<%
option explicit
%>
<table align="center">
<tr><td><font face="Comic Sans MS" size="4" color="#3333ff">Home Security System Via
Computer Network</font><br>
<font color=DeepSkyBlue><center>King Mongkut's Institute of Technology Ladkrabang,
 :Majoring in Electronics Engineering</center></font><br>
</td>
<td align="center"><img src='pict/hill.gif' width='100' height="100" ></td></tr><br>

<table align='center' border='0' cellspacing="8" width="750">
<tr><td align="center" bgcolor="#ffccff" colspan="4">
<font face="Arial" size="2" color="#ff3399">Light Switch Control</font>
<form name="form1" action = 'proj1.asp' method ='post'>
switch1 off <input type = 'radio' name= 'sw1' value='switch no.1 off'><br>
switch2 off <input type = 'radio' name= 'sw2' value='switch no.2 off'><br>
switch3 off <input type = 'radio' name= 'sw3' value='switch no.3 off'><br>
switch1 on  <input type = 'radio' name= 'sw1' value='switch no.1 on'><br>
switch2 on  <input type = 'radio' name= 'sw2' value='switch no.2 on'><br>
switch3 on  <input type = 'radio' name= 'sw3' value='switch no.3 on'><br><br>
<input type= 'submit' value='send'> <input type='reset' value='clear'>
</form>
</td>

<td align="center" >
<table align='center' border=1 width='260'>
<tr align="center" bgcolor=darkgray ><td height=60 colspan = 4 ><font face="Arial" size="3"
color=Blue>Status of lights</font></td></tr><br>
        <tr><td align='center' width="60" height="60">
                <%if request.form("sw1")="switch no.1 on" then%>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<img src='pict/deco_60.jpg' width='60' height="60">
</td>
<td bgcolor=PaleTurquoise align='center' height="60"><%response.write "<font color 'red'>"&request.form("sw1")&"</font>"%><br>
<%response.write ("<font color='blue'>"&now&"</font>")%>
</td>
<%else%>
<img src='pict/cool_60.jpg' width='60' height="60" ></td>
<td bgcolor=PaleTurquoise align='center' height="60"><%response.write "<font color='red'>"&request.form("sw1")&"</font>"%><br>
<%response.write ("<font color='blue'>"&now&"</font>")%><br>
</td>
<%end if%>
</tr>
<tr><td align='center' width="60" height="60">
<%if request.form("sw2")="switch no.2 on" then%>
<img src='pict/deco_60.jpg' width='60' height="60"></td>
<td bgcolor=PaleTurquoise align='center' height="60"><%response.write "<font color='red'>"&request.form("sw2")&"</font>"%><br>
<%response.write ("<font color='blue'>"&now&"</font>")%>
</td>
<%else%>
<img src='pict/cool_60.jpg' width='60' height="60" ></td>
<td bgcolor=PaleTurquoise align='center' height="60"><%response.write "<font color='red'>"&request.form("sw2")&"</font>"%><br>
<%response.write ("<font color='blue'>"&now&"</font>")%><br>
</td>
<%end if%>
tr>
<tr><td align='center' width="60" height="60">
<%if request.form("sw3")="switch no.3 on" then%>
<img src='pict/deco_60.jpg' width='60' height="60"></td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                <td bgcolor=PaleTurquoise align='center' height="60"><%response.write "<font
color='red'>"&request.form("sw3")&"</font>"%><br>
                <%response.write ("<font color='blue'>"&now&"</font>")%>
                </td>
                <%else%>
                <img src='pict/cool_60.jpg' width='60' height="60" ></td>
                <td bgcolor=PaleTurquoise align='center' height="60"><%response.write "<font
color='red'>"&request.form("sw3")&"</font>"%><br>
                <%response.write ("<font color='blue'>"&now&"</font>")%><br>
                </td>
                <%end if%>
        </tr>
</table><br>
</td>


<td align="center"  width = "180">
<br><table>
        <tr><td align="center"><a href="proj1.asp"><font color="#ff9900">light switch control
</font></a></td></tr>
        <tr><td align="center"><a href="test.asp"><font color="#ff9900">status of sensors
</font></a></td></tr>
        <tr><td align="center"><a href="temp.asp"><font color="#ff9900">temperature display
</font></a></td></tr>
        <tr><td align="center"><a href="home_feature.asp"><font color="#ff9900">home
feature</font></a></td></tr>
        <tr><td align="center"><a href="aboutus.asp"><font color="#ff9900">about us</font>
</a></td></tr>
        </table>
</td>
</tr>
</table><br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<center><font color="#00ffff">Remarks:
<img src='pict/cool_60.jpg' width='20' height="20" ><font color="#00ffff">=off, </font>
<img src='pict/deco_60.jpg' width='20' height="20" ><font color="#00ffff">=on </font>
</center>

<%

        dim fso
        dim writestream
        set fso = createobject("scripting.filesystemobject")
        set writestream = fso.CreateTextFile(server.MapPath("sw.txt"),true)
        select case request.form("sw1")
                case "switch no.1 off"
                writestream.writeline("0")
                case "switch no.1 on"
                writestream.writeline("1")
        end select
        select case request.form("sw2")
                case "switch no.2 off"
                writestream.writeline("2")
                case "switch no.2 on"
                writestream.writeline("3")
        end select
        select case request.form("sw3")
                case "switch no.3 off"
                writestream.writeline("4")
                case "switch no.3 on"
                writestream.writeline("5")
        End Select
        writestream.close
%>
```

----------ASP SENSOR----------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<%
        Set fso = CreateObject("Scripting.FileSystemObject")
        Set readstream = fso.opentextfile(server.MapPath("test.txt"),1)
        status = ReadStream.Readline
%>
        <center><STRONG><font face=sans-serif size="3" color=blue>Status of sensors</font>
</STRONG></center><br>
        <tr>
<table align='center' border=1>
        <tr bgcolor=#cccccc><td><font size=3 color=crimson>status</font></td>
                <td><center><font size=3 color=crimson>position</font></center></td>
                <td><font size=3 color=crimson>IN</font></td>
                <td><font size=3 color=crimson>OUT</font></td>
        </tr>
<%if left(status,1) = "a" then%>
        <tr><td width="30"><center><img src='pict/red.gif' width='30' height="30" ></center>
</td>
<%else%>
                <td  align="center" width="30"><img src='pict/green.gif' width='30'
height="30" ></td>
<%end if%>
                <td align="center"><a href="details.asp"><font color=Blue>Front door</font>
</a></td>
                <%for i = 1 to len(status) step 1
                        if mid(status,i,1) = "a" then
                        j = j + 1
                        else
                        k = k + 1
                        end if
                next
                        %>
                <td width="30" align= "center"><%=j%><font color=Crimson></td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                <td width="30" align= "center" ><%=k%><font color=Crimson></td>
        </tr>
<%
        Set fso1 = CreateObject("Scripting.FileSystemObject")
        Set readstream1 = fso1.opentextfile(server.MapPath("test1.txt"),1)
        status1 = readStream1.Readline
%>

<%if left(status1,1) = "c" then%>
        <tr><td  align="center" width="30"><img src='pict/red.gif' width='30' height="30" >
</td>
<%else%>
                <td  align="center" width="30"><img src='pict/green.gif' width='30'
height="30" ></td>
<%end if%>
                <td align="center"><a href="details1.asp"><font color="Blue">Back door
</font></a></td>
                <%for l = 1 to len(status1) step 1
                    if mid(status1,l,1) = "c" then
                    m = m + 1
                    else
                    n = n + 1
                    end if
                next
                    %>
                <td width="30" align= "center"><%=m%><font color=Crimson></td>
                <td width="30" align="center"><%=n%><font color=Crimson></td>

        tr
    table
    tr><br>
    center><font color= "blue"> Remarks: </font><img src='pict/green.gif' width='20' height="20" >
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<font color="blue"> =Normal. </font>
<img src='pict/red.gif' width='20' height="20" ><font color="blue"> =Abnormal </font><center>

<p align=right ><a href="proj1.asp"><font color=blue>go to main</font></a>
------------------------FRONT DOOR-----------------------

<%
  set FSO = CreateObject("Scripting.FileSystemObject")
  set readstream = FSO.OpenTextFile(server.MapPath("test.txt"),1)
  if left(readstream.Readline,1) = "a" then
                set textstream = FSO.OpenTextFile(server.MapPath("gate_ab.txt").1)
                response.write "<font color='blue' size='+2'>"&textstream.ReadAll&" 
</font>"&now
  else
                set textstream = FSO.OpenTextFile(server.MapPath("gate_normal.txt"),1)
                response.write "<font color='blue' size='+2'>"&textstream.ReadAll&" 
</font>"&now
  end if

%>
<img src="pict\gate.gif" width="800"></br>
<a href="proj1.asp"><font color="red">go to main</font></a></br>
<a href="test.asp"><font color="red">previous page</font></a>
------------------------ WINDOW----------------------
<%
  set FSO = CreateObject("Scripting.FileSystemObject")
  set readstream = FSO.OpenTextFile(server.MapPath("test1.txt").1)
  if left(readstream.Readline,1) = "c" then
                set textstream = FSO.OpenTextFile(server.MapPath("back_ab.txt").1)
                response.write "<font color='blue' size='+2'>"&textstream.ReadAll& "</font>
<font color=red>ABNORMAL</font>"&" "&now
  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                set textstream = FSO.OpenTextFile(server.MapPath("back_normal.txt"),1)
                response.write "<font color='blue' size='+2'>"&textstream.ReadAll&" 
    font>"&now
    end if

"a>
    img src="pict\back.gif" ></br>
    a href="proj1.asp" ><font color="red">go to main</font></a></br>
    a href="test.asp"><font color="red">previous page</font></a>
----------HOME FEATURE----------
<center ><font size=5 color=blue>HOME FEATURE</font></center>
    img src="pict\homefeature.jpg" >
    a href="proj1.asp"><font color="red">go to main</font></a></br>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

----------VISUAL BASIC PROGRAM----------

```
Option Explicit
Private switch As New Form1

Public Sub sw1_off(data_out As Variant)
data_out = "0"
Dim i As Integer
For i = 1 To 1000
switch.MSComm1.RTSEnable = True
switch.MSComm1.Output = data_out
switch.MSComm1.RTSEnable = False
Next i
End Sub

Public Sub sw2_off(data_out As Variant)
data_out = "2"
switch.MSComm1.RTSEnable = True
switch.MSComm1.Output = data_out
switch.MSComm1.RTSEnable = False
End Sub

Public Sub sw3_off(data_out As Variant)
data_out = "4"
switch.MSComm1.RTSEnable = True
switch.MSComm1.Output = data_out
switch.MSComm1.RTSEnable = False
End Sub
Public Sub sw1_on(data_out As Variant)
data_out = "1"
Dim i As Integer
For i = 1 To 1000
    switch.MSComm1.RTSEnable = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    switch.MSComm1.Output = data_out
    switch.MSComm1.RTSEnable = False
  Next i
End Sub
Public Sub sw2_on(data_out As Variant)
data_out = "3"
switch.MSComm1.RTSEnable = True
switch.MSComm1.Output = data_out
switch.MSComm1.RTSEnable = False
End Sub
Public Sub sw3_on(data_out As Variant)
data_out = "5"
switch.MSComm1.RTSEnable = True
switch.MSComm1.Output = data_out
switch.MSComm1.RTSEnable = False
End Sub
Public Sub open_port()
  switch.MSComm1.PortOpen = True
End Sub
----------PICBASIC.ASM----------

PM_USED                 EQU     1

        INCLUDE

#define         OSC             4
#define         CHAR_PACING             1000

RAM_START               EQU     0000Ch
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RAM_END                 EQU   0004Fh
RAM_BANKS               EQU   00001h
BANK0_START             EQU   0000Ch
BANK0_END               EQU   0004Fh
EEPROM_START            EQU   02100h
EEPROM_END              EQU   0213Fh

R0                  EQU   RAM_START + 000h
R1                  EQU   RAM_START + 002h
R2                  EQU   RAM_START + 004h
R3                  EQU   RAM_START + 006h
R4                  EQU   RAM_START + 008h
R5                  EQU   RAM_START + 00Ah
R6                  EQU   RAM_START + 00Ch
R7                  EQU   RAM_START + 00Eh
R8                  EQU   RAM_START + 010h
FLAGS               EQU   RAM_START + 012h
GOP                 EQU   RAM_START + 013h
RM1                 EQU   RAM_START + 014h
RM2                 EQU   RAM_START + 015h
RR1                 EQU   RAM_START + 016h
RR2                 EQU   RAM_START + 017h
 i                  EQU   RAM_START + 018h
 item               EQU   RAM_START + 019h
 temp               EQU   RAM_START + 01Ah
 PORTL              EQU   PORTB
 PORTH              EQU   PORTA
 TRISL              EQU   TRISB
 TRISH              EQU   TRISA
#define _PORTA_4            PORTA, 004h
#define _PORTA_3            PORTA, 003h
#define _PORTA_2            PORTA, 002h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define _PORTA_1            PORTA, 001h
#define _PORTB_7        PORTB, 007h
#define _PORTA_0            PORTA, 000h
#define _PORTB_0        PORTB, 000h
#define _PORTB_1        PORTB, 001h
#define _PORTB_2        PORTB, 002h

: Constants.
_T2400          EQU     00000h
_T1200          EQU     00001h
_T9600          EQU     00002h
_T300           EQU     00003h
_N2400          EQU     00004h
_N1200          EQU     00005h
_N9600          EQU     00006h
_N300           EQU     00007h
_OT2400         EQU     00008h
_OT1200         EQU     00009h
_OT9600         EQU     0000Ah
_OT300          EQU     0000Bh
_ON2400         EQU     0000Ch
_ON1200         EQU     0000Dh
_ON9600         EQU     0000Eh
_ON300          EQU     0000Fh
_MSBPRE                 EQU     00000h
_LSBPRE         EQU     00001h
_MSBPOST                EQU     00002h
_LSBPOST                EQU     00003h
_LSBFIRST               EQU     00000h
_MSBFIRST               EQU     00001h
_CLS            EQU     00000h
_HOME           EQU     00001h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BELL            EQU     00007h
BKSP            EQU     00008h
TAB             EQU     00009h
CR              EQU     0000Dh
UnitOn          EQU     00012h
UnitOff         EQU     0001Ah
UnitsOff        EQU     0001Ch
LightsOn        EQU     00014h
LightsOff       EQU     00010h
Dim             EQU     0001Eh
Bright          EQU     00016h
        INCLUDE "OOO.MAC"
        INCLUDE


        LABEL?L     _cmeon7
        MOVE?CB     0FFh, TRISA
        MOVE?CB     000h, TRISB

        LABEL?L     _Main
        CMPNE?TCL   _PORTA_4, 001h, L00001
        MOVE?CB     004h, _item
        LABEL?L     L00001
        CMPNE?TCL   _PORTA_3, 001h, L00003
        MOVE?CB     003h, _item
        LABEL?L     L00003
        CMPNE?TCL   _PORTA_2, 001h, L00005
        MOVE?CB     002h, _item
        LABEL?L     L00005
        CMPNE?TCL   _PORTA_1, 001h, L00007
        MOVE?CB     001h, _item
        LABEL?L     L00007
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SERPIN?T        _PORTB_7
SERMODE?C       _T9600
SEROUT?B        _item
SERPIN?T        _PORTA_0
SERMODE?C       _T9600
SERIN?B         _temp
CMPNE?BCL       _temp, 031h, L00009
MOVE?CT         000h, _PORTB_0
LABEL?L         L00009
CMPNE?BCL       _temp, 030h, L00011
MOVE?CT         001h, _PORTB_0
LABEL?L         L00011
CMPNE?BCL       _temp, 033h, L00013
MOVE?CT         000h, _PORTB_1
LABEL?L         L00013
CMPNE?BCL       _temp, 032h, L00015
MOVE?CT         001h, _PORTB_1
LABEL?L         L00015
CMPNE?BCL       _temp, 035h, L00017
MOVE?CT         000h, _PORTB_2
LABEL?L         L00017
CMPNE?BCL       _temp, 034h, L00019
MOVE?CT         001h, _PORTB_2
LABEL?L         L00019
GOTO?L          _Main

END
```

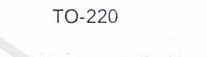
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR

BD243/A/B/C

Medium Power Linear and Switching Applications

- Complement to BD244, BD244A, BD244B and BD244C respectively



1.Base 2.Collector 3.Emitter

NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings T_C=25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Base Voltage		
	: BD243	45	V
	: BD243A	60	V
	: BD243B	80	V
	: BD243C	100	V
V_{CEO}	Collector-Emitter Voltage		
	: BD243	45	V
	: BD243A	60	V
	: BD243B	80	V
	: BD243C	100	V
V_{EBO}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	6	A
I_{CP}	*Collector Current (Pulse)	10	A
I_B	Base Current	2	A
P_C	Collector Dissipation (T_C=25°C)	65	W
T_J	Junction Temperature	150	°C
T_{STG}	Storage Temperature	- 65 ~ 150	°C

Electrical Characteristics T_C=25°C unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
V_{CEO}(sus)	* Collector-Emitter Sustaining Voltage					
	: BD243	I_C=30mA, I_B=0	45			V
	: BD243A		60			V
	: BD243B		80			V
	: BD243C		100			V
I_{CEO}	Collector Cut-off Current : BD243/243A	V_{CE} = 30V, I_B = 0			0.7	mA
	: BD243B/243C	V_{CE} = 60V, I_B = 0			0.7	mA
I_{CES}	Collector Cut-off Current : BD243	V_{CE} = 45V, V_{BE} = 0			0.4	mA
	: BD243A	V_{CE} = 60V, V_{BE} = 0			0.4	mA
	: BD243B	V_{CE} = 80V, V_{BE} = 0			0.4	mA
	: BD243C	V_{CE} = 100V, V_{BE} = 0			0.4	mA
I_{EBO}	Emitter Cut-off Current	V_{EB} = 5V, I_C = 0			1	mA
h_{FE}	*DC Current Gain	V_{CE} = 4V, I_C = 0.3A	30			
		V_{CE} = 4V, I_C = 3A	15			
V_{CE}(sat)	*Collector-Emitter Saturation Voltage	I_C = 6A, I_B = 1A			1.5	V
V_{BE}(on)	*Base-Emitter ON Voltage	V_{CE} = 4V, I_C = 6A			2	V

*Pulse Test: PW=300μs, Duty Cycle≤20% Pulsed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Characteristics

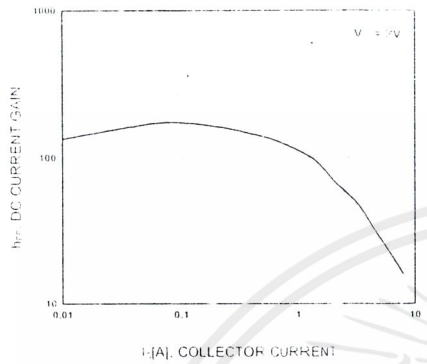


Figure 1. DC current Gain

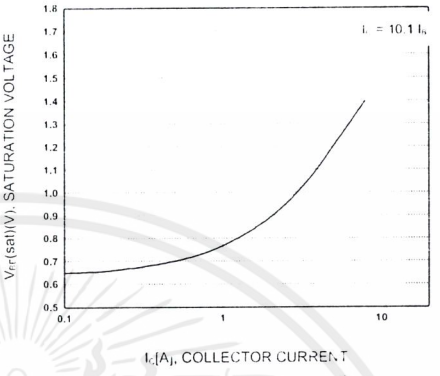


Figure 2. Base-Emitter Saturation Voltage

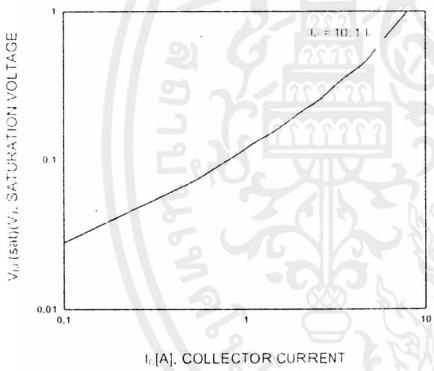


Figure 3. Collector-Emitter Saturation Voltage

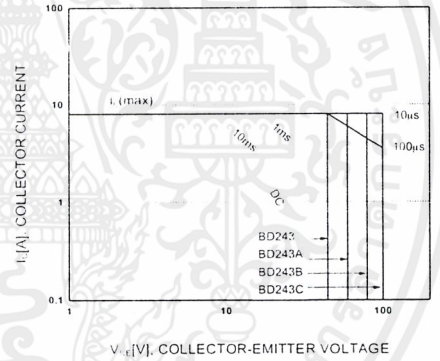


Figure 4. Safe Operating Area

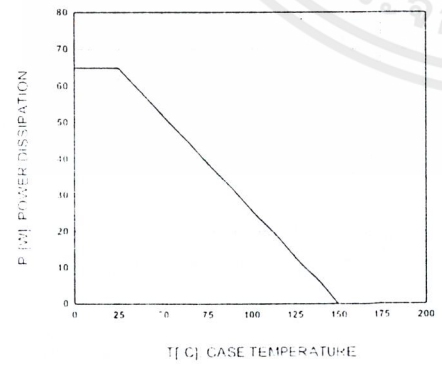
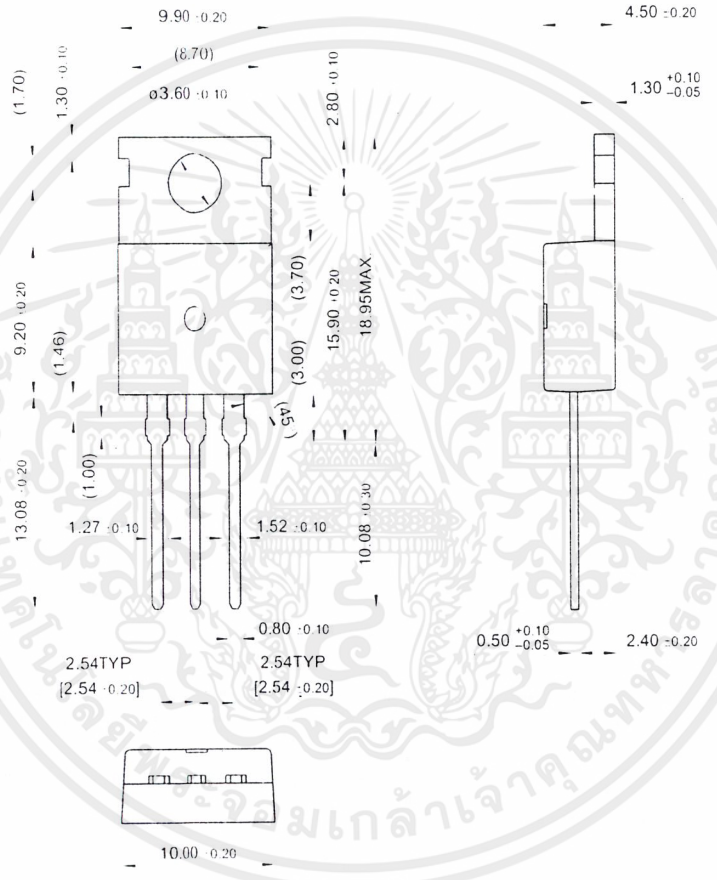


Figure 5. Power Derating

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Package Demensions

TO-220



Dimensions in Millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™
Bottomless™
CoolFET™
CROSSVOLT™
E^2CMOS™
FACT™
FACT Quiet Series™
FAST®
FASTr™
GTO™
HiSeC™
ISOPLANAR™
MICROWIRE™
POP™
PowerTrench®
QFET™
QS™
Quiet Series™
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
SyncFET™
TinyLogic™
UHC™
VCX™

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR INTERNATIONAL.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

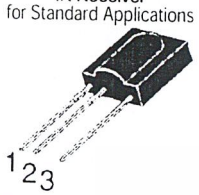
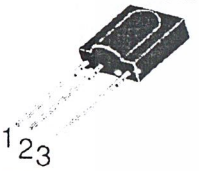
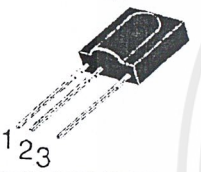
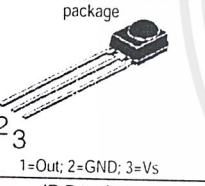
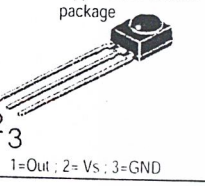
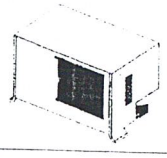
Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

©2002 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vishay Receivers for IR-Remote Control

VISHAY

PRODUCT DESCRIPTION	TYPE	DIM. FIG.	FO(KHZ)	FEATURES
IR Receiver for Standard Applications 	TSOP1730 TSOP1733 TSOP1736 TSOP1737 TSOP1738 TSOP1740 TSOP1756	72 72 72 72 72 72 72	30 33 36 36.7 38 40 56	• Internal filter for PCM Frequency • Improved shieding against all electrical field disturbances • TTL and CMOS compatibility • Output active low • Continuous data transmissions possible (1200 bit/s) • Suitable burst length = / > 10 cycles/burst • 1 = GND ; 2 = Vs ; 3. = Out.
IR Receiver for use in Disturbed Ambience 	TSOP1230 TSOP1233 TSOP1236 TSOP1237 TSOP1238 TSOP1240 TSOP1256	72 72 72 72 72 72 72	30 33 36 36.7 38 40 56	• Enhanced immunity against all kind of disturbance light • No occurance of disturbance pulses at the output • Continuous data transmission possible (1200 bit/s) • Suitable burst length = / > 10 cycles/burst • tp/T = 0.5 • 1 = GND ; 2 = Vs ; 3. = Out.
IR Receiver for Short Burst Applications 	TSOP1130 TSOP1133 TSOP1136 TSOP1137 TSOP1138 TSOP1140 TSOP1156	72 72 72 72 72 72 72	30 33 36 36.7 38 40 56	• Enhanced data rate of 2400 bit/s. • Operation with short bursts possible(= / > 6 cycles/burst) • tp/T = 0.5 • 1 = GND ; 2 = Vs ; 3. = Out.
IR Receiver for Standard Applications in small package  1=Out; 2=GND; 3=Vs	TSOP1830 TSOP1833 TSOP1836 TSOP1837 TSOP1838 TSOP1840 TSOP1856	73 73 73 73 73 73 73	30 33 36 36.7 38 40 56	• Internal filter for PCM frequency • TTL and CMOS compatibility • Output active low • Small size package • Enhanced immunity against all kind of disturbance light. • No occurance of disturbance pulses at the output • Short setting time after power on(< 200 us)
IR Receiver for Standard Applications in small package  1=Out ; 2= Vs ; 3=GND	TSOP1830 TSOP1833 TSOP1836 TSOP1837 TSOP1838 TSOP1840 TSOP1856	73 73 73 73 73 73 73	30 33 36 36.7 38 40 56	• Internal filter for PCM frequency • TTL and CMOS compatibility • Output active low • Small size package • Enhanced immunity against all kind of disturbance light. • No occurance of disturbance pulses at the output • Short setting time after power on(< 200 us)
IR Receiver for Standard Applications in SMD package	TFMM5300 TFMM5330 TFMM5360 TFMM5370 TFMM5380 TFMM5400 TFMM5560	75 75 75 75 75 75 75	30 33 36 36.7 38 40 56	• Internal filter for PCM frequency • Enhanced immunity against all electrical field disturbances • TTL and CMOS compatibility • Output active low • Continuous data transmission possible(1200 bit/s) • Suitable burst length = / > 10 cycles/burst • tp/T = 0.5 • Top-view package
IR Receiver for Standard Applications in SMD package 	TFMM5300 TFMM5330 TFMM5360 TFMM5370 TFMM5380 TFMM5400 TFMM5560	75 75 75 75 75 75 75	30 33 36 36.7 38 40 56	• Internal filter for PCM frequency • Enhanced immunity against all electrical field disturbances • TTL and CMOS compatibility • Output active low • Continuous data transmission possible(1200 bit/s) • Suitable burst length = / > 10 cycles/burst • tp/T = 0.5 • Top-view package

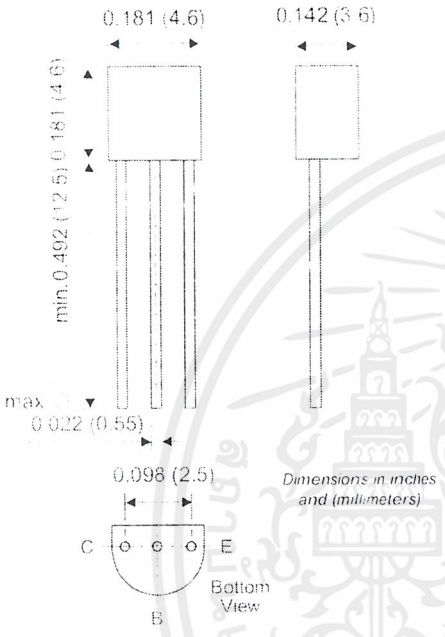
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL SEMICONDUCTOR

BC546 thru BC548

Small Signal Transistors (NPN)

TO-226AA (TO-92)



Features

- NPN Silicon Epitaxial Planar Transistors
- These transistors are subdivided into three groups A, B, and C according to their current gain. The type BC546 is available in groups A and B, however, the types BC547 and BC548 can be supplied in all three groups. As complementary types the PNP transistors BC556...BC558 are recommended.
- On special request, these transistors are also manufactured in the pin configuration TO-18.

Mechanical Data

Case: TO-92 Plastic Package
Weight: approx. 0.18g
Packaging Codes/Options:
E6/Bulk – 5K per container, 20K/box
E7/4K per Ammo mag., 20K/box

Maximum Ratings & Thermal Characteristics

Ratings at 25°C ambient temperature unless otherwise specified

Parameter		Symbol	Value	Unit
Collector-Base Voltage	BC546 BC547 BC548	V_{CBO}	80 50 30	V
Collector-Emitter Voltage	BC546 BC547 BC548	V_{CES}	80 50 30	V
Collector-Emitter Voltage	BC546 BC547 BC548	V_{CEO}	65 45 30	V
Emitter-Base Voltage	BC546, BC547 BC548	V_{EBO}	6 5	V
Collector Current		I_C	100	mA
Peak Collector Current		I_{CM}	200	mA
Peak Base Current		I_{BM}	200	mA
Peak Emitter Current		$-I_{EM}$	200	mA
Power Dissipation at T_{amb} = 25°C		P_{tot}	500(1)	mW
Thermal Resistance Junction to Ambient Air		$R_{\Theta JA}$	250(1)	°C/W
Junction Temperature		T_J	150	°C
Storage Temperature Range		T_S	–65 to +150	°C

Note: (1) Valid provided that leads are kept at ambient temperature at a distance of 2 mm from case.

1/24/01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL SEMICONDUCTOR

BC546 thru BC548

Small Signal Transistors (NPN)

Electrical Characteristics (T_j = 25°C unless otherwise noted)

Parameter		Symbol	Test Condition	Min	Typ	Max	Unit
Small Signal Current Gain	Current gain group A	h_{fe}	V_{CE} = 5 V, I_C = 2 mA, f = 1 kHz	—	220	—	—
	B			—	330	—	
	C			—	600	—	
Input Impedance	Current gain group A	h_{ie}	V_{CE} = 5 V, I_C = 2 mA, f = 1 kHz	1.6	2.7	4.5	kΩ
	B			3.2	4.5	8.5	
	C			6	8.7	15	
Output Admittance	Current gain group A	h_{oe}	V_{CE} = 5 V, I_C = 2 mA, f = 1kHz	—	18	30	µS
	B			—	30	60	
	C			—	60	110	
Reverse Voltage Transfer Ratio	Current gain group A	h_{re}	V_{CE} = 5 V, I_C = 2 mA, f = 1kHz	—	$1.5 \cdot 10^{-4}$	—	—
	B			—	$2 \cdot 10^{-4}$	—	
	C			—	$3 \cdot 10^{-4}$	—	
DC Current Gain	Current gain group A	h_{FE}	V_{CE} = 5 V, I_C = 10 µA	—	90	—	—
	B			—	150	—	
	C			—	270	—	
	Current gain group A		V_{CE} = 5 V, I_C = 2 mA	110	180	220	
	B			200	290	450	
	C			420	500	800	
	Current gain group A		V_{CE} = 5 V, I_C = 100 mA	—	120	—	
	B			—	200	—	
	C			—	400	—	
Collector Saturation Voltage		V_{CEsat}	I_C = 10 mA, I_B = 0.5 mA	—	80	200	mV
			I_C = 100 mA, I_B = 5 mA	—	200	600	
Base Saturation Voltage		V_{BEsat}	I_C = 10 mA, I_B = 0.5 mA	—	700	—	mV
			I_C = 100 mA, I_B = 5 mA	—	900	—	
Base-Emitter Voltage		V_{BE}	V_{CE} = 5 V, I_C = 2 mA	580	660	700	mV
			V_{CE} = 5 V, I_C = 10 mA	—	—	720	
Collector-Emitter Cutoff Current	BC546	I_{CES}	V_{CE} = 80 V	—	0.2	15	nA
	BC547		V_{CE} = 50 V	—	0.2	15	nA
	BC548		V_{CE} = 30 V	—	0.2	15	nA
	BC546		V_{CE} = 80 V, T_j = 125°C	—	—	4	µA
	BC547		V_{CE} = 50 V, T_j = 125°C	—	—	4	µA
	BC548		V_{CE} = 30 V, T_j = 125°C	—	—	4	µA
Gain-Bandwidth Product		f_T	V_{CE} = 5 V, I_C = 10 mA, f = 100 MHz	—	300	—	MHz
Collector-Base Capacitance		C_{CBO}	V_{CB} = 10 V, f = 1 MHz	—	3.5	6	pF
Emitter-Base Capacitance		C_{EBO}	V_{EB} = 0.5 V, f = 1 MHz	—	9	—	pF
Noise Figure	BC546, BC547 BC548	F	V_{CE} = 5 V, I_C = 200 µA, R_G = 2 kΩ, f = 1 kHz, Δf = 200 Hz	—	2	10	dB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL SEMICONDUCTOR

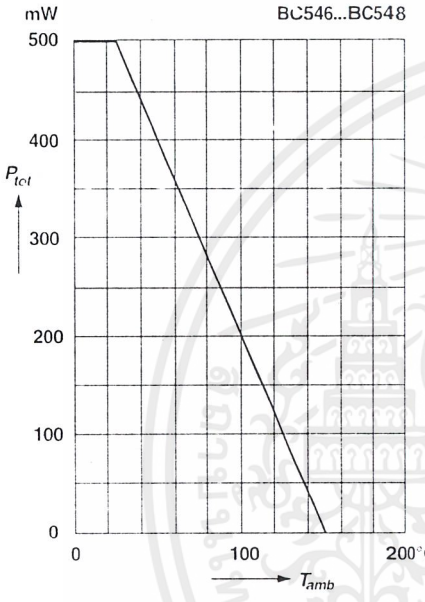
BC546 thru BC548

Small Signal Transistors (NPN)

Ratings and Characteristic Curves (TA = 25°C unless otherwise noted)

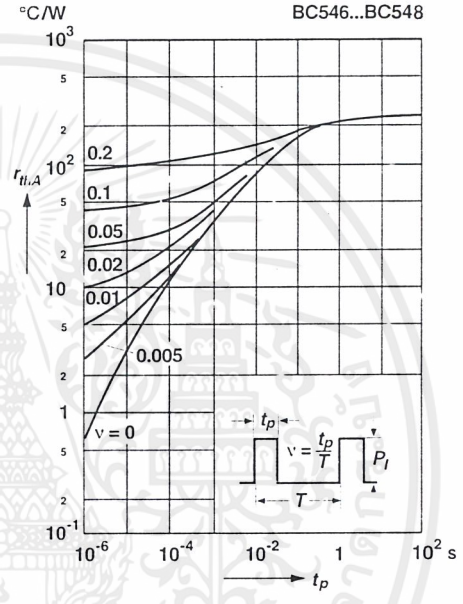
Admissible power dissipation versus temperature

Valid provided that leads are kept at ambient temperature at a distance of 2 mm from case

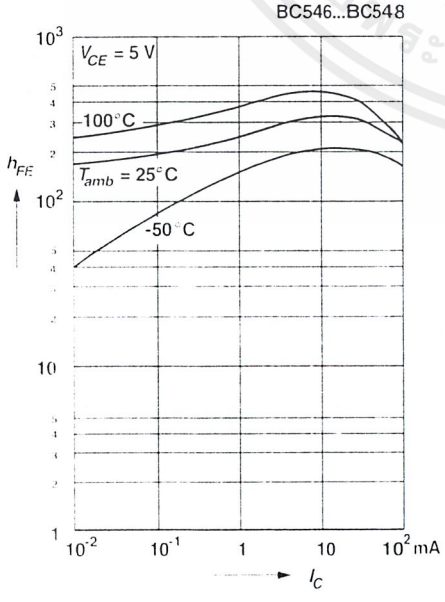


Pulse thermal resistance versus pulse duration

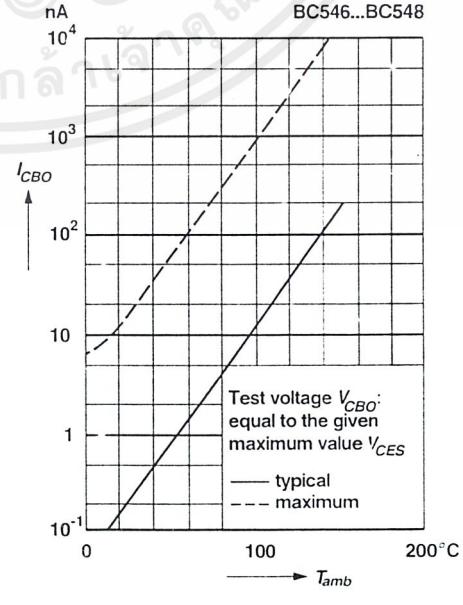
Valid provided that leads are kept at ambient temperature at a distance of 2 mm from case



DC current gain versus collector current



Collector-base cutoff current versus ambient temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

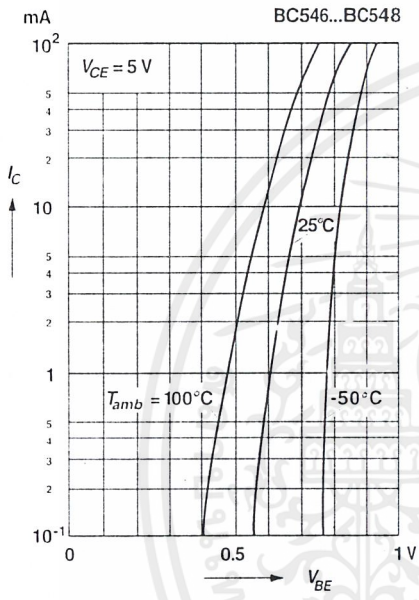
GENERAL SEMICONDUCTOR

BC546 thru BC548

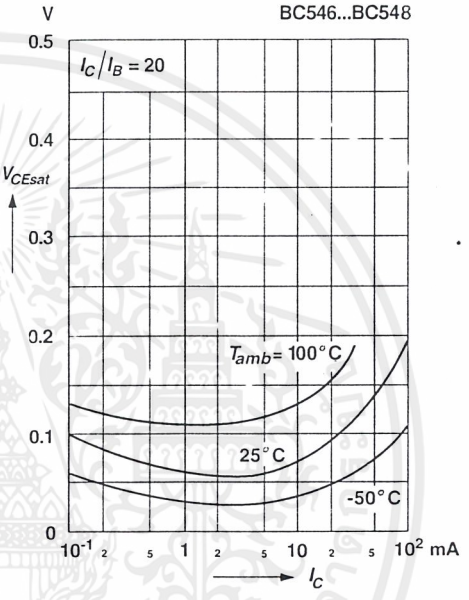
Small Signal Transistors (NPN)

Ratings and Characteristic Curves (TA = 25°C unless otherwise noted)

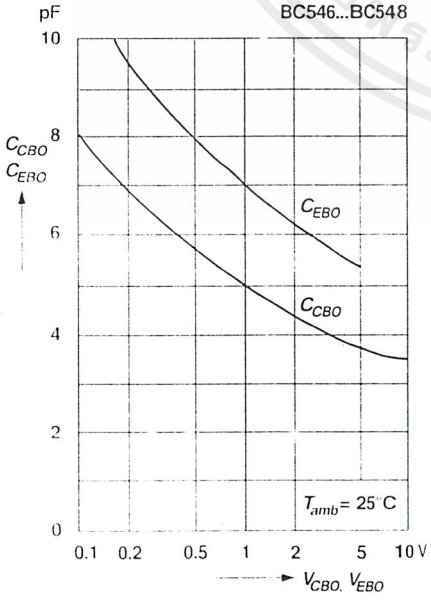
Collector current versus base-emitter voltage



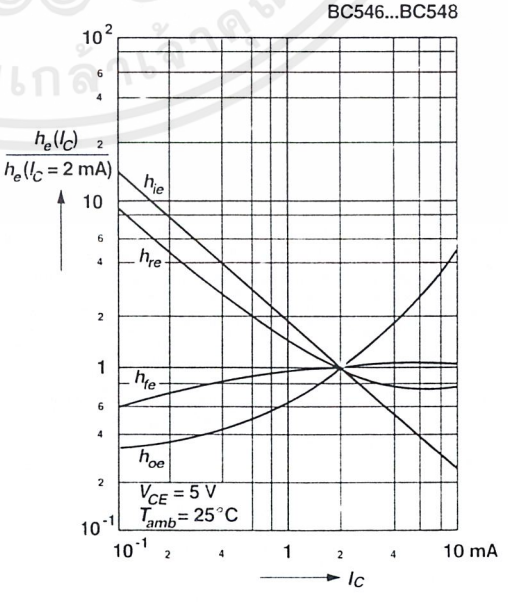
Collector saturation voltage versus collector current



Collector-base capacitance, Emitter-base capacitance versus reverse bias voltage



Relative h-parameters versus collector current



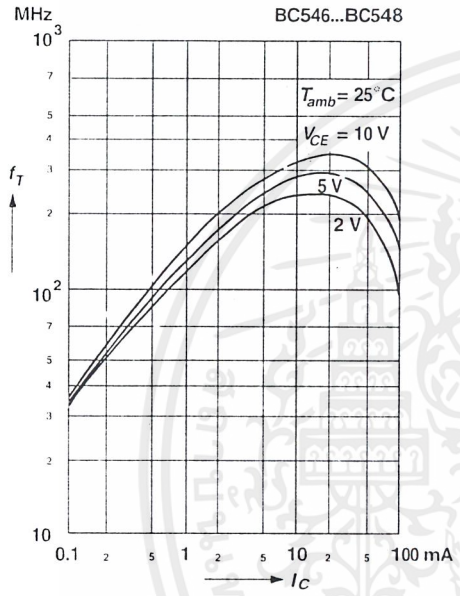
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BC546 thru BC548

Small Signal Transistors (NPN)

Ratings and Characteristic Curves (T_A = 25°C unless otherwise noted)

Gain-bandwidth product versus collector current



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลือเป็นอย่างดี ทั้งจากอาจารย์ที่ปรึกษา รุ่นพี่ และเพื่อนๆ ที่คอยให้คำปรึกษาและแนะนำ ถ้าไม่มีบุคคลเหล่านี้ โครงงานชิ้นนี้ก็ไม่สามารถสำเร็จลุล่วงได้ ดังนั้นจึงขอขอบคุณอาจารย์ที่ปรึกษา รุ่นพี่ และเพื่อนๆไว้ ณ โอกาสนี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้