สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบความเรียบร้อย และความปลอดภัยของบ้านผ่านอินเทอร์เน็ต

HOME SECURITY AND MONITORING SYSTEM VIA INTERNET

โดย

นายศราวุธ	พานชัยเจริญ	40010764
นายศุภชัย	ชนะฤทธิชัย	40010790
นายศุภชาติ	นาคะปักษิราช	40010793

เลขหมู่.....
เลขทะเบียน..... 42278
วัน, เดือน, ปี..... 16 พ.ค. 2545

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบความเรียบร้อย และความปลอดภัยของบ้านผ่านอินเทอร์เน็ต

HOME SECURITY AND MONITORING SYSTEM VIA INTERNET

โดย

นายศราวุธ พานชัยเจริญ 40010764

นายศุภชัย ชนะฤทธิชัย 40010790

นายศุภชาติ นาคะปักษิราช 40010793

อาจารย์ที่ปรึกษา

อาจารย์อิทธิภูมิ บุญพิคำ

ปริญญานิพนธ์สำหรับวิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2543

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบความเรียบร้อยและความปลอดภัยของบ้านผ่านอินเทอร์เน็ต

ผู้จัดทำ

นายศราวุธ	พานชัยเจริญ	40010764
นายศุภชัย	ชนะฤทธิชัย	40010790
นายศุภชาติ	นาคะปักษิราช	40010793

.....

(อาจารย์อิทธิภูมิ บุญพิคำ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์	ระบบตรวจสอบความเรียบร้อยและความปลอดภัยของบ้านผ่านอินเทอร์เน็ต		
	Home Security and Monitoring system via Internet		
จัดทำโดย	นายศราวุธ	พานชัยเจริญ	40010764
	นายศุภชัย	ชนะฤทธิชัย	40010790
	นายศุภชาติ	นาคะปักษิราช	40010793
อาจารย์ที่ปรึกษา	อาจารย์อิทธิภูมิ	บุญพิคำ	

ปริญญานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา

(อาจารย์อิทธิภูมิ บุญพิคำ)

วันที่ 4/01/....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

เหนือสิ่งอื่นใดผู้จัดทำปริญญานิพนธ์ขอขอบพระคุณคุณพ่อคุณแม่ ที่ได้ประทานทุกสิ่งทุกอย่างให้สามารถทำงานต่างๆให้ลุล่วงไปด้วยดี รวมทั้งอาจารย์อิทธิภูมิ บุญพิคำ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานครั้งนี้ ที่ได้ให้คำปรึกษาและความรู้ในการจัดทำโครงงานครั้งนี้ และคณาจารย์ทุกท่านรวมถึงเพื่อนๆ ที่ได้ให้ความรู้ ความช่วยเหลือ และคำแนะนำต่างๆจนทำให้โครงงานครั้งนี้สำเร็จลุล่วงไปด้วยดี และท้ายสุดนี้ขอขอบพระคุณผู้ที่ไม่ได้กล่าวถึงในข้างต้น พวกท่านอยู่ในใจเราตลอด ขอขอบพระคุณอย่างสูงไว้ ณ ที่นี้ด้วย

.....
(นายศราวุธ พานชัยเจริญ)

.....
(นายศุภชัย ชนะฤทธิชัย)

.....
(นายศุภชาติ นาคะปักษิราช)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบความเรียบร้อย และความปลอดภัยของบ้านผ่านอินเทอร์เน็ต

นายศราวุธ พานชัยเจริญ
นายศุภชัย ชนะฤทธิชัย
นายศุภชาติ นาคะปักษิราช
อาจารย์อิทธิภูมิ บุญพิคำ (อาจารย์ที่ปรึกษา)
ปีการศึกษา 2543

บทคัดย่อ

โครงงานนี้เสนอระบบตรวจสอบความเรียบร้อย และความปลอดภัยของบ้านผ่าน Internet รายละเอียดของโครงงานนี้ได้แก่ ส่วน interface ระหว่างไมโครคอนโทรลเลอร์ และ Target Server ให้การควบคุมจำนวนอุปกรณ์ไฟฟ้าเป้าหมายได้ และส่วนของ control station หรือ Source Server ซึ่งโปรแกรมสำหรับการติดต่อระหว่าง Server ทั้งสองผ่าน Internet ได้ถูกเขียนขึ้น ผลการทำงานสามารถให้ Server ทั้งสองส่งผ่านข้อมูลพื้นฐานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Home Security and Monitoring system via an Internet

Mr.Sarawut Panchaicharoen
Mr.Supachai Chanarittichai
Mr.Supachad Nakapugsirach
Mr.Ittibhoom Boonpikum(Advisor)
Education Year 2000

Abstract

This project presents home security and monitoring system via an internet which are interface between microcontroller and target server or the server that connects directly with target electronics devices. This allow the system be able to control more devices. Also , a program has been written to be able to link between two server , which are control station or source server and target station or target server. At the end of this work e are able to send basic information via an internet and displayed on the other server.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
บทที่ 2 มาตรฐานการสื่อสารพอร์ตอนุกรม	2
2.1 ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟสตามมาตรฐานต่างๆ	2
2.1.1การอินเทอร์เฟสตามมาตรฐาน RS-232-C	2
2.1.2ลักษณะการอินเทอร์เฟสที่ใช้มาตรฐาน RS-423	4
2.1.3ลักษณะการอินเทอร์เฟสที่ใช้มาตรฐาน RS-422	5
2.1.4 ลักษณะการอินเทอร์เฟสแบบ current-loop	6
บทที่ 3 หลักการและความรู้ทั่วไปเกี่ยวกับไมโครคอนโทรลเลอร์ MCS51 และหลักการออกแบบวงจรอินฟราเรด	8
3.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูลMCS51	8
3.2 พอร์ทของ 8051	8
3.3 หน่วยความจำของMCS51	10
3.3.1 หน่วยความจำสำหรับเก็บโปรแกรม(Program Memory)	10
3.3.2 หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)	10
3.4 ไทม์เมอร์/เคาท์เตอร์	11
3.4.1 โครงสร้างของ ไทม์เมอร์ และ เคาท์เตอร์ มีโครงสร้างดังนี้	11
3.4.2 Timer/Counter Mode Control Register(TMOD)อยู่ใน SFR ตำแหน่งที่(89H)	11
3.4.3 Timer/Counter Control Register(TCON)อยู่ใน SFR ตำแหน่งที่(088H)	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 รีจิสเตอร์ฟังก์ชั่นพิเศษ 13
3.5.1Accumulator (ACC)อยู่ในตำแหน่ง 0E0h 13
3.5.2 B Register (B) อยู่ที่ตำแหน่ง OFOh 13
35.3 Program Status Word อยู่ที่ตำแหน่ง ODOh 13
3.5.4 Stact Pointer (SP) อยู่ที่ตำแหน่ง 081 h 13
3.5.5 Data Pointer Register อยู่ที่ตำแหน่ง 82h และ 83h 14
3.5.6 Port (0-3)อยู่ที่ตำแหน่ง 80h,90h,0A0h,0B0h 14
3.5.7 Serail Data Buffer (SBUF) อยู่ที่ตำแหน่ง 99h 14
3.5.8 Interrupt Priority Register (IP) อยู่ตำแหน่ง 14
3.5.9 Enterrupt Enable Register (IE) อยู่ตำแหน่ง 0A8h 14
3.5.10 Timer/Counter Mode Control Register (TMOD) อยู่ที่ตำแหน่ง 089h 14
3.5.11 Timer/Counter Control Register (TCON) 14
3.5.12 Power Control Register(PCON) อยู่ที่ตำแหน่ง 87h 15
3.6 การออกแบบวงจรอินฟราเรด 16
บทที่ 4 การออกแบบวงจรส่วนควบคุมไฟ 18
4.1 ระบบควบคุมไฟทั่วไป 18
4.1.1 หลักการทั่วไป 18
4.2 การออกแบบวงจรควบคุมไฟ 18
4.2.1 วงจรแรงดันอ้างอิง +5 โวลท์ 18
4.2.2 วงจรทางแรงดันไฟสูง 19
บทที่ 5 การทำงานของส่วนโปรแกรม 20
5.1 โปรแกรมในส่วนไมโครคอนโทลเลอร์ 20
5.2 โปรแกรมในส่วนของ Delphi 21
บทที่ 6 ผลการทดลอง 26
บทที่ 7 สรุปและวิจารณ์ 28
บรรณานุกรม VIII
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

เรื่อง	หน้า
รูปที่ 1 แสดง Block Diagram ของระบบ	1
รูปที่ 2.1 วงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-232-C	3
รูปที่ 2.2 วงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-423	4
รูปที่ 2.3(ก) วงจรขับและรับสัญญาณที่ใช้มาตรฐาน RS-422	5
รูปที่ 2.3(ข) วงจรขับและรับสัญญาณที่ใช้มาตรฐาน RS-485	6
รูปที่ 2.4 วงจรที่ใช้การอินเทอร์เฟสแบบ current-loop	7
รูปที่ 3.1 การจัดวางขาของ 8051	9
รูปที่ 3.2 แสดงรายละเอียดของรีจิสเตอร์ TMOD	11
รูปที่ 3.3 แสดงรายละเอียดของรีจิสเตอร์ TCON	12
รูปที่ 3.4 แสดงวงจรอินฟราเรดทางด้านส่ง	17
รูปที่ 4.1 แสดงส่วนวงจรแรงดันไฟอ้างอิง +5 โวลต์	19
รูปที่ 4.2 ส่วนออปโตโซเลทและวงจรทางแรงดันสูง	19
รูปที่ 5.1. แสดง Flow Chart การทำงานของโปรแกรมของไมโครคอนโทรเลอร์	21
รูปที่ 5.1(ก) แสดง Flow Chart การทำงานของโปรแกรมไมโครคอนโทรเลอร์ด้าน Sensor	21
รูปที่ 5.1(ข) แสดง Flow Chart การทำงานของโปรแกรมไมโครคอนโทรเลอร์ด้าน Line Switch	22
รูปที่ 5.2 แสดง Flow Chart การทำงานของโปรแกรมของ Delphi	23
รูปที่ 5.2(ก) แสดง Flow Chart การทำงานของโปรแกรมของ Delphi ด้าน Server	23
รูปที่ 5.2(ข) แสดง Flow Chart การทำงานของโปรอกรมของ Delphi ด้าน Client	24
รูปที่ 5.3 รูปการแสดงหน้าจอของโปรแกรม Delphi ด้าน Server	25
รูปที่ 5.4 รูปการแสดงหน้าจอของโปรแกรม Delphi ด้าน Client	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงการเลือกโหมด Timer/Counter 12

ตารางที่ 6.1 แสดงผลการทดลอง 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Douglas V.Hall, 'Microprocessors and Interfacing Programming and Hardware', McGraw-Hill Book . Singapore , 1992
2. กฤษฎา ใจเย็น , ชัยวัฒน์ ลิ้มพรจิตวิไล , "เรียนรู้การเชื่อมต่อPC กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม" , อินโนเวตีฟ เอ็กเพอริเมนต์ , กรุงเทพ , 119 หน้า , 2542
3. รศ.สมยศ จุณณะปิยะ , "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกลู MCS-51" , คณะวิศวกรรมศาสตร์ สจล. , กรุงเทพ , 2543
4. สัจจะ จรัสรุ่งรวีวร , จักรพงษ์ สุขประเสริฐ , "เริ่มต้นอย่างมืออาชีพด้วย Delphi5 ฉบับสมบูรณ์" , อินโฟเพรส , กรุงเทพ , 2543

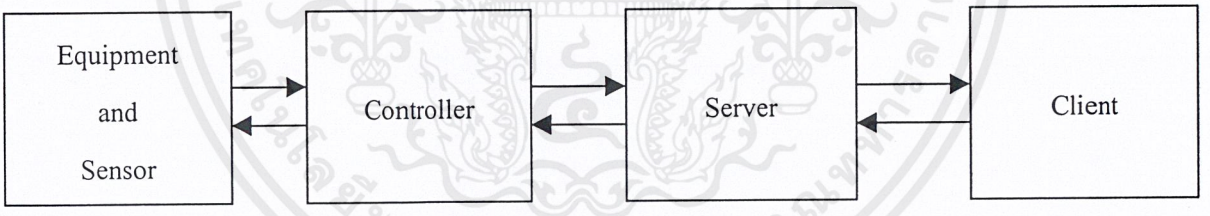
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันเทคโนโลยีได้พัฒนาไปอย่างรวดเร็วในทุกๆด้าน และการนำเทคโนโลยีต่าง ๆ มาประยุกต์เข้าด้วยกันก็จะเกิดสิ่งประดิษฐ์ใหม่ๆ ซึ่งช่วยอำนวยความสะดวก และเกิดประโยชน์มากมาย

โปรแกรมรักษาความปลอดภัยภายในอาคารนี้ ได้นำเทคโนโลยีต่างๆเหล่านั้นมาประยุกต์เข้าด้วยกันซึ่งตัว sensor ที่ใช้ในการตรวจเช็คนี้ จะใช้อินฟราเรดเป็นตัวตรวจจับ แล้วจึงส่งสัญญาณไปยังคอนโทรลเลอร์ เพื่อสั่งให้ทำงานต่อไป

ซึ่งโปรแกรมนี้จะถูกควบคุมผ่านทางคอมพิวเตอร์ ซึ่งแสดงผลโดยโปรแกรม Delphi ซึ่งทำหน้าที่แสดงผล ทำการประมวลผลคำสั่ง แล้วส่งข้อมูลคำสั่งผ่านพอร์ทอนุกรม มาที่ไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์จะทำหน้าที่สั่งงาน และสามารถควบคุมอุปกรณ์ภายในบ้านโดยผ่านเครือข่ายอินเทอร์เน็ต



รูปที่ 1 แสดง Block Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
มาตรฐานการสื่อสารผ่านพอร์ทอนุกรม

ในการสื่อสารโดยใช้คอมพิวเตอร์นั้น มีรูปแบบที่ใช้ในการสื่อสารอยู่ 2 อย่างใหญ่ๆคือ การสื่อสารแบบอนุกรม และการสื่อสารแบบขนาน การสื่อสารแบบขนานนั้นสามารถส่งข้อมูลได้ทีละมากๆ เพราะมีสายสัญญาณหลายเส้น แต่ก็เป็นอุปสรรคในการสื่อสารระยะทางที่ไกล การสื่อสารในระยะทางที่ไกล การสื่อสารแบบอนุกรมจึงได้เปรียบในจุดที่ใช้สัญญาณเพียงแค่สองเส้น จึงเป็นการประหยัดและสะดวกในการใช้งาน

การสื่อสารข้อมูลหรืออินเทอร์เฟสถึงกันนั้น ย่อมต้องมีมาตรฐานในการสื่อสาร เพื่อให้เป็นที่เข้าใจกันทั้งฝ่ายส่งและฝ่ายรับ ที่ชื่อเรียกต่างกันไปตามมาตรฐานต่างๆ เช่น อาร์เอส-232 (RS-232; Recommended Standard – 232), RS-422,RS-485 และอื่นๆเป็นต้น

2.1 ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟสตามมาตรฐานต่างๆ

ปกติการอินเทอร์เฟสที่ออกแบบและจัดให้ใช้ควบคุมและตรวจสอบ (sense) ข้อมูล จะเป็นการอินเทอร์เฟสที่ใช้สัญญาณดิจิตอลเป็นหลักสัญญาณพวกนี้จะถูกส่งและรับมาจากรีจีสเตอร์ อินพุท/เอาท์พุท, อินพุทของ Interrupt-request, พอร์ทที่ใช้ในการทำ DMA และไทม์เมอร์/เคาน์เตอร์ การอินเทอร์เฟสที่กล่าวมานี้ทุกตัวเป็นการใช้สัญญาณดิจิตอลที่มีระดับสัญญาณแบบ TTL ในการอินเทอร์เฟส ดังนั้นถ้าเรานำอุปกรณ์อื่นที่ใช้ระดับสัญญาณ TTL ในการอินเทอร์เฟสด้วยเช่นกัน การทำอินเทอร์เข้ากับเครื่อง PC จะทำได้โดยตรง แต่ในหลายกรณีสัญญาณที่ใช้ไม่ได้เป็นระดับสัญญาณ TTL หรือไม่ได้เป็นสัญญาณดิจิตอล และระยะทางที่ทำการอินเทอร์เฟสมีระยะห่างมากซึ่งก่อให้เกิดปัญหาหลายประการ เราสามารถแก้ปัญหาเหล่านี้ได้โดยการใช้การอินเทอร์เฟสแบบต่างๆ ดังที่จะกล่าวต่อไป

2.1.1 การอินเทอร์เฟสตามมาตรฐาน RS-232-C

มาตรฐาน RS-232-C เป็นมาตรฐานที่ได้รับการพัฒนามานานและถูกใช้งานกันอย่างแพร่หลาย เราใช้ RS-232-C เชื่อมต่อ DTE (Data Terminal Equipment) เข้ากับ DCE (Data Communication Equipment) เช่น การต่อเทอร์มินัลเข้ากับโมเด็ม มาตรฐาน RS-232-C กล่าวถึงลักษณะทางกล, ลักษณะของสัญญาณไฟฟ้าและลักษณะการทำงานที่ใช้ในการอินเทอร์เฟส ตัวอย่างของอุปกรณ์ที่ใช้ในการอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

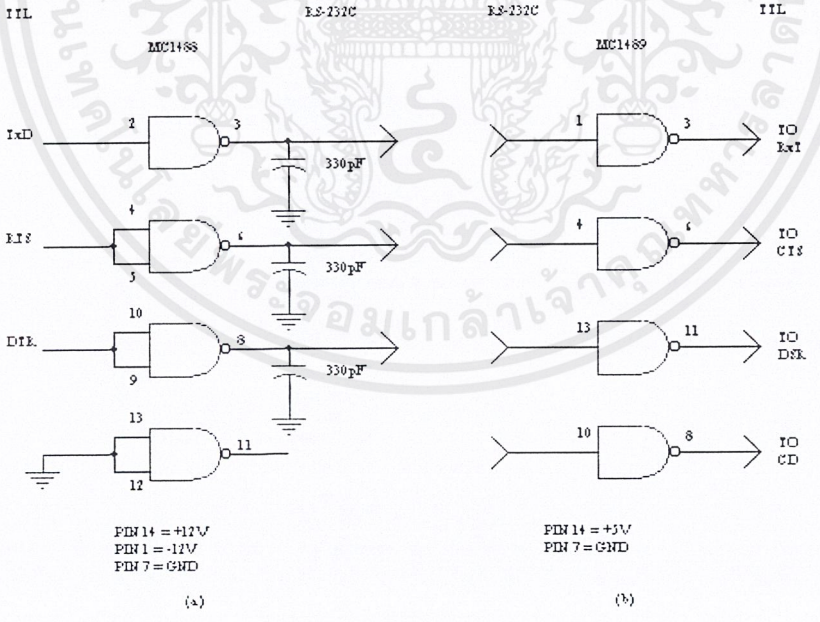
เทอร์เฟสตามมาตรฐาน RS-232-C ได้แก่ เทอร์มินัล, พล็อตเตอร์, ลอจิกอนาไลเซอร์ (Logic Analyzer) และเครื่องพิมพ์ ถ้าการประยุกต์ใช้งานของชิ้นงาน ต้องการทำอินเทอร์เฟสอุปกรณ์ เข้ากับการอินเทอร์-เฟสมาตรฐาน RS-232-C เราจำเป็นต้องแปลงระดับสัญญาณ TTL ให้เป็นระดับสัญญาณแบบอื่น ซึ่งรายละเอียดของระดับสัญญาณที่ใช้ได้กล่าวไว้ในมาตรฐาน RS-232-C

ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟส

มาตรฐาน RS-232-C ใช้สัญญาณเพียงเส้นเดียวในการส่งสัญญาณ โดยสัญญาณที่ส่ง ส่งไปได้ในทิศทางเดียว ในกรณีที่อัตราเร็วในการส่งข้อมูลมีค่าเท่ากับ 20k bps(กิโลบิทต่อวินาที) ซึ่งค่านี้เป็นค่าสูงสุดที่ใช้ในการส่งข้อมูล ระยะทางที่ใช้ส่งข้อมูลไม่ควรเกิน 50 ฟุต (ตามข้อกำหนดในมาตรฐาน) สำหรับการแทนแรงดันของระดับสัญญาณ (ในบทนี้เราแทนระดับสัญญาณด้วยลอจิกบวก) มีข้อกำหนดดังนี้ “1” แทนระดับแรงดันที่มีค่าระหว่าง +5 โวลท์ถึง +15 โวลท์ “0” แทนระดับแรงดันที่มีค่าระหว่าง –5 โวลท์ถึง –15 โวลท์

วงจรแปลงระดับแรงดัน

ตัวอย่างของวงจรที่ใช้แปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณที่กำหนดไว้ในมาตรฐาน RS-232-C และแปลงกลับจากระดับแรงดันในมาตรฐาน RS-232-C ไปเป็นระดับสัญญาณ TTL แสดงไว้ในรูปที่ 2.1



รูปที่ 2.1 วงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-232-C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

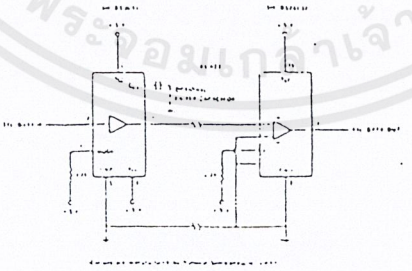
2.1.2 ลักษณะการอินเทอร์เฟสที่ใช้มาตรฐาน RS-423

มาตรฐาน RS-423 เป็นมาตรฐานที่ได้รับการพัฒนามาจากมาตรฐาน RS-232-C อุปกรณ์ที่ผลิตขึ้นมาใหม่ๆ มักจะใช้การอินเทอร์เฟสแบบนี้ โดยเฉพาะอุปกรณ์ที่ต้องการให้อัตราเร็วในการส่งข้อมูลมีค่าสูง ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟส

มาตรฐาน RS-423 ใช้สายสัญญาณเส้นเดียวในการส่งสัญญาณ โดยสัญญาณที่ส่งๆ ไปได้ทิศทางเดียว อัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 100k bps ที่ระยะห่าง 40 ฟุต ตัวรับข้อมูลเป็นแบบ balanced-1 (ดูรูปที่ 2.2) ดังนั้นตัวรับข้อมูล (receiver) จึงรับข้อมูลแบบขยายความแตกต่างของสัญญาณระหว่างสายกราวนด์กับตัวขับสัญญาณ(driver) การทำเช่นนี้ช่วยแก้ปัญหาในกรณีที่เกิดความแตกต่างระหว่างแรงดันที่กราวนด์ของตัวรับข้อมูล กับตัวขับสัญญาณ สำหรับการแทนระดับแรงดันนั่น ลอจิก “1” แทนระดับแรงดันที่อยู่ระหว่าง +4 โวลท์ ถึง +6 โวลท์ ส่วนลอจิก “0” แทนระดับแรงดันที่มีค่าระหว่าง –4 โวลท์ ถึง –6 โวลท์

วงจรแปลงระดับแรงดันของมาตรฐาน RS-423

วงจรที่ใช้แปลงระดับแรงดันจากระดับสัญญาณ TTL ไปเป็นระดับสัญญาณที่ใช้ในมาตรฐาน RS-423 และแปลงระดับสัญญาณจาก RS-423 กลับไปเป็นระดับสัญญาณ TTL แสดงไว้ในรูป 2.2



รูปที่ 2.2 วงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-423

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ลักษณะการอินเทอร์เฟสที่ใช้มาตรฐาน RS-422

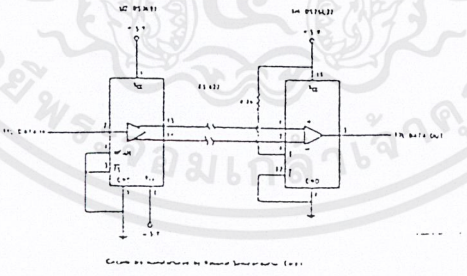
มาตรฐาน RS-422 ได้พัฒนาจากมาตรฐาน RS-423 ทำให้อัตราเร็วในการส่งข้อมูลมีค่าสูงขึ้นและระยะทางที่ใช้ส่งข้อมูลระหว่างตัวส่งและตัวรับมีระยะทางไกลขึ้น

ลักษณะสัญญาณที่ใช้ในการอินเทอร์เฟส

มาตรฐาน RS-422 ใช้ในการส่งข้อมูลในลักษณะของ one-way balanced-line (ดูรูป 2.3) โดยอัตราเร็วในการส่งข้อมูลมีค่าสูงถึง 10M bps ระยะห่างในการส่งข้อมูลสามารถขยายได้ถึง 4000 ฟุต ระดับแรงดันที่ส่งจากตัวขับสัญญาณจะมีค่าระหว่าง ±2โวลท์ ถึง±6โวลท์ นอกจากนี้ตัวขับสัญญาณสามารถจับสัญญาณที่มีระดับต่ำกว่า 200mV ได้

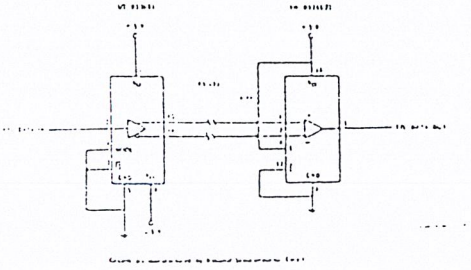
วงจรแปลงระดับแรงดันที่ใช้ในมาตรฐาน RS-422

วงจรที่ใช้แปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณที่ใช้ในมาตรฐาน RS-422 และแปลงจาก RS-422 กลับไปเป็นระดับสัญญาณ TTL แสดงไว้ในรูป 2.3 นอกจากนี้ผู้ผลิตบางบริษัทได้ทำวงจรขับสัญญาณเป็นแบบ tri-state ทำให้เราสามารถส่งข้อมูลได้สองทิศทางบนสายคู่เดียว(single pair) ซึ่งเป็นมาตรฐานใหม่คือ RS-485 คุณสมบัติข้อนี้ทำให้เราสามารถใช้มาตรฐาน RS-485 ในเนตเวิร์กที่มีโครงสร้างเป็นแบบ multidrop ซึ่งอุปกรณ์หลายๆ ตัวสามารถรับและส่งข้อมูลแบบ half duplex บนสายคู่เดียวได้



รูปที่ 2.3(ก) วงจรขับและรับสัญญาณที่ใช้มาตรฐาน RS-422

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

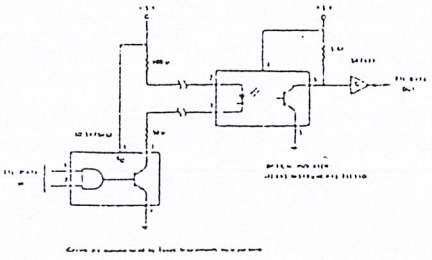


รูปที่ 2.3(ข) วงจรขับและรับสัญญาณที่ใช้มาตรฐาน RS-485

2.1.4 ลักษณะการอินเทอร์เฟสแบบ current-loop

การอินเทอร์เฟสแบบ current-loop มีข้อดีคือสามารถส่งข้อมูลได้ในระยะทางไกลๆ และค่าใช้จ่ายในการอินเทอร์เฟสแบบนี้มีราคาไม่สูงนัก หลักการของการอินเทอร์เฟสแบบนี้มีดังนี้ เมื่อลูปถูกปิดวงจร (closed-loop) ระดับแรงดันจะถูกเปลี่ยนเป็นกระแสตามสมการ V=IR เมื่อลูปถูกเปิดวงจรจะไม่มีกระแสไหลในลูป เนื่องจากวงจรของการอินเทอร์เฟสแบบนี้มีค่าอิมพิแดนซ์ต่ำดังนั้นจึงทนต่อสัญญาณรบกวนได้ดี เรามักใช้การอินเทอร์เฟสแบบนี้ในกรณีที่ต้องเดินสายผ่านบริเวณที่มีสัญญาณรบกวนมากๆ นอกจากนี้เราสามารถใช้การอินเทอร์เฟสแบบนี้แยกกราวนด์ของระบบสองระบบออกจากกัน ดังแสดงในรูป 2.4 ซึ่งเป็นการแยก current-loop ของวงจรผลิตสัญญาณ (transmitter) ออกจากวงจรรับสัญญาณ (receiver) วงจรนี้สามารถใช้ส่งข้อมูลได้ในอัตราเร็ว 50k bps โดยระยะทางที่ใช้ส่งข้อมูลมีค่าได้ไม่เกิน 3000 ฟุต ตัวจำกัดระยะทางที่ใช้ในการส่งข้อมูลคือ ความต้านทางของสายที่ประกอบกันอยู่เป็นลูป ซึ่งไม่ควรเกิน 30 โอห์ม (สำหรับวงจรในรูปที่ 2.4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 วงจรที่ใช้การอินเทอร์เฟสแบบ current-loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการและความรู้ทั่วไปเกี่ยวกับไมโครคอนโทรลเลอร์ MCS51 และการออกแบบวงจรอินฟราเรด

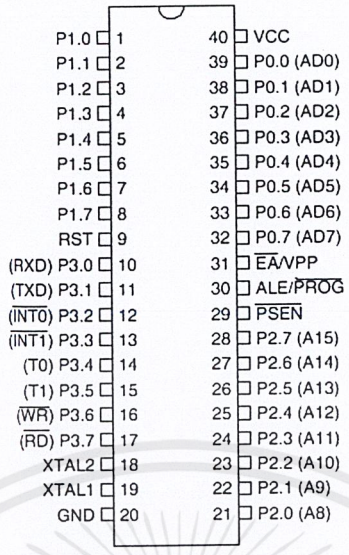
3.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูลMCS51

- ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์8051 และ 8031 สำหรับเบอร์8052 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memmory) ขนาด 128 ไบต์สำหรับเบอร์8052ขึ้นไปมีถึง 256 ไบต์
- หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มีไทม์เมอร์เคาท์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3ชุด)ทำงานได้ 4 โหมด
- รับอินเตอร์รัพท์ได้ 6 แหล่ง 5 เวคเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6เวคเตอร์
- มีพอร์ทสำหรับรับส่งข้อมูลอนุกรม (UART) 2 พอร์ทแบบ Full Duplex เลือกรูปได้ 4 โหมด
- มีคำสั่งในการ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

3.2 พอร์ทของ 8051

8051 เป็น ไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆ แสดงดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 การจัดวางขาของ 8051

พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0)ใช้งานได้ 2 หน้าที่คือ แอดเดรสบัสและดาต้าบัสเมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นไอโอพอร์ท ถ้าต้องการให้ทำงานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" ไปยังพอร์ทนี้

พอร์ท 1 (ขา 1-8)มีทั้งหมด 8 บิตคือ (P1.0-P1.7)มีโครงสร้างคล้ายพอร์ท 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน Internal Pull up Register

พอร์ท 2 (ขา 21-28)มีทั้งหมด 8 บิต คือขา(P2.7-P2.0) มีโครงสร้างคล้าย พอร์ท 0 โดยมี FET ตัวล่างเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน(Internal pull up) พอร์ทนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15-A8) และเป็นไอโอพอร์ทใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อน

พอร์ท 3 (ขา 10-17)มีทั้งหมด 8 บิต คือขา(P3.7-P3.0) มีโครงสร้างคล้ายพอร์ท 1 ทำงานได้ 2หน้าที่คือใช้เป็นไอโอพอร์ท ถ้าจะโปรแกรมให้เป็นอินพุทพอร์ทต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อน และอีกหน้าที่หนึ่งก็คือใช้ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไปสัญญาณต่างๆมีดังนี้

- P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม(UART)
- P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม(UART)
- P3.2/INT0 (External Interupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0
- P3.3/INT1 (External Interupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- P3.4/T0(Counter 0 External Intput) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจร Counter 0 (เป็นอินพุทโหมดเคาท์เตอร์)
- P3.5/T1(Counter 1 External Intput) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจร Counter 1 (เป็นอินพุทโหมดเคาท์เตอร์)
- P3.6/$\overline{WR}$ ($\overline{External}$ Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลลงในหน่วยความจำข้อมูลภายนอก
- P3.7/RD ($\overline{External}$ Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลลงในหน่วยความจำข้อมูลภายนอก
- ALE($\overline{ขา\ 30}$)เป็นขาส่งสโตรบสำหรับใช้ในการแลตซ์แอดเดรสไบต์ต่ำ(A7-A0)ที่ส่งออกมาจากพอร์ท 0 สัญญาณนี้จะแอคทีฟทุกๆ 2 ครั้งใน 1 แมชชีนไซเคิล
- PSEN(ขา 29)เป็นขาสโตรบที่ใช้สำหรับอ่านข้มูลจาก Program Memory ภายนอก สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้
- EA (ขา 31) ใช้เลือกหน่วยความจำโปรแกรมภายนอก

  ป้อน “0” จะอ่านโปรแกรมจากภายนอกชิพ

  ป้อน “1” จะอ่านโปรแกรมจากภายในชิพ

- RST (ขา 9) ขารีเซ็ท จะรีเซ็ทก็ต่อเมื่อป้อนลอจิก “1” เข้าที่นี้อย่างน้อย 2 แมชชีนไซเคิล
- XTAL1(ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุทเข้าสู่วงจรออสซิลเลเตอร์ภายใน
- XTAL2(ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาท์พุทของวงจรออสซิลเลเตอร์ภายใน

3.3 หน่วยความจำของMCS51

หน่วยความจำที่ใช้กับMCS-51มีอยู่2 ชนิดคือ

3.3.1 หน่วยความจำสำหรับเก็บโปรแกรม(Program Memory)

เป็นหน่วยความจำที่ใช้เก็บโปรแกรมสั่งงาน บรรจุอยู่ในชิพ 8051 ส่วนที่เป็น Program Memory ก็คือ ROM ขนาด 4 กิโลไบต์นั่นเอง แต่ถ้าเป็นเบอร์ 8052 จะมีROM ขนาด 8 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

หน่วยความจำสำหรับเก็บข้อมูล แบ่งเป็น 2 ส่วนคือ หน่วยความจำภายในชิพมีเพียง 128 ไบต์ สำหรับเบอร์และ 256 ไบต์สำหรับเบอร์ 8052 ขึ้นไปและหน่วยความจำภายนอกชิพมีความจุ 64 กิโลไบต์

3.4 ไทม์เมอร์/เคาท์เตอร์

ไทม์เมอร์เคาท์เตอร์สามารถเลือกให้มีการทำงานเป็นไทม์เมอร์ หรือเคาท์เตอร์อย่างใดอย่างหนึ่ง โดยเลือกที่บิต C/T ในรีจิสเตอร์ใช้งานเฉพาะ TMOD โหมดไทม์เมอร์และเคาท์เตอร์จะใช้ up Counter Register (TH_x ,TL_x)ตัวเดียวกันซึ่งเป็นแบบนับขึ้น

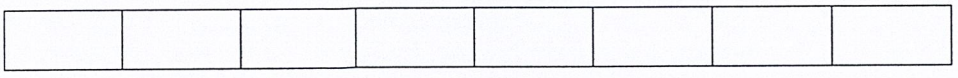
โหมดไทม์เมอร์ up Counter Register (TH_x ,TL_x) จะถูกเพิ่มค่าทุกๆ 1 แมชชีนไซเคิล(12 คาบเวลาของCPU osc)โหมดนี้ไม่ต้องป้อนสัญญาณจากภายนอกเข้ามาแต่จะใช้สัญญาณ CPU osc

โหมดเคาท์เตอร์ up Counter Register (TH_x ,TL_x) จะถูกเพิ่มค่าทีละหนึ่งเมื่อป้อนสัญญาณคล๊อกจากภายนอกเข้ามา 1 ลูกเข้ามาทางขา T0 (pin) หรือT1 (pin) อยู่ที่ขา 14 และขา 15 ตามลำดับโดยไม่สนใจคาบเวลาของพัลซ์แต่ละลูก การตรวจสอบพัลซ์ที่เข้ามาทางขานี้โดยจะตรวจสองทุกๆ S5P2 ของแต่ละแมชชีนไซเคิล ดังนั้นการตรวจสอบคล้อค แต่ละลูกจะต้องใช้เวลาถึง 2 แมชชีนไซเคิล(1/24 คาบเวลาของ CPU osc)

3.4.1 โครงสร้างของ ไทม์เมอร์ และ เคาท์เตอร์ มีโครงสร้างดังนี้

- เคาท์เตอร์แบบนับขึ้น up Counter Register(TH_x, TL_x)
- ส่วนเลือกโหมดไทม์เมอร์และเคาท์เตอร์ เลือกที่บิต C/T
- ส่วนควบคุมการนับ และหยุดนับ (Start Counter)ควบคุมที่บิต TR_x ,GATE และสัญญาณจากภายนอกที่ขา INT_x(pin)
- ในโหมดเคาท์เตอร์จะรับอินพุทพัลซ์จากภายนอกที่ขา T_x pin
- ในโหมดไทม์เมอร์จะรับโหมดไทม์เมอร์จะรับอินพุทจากคล้อคซีพียูที่หารด้วย 12 แล้ว
- โหมดเคาท์เตอร์และไทม์เมอร์ใช้เคาท์เตอร์แบบตัวเดียวกันแบบนับขึ้น

3.4.2 Timer/Counter Mode Control Register(TMOD)อยู่ใน SFR ตำแหน่งที่(89H)



TIMER 0 TIMER 1

รูปที่ 3.2 แสดงรายละเอียดของรีจิสเตอร์ TMOD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$GATE_x$ - เป็นบิตตเลือกการสตาร์ทไทม์เมอร์เคาท์เตอร์ x

0 ควบคุมโดย Software (Internal Control)

1 ควบคุมโดย Hardware (External Control โดยการทริกจากภายนอก)

C/T - บิตเลือกการทำงานของไทม์เมอร์เคาท์เตอร์โดยเลือกดังนี้

ถ้า C/T = 0 เป็นการเลือกโหมดไทม์เมอร์(นับจำนวนแมชชีนไซเคิล)

ถ้า C/T = 1 เป็นการเลือกโหมดไทม์เมอร์(นับจำนวนพัลซ์จากภายนอก)

M1,M0 - เลือกโหมดการทำงานได้ 4 โหมด

ตารางที่ 3.1 แสดงการเลือกโหมด Timer/Counter

M1	M0	โหมด	การทำงาน
0	0	0	13 บิตไทม์เมอร์หรือเคาท์เตอร์
0	1	1	16 บิตไทม์เมอร์หรือเคาท์เตอร์
1	0	2	8 บิตไทม์เมอร์หรือเคาท์เตอร์แบบโหลดซ้ำได้อัตโนมัติ
1	1	3	บิตไทม์เมอร์หรือเคาท์เตอร์ โดยใช้ TL0 8 บิตไทม์เมอร์ โดยใช้ TH0
1	1	4	

3.4.3 Timer/Counter Control Register(TCON)อยู่ใน SFR ตำแหน่งที่(088H)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 3.3 แสดงรายละเอียดของรีจิสเตอร์ TCON

TF1 - แฟลกแสดงการเกิดโอเวอร์โฟลว์ขอวไทม์เมอร์ 1 จะเซ็?เมื่อไทม์เมอร์ 1 เกิดโอเวอ์โฟลว์และจะถูกเคลียร์เองเมื่อซีพียูย้ายการทำงานไปที่โปรแกรมบริการอินเตอร์รัปต์หรือใช้คำสั่ง CLR TF1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TR1 - บิตควบคุมการนับของไทม์เมอร์เคาท์เตอร์ 1 ควบคุมจากโปรแกรม

1 ไทม์เมอร์หรือ เคาท์เตอร์ 1 เริ่มทำงานต่อ

0 ไทม์เมอร์หรือเคาท์เตอร์ 1 หยุดทำงาน

TF0 - แฟลกแสดงการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 0 ถูกเซ็ทเมื่อไทม์เมอร์ 0 เกิดโอเวอร์โฟลว์ เช่นเดียวกับ TF1

TR0 - เช่นเดียวกับ TR1 แต่ใช้ควบคุมไทม์เมอร์ 0

IE1 - แฟลกแสดงการเกิดสัญญาณอินเตอร์รัปต์ภายนอกหมายเลข 1เมื่อมีสัญญาณอินเตอร์รัปต์เข้ามาที่ขา $\overline{\text{INT 0}}$ และถูกเคลียร์เองโดยคำสั่ง RETI ในโปรแกรมส่วนบริการอินเตอร์รัปต์

IT1 - แฟลกเลือกประเภทการตรวจสอบสัญญาณอินเตอร์รัปต์ที่เกิดขึ้นที่ขา INT 1 โดย

1 จะตรวจสอบการเปลี่ยนระดับแบบขอบขาลงที่ขาINT 1

0 จะตรวจสอบระดับศูนย์ของสัญญาณที่ขา INT 1

IE0 - แฟลกแสดงการเกิดสัญญาณอินเตอร์รัปต์ภายนอกหายเลข 0 เมื่อมีสัญญาณอินเตอร์รัปต์เข้ามาที่ขา INT 1 และถูกเคลียร์โดยคำสั่ง RETI ที่อยู่ในโปรแกรมส่วนบริการอินเตอร์รัปต์

IT0 - เช่นเดียวกับ IT1

3.5 รีจิสเตอร์ฟังก์ชั่นพิเศษ

Special Function Register (SFR)เป็นรีจิสเตอร์ที่ใช้งานในหน้าที่ต่าง ๆกันซึ่งบรรจุไว้ในหน่วยความจำภายใน(Internal Ram) ตำแหน่ง (80h-FFh) โดยมีรายละเอียดดังนี้

3.5.1Accumulator (ACC)อยู่ในตำแหน่ง 0E0h

เป็นรีจิสเตอร์ 8 บิต ที่ใช้ประมวณผลทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร หรือเป็นทางผ่านในการเคลื่อนย้ายข้อมูล

3.5.2 B Register (B) อยู่ที่ตำแหน่ง OF0h

เป็นรีจิสเตอร์ 8 บิต ที่ใช้ในคำสั่งคูณ และหาร รวมกับรีจิสเตอร์ A เช่นMUL AB,DIV AB

3.5.3 Program Status Word อยู่ที่ตำแหน่ง ODOh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นรีจิสเตอร์ 8 บิต แต่ใช้งานเพียง 7 บิต 5 บิต เป็นแฟลกซ์และอีก 2 บิต เป็นบิตเลือกรีจิสเตอร์แบงค์เลือกได้ 3 แบงค์ คือแบงค์ 0-3 นั่นเอง

3.5.4 Stact Pointer (SP) อยู่ที่ตำแหน่ง 081 h

เป็นรีจิสเตอร์ 8 บิต ที่ใช้ชี้ตำแหน่งของสแตคทุกครั้งที่ถูกรีเซ็ท ก็จะมีค่าเท่ากับ 07H ทันทีจะเพิ่มขึ้น 1 เสมอ เมื่อทำคำสั่ง CALL,PUSH และจะลดลงเมื่อทำคำสั่ง RET,RETI,POP ฯลฯ

3.5.5 Data Pointer Register อยู่ที่ตำแหน่ง 82h และ 83h

เป็นรีจิสเตอร์ 16 บิต แบ่งเป็น 8 บิตล่าง และ 8 บิตบน มีชื่อว่า DPH และ DPL อยู่ที่ตำแหน่ง 83h และ 82h ตามลำดับ มีประโยชน์ในการชี่แอดเดรสของหน่วยความจำ เช่น MOVX A,@DPTR,JMP@A+DPTR,MOVC A,@A+DPTR

3.5.6 Port (0-3)อยู่ที่ตำแหน่ง 80h,90h,0A0h,0B0h

เป็นพอร์ท 8 บิต มีชื่อว่า P0,P1,P2,P3 อยู่ที่ตำแหน่ง 80h,90h,0A0h,0B0h ตามลำดับ

3.5.7 Serail Data Buffer (SBUF) อยู่ที่ตำแหน่ง 99h

เป็นรีจิสเตอร์ 8 บิตที่ใช้เก็บข้อมูลก่อนที่จะส่งข้อมูลอนุกรมออกไปหรือเป็นบัฟเฟอร์เก็บข้อมูลในช่วงที่รับข้อมูลเข้ามา SUBF จะมีอยู่ 2 ตัว คือ รับและส่งแยกกัน การรับส่งข้อมูลอนุกรมมี 4 โหมด ด้วยกัน

3.5.8Interrupt Priority Register (IP) อยู่ตำแหน่ง

เป็นรีจิสเตอร์ควบคุมลำดับความสำคัญของอินเตอร์รัพต์

3.5.9Enterrupt Enable Register (IE) อยู่ตำแหน่ง 0A8h

เป็นรีจิสเตอร์ที่ใช้กำหนดให้ทำหรือไม่ทำอินเตอร์รัพท์ (Enable และ Disable) อินเตอร์รัพท์ จาก 6 แหล่ง 5 Vecter เช่นถ้าต้องการให้พอร์ทอนุกรมทำอินเตอร์รัพต์ได้ก็ต้องเซ็ทบิต ES (บิต D4 ในIE) ด้วยคำสั่งดังนี้ SETB ES หรือจะห้ามไม่ให้พอร์ทอนุกรมทำอินเตอร์รัพต์ ก็ต้องใช้คำสั่ง CLR ES

3.5.10 Timer/Counter Mode Control Register (TMOD) อยู่ที่ตำแหน่ง 089h

เป็นรีจิสเตอร์เลือกการทำงานเลือกได้ 4 โหมดโดยเลือกที่บิต M0และM1 ส่วนบิต Gate เป็นบิตที่ใช้เลือกการสตาร์ท Timer/Counter โดยควบคุมร่วมกับ TRx ในTCON ส่วนบิตC/Tใช้เลือกTimer หรือ Counter ดังรายละเอียดกล่าวในเรื่องTimer/Counter ที่ผ่านมาแล้ว

3.5.11 Timer/Counter Control Register (TCON)

เป็นรีจิสเตอร์ที่ใช้ควบคุมให้ Timer/Counter เริ่มนับโดยเซ็ทที่บิตTRx(TR0 หรือTR1 นั่นเอง)อีกหน้าที่หนึ่งใช้แสดงผลการเกิดโอเวอร์โฟลว์ของTimer/Counterโดยจะแสดงที่บิต TFx (TF0 หรือ TF1นั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอง)อีกหน้าที่หนึ่งคือแสดงสถานะเมื่อเกิดการอินเตอร์รัปต์จากภายนอกโดยแสดงที่บิต IE0หรือIE1 หน้าที่สุดท้ายคือบิตเลือกสัญญาณอินเตอร์รัปต์ว่าจะเอาแบบ Trig ที่ขอบขาลงหรือ Trigที่ระดับศูนย์รายละเอียดกล่าวไว้ในเรื่องของ Timer/Counter

3.5.12 Power Control Register(PCON) อยู่ที่ตำแหน่ง 87h

เป็นรีจิสเตอร์ที่ใช้ควบคุมการประหยัดพลังงานมีการทำงาน 2โหมด คือ Ideal mode และ Power down mode และยังใช้เพิ่ม Baud Rate เป็น 1 เท่าหรือ 2 เท่าโดยเซ็ทที่บิต SMOD

รายละเอียดของ PCON มีดังนี้

- SMOD = 0 ความเร็ว Buad Rate เมื่อส่งข้อมูลอนุกรมโหมด 1,2,3 จะเป็น 1 เท่า
- SMOD = 1 ความเร็ว Buad Rate เมื่อส่งข้อมูลอนุกรมโหมด 1,2,3 จะเป็น 2 เท่า(Doubbl Baud Rate)
- SMOD = 1 GF1และ GF0 เป็นแฟลกซ์ที่ใช้งานทั่วไป
- PD (Power Down Mode) เมื่อถูกเซ็ทให้เป็น 1 จะเข้าสู่โหมดนี้ทันที
- IDL (Ideal Mode)เมื่อถูกเซ็ทเป็น 1 จะเข้าสู่โหมดนี้ทันที
- Serial Port Control Register(SCON) อยู่ที่ตำแหน่ง 89h

ใช้เลือกโหมดการทำงาน(เลือกได้ 4 โหมด) เลือกที่ SM1 ,SM0 ส่วน SM2 ใช้เลือกการทำงานแบบ Processor เดี่ยวหรือหลาย Processor บิต REN ใช้ควบคุมให้รับหรือไม่รับข้อมูลเข้ามา บิต TB8 คือที่ 1 บิตที่ 9 ที่ส่งออกไป และ RB8 ก็คือ ข้อมูลบิตที่ 9 ที่รับเข้ามาได้สุดท้าย คือ TI และ RI ก็คือ Transmit interupt Flag และ Receive Interupt Flag ตามลำดับ (TI จะเซ็ทเมื่อสิ้นสุดส่งข้อมูลออกไป 1 ไบต์และ RI จะเซทเมื่อสิ้นสุดรับข้อมูลเข้ามา 1 ไบต์)

3.6 การออกแบบวงจรอินฟราเรด

วงจรอินฟราเรด (Infrared) นี้จะติดตั้งตัวส่งและตัวรับบริเวณประตูที่เป็นทางเข้าออก โดยใช้สัญญาณอินพุทจากวงจรอินฟราเรดทางด้านตัวส่ง ส่งไปยังตัวรับซึ่งในสภาวะปกติที่ไม่มีอะไรมาบังระหว่างตัวส่งกับตัวรับแสดงว่าไม่มีผู้ผ่านเข้าออก แต่ถ้ามีการขวางทางของสัญญาณเกิดขึ้นก็จะแสดงว่ามีผู้ผ่านเข้าออก และจะส่งสัญญาณไปว่าผู้นั้นเข้าหรือออกจากห้องนั้น โดยสามารถตรวจสอบได้จากการเปลี่ยนแปลงของสัญญาณจากตัวรับอินฟราเรด เพราะที่ประตูนี้เราจะติดตั้งวงจรอยู่สองชุด โดยกำหนดว่าชุดไหนส่งสัญญาณเข้ามาก่อนหลังเป็นตัวตรวจว่ามีคนเข้าหรือออกจากห้องนั้น

ส่วนการออกแบบวงจรนี้ เราจะใช้ Ic 555 สร้างสัญญาณพัลส์โดยการกระตุ้นของตัวเก็บประจุ Ca ผ่าน Ra และ Rb เมื่อแรงดันคร่อม Ca มีค่า (2/3)Vcc จะทำให้วงจรเปรียบเทียบแรงดันชุดบนทำงานไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระตุ้นให้วงจรควบคุมฟลิปฟลอปทำงานไปอีกต่อหนึ่งเป็นผลให้ Ca คายประจุผ่าน Rb และขา 7 ลงกราวน์ ช่วงนี้เอาท์พุทขา 3 จะมีแรงดันต่ำและเมื่อแรงดันที่ Ca ลดลงมาถึง (1/3)Vcc ก็จะไปกระตุ้นให้วงจรเปรียบเทียบชุดล่างทำงานกระตุ้นวงจรฟลิปฟลอป ทำให้แรงดันเอาท์พุทสูงขึ้นและ Ca ไม่สามารถคายประจุได้อีกเพราะทรานซิสเตอร์ภายในซึ่งทำหน้าที่เป็นสวิตซ์จะอยู่ในสภาวะไม่ทำงาน ดังนั้น Ca จะเริ่มการเก็บประจุอีกจนแรงดันเท่ากับ (2/3)Vcc ก็จะเริ่มทำงานแบบเดิมอีกครั้ง ลักษณะเป็นดังรูปที่

จากลักษณะการทำงานของวงจร จะเห็นว่าช่วงความกว้างของพัลส์ที่ได้ทางเอาท์พุทจะขึ้นอยู่กับการเก็บประจุ และการคายประจุ ของ Ca จะได้ว่า

ช่วงการเก็บประจุ ; $T1 = 0.693(Ra+Rb)Ca$

ช่วงการคายประจุ ; $T2 = 0.693RaCa$

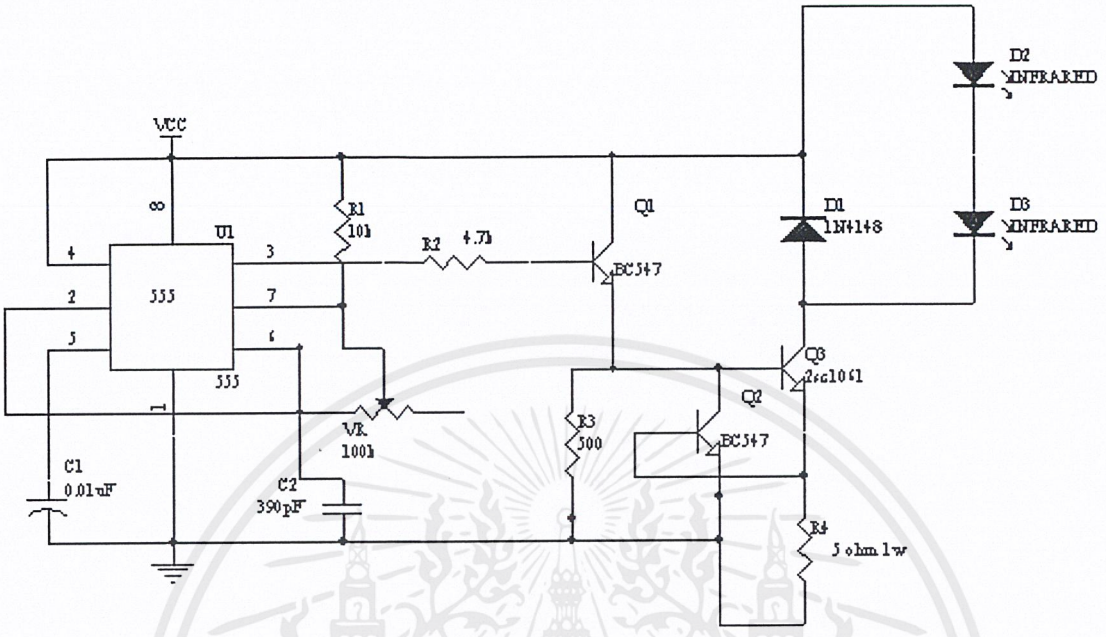
$f = 1/(T1+T2)$

วงจรอินฟราเรดทางด้านส่งนี้ ใช้ Ic 555 ในการสร้างพัลส์ โดยความถี่ที่สร้างขึ้นกำหนดจาก Ra Rb และ Ca จากสูตรที่กล่าวมาแล้ว ในวงจรนี้เราสร้างสัญญาณพัลส์ความถี่ประมาณ 38 kHzเพื่อให้ตรงกับความถี่ของวงจรทางด้านรับอินฟราเรด สัญญาณพัลส์ที่สร้างได้นี้จะถูกส่งไปยังส่วนไดรเวอร์ โดยใช้ เพาเวอร์ทรานซิสเตอร์ เบอร์ 2sc1061 ส่วน BC547 ทำหน้าที่เป็นสวิทซ์ และทำหน้าที่จำกัดกระแสที่ไหลผ่าน MLED 81 (ตัวส่งอินฟราเรด) ตามลำดับ

ส่วนทางด้านของตัวรับอินฟราเรดนั้น เราจะใช้เป็น Photo Modules เบอร์ TSOP1838 ที่สามารถรับสัญญาณอินฟราเรดที่ 38 kHz ซึ่งในตัว Photo Modules TSOP1838 จะประกอบด้วยส่วนรับแสงอินฟราเรด เฟสล็อคลูปที่ความถี่ 38 kHz และภาคขยายสัญญาณเอาท์พุท ซึ่งในส่วนของตัวรับนี้จะ Active Low คือ ในสภาวะปกติที่สามารถรับส่งสัญญาณได้ตามปกติ เอาท์พุทที่ได้จากตัวรับนี้จะเป็น Low (0) และถ้ามีการขวางการส่งของสัญญาณ เอาท์พุทที่ได้จะเป็น Hi (1)c]และเอาท์พุทที่ได้นี้ก็จะถูกส่งไปยังไมโครคอนโทรลเลอร์ เพื่อนำไปใช้ในการสั่งงานอย่างอื่นต่อไปลักษณะของวงจรทางด้านส่งเป็ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 3.4 แสดงวงจรอินฟราเรดทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42278

บทที่ 4
การออกแบบวงจรส่วนควบคุมไฟ

4.1 ระบบควบคุมไฟทั่วไป

4.1.1 หลักการทั่วไป

ในระบบควบคุมไปแสงสีต่างๆนั้นมีหลายขนาดและหลายชนิด มีตั้งแต่ที่สามารถขับหลอดได้ไม่กี่วัตต์ขึ้นไปจนถึงระดับพันวัตต์เลยทีเดียว แต่ในระบบเก่าๆ การขับหลอดไฟทางเอาท์พุทเป็นการเชื่อมต่อกับวงจรทางแรงดันสูง จึงเป็นอันตรายต่อวงจรและผู้ใช้งานเองด้วย เนื่องจากเมื่อเกิดการขัดข้องขึ้นที่วงจรก็จะทำให้แรงดันสูงผ่านเข้ามาสู่ทางอินพุทได้ ผู้ใช้งานจะถูกไฟฟ้าดูดได้ง่าย

ระบบการควบคุมได้มีการพัฒนาและแก้ไขข้อบกพร่องนี้ โดยการแยกวงจรทางภาคอินพุทกับเอาท์พุทออกจากกัน โดยอาศัยหม้อแปลงเป็นตัวแยก แต่ก็ยังไม่ดีเท่าควร เนื่องจากต้องสิ้นเปลืองเนื้อที่ในการติดตั้งเข้าไปในกล่อง และประสิทธิภาพการทำงานก็ไม่ดีพอ เช่นเดียวกับระบบที่ใช้รีเลย์ทางเอาท์พุท ซึ่งรีเลย์จะมีอายุการใช้งานของหน้าสัมผัสน้อย และมีคราบเขม่าจับที่หน้าสัมผัสจึงทำให้มีปัญหาเวลาใช้งานจริง

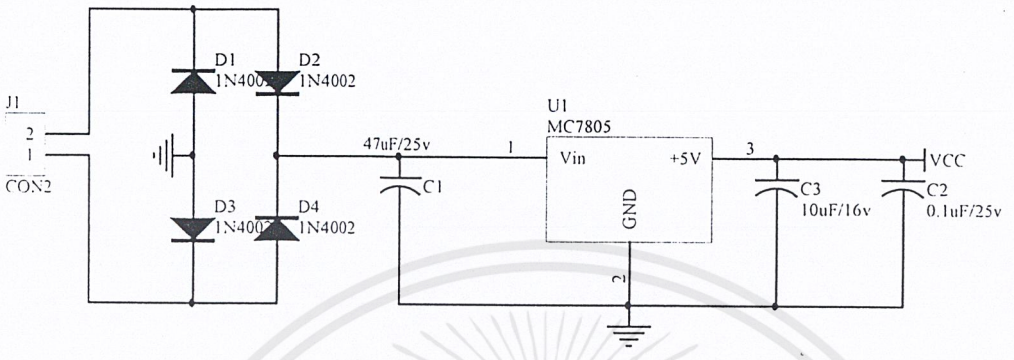
การแก้ปัญหาที่ดีที่สุดในการแยกวงจรควบคุมทางอินพุทกับเอาท์พุทแรงดันไฟสูงออกจากกันสามารถทำได้โดยการใช้อุปกรณ์เชื่อมโยงทางแสงที่เรียกว่า ออปโต้ไอโซเลเตอร์ (optocouple isolator) เป็นระบบที่นิยมใช้งานและออกแบบกันมากในขณะนี้ เพราะไม่มีปัญหาเรื่องหน้าสัมผัส การลัดวงจรถึงกันระหว่างอินพุทกับเอาท์พุท ซึ่งออปโต้ไอโซเลเตอร์นี้มีขนาดเล็กบรรจุอยู่ภายในตัวถังแบบ DIP ไอซี 6 ขา และที่สำคัญคือราคาถูกกว่าระบบอื่นๆด้วย

4.2 การออกแบบวงจรควบคุมไฟ

4.2.1 วงจรแรงดันอ้างอิง +5 โวลท์

ในการใช้งานชุดควบคุมไฟ จะต้องอาศัยแรงดันอ้างอิง (Vcc) เป็นไฟเลี้ยงอุปกรณ์ ไอซีต่างๆและยังเป็นแรงดันอ้างอิงในวงจรเปรียบเทียบแรงดัน โดยวงจรมีการทำงานก็คือ แรงดันไฟเอซีผ่าน ฟูลเวฟบริดจ์เรคติไฟร์ (Full Wave bridge rectifier) แรงดันดีซีที่ได้คร่อมตัวเก็บประจุ เพื่อกรองแรงดันให้เรียบก่อนผ่านวงจรแรงดันคงที่ (Regulate) ไอซี 7805 ซึ่งจะได้เอาท์พุทเป็น แรงดันไฟตรง +5 โวลท์ ตามรูปที่ 4.1

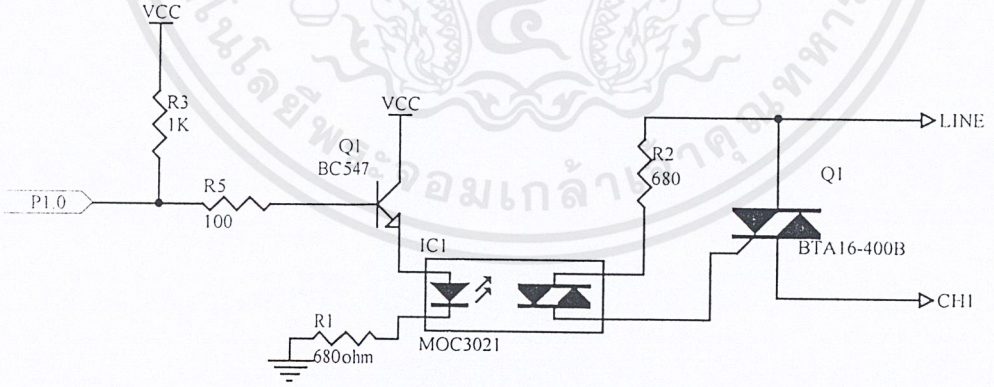
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.1 แสดงส่วนวงจรแรงดันไฟอ้างอิง +5 โวลต์

4.2.2 วงจรทางแรงดันไฟสูง

เอาท์พุทที่ได้จะไป On วงจรออปโต้ไอโซเลท (Opto Isolate) ตามรูปที่ 4.2 ซึ่งแรงดันที่ตกคร่อมจะทำการไบอัส ไตรแอกในส่วนแรงดันสูงทำให้สามารถควบคุมกระแสที่ไหล หรือเป็นการควบคุมความสว่างของหลอดไฟนั่นเอง



รูปที่4.2 ส่วนออปโตโซเลทและวงจรทางแรงดันสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทำงานของส่วนโปรแกรม

ในการเขียนโปรแกรมส่งและรับข้อมูล นั้นประกอบด้วย ไมโครคอนโทรลเลอร์ 2 ตัว และ เครื่องคอมพิวเตอร์ที่ใช้เป็น Server ซึ่งการส่งและรับข้อมูลนั้น ทุก ๆ ตัวที่กล่าวมาจะได้รับข้อมูลทั้งหมด จึงต้องทำการระบุ address ของแต่ละตัว เพื่อไม่ให้เกิดความผิดพลาด โดยจะส่งไปเป็น byteแรกของข้อมูล เมื่อทุกตัวได้รับก็จะเช็ค address ว่าเป็นข้อมูลของมันหรือไม่ ถ้าใช่จึงนำข้อมูลมาประมวลผลต่อไป

5.1 โปรแกรมในส่วนไมโครคอนโทลเลอร์

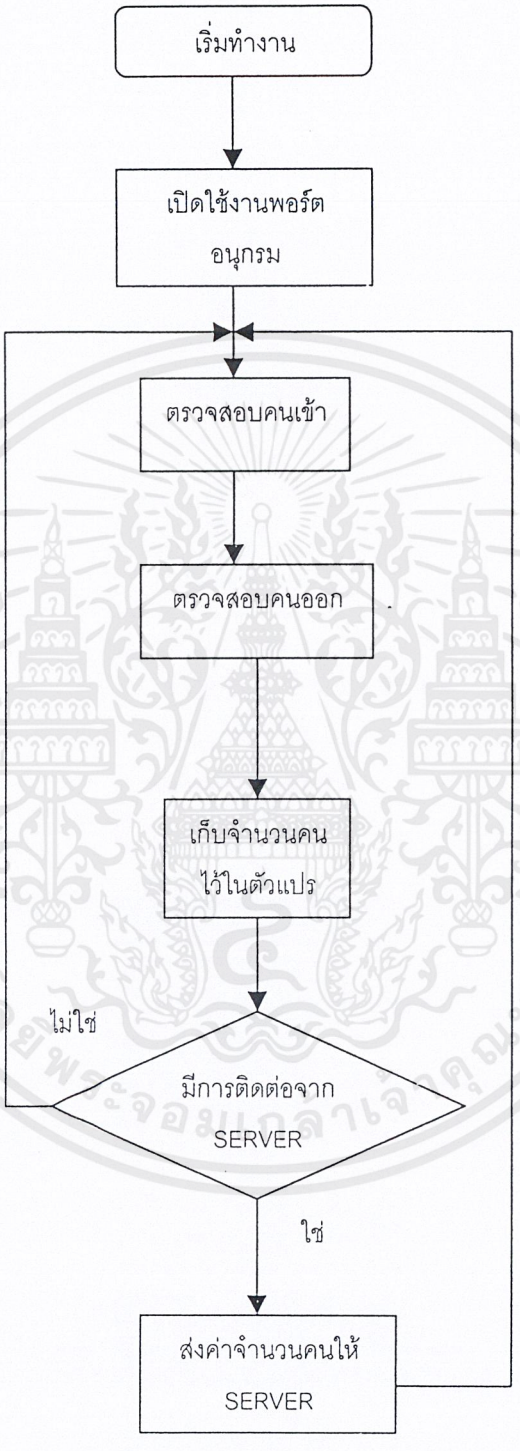
จาก Flow Chart รูปที่5.1 จะเห็นว่า โปรแกรมนี้จะเริ่มจากการที่ตัวของไมโครคอนโทรลเลอร์ จะรอสัญญาณจาก sensor ตัวรับอินฟราเรดซึ่งจะส่งมาว่า มีคนเข้าหรือออกจากห้อง และถ้าตรวจพบว่ามีคนเข้ามาคอนโทรลเลอร์ก็จะสั่งให้อุปกรณ์ไฟฟ้าทำงาน แต่ถ้าไม่พบอะไรก็จะเข้าไปเช็คว่ามีการสั่งงานจากคอมพิวเตอร์หรือไม่ ถ้าไม่มีก็วนกลับไปตรวจเช็คที่ตัว sensor ใหม่ แต่ถ้าคอมพิวเตอร์มีคำสั่งส่งมาให้ทำงาน คอนโทรลเลอร์ก็จะนำค่าที่ได้นี้ ไปเทียบกับค่าในตารางคำสั่งว่าคำสั่งนี้ให้ทำอะไรกับอุปกรณ์ไฟฟ้าตัวใดต่อไป แล้วจึงวนกลับไปตรวจเช็คสัญญาณต่อไป แต่ถ้าสัญญาณจาก sensor ส่งเข้ามาว่าคนออกจากห้องไปหมดแล้ว คอนโทรลเลอร์ก็จะสั่งงานให้อุปกรณ์ไฟฟ้าทุกตัวเลิกใช้งานทันที แต่เราก็สามารถสั่งให้อุปกรณ์ไฟฟ้าทำงานได้ แม้จะไม่มีใครอยู่ในห้องก็ตาม โดยสั่งงานจากคอมพิวเตอร์

5.2 โปรแกรมในส่วนของ Delphi

Flow chart ในรูปที่ 5.2 โปรแกรม Delphi แบ่งเป็น 2 ส่วน คือ Server Window (รูปที่5.2.1) สามารถที่จะสั่งงานผ่าน Window นี้ได้เลย และมันยังเก็บสถานะของแต่ละอุปกรณ์ไว้ เพื่อรายงานแก่ผู้ใช้งานระยะไกลได้อีกด้วย แล้วมันยังสามรถสั่งปิดไฟทั้งหมดอย่างอัตโนมัติเมื่อไม่มีใครอยู่ในห้องแล้ว ซึ่งมันทราบว่ามีคนอยู่เท่าไรในห้องโดยการถามจากไมโครคอนโทรลเลอร์ และอีกส่วนคือ Client Window (รูปที่5.2.2) เป็น Window สำหรับผู้ใช้งานในระยะไกล (แต่อยู่ในระบบ LAN เดียวกัน) สามารถสั่งงานได้โดยสั่งผ่าน Server Window อีกทีหนึ่ง โดยต้องกำหนด IF address ของเครื่อง และ Port ที่จะทำการติดต่อของเครื่องที่เป็น Server โดยสถานะของอุปกรณ์แต่ละตัว มันจะทราบโดยส่งข้อมูลไปถามที่ Server

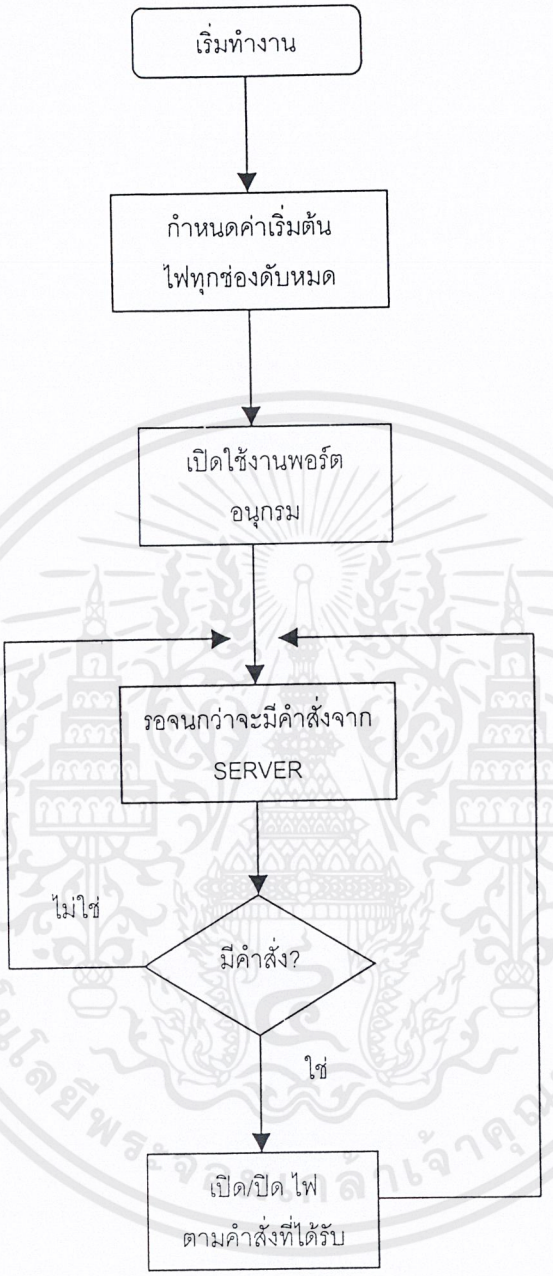
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่5.1 แสดง Flow Chart การทำงานของโปรแกรมของไมโครคอนโทรเลอร์



รูปที่ 5.1(ก) แสดง Flow Chart การทำงานของโปรแกรมไมโครคอนโทรเลอร์ด้าน Sensor

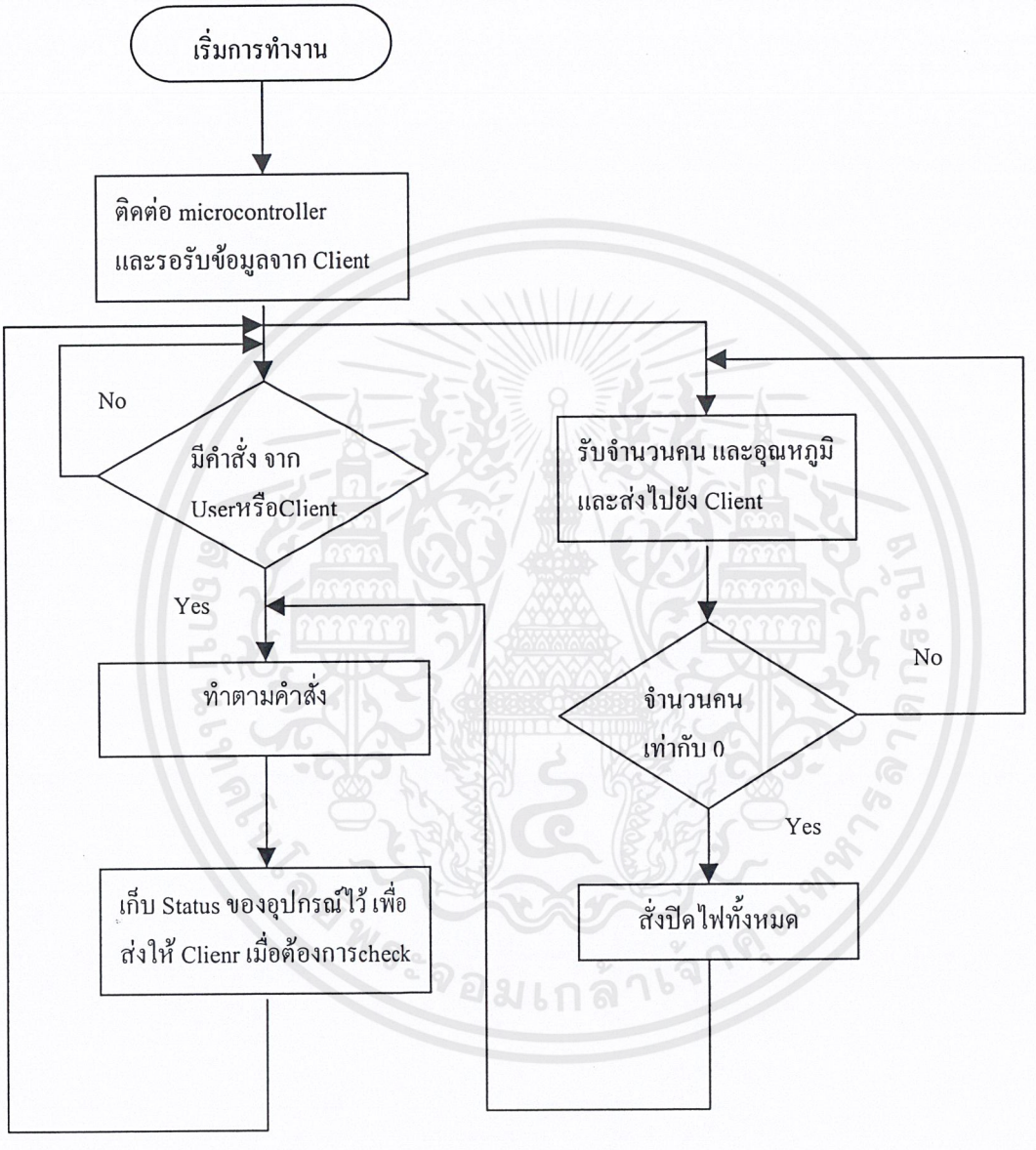
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1(ข) แสดง Flow Chart การทำงานของโปรแกรมไมโครคอนโทรเลอร์ด้าน Line Switch

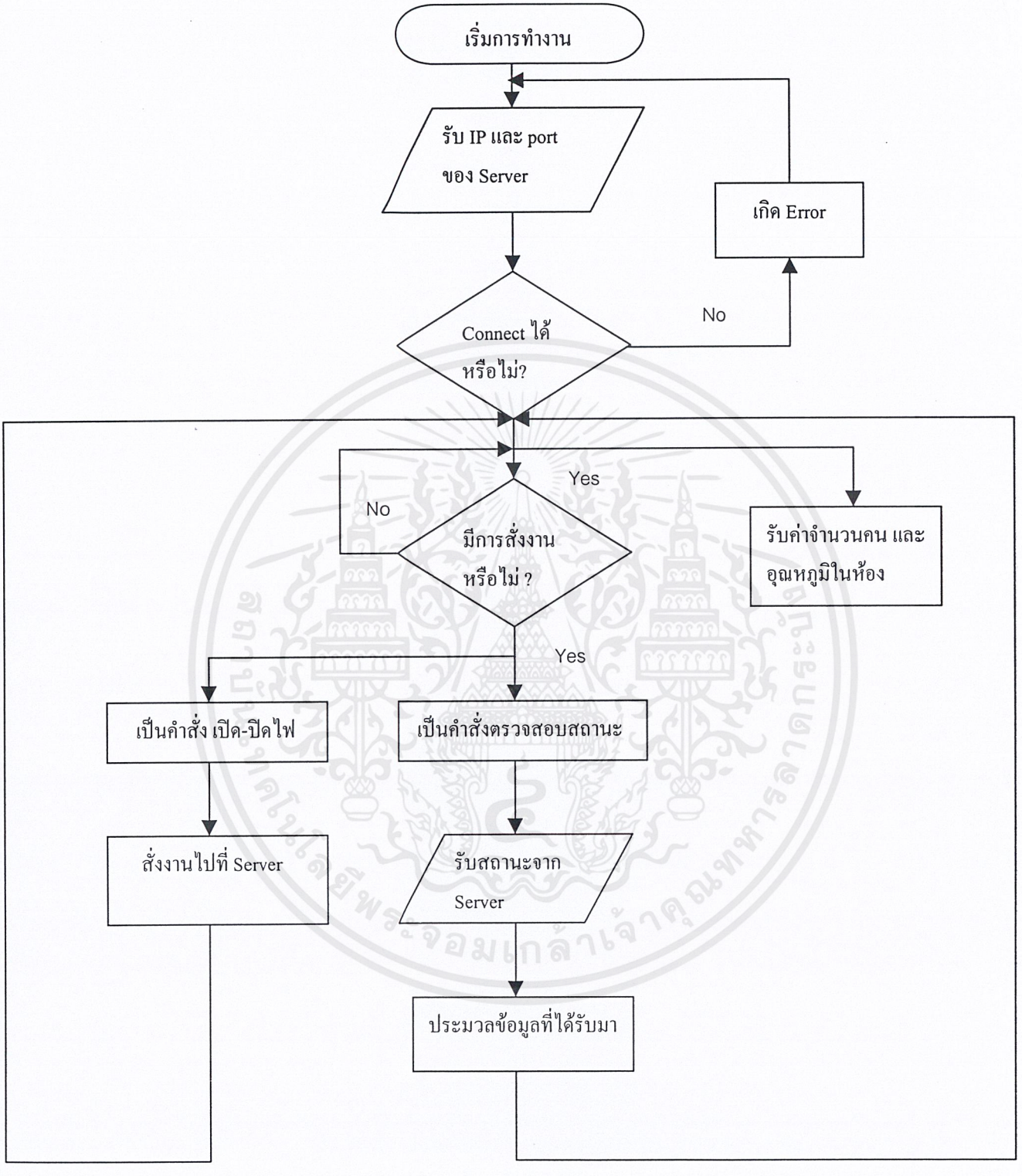
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.2 แสดง Flow Chart การทำงานของโปรแกรมของ Delphi



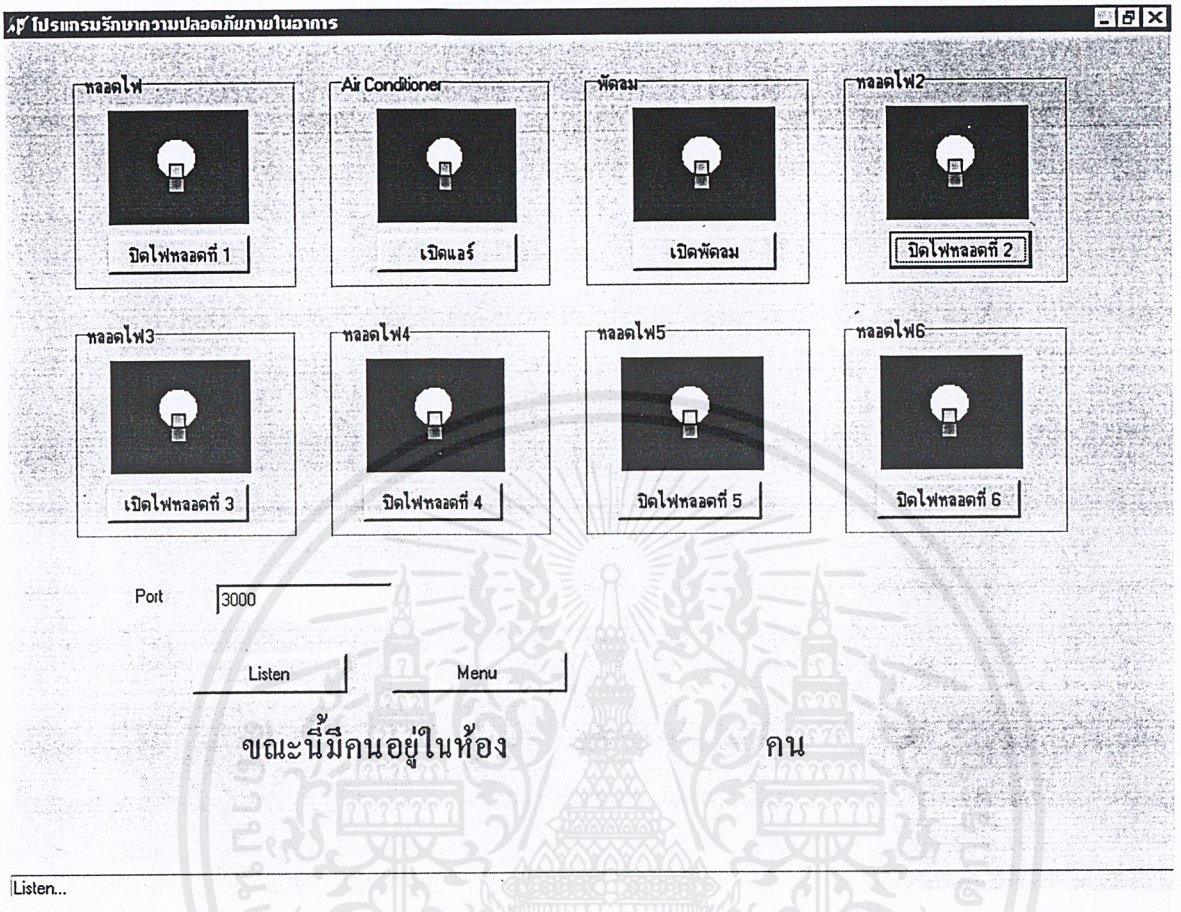
รูปที่5.2(ก) แสดง Flow Chart การทำงานของโปรแกรม Delphi ด้าน Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



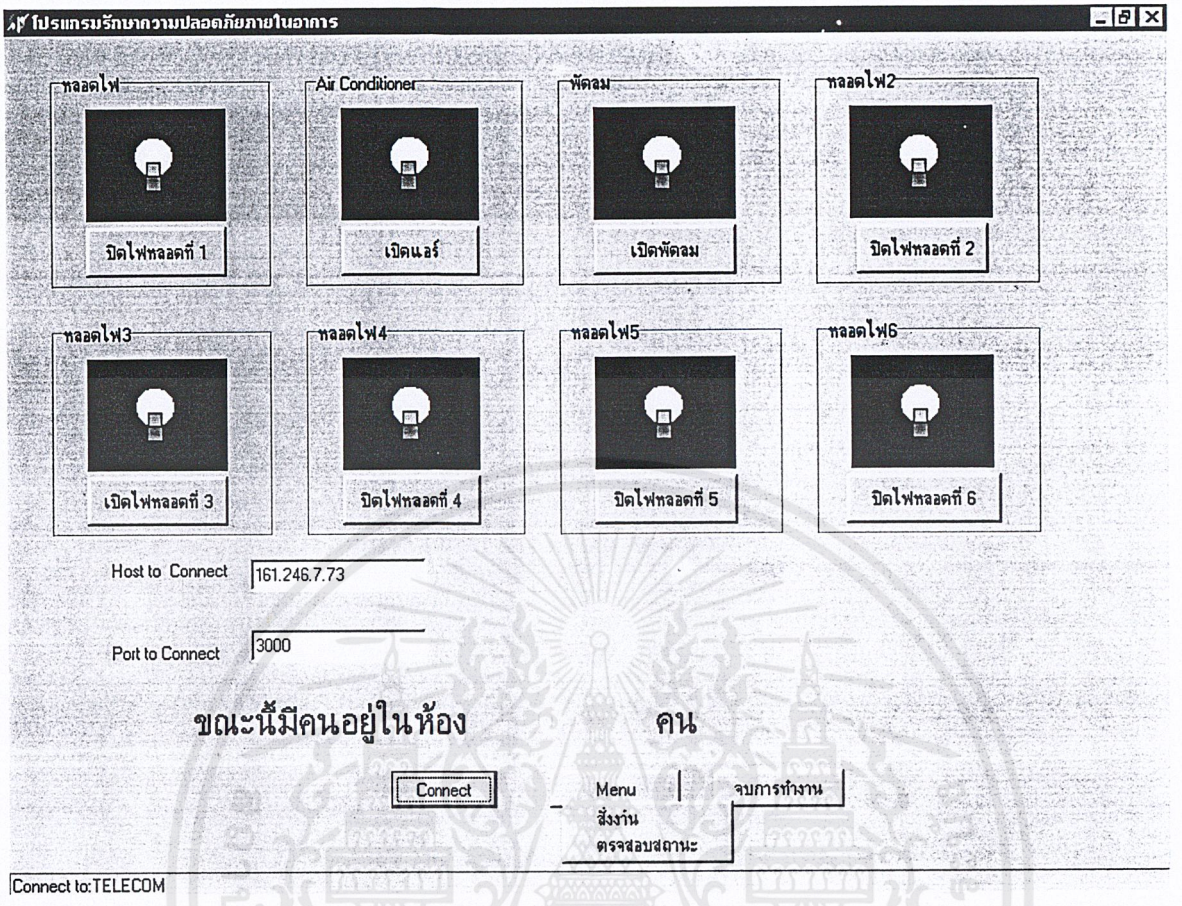
รูปที่5.2(ข) แสดง Flow Chart การทำงานของโปรแกรมของ Delphi ด้าน Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 รูปการแสดงหน้าจอของโปรแกรม Delphi ทางด้าน Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 รูปการแสดงหน้าจอของโปรแกรม Delphi ทางด้าน Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ผลการทดลอง

ในโครงงานนี้ จะมีการใช้หลอดไฟในการแสดงผลการทำงาน และมีการใช้ Sensor ในการควบคุม และแสดงค่า อุณหภูมิด้วยซึ่งผลที่ได้จากการทดลอง ได้แสดงไว้ในตารางการทดลองข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการ	ไฟหลอดที่1	แอร์	พัดลม	ไฟหลอดที่2	ไฟหลอดที่3	ไฟหลอดที่4	ไฟหลอดที่5	ไฟหลอดที่6	จำนวนคนในห้อง
สภาวะเริ่มต้น	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
สั่งเปิดไฟดวงที่1	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
สั่งเปิดair	ติด	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
สั่งปิดไฟดวงที่1	ดับ	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
สั่งเปิดอุปกรณ์ทั้งหมด	ติด	ติด	ติด	ติด	ติด	ติด	ติด	ติด	0 คน
สั่งปิดอุปกรณ์ทั้งหมด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
มีคนเข้ามาในห้อง 1 คน	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	1 คน
มีคนเข้ามาในห้อง 3 คน	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	3 คน
สั่งเปิดไฟดวงที่1 และ พัดลม	ติด	ดับ	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	3 คน
เมื่อคนออกจากห้องทั้งหมด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
เมื่อมีการสั่งเปิด air จาก Client	ดับ	ติด	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน
เมื่อมีการสั่งปิด air จาก Client	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	ดับ	0 คน

ตารางที่ 6.1 ตารางแสดงผลการทดลอง

บทที่ 7
สรุปและวิจารณ์

จากการทดลองกาทำงานของระบบ Sensor และการสั่งงานควบคุมอุปกรณ์ไฟฟ้าต่างๆนั้น พบว่า เราสามารถ กำหนดควบคุมการทำงานของอุปกรณ์แต่ละตัวได้ โดยการควบคุมผ่านเครื่องคอมพิวเตอร์ และสามารถแสดงสถานะของอุปกรณ์แต่ละตัวได้ว่า ขณะนี้มีอุปกรณ์ตัวใดบ้างที่มีการใช้งานอยู่บ้าง ซึ่งเราสามารถทำการควบคุมอุปกรณ์ได้ ถึงแม้ว่าจะไม่มีใครอยู่ภายในห้องนั้นเลยก็ตาม และเรายังสามารถควบคุมอุปกรณ์ต่างๆได้ จากที่อื่นๆ โดยการ connect มายังเครื่อง Server ซึ่งเราจะต้องทราบ IP address และ Port ที่เครื่อง Server ได้กำหนดขึ้นมาให้ถูกต้อง

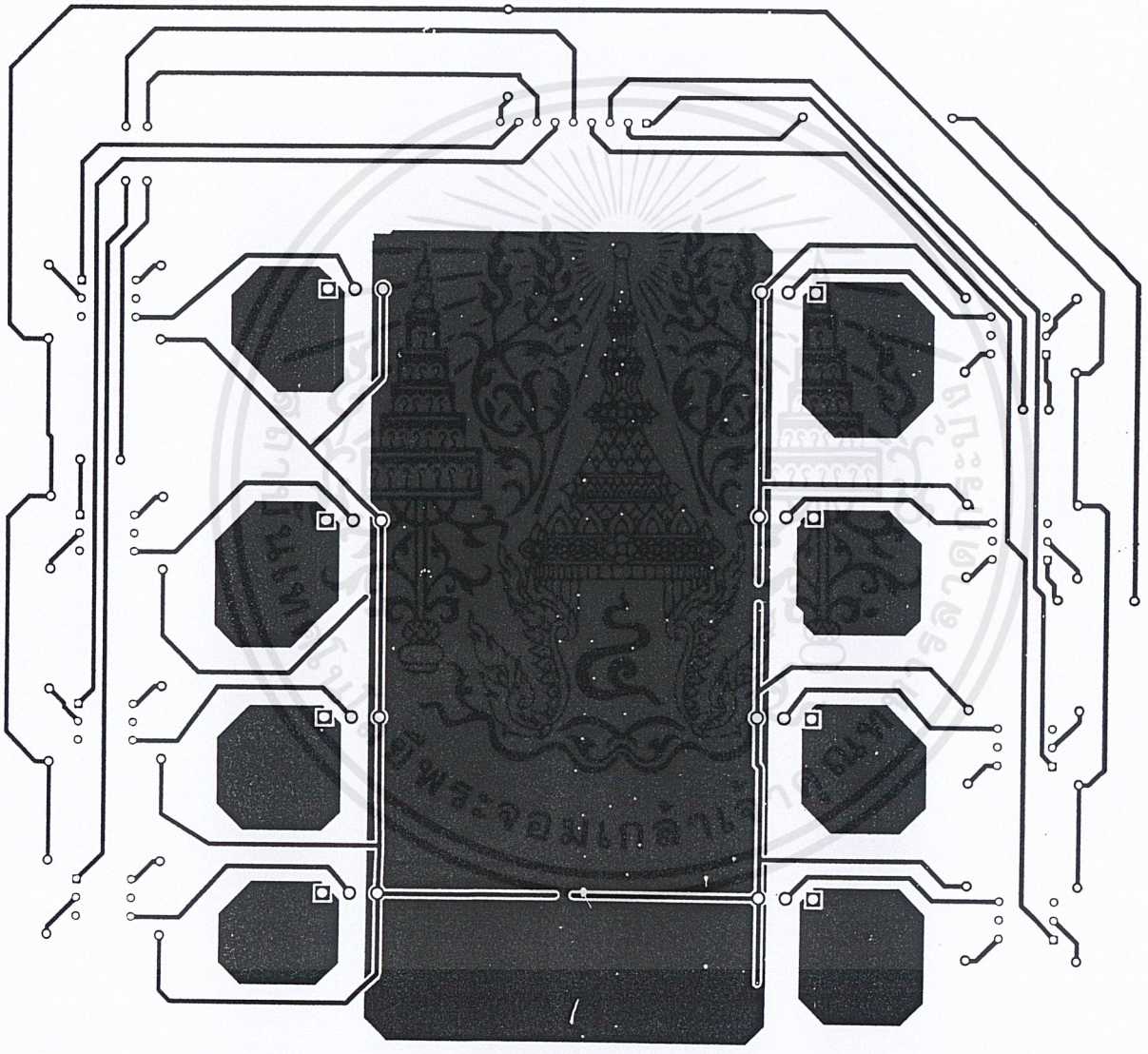
แต่ถ้าเกิดปัญหาคือ ไม่สามารถทำการสั่งงานผ่านคอมพิวเตอร์ได้ อาจมีสาเหตุมาจาก

1. อุปกรณ์ในส่วนของสวิตซ์เกิดการเสียหาย
2. ฟิวส์ขาด เนื่องจากใช้กระแสมากเกินไป
3. ตำแหน่งของตัว Sensor Infrared ติดตั้งไม่ถูกตำแหน่ง
4. port ที่ใช้เชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ กับบอร์ดคอนโทรลเลอร์ ต่อไม่แน่น
5. เป็น bug ที่หลงอยู่ภายในโปรแกรมที่นำมาใช้ควบคุม
6. ระบบ Internet ที่ใช้ในการติดต่อมีการเกิดข้อขัดข้องขึ้น
7. มีสัญญาณรบกวนเกิดขึ้นในสายส่งสัญญาณที่ใช้ติดต่อระหว่างเครื่องคอมพิวเตอร์ และ บอร์ดคอนโทรลเลอร์
8. Microcontroller เกิดการเสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

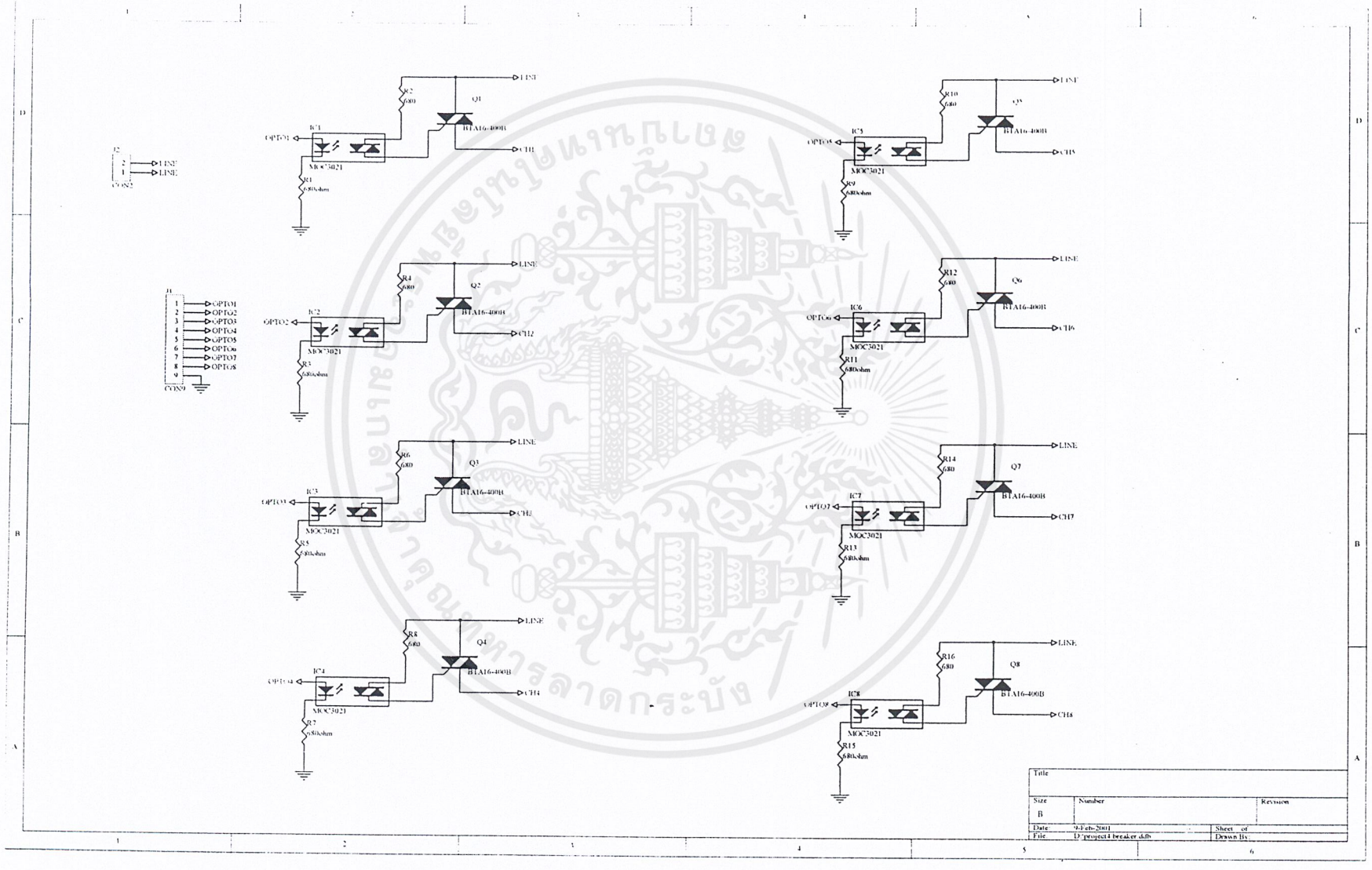
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

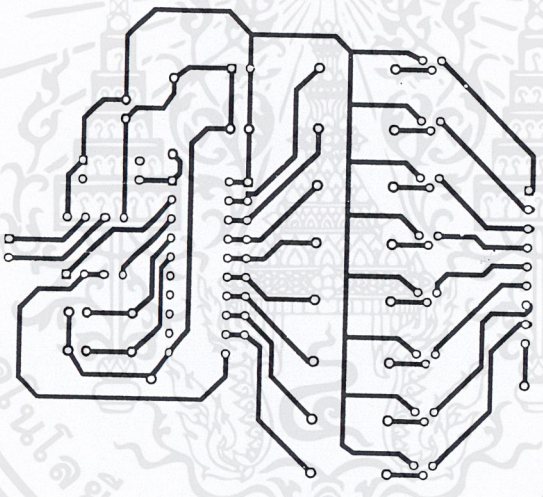


รูปภาพแสดงลายของวงจร Line Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

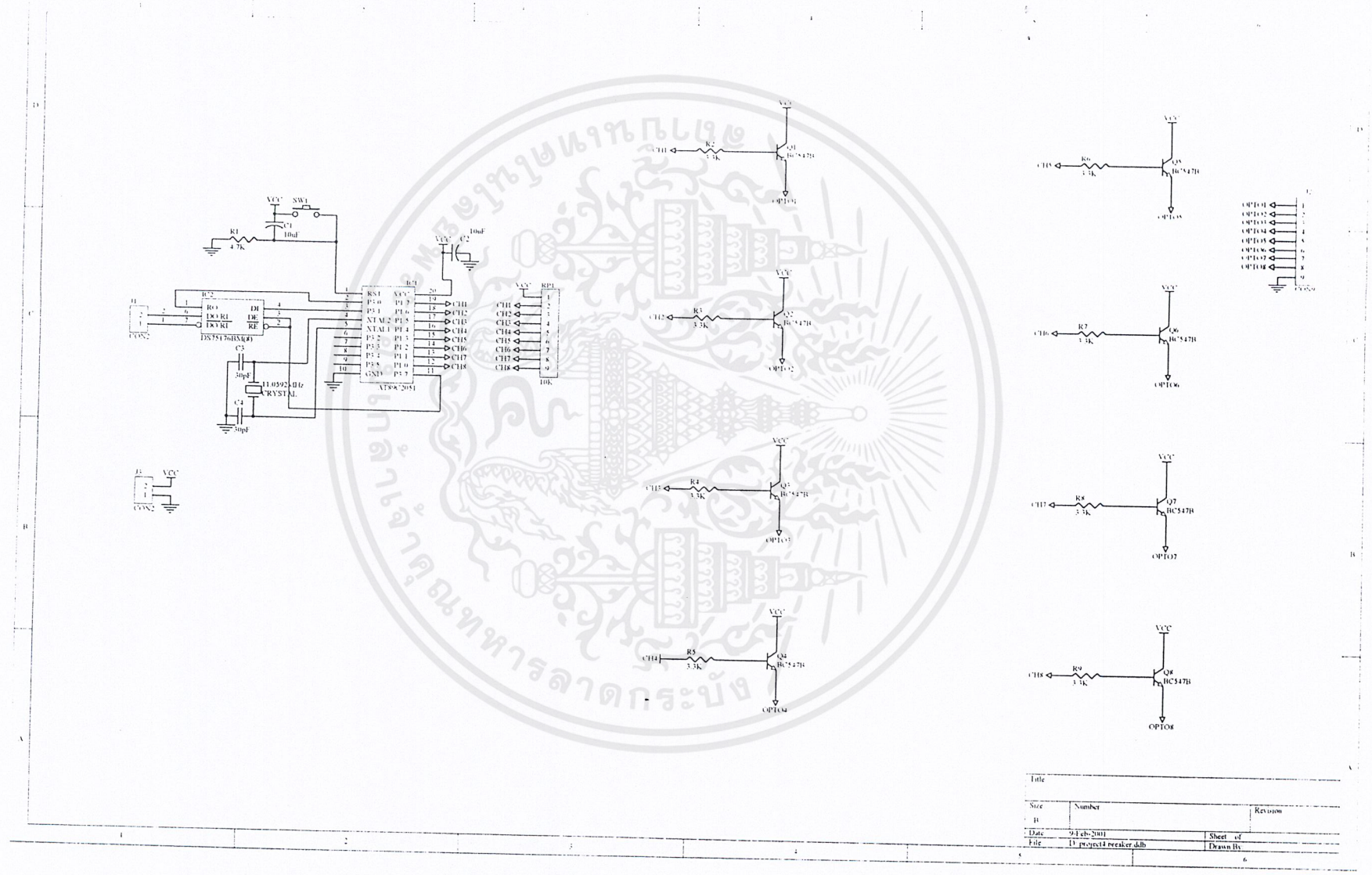


รูปภาพแสดงวงจร Line Switch



รูปภาพแสดงลายวงจรของวงจรไมโครคอนโทรลเลอร์ควบคุม Line Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

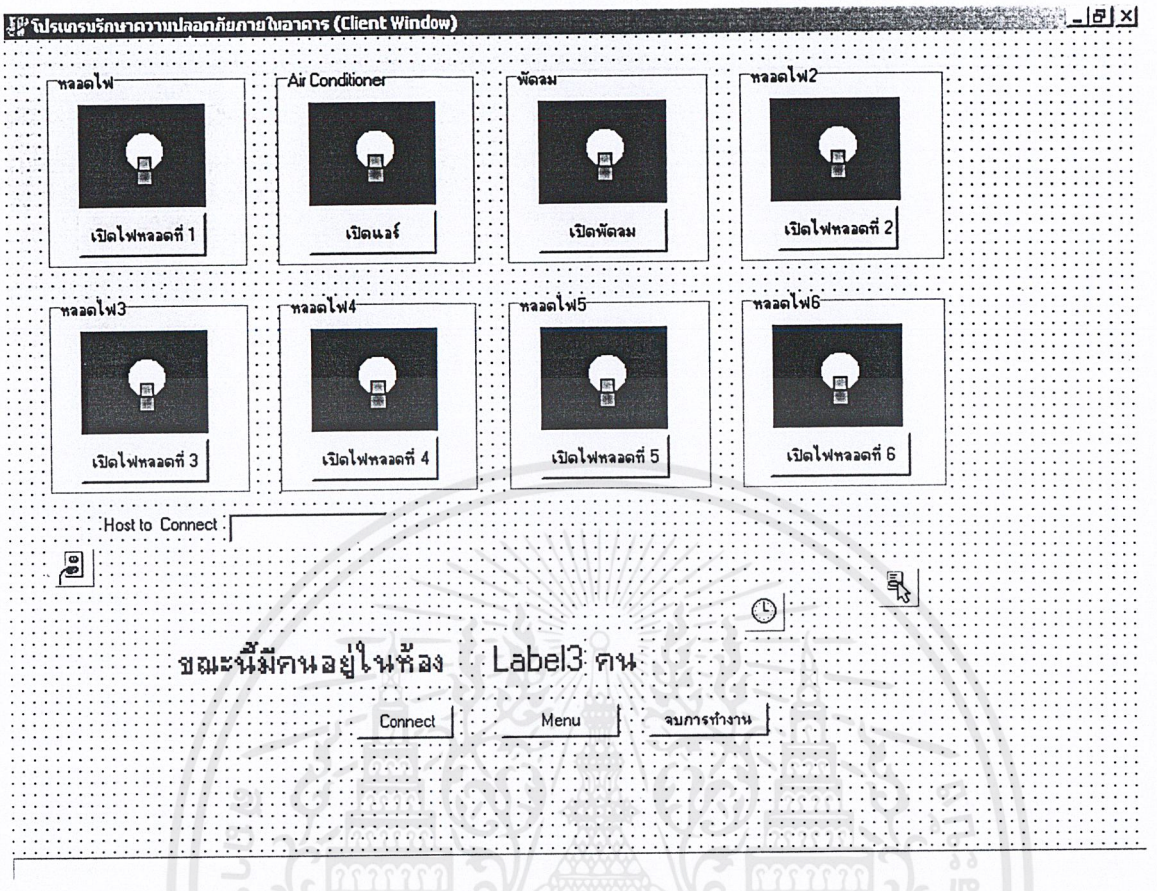


รูปภาพแสดงวงจรไมโครคอนโทรลเลอร์ควบคุม Line Switch

SOURCE CODE

ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการวาง Component ต่าง ๆ ในโปรแกรมDelphi ด้าน Client window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Delphi ฝั่ง Client

```
unit ProjectClient;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, ScktComp, ComCtrls, Menus;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    GroupBox5: TGroupBox;
    GroupBox6: TGroupBox;
    GroupBox7: TGroupBox;
    GroupBox8: TGroupBox;
    ImageLamp: TImage;
    ImageAir: TImage;
    ImageFan: TImage;
    ImageLamp2: TImage;
    ImageLamp3: TImage;
    ImageLamp4: TImage;
    ImageLamp5: TImage;
    Imagelamp6: TImage;
    ClientSocket: TClientSocket;
    Label1: TLabel;
    Edit1: TEdit;
    StatusBar1: TStatusBar;
    btnConnect: TButton;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    btnMenu: TButton;
    pmnMenuFrom: TPopupMenu;
    mnuCommand: TMenuItem;
    mnuCheck: TMenuItem;
    btnExit: TButton;
    btnLamp1: TButton;
    btnAir: TButton;
    btnFan: TButton;
    btnLamp2: TButton;
    btnLamp3: TButton;
    btnLamp4: TButton;
    btnLamp5: TButton;
    btnLamp6: TButton;
    Timer1: TTimer;
    procedure FormCreate(Sender: TObject);
    procedure ClientSocketConnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure btnConnectClick(Sender: TObject);
    procedure ClientSocketError(Sender: TObject; Socket:
TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
    procedure ClientSocketDisconnect(Sender: TObject;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Socket: TCustomWinSocket);
   procedure ClientSocketRead(Sender: TObject;
Socket: TCustomWinSocket);
   procedure mnuCommandClick(Sender:
TObject);
   procedure mnuCheckClick(Sender: TObject);
   procedure btnExitClick(Sender: TObject);
   procedure btnMenuClick(Sender: TObject);
   procedure btnLamp1Click(Sender: TObject);
   procedure btnAirClick(Sender: TObject);
   procedure btnFanClick(Sender: TObject);
   procedure btnLamp2Click(Sender: TObject);
   procedure btnLamp3Click(Sender: TObject);
   procedure btnLamp4Click(Sender: TObject);
   procedure btnLamp5Click(Sender: TObject);
   procedure btnLamp6Click(Sender: TObject);
   procedure Timer1Timer(Sender: TObject);

  private
   { Private declarations }
  public
   { Public declarations }
  end;

var
  Form1: TForm1;
  s1,s2,s3,s4,s5,s6,s7,s8 : string[8];
  count : integer;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
     s1:='a';
     s2:='b';
     s3:='c';
     s4:='d';
     s5:='e';
     s6:='f';
     s7:='g';
     s8:='h';
     count := 0;
     Label3.Caption := InttoStr(count);
     btnMenu.Enabled := False;
end;

procedure TForm1.ClientSocketConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  Statusbar1.Panels[0].Text:='Connect
to:'+socket.RemoteHost;
  btnMenu.Enabled := True;
  btnLamp1.Enabled := True;
  btnAir.Enabled := True;
  btnFan.Enabled := True;
  btnLamp2.Enabled := True;
  btnLamp3.Enabled := True;
  btnLamp4.Enabled := True;
  btnLamp5.Enabled := True;
  btnLamp6.Enabled := True;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

procedure TForm1.btnConnectClick(Sender:
TObject);
var hostname:shortstring;  portNo:integer;
begin
 if ClientSocket.Active=true then
 begin
 ClientSocket.Close;
 btnConnect.Caption:='Connect';
 Timer1.Enabled:=False;
 end
 else
 begin
 hostname:=edit1.text;
 portNo:=3000;
 ClientSocket.host:=hostname;
 ClientSocket.port:=PortNO;
 ClientSocket.open;
 btnConnect.Caption:='Disconnect';
 Timer1.Enabled:=True;
 end;
end;

procedure TForm1.ClientSocketError(Sender:
TObject;
 Socket: TCustomWinSocket; ErrorEvent:
TErrorEvent;
 var ErrorCode: Integer);
begin
 Statusbar1.Panels[0].Text:='Error';
end;

procedure TForm1.ClientSocketDisconnect(Sender:
TObject;
 Socket: TCustomWinSocket);
begin
 Statusbar1.Panels[0].Text:='Now Client Disconnect';
end;

procedure TForm1.ClientSocketRead(Sender: TObject;
 Socket: TCustomWinSocket);
var str,a2,a3:String;
   Counter,i,n : integer;
begin
 str:= str + Socket.ReceiveText;
 n := Length(str);
    for i := 1 to n-1 do
    begin
    if str[i]='O' then
    MessageDlg('à¡Ô´ Error'+#13#10+'Comport ที่
Server อาจจะไม่ได้ Connect',mtError,[mbYes],0)
    else if str[i]='A' then
    begin
    s1 := str[i];
    btnLamp1.Caption := 'ปิดไฟหลอดที่ 1';
    ImageLamp.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    end
    else if str[i]='a' then
    begin
    s1 := str[i];
    btnLamp1.Caption := 'เปิดไฟหลอดที่ 1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ImageLamp.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    end
    else if str[i]='B' then
    begin
    s2 := str[i];
    btnAir.Caption := 'ปิดแอร์';
    ImageAir.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    end
    else if str[i]='b' then
    begin
    s2 := str[i];
    btnAir.Caption := 'เปิดแอร์';
    ImageAir.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    end
    else if str[i]='C' then
    begin
    s3 := str[i];
    btnFan.Caption := 'ปิดพัดลม';
    ImageFan.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    end
    else if str[i]='c' then
    begin
    s3 := str[i];
    btnFan.Caption := 'เปิดพัดลม';
    ImageFan.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    end
    else if str[i]='D' then
    begin
    s4 := str[i];
    btnLamp2.Caption := 'ปิดไฟหลอดที่ 2';
    ImageLamp2.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    end
    else if str[i]='d' then
    begin
    s4 := str[i];
    btnLamp2.Caption := 'เปิดไฟหลอดที่ 2';
    ImageLamp2.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    end
    else if str[i]='E' then
    begin
    s5 := str[i];
    btnLamp3.Caption := 'ปิดไฟหลอดที่ 3';
    ImageLamp3.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    end
    else if str[i]='e' then
    begin
    s5 := str[i];
    btnLamp3.Caption := 'เปิดไฟหลอดที่ 3';
    ImageLamp3.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    end
    else if str[i]='F' then
    begin
    s6 := str[i];
    btnLamp4.Caption := 'ปิดไฟหลอดที่ 4';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      ImageLamp4.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
      end
      else if str[i]='f' then
      begin
      s6 := str[i];
      btnLamp4.Caption := 'เปิดไฟหลอดที่ 4';
      ImageLamp4.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
      end
      else if str[i]='G' then
      begin
      s7 := str[i];
      btnLamp5.Caption := 'ปิดไฟหลอดที่ 5';
      ImageLamp5.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
      end
      else if str[i]='g' then
      begin
      s7 := str[i];
      btnLamp5.Caption := 'เปิดไฟหลอดที่ 5';
      ImageLamp5.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
      end
      else if str[i]='H' then
      begin
      s8 := str[i];
      btnLamp6.Caption := 'ปิดไฟหลอดที่ 6';
      ImageLamp6.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
      end
      else if str[i]='h' then
      begin
      s8 := str[i];
      btnLamp6.Caption := 'เปิดไฟหลอดที่ 6';
      ImageLamp6.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
      end;
      if n=9 then
      begin
          a2:= str[9];
          a3 := IntToStr(Ord(a2[1]));
          if a3 > '0' then
          Label3.Caption:= a3
          else Label3.Caption:='0';
      end;
      end;

end;

procedure TForm1.mnuCommandClick(Sender: TObject);
begin
  ClientSocket.Socket.SendText(s1);
  ClientSocket.Socket.SendText(s2);
  ClientSocket.Socket.SendText(s3);
  ClientSocket.Socket.SendText(s4);
  ClientSocket.Socket.SendText(s5);
  ClientSocket.Socket.SendText(s6);
  ClientSocket.Socket.SendText(s7);
  ClientSocket.Socket.SendText(s8);
  ClientSocket.Socket.SendText('M');
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.mnuCheckClick(Sender:
TObject);
begin
   ClientSocket.Socket.SendText('K');
end;

procedure TForm1.btnExitClick(Sender: TObject);
var resp : word;
begin
     resp := MessageDlg('  Exit? '
,mtConfirmation,[mbYes,mbNo],0);
     if resp = idYes then
     begin
          Clientsocket.Close;
          Close;
     end
     else
          MessageDlg('กลับสู่การ
ทำงาน',mtWarning,[mbOK],0);
end;

procedure TForm1.btnMenuClick(Sender:
TObject);
begin
   MessageDlg('ต้องคลิ๊กขวาที่ปุ่ม Menu เพื่อเริ่ม
ทำงาน' ,mtInformation,[mbOK],0);
end;

procedure TForm1.btnLamp1Click(Sender:
TObject);
begin
     if s1 = 'a' then
     begin
     ImageLamp.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
     s1:='A';
     btnLamp1.Caption := 'ปิดไฟหลอดที่ 1';
     end
     else if s1 = 'A' then
     begin
     ImageLamp.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
     s1:='a';
     btnLamp1.Caption := 'เปิดไฟหลอดที่ 1';
     end;
     ClientSocket.Socket.SendText(s1);
end;

procedure TForm1.btnAirClick(Sender: TObject);
begin
     if s2 = 'b' then
     begin
     ImageAir.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
     s2:='B';
     btnAir.Caption := 'ปิดแอร์';
     end
     else if s2 = 'B' then
     begin
     ImageAir.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
     s2:='b';
     btnAir.Caption := 'เปิดแอร์';
```

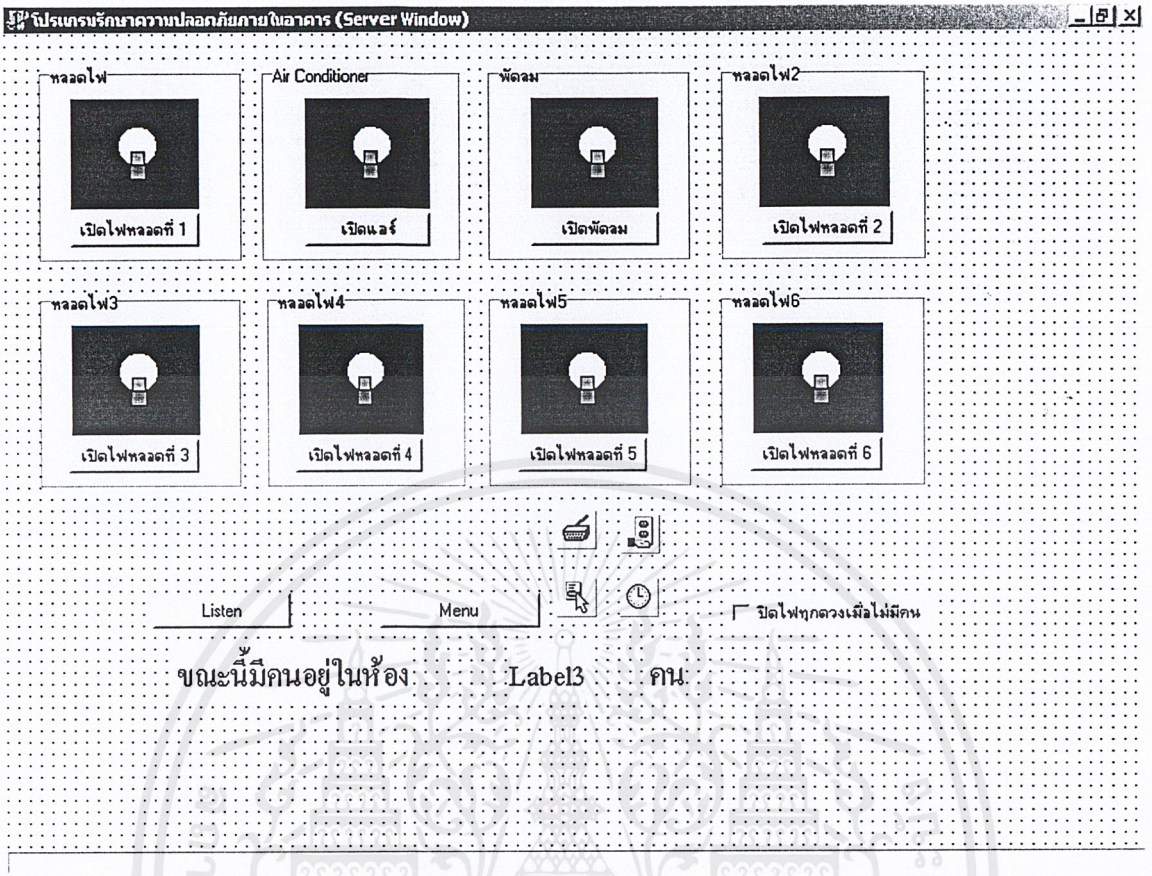
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;
    ClientSocket.Socket.SendText(s2);
end;

procedure TForm1.btnFanClick(Sender: TObject);
begin
    if s3 = 'c' then
    begin
    ImageFan.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s3:='C';
    btnFan.Caption := 'ปิดพัดลม';
    end
    else if s3 = 'C' then
    begin
    ImageFan.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s3:='c';
    btnFan.Caption := 'เปิดพัดลม';
    end;
    ClientSocket.Socket.SendText(s3);
end;

procedure TForm1.btnLamp2Click(Sender:
TObject);
begin
    if s4 = 'd' then
    begin
    ImageLamp2.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s4:='D';
    btnLamp2.Caption := 'ปิดไฟหลอดที่ 2';
    end
    else if s4 = 'D' then
    begin
    ImageLamp2.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s4:='d';
    btnLamp2.Caption := 'เปิดไฟหลอดที่ 2';
    end;
    ClientSocket.Socket.SendText(s4);
end;

procedure TForm1.btnLamp3Click(Sender: TObject);
begin
    if s5 = 'e' then
    begin
    ImageLamp3.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s5:='E';
    btnLamp3.Caption := 'ปิดไฟหลอดที่ 3';
    end
    else if s5 = 'E' then
    begin
    ImageLamp3.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s5:='e';
    btnLamp3.Caption := 'เปิดไฟหลอดที่ 3';
    end;
    ClientSocket.Socket.SendText(s5);
end;

procedure TForm1.btnLamp4Click(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if s6 = 'f' then
    begin
    ImageLamp4.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s6:='F';
    btnLamp4.Caption := 'ปิดไฟหลอดที่ 4';
    end
    else if s6 = 'F' then
    begin
    ImageLamp4.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s6:='f';
    btnLamp4.Caption := 'เปิดไฟหลอดที่ 4';
    end;
    ClientSocket.Socket.SendText(s6);
end;

procedure TForm1.btnLamp5Click(Sender:
TObject);
begin
    if s7 = 'g' then
    begin
    ImageLamp5.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s7:='G';
    btnLamp5.Caption := 'ปิดไฟหลอดที่ 5';
    end
    else if s7 = 'G' then
    begin
    ImageLamp5.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s7:='g';
    btnLamp5.Caption := 'เปิดไฟหลอดที่ 5';
    end;
    ClientSocket.Socket.SendText(s7);
end;

procedure TForm1.btnLamp6Click(Sender: TObject);
begin
    if s8 = 'h' then
    begin
    ImageLamp6.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s8:='H';
    btnLamp6.Caption := 'ปิดไฟหลอดที่ 6';
    end
    else if s8 = 'H' then
    begin
    ImageLamp6.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s8:='h';
    btnLamp6.Caption := 'เปิดไฟหลอดที่ 6';
    end;
    ClientSocket.Socket.SendText(s8);
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    ClientSocket.Socket.SendText('K');

end;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการวาง Component ต่าง ๆ ในโปรแกรมDelphi ด้าน Server window

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Delphi ฝั่ง Server

```
unit Mainpage;

interface

uses
  Windows, Messages, SysUtils, Classes,
Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls,
CPort,ComCtrls,OleCtrls, ScktComp, Menus;

type
  TForm1 = class(TForm)
    GroupBoxLamp: TGroupBox;
    GroupBoxAir: TGroupBox;
    GroupBoxFan: TGroupBox;
    ImageFan: TImage;
    ImageAir: TImage;
    ImageLamp: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    GroupBox1: TGroupBox;
    ImageLamp2: TImage;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    GroupBox5: TGroupBox;
    ImageLamp6: TImage;
    ImageLamp5: TImage;
    ImageLamp4: TImage;
    ImageLamp3: TImage;
    StatusBar1: TStatusBar;
    ServerSocket: TServerSocket;
    Label4: TLabel;
    btnListen: TButton;
    ComPort1: TComPort;
    btnMenu: TButton;
    pmnuMenu: TPopupMenu;
    mnuConnect: TMenuItem;
    N2: TMenuItem;
    mnuExit: TMenuItem;
    btnLamp1: TButton;
    btnAir: TButton;
    btnFan: TButton;
    btnLamp2: TButton;
    btnLamp3: TButton;
    btnLamp4: TButton;
    btnLamp5: TButton;
    btnLamp6: TButton;
    Timer: TTimer;
    cbZero: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure
  Comport1RxChar(Sender:TObject;Inque:integer);
    procedure ServerSocketAccept(Sender: TObject;
    Socket: TCustomWinSocket);
    procedure ServerSocketListen(Sender: TObject;
      Socket: TCustomWinSocket);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure ServerSocketClientError(Sender:
TObject;
      Socket: TCustomWinSocket; ErrorEvent:
TErrorEvent;
      var ErrorCode: Integer);
    procedure btnListenClick(Sender: TObject);
    procedure ServerSocketClientRead(Sender:
TObject;
      Socket: TCustomWinSocket);
    procedure mnuConnectClick(Sender: TObject);
    procedure mnuExitClick(Sender: TObject);
    procedure btnMenuClick(Sender: TObject);
    procedure senddata(addr,data: string);
    procedure btnLamp1Click(Sender: TObject);
    procedure btnAirClick(Sender: TObject);
    procedure btnFanClick(Sender: TObject);
    procedure btnLamp2Click(Sender: TObject);
    procedure btnLamp3Click(Sender: TObject);
    procedure btnLamp4Click(Sender: TObject);
    procedure btnLamp5Click(Sender: TObject);
    procedure btnLamp6Click(Sender: TObject);
    procedure TimerTimer(Sender: TObject);
    procedure ServerSocketConnecting
  (Sender:TObject;
    Socket: TCustomWinSocket);

  private
    { Private declarations }
    ConnectSocket:TcustomWinSocket;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  s1,s2,s3,s4,s5,s6,s7,s8:string;
  x1,x2,x3,x4,x5,x6,x7,x8 : string;
  N,M,lenght,count: Integer;
  data,temp,check,addr:string;
  Error:boolean;

implementation

{$R *.DFM}

Procedure TForm1.FormCreate(Sender: TObject);
begin
    x1:='a';
    x2:='b';
    x3:='c';
    x4:='d';
    x5:='e';
    x6:='f';
    x7:='g';
    x8:='h';
    s1:='a';
    s2:='b';
    s3:='c';
    s4:='d';
    s5:='e';
    s6:='f';
    s7:='g';
    s8:='h';
    N := 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    count:=0;
    label3.Caption:= inttostr(count);
end;

procedure TForm1.Comport1RxChar
(Sender:TObject;Inque:Integer);
var Str,Str1,Str2,Str3:string;
begin
    if ComPort1.Parity.Bits=prSpace then
    ComPort1.ReadStr(Str,Inque);
    if Str='' then exit;
    Str2:=Str[2];
    Str3:=Str[3];
    if Str2='N' then
        begin
        Str1:=IntToStr(Ord(Str[3]));
        Label3.Caption:=Str1;
        Count:=StrToInt(Str1);
        end;
    if cbZero.Checked=true then
    begin
    if Count = 0 then
      begin
      x1:='A';    btnLamp1.Click;
      x2:='B';    btnAir.Click;
      x3:='C';    btnFan.Click;
      x4:='D';    btnLamp2.Click;
      x5:='E';    btnLamp3.Click;
      x6:='F';    btnLamp4.Click;
      x7:='G';    btnLamp5.Click;
      x8:='H';    btnLamp6.Click;
      end;
    end;
end;

procedure TForm1.ServerSocketAccept(Sender: TObject;
 Socket: TCustomWinSocket);
begin
 ConnectSocket:=Socket;
 statusbar1.Panels[0].text:='connection accepted';
end;

procedure TForm1.ServerSocketConnecting
(Sender:TObject;
  Socket: TCustomWinSocket);
begin
 Statusbar1.Panels[0].text:='Connecting.....';
end;

procedure TForm1.ServerSocketClientError(Sender:
TObject;
 Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
 var ErrorCode: Integer);
begin
 Statusbar1.Panels[0].Text:='Error';
end;

procedure TForm1.ServerSocketListen(Sender: TObject;
  Socket: TCustomWinSocket);
begin
   statusbar1.panels[0].text:='Listen...';
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.btnListenClick(Sender:
TObject);
var portNo:integer;
begin
      portNO:=3000;
      ServerSocket.port:=PortNo;
      ServerSocket.open;
end;

procedure TForm1.ServerSocketClientRead
(Sender: TObject;
 Socket: TCustomWinSocket);
 var  A , s9 :String;
      i,n : integer;

begin
 A:= A + Socket.Receivetext;
 n :=  Length(A);
 s9:=chr(Count);
 //s9:= InttoStr(count);
 for i := 0 to n do
 if A[i] = 'K' then
 begin
 //if Comport1.Connected then
 //begin
 ServerSocket.Socket.Connections[0].SendText
(x1);
 ServerSocket.Socket.Connections[0].SendText
(x2);
 ServerSocket.Socket.Connections[0].SendText
(x3);
 ServerSocket.Socket.Connections[0].SendText(x4);
 ServerSocket.Socket.Connections[0].SendText(x5);
 ServerSocket.Socket.Connections[0].SendText(x6);
 ServerSocket.Socket.Connections[0].SendText(x7);
 ServerSocket.Socket.Connections[0].SendText(x8);
 ServerSocket.Socket.Connections[0].SendText(s9);
 end
 //else ServerSocket.Socket.Connections[0].SendText
('O');
 //end;

 else if A[i] = 'A' then
 begin
 s1 := 'a';
 btnLamp1.Click;
 end
 else if A[i] = 'a'  then
 begin
 s1 := 'A';
 btnLamp1.Click;
 end
 else if A[i] = 'B' then
 begin
 s2 := 'b';
 btnAir.Click;
 end
 else if A[i] = 'b' then
 begin
 s2 := 'B';
 btnAir.Click;
 end
 else if A[i] = 'C' then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
s3 := 'c';
btnFan.Click;
end
else if A[i] = 'c' then
begin
s3 := 'C';
btnLamp1.Click;
end
else if A[i] = 'D' then
begin
s4 := 'd';
btnLamp2.Click;
end
else if A[i] = 'd' then
begin
s4 := 'D';
btnLamp2.Click;
end
else if A[i] = 'E' then
begin
s5 := 'e';
btnLamp3.Click;
end
else if A[i] = 'e' then
begin
s5 := 'E';
btnLamp3.Click;
end
else if A[i] = 'F' then
begin
s6 := 'f';
btnLamp4.Click;
end
else if A[i] = 'f' then
begin
s6 := 'F';
btnLamp4.Click;
end
else if A[i] = 'G' then
begin
s7 := 'g';
btnLamp5.Click;
end
else if A[i] = 'g' then
begin
s7 := 'G';
btnLamp5.Click;
end
else if A[i] = 'H' then
begin
s8 := 'h';
btnLamp6.Click;
end
else if A[i] = 'h' then
begin
s8 := 'H';
btnLamp6.Click;
end;

end;

procedure TForm1.mnuConnectClick(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if Comport1.Connected then
  begin
    Comport1.Close;
    mnuConnect.Checked := False;
    btnLamp1.Enabled := False;
    btnAir.Enabled := False;
    btnFan.Enabled := False;
    btnLamp2.Enabled := False;
    btnLamp3.Enabled := False;
    btnLamp4.Enabled := False;
    btnLamp5.Enabled := False;
    btnLamp6.Enabled := False;
    Timer.Enabled := False;
  end
  else
  begin
    Comport1.Open;

ComPort1.FlowControl.ControlRTS:=rtsEnable;
    mnuConnect.Checked := True;
    btnLamp1.Enabled := True;
    btnAir.Enabled := True;
    btnFan.Enabled := True;
    btnLamp2.Enabled := True;
    btnLamp3.Enabled := True;
    btnLamp4.Enabled := True;
    btnLamp5.Enabled := True;
    btnLamp6.Enabled := True;
    Timer.Enabled := True;
  end;
end;
```

```
procedure TForm1.mnuExitClick(Sender: TObject);
var resp : Word;
begin
    resp := MessageDlg(' Exit? ' ,mtConfirmation,
[mbYes,mbNo],0);
    if resp = idYes then
    begin
        Comport1.Close;
        ServerSocket.Close;
        Close;
    end
    else
        MessageDlg('กลับสู่การทำงาน',mtWarning,
[mbOK],0);
end;


procedure TForm1.btnMenuClick(Sender: TObject);
begin
  MessageDlg('คุณต้องคลิ๊กขวาที่ปุ่ม Menu เพื่อเริ่ม
ทำงาน' ,mtInformation,[mbOK],0);
end;

procedure TForm1.senddata(addr,data: string);
begin
    if ComPort1.Connected=true then
    begin
   ComPort1.FlowControl.ControlRTS:=rtsDisable;
    ComPort1.BeginUpdate;
    ComPort1.Parity.Bits:=prMark;
    ComPort1.EndUpdate;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ComPort1.WriteStr(addr);
    ComPort1.BeginUpdate;
    ComPort1.Parity.Bits:=prSpace;
    ComPort1.EndUpdate;
    ComPort1.WriteStr(Data);

ComPort1.FlowControl.ControlRTS:=rtsEnable;
    end;
end;
procedure TForm1.btnLamp1Click(Sender:
TObject);
begin
    addr := '1';
    if x1 = 'a' then
    begin
 ImageLamp.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s1:='A';
    temp := s1;
    btnLamp1.Caption := 'ปิดไฟหลอดที่ 1';
    end
    else if x1 = 'A' then
    begin
    ImageLamp.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s1:='a';
    temp := s1;
    btnLamp1.Caption := 'เปิดไฟหลอดที่ 1';
    end;
    x1 := s1;
    Data:=x1;
    senddata(addr,data); Senddata(Addr,Data);
end;

procedure TForm1.btnAirClick(Sender: TObject);
begin
    addr := '1';
    if x2 = 'b' then
    begin
    ImageAir.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s2:='B';
    temp := s2;
    btnAir.Caption := 'ปิดแอร์';
    end
    else if x2 = 'B' then
    begin
    ImageAir.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s2:='b';
    temp := s2;
    btnAir.Caption := 'เปิดแอร์';
    end;
    x2 := s2;
    Data:=x2;
    Senddata(Addr,Data); Senddata(Addr,Data);
end;

procedure TForm1.btnFanClick(Sender: TObject);
begin
    addr := '1';
    if x3 = 'c' then
    begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ImageFan.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s3:='C';
    temp := s3;
    btnFan.Caption := 'ปิดพัดลม';
    end
    else if x3 = 'C' then
    begin
    ImageFan.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s3:='c';
    temp := s3;
    btnFan.Caption := 'เปิดพัดลม';
    end;
    x3 := s3;
    Data:=x3;
    Senddata(Addr,Data); Senddata(Addr,Data);
end;

procedure TForm1.btnLamp2Click(Sender:
TObject);
begin
    addr := '1';
    if x4 = 'd' then
    begin
    ImageLamp2.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s4:='D';
    temp := s4;
    btnLamp2.Caption := 'ปิดไฟหลอดที่ 2';
    end
    else if x4 = 'D' then
    begin
    ImageLamp2.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s4:='d';
    temp := s4;
    btnLamp2.Caption := 'เปิดไฟหลอดที่ 2';
    end;
    x4 := s4;
    Data:=x4;
    Senddata(Addr,Data); Senddata(Addr,Data);
end;

procedure TForm1.btnLamp3Click(Sender: TObject);
begin
    addr := '1';
    if x5 = 'e' then
    begin
    ImageLamp3.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s5:='E';
    temp := s5;
    btnLamp3.Caption := 'ปิดไฟหลอดที่ 3';
    end
    else if x5 = 'E' then
    begin
    ImageLamp3.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s5:='e';
    temp := s5;
    btnLamp3.Caption := 'เปิดไฟหลอดที่ 3';
    end;
    x5 := s5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Data:=x5;
    Senddata(Addr,Data); Senddata(Addr,Data);
end;
procedure TForm1.btnLamp4Click(Sender:
TObject);
begin
    addr := '1';
    if x6 = 'f' then
    begin
    ImageLamp4.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s6:='F';
    temp := s6;
    btnLamp4.Caption := 'ปิดไฟหลอดที่ 4';
    end
    else if x6 = 'F' then
    begin
    ImageLamp4.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s6:='f';
    temp := s6;
    btnLamp4.Caption := 'เปิดไฟหลอดที่ 4';
    end;
    x6 := s6;
    Data:=x6;
    Senddata(Addr,Data); Senddata(Addr,Data);
end;

procedure TForm1.btnLamp5Click(Sender:
TObject);
begin
    addr := '1';
    if x7 = 'g' then
    begin
    ImageLamp5.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s7:='G';
    temp := s7;
    btnLamp5.Caption := 'ปิดไฟหลอดที่ 5';
    end
    else if x7 = 'G' then
    begin
    ImageLamp5.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
    s7:='g';
    temp := s7;
    btnLamp5.Caption := 'เปิดไฟหลอดที่ 5';
    end;
    x7 := s7;
    Data:=x7;
    Senddata(Addr,Data); Senddata(Addr,Data);
end;

procedure TForm1.btnLamp6Click(Sender: TObject);
begin
    addr := '1';
    if x8 = 'h' then
    begin
    ImageLamp6.Picture.Bitmap.LoadFromFile
('LampOn.bmp');
    s8:='H';
    temp := s8;
    btnLamp6.Caption := 'ปิดไฟหลอดที่ 6';
    end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      else if x8 = 'H' then
      begin
      ImageLamp6.Picture.Bitmap.LoadFromFile
('LampOff.bmp');
      s8:='h';
      temp := s8;
      btnLamp6.Caption := 'เปิดไฟหลอดที่ 6';
      end;
      x8 := s8;
      Data:=x8;
      Senddata(Addr,Data); Senddata(Addr,Data);
end;

procedure TForm1.TimerTimer(Sender: TObject);
begin
      if not ComPort1.Connected then
        MessageDlg('Please Click Connect
Button',mtError,[mbOK],0)
      else
        begin
      addr:='2';
      data:='N';

ComPort1.FlowControl.ControlRTS:=rtsDisable;
      ComPort1.BeginUpdate;
      ComPort1.Parity.Bits:=prMark;
      ComPort1.EndUpdate;
      ComPort1.WriteStr(addr);
      ComPort1.BeginUpdate;
      ComPort1.Parity.Bits:=prSpace;
      ComPort1.EndUpdate;
      ComPort1.WriteStr(data);
ComPort1.FlowControl.ControlRTS:=rtsEnable;
  end;
      if cbZero.Checked=true then
begin
if Count = 0 then
  begin
  x1:='A';    btnLamp1.Click;
  x2:='B';    btnAir.Click;
  x3:='C';    btnFan.Click;
  x4:='D';    btnLamp2.Click;
  x5:='E';    btnLamp3.Click;
  x6:='F';    btnLamp4.Click;
  x7:='G';    btnLamp5.Click;
  x8:='H';    btnLamp6.Click;
  end;
end;
                    end;

                    end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมไมโครคอนโทรลเลอร์

```
;PROGRAM FOR DETECT IN/OUT
ADDR    EQU 25H
DATA    EQU 26H
BUF     EQU 27H
BFF     EQU 28H
NUM     EQU 29H

SEN1    BIT P1.7
SEN2    BIT P1.6
LED1    BIT P1.5
LED2    BIT P1.4

        ORG 0000H
        MOV ADDR,#'2'
        ;ADDRESS IS '2'
        SETB SEN1
        SETB SEN2
    MOV NUM,#00H

INITIAL:
        MOV     SCON,#0F0H              ;Set
serial mode3 multiprocessor mode
        MOV     PCON,#80H
        ;SMOD=1
        MOV     TMOD,#20H
        ;Timer1 mode2
        MOV     TH1,#0FFH
        ;BAUD 57.6 K
        SETB    TR1                     ;Start
Timer1
    CLR   P3.7          ;Set chip 75176 for recieve

RST:  CLR LED1
    CLR LED2
    MOV BUF,#01111000b
    MOV BFF,#01111000b
    MOV 40H,#00H
    MOV 41H,#00H
MAIN:  CALL D1
    CALL D2
    JMP CK
LOOP:  JMP READ_A
NEXT:  CJNE A,#'N',MAIN
    CALL SEND_A
    JMP MAIN
CK:   MOV A,40H
    CJNE A,#00H,CK5
    JMP END_C
CK5:  CJNE A,#05H,CK3
    JMP CK1
CK3:  CJNE A,#03H,END_C
CK1:  MOV A,41H
    CJNE A,#00H,CK4
    JMP MAIN
CK4:  CJNE A,40H,CK2
    JMP END_C
CK2:  MOV A,40H
    CJNE A,#05H,ADD_N
    MOV A,NUM
    CJNE A,#00H,DEC_N
    SJMP END_C
DEC_N:  DEC NUM
    SETB LED1
    CLR LED2
    SJMP END_C
ADD_N:  INC NUM
    SETB LED2
    CLR LED1
END_C:  MOV 40H,#00H
    MOV 41H,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    JMP LOOP


READ_A: JNB RI,MAIN
        MOV A,SBUF
        CLR RI
        CJNE A,ADDR,END_R
        CLR SM2
        JNB RI,$
        MOV A,SBUF
        SETB SM2
END_R:  JMP NEXT

;------------------
;  SEND DATA
;------------------
SEND_A: CLR TI
        SETB P3.7
        SETB TB8
        MOV A,#'N'
        MOV SBUF,A
        JNB TI,$
        CLR TI
        CLR TB8
        MOV A,NUM
        MOV SBUF,A
        JNB TI,$
        CLR TI
        CLR P3.7
        RET

;----------------------------
;   CHECK SENSOR1
;----------------------------
D1:     JNB SEN1,SET_BF1    ;CHECK SEN1
        CLR BUF.1                   ;IF P1.7 ON
BUF.1=0
        SJMP CK_B5

SET_BF1: SETB BUF.1                 ;IF P1.7 OFF
BUF.1=1
CK_B5:  JNB BUF.5,ORB1
ANDB1:  MOV C,BUF.3
        ANL C,BUF.1                 ;BUF.3 AND
BUF.1
        MOV BUF.3,C
        JNB BUF.3,CLR_B5
        SETB BUF.3
        SETB BUF.5
        SETB BUF.1
        SJMP END_D1
CLR_B5: CLR BUF.5           ;SEN1 ON CLEAR BUF.5
        SJMP END_D1
ORB1:   MOV C,BUF.3
        ORL C,BUF.1                 ;BUF.3 OR
BUF.1
        MOV BUF.3,C
        JB  BUF.3,SET_B5
        SJMP END_D1
SET_B5: SETB BUF.5                  ;IF PIN1 OFF
THEN SET BUF.5
        SETB BUF.3
        SETB BUF.1
    MOV A,40H
    CJNE A,#00H,CK1_1
    MOV 40H,#05H
    SJMP END_D1
CK1_1:  CJNE A,#05H,CK1_2
    SJMP END_D1
CK1_2:  MOV 41H,#05H
END_D1: RET

;----------------------------
;   CHECK SENSOR2
;----------------------------
D2:   JNB SEN2,SET_BF2    ;CHECK PIN2
    CLR BFF.2         ;IF SEN2 ON  BUF.2=0
        SJMP CK_B6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SET_BF2: SETB BFF.2          ;IF SEN2 OFF BUF.2=1
CK_B6:  JNB BFF.6,ORB2
ANDB2:  MOV C,BFF.4
     ANL C,BFF.2          ;BUF.4 AND BUF.2
     MOV BFF.4,C
     JNB BFF.4,CLR_B6
          SJMP END_D2
CLR_B6: CLR BFF.6          ;SEN2 ON CLEAR BUF.6
          SJMP END_D2
ORB2:   MOV C,BFF.4          ;BUF.4 OR BUF.2
     ORL C,BFF.2
     MOV BFF.4,C
     JB  BFF.4,SET_B6
          SJMP END_D2
SET_B6: SETB BFF.6          ;IF P1.4 OFF THEN SET
BUF.6
          SETB BFF.4
          SETB BFF.2
     MOV A,40H
     CJNE A,#00H,CK2_1
     MOV 40H,#03H
     SJMP END_D2
CK2_1:  CJNE A,#03H,CK2_2
     SJMP END_D2
CK2_2:  MOV 41H,#03H
END_D2: RET
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;PROGRAM CONTROL SWITCH 8
CHANNELS
;BY SUPACHAI CHANARITTICHAI
;ELECTRONICS ENGINEERING
ADDR    EQU 25H
DATA    EQU 26H
LIGHT_A BIT P1.7
LIGHT_B BIT P1.6
LIGHT_C BIT P1.5
LIGHT_D BIT P1.4
LIGHT_E BIT P1.3
LIGHT_F BIT P1.2
LIGHT_G BIT P1.1
LIGHT_H BIT P1.0

        ORG 0000H
        MOV ADDR,#'1'
        ;ADDRESS IS '1'
        MOV P1,#00H
INITIAL:
        MOV     SCON,#0F0H      ;Set
serial mode3 multiprocessor mode
        MOV     PCON,#80H
        ;SMOD=1
        MOV     TMOD,#20H
        ;Timer1 mode2
        MOV     TH1,#0FFH
        ;BAUD 57.6 K
        SETB    TR1             ;Start
Timer1
        CLR     P3.7            ;Set
chip 75176 for resive

RST:    MOV P1,#00H
        CLR P3.7

MAIN:   CALL READ_A
        MOV DATA,A
;-------------------------
;           TABLE
;-------------------------
CHECKR: CJNE A,#'R',CK1
        JMP RST                 ;IF 'R' = RESET
CK1:    CJNE A,#'A',CK2         ;IF 'A' OPEN
LIGHT A
        SETB LIGHT_A
        JMP MAIN
CK2:    CJNE A,#'B',CK3         ;IF 'B' OPEN
LIGHT B
        SETB LIGHT_B
        JMP MAIN
CK3:    CJNE A,#'C',CK4         ;IF 'C' OPEN
LIGHT C
        SETB LIGHT_C
        JMP MAIN
CK4:    CJNE A,#'D',CK5         ;IF 'D' OPEN
LIGHT D
        SETB LIGHT_D
        JMP MAIN
CK5:    CJNE A,#'E',CK6         ;IF 'E' OPEN
LIGHT E
        SETB LIGHT_E
        JMP MAIN
CK6:    CJNE A,#'F',CK7         ;IF 'F' OPEN
LIGHT F
        SETB LIGHT_F
        JMP MAIN
CK7:    CJNE A,#'G',CK8         ;IF 'G' OPEN
LIGHT G
        SETB LIGHT_G
        JMP MAIN
CK8:    CJNE A,#'H',CK9         ;IF 'H' OPEN
LIGHT H
        SETB LIGHT_H
        JMP MAIN
CK9:    CJNE A,#'a',CK10    ;IF 'a' CLOSE LIGHT A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR LIGHT_A
        JMP MAIN
CK10:   CJNE A,#'b',CK11    ;IF 'b' CLOSE LIGHT B
        CLR LIGHT_B
        JMP MAIN
CK11:   CJNE A,#'c',CK12    ;IF 'c' CLOSE LIGHT C
        CLR LIGHT_C
        JMP MAIN
CK12:   CJNE A,#'d',CK13    ;IF 'd' CLOSE LIGHT D
        CLR LIGHT_D
        JMP MAIN
CK13:   CJNE A,#'e',CK14    ;IF 'e' CLOSE LIGHT E
        CLR LIGHT_E
        JMP MAIN
CK14:   CJNE A,#'f',CK15    ;IF 'f' CLOSE LIGHT F
        CLR LIGHT_F
        JMP MAIN
CK15:   CJNE A,#'g',CK16    ;IF 'g' CLOSE LIGHT G
        CLR LIGHT_G
        JMP MAIN
CK16:   CJNE A,#'h',MAIN    ;IF 'h' CLOSE LIGHT H
        CLR LIGHT_H
        JMP MAIN                    ;ELSE GOTO
MAIN

;------------------
;   READ DATA
;------------------
READ_A: JNB RI,$
        MOV A,SBUF
        CLR RI
        CJNE A,ADDR,END_R
        CLR SM2
        JNB RI,$
        MOV A,SBUF
        SETB SM2
        MOV DATA,A
END_R:  RET

;------------------
;   SEND DATA
;------------------
SEND_A: CLR TI
        SETB P3.7
        SETB TB8
        MOV A,#'L'
        MOV SBUF,A
        JNB TI,$
        CLR TI
        CLR TB8
        MOV A,DATA
        MOV SBUF,A
        JNB TI,$
        CLR TI
        CLR P3.7
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GlobalOptoisolator™

VDE | UL | CSA | SETI | SEMKO | DEMKO | NEMKO | BABT

6-Pin DIP Zero-Cross Optoisolators Triac Driver Output
(250 Volts Peak)

MOC3031
[IFT = 15 mA Max]
MOC3032
[IFT = 10 mA Max]
MOC3033*
[IFT = 5 mA Max]

*Motorola Preferred Device

The MOC3031, MOC3032 and MOC3033 devices consist of gallium arsenide infrared emitting diodes optically coupled to a monolithic silicon detector performing the function of a Zero Voltage crossing bilateral triac driver.

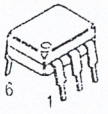
They are designed for use with a triac in the interface of logic systems to equipment powered from 115 Vac lines, such as teletypewriters, CRTs, printers, motors, solenoids and consumer appliances, etc.

- Simplifies Logic Control of 115 Vac Power
- Zero Voltage Crossing
- dv/dt of 2000 V/µs Typical, 1000 V/µs Guaranteed
- *To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.*

Recommended for 115 Vac(rms) Applications:

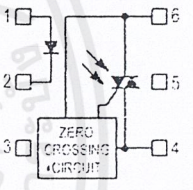
- Solenoid/Valve Controls
- Lighting Controls
- Static Power Switches
- AC Motor Drives
- Temperature Controls
- E.M. Contactors
- AC Motor Starters
- Solid State Relays

STYLE 6 PLASTIC



STANDARD THRU HOLE
CASE 730A–04

COUPLER SCHEMATIC



1. ANODE
2. CATHODE
3. NC
4. MAIN TERMINAL
5. SUBSTRATE
   DO NOT CONNECT
6. MAIN TERMINAL

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INFRARED LED			
Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
Total Power Dissipation @ T_A = 25°C Negligible Power in Output Driver Derate above 25°C	P_D	120 1.41	mW mW/°C
OUTPUT DRIVER			
Off-State Output Terminal Voltage	V_{DRM}	250	Volts
Peak Repetitive Surge Current (PW = 100 µs, 120 pps)	I_{TSM}	1	A
Total Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	150 1.76	mW mW/°C
TOTAL DEVICE			
Isolation Surge Voltage(1) (Peak ac Voltage, 60 Hz, 1 Second Duration)	V_{ISO}	7500	Vac(pk)
Total Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	250 2.94	mW mW/°C
Junction Temperature Range	T_J	−40 to +100	°C
Ambient Operating Temperature Range(2)	T_A	−40 to +85	°C
Storage Temperature Range(2)	T_{stg}	−40 to +150	°C
Soldering Temperature (10 s)	T_L	260	°C

1. Isolation surge voltage, V_{ISO}, is an internal device dielectric breakdown rating.
   For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.

GlobalOptoisolator is a trademark of Motorola, Inc.

(Replaces MOC3030/D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
INPUT LED						
Reverse Leakage Current (V_R = 3 V)		I_R	—	0.05	100	μA
Forward Voltage (I_F = 30 mA)		V_F	—	1.3	1.5	Volts
OUTPUT DETECTOR (I_F = 0 unless otherwise noted)						
Leakage with LED Off, Either Direction (Rated V_{DRM}(1))		I_{DRM1}	—	10	100	nA
Peak On-State Voltage, Either Direction (I_{TM} = 100 mA Peak)		V_{TM}	—	1.8	3	Volts
Critical Rate of Rise of Off-State Voltage		dv/dt	1000	2000	—	V/μs
COUPLED						
LED Trigger Current, Current Required to Latch Output (Main Terminal Voltage = 3 V(2))	MOC3031	I_{FT}	—	—	15	mA
	MOC3032		—	—	10	
	MOC3033		—	—	5	
Holding Current, Either Direction		I_H	—	250	—	μA
Isolation Voltage (f = 60 Hz, t = 1 sec)		V_{ISO}	7500	—	—	Vac(pk)
ZERO CROSSING						
Inhibit Voltage (I_F = Rated I_{FT}, MT1–MT2 Voltage above which device will not trigger.)		V_{IH}	—	5	20	Volts
Leakage in Inhibited State (I_F = Rated I_{FT}, Rated V_{DRM}, Off State)		I_{DRM2}	—	—	500	μA

1. Test voltage must be applied within dv/dt rating.
2. All devices are guaranteed to trigger at an I_F value less than or equal to max I_{FT}. Therefore, recommended operating I_F lies between max I_{FT} (15 mA for MOC3031, 10 mA for MOC3032, 5 mA for MOC3033) and absolute max I_F (60 mA).

TYPICAL ELECTRICAL CHARACTERISTICS

T_A = 25°C

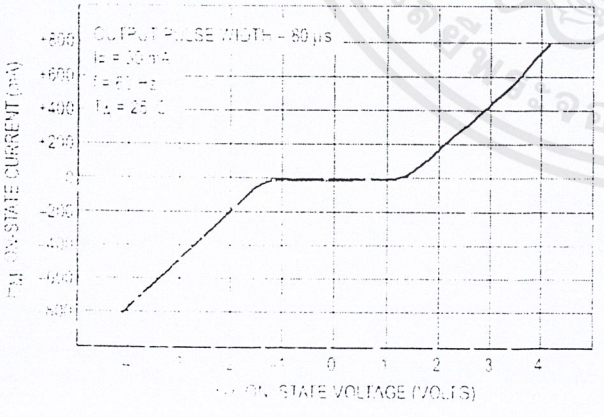


Figure 1. On-State Characteristics

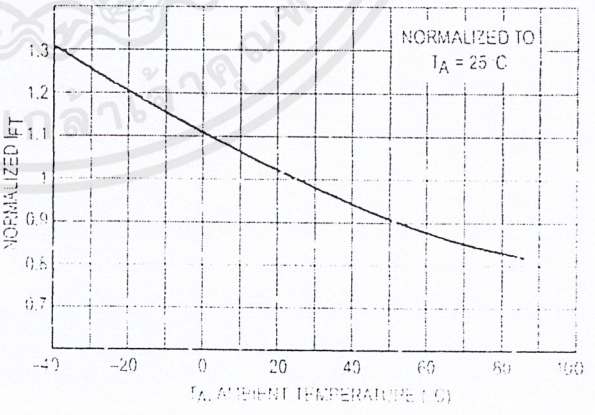


Figure 2. Trigger Current versus Temperature

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

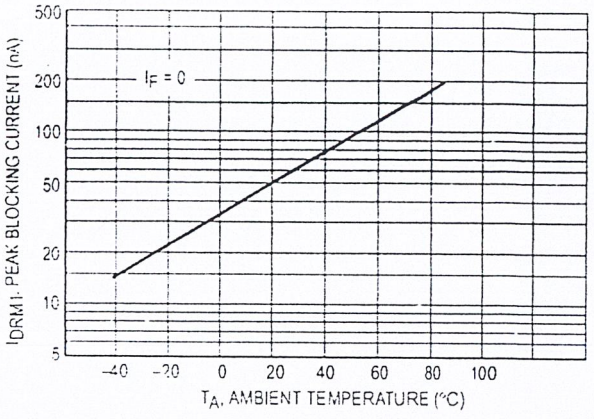


Figure 3. I_{DRM1}, Peak Blocking Current versus Temperature

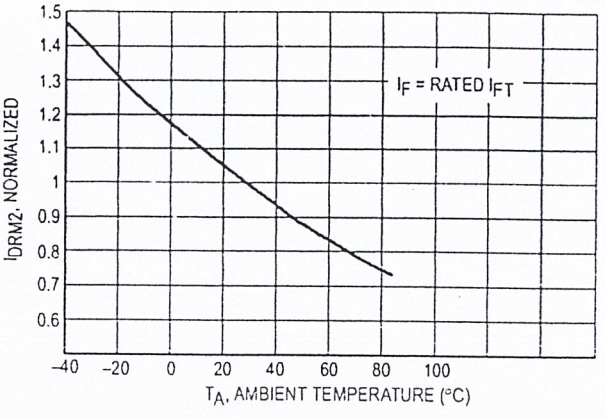


Figure 4. I_{DRM2}, Leakage in Inhibit State versus Temperature

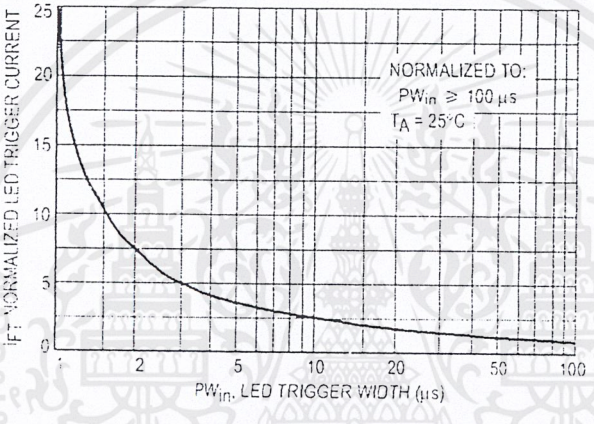
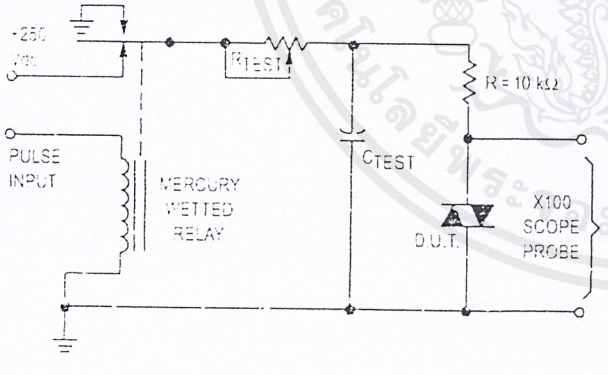


Figure 5. LED Current Required to Trigger versus LED Pulse Width



1. The mercury wetted relay provides a high speed repeated pulse to the D.U.T.
2. 100x scope probes are used, to allow high speeds and voltages.
3. The worst–case condition for static dv/dt is established by triggering the D.U.T. with a normal LED input current, then removing the current. The variable R_{TEST} allows the dv/dt to be gradually increased until the D.U.T. continues to trigger in response to the applied voltage pulse, even after the LED current has been removed. The dv/dt is then decreased until the D.U.T. stops triggering. τ_{RC} is measured at this point and recorded.

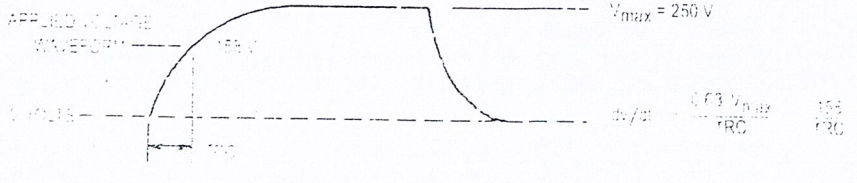
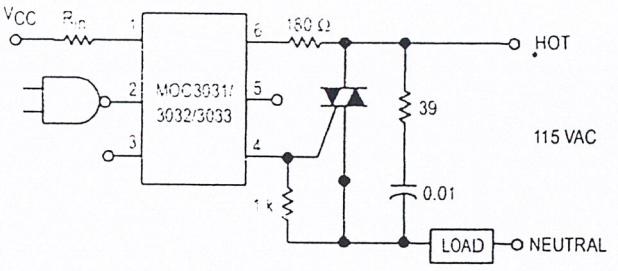


Figure 6. Static dv/dt Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

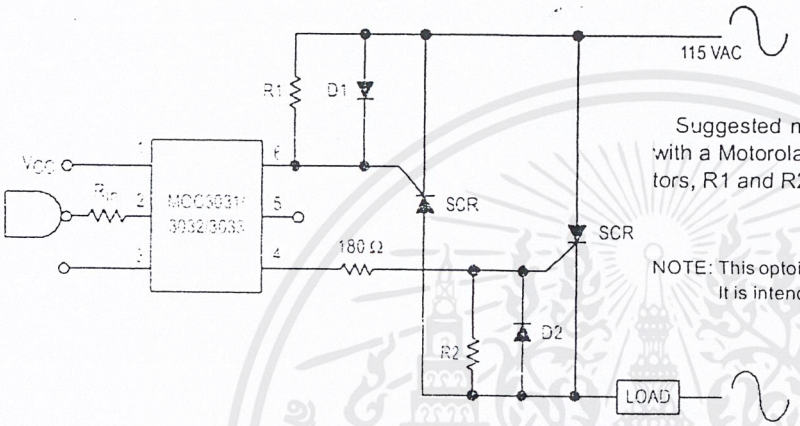


* For highly inductive loads (power factor < 0.5), change this value to 360 ohms.

Typical circuit for use when hot line switching is required. In this circuit the "hot" side of the line is switched and the load connected to the cold or neutral side. The load may be connected to either the neutral or hot line.

R_{in} is calculated so that I_F is equal to the rated I_{FT} of the part, 5 mA for the MOC3033, 10 mA for the MOC3032, or 15 mA for the MOC3031. The 39 ohm resistor and 0.01 μF capacitor are for snubbing of the triac and may or may not be necessary depending upon the particular triac and load used.

Figure 7. Hot–Line Switching Application Circuit



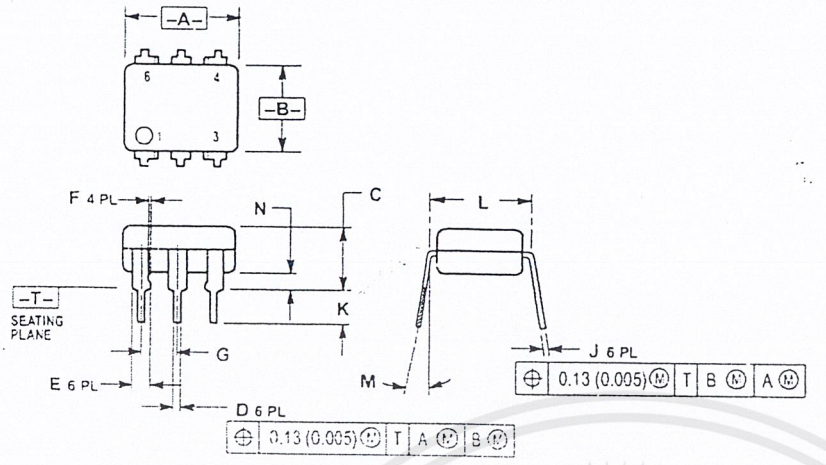
Suggested method of firing two, back–to–back SCR's, with a Motorola triac driver. Diodes can be 1N4001; resistors, R1 and R2, are optional 1 k ohm.

NOTE: This optoisolator should not be used to drive a load directly. It is intended to be a trigger device only.

Figure 8. Inverse–Parallel SCR Driver Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PACKAGE DIMENSIONS

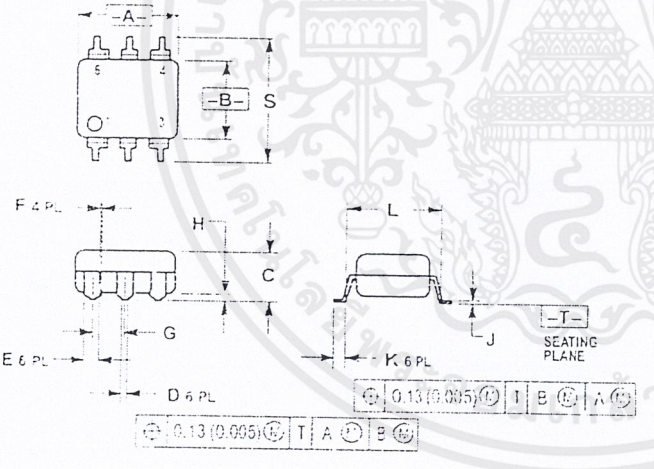


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.015	0.100	0.38	2.54

STYLE 6
PIN 1. ANODE
2. CATHODE
3. NC
4. MAIN TERMINAL
5. SUBSTRATE
6. MAIN TERMINAL

CASE 730A–04
ISSUE G



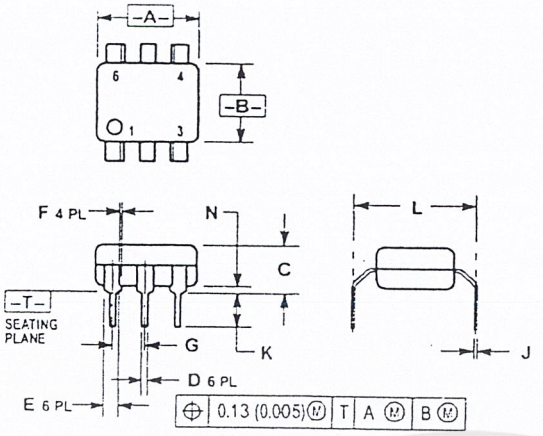
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
H	0.045	0.055	1.14	1.40
J	0.008	0.012	0.20	0.30
K	0.115	0.135	2.92	3.43
L	0.300 BSC		7.62 BSC	
S	0.332	0.368	8.43	9.35

*Consult factory for leadform option availability

CASE 730C–04
ISSUE D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.400	0.425	10.16	10.80
N	0.015	0.040	0.38	1.02

*Consult factory for leadform option availability

CASE 730D–05
ISSUE D

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and M are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:
USA/EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://Design-NET.com

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki,
6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ หาผลประโยชน์การค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes

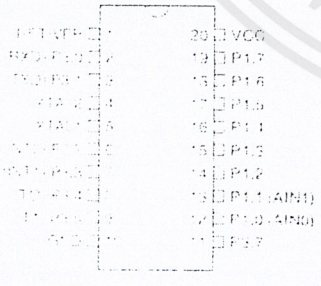
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Configuration

PDIP/SOIC



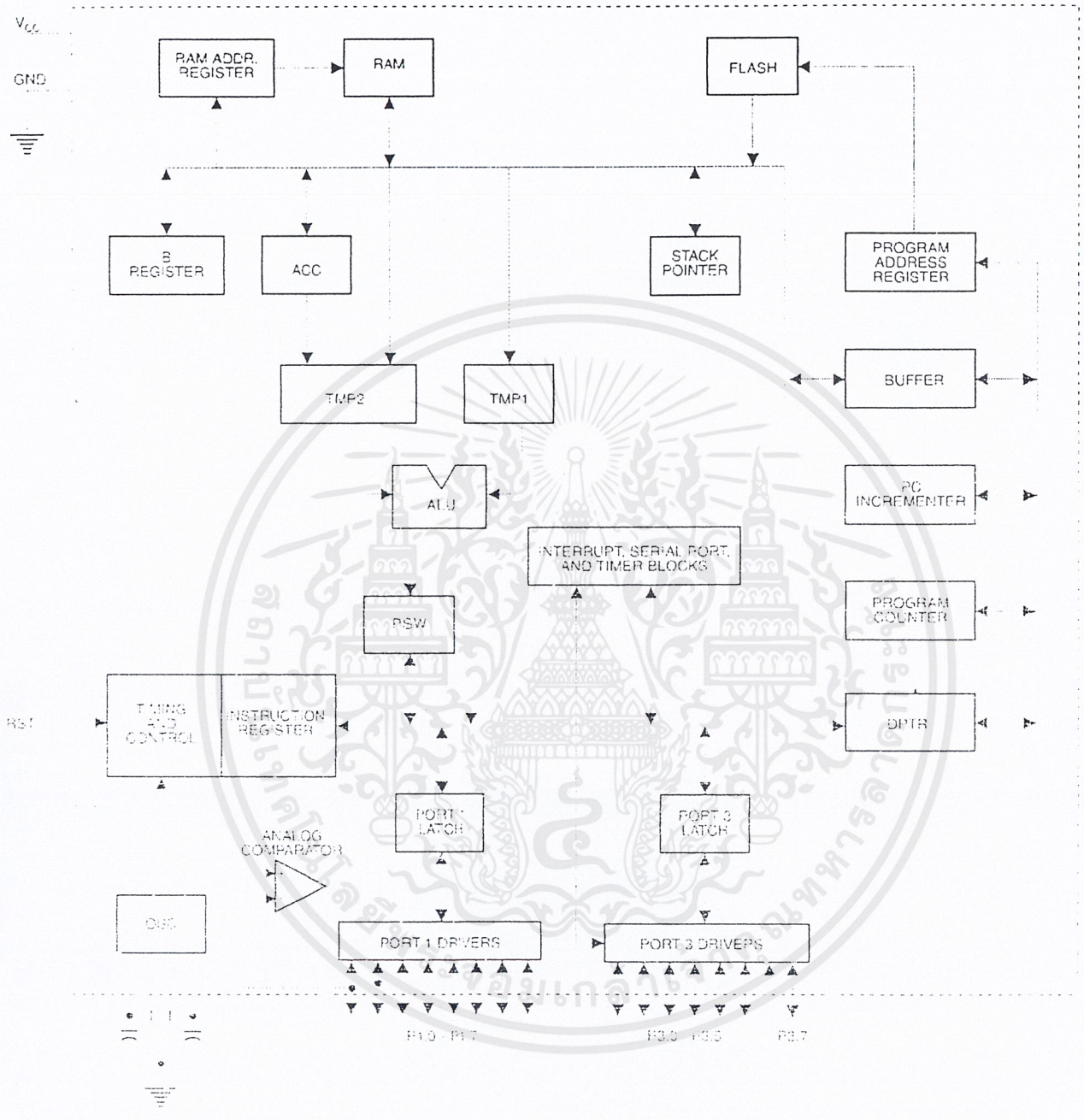
8-bit Microcontroller with 2K Bytes Flash

AT89C2051

Rev. 0368I–02/01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bi-irectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-irectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

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

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

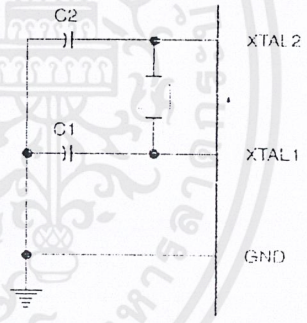
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

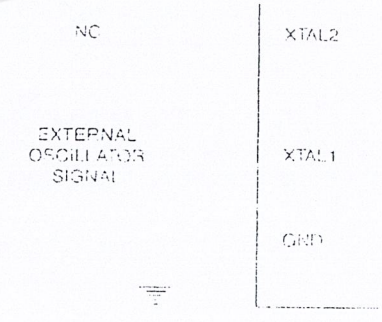
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XXX00000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0XX00000							0AFH
0A0H								0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H		SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Restrictions on Certain Instructions

The AT89C2051 and is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes[1]

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. *The Lock Bits can only be erased with the Chip Erase operation.*

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power-down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming The Flash

The AT89C2051 is shipped with the 2K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
   Apply power between V_{CC} and GND pins
   Set RST and XTAL1 to GND
2. Set RST to "H"
   Set P3.2 to "H"
3. Apply the appropriate combination of "H" or "L" logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7
5. Raise RST to 12V to enable programming.
6. Raise P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic "H" level and set pins P3.3 to P3.7 to the appropiate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2K bytes array or until the end of the object file is reached.
10. Power-off sequence:
    Set XTAL1 to "L"
    Set RST to "L"
    Turn V_{CC} power off

Data Polling: The AT89C2051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from "L" to "H".
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 21H indicates 89C2051

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

Mode		RST/VPP	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data[1][3]		12V		L	H	H	H
Read Code Data[1]		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V	[2]	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
2. Chip Erase requires a 10 ms PROG pulse.
3. P3.1 is pulled Low during programming to indicate RDY/BSY.

Figure 3. Programming the Flash Memory

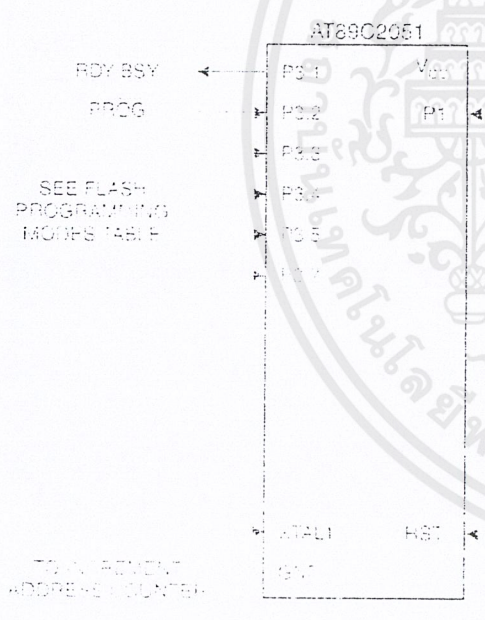
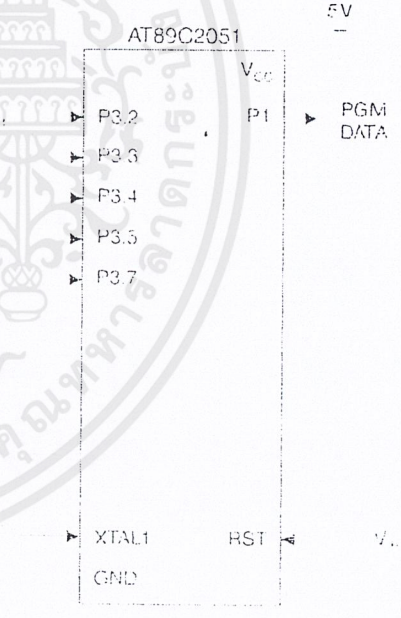


Figure 4. Verifying the Flash Memory



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

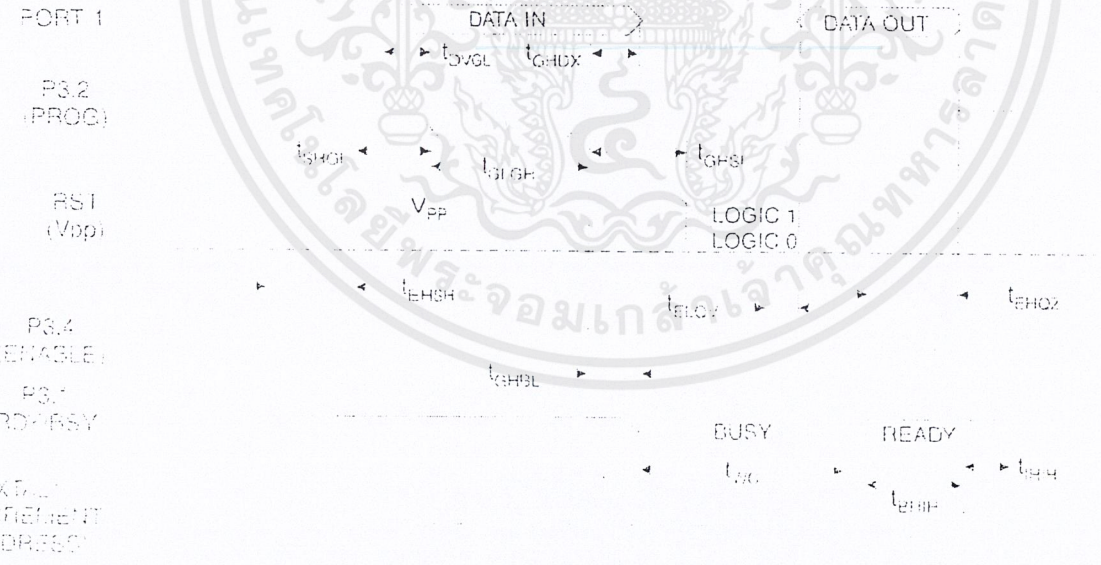
Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold after $\overline{PROG}$	1.0		µs
t_{EHSH}	P3.4 ($\overline{ENABLE}$) High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$ to Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

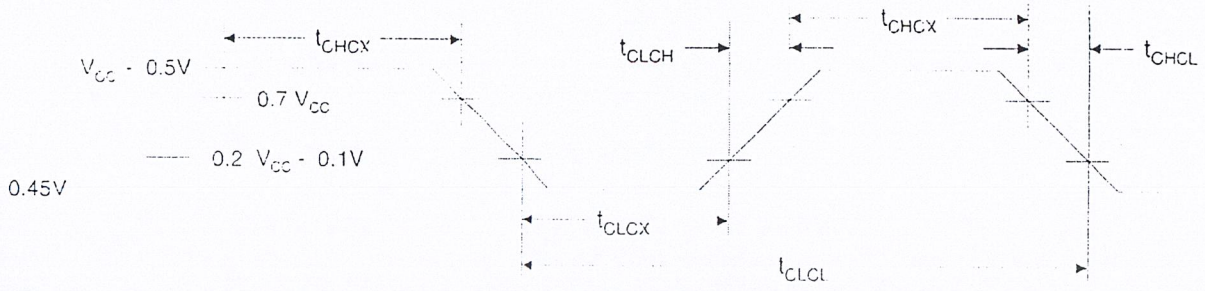
T_A = -40°C to 85°C, V_{CC} = 2.0V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.2 V_{CC} - 0.1	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage[1] (Ports 1, 3)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 2.7V		0.5	V
V_{OH}	Output High-voltage (Ports 1, 3)	I_{OH} = -80 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -30 µA	0.75 V_{CC}		V
		I_{OH} = -12 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-750	µA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	0 < V_{IN} < V_{CC}		±10	µA
V_{OS}	Comparator Input Offset Voltage	V_{CC} = 5V		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, V_{CC} = 6V/3V		15/5	mA
		Idle Mode, 12 MHz, V_{CC} = 6V/3V P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power-down Mode[2]	V_{CC} = 6V P1.0 & P1.1 = 0V or V_{CC}		100	µA
		V_{CC} = 3V P1.0 & P1.1 = 0V or V_{CC}		20	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 20 mA
   Maximum total I_{OL} for all output pins: 80 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	V_{CC} = 2.7V to 6.0V Min	Max	V_{CC} = 4.0V to 6.0V Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

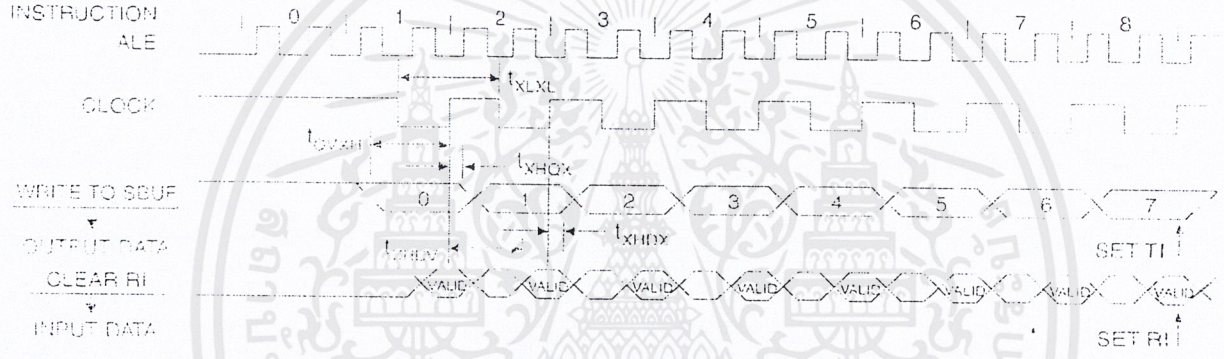
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

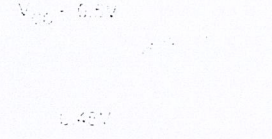
V_{CC} = 5.0V ± 20%; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Osc Min	12 MHz Osc Max	Variable Oscillator Min	Variable Oscillator Max	Units
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold after Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

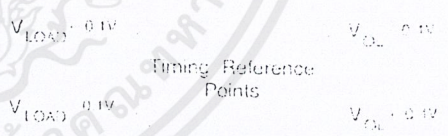


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]

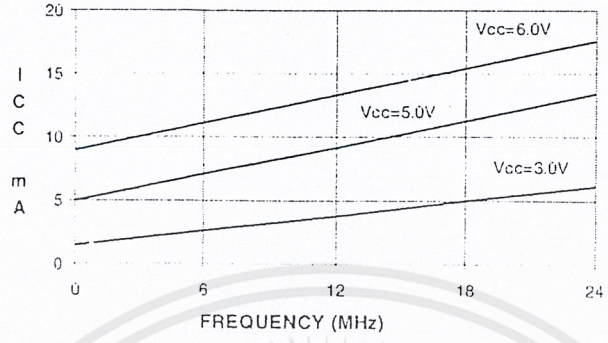


Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

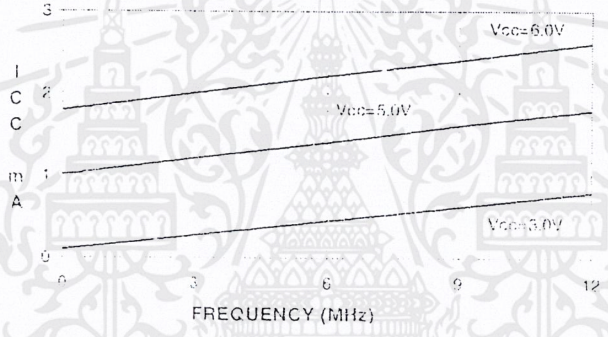
AT89C2051

TYPICAL ICC - ACTIVE (85°C)



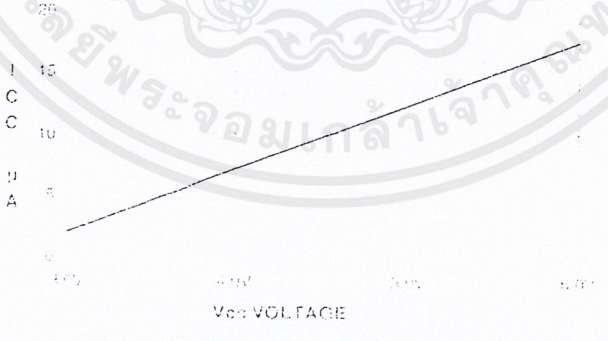
AT89C2051

TYPICAL ICC - IDLE (85°C)



AT89C2051

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



Notes:
1. XTAL1 tied to GND for I_{CC} (power-down)
2. P1.0 and P1.1 = V_{CC} or GND
3. Lock bits programmed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

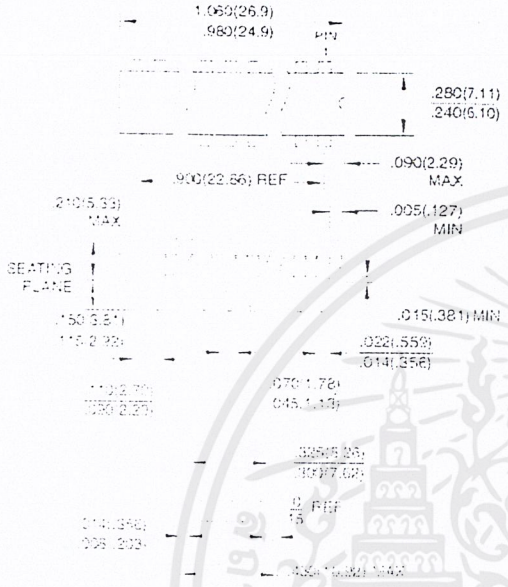
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89C2051-12PC	20P3	Commercial (0°C to 70°C)
		AT89C2051-12SC	20S	
		AT89C2051-12PI	20P3	Industrial (-40°C to 85°C)
		AT89C2051-12SI	20S	
24	4.0V to 6.0V	AT89C2051-24PC	20P3	Commercial (0°C to 70°C)
		AT89C2051-24SC	20S	
		AT89C2051-24PI	20P3	Industrial (-40 C to 85°C)
		AT89C2051-24SI	20S	

Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

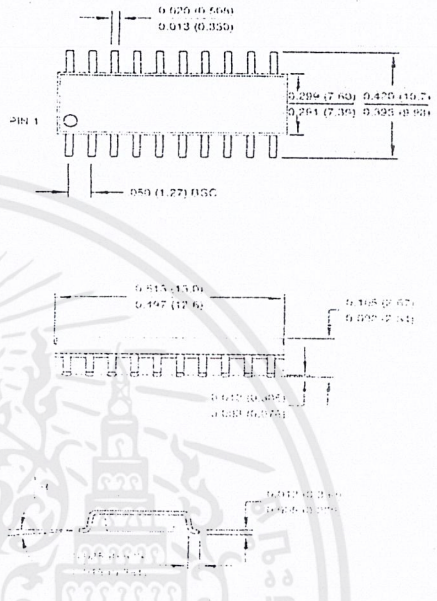
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

20P3, 20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-001 AD



20S, 20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)
Dimensions in Inches and (Millimeters)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 2000.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0804I–02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้