สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมและแสดงผลการทำงานของอุปกรณ์ไฟฟ้าแบบไร้สาย

(RF Wireless Remote and Monitoring Controller)

จัดทำโดย

นายวศิน อนันต์

นางสาววันวิสาข์ โรจน์บัณฑิต

วพ.
ก357ค
2550

เลขหมู่.....
เลขทะเบียน 82434
วัน,เดือน,ปี 11 ก.ค. 2551

b 11949466
i

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2550

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมและแสดงผลการทำงานของอุปกรณ์ไฟฟ้าแบบไร้สาย
(RF Wireless Remote and Monitoring Controller)

ผู้จัดทำ

1.นายวศิน	อนันต์	รหัส	47010677	ชั้นปีที่	4C
2.นางสาววันวิสาข์	โรจน์บัณฑิต	รหัส	47010690	ชั้นปีที่	4C

.....อาจารย์ที่ปรึกษา

(ผศ. พลผดุง ผดุงกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมและแสดงผลการทำงานของอุปกรณ์ไฟฟ้าแบบไร้สาย

นาย วศิน อนันต์ รหัส 47010677
น.ส. วันวิสาข์ โรจน์บัณฑิต รหัส47010690
ผ.ศ. ผลผดุง ผดุงกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการสร้างเครื่องควบคุมและแสดงผลการทำงานของอุปกรณ์ไฟฟ้าแบบไร้สายโดยผ่านทางคลื่นวิทยุ RF และใช้วิธีการส่งด้วยการมอดูเลตแบบ ASK โดยสัญญาณข้อมูลที่เข้ามามอดูเลตจะเป็นสัญญาณดิจิตอลที่ได้จากไมโครโปรเซสเซอร์ ส่วนทางเครื่องรับก็จะประกอบด้วยวงจรดีมอดูเลตสัญญาณ เพื่อให้ได้สัญญาณข้อมูลดิจิตอลออกมา โดยสัญญาณดิจิตอลนี้จะใช้แทนรหัสของอุปกรณ์ไฟฟ้าตัวที่ต้องการควบคุม โดยจะใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 เบอร์ P89V51RD2 เป็นตัวควบคุม ซึ่งจะสามารถเลือกตัวอุปกรณ์ไฟฟ้าตัวที่ต้องการควบคุมได้และสามารถตั้งเวลาได้ ทำให้สามารถควบคุมอุปกรณ์ไฟฟ้าได้โดยไม่ต้องมีสายสัญญาณ ในขณะเดียวกันที่ตัวรับจะสามารถส่งสถานะการทำงานของอุปกรณ์ไฟฟ้ากลับมาแสดงผลที่ตัวส่งได้เช่นเดียวกัน

ซึ่งในเทอมหลังนี้จะศึกษาการทำงานการสื่อสารในแบบขนาน ออกแบบโปรแกรม Visaul Basic เพื่อกำหนดเวลาในการเปิดหรือปิดอุปกรณ์ไฟฟ้าในแต่ละตัว เพิ่มหน้าจอ LCD เพื่อแสดงเวลาในขณะที่ปิดคอมพิวเตอร์ และเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ไฟฟ้าโดยใช้โปรแกรมภาษา C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RF Wireless Remote and Monitoring Controller

Mr.Wasin Anant 47010677

MissWanvisa Rojbundit 47010690

Ass Prof. Ponpadung Padungkun Advisor

Educational Year 2007

Abstact

This thesis presents a RF wireless remote and monitoring Controller which transfers data by radio frequency using ASK modulation. The modulated signal is a digital signal from microprocessor to assign the code of electrical device which is wanted to control . The receiver consists of ASK demodulator circuits. The digital signal is a code of electrical device which a uses wants to control. The controller is number P89V51RD2 in type 8051 microcontroller which is able to select the electrical device and determine the operating time for the electrical device . For the principal above, it does not have the transmission line to control electrical device. At the same time the receiver can transfers working data of electrical device return to transmitter.

In this term we will study about working of communication in parallel form ,designing Visual Basic program to concrol time in open and close each electrical device, increasing LCD screen to show time during open the computer and writing program to control electrical devices by C language.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ

อาจารย์ ผศ.พลผดุง ผดุงกุล (อาจารย์ที่ปรึกษา) และอาจารย์ภาควิชาอิเล็กทรอนิกส์ทุกท่าน ที่ให้การอุปการะในการให้คำปรึกษาและแนะนำเกี่ยวกับโครงงานเครื่องควบคุมและแสดงผลการทำงานของอุปกรณ์ไฟฟ้าแบบไร้สาย และให้ยืมใช้เครื่องมืออิเล็กทรอนิกส์ในการทดลอง และสั่งสอนให้ความรู้จนสามารถนำมาประยุกต์ใช้งานในการทำโครงงานครั้งนี้

คุณพ่อ คุณแม่ ที่คอยให้ความเป็นห่วง และให้กำลังใจพวกเราพวกเราเสมอมา ทั้งในด้านการทำงานและการเรียน

และก็ขอขอบคุณพี่เกี้ยงที่คอยช่วยเหลือและช่วยให้คำปรึกษาที่ดีเสมอมา และไม่เคยทอดทิ้งเรา...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	1
Abstract	2
กิตติกรรมประกาศ	3
สารบัญ	4
สารบัญรูป	6
บทที่ 1 บทนำ	8
บทที่ 2 ทฤษฎี	10
2. 1 Microcontroller เบอร์ P89V51RD2	10
2.2 DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)	14
2.2.1 โหมดการทำงานของ DS1307	17
2.2.2 โหมดการเขียนข้อมูล	17
2.2.3 โหมดการอ่านข้อมูล	17
2.3 พอร์ตขนาน	19
2.4. หลักการของบัส I2C	20
2.4.1 การทำงานบนบัส I2C	22
2.5. รีเลย์	25
2.5.1 หลักการของรีเลย์	25
บทที่ 3 การออกแบบ	26
3.1 ส่วนตั้งเวลาเปิด ปิด และเลือกอุปกรณ์	27
3.2 การส่งข้อมูลผ่านพอร์ตขนาน	27
3.3 วงจรรีเลย์	28
โปรแกรมภาคส่ง	34
โปรแกรมภาครับตัวที่ 1	65
โปรแกรมภาครับตัวที่ 2	68
บทที่ 4 การทดลอง	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การเชื่อมต่อบอร์ดกับพีซี 71
บทที่ 5 ผลการทดลอง 74
5.1 กราฟแสดงผลเมื่อไม่ได้ป้อนข้อมูล 74
5.2 กราฟแสดงผลไปควบคุมLED1 76
5.3 กราฟแสดงผลไปควบคุมLED1 และ LED8 78
5.4 กราฟแสดงผลไปควบคุมLED9 80
5.5 กราฟแสดงผลไปควบคุมLED9 และ LED15 82
5.6 กราฟแสดงผลไปควบคุมLED3 , LED4 , LED12 , LED13 84
5.7 กราฟแสดงผลไปควบคุมLEDทุกตัว 86
บทที่ 6 สรุปและวิจารณ์ 88
ปัญหาและการแก้ไข 88
เอกสารอ้างอิง 89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ 2.1 แสดงลักษณะขาต่างของ MCS-51 11

รูปที่ 2.2 การจัดขาของ IC DS1307 14

รูปที่ 2.3 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล 18

รูปที่ 2.4 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล 18

รูปที่ 2.5 พอร์ตขนาน 19

รูปที่ 2.6 ไดอะแกรมแสดงสถานะต่าง ๆ บนระบบบัส I2C 21

รูปที่ 2.7 รูปแบบของข้อมูลกำหนดแอดเดรสของอุปกรณ์บนระบบบัส I2C 22

รูปที่ 2.8 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงแบบ 7 บิต ของระบบบัส I2C 23

รูปที่ 2.9 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงแบบ 10 บิต ของระบบบัส I2C 24

รูปที่ 3.1 วงจรภาคส่ง 29

รูปที่ 3.2 วงจรโมดูลรับ ส่ง 30

รูปที่ 3.3 วงจรภาครับ 31

รูปที่ 3.4 วงจรสร้างฐานเวลา 32

รูปที่ 3.5 วงจรRELAYควบคุมหลอดไฟ 33

รูปที่ 4.1 หน้าจอVB ขณะกดปุ่ม D7 72

รูปที่ 4.2 LED7 ติด เมื่อกดปุ่ม D7 72

รูปที่ 4.3 หน้าจอVB ขณะกดปุ่ม /CO 73

รูปที่ 4.4 LED7 ติด เมื่อกดปุ่ม D5 73

รูปที่ 5.1 กราฟแสดงผลเมื่อไม่ได้ป้อนข้อมูล 74

รูปที่ 5.2 รูปแสดงผลเมื่อไม่ได้ป้อนข้อมูลออกหน้าจอคอมพิวเตอร์ 75

รูปที่ 5.3 กราฟแสดงผลไปควบคุมLED1 76

รูปที่ 5.4 รูปแสดงผลไปควบคุมLED1ออกหน้าจอคอมพิวเตอร์ 77

รูปที่ 5.5 กราฟแสดงผลไปควบคุมLED1 และ LED8 78

รูปที่ 5.6 รูปแสดงผลไปควบคุมLED1 และ LED8ออกหน้าจอคอมพิวเตอร์ 79

รูปที่ 5.7 รูปแสดงผลไปควบคุมLED9 80

รูปที่ 5.8 รูปแสดงผลไปควบคุมLED9 ออกหน้าจอคอมพิวเตอร์ 81

รูปที่5.9 กราฟแสดงผลไปควบคุมLED9 และ LED15 82

รูปที่5.10 กราฟแสดงผลไปควบคุมLED9 และ LED15 ออกหน้าจอคอมพิวเตอร์ 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.11 กราฟแสดงผลไปควบคุมLED3 , LED4 , LED12 , LED13 84
รูปที่ 5.12 กราฟแสดงผลไปควบคุมLED3 , LED4 , LED12 , LED13 ออกหน้าจอคอมพิวเตอร์ 85
รูปที่13 กราฟแสดงผลไปควบคุมLEDทุกตัว 86
รูปที่14 รูปแสดงผลไปควบคุมLED ทุกตัวออกหน้าจอคอมพิวเตอร์ 87

สารบัญตาราง

ตารางที่ 2.1 แสดงหน้าที่ของขาต่างๆใน MCS-51 13
ตารางที่ 2.2 โครงสร้างของหน่วยความจำภายในของ DS1307 15
ตารางที่ 2.3 ข้อมูลที่เก็บอยู่ในรีจิสเตอร์ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โครงงานนี้เป็นการสร้างเครื่องควบคุมและแสดงผลการทำงานของอุปกรณ์ไฟฟ้าแบบไร้สาย โดยจะควบคุมผ่านหน้าจอคอมพิวเตอร์ซึ่งใช้โปรแกรม Visual Basic ในการออกแบบ เชื่อมต่อไมโครคอนโทรลเลอร์กับพีซี ผ่านพอร์ตขนาน และส่งข้อมูลผ่านทางคลื่นวิทยุ RF และใช้วิธีการส่งด้วยการมอดูเลตแบบ ASK โดยสัญญาณข้อมูลที่เข้ามามอดูเลตจะเป็นสัญญาณดิจิตอลที่ได้จากไมโครโปรเซสเซอร์ ส่วนทางเครื่องรับก็จะประกอบด้วยวงจรดีมอดูเลตสัญญาณ เพื่อให้ได้สัญญาณข้อมูลดิจิตอลออกมา โดยสัญญาณดิจิตอลนี้จะใช้แทนรหัสของอุปกรณ์ไฟฟ้าตัวที่ต้องการควบคุม โดยจะใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 เบอร์ P89V51RD2 เป็นตัวควบคุม ซึ่งจะสามารถเลือกตัวอุปกรณ์ไฟฟ้าตัวที่ต้องการควบคุมได้และสามารถตั้งเวลาได้ ทำให้สามารถควบคุมอุปกรณ์ไฟฟ้าได้โดยไม่ต้องมีสายสัญญาณ ในขณะเดียวกันที่ตัวรับจะสามารถส่งสถานะการทำงานของอุปกรณ์ไฟฟ้ากลับมาแสดงผลที่ตัวส่งได้เช่นเดียวกัน

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของโปรแกรม Visual Basic เพื่อกำหนดข้อมูลที่ต้องการให้เชื่อมต่อไปยังไมโครคอนโทรลเลอร์ โดยผ่านพอร์ตขนาน
2. เพื่อศึกษาการรับข้อมูลที่ได้จากภายนอก มาเก็บเรียบเรียงในไมโครคอนโทรลเลอร์
3. เพื่อศึกษาการส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังโมดูล โดยผ่านพอร์ตอนุกรม
4. เพื่อศึกษาการทำงานของระบบสื่อสารไร้สายโดยใช้การมอดูเลทแบบดิจิตอล ชนิด Amplitude Shift Keying (ASK)

ขอบเขตของโครงงาน

1. มีตัวส่ง 1 ชุด และ ตัวรับ 2 ชุด ตัวรับแต่ละชุดควบคุมหลอดไฟ 8 ดวง
2. ตัวส่ง และตัวรับ มีการรับส่งผ่านทางพอร์ตอนุกรม
3. ที่ตัวรับจะมีการควบคุมให้มีการเปิด-ปิดไฟโดยใช้ Relay
4. เมื่อตัวรับได้รับสัญญาณที่ส่งมาแล้วและได้ทำตามคำสั่งที่ตั้งไว้แล้วจะมีการส่งสัญญาณกลับไปให้ที่ตัวส่งว่าได้รับสัญญาณแล้ว เพื่อแสดงผลการทำงานของหลอดไฟแต่ละดวงออกทางหน้าจอคอมพิวเตอร์
5. ใช้ Microcontroler ในการควบคุมการทำงาน โดยกำหนดการทำงานที่หน้าจอคอมพิวเตอร์ผ่านโปรแกรม Visual Basic ไปยังไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. มีฐานเวลาจริงเชื่อมต่อกับไมโครคอนโทรลเลอร์เพื่อ ให้สามารถเปรียบเทียบเวลาจริงหากปิดคอมพิวเตอร์
7. ใช้โปรแกรม Visual Basic ในการสั่งคำสั่งข้อมูลผ่านพีซี ทางพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 Microcontroller เบอร์ P89V51RD2

เป็นไมโครคอนโทรเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช(flash memory) ของ Philips Semiconductor ซึ่งมีคุณสมบัติทางเทคนิคที่โดดเด่นดังนี้

1. เป็นไมโครคอนโทรลเลอร์ 8 บิต ที่เข้ากันได้กับไมโครคอนโทรลเลอร์ MCS-51พื้นฐาน
2. หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ลบและเขียนใหม่ได้ถึง 10,000 ครั้ง ขนาดของหน่วยความจำโปรแกรมสูงถึง 64 กิโลไบต์
3. หน่วยความจำข้อมูลแรมภายในมีขนาด 1 กิโลไบต์
4. โปรแกรมข้อมูลลงในหน่วยความจำโปรแกรมแบบ ในระบบ (ISP : In-system programming)
5. ความถี่สัญญาณนาฬิกาสูงสุด 40 MHz ในกรณีทำงานด้วยสัญญาณนาฬิกาภายใน 12 ลูกต่อแมชีนไซเคิล

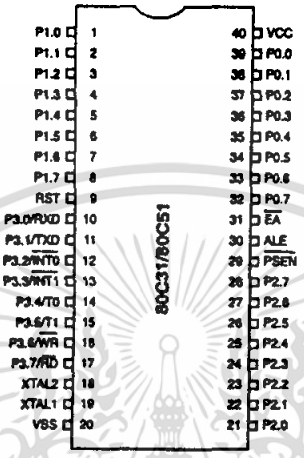
P89V51RD2 ได้รับการกำหนดให้ทำงานเบื้องต้นในโหมดสัญญาณนาฬิกา 12 ลูกต่อแมทชีนไซเกิล สามารถเลือกเปลี่ยนเป็น 6 สัญญาณนาฬิกาต่อแมทชีนไซเกิลได้ด้วยการโปรแกรมแบบ ISP แต่เมื่อเปลี่ยนแล้ว หากต้องการเปลี่ยนกลับเป็น 2 สัญญาณนาฬิกาต้องใช้การโปรแกรมแบบขนาน

1. ขาพอร์ต 8 บิต 4 พอร์ต แบบกึ่งสองทิศทางเป็นได้ทั้งอินพุต และเอาต์พุต
2. อุปกรณ์เพอริเฟอรัลภายในไมโครคอนโทรลเลอร์สามารถทำงานด้วยความเร็ว 12 สัญญาณนาฬิกาภายในต่อแมทชีนไซเกิลได้ แม้ว่าซีพียูจะทำงานด้วยความเร็ว 6 สัญญาณนาฬิกาภายในต่อแมทชีนไซเกิล
3. มีวงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์
4. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 ตัว(ไทเมอร์ 0,1 และ 2
5. มีรีจิสเตอร์ตัวชี้ตำแหน่งข้อมูลหรือ DPTR 2 ตัว
6. สามารถรับรองแหล่งกำเนิดอินเตอร์รัปต์ได้ 8 ประเภท
7. กำหนดนัยสำคัญของการตอบสนองอินเตอร์รัปต์ได้ 4 ระดับ
8. สามารถติดต่อหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. มีวอตช์อกไทเมอร์
10. มีโมดูลวงจรนับโปรแกรมได้ (PCA: Programmable Counter Array) ซึ่งบรรจุวงจรตรวจจับสัญญาณ(capture),เปรียบเทียบสัญญาณ(compare)วงจรมอดูเลชั่นทางความกว้างพัลส์(PWM) และวอตช์ด็อกไทเมอร์(watchdog timer)

MCS-51 Pin & Timing Diagram



รูปที่ 2.1 แสดงลักษณะขาต่างของ MCS-51

Pin	I / O / Z	หน้าที่ / การทำงาน
P0.0-P0.7	I / O / Z	Port 0 เป็น Open drain bidirectional I/O Port 8 Bit เมื่อใช้เป็น Output มันสามารถรับ (Sink) กระแสของ Input แบบ LS TTL ได้ 8 ตัว สำหรับ Port 0 Bit ที่ถูกส่งค่า '1' ออกมา จะทำให้ขานั้นเป็น Float ซึ่งสามารถใช้งานเป็น High impedance Input ได้ นอกจากนี้ ยังทำหน้าที่เป็น Addr. Bus ส่วนของ Low byte (PCL) สลับกับ Data Bus ในกรณีที่มีการใช้งาน External Memory ในการนี้ Logic '1' จะถูกสร้างจาก Strong Internal pull-up และท้ายสุด Port 0 จะใช้ส่งค่าที่มีอยู่ใน Internal program memory ออกมาเพื่อการตรวจสอบ (Program verification) แต่จะต้องต่อ External pull-up ให้ด้วย
P1.0-P1.7	I / O	Port 1 เป็น Bidirectional I/O Port 8 Bit with Internal pull-up เมื่อใช้เป็น Output มันสามารถรับ/จ่าย (Sink/Source) กระแสของ Input แบบ LS TTL ได้ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ตัว สำหรับ Port 1 Bit ที่ถูกส่งค่า '1' ออกมา จะทำให้ขานั้นเป็น High จาก Internal pull-up ซึ่งสามารถใช้งานเป็น Input ได้ และเมื่อถูกป้อนด้วย Logic '0' นั้น มันจะให้กระแสออกมาได้ซึ่งกระแสจะไหลมาจาก Internal pull-up นั่นเอง (IIL, on data sheet) นอกจากนี้ สำหรับ 80C52, 83C154 and 83C154D ขา Port 1.0 และ P1.1 จะสามารถใช้งานเป็น T2, T2EX (T2: Timer 2 input, T2EX: Timer 2 "Capture" input)
P2.0-P2.7	I / O	Port 2 เป็น Bidirectional I/O Port 8 Bit with Internal pull-up เมื่อใช้เป็น Output มันสามารถรับ/จ่าย (Sink/Source) กระแสของ Input แบบ LS TTL ได้ 4 ตัว นอกจากนี้ Port 2 ยังทำหน้าที่เป็น Address Bus ในส่วนของ High byte (PCH) ในกรณีที่มีการใช้งาน External Memory ที่ใช้การอ้าง Address แบบ 16 Bit ในการนี้ Logic '1' จะถูกสร้างจาก Strong Internal pull-up
P3.0-P3.7	I / O I O I I I I O O	Port 3 เป็น Bidirectional I/O Port 8 Bit with Internal pull-up นอกจากนี้แล้ว ขา ต่างๆ ของ Port 3 จะมีการใช้งานเป็น Special feature ของ MCS-51 ดังต่อไปนี้ P3.0: RXD (Serial input port) P3.1: TXD (Serial output port) P3.2: INT0 (External interrupt 0) P3.3: INT1 (External interrupt 1) P3.4: T0 (Timer 0 external input) P3.5: T1 (Timer 1 external input) P3.6: WR (External data memory write strobe) P3.7: RD (External data memory read strobe)
RST	I	Reset เป็นสัญญาณป้อนให้กับ MCS เพื่อเป็นการกำหนดค่าเริ่มต้นต่างๆ การทำ Reset นี้จะต้องทำทุกครั้งที่จ่ายไฟให้กับวงจร ในการ Reset สัญญาณ RST ต้องมีอยู่อย่างน้อย 2 Machine cycles จึงจะทำการ Reset ได้
ALE	O	Address Latch Enable เป็นสัญญาณ จะไป ทำให้ค่าของ PCL ถูกเก็บเข้าที่ LATCH ในระหว่างที่ ค่าของ PCL ออกมาที่ P0 สัญญาณ ALE นี้ จะมีออกมาตลอดเวลา (ไม่ว่าจะมีการใช้ External memory หรือไม่) เป็น

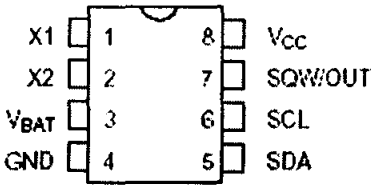
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ความถี่ 1/6 ของสัญญาณ Clock ซึ่งสามารถนำไปใช้เป็นฐานเวลาให้กับวงจรภายนอกได้ แต่อย่างไรก็ดี ในกรณีที่เป็นการติดต่อกับ External data memory สัญญาณ ALE จะข้ามไปหนึ่ง ลูก
$\overline{\text{PSEN}}$	O	Program Store Enable สัญญาณ $\overline{\text{PSEN}}$ ซึ่งเป็นสัญญาณที่ใช้ในการ Fetch คำสั่งจาก External program memory นั้น จะไม่ทำงานเมื่อเป็นการ Fetch คำสั่งจาก Internal program memory สัญญาณ $\overline{\text{PSEN}}$ นี้จะมีเกิดขึ้น สองครั้ง ต่อหนึ่ง Machine cycle ยกเว้นในกรณีที่เป็นการติดต่อกับ External data memory สัญญาณ $\overline{\text{PSEN}}$ จะข้ามไป
$\overline{\text{EA}}$	I	External Access โดยการต่อขาสัญญาณ $\overline{\text{EA}}$ เป็น 'High' การ Fetch คำสั่งที่ Address น้อยกว่า 0FFFh (4K สำหรับ 80C51) จะได้จาก Internal program memory และถ้าเป็น Address ตั้งแต่ 1000h ก็ จะเป็นการอ่านจาก External program memory ถ้าต่อขา $\overline{\text{EA}}$ เข้ากับ 'Low' การ Fetch คำสั่งทั้งหมดจะกระทำกับ External program memory
XTAL1	I	Input to inverting oscillator amplifier และใช้สำหรับการป้อน External clock
XTAL2	O	Output to inverting oscillator amplifier

ตารางที่ 2.1 แสดงหน้าที่ของขาต่างๆใน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)



รูปที่ 2.2 การจัดขาของ IC DS1307

VCC	Primary Power Supply
X1, X2	32.768 kHz Crystal Connection
VBAT	+3 Volt Battery Input
GND	Ground
SDA	Serial Data
SCL	Serial Clock
SQW/OUT	Square wave/Output Driver

เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรธินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100 ซึ่งมีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้ และมีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I2C โดยทำงานเป็นอุปกรณ์สเลฟเสมอ ส่วนประกอบหลักที่สำคัญคือ วงจรออสซิลเลเตอร์ เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง มีการเก็บค่าของเวลาไว้ในหน่วยความจำนอนโวลาไทล์แรม64ไบต์ ซึ่งจัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์และเก็บข้อมูลทั่วไป 56 ไบต์ วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากต่ำกว่า 1.25 x Vbat ก็จะควบคุมให้ DS1307 หยุดการทำงาน ทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นในการใช้งานต้องระมัดระวังอย่าให้ไฟเลี้ยงต่ำกว่า 1.25 x Vbat หรือประมาณ 3.75 V ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า Vbat ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดสรรหน่วยความจำภายใน DS1307

การจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307ประกอบด้วย พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00-06 เป็นพื้นที่ของรีจีสเตอร์ค่าเวลาใช้ในเก็บข้อมูลเกี่ยวกับเวลา ที่แอดเดรส 07 เป็นพื้นที่ของรีจีสเตอร์ควบคุมการทำงานของ DS1307

รีจิสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07มีรายระเอียดของแต่ละบิตดังนี้

OUT(Output control) : ใช้ควบคุมระดับลอจิกที่ขา SQW OUT ในกรณีที่ดิสเอเบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น 1 ที่ขา SQW OUT ก็จะเป็น 1 ถ้าบิตนี้เป็น 0 ที่ขา SQW OUT ก็จะเป็น 0

SQWE (Square Wave Enable) : ใช้เอ็นเอเบิลวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SQW OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออกให้กำหนดบิตนี้เป็น 1

BIT 7	6	5	4	3	2	1	0	
00H	CH	10 SECONDS		SECONDS				00–59
01H	X	10 MINUTES		MINUTES				00–59
02H	X	12/24	10 HR A/P	10 HR	HOURS			01–12 00–23
03H	X	X	X	X	X	DAY		1–7
04H	X	X	10 DATE		DATE			01–28/29 01–30 01–31
05H	X	X	X	10 MONTH	MONTH			01–12
06H	10 YEAR				YEAR			00–99
07H	OUT	X	X	SQWE	X	X	RS1	RS0 Control Register
08H – 3FH	RAM 56 BYTE							USER RAM

ตารางที่2.2 โครงสร้างของหน่วยความจำภายในของ DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่เก็บอยู่ในรีจิสเตอร์ 00H-06H จะเป็นแบบ BCD (Binary Code Decimal)

00H	SECOND	เก็บค่าวินาที โดยบิตที่สำคัญ คือบิต 7 CH (Clock Halt) เมื่อบิต 7 นี้เป็น 1 จะทำ ให้สัญญาณ นาฬิกา หยุดเดิน
01H	MINUTES	เก็บค่านาที 00-59
02H	HOUR	เก็บค่าชั่วโมง เมื่อบิต 6 เป็นบิตที่กำหนดการแสดงผแบบ 12/24 ช.ม เมื่อบิต 6 =1 จะเก็บค่า 01-12 ,เมื่อบิต 6 =0 จะเก็บค่า 01-24
03H	DAY	เก็บค่าวัน 1-7
04H	DATE	เก็บค่าที่ 01-31
05H	MONTH	เก็บค่าเดือน 01-12
06H	YEAR	เก็บค่าปี 00-99
07H	Control Register	รีจิสเตอร์ควบคุมการทำงานของขา SQW \OUT ถ้าบิต 4 SQWE = 1 กำหนดให้ขา SQW \OUT กำเนิดความถี่ ตามความถี่ที่กำหนดโดย RS1,RS0 ถ้าบิต 4 SQWE = 0 กำหนดให้ขา SQW \OUT จะมีค่าตามบิต 7 คือขา OUT
08H-3FH	UserRam	หน่วยความจำใช้งานทั่วไป 56 byte

RS1	RS0	ความถี่ output ที่ขา SQW/OUT
0	0	1Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.3 ข้อมูลที่เก็บอยู่ในรีจิสเตอร์

2.2.1 โหมดการทำงานของ DS1307

มี 2 โหมดคือ โหมดเขียนข้อมูล และ โหมดอ่านข้อมูล ในการใช้งานปกติจะใช้เฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

2.2.2 โหมดการเขียนข้อมูล

เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนดสภาวะเริ่มต้น (START : S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้ง จึงสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด (STOP : P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

2.2.3 โหมดการอ่านข้อมูล

เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่าน ซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยDS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูล ให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้งตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

82434

สภาวะ เริ่มต้น	ข้อมูลกำหนด แอดเดรส	R/W		แอดเดรส ของแรม		ข้อมูล		ข้อมูล		สภาวะ หยุด
S	1101000	0	A	xxxxxxxx	A	xxxxxxxx	A	Xxxxxxxx	A	P

รูปที่ 2.3 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล

สภาวะ เริ่มต้น	ข้อมูลกำหนด แอดเดรส	R/W		ข้อมูล		ข้อมูล		ข้อมูล		สภาวะ หยุด
S	1101000	1	A	xxxxxxxx	A	xxxxxxxx	A	Xxxxxxxx	A	P

รูปที่2.4 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.พอร์ตขนาน

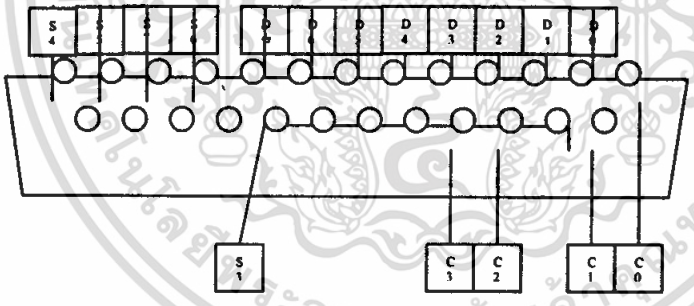
การส่งข้อมูลผ่านพอร์ตขนาน

ลักษณะแรงดันที่ส่งออกมาเป็น TTL โดยสัญญาณลอจิก 1 เท่ากับ 5 โวลต์ และลอจิก 0 จะเท่ากับ 0 โวลต์ และพอร์ตขนานของเครื่องคอมพิวเตอร์ประกอบด้วยสัญญาณทั้งหมด 25 เส้นสัญญาณ แบ่งเป็น 3 กลุ่ม

Data Port จำนวน 8 เส้นสัญญาณ

Status Port จำนวน 5 เส้นสัญญาณ

Control Port จำนวน 4 เส้นสัญญาณ



รูปที่2.5 พอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักการของบัส I2C

บัส I2C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือ เรียกว่า โปรโตคอล (protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และ อุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่งต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูลเรียกว่า ตัวส่ง (transmitter) อุปกรณ์ที่เป็นผู้รับข้อมูลเรียกว่า ตัวรับ (receiver) ในอุปกรณ์บนบัส I2Cสามารถเป็นได้ทั้งตัวรับและตัวส่งบางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I2C เรียดว่า มาสเตอร์ (master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I2C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I2C คือ

1.) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

2.) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลง ขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาควบคุมแทน

สภาวะที่เกิดขึ้นบนบัส I2C มีด้วยกัน 5 สภาวะ ดังนี้

1.) บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั้นหมายความว่า การถ่ายทอดข้อมูลสมารถเริ่มต้นขึ้นได้

2.) เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่าสภาวะเริ่มต้น (START)

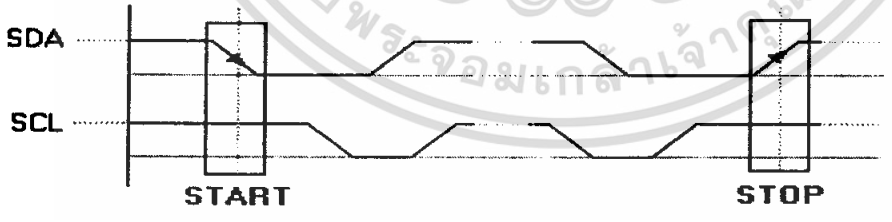
3.) หยุดการถ่ายทอดข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.) ข้อมูลดำรงอยู่บนบัส (data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น "0" หรือ "1" ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือ สภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้น เกิดความผิดพลาดขึ้น

รับข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูล มา 1 บิตเรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์ จะทำการส่งสัญญาณรับรู้พิเศษ ซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ ที่ส่งมาจาก ตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้ เพื่อตอบสนอง ให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

5.) รับข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูล มา 1 บิตเรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์ จะทำการส่งสัญญาณรับรู้พิเศษ ซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ ที่ส่งมาจาก ตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้ เพื่อตอบสนอง ให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว



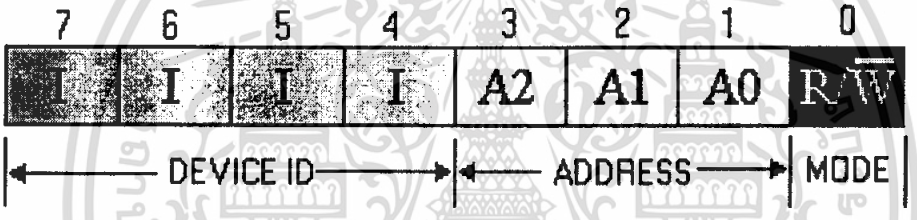
รูปที่ 2.6 ไดอะแกรมแสดงสถานะต่าง ๆ บนระบบบัส I2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 **การทำงานบนบัส I2C**

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่าง ๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I2C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มากใช้การอ้างถึงแบบ 7 บิต ก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้ เรียบร้อยแล้วก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้น หัวใจสำคัญในอันดับแรกของการทำงานบนบัส I2C คือการอ้างถึงอุปกรณ์แต่ละตัว ต่อไปนี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ



รูปที่ 2.7 รูปแบบของข้อมูลกำหนดแอดเดรสของอุปกรณ์บนระบบบัส I2C

การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อหรือ ข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 2-22 ใน 7 บิตบนรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็นบิตกำหนดแอดเดรส คงที่ (fixed address bit) จำนวน 4 บิตซึ่งข้อมูลนี้อุปกรณ์แต่บะตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเนบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ ที่มีการเชื่อมต่อแบบบัส I2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลยังอุปกรณ์ สเลฟตัว นั้นๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์ส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลฟ ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวเอย่างไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุม เพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

ข้อมูลในไบต์ต่อมา คือ ข้อมูลที่ทำการถ่ายทอดจริง (data) หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณ รับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลยังคงสามารถดำเนินต่อไปได้ ในรูป ที่ 2-23 แสดงรูปแบบข้อมุลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I2C ของการอ้างถึงแบบ 7 บิต

	ข้อมูลไบต์ที่ 1 เลือกแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ และจุดมุ่งหมายในการติดต่อ(อ่านหรือเขียน)			ข้อมูลไบต์ที่ 2 ส่งข้อมูลควบคุมการทำงานของอุปกรณ์ที่ต้องการติดต่อ control word			ข้อมูลไบต์ต่อไป ข้อมูลจริงที่ต้องการถ่ายทอด		
เริ่มต้น	A6-A0	R/W=0	รับรู้	C6-C0	R/W	รับรู้	D7-D0	รับรู้	หยุด

รูปที่2.8 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงแบบ 7 บิต ของระบบบัส I2C

ในการอ้างถึงแบบ 10 บิต

ในการอ้างถึงแบบนี้ ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกับแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิต บนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิ ถัดมาเป็นบิตแอดเดรส ของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือหรือข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูล ไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ข้อมูลไบต์ที่1 เลือกการอ้างอิงถึงแบบ10บิต			ข้อมูลไบต์ที่ 2 ข้อมูลแอดเดรส 8บิตต่ำ		ข้อมูลไบต์ที่3 ส่งข้อมูลควบคุม การทำงาน ของอุปกรณ์ที่ต้องการ ติดต่อหรือcontrol word			ข้อมูลไบต์ต่อไป ข้อมูลจริงที่ต้องการ ถ่ายทอด		
เริ่มต้น	1 1 1 1 0 A9 A8	R/W = 0	รับรู้	A7-A0	รับรู้	C6-C0	R/W	รับรู้	D7-D0	รับรู้	หยุ

รูปที่ 2.9 รูปแบบของข้อมูลที่ใช้ในการอ้างถึงแบบ 10 บิต ของระบบบัส I2C

เช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2-24 แสดงรูปแบบข้อมูลอนุกรม ของการอ้างถึงแบบ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5รีเลย์

เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (solenoid)รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor)ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา
2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

2.5.1 หลักการของรีเลย์

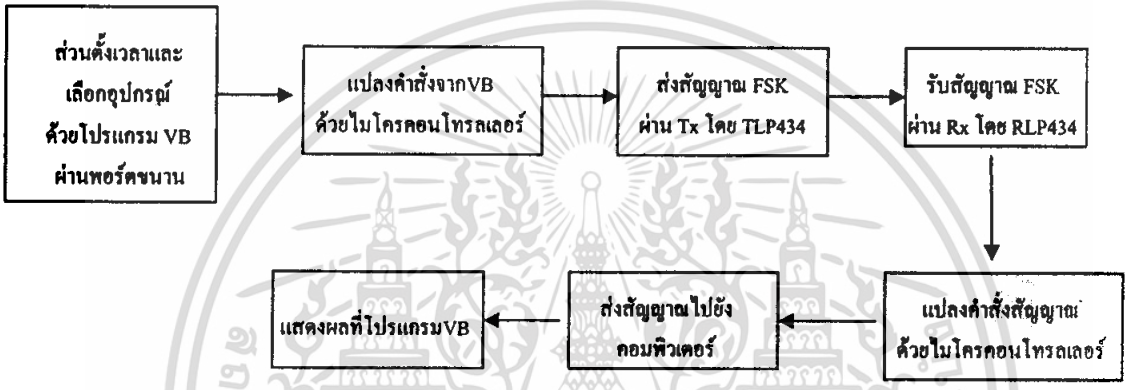
รีเลย์คืออุปกรณ์ที่เราควบคุมมันด้วยไฟฟ้าเพื่อให้ไปปิด หรือเปิดหน้าสัมผัส ทำหน้าที่เหมือนกับสวิตช์ (ซึ่งจะไปควบคุมวงจรเดียวกันหรือวงจรอื่นๆ อีกทอดหนึ่งหน้าสัมผัสมีหลายแบบ(ภาวะปกติ)เมื่อไม่มีกระแสป้อนเข้าขดลวด (หน้าสัมผัสของรีเลย์อยู่ในลักษณะใดจึงมักมีอักษรย่อเพิ่มเติมเข้ามาอีก เช่น NO)ย่อมาจาก normally open (หมายถึงภาวะปกติหน้าสัมผัสกับขั้วแยกจากกัน และจะแตะกันก็ต่อเมื่อขดลวดของรีเลย์ ได้รับปริมาณกระแสมากพอ อีกตัวหนึ่ง คือ NC)ย่อมาจาก normally closed (หมายถึงภาวะปกติหน้าสัมผัสกับขั้วจะแตกกัน และจะแยกจากกันก็ต่อเมื่อขดลวดของรีเลย์ได้รับกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

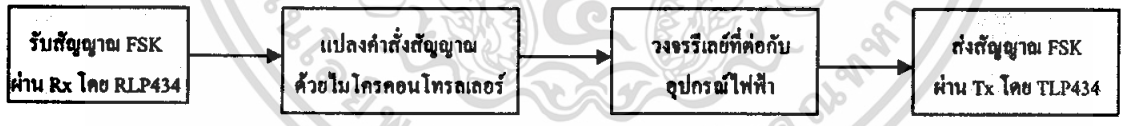
บทที่ 3
การออกแบบ

ในชิ้นงานนี้เราสามารถแสดงขั้นตอนการทำงานได้เป็น Block diagram ดังนี้

<u>ภาคส่ง</u>



<u>ภาครับ</u>



จากการป้อนคำสั่งของผู้ใช้ผ่านเครื่องคอมพิวเตอร์ โดยใช้โปรแกรมVisual Basic โดยสั่งออกทางพอร์ตขนานของเครื่องคอมพิวเตอร์ เพื่อส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ ซึ่งจะได้ข้อมูลเป็นดิจิตอลขนาด 8 บิต โดยจะส่งมาทั้งหมด 7 ชุด เมื่อรับค่าที่ได้มาแล้วจะเก็บข้อมูลไว้ในตัวไมโครคอนโทรลเลอร์ แล้วนำข้อมูลที่ได้ไปเทียบกับเวลาจริงเมื่อตรงกันก็จะส่งสัญญาณไปยังภาครับและก็จะรับสัญญาณกลับมาเพื่อส่งข้อมูลกลับไปยังคอมพิวเตอร์

ที่ภาครับเมื่อเรารับข้อมูลมาแล้วเราจะแปลงสัญญาณจากอนุกรมเป็นสัญญาณดิจิตอลอีกครั้งโดยใช้ไมโครคอนโทรลเลอร์ ซึ่งจะได้ข้อมูลเป็นสัญญาณดิจิตอล ออกทางพอร์ตของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ และนำลอจิกจากพอร์ตของไมโครคอนโทรลเลอร์ไปควบคุมอุปกรณ์ไฟฟ้าต่างๆโดยตรง และจะส่งสัญญาณกลับไปยังภาคส่งเพื่อบอกให้รู้ว่าได้รับสัญญาณแล้ว

3.1 ส่วนตั้งเวลาเปิด ปิด และเลือกอุปกรณ์

ทำงานโดยรับคำสั่งจากคอมพิวเตอร์ออกทางพอร์ตขนานส่งเข้าสู่ไมโครคอนโทรลเลอร์ โดยคำสั่งนี้จะเป็นข้อมูลขนาด 8 บิต จำนวน 7 ชุด

ชุดที่ 1 เป็นการเลือกอุปกรณ์

ชุดที่ 2 เป็นการส่งค่าวันเปิด

ชุดที่ 3 เป็นการส่งค่าชั่วโมงที่เปิด

ชุดที่ 4 เป็นการส่งค่านาทีที่เปิด

ชุดที่ 5 เป็นการส่งค่าวันปิด

ชุดที่ 6 เป็นการส่งค่าชั่วโมงที่ปิด

ชุดที่ 7 เป็นการส่งค่านาทีที่ปิด

ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของ Microcontroller นั้นเราใช้ภาษาซีในการเขียนโปรแกรม และใช้โปรแกรม Keil ในการสร้างไฟล์ที่เป็นนามสกุล .hex ออกมาแล้วใช้โปรแกรม Flash Magic ในการเบิร์นข้อมูลลงในตัว

3.2 การส่งข้อมูลผ่านพอร์ตขนาน

ลักษณะแรงดันที่ส่งออกมาเป็น TTL โดยสัญญาณลอจิก 1 เท่ากับ 5 โวลต์ และลอจิก 0 จะเท่ากับ 0 โวลต์ ในการการเขียนโปรแกรมจะใช้โพรซีเจอร์ Out จากไฟล์ input 32.dll โดยค่าของ Portnumber จะเท่ากับ H378 สำหรับ LPT1 และ H278 สำหรับ LPT2 ส่วน Data นั้นสามารถกำหนดให้มีค่าอยู่ระหว่าง 0 ถึง 255

การเขียนโปรแกรมเชื่อมไมโครคอนโทรลเลอร์กับพีซีผ่านพอร์ตขนาน

เชื่อมต่อกับพีซีโดยผ่านสายเชื่อมต่อพอร์ตขนาน จากนั้นเขียนโปรแกรมด้วย Visual Basicแล้วสั่งงานส่งข้อมูลที่ต้องการไปยังไมโครคอนโทรลเลอร์ โดยมีตารางการเชื่อมข้อมูลดังนี้

Bit	ชื่อขาไมโครคอนโทรลเลอร์
D0	P1.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

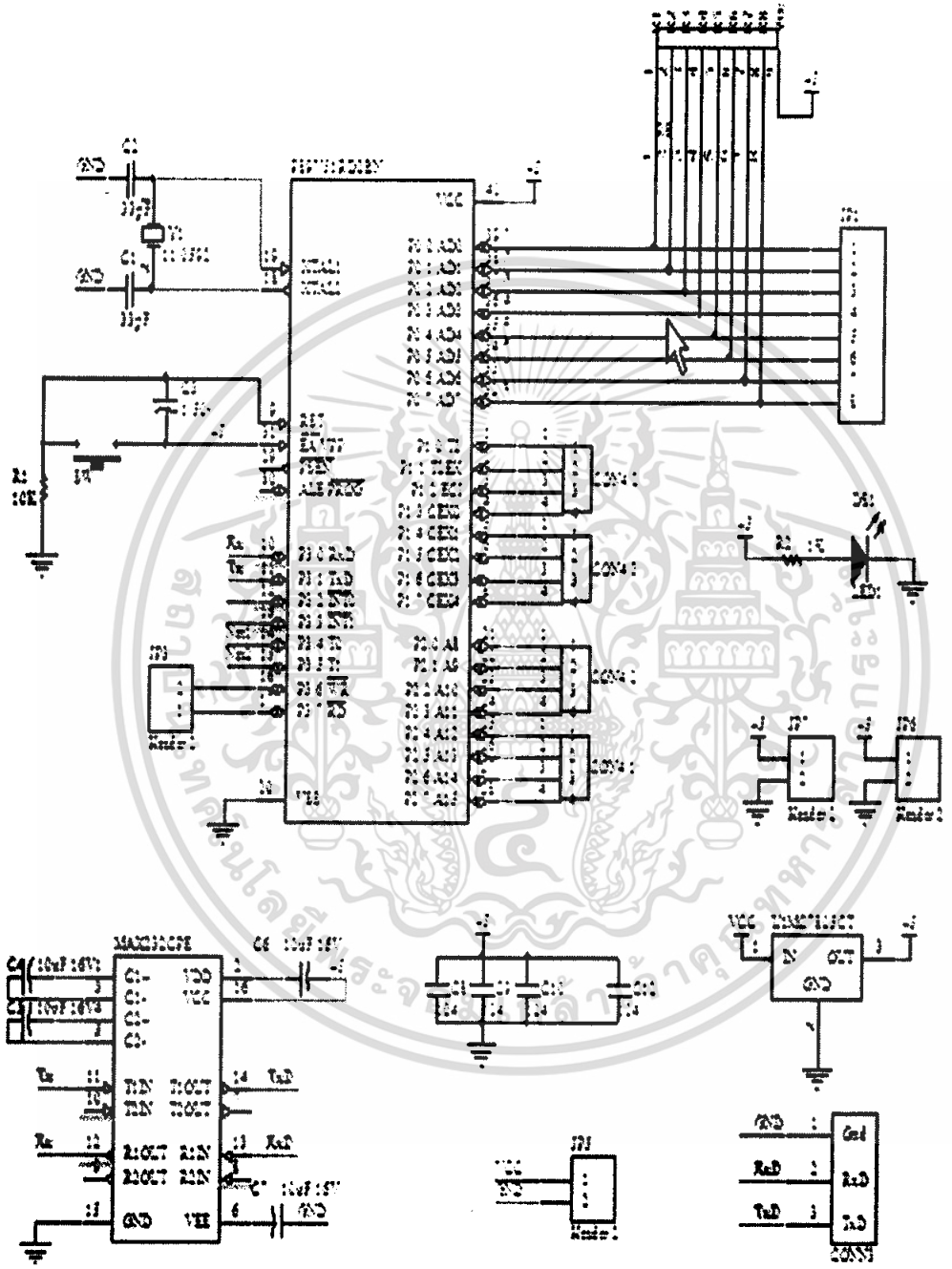
D1	P1.1
D2	P1.2
D3	P1.3
D4	P1.4
D5	P1.5
D6	P1.6
D7	P1.7
S3	P2.0
S4	P2.1
S5	P2.2
S6	P2.3
S7	P2.4
C0	P3.2

3.3 วงจรรีเลย์

โดยรีเลย์ที่ใช้จะเป็นรีเลย์ 12V โดยการทำงานของรีเลย์คือรีเลย์จะทำงานเมื่อมีความต่างศักย์คร่อมรีเลย์เท่ากับ 12V และจะทำงานเมื่ออินพุทที่ขอเบสของทรานซิสเตอร์ที่ค่าเป็น High เนื่องจากกระแส Ib จะมีค่ามากจนทำให้ทรานซิสเตอร์นำกระแสจนอิ่มตัวกระแสจะไหลผ่านรีเลย์ลงกราวด์ได้ความต่างศักย์คร่อมรีเลย์ประมาณ 12V รีเลย์จึงทำงาน ส่วนเมื่ออินพุทเป็น Low ทรานซิสเตอร์จะไม่นำกระแสเนื่องจาก Vbe น้อยกว่า 0.7 ทำให้ความต่างศักย์คร่อมรีเลย์ไม่มีรีเลย์จึงไม่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

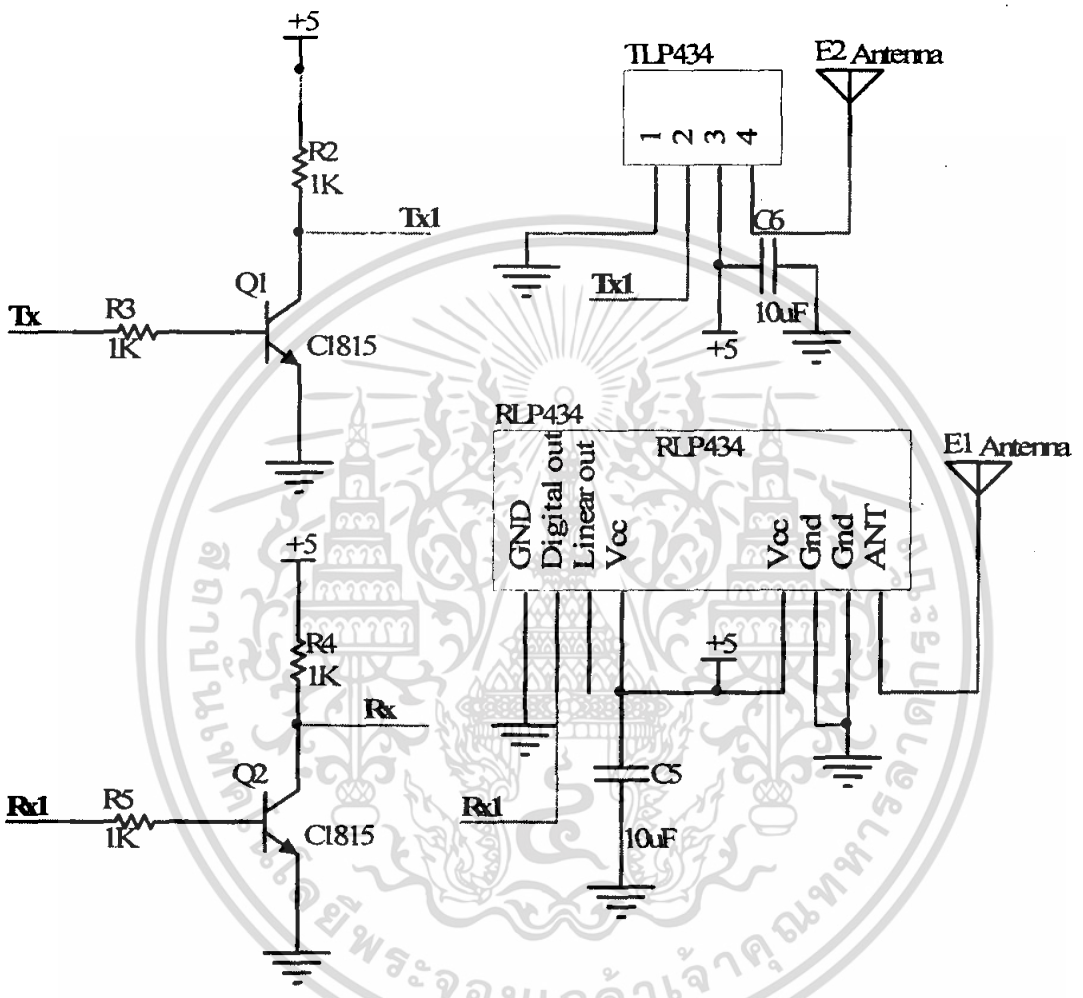
วงจรภาคส่ง



รูปที่ 3.1 วงจรภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

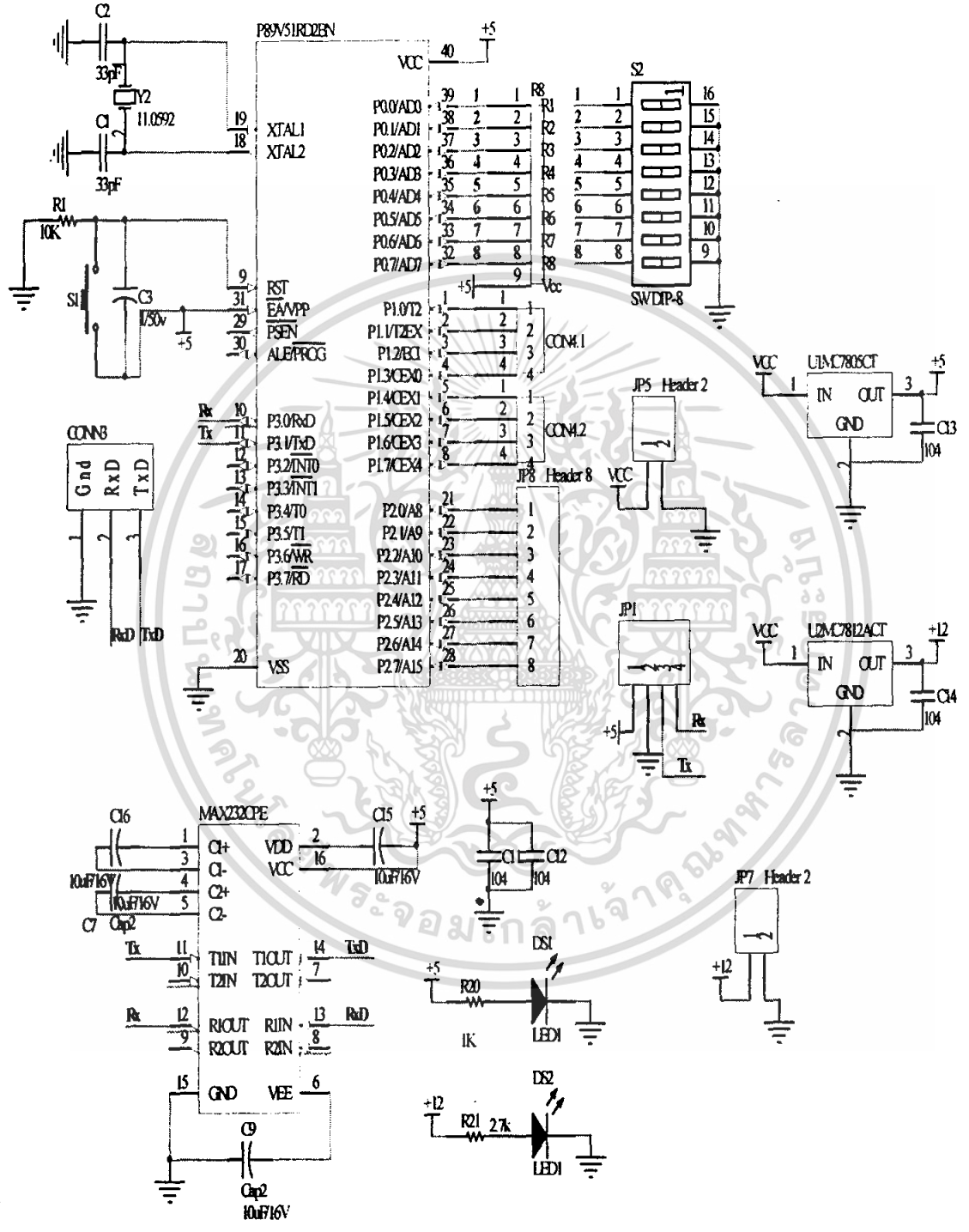
วงจรโมดูลรับ-ส่ง



รูปที่ 3.2 วงจรโมดูลรับ ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

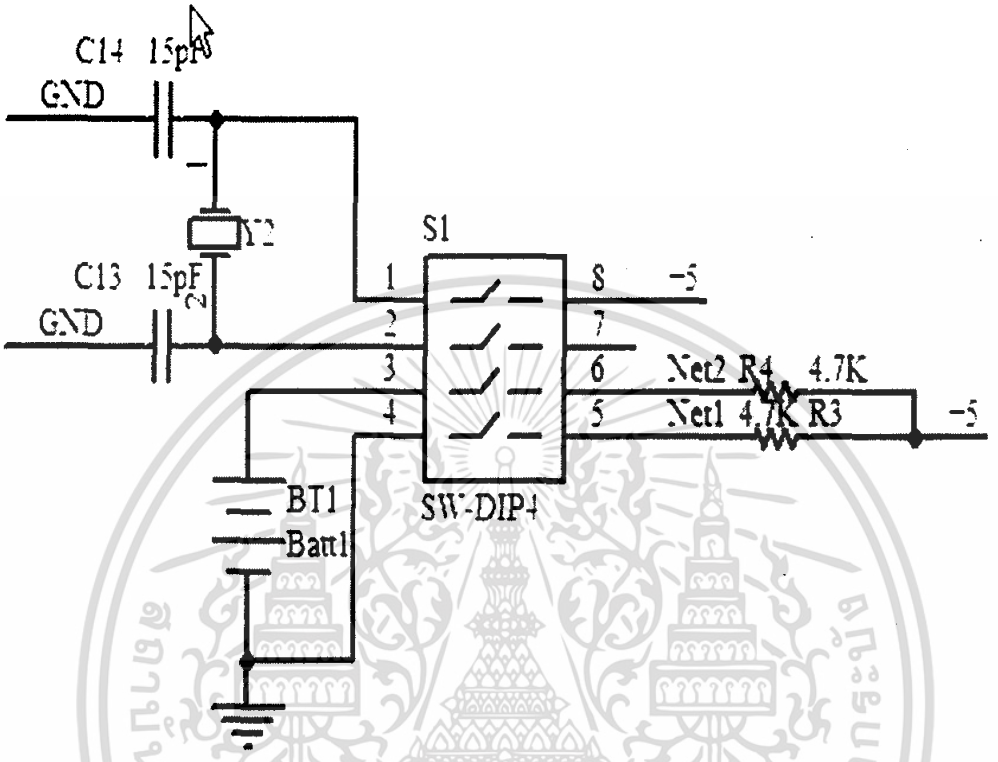
วงจรภาครับ



รูปที่ 3.3 วงจรภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

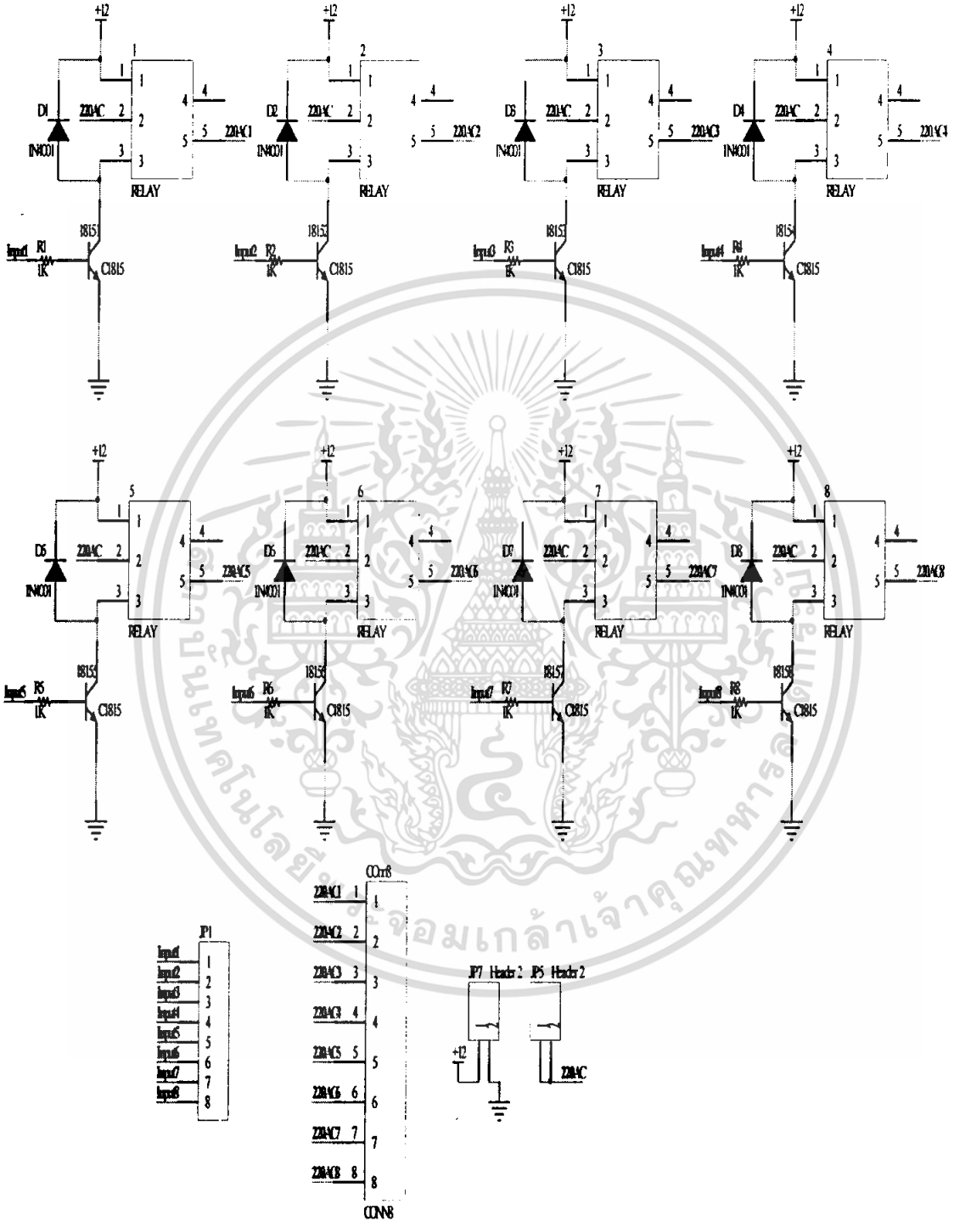
วงจรของฐานเวลา



รูปที่ 3.4 วงจรสร้างฐานเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรRELAYควบคุมหลอดไฟ



รูปที่ 3.5 วงจรRELAYควบคุมหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาคส่ง

```
#include <P89V51Rx2.h>
#include <LCD.H>
#include <I2C.H>
#include <DS1307.H>
#define P_LCD P0
#define P_VB P1
#define P_LED P2
sbit IN1  = P3^2;
sbit DS1  = P3^4;
sbit DS2  = P3^5;
sbit LCD1 = P3^6;
sbit LCD2 = P3^7;
sbit Led0 = P2^0;
sbit Led1 = P2^1;
sbit Led2 = P2^2;
sbit Led3 = P2^3;
sbit Led4 = P2^4;
sbit Led5 = P2^5;
sbit Led6 = P2^6;
sbit Led7 = P2^7;
sbit Vb0 = P1^0;
sbit Vb1 = P1^1;
sbit Vb2 = P1^2;
sbit Vb3 = P1^3;
sbit Vb4 = P1^4;
sbit Vb5 = P1^5;
sbit Vb6 = P1^6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sbit Vb7 = P1^7;
sbit Sw0 = P1^0;
sbit Sw1 = P1^1;
sbit Sw2 = P1^2;
sbit Sw3 = P1^3;
sbit Sw4 = P1^4;
sbit Sw5 = P1^5;
sbit Sw6 = P1^6;
sbit Sw7 = P1^7;
int j,m;
intledd1,ledd2,ledd3,ledd4,ledd5,ledd6,ledd7,ledd8,ledd9,ledd10,ledd11,ledd12,ledd13,ledd14,ledd15
,ledd16;
unsigned char
sec,min,hour,day,date,month,year,control,ans,jay,Gade1,Gade2,Kieng1,Kieng2,Kieng3,Kieng4,dat,da
t2;
unsigned char led_vb,date_vb,open_hour_vb,open_min_vb,date2_vb,close_hour_vb,close_min_vb;
unsigned char Group1_1;
unsigned char Group2_1;
unsigned char
LedLed1,LedLed2,LedLed3,LedLed4,LedLed5,LedLed6,LedLed7,LedLed8,LedLed9,LedLed10,Led
Led11,LedLed12,LedLed13,LedLed14,LedLed15,LedLed16;
unsigned char
LedVb1,LedVb2,LedVb3,LedVb4,LedVb5,LedVb6,LedVb7,LedVb8,LedVb9,LedVb10,LedVb11,Le
dVb12,LedVb13,LedVb14,LedVb15,LedVb16;
xdata unsigned char
dayo1[5],dayo2[5],dayo3[5],dayo4[5],dayo5[5],dayo6[5],dayo7[5],dayo8[5],dayo9[5],dayo10[5],day
o11[5],dayo12[5],dayo13[5],dayo14[5],dayo15[5],dayo16[5];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
xdata unsigned char
dayc1[5],dayc2[5],dayc3[5],dayc4[5],dayc5[5],dayc6[5],dayc7[5],dayc8[5],dayc9[5],dayc10[5],dayc
11[5],dayc12[5],dayc13[5],dayc14[5],dayc15[5],dayc16[5];
xdata unsigned char
oph1[5] ,oph2[5] ,oph3[5] ,oph4[5] ,oph5[5] ,oph6[5] ,oph7[5] ,oph8[5] ,oph9[5] ,oph10[5] ,oph11[5]
,oph12[5] ,oph13[5] ,oph14[5] ,oph15[5] ,oph16[5] ;
xdata unsigned char
opm1[5] ,opm2[5] ,opm3[5] ,opm4[5] ,opm5[5] ,opm6[5] ,opm7[5] ,opm8[5] ,opm9[5] ,opm10[5] ,o
pm11[5] ,opm12[5] ,opm13[5] ,opm14[5] ,opm15[5] ,opm16[5] ;
xdata unsigned char
clh1[5] ,clh2[5] ,clh3[5] ,clh4[5] ,clh5[5] ,clh6[5] ,clh7[5] ,clh8[5] ,clh9[5] ,clh10[5] ,clh11[5] ,clh12[
5] ,clh13[5] ,clh14[5] ,clh15[5] ,clh16[5] ;
xdata unsigned char
clm1[5] ,clm2[5] ,clm3[5] ,clm4[5] ,clm5[5] ,clm6[5] ,clm7[5] ,clm8[5] ,clm9[5] ,clm10[5] ,clm11[5]
,clm12[5] ,clm13[5] ,clm14[5] ,clm15[5] ,clm16[5] ;
void init_serial(void)
{
 TMOD = 0x21;
 SCON = 0x50;
 TH1  = 0xA0;
 TR1  = 1;
 TI   = 1;
 RI   = 0;
 TL1  = 0xA0;
}
void display_datetime(void)
{
 sec  = DS1307_rd(0x00);
 min  = DS1307_rd(0x01);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  hour  = DS1307_rd(0x02);
  day   = DS1307_rd(0x03);                  // Ex: day  = Sun,Mon,Tue,...
  date  = DS1307_rd(0x04);                  // Ex: date = 1,2,3,4,5,6,...
  month = DS1307_rd(0x05);
  year  = DS1307_rd(0x06);
  //------Show Date------//
  lcd_origin();
  lcd_command(0xC0);
  send_to_lcd(date);
  lcd_text('/');
  send_to_lcd(month);
  lcd_text('/');
  send_to_lcd(year);
  //------Show Time------//
  lcd_origin();
  lcd_command(0x80);
  send_to_lcd(hour);
  lcd_text(':');
  send_to_lcd(min);
  lcd_text(':');
  send_to_lcd(sec);
}
void vb_clk(void)
{
  while(Vb7 == 0)
  {}
  while(Vb7 == 1)
  {}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Receive()
{
 led_vb = P1;
 vb_clk();
 date_vb = P1;
 vb_clk();
 open_hour_vb = P1;
 vb_clk();
 open_min_vb = P1;
 vb_clk();
 date2_vb = P1;
 vb_clk();
 close_hour_vb = P1;
 vb_clk();
 close_min_vb = P1;
}
void Receive2(unsigned char le,unsigned char da,unsigned char op_hr,unsigned char op_mi,unsigned char da2,unsigned char cl_hr,unsigned char cl_mi)
{
 if(le == 0x01)
 {
        dayo1[ledd1] = da;
        oph1[ledd1]  = op_hr;
        opm1[ledd1]  = op_mi;
        dayc1[ledd1] = da2;
        clh1[ledd1]  = cl_hr;
        clm1[ledd1]  = cl_mi;
        ledd1++;
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(le == 0x02)
{
        dayo2[ledd2] = da;
        oph2[ledd2]  = op_hr;
        opm2[ledd2]  = op_mi;
        dayc2[ledd2] = da2;
        clh2[ledd2]  = cl_hr;
        clm2[ledd2]  = cl_mi;
        ledd2++;
}
if(le == 0x03)
{
        dayo3[ledd3] = da;
        oph3[ledd3]  = op_hr;
        opm3[ledd3]  = op_mi;
        dayc3[ledd3] = da2;
        clh3[ledd3]  = cl_hr;
        clm3[ledd3]  = cl_mi;
        ledd3++;
}
if(le == 0x04)
{
        dayo4[ledd4] = da;
        oph4[ledd4]  = op_hr;
        opm4[ledd4]  = op_mi;
        dayc4[ledd4] = da2;
        clh4[ledd4]  = cl_hr;
        clm4[ledd4]  = cl_mi;
        ledd4++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
if(le == 0x05)
{
        dayo5[ledd5] = da;
        oph5[ledd5]  = op_hr;
        opm5[ledd5]  = op_mi;
        dayc5[ledd5] = da2;
        clh5[ledd5]  = cl_hr;
        clm5[ledd5]  = cl_mi;
        ledd5++;
}
if(le == 0x06)
{
        dayo6[ledd6] = da;
        oph6[ledd6]  = op_hr;
        opm6[ledd6]  = op_mi;
        dayc6[ledd6] = da2;
        clh6[ledd6]  = cl_hr;
        clm6[ledd6]  = cl_mi;
        ledd6++;
}
if(le == 0x07)
{
        dayo7[ledd7] = da;
        oph7[ledd7]  = op_hr;
        opm7[ledd7]  = op_mi;
        dayc7[ledd7] = da2;
        clh7[ledd7]  = cl_hr;
        clm7[ledd7]  = cl_mi;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ledd7++;
}
if(le == 0x08)
{
        dayo8[ledd8] = da;
        oph8[ledd8]  = op_hr;
        opm8[ledd8]  = op_mi;
        dayc8[ledd8] = da2;
        clh8[ledd8]  = cl_hr;
        clm8[ledd8]  = cl_mi;
        ledd8++;
}
if(le == 0x09)
{
        dayo9[ledd9] = da;
        oph9[ledd9]  = op_hr;
        opm9[ledd9]  = op_mi;
        dayc9[ledd9] = da2;
        clh9[ledd9]  = cl_hr;
        clm9[ledd9]  = cl_mi;
        ledd9++;
}
if(le == 0x10)
{
        dayo10[ledd10] = da;
        oph10[ledd10]  = op_hr;
        opm10[ledd10]  = op_mi;
        dayc10[ledd10] = da2;
        clh10[ledd10]  = cl_hr;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        clm10[ledd10] = cl_mi;
        ledd10++;
}
if(le == 0x11)
{
        dayo11[ledd11] = da;
        oph11[ledd11] = op_hr;
        opm11[ledd11] = op_mi;
        dayc11[ledd11] = da2;
        clh11[ledd11] = cl_hr;
        clm11[ledd11] = cl_mi;
        ledd11++;
}
if(le == 0x12)
{
        dayo12[ledd12] = da;
        oph12[ledd12] = op_hr;
        opm12[ledd12] = op_mi;
        dayc12[ledd12] = da2;
        clh12[ledd12] = cl_hr;
        clm12[ledd12] = cl_mi;
        ledd12++;
}
if(le == 0x13)
{
        dayo13[ledd13] = da;
        oph13[ledd13] = op_hr;
        opm13[ledd13] = op_mi;
        dayc13[ledd13] = da2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        clh13[ledd13]  = cl_hr;
        clm13[ledd13]  = cl_mi;
        ledd13++;
}
if(le == 0x14)
{
        dayo14[ledd14] = da;
        oph14[ledd14]  = op_hr;
        opm14[ledd14]  = op_mi;
        dayc14[ledd14] = da2;
        clh14[ledd14]  = cl_hr;
        clm14[ledd14]  = cl_mi;
        ledd14++;
}
if(le == 0x15)
{
        dayo15[ledd15] = da;
        oph15[ledd15]  = op_hr;
        opm15[ledd15]  = op_mi;
        dayc15[ledd15] = da2;
        clh15[ledd15]  = cl_hr;
        clm15[ledd15]  = cl_mi;
        ledd15++;
}
if(le == 0x16)
{
        dayo16[ledd16] = da;
        oph16[ledd16]  = op_hr;
        opm16[ledd16]  = op_mi;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dayc16[ledd16] = da2;
        clh16[ledd16]  = cl_hr;
        clm16[ledd16]  = cl_mi;
        ledd16++;
 }
}

void Send_To_51()
{
  for(m=0;m<3;m++)
  {
        SBUF = 0x11;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
       }
       for(m=0;m<3;m++)
  {
        SBUF = 0x22;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
       }
       for(m=0;m<3;m++)
  {
        SBUF = 0x44;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
       }
       for(m=0;m<1;m++)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SBUF = Group1_1;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
       }
}

void Send_To_51_2()
{
  for(m=0;m<3;m++)
  {
        SBUF = 0x12;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
       }
      for(m=0;m<3;m++)
  {
        SBUF = 0x23;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
      }
      for(m=0;m<3;m++)
  {
        SBUF = 0x45;
        while(~TI);                 // Wait until data send to serial port finish
        TI = 0;                     // Clear TI flag
      }
      for(m=0;m<1;m++)
  {
        SBUF = Group2_1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            while(~TI);                 // Wait until data send to serial port finish
            TI = 0;                     // Clear TI flag
          }
}
void Convert()
{
 Group1_1 =
((LedLed1)|(LedLed2)|(LedLed3)|(LedLed4)|(LedLed5)|(LedLed6)|(LedLed7)|(LedLed8));
 Group2_1 =
((LedLed9)|(LedLed10)|(LedLed11)|(LedLed12)|(LedLed13)|(LedLed14)|(LedLed15)|(LedLed16));
}
void Display()
{
 for(j=0;j<ledd1;j++)
 {
  if(dayo1[j] == day)
  {
        if((oph1[j] == hour)&&(opm1[j] == min))
        {
              LedLed1 = 0x01;
              LedVb1 = 0x11;
              jay = 1;
        }
  }
  if(dayc1[j] == day)
  {
        if((clh1[j] == hour)&&(clm1[j] == min))
        {
              LedLed1 = 0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LedVb1 = 0x01;
                jay = 1;
        }
 }
}
for(j=0;j<ledd2;j++)
{
 if(dayo2[j] == day)
 {
        if((oph2[j] == hour)&&(opm2[j] == min))
        {
                LedLed2 = 0x02;
                LedVb2 = 0x12;
                        jay = 1;
        }
 }
 if(dayc2[j] == day)
 {
        if((clh2[j] == hour)&&(clm2[j] == min))
        {
                LedLed2 = 0x00;
                LedVb2 = 0x02;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd3;j++)
{
 if(dayo3[j] == day)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 {
        if((oph3[j] == hour)&&(opm3[j] == min))
        {
                LedLed3 = 0x04;
                LedVb3 = 0x13;
                jay = 1;
        }
 }
 if(dayc3[j] == day)
 {
        if((clh3[j] == hour)&&(clm3[j] == min))
        {
                LedLed3 = 0x00;
                LedVb3 = 0x03;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd4;j++)
{
 if(dayo4[j] == day)
 {
        if((oph4[j] == hour)&&(opm4[j] == min))
        {
                LedLed4 = 0x08;
                LedVb4 = 0x14;
                        jay = 1;
        }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if(dayc4[j] == day)
 {
        if((clh4[j] == hour)&&(clm4[j] == min))
        {
              LedLed4 = 0x00;
              LedVb4 = 0x04;
                    jay = 1;
        }
 }
}
for(j=0;j<ledd5;j++)
{
 if(dayo5[j] == day)
 {
        if((oph5[j] == hour)&&(opm5[j] == min))
        {
              LedLed5 = 0x10;
              LedVb5 = 0x15;
                    jay = 1;
        }
 }
 if(dayc5[j] == day)
 {
        if((clh5[j] == hour)&&(clm5[j] == min))
        {
              LedLed5 = 0x00;
              LedVb5 = 0x05;
                    jay = 1;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
}
for(j=0;j<ledd6;j++)
{
 if(dayo6[j] == day)
 {
        if((oph6[j] == hour)&&(opm6[j] == min))
        {
                LedLed6 = 0x20;
                LedVb6 = 0x16;
                        jay = 1;
        }
 }
 if(dayc6[j] == day)
 {
        if((clh6[j] == hour)&&(clm6[j] == min))
        {
                LedLed6 = 0x00;
                LedVb6 = 0x06;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd7;j++)
{
 if(dayo7[j] == day)
 {
        if((oph7[j] == hour)&&(opm7[j] == min))
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LedLed7 = 0x40;
                LedVb7 = 0x17;
                        jay = 1;
        }
 }
 if(dayc7[j] == day)
 {
        if((clh7[j] == hour)&&(clm7[j] == min))
        {
                LedLed7 = 0x00;
                LedVb7 = 0x07;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd8;j++)
{
 if(dayo8[j] == day)
 {
        if((oph8[j] == hour)&&(opm8[j] == min))
        {
                LedLed8 = 0x80;
                LedVb8 = 0x18;
                        jay = 1;
        }
 }
 if(dayc8[j] == day)
 {
        if((clh8[j] == hour)&&(clm8[j] == min))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                LedLed8 = 0x00;
                LedVb8 = 0x08;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd9;j++)
{
 if(dayo9[j] == day)
 {
        if((oph9[j] == hour)&&(opm9[j] == min))
        {
                LedLed9 = 0x01;
                LedVb9 = 0x19;
                        jay = 1;
        }
 }
 if(dayc9[j] == day)
 {
        if((clh9[j] == hour)&&(clm9[j] == min))
        {
                LedLed9 = 0x00;
                LedVb9 = 0x09;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd10;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
 if(dayo10[j] == day)
 {
        if((oph10[j] == hour)&&(opm10[j] == min))
        {
                LedLed10 = 0x02;
                LedVb10 = 0x1A;
                        jay = 1;
        }
 }
 if(dayc10[j] == day)
 {
        if((clh10[j] == hour)&&(clm10[j] == min))
        {
                LedLed10 = 0x00;
                LedVb10 = 0x0A;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd11;j++)
{
 if(dayo11[j] == day)
 {
        if((oph11[j] == hour)&&(opm11[j] == min))
        {
                LedLed11 = 0x04;
                LedVb11 = 0x1B;
                        jay = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    }
    if(dayc11[j] == day)
    {
        if((clh11[j] == hour)&&(clm11[j] == min))
        {
            LedLed11 = 0x00;
            LedVb11 = 0x0B;
                jay = 1;
        }
    }
}
for(j=0;j<ledd12;j++)
{
    if(dayo12[j] == day)
    {
        if((oph12[j] == hour)&&(opm12[j] == min))
        {
            LedLed12 = 0x08;
            LedVb12 = 0x1C;
                jay = 1;
        }
    }
    if(dayc12[j] == day)
    {
        if((clh12[j] == hour)&&(clm12[j] == min))
        {
            LedLed12 = 0x00;
            LedVb12 = 0x0C;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    jay = 1;
            }
     }
    }
    for(j=0;j<ledd13;j++)
    {
     if(dayo13[j] == day)
     {
            if((oph13[j] == hour)&&(opm13[j] == min))
            {
                    LedLed13 = 0x10;
                    LedVb13 = 0x1D;
                        jay = 1;
            }
     }
     if(dayc13[j] == day)
     {
            if((clh13[j] == hour)&&(clm13[j] == min))
            {
                    LedLed13 = 0x00;
                    LedVb13 = 0x0D;
                        jay = 1;
            }
     }
    }
    for(j=0;j<ledd14;j++)
    {
     if(dayo14[j] == day)
     {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if((oph14[j] == hour)&&(opm14[j] == min))
        {
                LedLed14 = 0x20;
                LedVb14 = 0x1E;
                        jay = 1;
        }
 }
 if(dayc14[j] == day)
 {
        if((clh14[j] == hour)&&(clm14[j] == min))
        {
                LedLed14 = 0x00;
                LedVb14 = 0x0E;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd15;j++)
{
 if(dayo15[j] == day)
 {
        if((oph15[j] == hour)&&(opm15[j] == min))
        {
                LedLed15 = 0x40;
                LedVb15 = 0x1F;
                        jay = 1;
        }
 }
 if(dayc15[j] == day)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 {
        if((clh15[j] == hour)&&(clm15[j] == min))
        {
                LedLed15 = 0x00;
                LedVb15 = 0x0F;
                        jay = 1;
        }
 }
}
for(j=0;j<ledd16;j++)
{
 if(dayo16[j] == day)
 {
        if((oph16[j] == hour)&&(opm16[j] == min))
        {
                LedLed16 = 0x80;
                LedVb16 = 0x10;
                        jay = 1;
        }
 }
 if(dayc16[j] == day)
 {
        if((clh16[j] == hour)&&(clm16[j] == min))
        {
                LedLed16 = 0x00;
                LedVb16 = 0x00;
                        jay = 1;
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  Convert();
}
void Vb_Led()
{
 if(sec == 0x10)
 {
  P2 = LedVb1;
 }
 if(sec == 0x11)
 {
  P2 = LedVb2;
 }
 if(sec == 0x12)
 {
  P2 = LedVb3;
 }
 if(sec == 0x13)
 {
  P2 = LedVb4;
 }
 if(sec == 0x14)
 {
  P2 = LedVb5;
 }
 if(sec == 0x15)
 {
  P2 = LedVb6;
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(sec == 0x16)
{
 P2 = LedVb7;
}
if(sec == 0x17)
{
 P2 = LedVb8;
}
if(sec == 0x18)
{
 P2 = LedVb9;
}
if(sec == 0x19)
{
 P2 = LedVb10;
}
if(sec == 0x20)
{
 P2 = LedVb11;
}
if(sec == 0x21)
{
 P2 = LedVb12;
}
if(sec == 0x22)
{
 P2 = LedVb13;
}
if(sec == 0x23)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
   P2 = LedVb14;
  }
  if(sec == 0x24)
  {
   P2 = LedVb15;
  }
  if(sec == 0x25)
  {
   P2 = LedVb16;
  }
}
void init_interrupt()
{
 EX0 = 1;          //Enable External 0
 IT0 = 0;          //Interrupt Type : Falling Edge
 EA = 1;           //Enable Interrupt All (a must have one!)
}
void VB() interrupt 0  //P3_2
{
 while(Vb7 == 1)
 {}
 Receive();
 while(Vb7 == 0)
 {}
 Receive2(led_vb,date_vb,open_hour_vb,open_min_vb,date2_vb,close_hour_vb,close_min_vb);
}
void main()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//------------------------------C 104 = 0.1uF Vcc+Gnd-----------------------------------//
P0 = 0x00;              // LCD
P1 = 0xFF;              // VB
P2 = 0x00;              // Led   P2 = 0xFF;
IN1 = 0;                        // (P3^2 Interrupt0)  (P3^4,P3^5 Control DS1307)  (P3^6,P3^7
Control LCD)
DS1 = 0;
DS2 = 0;
LCD1 = 0;
LCD2 = 0;
lcd_init();             // Initial LCD
init_serial();    // Initial TxRx
init_interrupt(); // Initial Interupt
ledd1 = 0;
ledd2 = 0;
Group1_1 = 0;
Group2_1 = 0;
DS1307_wrtime(0x12,0x45,0x53);
DS1307_wrdate(0x04,0x12,0x03,0x08);
while(1)
{
        Display();

if((sec==0x00)|(sec==0x01)|(sec==0x10)|(sec==0x11)|(sec==0x20)|(sec==0x21)|(sec==0x30)|(sec==
0x31)|(sec==0x40)|(sec==0x41)|(sec==0x50)|(sec==0x51))
        {
                        dat = Group1_1;
                        for(m=0;m<1;m++)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SBUF = 0x11;            //0001 0001
                while(!TI);
                TI=0;
                }
                for(m=0;m<1;m++)
        {
                SBUF = 0x22;            //0010 0010
                while(!TI);
                TI=0;
                }
                for(m=0;m<1;m++)
        {
                SBUF = 0x44;            //0100 0100
                while(!TI);
                TI=0;
                }
                for(m=0;m<1;m++)
        {
                SBUF = dat;
                while(~TI);                     // Wait until data send to serial
port finish
                TI = 0;                         // Clear TI flag
                }
        }
if((sec==0x05)|(sec==0x06)|(sec==0x15)|(sec==0x16)|(sec==0x25)|(sec==0x26)|(sec==0x35)|(sec==
0x36)|(sec==0x45)|(sec==0x46)|(sec==0x55)|(sec==0x56))
        {
                dat2 = Group2_1;
                for(m=0;m<1;m++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                    SBUF = 0x41;            //0001 0001
                    while(!TI);
                    TI=0;
                    }
                    for(m=0;m<1;m++)
                {
                    SBUF = 0x31;            //0010 0010
                    while(!TI);
                    TI=0;
                    }
                    for(m=0;m<1;m++)
                {
                    SBUF = 0x21;            //0100 0100
                    while(!TI);
                    TI=0;
                    }
                    for(m=0;m<1;m++)
                    {
                    SBUF = dat2;
                    while(~TI);                     // Wait until data send to serial
port finish
                    TI = 0;                         // Clear TI flag
                    }
            }
  display_datetime();
      Kieng1 = ((Group1_1)&(0x0F));                                  // .... xxxx
Gade1 Led1-4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Kieng2 = (((Group1_1)&(0xF0))>>4);                    // xxxx ....
Gade1 Led5-8
        Kieng3 = ((Group2_1)&(0x0F));                         // .... xxxx
Gade2 Led9-12
        Kieng4 = (((Group2_1)&(0xF0))>>4);                    // xxxx ....
Gade2 Led13-16
        LedVb1  = (((Kieng1)&(0x01))<<4)|(0x01);              // 0001 0000
        LedVb2  = (((Kieng1)&(0x02))<<3)|(0x02);              // 0001 0001
        LedVb3  = (((Kieng1)&(0x04))<<2)|(0x03);              // 0001 0010
        LedVb4  = (((Kieng1)&(0x08))<<1)|(0x04);              // 0001 0011
        LedVb5  = (((Kieng2)&(0x01))<<4)|(0x05);              // 0001 0100
        LedVb6  = (((Kieng2)&(0x02))<<3)|(0x06);
        LedVb7  = (((Kieng2)&(0x04))<<2)|(0x07);
        LedVb8  = (((Kieng2)&(0x08))<<1)|(0x08);
        LedVb9  = (((Kieng3)&(0x01))<<4)|(0x09);
        LedVb10 = (((Kieng3)&(0x02))<<3)|(0x0A);
        LedVb11 = (((Kieng3)&(0x04))<<2)|(0x0B);
        LedVb12 = (((Kieng3)&(0x08))<<1)|(0x0C);
        LedVb13 = (((Kieng4)&(0x01))<<4)|(0x0D);
        LedVb14 = (((Kieng4)&(0x02))<<3)|(0x0E);
        LedVb15 = (((Kieng4)&(0x04))<<2)|(0x0F);
        LedVb16 = (((Kieng4)&(0x08))<<1)|(0x00);
        Vb_Led();
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาครับตัวที่ 1

```
#include <P89V51Rx2.h>                    // Header file register for P89V51RD2BN
sbit Sw0 = P1^0;
sbit Sw1 = P1^1;
sbit Sw2 = P1^2;
sbit Sw3 = P1^3;
sbit Sw4 = P1^4;
sbit Sw5 = P1^5;
sbit Sw6 = P1^6;
sbit Sw7 = P1^7;
unsigned char ans,dat,m,n,x,r,r2,rec,j;
 void delay(int time)
{
        int i,j;                          // For keep counter loop
        for(i=0;i<time;i++)               // Loop delay
        for(j=0;j<200;j++);
}
void Receive()
{
  while(!RI);
        RI=0;
        rec = SBUF;
}
void main(void)
{
        unsigned char i=0;
        TMOD = 0x21;            // Timer1 Mode2(8 bit auto reload) for serial port
        SCON = 0x50;            // Mode serial port TX/RX data
        TH1 = 0xA0;                     // Set 300 bps Timer1 default
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TL1 = 0xA0;
        RI = 0;                                 // Clear RI flag
        TI = 0;                 // Clear TI flag
        //EA = 1;                                       // Enable interrupt all
        //ES0 = 1;              // Enable interrupt serial port
        TR1 = 1;                                // Start Timer1    c xc
        P2 = 0x00;
while(1)
{
  while(!RI);
        RI=0;
        ans = SBUF;
        switch (ans)
          {
                case 0x11:
                {
                 while(ans != 0x22)
                 {
                  while(!RI);
                        RI=0;
                        ans = SBUF;
                }
                while(ans == 0x22)
                {
                        while(!RI);
                        RI=0;
                        ans = SBUF;
                }
                while(ans != 0x44)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                while(!RI);
                RI=0;
                ans = SBUF;
            }
            while(ans == 0x44)
            {
                while(!RI);
                RI=0;
                ans = SBUF;
                x = ans;
                P2 = x;
            }
        break;
        }
        default :
        break;
    }
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาครับตัวที่ 2

```
#include <P89V51Rx2.h>                    // Header file register for P89V51RD2BN
sbit Sw0 = P1^0;
sbit Sw1 = P1^1;
sbit Sw2 = P1^2;
sbit Sw3 = P1^3;
sbit Sw4 = P1^4;
sbit Sw5 = P1^5;
sbit Sw6 = P1^6;
sbit Sw7 = P1^7;
unsigned char ans,dat,m,n,x,r,r2,rec,j;
 void delay(int time)
{
        int i,j;                          // For keep counter loop
        for(i=0;i<time;i++)               // Loop delay
        for(j=0;j<200;j++);
}
void Receive()
{
  while(!RI);
        RI=0;
        rec = SBUF;
}
void main(void)
{
        unsigned char i=0;
        TMOD = 0x21;              // Timer1 Mode2(8 bit auto reload) for serial port
        SCON = 0x50;              // Mode serial port TX/RX data
        TH1 = 0xA0;                       // Set 300 bps Timer1 default
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TL1 = 0xA0;
        RI = 0;                                 // Clear RI flag
        TI = 0;                 // Clear TI flag
        //EA = 1;                                       // Enable interrupt all
        //ES0 = 1;              // Enable interrupt serial port
        TR1 = 1;                                // Start Timer1    c xc
        P2 = 0x00;
while(1)
{
  while(!RI);
        RI=0;
        ans = SBUF;
        switch (ans)
          {
                case 0x41:
                {
                while(ans != 0x31)
                {
                 while(!RI);
                        RI=0;
                        ans = SBUF;
                }
                while(ans == 0x31)
                {
                        while(!RI);
                        RI=0;
                        ans = SBUF;
                }
                while(ans != 0x21)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                while(!RI);
                RI=0;
                ans = SBUF;
            }
            while(ans == 0x21)
            {
                while(!RI);
                RI=0;
                ans = SBUF;
                x = ans;
                P2 = x;
            }
        break;
        }
        default :
        break;
    }
}
```

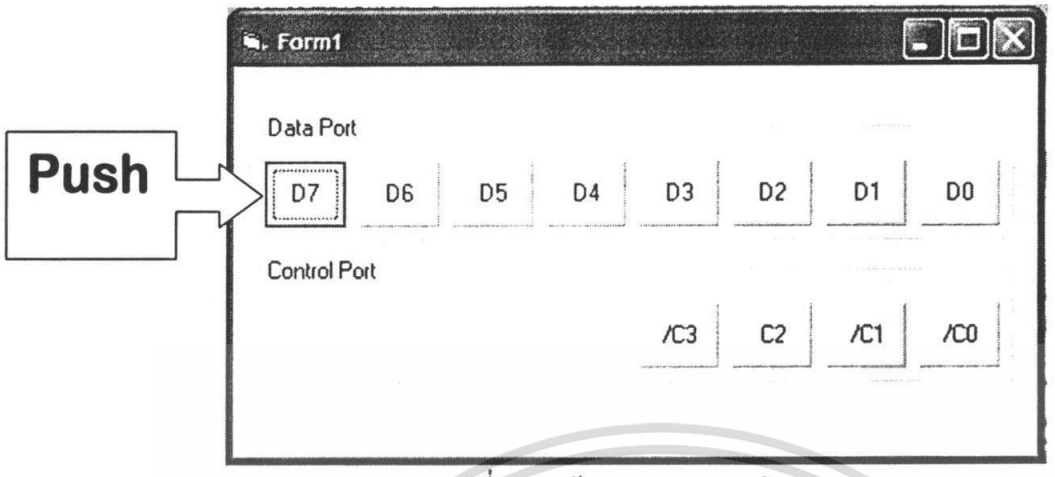
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลอง

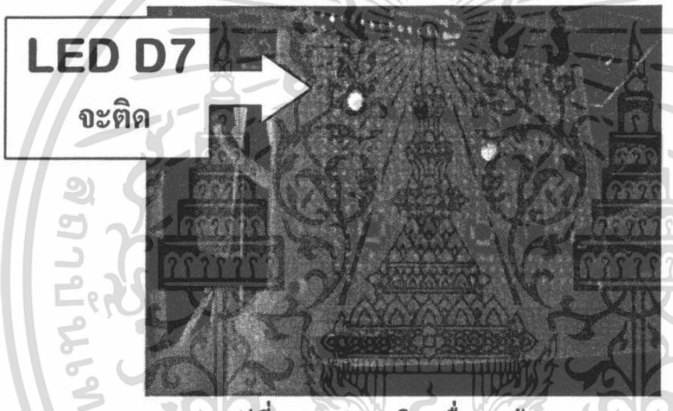
4.1 การเชื่อมต่อบอร์ดกับพีซี

นำบอร์ดทดลองมาเชื่อมต่อกับพีซี ผ่านสายเชื่อมต่อพอร์ตขนาน จากนั้นเขียนโปรแกรมด้วย Visual Basic แล้วสั่งงานการแสดงผล LED ที่บอร์ดทดลอง ในการการเขียนโปรแกรมจะใช้โพรซีเจอร์ Out จากไฟล์ input 32.dll โดยค่าของ Portnumber จะเท่ากับ H378 สำหรับ LPT1 และ H278 สำหรับ LPT2 ส่วน Data นั้นสามารถกำหนดให้มีค่าอยู่ระหว่าง 0 ถึง 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

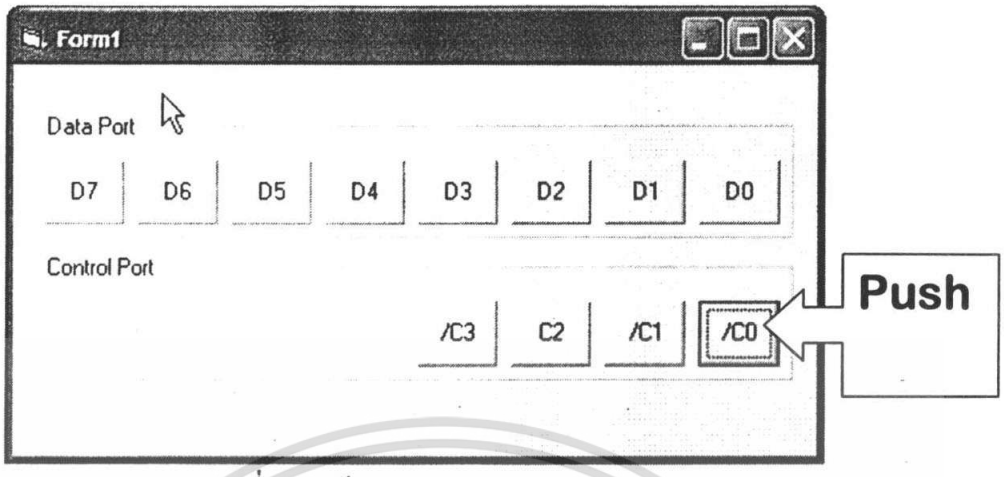


รูปที่ 4.1 หน้าจอVB ขณะกดปุ่ม D7

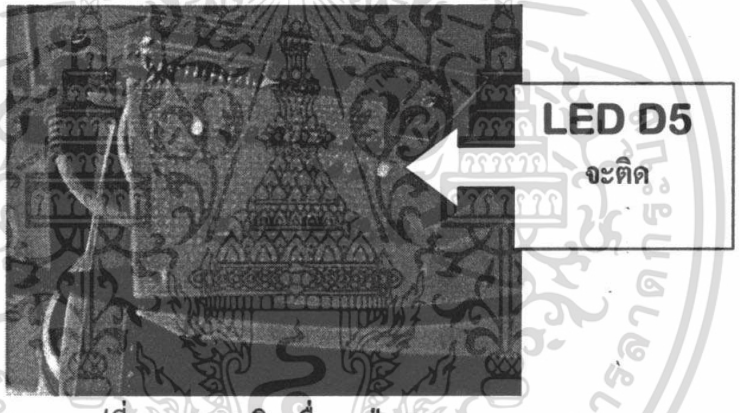


รูปที่ 4.2 LED7 ติด เมื่อกดปุ่ม D7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 หน้าจอVB ขณะกดปุ่ม /CO



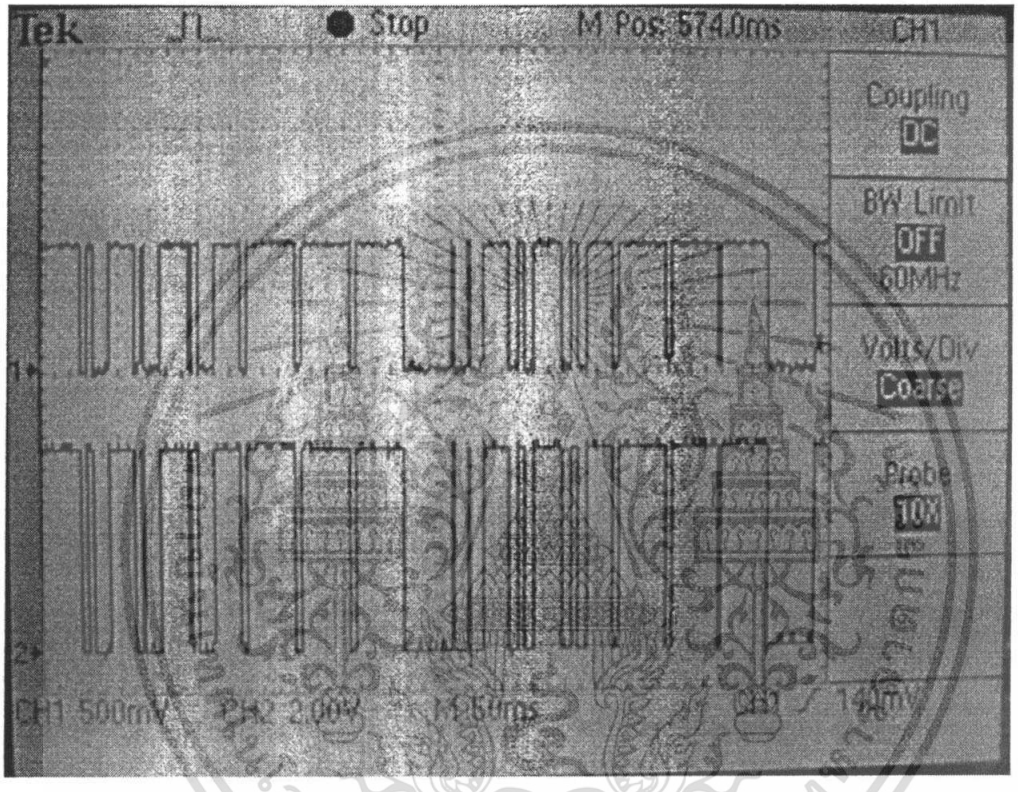
รูปที่ 4.4 LED7 ติด เมื่อกดปุ่ม D5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

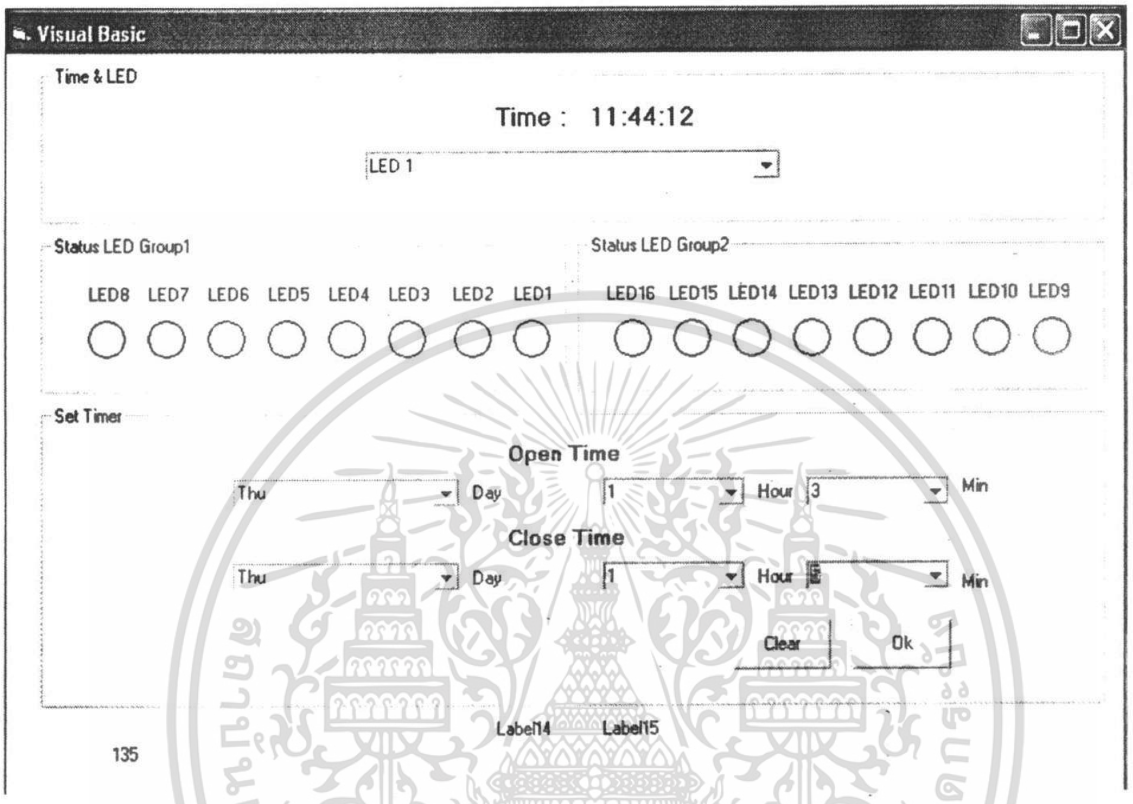
1. กราฟแสดงผลเมื่อไม่ได้ป้อนข้อมูล



รูปที่ 5.1 กราฟแสดงผลเมื่อไม่ได้ป้อนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

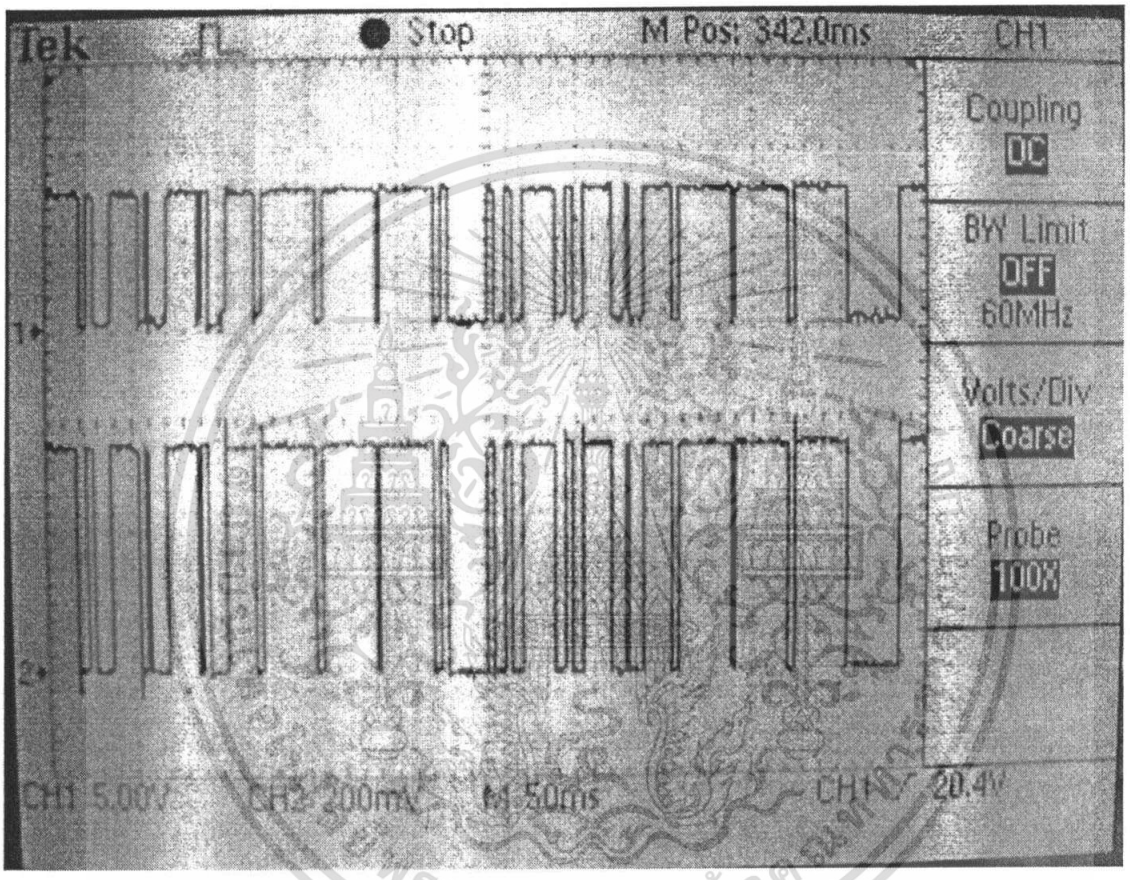
แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่ 5.2 รูปแสดงผลเมื่อไม่ได้ป้อนข้อมูลออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

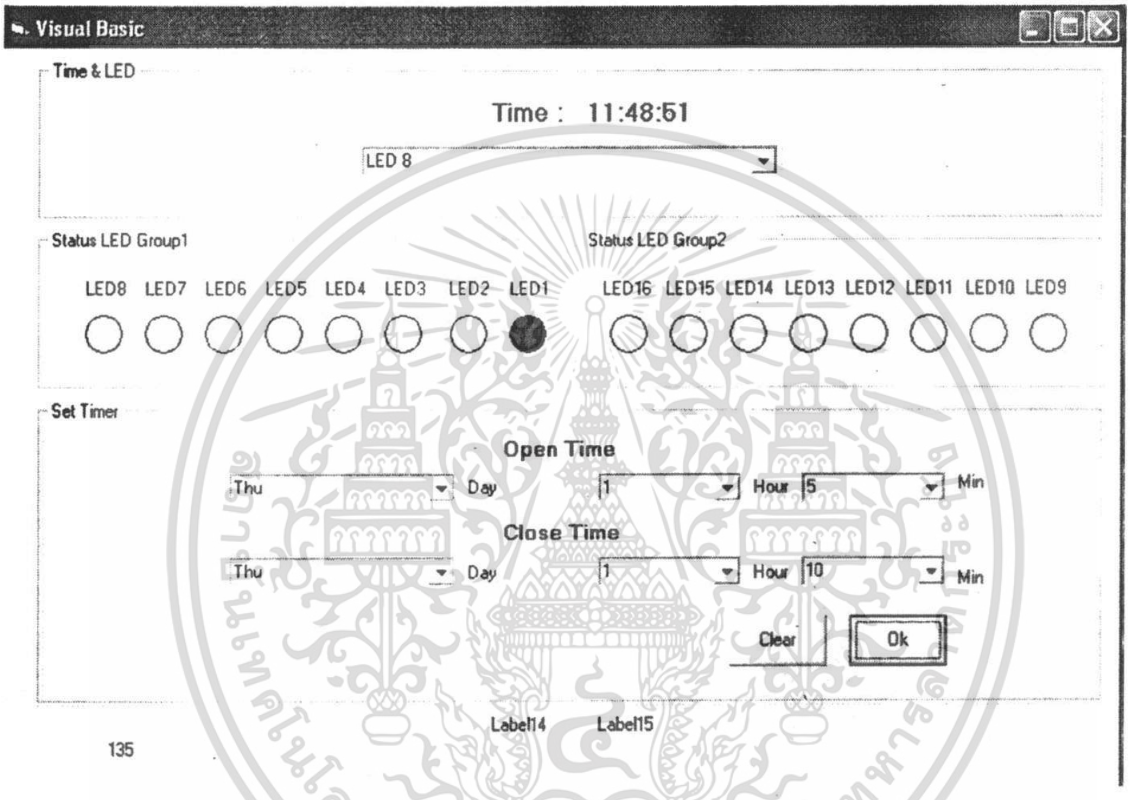
2. กราฟแสดงผลไปควบคุมLED1



รูปที่ 5.3 กราฟแสดงผลไปควบคุมLED1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

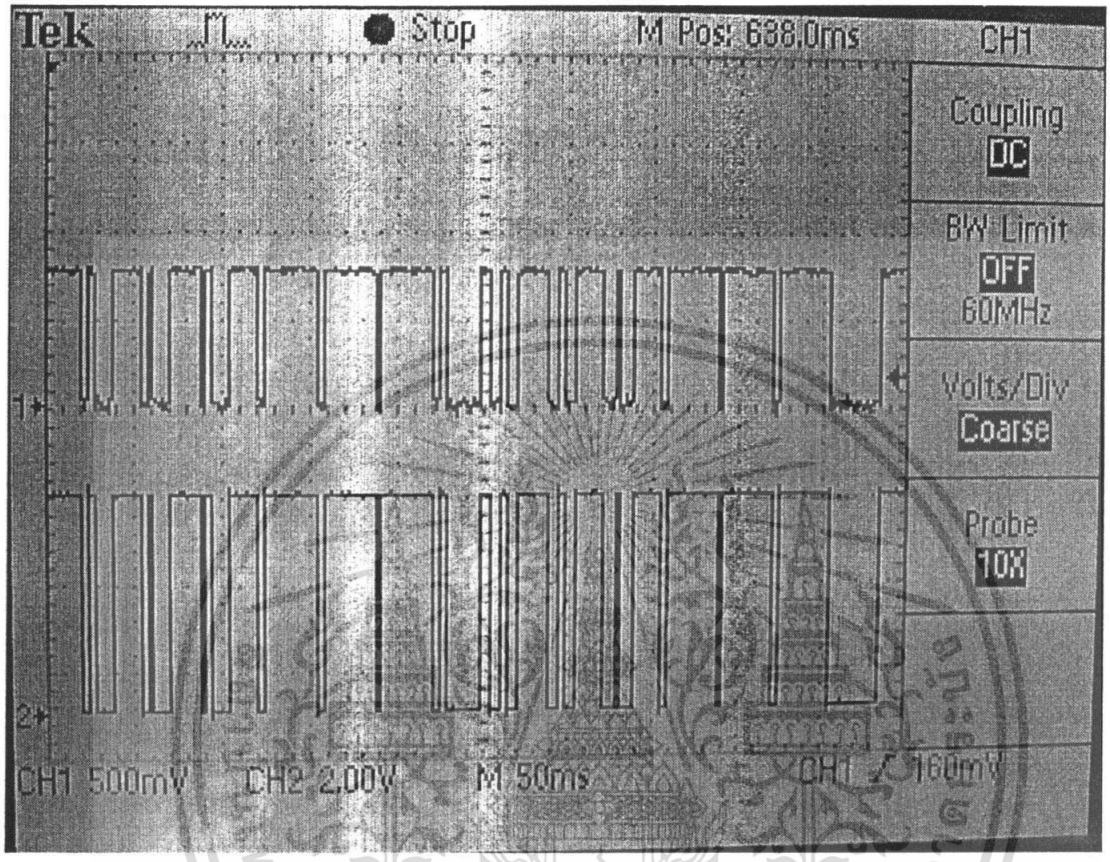
แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่ 5.4 รูปแสดงผลไปควบคุมLED1ออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

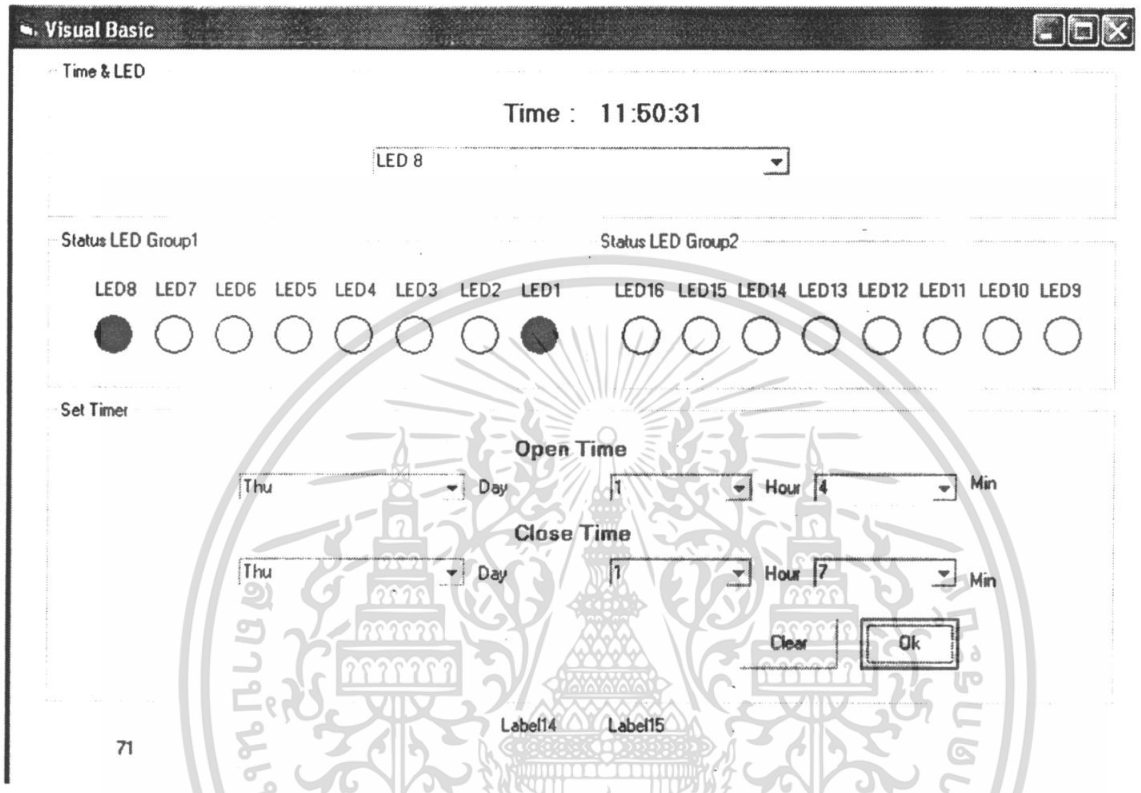
3. กราฟแสดงผลไปควบคุมLED1 และ LED8



รูปที่ 5.5 กราฟแสดงผลไปควบคุมLED1 และ LED8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

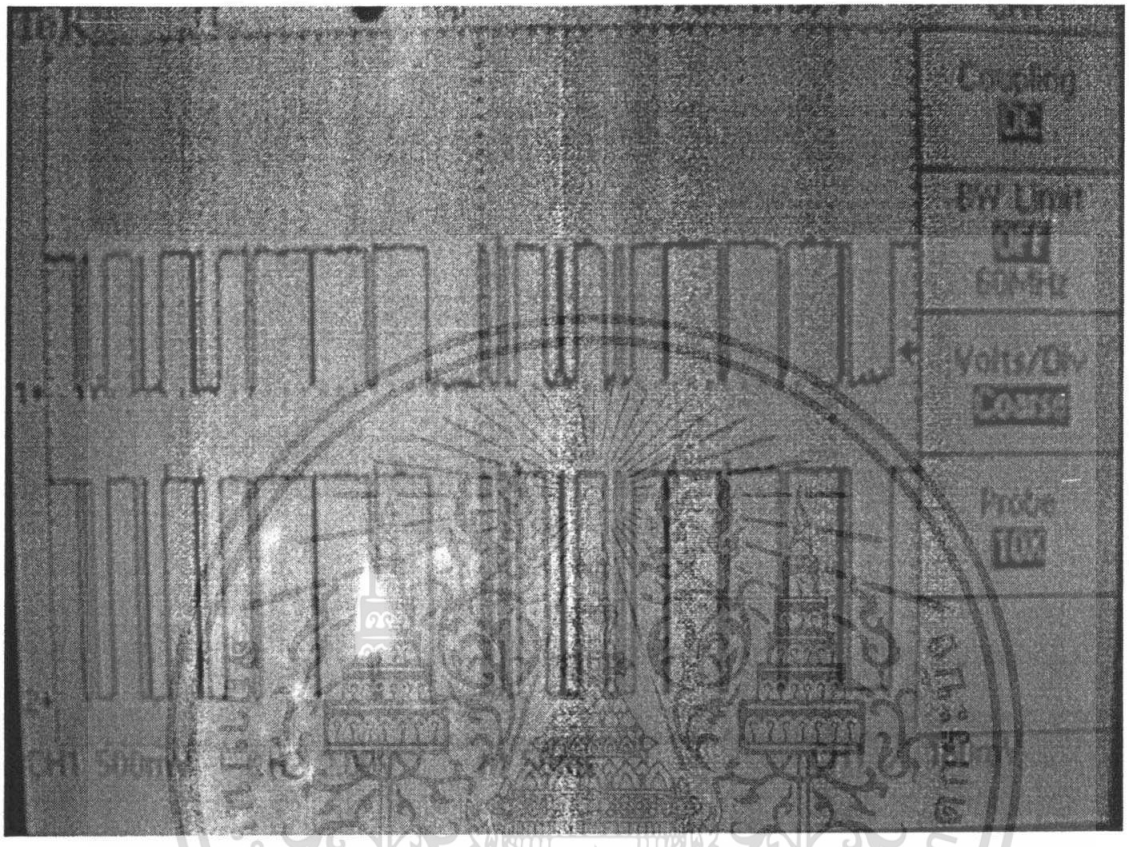
แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่ 5.6 รูปแสดงผลไปควบคุมLED1 และ LED8ออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

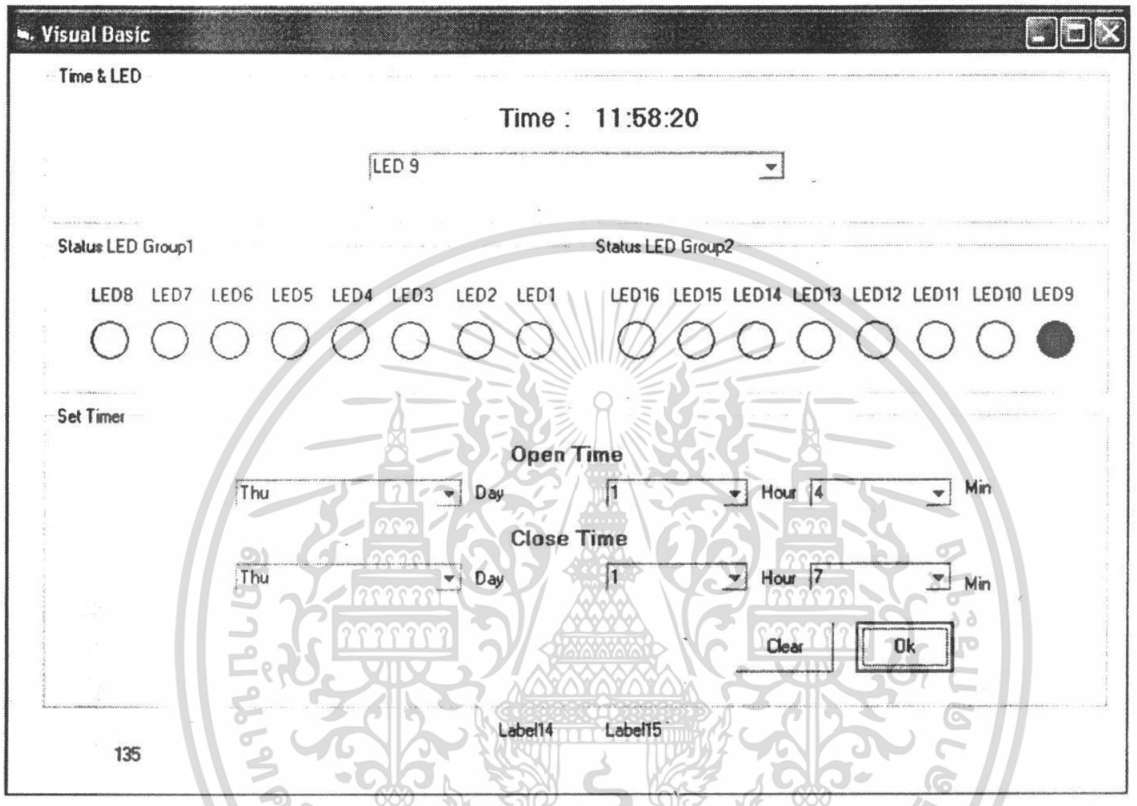
4. กราฟแสดงผลไปควบคุมLED9



รูปที่ 5.7 รูปแสดงผลไปควบคุมLED9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

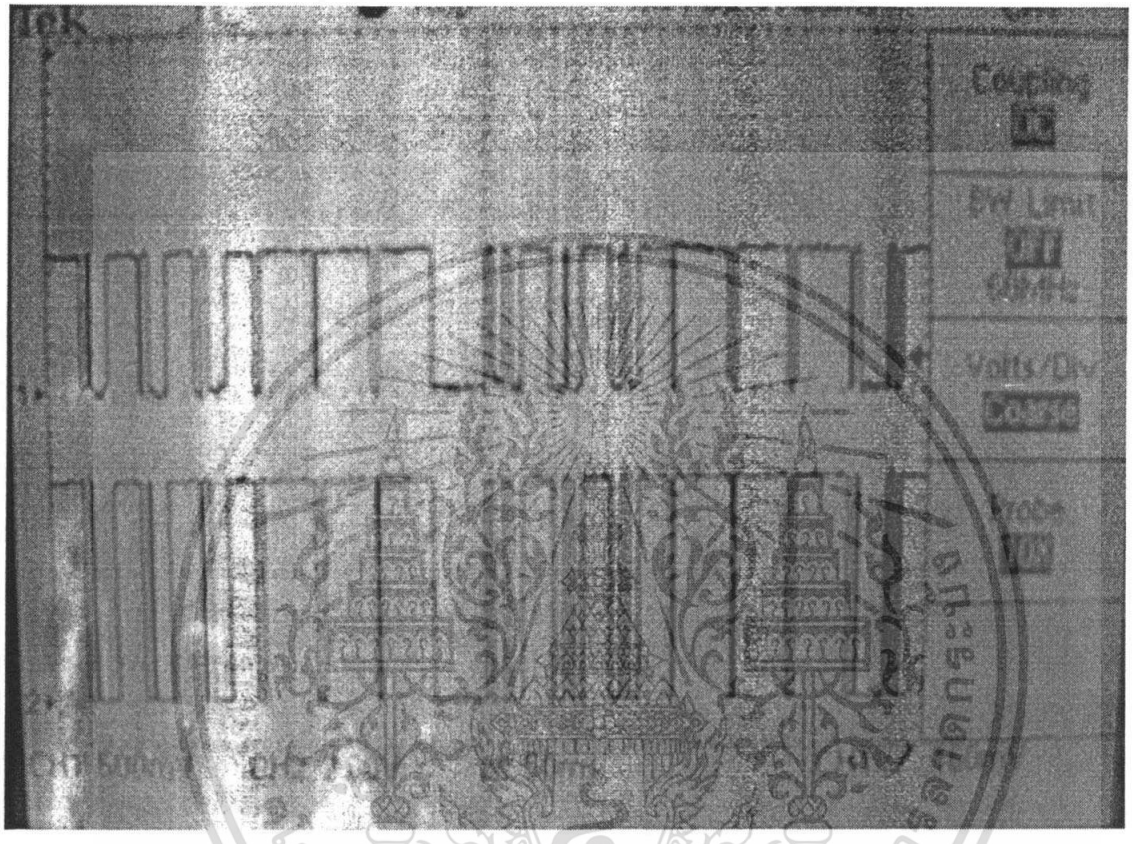
แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่ 5.8 รูปแสดงผลไปควบคุมLED9 ออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

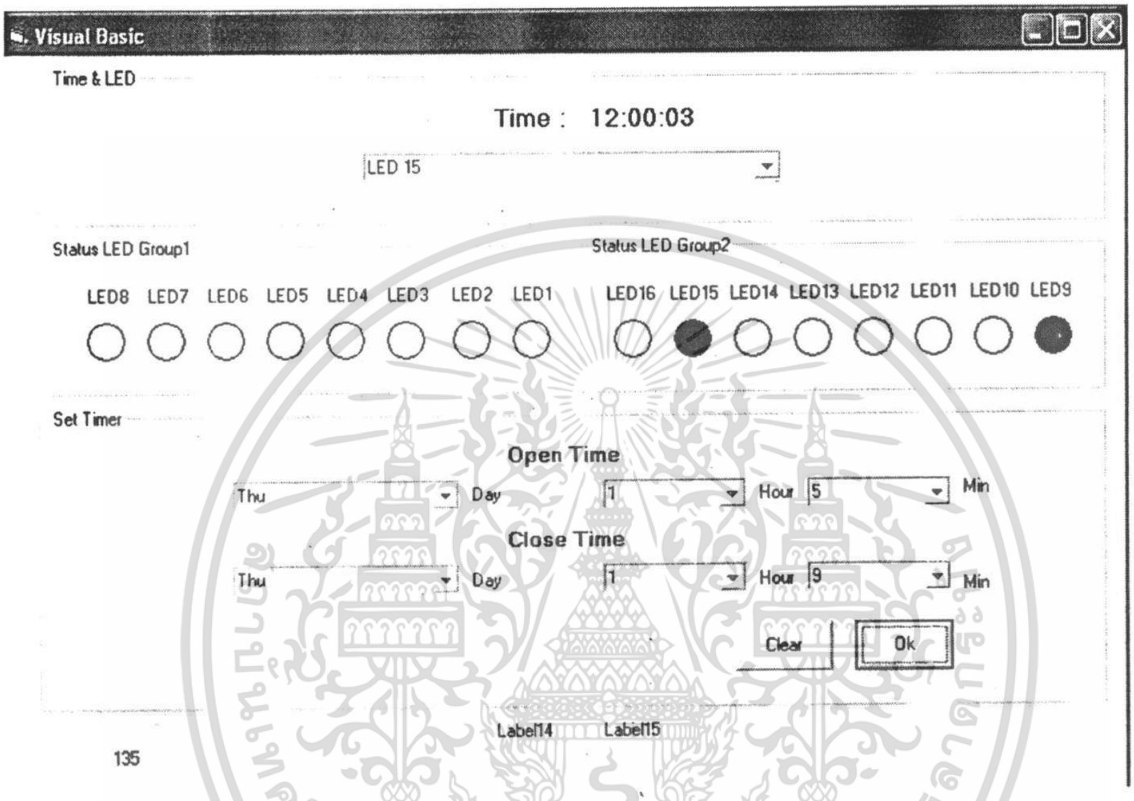
5.กราฟแสดงผลไปควบคุมLED9 และ LED15



รูปที่5.9 กราฟแสดงผลไปควบคุมLED9 และ LED15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

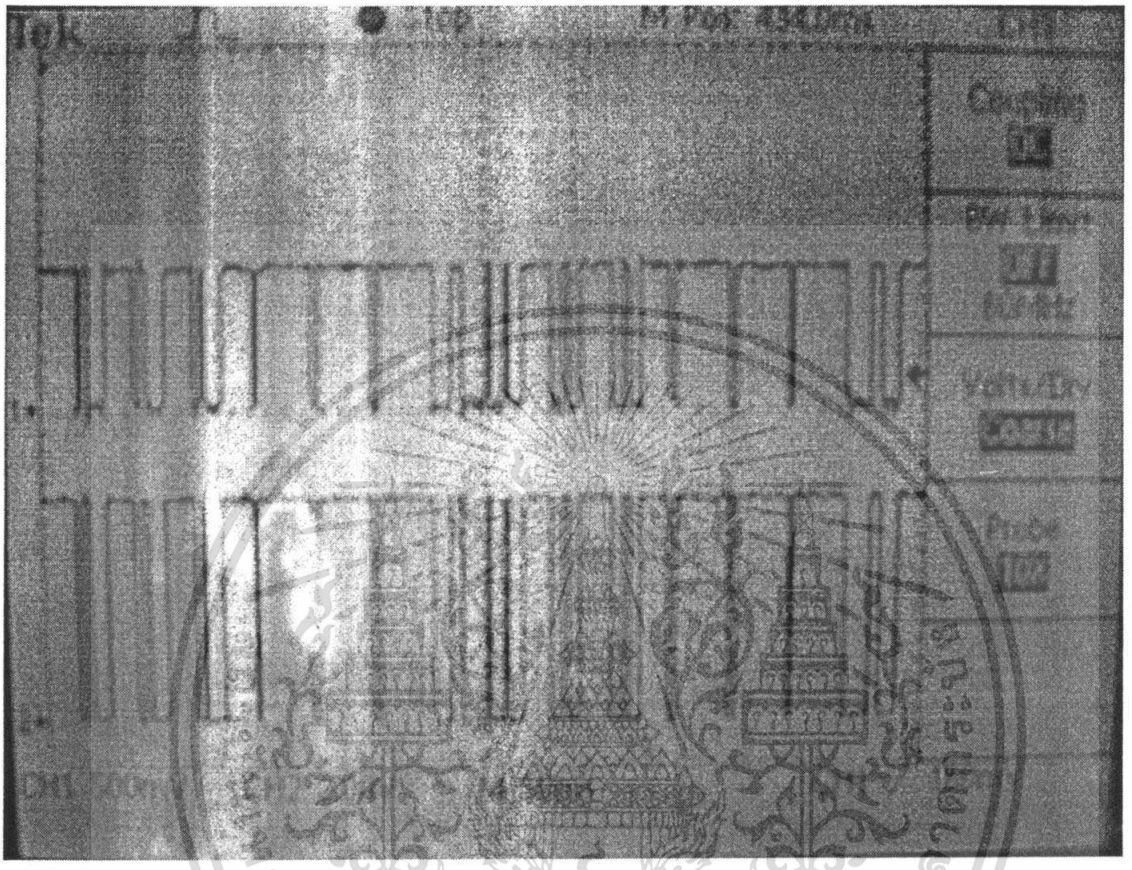
แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่5.10 กราฟแสดงผลไปควบคุมLED9 และ LED15 ออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

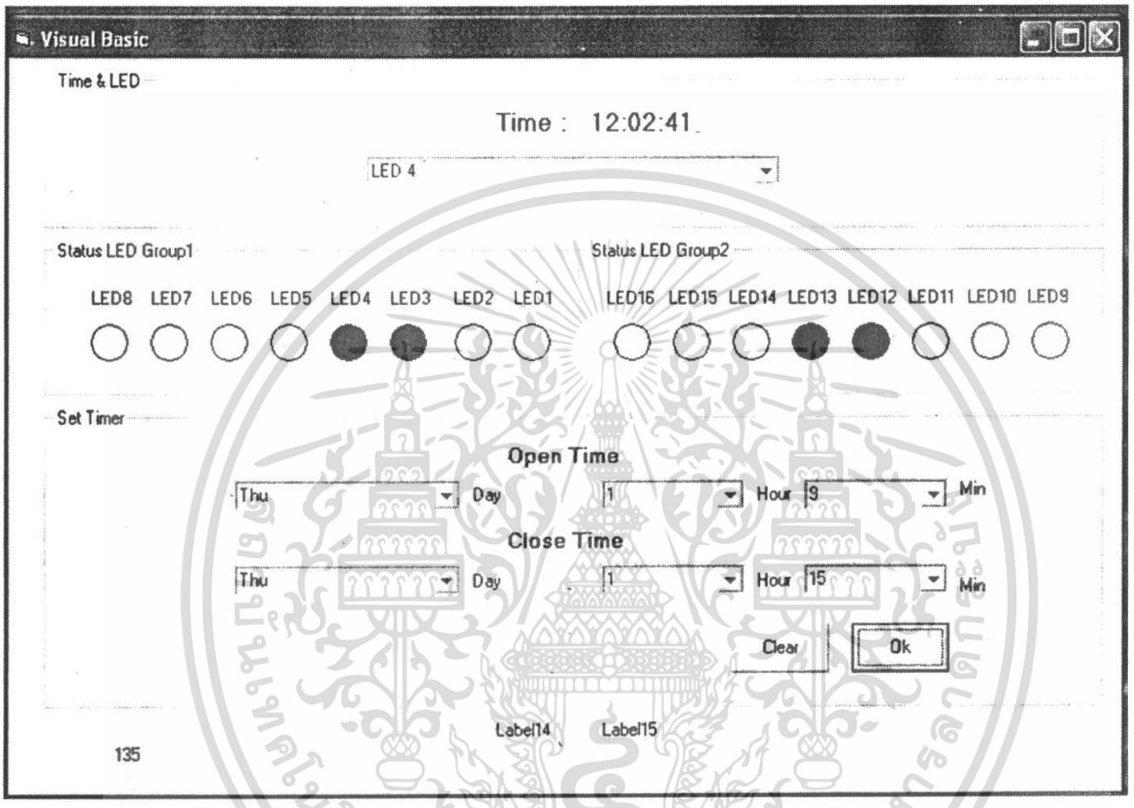
6.กราฟแสดงผลไปควบคุมLED3 , LED4 , LED12 , LED13



รูปที่ 5.11 กราฟแสดงผลไปควบคุมLED3 , LED4 , LED12 , LED13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

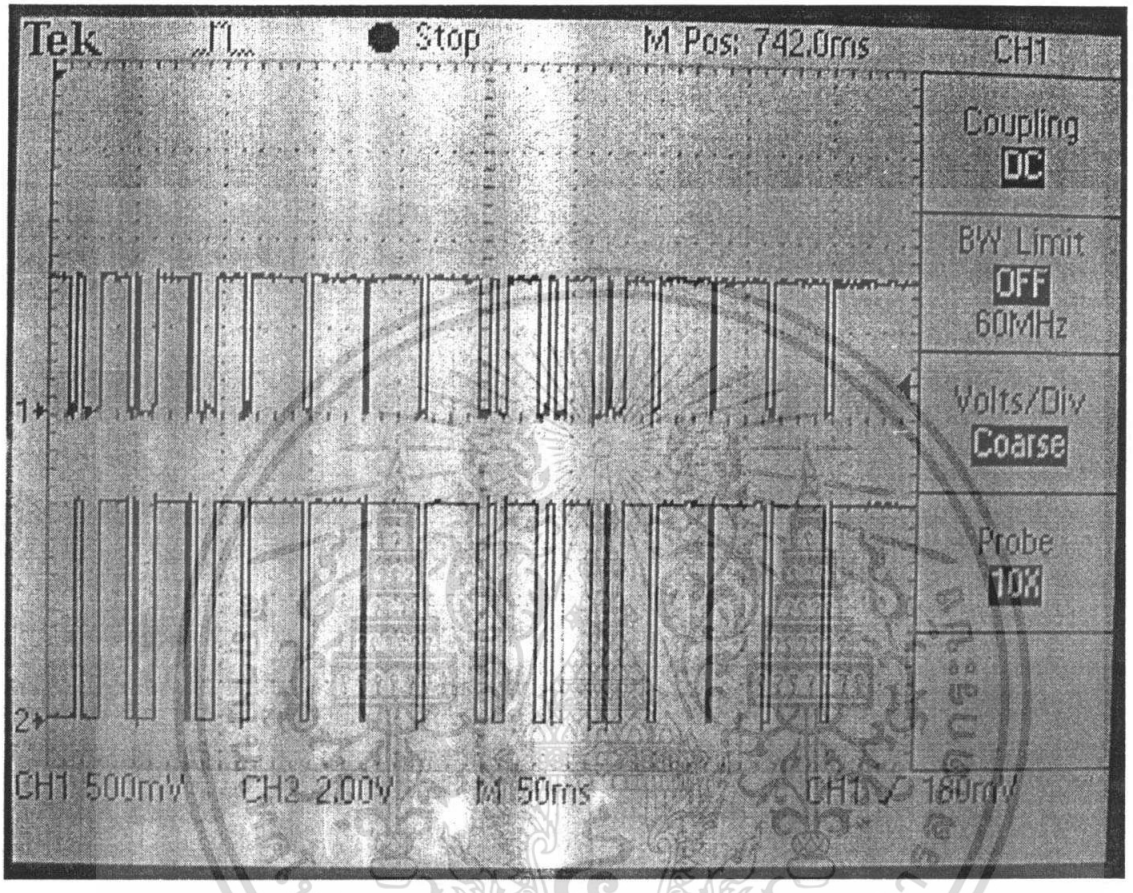
แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่ 5.12 กราฟแสดงผลไปควบคุมLED3 , LED4 , LED12 , LED13 ออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

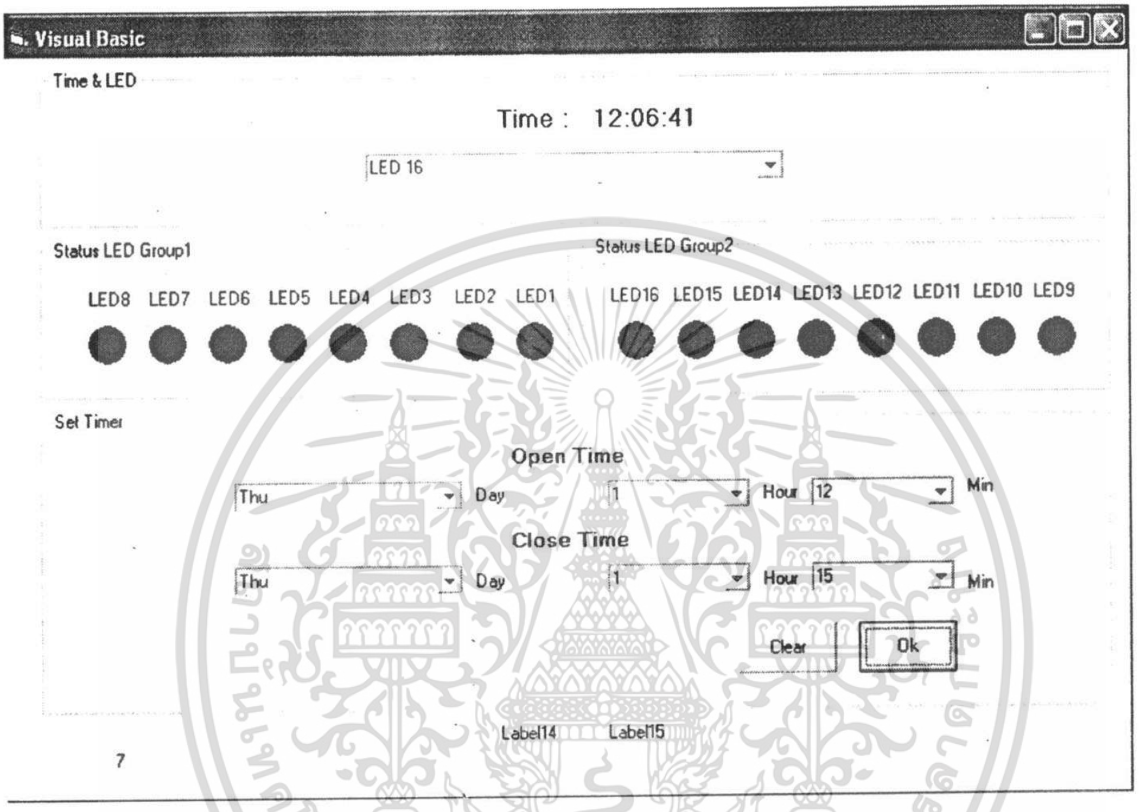
7. กราฟแสดงผลไปควบคุมLEDทุกตัว



รูปที่13 กราฟแสดงผลไปควบคุมLEDทุกตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลออกหน้าจอคอมพิวเตอร์



รูปที่14 รูปแสดงผลไปควบคุมLED ทุกตัวออกหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์

วงจรในโครงงานนี้สามารถแบ่งออกได้เป็น 2 ส่วน คือ ส่วนวงจรภาครับ และ ส่วนของวงจรภาคส่ง โดยส่วนที่ควบคุมการทำงานของวงจรจะอยู่ที่วงจรภาคส่ง

-วงจรภาคส่ง

ในวงจรส่วนนี้จะเริ่มต้นจากการรับค่าคำสั่งจากคอมพิวเตอร์คำสั่งนี้จะถูกส่งไปยังไมโครคอนโทรลเลอร์จากนั้นไมโครคอนโทรลเลอร์จะส่งข้อมูลขนาด 8บิต จำนวน 3 ชุด ที่ใช้ในการควบคุมการเปิด-ปิด อุปกรณ์ไฟฟ้า

-วงจรภาครับ

วงจรนี้จะรับค่าสัญญาณแล้วทำการแปลงสัญญาน แล้วก็นำข้อมูลนี้ไปผ่านวงจรเพื่อควบคุมการเปิด-ปิด อุปกรณ์ไฟฟ้าต่อไป

-ข้อมูลที่ใช้ควบคุมการเปิด-ปิด อุปกรณ์ไฟฟ้า

ไมโครคอนโทรลเลอร์จะส่งคำสั่งควบคุมนี้ออกมาครั้งละ3ชุด โดยชุดแรกคือการเลือกอุปกรณ์ที่จะส่ง ชุดที่สองคือบิตคั่น ชุดที่สาม คือ คำสั่งในการเปิดหรือปิดอุปกรณ์ไฟฟ้าที่อุปกรณ์นั้น

ปัญหาและการแก้ไข

- การส่งแบบ ASK มีข้อเสียคือ มี noise รบกวนมาก จึงทำให้เกิดความผิดพลาดได้ง่าย
- ตรวจสอบความผิดพลาดของโปรแกรมได้ยาก เนื่องจากตัวโปรแกรมมีจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. เอกสารประกอบการเรียนวิชา Electronics Communications
2. หนังสือ ไมโครคอนโทรลเลอร์ MCS-51
3. Data sheet : TLPRLP434(module) , TLPRLP434A(transmitt)
4. หนังสือ เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ ด้วย Visual Basic
5. หนังสือ คู่มือการใช้งาน Protel DXP
6. หนังสือคู่มือการเขียนภาษาC ฉบับเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้