สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบเก็บข้อมูลสภาพแวดล้อมเพื่อการเกษตร

นำเสนอโดย

นาย ธเนศ ทิพย์เลอเลิศ

นาย บัณฑิต สุทธิกิติวงศ์

เลขหมู่.....
เลขทะเบียน 73149
วัน,เดือน,ปี • 6 ก.ค. 2550

b 11-121-110
.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเก็บข้อมูลสภาพแวดล้อมเพื่อการเกษตร

นำเสนอโดย

นาย ธเนศ ทิพย์เลอเลิศ รหัส 45010336

นาย บัณฑิต สุทธิกิติวงศ์ รหัส 45010421

อาจารย์ที่ปรึกษา

รศ. ขนิษฐา แซ่ตั้ง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2548

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเก็บข้อมูลสภาพแวดล้อมเพื่อการเกษตร

(ENVIRONMENT DATA LOGGER SYSTEM FOR AGRICULTURE)

ผู้จัดทำ

1. นาย ธเนศ ทิพย์เลอเลิศ รหัสนักศึกษา 45010336
2. นาย บัณฑิต สุทธิกิติวงศ์ รหัสนักศึกษา 45010421

ลงชื่อ.....

(.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเก็บข้อมูลสภาพแวดล้อมเพื่อการเกษตร

นาย ธเนศ ทิพย์เลอเลิศ รหัส 45010336
นาย บัณฑิต สุทธิกิติวงศ์ รหัส 45010421
รศ. ขนิษฐา แซ่ตั้ง (อาจารย์ที่ปรึกษา)
ปีการศึกษา 2548

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอการออกแบบระบบเก็บข้อมูลแบบหลายช่องสัญญาณ เพื่อใช้ในการเก็บข้อมูลต่างๆ การทำงานของระบบนี้ทั้งหมดจะถูกควบคุมด้วยไมโครคอนโทรลเลอร์ ตระกูลMCS- 51 ในการสุ่มเก็บข้อมูล ข้อมูลที่ได้จะถูกเก็บในหน่วยความจำภายนอกที่มีขนาด 32 กิโลไบต์ มีตัวแสดงเวลาจริง มีหน้าจอ LCD ไว้แสดงขั้นตอนต่างๆ ในแต่ละช่วงการทำงาน และส่งข้อมูลออกไปยังคอมพิวเตอร์ด้วยอัตราเร็ว 9600 บิตต่อวินาทีโดยผ่านพอร์ตอนุกรม RS 232 เพื่อแสดงผลออกมาทางโปรแกรมดาต้าล็อกเกอร์ซึ่งข้อมูลที่เข้ามานี้จะนำไปเก็บในฐานข้อมูลเพื่อนำไปวิเคราะห์ผลทางทฤษฎีต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ENVIRONMENT DATA LOGGER SYSTEM FOR AGRICULTURE

Mr. Taned Tiplerlerd ID 45010336

Mr. Bandit Suthikitiwong ID 45010421

Assoc.Prof. Khanitta Sae-Tang

Educational Year 2005

Abstract

This thesis presents a design of Multichannel data logger system. A Micro-controller family MCS-51 plays the grand role in controlling the whole system. Analogue signal, amplitude up to 5 volts, can be as channel input up to 4 channels. Depend on the channel sampling rate defined by user. Have realtime clock and LCD monitor that shows status in work period. The converted 8 bits digital data are stored in RAM of 32 Kbytes before being sent data to a computer via a serial port at board rate 9600 bps . The record data could be displayed on monitor that show form of graph and data table.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเรื่อง **ระบบเก็บข้อมูลสภาพแวดล้อมเพื่อการเกษตร** สำเร็จได้ด้วยดีก็เพราะได้รับความเอื้อเฟื้อจาก **อาจารย์ ขนิษฐา แซ่ตั้ง** ซึ่งคอยให้คำปรึกษาและแนะแนวทางในการดำเนินงานรวมถึงแนะแนวทางแก้ปัญหาต่างๆที่เกิดขึ้น รวมทั้งเพื่อนๆที่คอยให้ช่วยเหลือในด้านต่างๆ

ธเนศ ทิพย์เลอเลิศ

(นาย ธเนศ ทิพย์เลอเลิศ)

บัณฑิต สุทธิกิตติวงศ์

(นาย บัณฑิต สุทธิกิตติวงศ์)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ

สารบัญ

บทที่ 1 1

บทนำ 1

1.1 แนวความคิดและโครงสร้างของโครงงาน 1

บทที่ 2 3

ทฤษฎีพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 3

2.1 คุณสมบัติไมโครคอนโทรลเลอร์ MCS-51 3

2.2 โครงสร้างสถาปัตยกรรม MCS-51 4

2.3 การจัดหน่วยความจำ 7

2.3.1 Program Memory 7

2.3.2 Data Memory 9

2.4 พอร์ตอินพุท / เอาท์พุทของ MCS-51 11

2.4.1 การใช้งานเป็นพอร์ตอินพุท 12

2.4.2 การใช้งานเป็นพอร์ตเอาท์พุท 12

2.5 ฐานเวลาในการทำงานของซีพียู MSC-51 12

2.6 การอินเตอร์เฟสโดยพอร์ต RS-232 14

2.6.1 มาตรฐาน RS-232 14

2.6.2 การอินเตอร์เฟสตามมาตรฐาน RS-232 14

บทที่ 3 16

การออกแบบและการทำงาน 16

3.1 การออกแบบภาคควบคุม 16

3.1.1 การสร้างฐานเวลาให้กับไมโครคอนโทรลเลอร์ 16

3.1.2 การเชื่อมต่อกับพอร์ตอนุกรมคอมพิวเตอร์ 23

3.1.3 การเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำภายนอก 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 30
สรุปผลการทำงาน 30
4.1 สรุปผลการทำงาน 30
4.2 แนวทางในการพัฒนาและประยุกต์การใช้งาน 30
4.3 ปัญหาในการทำงาน 31
ภาคผนวก
คู่มือการใช้ดาต้าล็อกเกอร์ ก
คู่มือการใช้โปรแกรมดาต้าล็อกเกอร์ ช
กิตติกรรมประกาศ ฌ
เอกสารอ้างอิง ฌ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

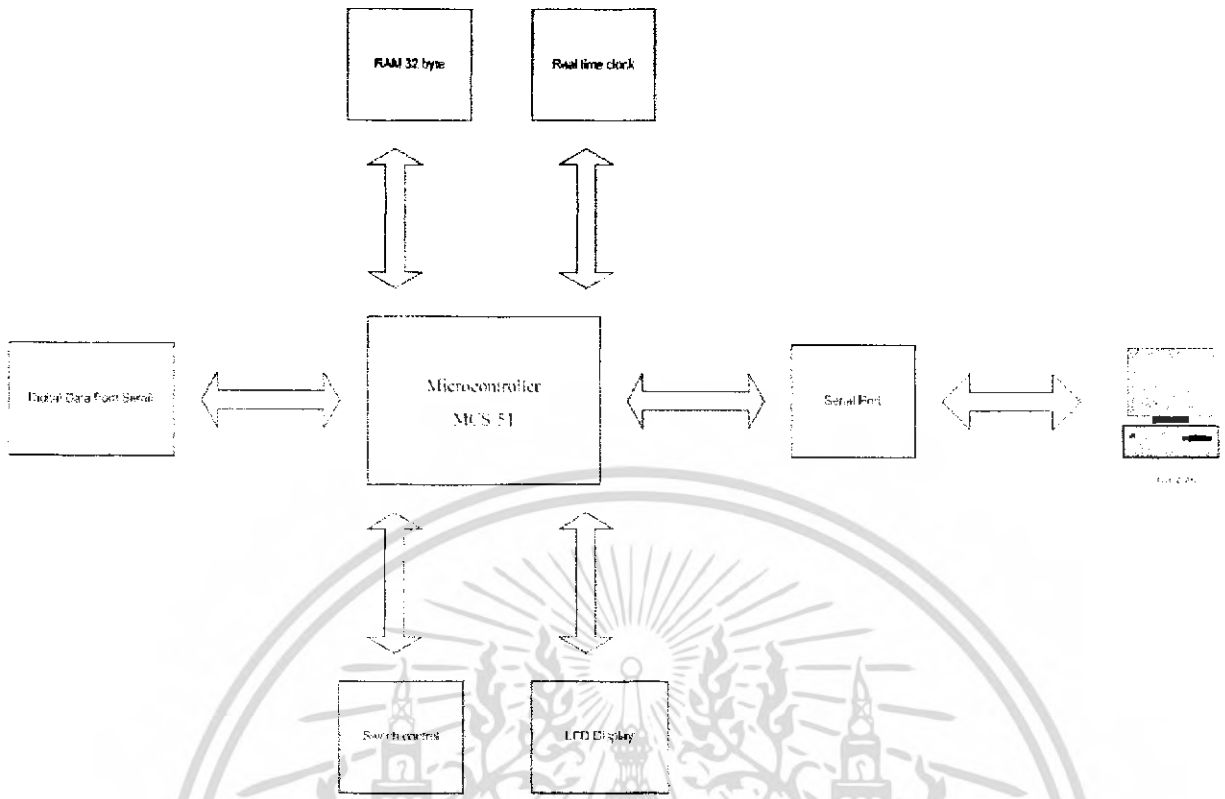
บทที่ 1
บทนำ

ดาต้าล็อกเกอร์ (data logger) คือ อุปกรณ์ที่ใช้ในการเก็บบันทึกข้อมูลในช่วงเวลาที่กำหนด เพื่อนำมาแสดงผลหรือนำข้อมูลเหล่านั้นมาประมวลผลภายหลัง การเก็บข้อมูลในลักษณะดังกล่าว อาจจะเป็นการบันทึกแบบอนาลอก เช่น การบันทึกเสียงและภาพแบบเก่า การเก็บบันทึกข้อมูลอย่างต่อเนื่องบนกระดาษกราฟ อย่างไรก็ดีการบันทึกแบบดิจิตอลจะมีข้อดีในแง่ของการจัดเก็บที่สามารถเก็บปริมาณได้มากในขนาดทางกายภาพที่เล็กกว่า นอกจากนี้สามารถนำมาประมวลผลด้วยดิจิตอลคอมพิวเตอร์ได้โดยตรง ข้อดีประการแรกดังกล่าวของดิจิตอลดาต้าล็อกเกอร์นั้น เหมาะกับการเก็บข้อมูลในบริเวณที่ห่างไกล ยากต่อการตรวจสอบ หรือ การบันทึกการเปลี่ยนแปลงนั้นเป็นเวลานานๆ เนื่องจากขนาดเล็กและเก็บข้อมูลได้มากกว่าการกินกำลังงานน้อยกว่าจากการใช้เทคโนโลยีของอุปกรณ์สารกึ่งตัวนำในปัจจุบัน แต่เนื่องจากปริมาณทางกายภาพที่ใช้เซ็นเซอร์(sensor)แปลงให้อยู่ในรูปของปริมาณทางไฟฟ้าแล้วเกือบทั้งหมดจะอยู่ในรูปแบบอนาลอกที่ต่อเนื่องแกนเวลา การเก็บแบบดิจิตอลนั้นต้องมีการสุ่มสัญญาณนั้นในเวลาใดๆ แล้วนำมาเข้ารหัสดิจิตอลเพื่อให้เหมาะกับการจัดเก็บและประมวลผล ดังนั้น การจัดเก็บแบบดิจตอลอาจจะขาดความต่อเนื่องหากการสุ่มค่ามีช่วงห่างเมื่อเทียบกับการเปลี่ยนแปลงของปริมาณทางกายภาพนั้นๆ ดังนั้นดิจิตอลดาต้าลอกเกอร์มักจะสามารถกำหนดช่วงการสุ่มได้ การนำไมโครคอนโทรลเลอร์จะช่วยลดความซับซ้อนของวงจรอิเล็กทรอนิกส์ที่ใช้ในการควบคุมกระบวนการแปลงสัญญาณ การสุ่มและการจัดเก็บข้อมูลแบบอัตโนมัติโปรแกรมได้ โครงงานนี้ได้ทำการศึกษาออกแบบระบบดิจิตอลดาต้าล็อกเกอร์แบบ 8 บิต สี่ช่องสัญญาณ ควบคุมด้วยไมโครคอนโทรลเลอร์ โดยใช้อุณหภูมิเป็นตัวอย่างของข้อมูลในการเก็บบันทึก

1.1 แนวความคิดและโครงสร้างของโครงงาน

แนวความคิดของโรงงานที่ต้องการสร้างคือ ดาต้าล็อกเกอร์ ที่สามารถวัดได้หลายช่องสัญญาณ โดยมีไมโครคอนโทรลเลอร์ MCS 51 เป็นอุปกรณ์ที่ใช้ในการควบคุมระบบ โดยเราสามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ และมีการส่งข้อมูลที่วัดได้นั้นมาบันทึกบนไมโครคอมพิวเตอร์ซึ่งอยู่ระยะไกลได้จากแนวความคิดของโครงงานดังกล่าวทำให้เราได้บล็อกไดอะแกรมของโครงงานดังรูปที่ 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่1.1 โครงสร้างของโครงงาน

จากแนวความคิดในการออกแบบระบบ ซึ่งกำหนดว่าสามารถใช้ในการจัดเก็บข้อมูลจากเซ็นเซอร์ได้ ทั้งหมด 4 ช่อง ระดับแรงดันอินพุทตั้งแต่ 0-5 โวลต์ แต่ในข้อจำกัดของเวลาในการดำเนินโครงงาน จะทำการวัดเฉพาะอุณหภูมิเพียงอย่างเดียว โดยใช้เซมิคอนดักเตอร์เซนเซอร์ วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ใช้ขนาด 8 บิต ข้อมูลดิจิตอลจะถูกเก็บบันทึกไว้ในหน่วยความจำแบบแรม(RAM) ขนาด 32 กิโลไบต์ตลอดในช่วงการเก็บบันทึก ก่อนที่จะอ่านหรือส่งต่อไปยังไมโครคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม (RS-232) การกำหนดรายระเอียดในการสุ่มเก็บข้อมูลนั้นสามารถกำหนดได้ว่าจะให้เก็บอย่างไรหรือเก็บทุกๆกี่นาที และสามารถตั้งเวลาในการเก็บข้อมูลหรือหยุดการเก็บข้อมูลได้โดยการโปรแกรมบนไมโครคอมพิวเตอร์ โปรแกรมบนไมโครคอนโทรลเลอร์โดยตรง ข้อมูลที่ได้จากดาต้าล็อกเกอร์จะเก็บในลักษณะไฟล์ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีพื้นฐานของไมโครคอนโทรลเลอร์MCS-51

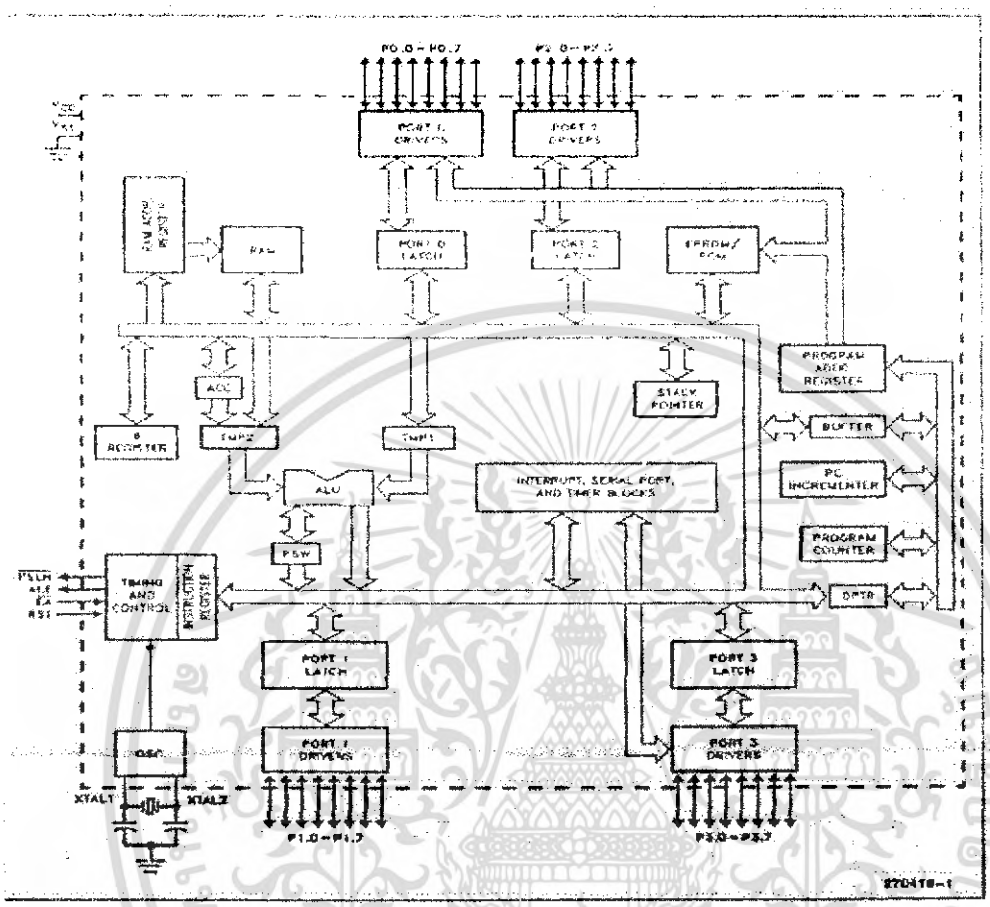
ไมโครคอนโทรลเลอร์ เป็นชื่ออุปกรณ์อิเล็กทรอนิคส์แบบหนึ่งที่บรรจุความสามารถมากมาย ไม่ว่าจะเป็นหน่วยประมวลผลหน่วยคำนวณทางคณิตศาสตร์และลอจิกวงจรรับสัญญาณอินพุทวงจรขับสัญญาณออกทางเอาท์พุตหน่วยความจำวงจรกำเนิดสัญญาณนาฬิกาทำให้ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลงในขณะที่มีขีดความสามารถสูงขึ้นดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้นิยมในการควบคุม โดยสามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.1 คุณสมบัติไมโครคอนโทรลเออร์ MCS-51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียู 8 บิต
- ภายในมีหน่วยความจำเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้เป็นพันครั้ง
- สายอินพุทและเอาท์พุทมีจำนวน 32 เส้นใช้เลือกแอดเดรสแยกต่างหากจากกันได้
- มีแรมบรรจุภายในขนาด 128 ไบต์ หรือ 256 บิต
- วงจรตั้งเวลา/วงจรนับมีขนาด 16 บิตอย่างน้อย 2 ตัว
- กำหนดเป็น UART(Universal Synchronous Asynchronous Receiver Transmitter) ส่งข้อมูลอนุกรมได้สองทิศทาง
- อินเทอร์รัปต์ แบ่งเป็น 2 ระดับจาก 5 หรือ 6 แหล่ง
- มีสัญญาณนาฬิกาอยู่ภายในตัว
- มีหน่วยความจำสำหรับเก็บข้อมูลภายในขนาด 4 หรือ 8 กิโลไบต์
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- คำสั่งทั้งหมดมี111คำสั่ง
- ทำงานด้วยเลขฐานสิบ และเลขฐานสิบหก
- ในเบอร์ AT89C2051, AT89C1051, จะลดขนาดลงเหลือแค่ 20 ขาโดยมี ROM ภายในจำนวน 2 กิโลไบต์ และ 1 กิโลไบต์ ตามลำดับ โดยจะไม่มี Port0 ,Port 2 ,ALE, PSEN, EA และ RD เนื่องจากได้อัดโปรแกรมไว้ในตัวแล้วขาเหล่านี้จะไม่ได้ใช้งาน

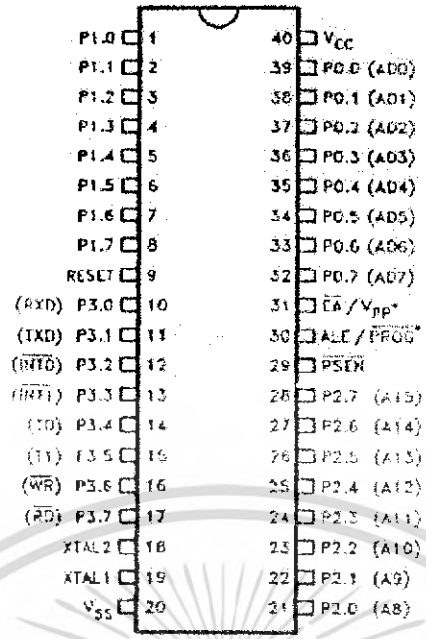
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างสถาปัตยกรรมของ MCS-51



รูปที่ 2.1 โครงสร้างภายใน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ตำแหน่งขาของชิปไมโครคอนโทรลเลอร์ MCS- 51

หน้าที่การใช้งานแต่ละขาของชิปไมโครคอนโทรลเลอร์MCS-51มีดังนี้

- **ขาVss (ขา20) สำหรับต่อลงกราวด์**
- **ขาVcc (ขา40) สำหรับต่อแหล่งจ่ายแรงดันกระแสไฟตรงขนาด 5โวลต์**
- **ขาพอร์ต 0 (ขา 32-39) มี 8 ขาใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต(P0.0-P0.7) แบบOpen drain Bi-directional พอร์ตนี้สามารถใช้งานเป็นพอร์ตอินพุท เอาท์พุททั่วไปได้และใช้ในการติดต่อหน่วยความจำสำหรับโปรแกรมและข้อมูลภายนอกชิปด้วย**
- **ขาพอร์ต 1 (ขา1-8) มี 8 ขาใช้สำหรับพอร์ต 1 (P1.0-P1.7)สามารถใช้เป็นอินพุท และเอาท์พุท พอร์ตทั่วไปได้ หากต้องการใช้งานเป็นพอร์ตอินพุท ต้องโหลดค่า ไปยังแต่ละบิตของพอร์ตนี้**
- **ขาพอร์ต 2 (ขา21-28) มี 8 ขาใช้เป็นขาสำหรับพอร์ต 2 ขนาด8 (P2.0-P2.7) มีโครงสร้างคล้ายพอร์ต 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทาน พูลอัพแทน พอร์ตนี้ทำงานได้สองหน้าที่คือ สามารถใช้เป็น แอดเดรสบัส ขนาด 8 บิต(A8-A15) และเป็นพอร์ตอินพุท เอาท์พุทใช้งานทั่วไป เมื่อจะใช้งานเป็นพอร์ตอินพุท ต้องส่ง ลอจิก "1" มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ FETอยู่ในสภาวะ off**
- **ขาพอร์ต 3 (ขา 10-17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (P3.0-3.7) ขนาด 8 บิต พอร์ตนี้สามารถใช้งานเป็นอินพุท เอาท์พุทพอร์ตทั่วไปได้ นอกจากนี้ยังใช้งานในหน้าที่พิเศษต่างๆ ดังนี้**

  ขาP3.0 ใช้รับข้อมูลจากภายนอกแบบอนุกรม

  ขาP3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาP3.2 ใช้เป็นอินพุทเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 0

ขาP3.3 ใช้เป็นอินพุทเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 1

ขาP3.4 สัญญาณอินพุทให้เคาน์เตอร์ของไทม์เมอร์ 0

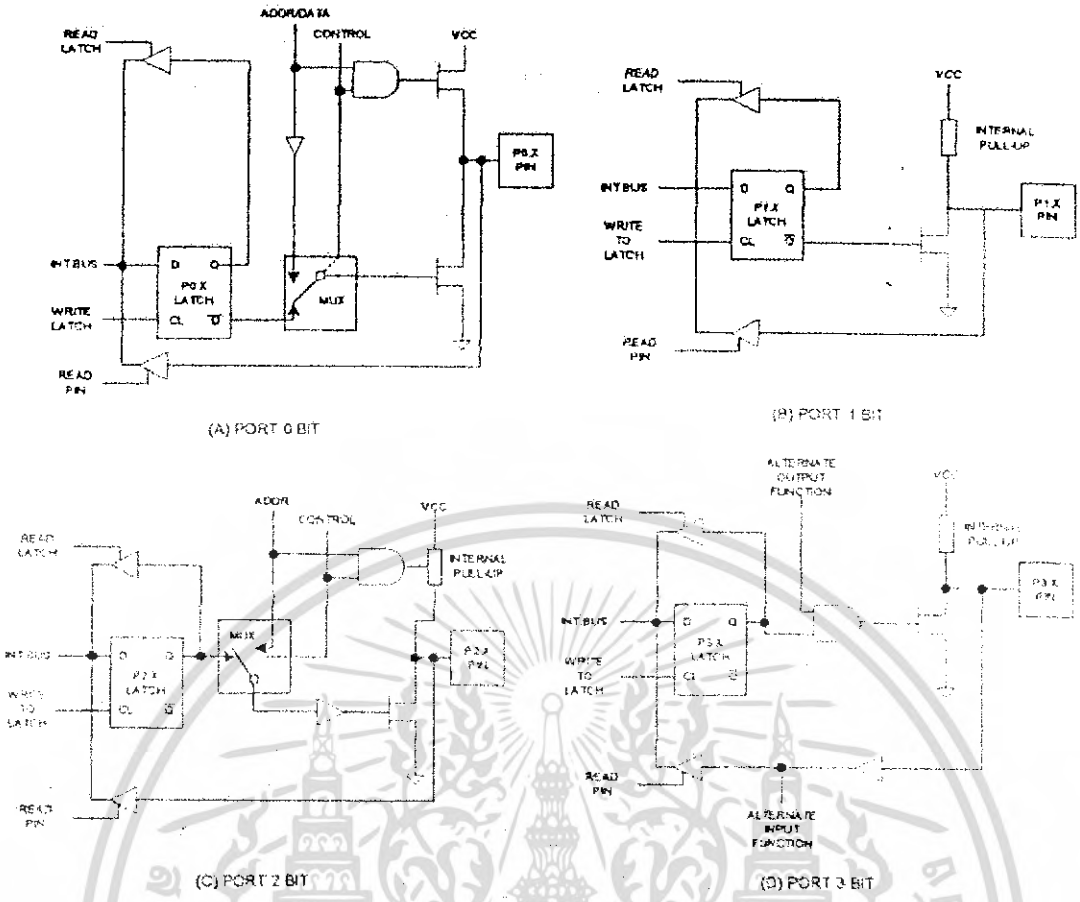
ขาP3.5 สัญญาณอินพุทให้เคาน์เตอร์ของไทม์เมอร์ 1

ขาP3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บภายนอกชิป

ขาP3.7 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลจากหน่วยความจำสำหรับเก็บภายนอกชิป

- ขา RT (ขา9) ใช้สำหรับการรีเซ็ทวงจรทุกอย่างภายในชิป เพื่อเริ่มต้นการทำงานใหม่
- ขา ALE/PROG (ขา30) เป็นขาใช้สำหรับส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำ (Address Latch Enable) พอร์ต 0 ในการติดต่อหน่วยความจำสำหรับโปรแกรมหรือข้อมูลภายนอก สัญญาณนี้จะแอคทีฟทุก ๆ 2 ครั้งใน 1 แมชชีนไซเคิล
- ขา PSEN (ขา 29) ใช้ส่งสัญญาณสโตรป เพื่ออ่านคำสั่งจากโปรมแกรมที่เก็บไว้ในหน่วยความจำภายนอกชิป (Program Strobe Enable)สัญญาณนี้จะส่งออกมาสองครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่านคำสั่งจากหน่วยความจำโปรแกรมภายในจะไม่มีสัญญาณออกที่ขานี้
- ขา EA/Vpp (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมที่อยู่ภายในชิปหรือภายนอกชิป หากขานี้มีสถานะเป็น 0 หมายถึง ให้ใช้โปรแกรมที่อยู่ภายนอก หากขานี้มีสถานะเป็น 1 หมายถึงบังคับให้ MCS-51 ใช้โปรมแกรมจากหน่วยความจำสำหรับโปรแกรมภายในชิป ส่วน MCS-51 ที่ไม่มีหน่วยความจำโปรแกรมสำหรับเก็บโปรแกรมภายในชิปให้ต่อขานี้ลงกราวด์เสมอ
- ขา XTAL1(ขา19) ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุทเข้าสู่วงจรออสซิลเลเตอร์
- ขา XTAL2 (ขา 18) ใช้ต่อคริสตอลภายนอก โดยเป็นเอาท์พุทออกจากวงจรออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.3 แสดงโครงสร้างแต่ละบิตภายในพอร์ตอินพุต/เอาท์พุทของ 8051

2.3 การจัดหน่วยความจำ

หน่วยความจำที่ใช้กับ MCS-51 มีอยู่ด้วยกัน 2 แบบ คือ

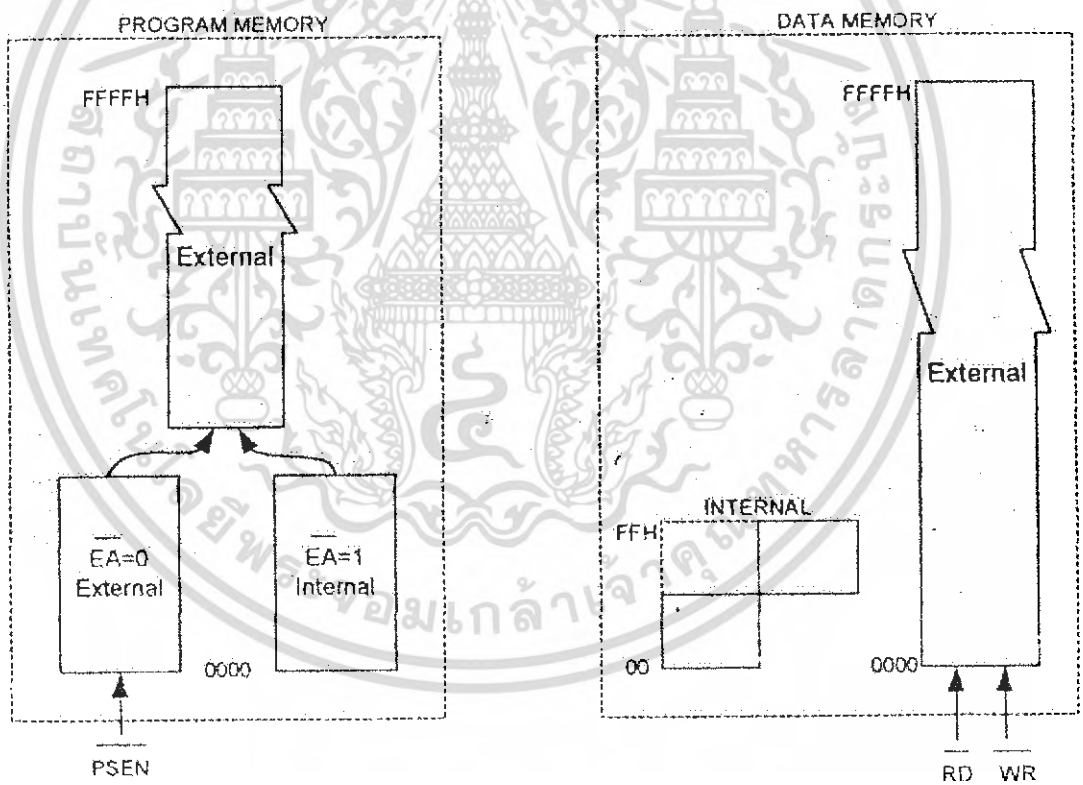
- หน่วยความจำสำหรับโปรแกรม (Program Memory)
- หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

2.3.1 Program Memory

เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ 8051 ทำงานเมื่อ 8051 ทำงานก็ต้องอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณควบคุมส่วนอื่น ๆ ตามการทำงานของแต่ละคำสั่งหน่วยความจำแบบนี้ จะต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้เขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็น รหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ การเขียนข้อมูลลงไปบน ROM จะต้องใช้เครื่องมือพิเศษในระหว่างการทำงานของ 8051 ผู้ใช้จะไม่สามารถใช้คำสั่ง ทำงานเขียนข้อมูลลงในหน่วยความจำแบบนี้ได้จำนวนตำแหน่งสูงสุดของหน่วยความจำแบบนี้ที่ 8051 จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่ง (Address) จะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000-FFFFh หน่วยความจำตำแหน่ง 0000-FFFFh จำนวน 4 กิโลไบต์ นั้นผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่ภายในหรือภายนอก 8051 ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน 8051 ก็ให้ป้อนสัญญาณสภาวะลอจิก High(l) เข้าที่ขา EA ของ 8051แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอก 8051 ก็ให้ต่อลอจิก Low(0) เข้าที่ขา EA ของ 8051 ส่วนหน่วยความจำตำแหน่ง 1FFFh ถึง FFFFh จะต้องต่ออยู่ภายนอก 8051 เสมอดังแสดงใน แผนภูมิหน่วยความจำ (Memory Map) ในรูปที่ 3-4

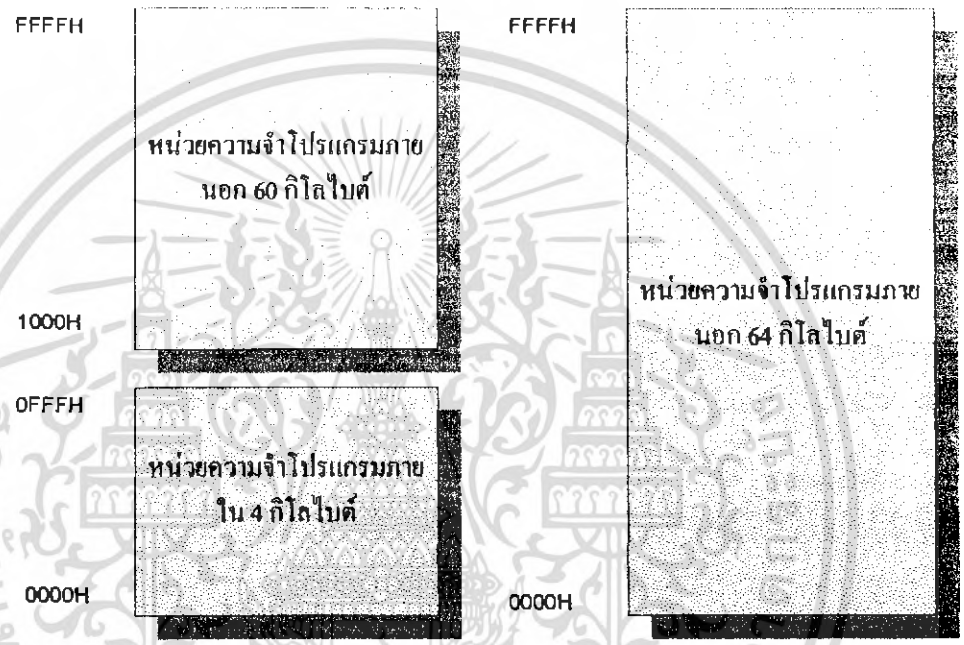


รูปที่ 2.4 แผนภูมิหน่วยความจำ

ในพื้นที่ของหน่วยความจำโปรแกรมไม่ว่าจะใช้งานจากภายในหรือภายนอกก็ตามต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H
- พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH
- พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H
- พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH
- พื้นที่สำหรับบริการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H
- พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH



รูปที่ 2.5 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

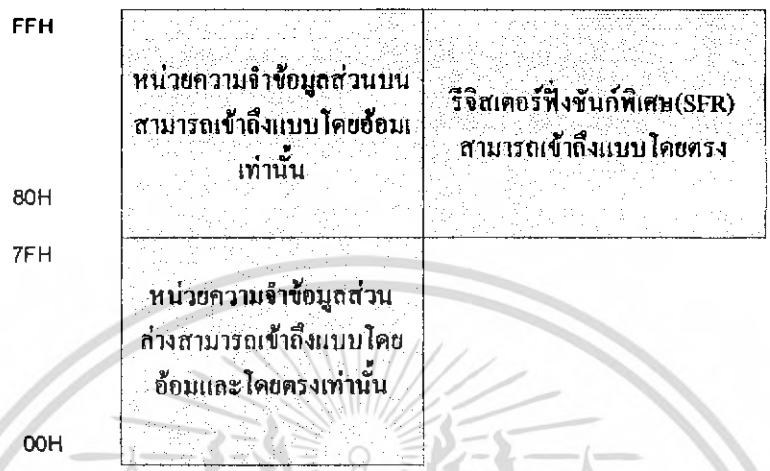
Internal Memory หมายถึง หน่วยความจำนั้นจะอยู่ภายใน 8051 ส่วน External Memory หมายถึง หน่วยความจำนั้นอยู่ภายนอก 8051

2.3.2 Data Memory

เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพักเก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของ 8051 การอ่านหรือการเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำประเภทนี้เป็นประเภท Random Access Memory ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย หน่วยความจำแบบ Data Memory ของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบต์ ที่ตำแหน่ง 00h-7Fh และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกของ 8051 มีได้สูงสุด 65536 ไบต์ อยู่ที่ตำแหน่ง 0000-FFFFh ดังแสดงในรูปที่ 3.5 โดยหน่วยความจำแบบ Data Memory ใน 8051 ที่ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

80h-FFh นั้นไม่ได้มีอยู่ทุกตำแหน่ง จะมีอยู่เฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่า Special Function Register(SFR) เพราะจะใช้หน่วยความจำเหล่านี้สำหรับงานพิเศษเท่านั้น อาจเป็น RAM หรือวงจรนับ วงจรตั้งเวลาก็ได้



รูป2.6 การจัดสรรพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

จะเห็นได้ว่าหน่วยความจำข้อมูลส่วนบนและรีจิสเตอร์ฟังก์ชั่นพิเศษมีตำแหน่งทับซ้อนกัน แต่จะใช้การติดต่อที่แตกต่างกัน โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์เช่นกัน มีแอดเดรสอยู่ที่ 80H-FFH แต่สามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ ฟังก์ชั่นพิเศษมีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกันกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ฟังก์ชั่นพิเศษใช้การเข้าถึงแบบโดยตรง

ตารางที่ 2-1 แสดงตำแหน่งแอดเดรสของรีจิสเตอร์ใช้งานทั่วไป

แอดเดรส	รีจิสเตอร์แบงค์	ชื่อรีจิสเตอร์ใช้งาน
00H – 07H	0	R0-R7
08H – 0FH	1	R0-R7
10H – 17H	2	R0-R7
18H – 1FH	3	R0-R7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2-2 แสดงการเลือก รีจิสเตอร์แบงค์ของรีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์	บิต RS0	บิต RS1	ตำแหน่งหน่วยความจำ
แบงค์ 0	0	0	0000H
แบงค์ 1	0	1	0008H
แบงค์ 2	1	0	0010H
แบงค์ 3	1	1	0018H

2.4 พอร์ตอินพุท/เอาท์พุทของ 8051

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีโครงสร้างของพอร์ตที่สามารถใช้งานแบบขนานได้จำนวนทั้งหมด 4 พอร์ต เรียกชื่อตามลำดับว่า พอร์ต 0 , 1 , 2 และ 3 เป็นพอร์ตขนาด 8 บิททั้งหมดการใช้งานพอร์ตสามารถทำได้ทั้งในลักษณะของเส้นสัญญาณเดี่ยว ๆ หรือกลุ่มของสัญญาณก็ได้ นอกจากนี้พอร์ต 0-3 ยังสามารถนำไปใช้งานอื่น ๆ ที่ไม่ใช่พอร์ตอินพุต/เอาท์พุตได้ โดยพอร์ต 0 ทำหน้าที่มัลติเพลกซ์ระหว่างบัสแอดเดรสไบต์ต่ำและบัสข้อมูล

ซึ่งพอร์ต 0 นั้นไม่สามารถที่จะใช้งานในการอ้างแอดเดรสไบต์ต่ำและอ่านเขียนข้อมูลไปยัง RAM ภายนอกได้พร้อม ๆ กัน ด้วยเหตุนี้จึงมีการต่อ ไอซี 74HC573 ร่วมเข้าไปเพื่อช่วยในการแลตช์แอดเดรสไบต์ต่ำให้กับหน่วยความจำภายนอก โดยขา ALE ของ MCU(Microcontroller Unit) จะเป็นตัวควบคุมจังหวะการแลตช์ให้กับ ไอซี 74HC573 และขณะที่ ไอซี 74HC573 แลตช์แอดเดรสไบต์ต่ำอยู่นั้น พอร์ต 0 จะทำการอ่านเขียนข้อมูลกับหน่วยความจำภายนอกได้นั้นเอง ซึ่งจากวงจรจะเห็นว่าหน่วยความจำภายนอกนั้น จะมีสัญญาณในการอ่าน เขียนข้อมูล โดยขาสัญญาณ RD และ WR ของ MCU เป็นตัวควบคุมในพอร์ต 2 นั้นไม่มีปัญหาแต่อย่างใด เพราะใช้ในการอ้างแอดเดรส ไบต์สูงเพียงหน้าที่เดียวเท่านั้น

สำหรับพอร์ต 3 นั้นนอกเหนือจากความสามารถเช่นพอร์ตปกติแล้ว สามารถนำไปเป็นขาสัญญาณของการอินเตอร์รัพท์ต่าง ๆ รวมทั้งสร้างสัญญาณควบคุม RD และ WR เพื่อทำหน้าที่อ่านหรือเขียนหน่วยความจำข้อมูลภายนอกด้วย การใช้งานพอร์ตลักษณะงานแบบอื่น ๆ ที่ไม่ใช้เป็นพอร์ตแบบอินพุท/เอาท์พุทนี้จะดำเนินการโดยซีพียูโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การใช้งานเป็นพอร์ตอินพุท

การใช้งานพอร์ตเป็นการอินพุทข้อมูลนั้นก่อนอื่นจะต้องส่งข้อมูลค่า "1" ออกมาที่บิตของพอร์ตก่อนเพื่อนหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทาน ซึ่งทำหน้าที่ Pull-Up ภายใน ซึ่งมีผลให้บิตนั้น ๆ ของพอร์ต 1 , 2 และ 3 มีสภาวะลอจิกสูง ตัวต้านทานเหล่านี้มีค่าประมาณ 50K ซึ่งเป็นค่าที่สูงมาก ทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิทของพอร์ต 0 นั้นแม้ว่าจะมีการทำงานที่คล้ายคลึงกับบิตของพอร์ตอื่น ๆ แต่เนื่องจากการที่ไม่มีตัวต้านทาน ทำหน้าที่ Pull-Up ภายในไว้ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นหยุดทำงาน ก็จะเป็นผลให้ขาสัญญาณนี้อยู่ในสภาวะ High Impedance แทน

2.4.2 การใช้งานเป็นพอร์ตเอาท์พุท

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับ ฟลิปฟลอป ซึ่งจะค้างค่านี้เอาไว้และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสภาวะเป็นลอจิกต่ำด้วย ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้นในกรณีที่เป็นการทำงานในแต่ละบิต พอร์ต 1 , 2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณจะเป็นสภาวะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั่นเอง ดังนั้นในการใช้งานพอร์ต 0 เป็นการเอาท์พุทข้อมูล จึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับไฟเลี้ยงแทน

2.5 ฐานเวลาในการทำงานของซีพียู 8051

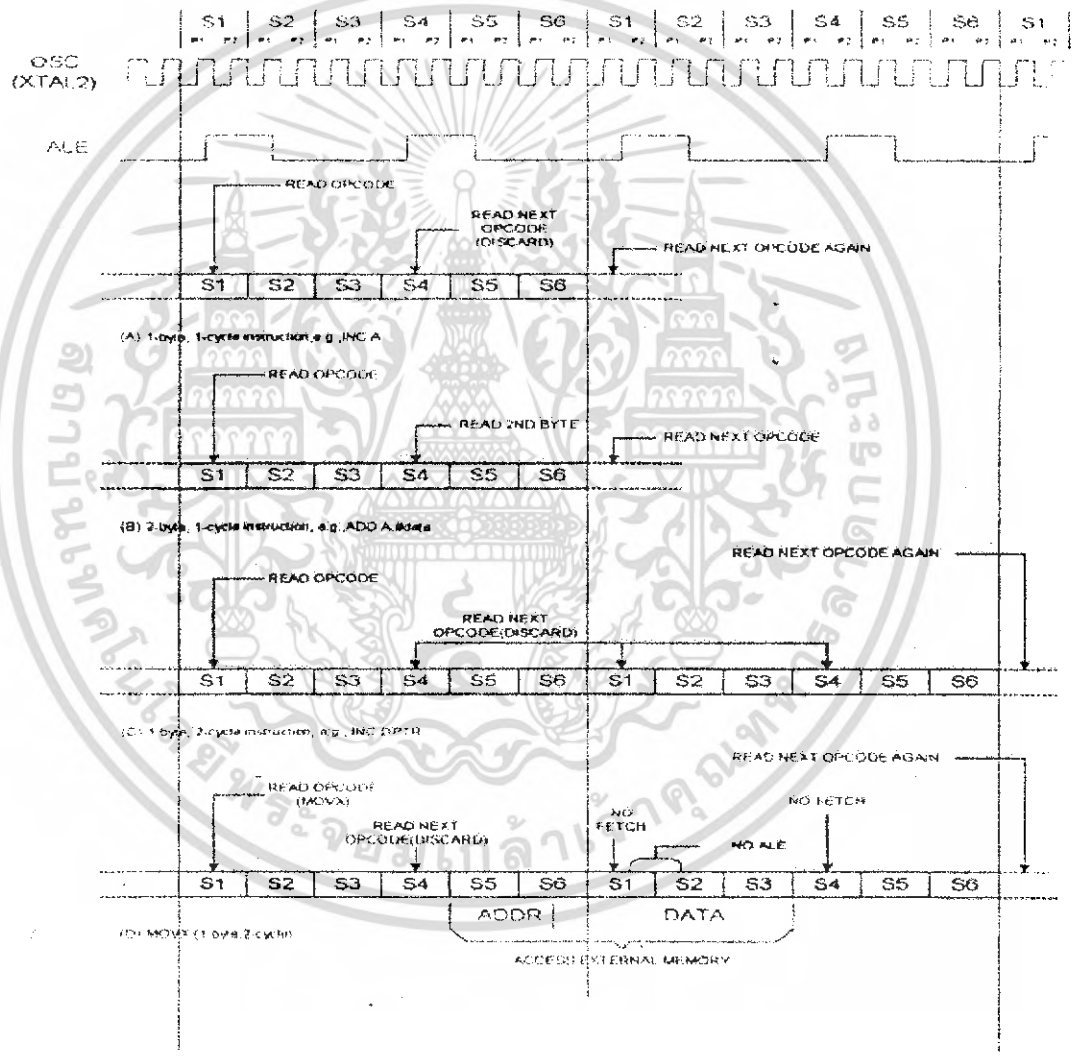
ซีพียูตระกูล MCS-51 มีวงจรออสซิเลเตอร์อยู่ภายใน สำหรับการสร้างพัลซ์ของสัญญาณนาฬิกา ซึ่งจะนำไปเป็นฐานเวลาหรือการกำหนดจังหวะการทำงานของหน่วย การทำงานทั้งหมดให้สอดคล้องกัน (Synchronization) โดยปกติแล้วก็มักจะทำโดยการใช้คริสตอลเชื่อมต่อเข้ากับขาสัญญาณ XTALI และ XTAL2 หรืออาจจะใช้สัญญาณนาฬิกาจากภายนอกก็ได้

พัลซ์ความถี่ของสัญญาณนาฬิกาจะเรียกว่า Pulse และคาบของสัญญาณนาฬิกานี้เรียกว่าคาบเวลาออสซิเลเตอร์ (Oscillator Period) คาบเวลาออสซิเลเตอร์จำนวน 2 คาบเรียกว่า State ซึ่งจะนำไปใช้เป็นช่วงเวลาพื้นฐานการทำงานย่อยของไมโครคอนโทรลเลอร์ ช่วงเวลาของ State จำนวน 6 ครั้งเรียกว่า แมชชีนไซเคิล (Machine Cycle) ดังนั้นค่าเวลาหนึ่งแมชชีนไซเคิลจะใช้เวลา 12 คาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออสซิเลเตอร์ ค่าของแมชชีนไซเคิลนี้จัดเป็นช่วงเวลาที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่งซึ่งหากว่าเป็นคำสั่งที่ซับซ้อนมากก็ต้องใช้เวลานานสองถึงสาม แมชชีนไซเคิล

จากรูปที่ 3-7 จะเห็นว่าหากทำคำสั่งที่ใช้เวลา 1 แมชชีนไซเคิลแล้ว จะมีการอ่านคำสั่งที่สเตท 1 หากคำสั่งมีขนาด 2 ไบท์ ก็จะมีการ Fetch ไบท์ที่สองของคำสั่งที่สเตทที่ 4 ของคำสั่ง ส่วนคำสั่งที่ใช้เวลา 2 แมชชีนไซเคิลนั้น จะมีการอ่าน Opcode ที่สเตทแรก และระหว่างนั้นจะไม่มีการ Fetch คำสั่งถัดไปเข้ามาจนกว่าจะทำคำสั่งนั้นเสร็จสิ้นจึงจะเริ่ม Fetch คำสั่งถัดไป



รูปที่2.7 ไซเคิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำว่า 1 แมชชีนไซเคิล คือช่วงเวลาการทำงาน ตั้งแต่ S1 จนถึง S6 การทำงานในหนึ่งคำสั่งจะใช้เวลาเพียง 1 μs ปรกติแล้วซีพียูจะ Run ด้วยความเร็วเท่ากับ 12 MHz ดังนั้นคล็อค 12 ลูกจะใช้เวลาเท่ากับ 1 μs

2.6 การอินเตอร์เฟสโดยพอร์ต RS-232

ลักษณะของการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกส่งออกมาทีละบิต จากตัวอุปกรณ์ส่งไปยังอุปกรณ์รับช่องสัญญาณในการส่งข้อมูลอาจจะใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารถูกกว่าแบบขนานแต่อัตราการส่งข้อมูลจะช้ากว่าการส่งแบบขนาน ในการส่งข้อมูลแบบอนุกรม ข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็นไบต์ จะถูกทยอยส่งทีละบิต และทางอุปกรณ์รับจะรองรับข้อมูลเข้ามาทีละบิต แล้วมารวมกันเป็นไบต์ซึ่งทางด้านอุปกรณ์รับจะต้องคอยตรวจสอบว่าบิตใดเป็นบิตเริ่มแรกของไบต์นั้น การตรวจสอบจะขึ้นอยู่กับรูปแบบของรหัสบิตที่ใช้ในการรับส่งข้อมูลแบบอนุกรม ระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น จำเป็นต้องมีมาตรฐานในการรับส่ง ซึ่งเป็นมาตรฐานที่นิยมมากที่สุดก็คือมาตรฐาน RS-232

2.6.1 มาตรฐาน RS-232

เพื่อที่จะให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันอย่างกว้างขวางที่สุดคือ RS-232 ซึ่งโดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรมอยู่ในตัวอยู่แล้วและจะทำหน้าที่รับส่งข้อมูลในแบบอนุกรม ตามจุดประสงค์ของมาตรฐาน RS-232C นั้น เพื่อจะสามารถเชื่อมต่อกันระหว่างอุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment : DTE) เช่น พอร์ตของคอมพิวเตอร์หลักหรืออุปกรณ์ปลายทางกับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment : DCE) หมายถึงอุปกรณ์ที่สามารถแปลงรูปคลื่นดิจิตอลไปเป็นสัญญาณที่เหมาะสมสำหรับการส่งผ่านสายโทรศัพท์หรือตัวกลางการสื่อสารอย่างอื่นโดยกระบวนการผสมสัญญาณ (Modulation)

2.6.2 การอินเตอร์เฟสตามมาตรฐาน RS-232C

มาตรฐาน RS-232C ใช้สัญญาณเพียงเส้นเดียวในการส่งสัญญาณโดยสัญญาณที่ส่งไปได้ทิศทางเดียว สำหรับความเร็วและระยะทางของการเชื่อมต่อ RS-232C สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้ถึง 200 Kbps ซึ่งเพียงพอสำหรับไมโครคอนโทรลเลอร์ที่มีอัตรา 110-9600 บอด ความยาวของสายเชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณระหว่าง DTE และ DCE โดยสัญญาณตามมาตรฐานของ RS-232 จำกัดอยู่ที่ 50 ฟุตหรือประมาณ 15 เมตร ซึ่งเพียงพอสำหรับการสื่อสารไมโครคอมพิวเตอร์กับอุปกรณ์กาขนอกสำหรับแรงดันของระดับสัญญาณจะถูกกำหนดลงใน 2 บริเวณ คือ

- แรงดันไฟบวก (สถานะ Space) อยู่ระหว่าง +5 ถึง +15 โวลท์สำหรับเอาต์พุตและระหว่าง +3 ถึง +15 สำหรับอินพุต ความแตกต่างมีไว้เผื่อกรณีที่แรงดันไฟฟ้าสูญหายเนื่องจากความยาวของสัญญาณ
- แรงดันไฟลบ (สถานะ Mark) ถูกกำหนดไว้ระหว่าง –5 ถึง –15 โวลต์ สำหรับเอาต์พุต และระหว่าง –3 ถึง –15 โวลท์สำหรับอินพุต

ซึ่งจะเห็นได้ว่าถ้าให้สายสัญญาณยาวเกินไป ระดับแรงดันไฟฟ้าจะตกลงเกินขอบเขตที่กำหนด นอกจากนี้ค่าความจุไฟฟ้าที่เกิดขึ้นจะมีผลกับคุณภาพของสัญญาณ โดยทำให้การเปลี่ยนสถานจากแรงดันไฟบวกไปเป็นแรงดันไฟลบนั้นไม่ชัดเจน เนื่องจาก RS-232C ไม่ได้ออกแบบให้นำไปใช้กับระยะทางไกล ดังนั้นถ้าอุปกรณ์อยู่ห่างกันมาก อาจจำเป็นต้องใช้โมเด็มหรือวิธีการอื่น ๆ แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและหลักการทำงาน

ระบบเก็บข้อมูลในโครงงานนี้ ประกอบด้วย 2 ส่วนใหญ่ๆ คือ ส่วนควบคุม และส่วนประมวลผล ซึ่งจะมีการประมวลผลทั้งบนไมโครคอนโทรลเลอร์ และบนคอมพิวเตอร์

3.1 การออกแบบภาคควบคุม

- ส่วนควบคุมระบบในที่นี้จะใช้ ไมโครคอนโทรลเลอร์ MCS-51 เป็นตัวควบคุม การทำงานทั้งหมดของระบบ ในส่วนควบคุมนี้จะรวมทั้งส่วนประมวลผลโดยไมโครคอนโทรลเลอร์ ข้อมูลที่ได้จากการสุ่มจะถูกพักไว้ที่หน่วยความจำขนาด 32 กิโลไบท์เพื่อส่งต่อไปประมวลผลยังคอมพิวเตอร์ต่อไป นอกจากนี้ยังใช้ตัวไมโครคอนโทรลเลอร์ ในแปลงสัญญาณที่มาจากพอร์ตอนุกรมเป็นสัญญาณขนาน
- ส่วนแสดงผลบนหน้าจอคอมพิวเตอร์ จะเป็นการนำเอาข้อมูลที่ได้จากอุปกรณ์คาต้าล็อกกอร์มาทำการประมวลผล และ พล็อตกราฟความสัมพันธ์ของค่าแรงดันกับอุณหภูมิ โดยสามารถเลือกดูเส้นกราฟของแต่ละช่องได้ด้วยการป้อนข้อมูลที่คีย์บอร์ดของคอมพิวเตอร์

ซึ่งรายละเอียดของหลักการออกแบบ และ หลักการทำงานของแต่ละส่วนนั้น จะได้กล่าว โดยละเอียดต่อไป

3.1.1 การสร้างฐานเวลาให้กับไมโครคอนโทรลเลอร์

I^2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูล อีกเส้นคือสายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า **สายข้อมูลอนุกรม หรือ SDA** (Serial Data line) ส่วนสายสัญญาณนาฬิกาที่มีชื่อเรียกว่า **สายสัญญาณนาฬิกาอนุกรม หรือ SCL** (Serial Clock line) ในการอธิบายต่อไปนี้จะเรียกสัญญาณทั้งสองว่า สาย **SDA** และ **SCL**

คุณสมบัติทั่วไปของบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง(bi-directory line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5v ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาท์พุตของอุปกรณ์ที่ติดต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collector)

อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติและสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

หลักการของบัส I^2C

บัส I^2C ประกอบด้วยสายสัญญาณ 2 เส้นคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ติดต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงก่อนอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (transmitter)

อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (receiver) อุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (master)

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ใช้ติดต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

1. การถ่ายทอดข้อมูลจะเกิดได้เมื่อบัสว่างเท่านั้น
2. ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

กรอบแยกด้านล่างเป็นการสรุปข้อมูลทางเทคนิคที่สำคัญของระบบบัส I^2C เพื่อใช้เป็นข้อมูลประกอบการเขียนโปรแกรมภาษาซี เพื่อควบคุมให้ไมโครคอนโทรลเลอร์ MCS-51 ติดต่อกับอุปกรณ์บนระบบบัส I^2C ได้

73149
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลพื้นฐานที่สำคัญของบัส I^2C

สภาวะที่เกิดขึ้นบนบัส I^2C มีด้วยกัน 5 สภาวะ ดังนี้

1. บัสว่าง (Bus no busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

2. เริ่มต้นการถ่ายข้อมูล (start data transfer) เกิดเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกว่า สภาวะเริ่มต้น

3. ข้อมูลดำรงอยู่บนบัส (data valid) เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดบนสาย SDA คือข้อมูลที่ทำการถ่ายทอดเมื่อสาย SCL เป็นลอจิกสูง สถานะ SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น "0" หรือ "1" ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดเวลาที่สาย SCL สถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

4. รับรู้ข้อมูล (acknowledge) เกิดหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิต เรียกว่า บิตรับรู้ (acknowledge bit) เป็นสถานะลอจิกสูง หลังการส่งข้อมูลมาครบ ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่อในตอนนั้นจะกำเนิดบิตรับรู้ที่มาลอจิกต่ำเพื่อตอบสนองให้ทราบว่าได้รับข้อมูลเรียบร้อย

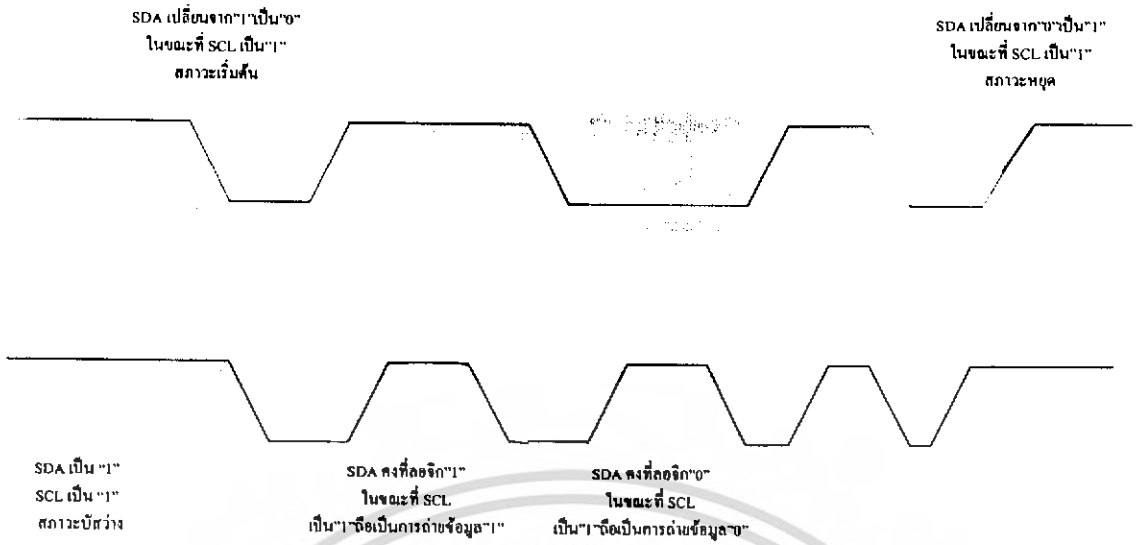
5. หยุดการถ่ายทอดข้อมูล *(stop data transfer) เกิดเมื่อสาย SDA มีการเปลี่ยนแปลงลอจิกจากต่ำไปสูง* ในที่สาย SCL มีลอจิกสูง เรียกสภาวะที่เกิดว่า สภาวะหยุด (STOP)

การทำงานบนบัส I^2C

เริ่มโดยการเข้าถึงอุปกรณ์เสียก่อน โดยการเข้าถึงนั้นจะใช้การเข้าถึงแบบ 7 หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่อบนบัสไม่มาก ใช้การเข้าถึงแบบ 7 บิตก็เพียงพอ แต่ในบางอุปกรณ์ต้องใช้การเข้าถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

การเข้าถึงแบบ 7 บิต (7-bit addressing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ไดอะแกรมเวลาแสดงสถานะต่างๆ บนระบบบัส I^2C

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ โดยมีรูปแบบแสดงในรูป ใน 7 บิตบนรวมทั้ง MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิต เป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะให้ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I^2C ส่วนในบิต SCL เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น “0” หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น “1” จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่าง ไอซีขยาย พอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาท์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง (data)

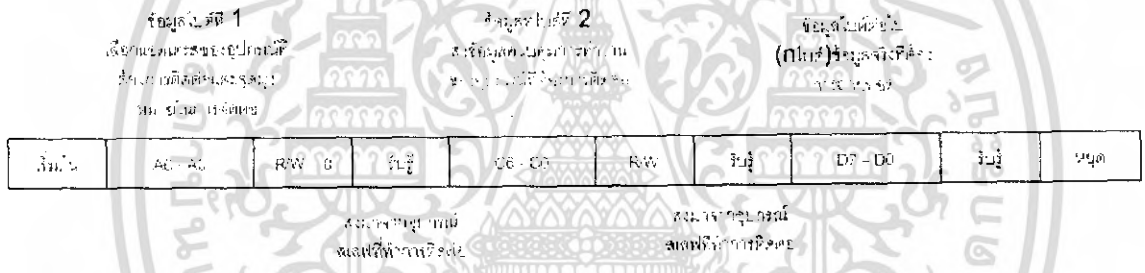
หลังจากถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

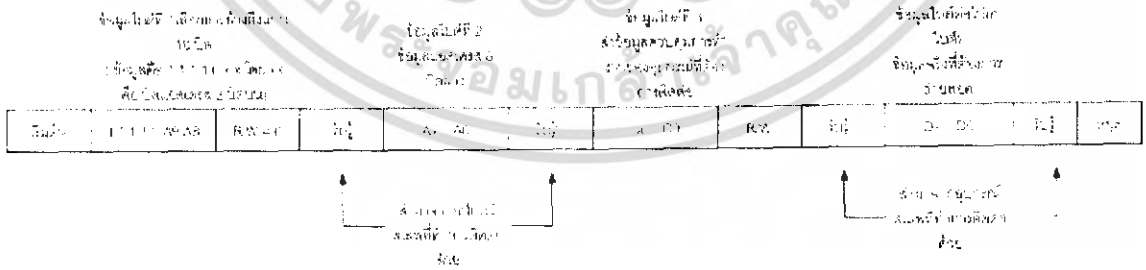
การเข้าถึงแบบ 10 บิต (10-bit addressing)

จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อ

เช่นเดียวกับการเข้าถึงแบบ 7 บิต หลังจากการถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้



รูปที่ 3.2 รูปแบบข้อมูลในการเข้าถึงแบบ 7 บิต

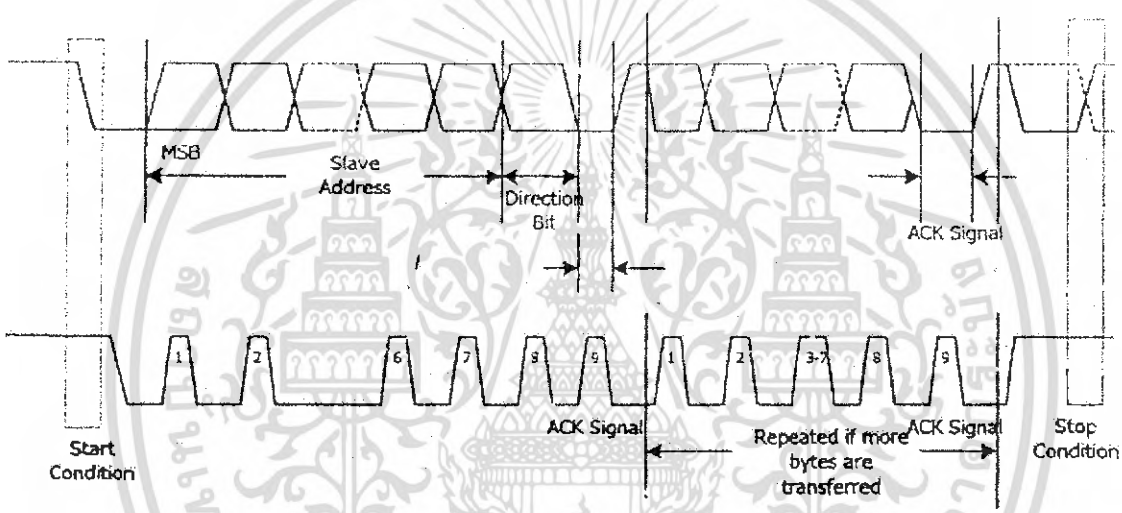


รูปที่ 3.3 รูปแบบข้อมูลในการเข้าถึงแบบ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ DS1307

วงจรออสซิเลเตอร์ ถือเป็นหัวใจหลักของไอซีกำหนดสัญญาณนาฬิกาจริง(Real time clock) เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการเอนเนเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่าคือ 1 Hz,4.096 kHz, 8.192 kHz และ 32 kHz โดยการกำหนดค่า RS0 และ RS1 ดังแสดงในตาราง 3.1 พร้อมกันนั้นจะมีการเก็บค่าของเวลาไว้ ในหน่วยความจำซึ่งมีขนาด 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปอีก 56 ไบต์



รูปที่ 3.4 การส่งข้อมูลบน Serial Bus ของ DS 1307

ตารางที่ 3.1 รีจิสเตอร์ควบคุม (Control Register) ที่ใช้สำหรับควบคุมการทำงานของขา SQW/OUT

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
OUT	X	X	SQWE	X	X	RS1	RS0

- OUT(Output control) บิตนี้ใช้ควบคุมระดับเอาท์พุตของขา SQW/OUT ในกรณีที่มีการดิสเอเบิลการกำเนิดสัญญาณพัลส์ ถ้า SQWE = 0 ระดับสัญญาณลอจิกที่ขา SQW/OUTจะเท่ากับ 0 และจะเท่ากับ 1 เมื่อ SQWE = 1
- SQWE (Square Wave Enable)บิตนี้เมื่อเซตค่าลอจิกเป็น 1 จะเป็นการ Enable เอาท์พุตออสซิเลเตอร์ ความถี่ของสัญญาณพัลส์จะขึ้นอยู่กับค่าของบิต RS0 และRS1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RS(Rate Select) บิตนี้ใช้ในการควบคุมความถี่เอาท์พุทพัลส์ ตารางที่ แสดงถึงรายละเอียดของการเลือกความถี่คลื่นพัลส์โดยบิตRS ทั้งสองตัว
- SDA และSCL (ขา 5 และ ขา6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบ I^2C ซึ่งในที่นี้จะต่อกับ P1.3 และ P 1.4 ตามลำดับ

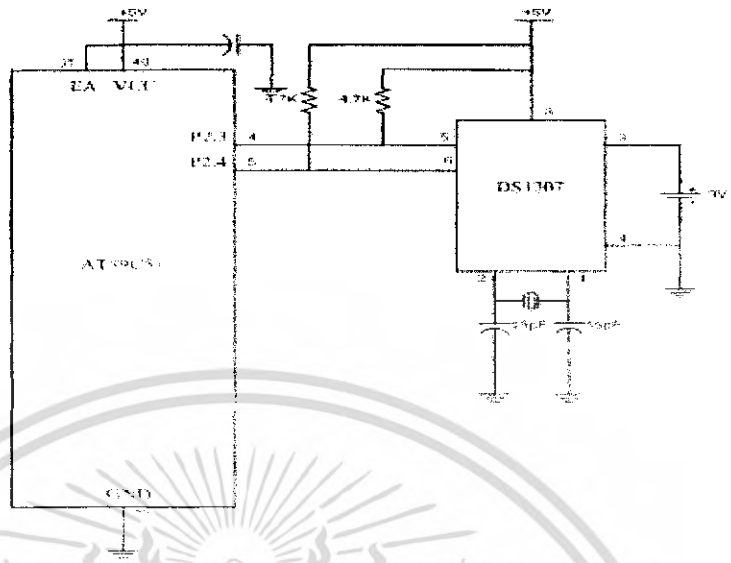
ตารางที่ 3.2 การเลือกความถี่ของสัญญาณพัลส์

RS1	RS0	SQW/OUT Frequency
0	0	1Hz
0	1	4.096Hz
1	0	8.192Hz
1	1	32.768Hz

ในการส่งข้อมูลทุกครั้งจะเริ่มต้นด้วยสภาวะเริ่มต้น (Start Condition) และสิ้นสุดการส่งข้อมูลด้วยสภาวะหยุด (Stop Condition) จำนวนไบต์ของข้อมูลที่ทำการส่งในระหว่างสภาวะเริ่มต้นจนถึงสภาวะหยุดการส่งนั้นจะไม่มีขีดจำกัด แต่ละถูกกำหนดโดยอุปกรณ์ควบคุม ในการส่งไบต์ข้อมูลนั้นอยู่ที่อุปกรณ์รับจะมีการส่งสัญญาณเพื่อตอบรับการส่งข้อมูล (Acknowledge Signal) ไปที่ตัวส่งเพื่อบอกให้รู้ว่าสามารถส่งข้อมูลได้หรือข้อมูลที่ส่งไปได้รับเรียบร้อยโดยอุปกรณ์ตัวรับจะต้องส่งสัญญาณดังกล่าวไปพร้อมกับบิตที่ 9 หลังจากที่มีการรับข้อมูลทุกครั้ง

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสถานะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า 1.25 V_{BAT} ควบคุมให้ DS1307 หยุดการทำงานดังนั้นในการใช้งานต้องระมัดระวังอย่าให้ไฟเลี้ยงต่ำกว่า 1.25V_{BAT} หรือประมาณ 3.75 โวลท์ ในกรณีที่ V_{BAT} เท่ากับ 3 โวลท์ หากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซีจะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยง ปรากฏขึ้นอีกครั้ง DS 1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

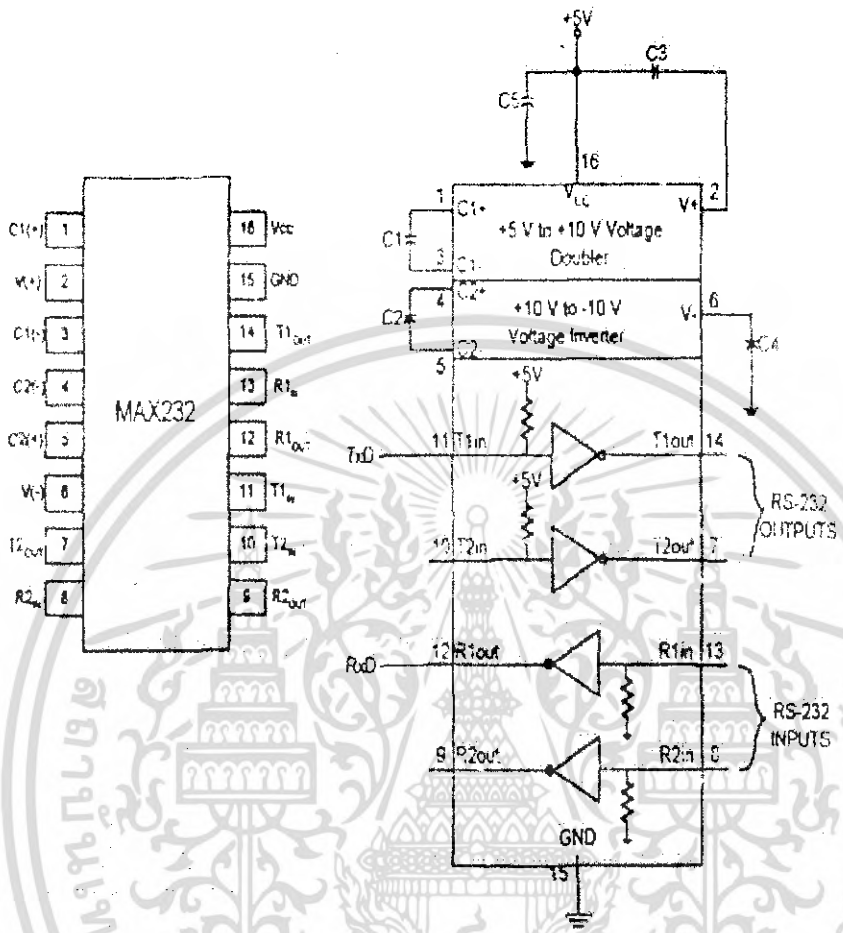


รูปที่ 3.5 การเชื่อมต่อไมโครคอนโทรลเลอร์กับไอซีรีลไทม์คล็อก

3.1.2 การเชื่อมต่อกับพอร์ตอนุกรมคอมพิวเตอร์

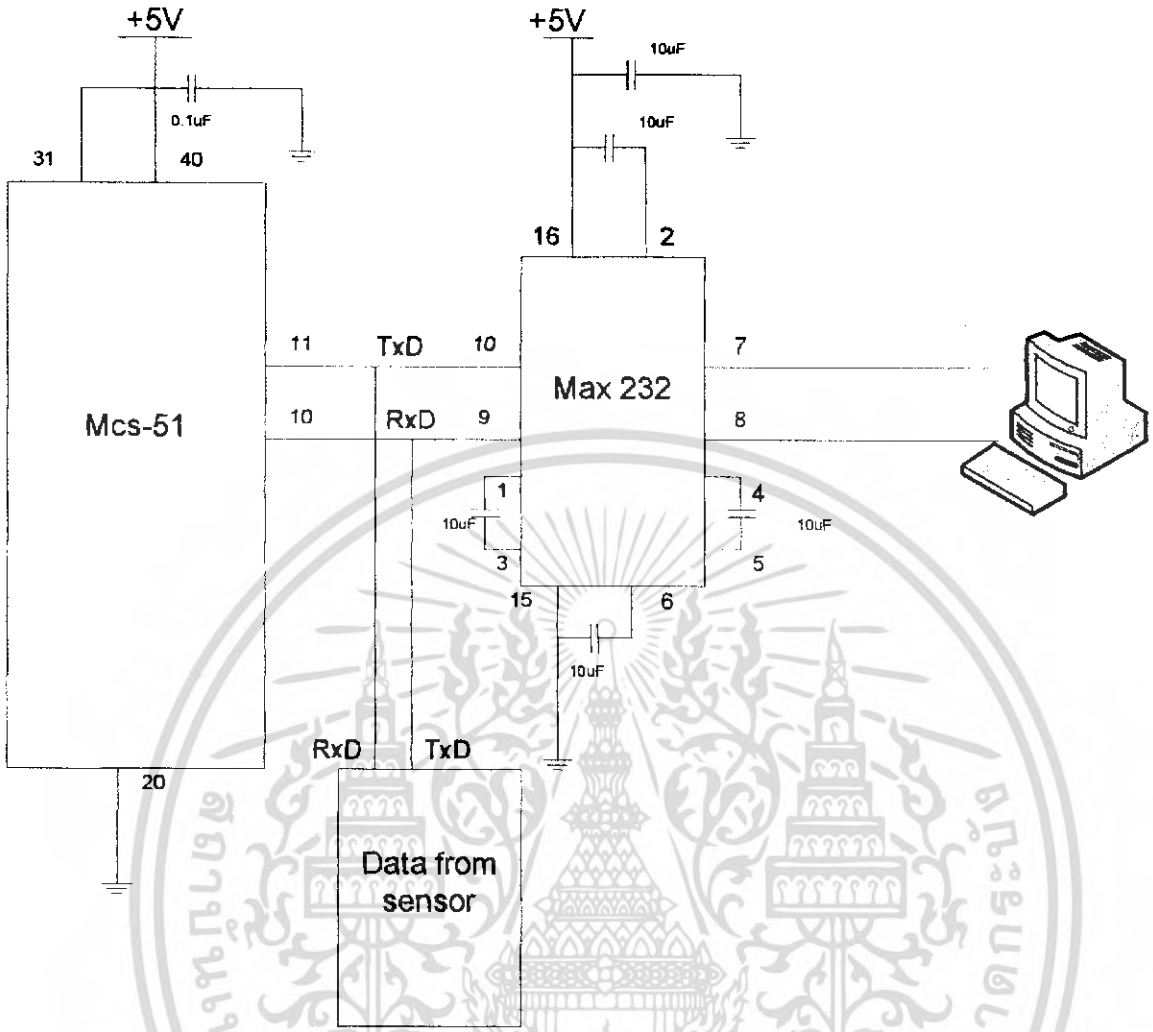
การใช้งานวงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 เป็นส่วนใหญ่แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ ± 3 V ถึง ± 12 V ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์จะอยู่ในระดับทีทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อกันโดยตรงได้ ต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ โครงงานนี้เราจะใช้ไอซี MAX232 ซึ่งทำหน้าที่ในการแปลงระดับสัญญาณข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับทีทีแอล (TTL) ไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับทีทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ได้ รูปที่ 4.7 แสดงให้เห็นถึงการจัดวางตำแหน่งขาสัญญาณและโครงสร้างภายในของไอซี MAX232 ที่จะใช้เชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 การจัดขาและโครงสร้างภายในของ MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

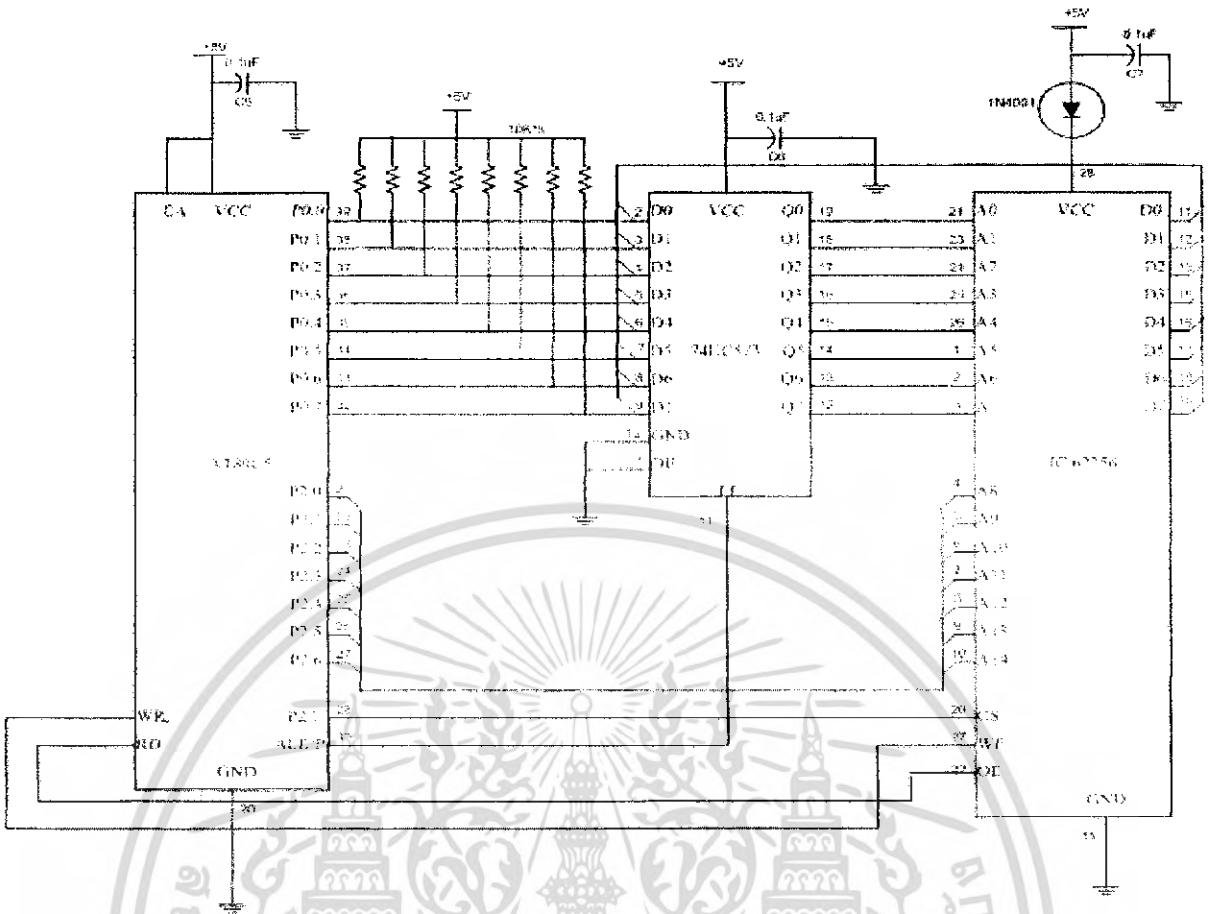


รูปที่ 3.8 การเชื่อมต่อโมดูล MAX-232 กับไมโครคอมพิวเตอร์

3.1.3 การเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำภายนอก

Static Ram เป็นหน่วยความจำชนิดหนึ่งที่ง่ายต่อการใช้งานโดยไอซีที่นำมาต่อเป็นหน่วยความจำภายนอกได้แก่ไอซีเบอร์ 62256 โดยมีหน่วยความจำ 32 กิโลไบต์ ในการคิดค่าขนาดของหน่วยความจำนั้นสามารถคำนวณได้จากขาแอดเดรสของแรม จะเห็นว่าแรมเบอร์ 62256 มีขาแอดเดรสคือ A0-A14 หรือ 15 บิต ขนาดของหน่วยความจำก็จะเท่ากับ 2^{15} ซึ่งเท่ากับ 32 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรใช้งานแรมภายนอก

ในการใช้งานหน่วยความจำภายนอกโดยจากวงจรในรูปที่ 4.8 จะใช้พอร์ต 0 ของไมโครคอนโทรลเลอร์ในการอ้างแอดเดรสไบต์ต่ำ และใช้พอร์ต 2 ในการอ้างแอดเดรสไบต์สูงให้กับแรมภายนอกโดยจะเห็นว่าพอร์ต 0 ต่อเข้ากับพอร์ต Data Bus ของแรมภายนอกซึ่งพอร์ต 0 นั้นไม่สามารถที่จะใช้งานในการอ้างแอดเดรสไบต์ต่ำ และ อ่านเขียนข้อมูลไปยังแรมได้พร้อมกัน ด้วยเหตุนี้จึงได้มีการต่อไอซี 74HC573 ร่วมเข้าไปเพื่อช่วยในการแลตช์แอดเดรสไบต์ต่ำให้กับแรมภายนอกโดยขา ALE ของไมโครคอนโทรลเลอร์จะเป็นตัวควบคุมจังหวะการแลตช์ให้กับไอซี 74HC573 และในขณะที่ไอซี 74HC573 แลตช์แอดเดรสไบต์ต่ำอยู่นั้น พอร์ต 0 จะทำการอ่านเขียนข้อมูลกับแรมภายนอกได้นั้นเอง จากวงจรจะเห็นว่าแรมภายนอกนั้นจะมีสัญญาณควบคุมในการอ่านเขียนข้อมูล โดยใช้ขาสัญญาณ RD และ WR ไมโครคอนโทรลเลอร์เป็นตัวควบคุมในพอร์ต 2 นั้น ไม่มีปัญหาแต่อย่างใดเพราะใช้ในการอ้างแอดเดรสไบต์สูงเพียงหน้าที่เดียวเท่านั้น

3.1.4 การติดต่อกับโมดูล LCD ของไมโครคอนโทรลเลอร์ MSC-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูล LCD เป็นอุปกรณ์แสดงผลแบบหนึ่งที่ได้รับความนิยมในการใช้งานสูงมาก โดยเฉพาะอย่างยิ่งเมื่อนำมาเชื่อมต่อกับไมโครคอนโทรลเลอร์ สำหรับในบทนี้จะนำเสนอการติดต่อระหว่างไมโครคอนโทรลเลอร์MCS-51 กับโมดูล LCD แบบอักษรขนาด 16 ตัวอักษร 2 บรรทัด ด้วยโปรแกรมภาษา C โดยโมดูล LCD ที่ใช้เป็นโมดูล LCD มาตรฐานที่ใช้ชิปเบอร์ HD44870 หรือเทียบเท่าเป็นตัวควบคุมการทำงาน

ข้อมูลเบื้องต้น ของโมดูล LCD

ในรูป แสดงการจัดขาของโมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด

V_{SS} (ขา 1) : ต่อกราวด์

V_{DD} (ขา 2) : ต่อ ไฟเลี้ยง +5V

VO (ขา 3) : เป็นขาอินพุทรับแรงดันเพื่อปรับความเข้มของการแสดงผล

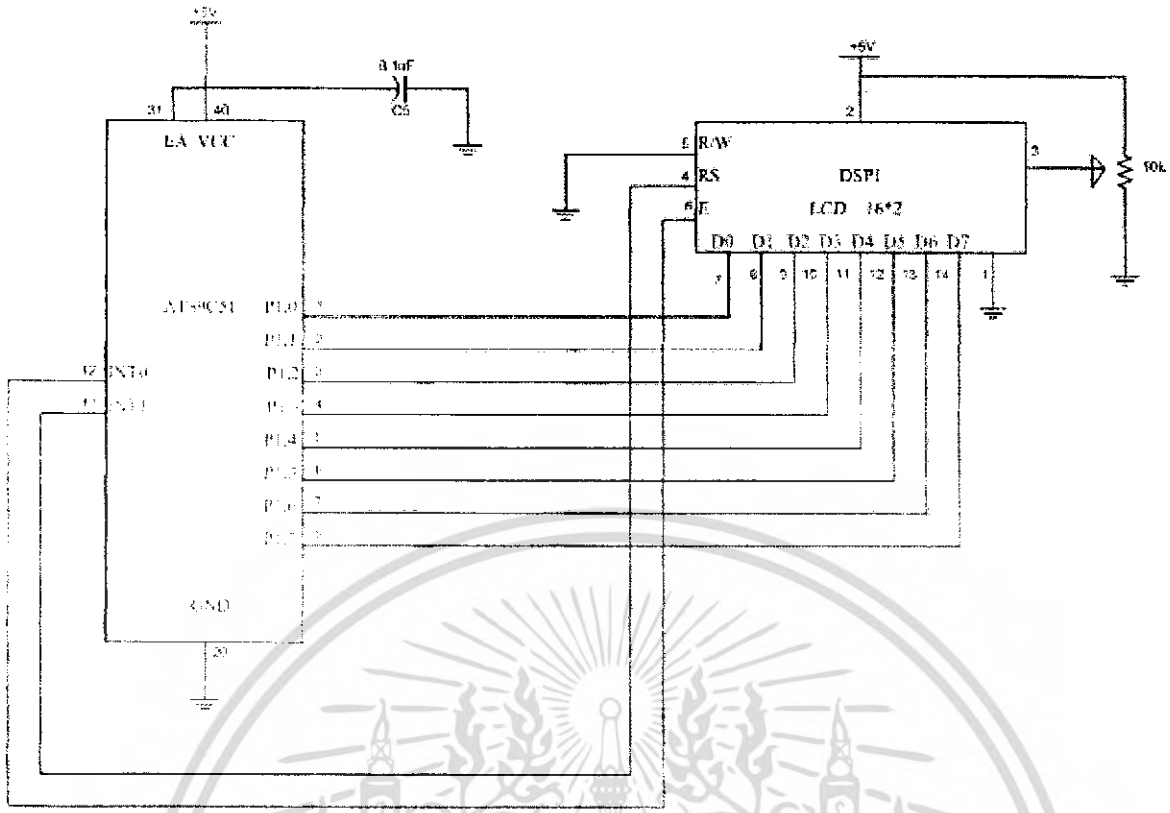
RS (ขา 4) : เป็นขาอินพุทใช้แยกชนิดของข้อมูลที่ประมวลผลว่าเป็นคำสั่งหรือเป็นข้อมูล โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าเป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลแสดงผล

R/W (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ โมดูล LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E(ขา 6) : เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดูล LCD ให้ทำงาน

D0-D7 (ขา 7-14) : เป็นขาข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การเชื่อมต่อโมดูล LCD กับไมโครคอมพิวเตอร์

3.1.5 การแสดงผลทางหน้าจอคอมพิวเตอร์

ส่วนแสดงผลทางหน้าจอคอมพิวเตอร์นั้นจะแสดงข้อมูลของเซ็นเซอร์แต่ละตัวที่ถูกจัดเก็บในดาต้าล็อกเกอร์ โดยเมื่อเราส่งข้อมูลเข้ามาแสดงผลทางพอร์ทอนุกรมแล้วข้อมูลจะถูกจัดเก็บในตารางแสดงผลและนำเอาข้อมูลเหล่านี้มาพล็อตกราฟแสดงความสัมพันธ์ของข้อมูลซึ่งซอร์ฟแวร์ที่ทางผู้จัดทำได้เลือกใช้นั้นคือ Visual Basic 6 สาเหตุที่ผู้จัดทำเลือกใช้โปรแกรมนี้เพราะ

Visual Basic ถือได้ว่าเป็นเครื่องมือที่ใช้ในการเขียนโปรแกรมบน Windows ที่ได้รับความนิยมสูงสุด ทั้งนี้ได้ถูกพัฒนามาอย่างต่อเนื่องหลายเวอร์ชันจนก้าวมาถึงเวอร์ชัน 6.0 ซึ่งได้รับความนิยมสูงสุดทั้งในประเทศและต่างประเทศและยังได้ถูกพัฒนาอย่างต่อเนื่องจากผู้ใช้งานทั่วโลก โดยโปรแกรมนี้มีจุดเด่นเหนือกว่าภาษาเขียนโปรแกรมอื่นๆคือ

1. สร้างแอพพลิเคชั่นได้ง่ายและรวดเร็ว

   Visual Basic 6.0 ได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยในการสร้างแอพพลิเคชั่นได้รวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอพพลิเคชั่นให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Rapid Application Development หรือ RAD

   ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำซากออกไป ขจัดสิ่งที่ไม่จำเป็นต้องรู้เกี่ยวกับการควบคุมฮาร์ดแวร์และการจัดตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ภาษาเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

การเขียนด้วยโปรแกรม Visual Basic 6.0จะเห็นว่า ภาษาBasicใน Visual Basic 6.0 นั้นอ่านง่าย คือ อ่านแล้วใกล้เคียงกับภาษาที่เราใช้งานปกติ อ่านแล้วสื่อความหมายเข้าใจได้ง่ายกว่าภาษาโปรแกรมอื่นๆ ทำให้ผู้ที่เริ่มเขียนโปรแกรมทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

แม้บรรดานักโปรแกรมหลายๆคนจะค่อนขอดว่าภาษา Basic ใน Visual Basic นั้นมีโครงสร้างไม่ค่อยดีนักเมื่อเทียบกับภาษาปาสคาล (Pascal) หรือภาษาจาวา (Java) แต่ต้องไม่ลืมว่าภาษา Basic ใช้เวลาการเรียนรู้ที่สั้นกว่า จึงง่ายต่อการเรียนรู้และเข้าใจของคนทั่วไป

3. รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

นอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic 6.0 ยังมีเครื่องมือช่วยให้การเขียนโปรแกรมเป็นเรื่องไม่ยุ่งยาก เพราะจะมีเครื่องมือช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยาก ตรวจสอบได้อัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์ลดลงได้มาก

นอกจากจะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้วยังมีเครื่องมือที่ใช้ทดสอบและแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิงและขอความช่วยเหลือในจุดที่ข้องใจ

เครื่องมือทั้งหมดที่กล่าวมาถูกจัดไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อๆว่า IDE ซึ่งมาจาก Integrated Development Environment) ทำให้เรียกใช้งานได้สะดวก ตั้งแต่เขียนโปรแกรม ทดสอบ แก้ไข สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆเข้าไปได้เรื่อยๆหรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้ออกเพื่อประหยัดเนื้อที่ฮาร์ดดิสก์

4. มีการพัฒนา ActiveX Control ให้ใช้งานมากมาย

การที่ Visual Basic ได้รับความนิยมอย่างสูง เนื่องจากมีบุคคลหรือบริษัททางด้านพัฒนาซอฟต์แวร์ได้สร้าง ActiveX Control ขึ้นมาใช้งานกับ Visual Basic คลอบคลุมการทำงานหลายด้าน เช่น ฮาร์ดแวร์ กราฟิก ฐานข้อมูล อินเตอร์เนต ซึ่งก็มีอยู่ทั้งในรูปของฟรีแวร์ แชร์แวร์ และซอฟต์แวร์ที่ขายกันจริงๆซึ่งถือเป็นเหตุผลสำคัญที่ทำให้ Visual Basic ถูกใช้งานอย่างแพร่หลายมากกว่า 10ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
สรุปผลการทำงาน

4.1 สรุปผลการทำงาน

จากการทดสอบการทำงานของส่วนประกอบสำคัญในระบบและโปรแกรมรองรับข้อมูล สรุปได้ดังนี้

- เรียลไทม์คล็อกสามารถสร้างสัญญาณอินเตอร์ได้ตามต้องการ
- สามารถรับส่งสัญญาณอนุกรมระหว่างคอมพิวเตอร์กับคอมพิวเตอร์
- สามารถอ่านและเขียนข้อมูลกับหน่วยความจำภายนอก
- หน้าจอแสดงผลสามารถแสดงขั้นตอนการทำงานต่างๆได้อย่างไม่ผิดพลาด
- โปรแกรมสามารถเก็บข้อมูลและแสดงผลข้อมูลออกมาได้

เมื่อนำส่วนประกอบทั้งหมดมาสร้างเป็นระบบเก็บข้อมูลอัตโนมัติ แล้วทดสอบการทำงานทั้งหมด สรุปได้ว่าระบบสามารถทำงานได้ถูกต้องตามฟังก์ชั่น ได้แก่

- สามารถตั้งค่าของเวลาได้ตามที่กำหนด
- สามารถส่งค่าข้อมูลต่างๆเข้าคอมพิวเตอร์ได้จริง
- สามารถเลือกช่องสัญญาณในการเก็บข้อมูลได้
- สามารถเลือกเวลาในการสุ่มเก็บข้อมูลได้

ในส่วนโปรแกรมดาต้าล็อกเกอร์

- โปรแกรมสามารถเก็บข้อมูลและนำผลออกมาแสดงในรูปของตาราง , กราฟและแผ่นรายงานข้อมูลได้

4.2 แนวทางในการพัฒนาและประยุกต์การใช้งาน

การพัฒนาระบบเก็บข้อมูลอัตโนมัติสามารถทำได้หลายอย่าง จึงขอยกตัวอย่างบางประการดังนี้

- การเพิ่มจำนวนช่องเก็บข้อมูล
- การเพิ่มความจุของหน่วยความจำภายนอก
- การเพิ่มความเร็วในการส่งข้อมูลไปยังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ปัญหาในการทำงาน

- ปัญหาทางด้านฮาร์ดแวร์

เนื่องจากการใช้พอร์ตของตัวไมโครคอนโทรลเลอร์เป็นจำนวนมากทำให้การเชื่อมต่อระหว่างแต่ละส่วนอาจมีการสับสนอีกทั้งนักศึกษาทำการเชื่อมไปยังส่วนต่างๆ ได้ไม่ดีพอ ซึ่งสาเหตุเหล่านี้ทำให้ผลการทดลองผิดพลาดได้

- ปัญหาทางด้านซอฟท์แวร์

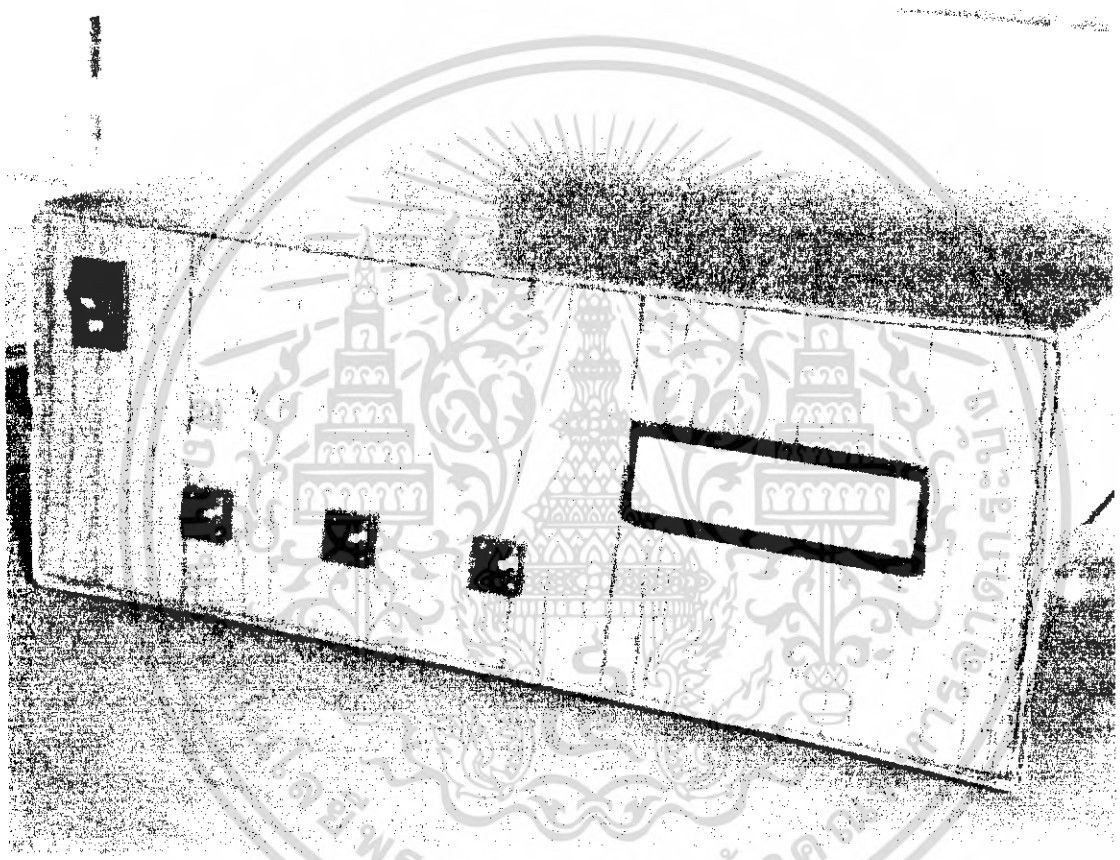
การเขียนโปรแกรมเข้าตัวไมโครคอนโทรลเลอร์เพื่อควบคุมอุปกรณ์อื่นๆ นั้น ในบางส่วนของตัวถูกควบคุมนักศึกษาไม่เข้าใจหลักการทำงานของมันได้ดีจึงไม่สามารถเขียนโปรแกรมควบคุมอุปกรณ์ได้ตามที่ต้องการได้

- ปัญหาในการทำงานร่วมของชิ้นงานระหว่างนักศึกษากับผู้ร่วมโครงงาน

ในการเชื่อมต่อทางพอร์ตอนุกรมระหว่างนักศึกษากับผู้ร่วมโครงงานไม่สามารถสื่อสารกันเพราะความไม่เข้าใจในรูปแบบของการสื่อสาร นักศึกษาจึงทำการจำลองการส่งข้อมูลเข้าทางพอร์ตอนุกรมแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้ดาต้าล็อกเกอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาต้าลอกเกอร์ประกอบด้วย

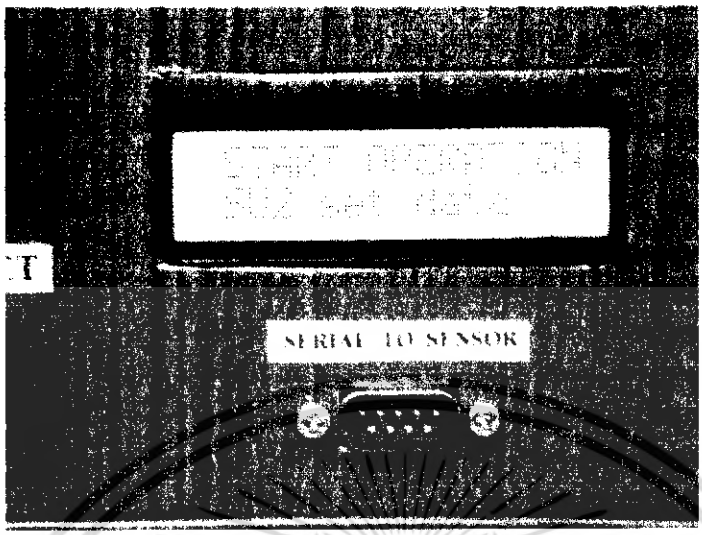
- หน้าจอแสดงผล
- สวิตช์ที่ 1
- สวิตช์ที่ 2
- สวิตช์รีเซ็ต
- สวิตช์เปิด ปิด
- พอร์ตอนุกรมที่ติดต่อกับเซ็นเซอร์และคอมพิวเตอร์

ขั้นตอนการทำงาน

1. เมื่อเปิดเครื่องถ้ามีข้อมูลที่ยังไม่ได้ส่งออก หน้าจอแสดงผลจะปรากฏดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าไม่มีข้อมูลที่ยังไม่ได้ส่งออก หน้าจอแสดงผลจะปรากฏดังรูป



2. กด SW2 เพื่อเริ่มการตั้งข้อมูลต่างๆ

- ตั้งข้อมูลวันเดือนปีและเวลา ณ ปัจจุบันโดยการกด

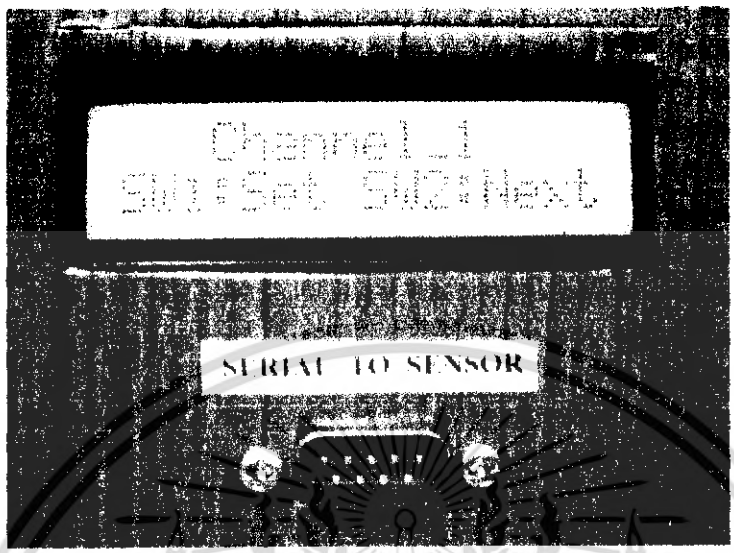
SW1 เพื่อเลื่อนช่องในการเซ็ตข้อมูลของวันเดือนปีและเวลา

SW2 เพื่อเลื่อนตัวเลขให้ตรงกับที่เราต้องการ

ค่าต่าง ๆ ที่กดจะปรากฏดังรูป

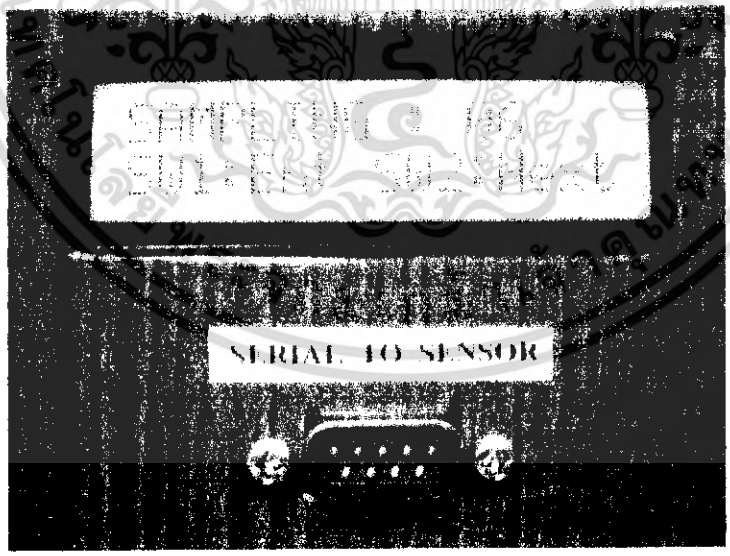
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือกช่องของเซ็นเซอร์ที่จะใช้ซึ่งมีให้เลือกมากสุดถึง 3 ช่อง ดังรูป



กด SW1 จะเข้าไปตั้งค่าเวลาในการซุ่มที่ช่องนั้น
กด SW2 เลื่อนช่องในการตั้งค่าเวลาที่จะสุ่ม

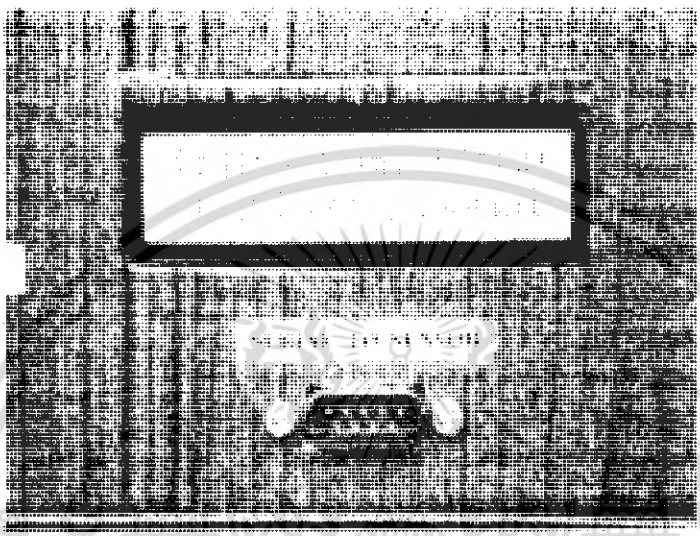
- ตั้งเวลาในการซุ่มข้อมูล ดังรูป



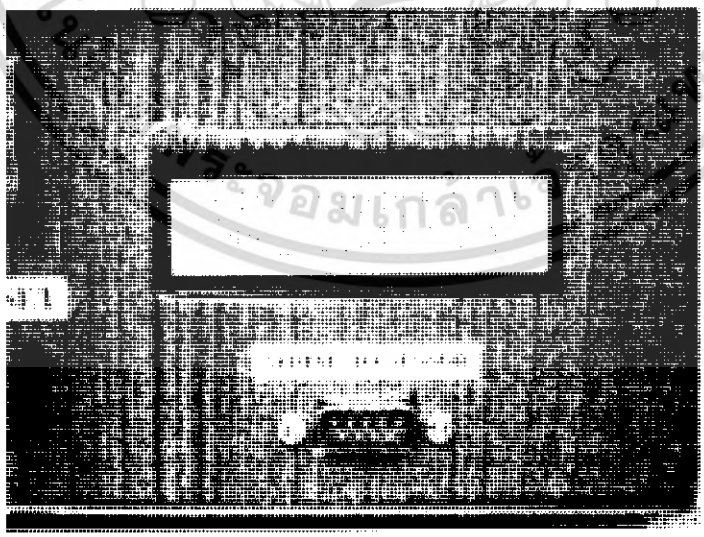
กด SW1 เลือกว่าจะซุ่มเวลาเป็นชั่วโมงหรือนาที
กด SW2 เลือกตัวเลขตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อป้อนข้อมูลต่างๆ ครบจะปรากฏหน้าจอดังรูป



เมื่อเก็บข้อมูลตามที่ต้องการแล้วให้กด SW2 ค้างไว้จนกว่ามีหน้าจอดังต่อไปนี้ปรากฏ



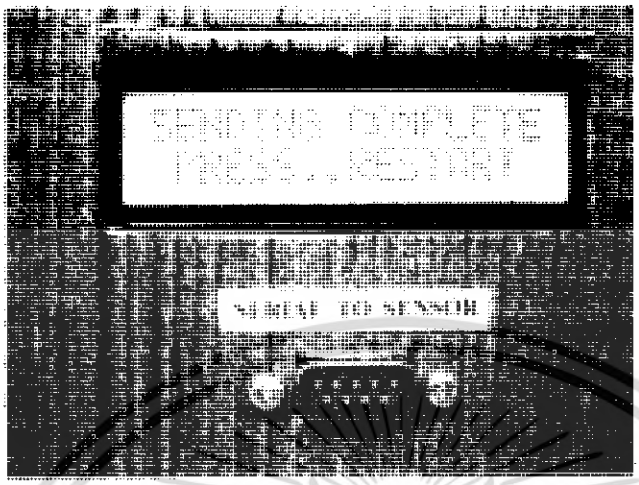
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้รีเซ็ตเครื่องใหม่จะมีหน้าจอปรากฏดังนี้

กด SW1 แล้วข้อมูลจะส่งไปที่คอมพิวเตอร์พร้อมกับมีหน้าจอนี้ปรากฏ

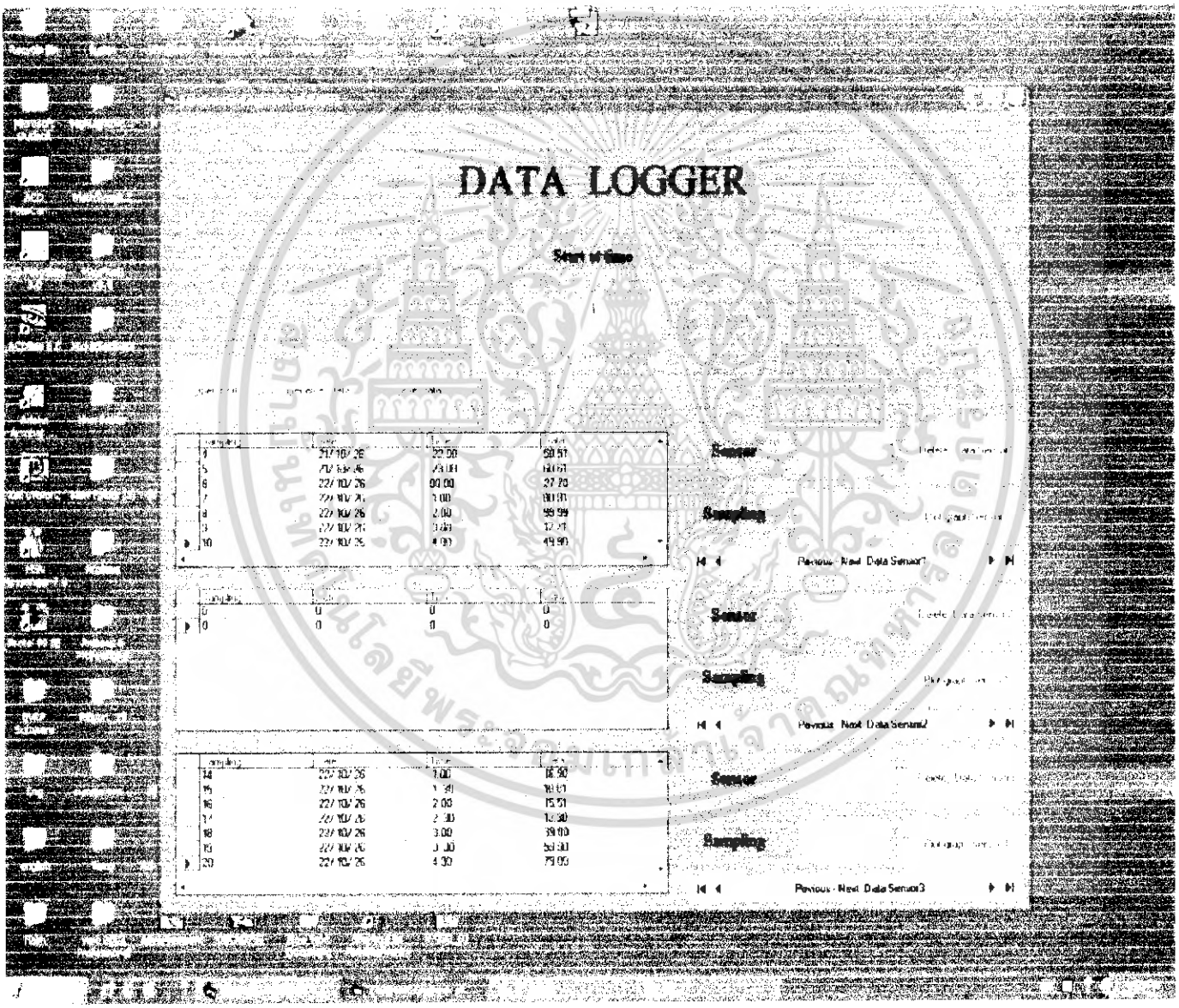
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการส่งเสร็จสิ้นจะมีหน้าจอน์ปรากฏ จากนั้นกดปุ่มรีเซ็ตเริ่มการทำงานใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้โปรแกรมDataLogger



รูปที่ 1 หน้าต่างหลักของโปรแกรมใช้งาน Data Logger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การรับค่าข้อมูลเข้ามาแสดงในตารางและบันทึกลงในดาต้าเบส

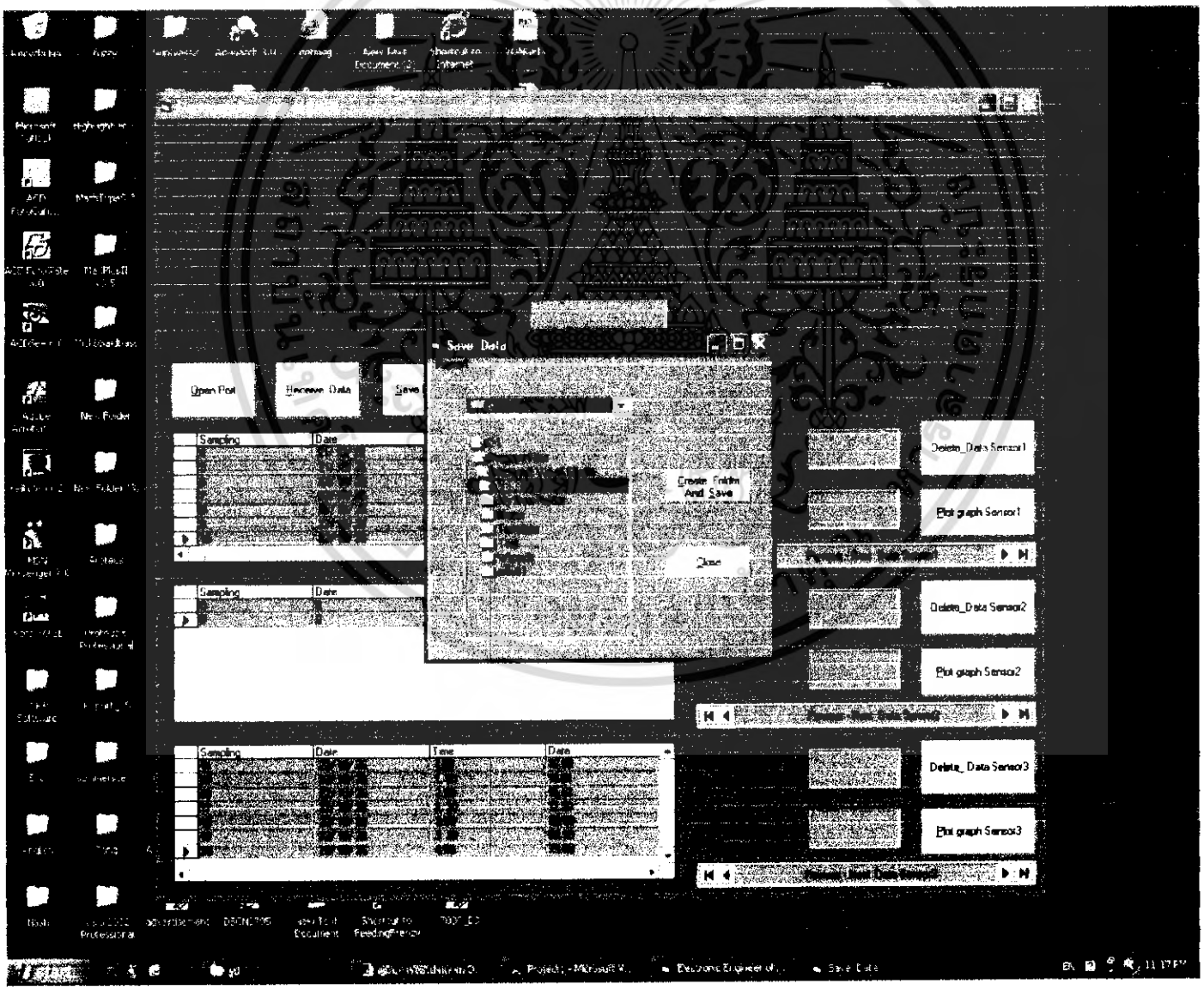
1.1 จากรูปที่ 1 ถ้ายังมีข้อมูลในตารางให้ทำการลบข้อมูลก่อนโดยกดปุ่ม Delete Data แล้วจึงกดปุ่ม Open Port เพื่อเปิดพอร์ทใช้งานและทำการกดปุ่ม SEND DATA เพื่อส่งข้อมูลรอจนหน้าจอเครื่องดาต้าล็อกเกอร์แสดงว่า SENDING COMPLETE จึงกดปุ่ม Receive Data เพื่อรอรับข้อมูล

1.2 ข้อมูลจะแสดงขึ้นที่ตาราง

1.3 ในช่องของ Start at time จะแสดง วัน เดือน ปี และ เวลาที่เริ่มทำการเก็บข้อมูล

1.4 ในช่องของ Sensor จะแสดงชื่อของชนิดเซ็นเซอร์

1.5 ในช่องของ Sampling จะแสดงช่างเวลาของการเก็บแต่ละครั้งของเซ็นเซอร์ชนิดนั้นๆ

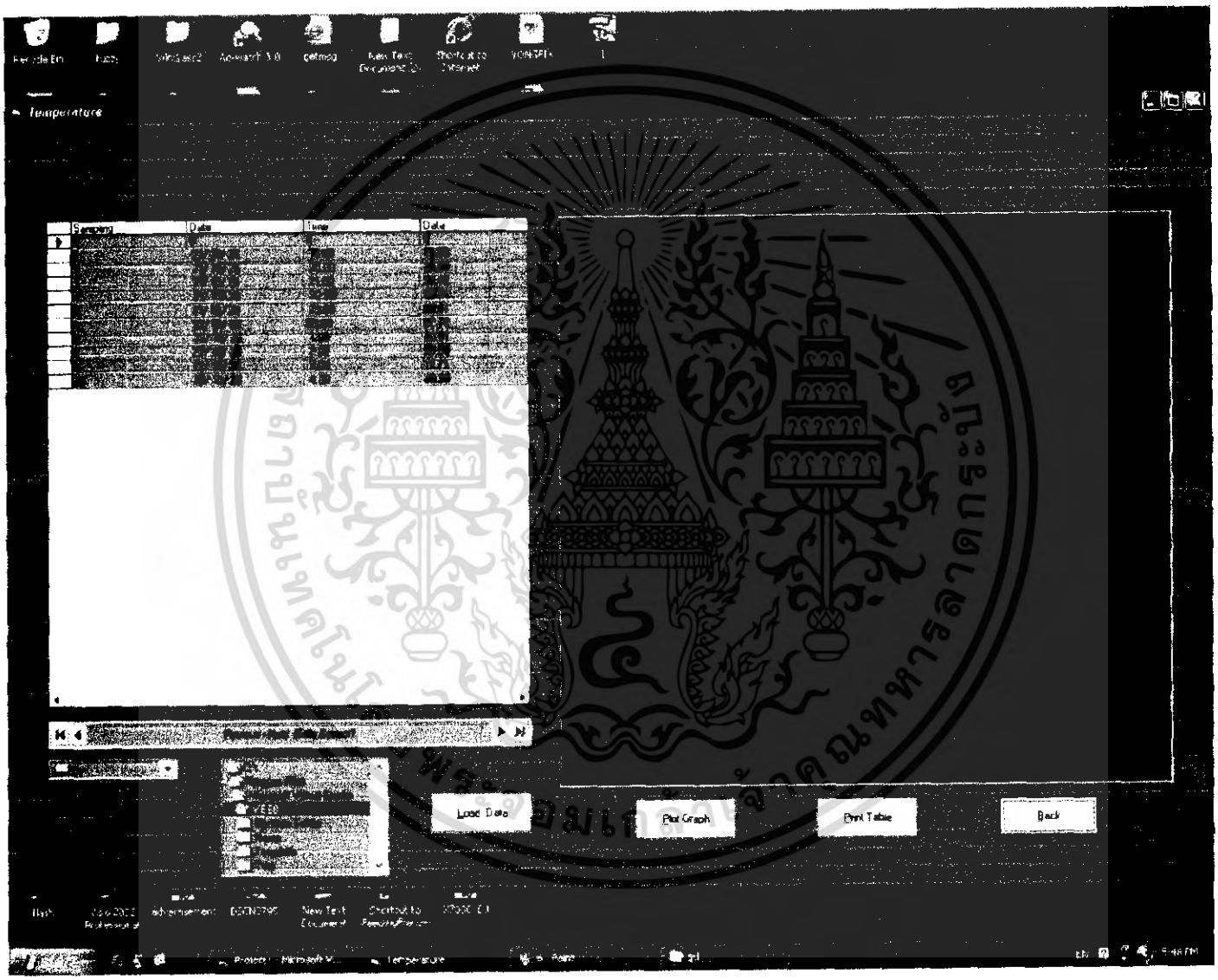


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2 แสดงหน้าจอเมื่อกดปุ่ม Save Data

1.6 ถ้าต้องการบันทึกข้อมูลนี้ไว้ให้กดปุ่ม Save Data แล้วหน้าจอจะแสดงดังรูปที่ 2 ทำการเลือกตำแหน่งที่จะบันทึกแล้วกดปุ่ม Create Folder And Save ตั้งชื่อโฟล์เดอร์และบันทึก

2. การนำข้อมูลมาพล็อตกราฟแสดงแนวโน้มความสัมพันธ์ของข้อมูล



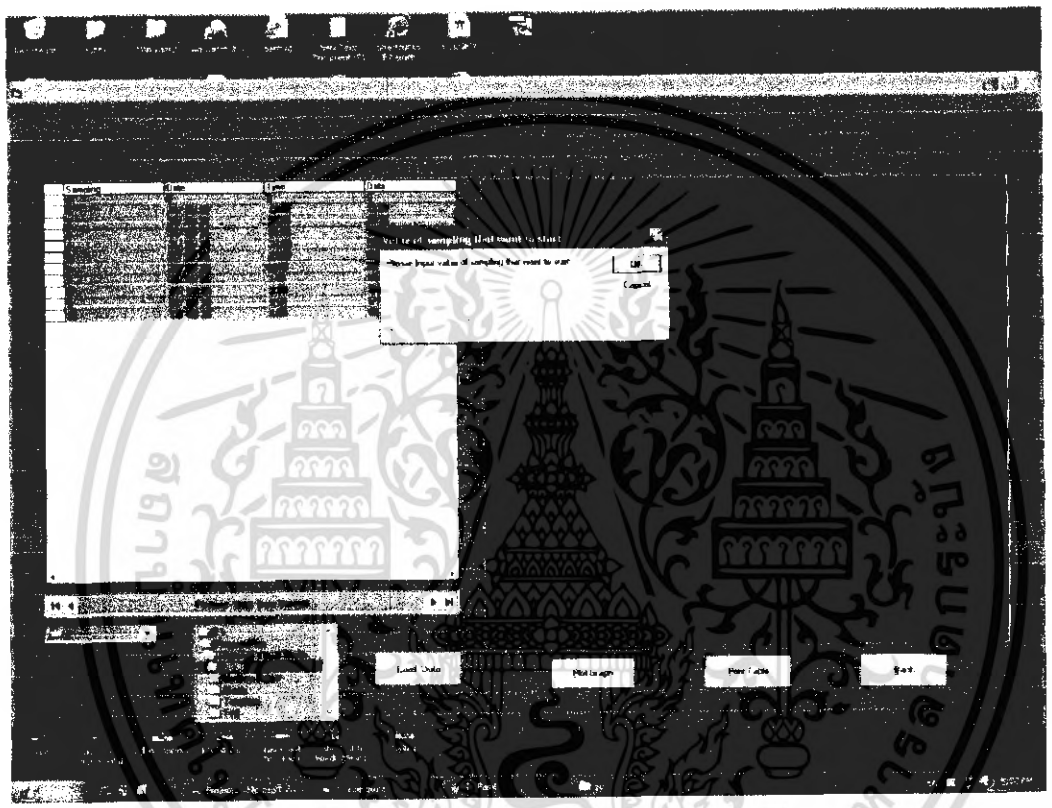
รูปที่ 3 แสดงหน้าจอเมื่อกดปุ่ม Plot Graph ที่หน้าจอหลักในรูปที่ 1

2.1 เมื่อกดปุ่ม Plot Graph ที่หน้าจอหลักหน้าจอดังรูปที่ 3 จะแสดงขึ้นมา

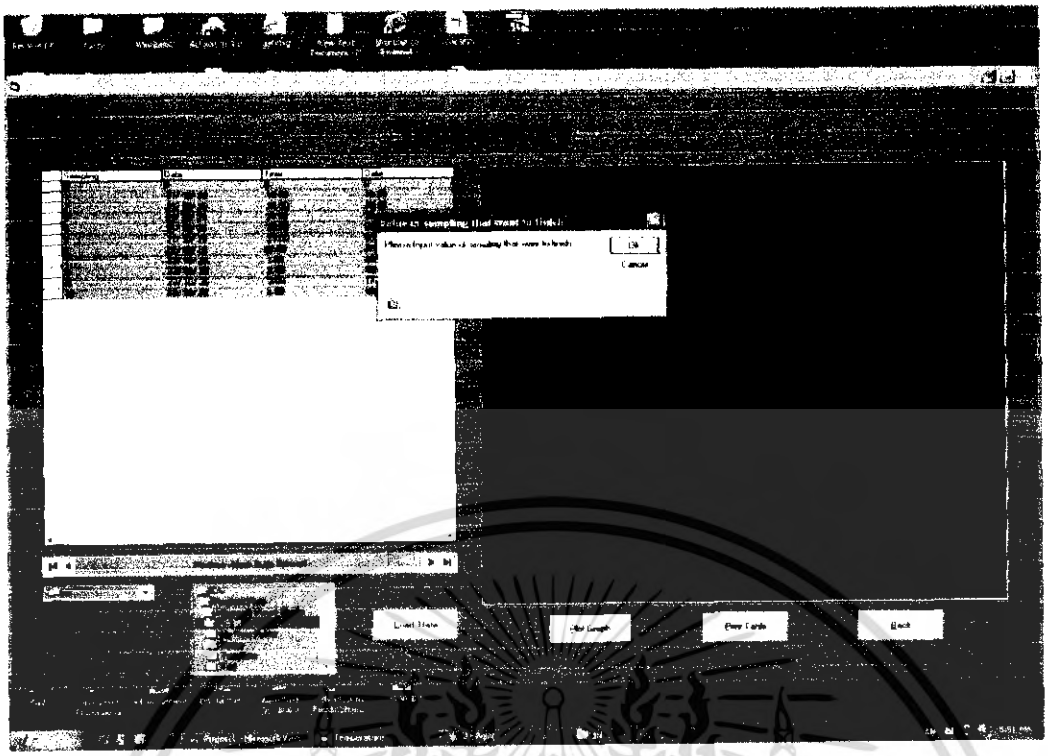
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ถ้าต้องการจะเลือกข้อมูลอื่นที่ไม่ได้แสดงอยู่ในตารางให้ทำการเลือกตำแหน่งของข้อมูลที่ต้องการจะนำมาพล็อตกราฟแล้วกดปุ่ม Load Data ข้อมูลชุดใหม่ก็จะมาแสดงที่ตาราง

2.3 กดปุ่ม Plot Graph เพื่อแสดงกราฟของข้อมูลในตาราง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 และ 5 แสดงหน้าจอหลังจากกดปุ่ม Plot Graph ในหน้าจอของรูปที่ 3

2.4 รูปที่ 4 แสดงหน้าจอหลังจากกดปุ่ม Plot Graph ให้ใส่ค่า Sampling เริ่มต้นที่จะนำมาพล็อตกราฟแล้วกดปุ่ม OK จะแสดงหน้าจอดังรูปที่ 5 ให้ใส่ค่า Sampling สุดท้ายที่จะนำมาพล็อตกราฟแล้วกดปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

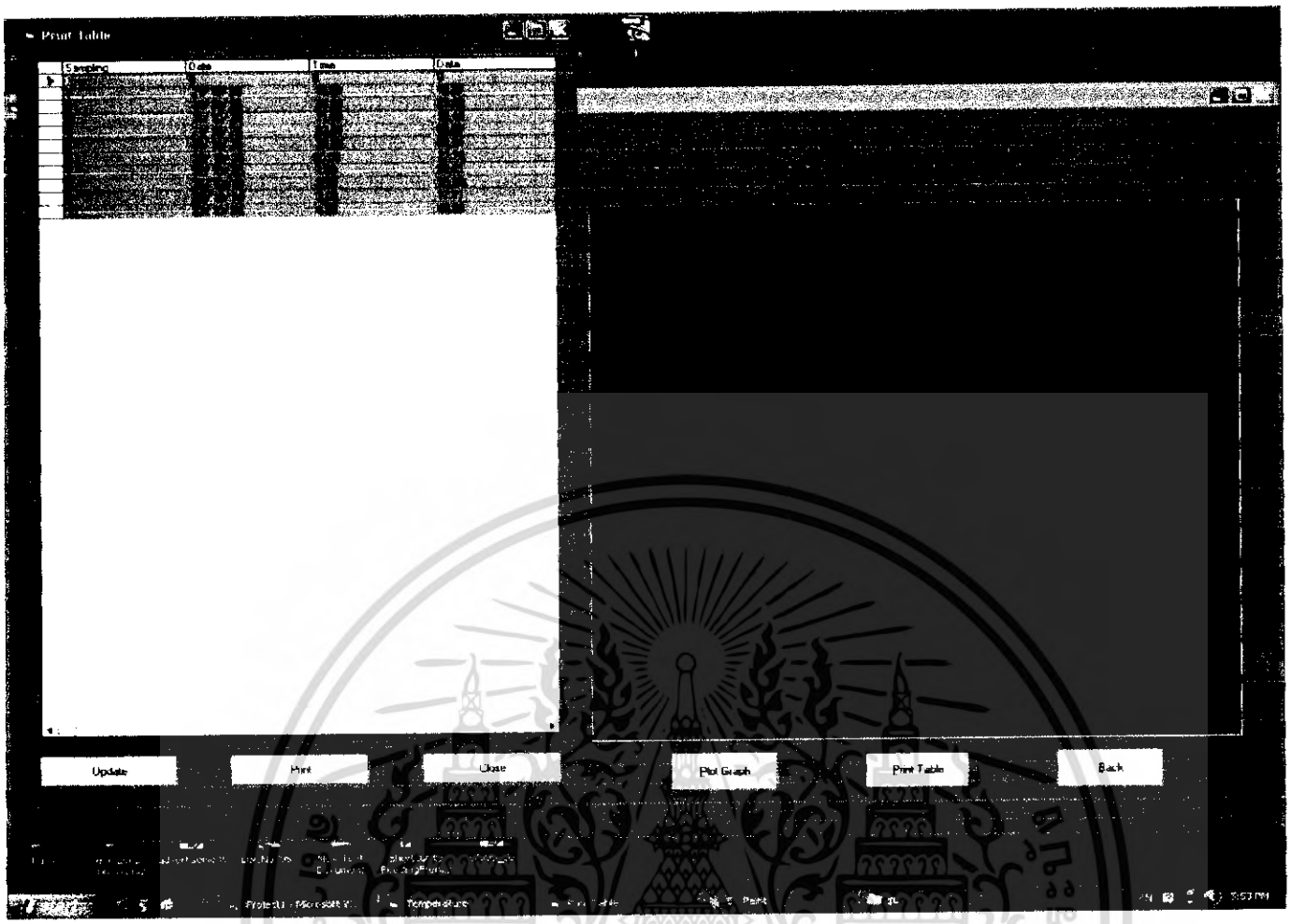
รูปที่ 6 แสดงกราฟที่ได้จากข้อมูลในตาราง

2.5 จะได้กราฟดังรูปที่ 6

3. การนำข้อมูลจากตารางมาแสดงภายนอกผ่านเครื่อง Printer

3.1 กดปุ่ม Print Table จะแสดงหน้าจอดังรูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 แสดงหน้าจอของตารางที่จะPrint

3.2 ถ้าตารางยังไม่เปลี่ยนให้กดปุ่ม Update แล้วกดปุ่มPrint ข้อมูลก็จะถูกพิมพ์ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-4323; Rev 9; 4/00

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

Features

Superior to Bipolar

- **Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)**
- **Low-Power Receive Mode in Shutdown (MAX223/MAX242)**
- **Meet All EIA/TIA-232E and V.28 Specifications**
- **Multiple Drivers and Receivers**
- **3-State Driver and Receiver Outputs**
- **Open-Line Detection (MAX243)**

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	4.7/10	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	6/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

For free samples & the latest literature: http://www.maxim-ic.com, or phone 1-800-998-8800.
For small orders, phone 1-800-835-8769.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CYPRESS

CY62256

256K (32K x 8) Static RAM

Features

- Temperature Ranges
  - —Commercial: 0°C to 70°C
  - —Industrial: –40°C to 85°C
  - —Automotive: –40°C to 125°C
- High speed: 55 ns and 70 ns
- Voltage range: 4.5V–5.5V operation
- Low active power (70 ns, LL version, Com'l and Ind'l)
  - —275 mW (max.)
- Low standby power (70 ns, LL version, Com'l and Ind'l)
  - —28 µW (max.)
- Easy memory expansion with $\overline{CE}$ and $\overline{OE}$ features
- TTL-compatible inputs and outputs
- Automatic power-down when deselected
- CMOS for optimum speed/power
- Package available in a standard 450-mil-wide (300-mil body width) 28-lead narrow SOIC, 28-lead TSOP-1, 28-lead reverse TSOP-1, and 600-mil 28-lead PDIP packages

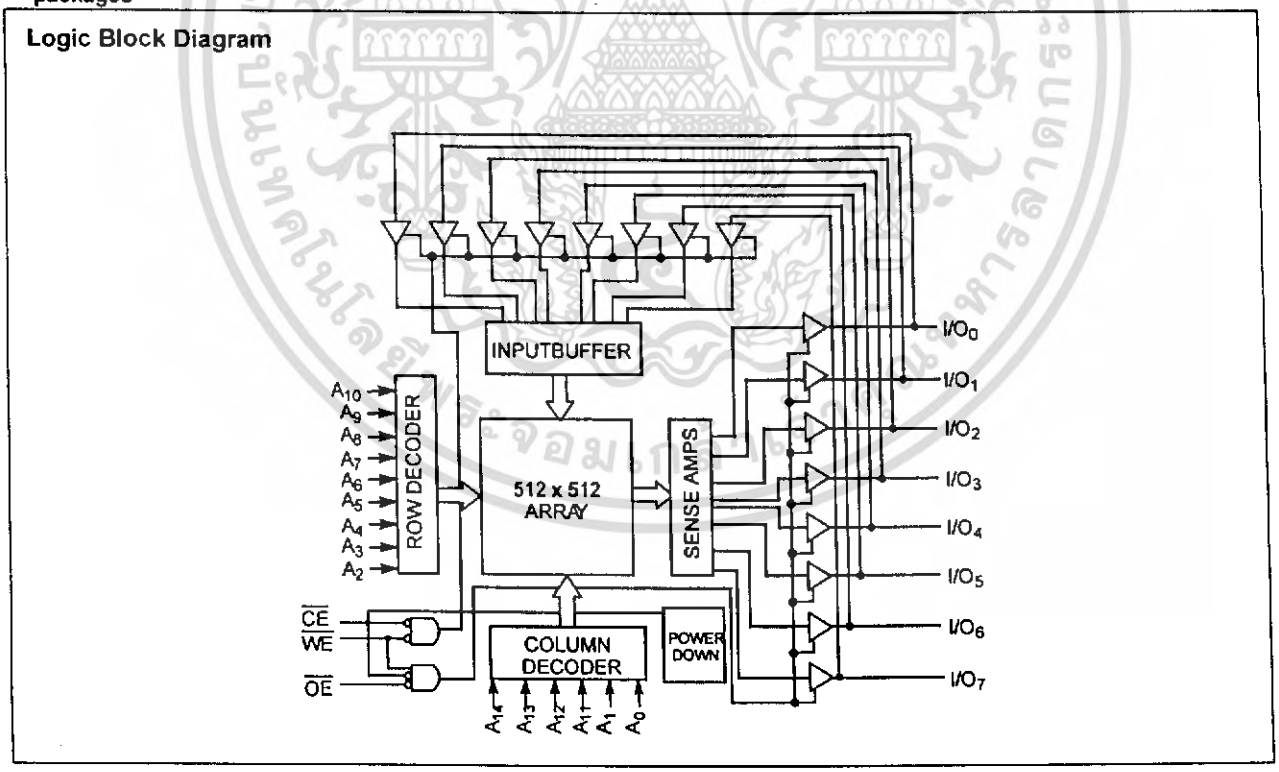
Functional Description[1]

The CY62256 is a high-performance CMOS static RAM organized as 32K words by 8 bits. Easy memory expansion is provided by an active LOW chip enable ($\overline{CE}$) and active LOW output enable ($\overline{OE}$) and three-state drivers. This device has an automatic power-down feature, reducing the power consumption by 99.9% when deselected.

An active LOW write enable signal ($\overline{WE}$) controls the writing/reading operation of the memory. When $\overline{CE}$ and $\overline{WE}$ inputs are both LOW, data on the eight data input/output pins (I/O_0 through I/O_7) is written into the memory location addressed by the address present on the address pins (A_0 through A_{14}). Reading the device is accomplished by selecting the device and enabling the outputs, $\overline{CE}$ and $\overline{OE}$ active LOW, while $\overline{WE}$ remains inactive or HIGH. Under these conditions, the contents of the location addressed by the information on address pins are present on the eight data input/output pins.

The input/output pins remain in a high-impedance state unless the chip is selected, outputs are enabled, and write enable ($\overline{WE}$) is HIGH.

Logic Block Diagram



Note:
1. For best practice recommendations, please refer to the Cypress application note "System Design Guidelines" on http://www.cypress.com.

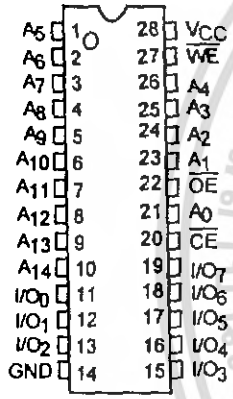
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Product Portfolio

Product		V_{CC} Range (V)			Speed (ns)	Power Dissipation			
						Operating, I_{CC} (mA)		Standby, I_{SB2} (µA)	
		Min.	Typ.[2]	Max.		Typ.[2]	Max.	Typ.[2]	Max.
CY62256	Commercial	4.5	5.0	5.5	70	28	55	1	5
CY62256L	Com'l / Ind'l				55/70	25	50	2	50
CY62256LL	Commercial				70	25	50	0.1	5
CY62256LL	Industrial				55/70	25	50	0.1	10
CY62256LL	Automotive				55	25	50	0.1	15

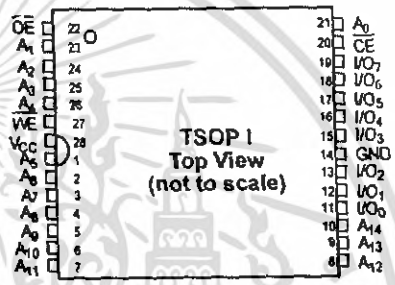
Pin Configurations

Narrow SOIC
Top View

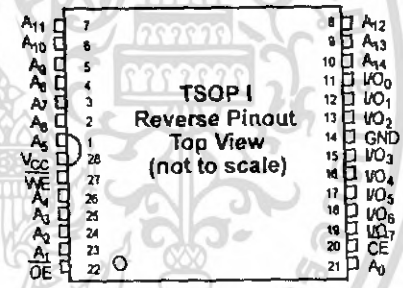


DIP
Top View

TSOP I
Top View
(not to scale)



TSOP I
Reverse Pinout
Top View
(not to scale)



Pin Definitions

Pin Number	Type	Description
1-10, 21, 23-26	Input	A_0-A_{14}. Address Inputs
11-13, 15-19,	Input/Output	I/O_0-I/O_7. Data lines. Used as input or output lines depending on operation
27	Input/Control	$\overline{WE}$. When selected LOW, a WRITE is conducted. When selected HIGH, a READ is conducted
20	Input/Control	$\overline{CE}$. When LOW, selects the chip. When HIGH, deselects the chip
22	Input/Control	$\overline{OE}$. Output Enable. Controls the direction of the I/O pins. When LOW, the I/O pins behave as outputs. When deasserted HIGH, I/O pins are three-stated, and act as input data pins
14	Ground	GND. Ground for the device
28	Power Supply	Vcc. Power supply for the device

Notes:
2. Typical specifications are the mean values measured over a large sample size across normal production process variations and are taken at nominal conditions (T_A = 25°C, V_{CC}). Parameters are guaranteed by design and characterization, and not 100% tested.

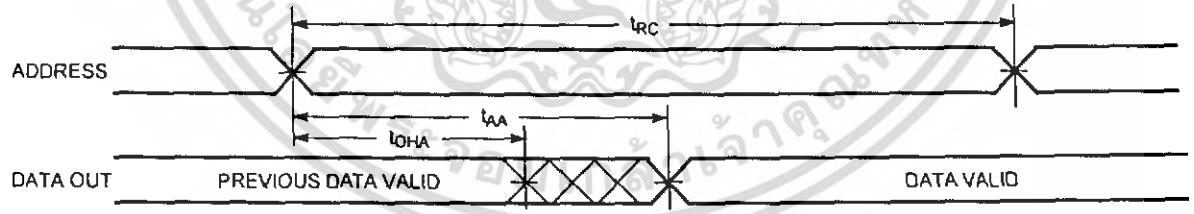
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Characteristics Over the Operating Range[7]

Parameter	Description	CY62256-55		CY62256-70		Unit
		Min.	Max.	Min.	Max.	
Read Cycle						
t_{RC}	Read Cycle Time	55		70		ns
t_{AA}	Address to Data Valid		55		70	ns
t_{OHA}	Data Hold from Address Change	5		5		ns
t_{ACE}	CE LOW to Data Valid		55		70	ns
t_{DOE}	OE LOW to Data Valid		25		35	ns
t_{LZOE}	OE LOW to Low-Z[8]	5		5		ns
t_{HZOE}	OE HIGH to High-Z[8, 9]		20		25	ns
t_{LZCE}	CE LOW to Low-Z[8]	5		5		ns
t_{HZCE}	CE HIGH to High-Z[8, 9]		20		25	ns
t_{PU}	CE LOW to Power-up	0		0		ns
t_{PD}	CE HIGH to Power-down		55		70	ns
Write Cycle[10, 11]						
t_{WC}	Write Cycle Time	55		70		ns
t_{SCE}	CE LOW to Write End	45		60		ns
t_{AW}	Address Set-up to Write End	45		60		ns
t_{HA}	Address Hold from Write End	0		0		ns
t_{SA}	Address Set-up to Write Start	0		0		ns
t_{PWE}	WE Pulse Width	40		50		ns
t_{SD}	Data Set-up to Write End	25		30		ns
t_{HD}	Data Hold from Write End	0		0		ns
t_{HZWE}	WE LOW to High-Z[8, 9]		20		25	ns
t_{LZWE}	WE HIGH to Low-Z[8]	5		5		ns

Switching Waveforms

Read Cycle No. 1[12, 13]



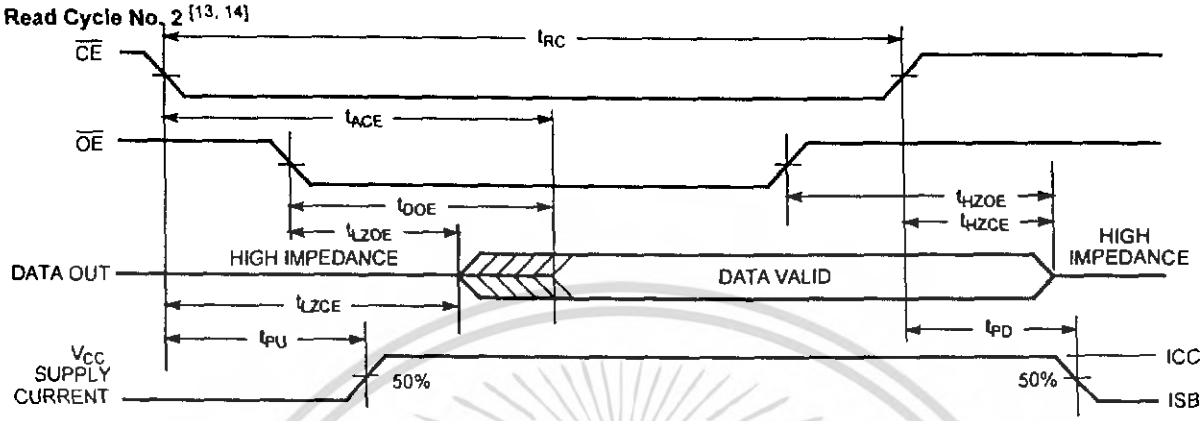
Notes:

7. Test conditions assume signal transition time of 5 ns or less, timing reference levels of 1.5V, input pulse levels of 0 to 3.0V, and output loading of the specified I_{OL}/I_{OH} and 100-pF load capacitance.
8. At any given temperature and voltage condition, t_{HZCE} is less than t_{LZCE}, t_{HZOE} is less than t_{LZOE}, and t_{HZWE} is less than t_{LZWE} for any given device.
9. t_{HZOE}, t_{HZCE}, and t_{HZWE} are specified with C_L = 5 pF as in (b) of AC Test Loads. Transition is measured ±500 mV from steady-state voltage.
10. The internal Write time of the memory is defined by the overlap of $\overline{CE}$ LOW and $\overline{WE}$ LOW. Both signals must be LOW to initiate a Write and either signal can terminate a Write by going HIGH. The data input set-up and hold timing should be referenced to the rising edge of the signal that terminates the Write.
11. The minimum Write cycle time for Write cycle #3 ($\overline{WE}$ controlled, $\overline{OE}$ LOW) is the sum of t_{HZWE} and t_{SD}.
12. Device is continuously selected. $\overline{OE}$, $\overline{CE}$ = V_{IL}.
13. $\overline{WE}$ is HIGH for Read cycle.

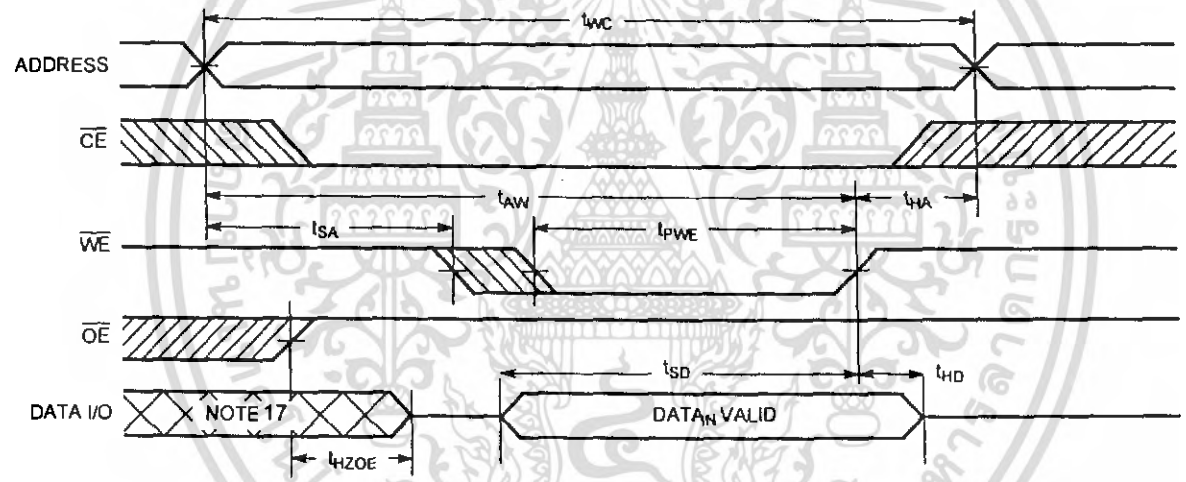
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Waveforms (continued)

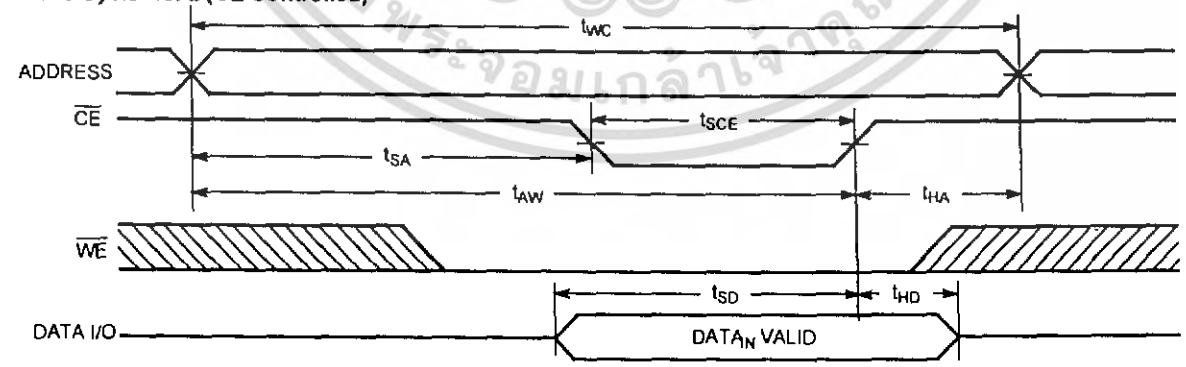
Read Cycle No. 2 [13, 14]



Write Cycle No. 1 (WE Controlled) [10, 15, 16]



Write Cycle No. 2 (CE Controlled) [10, 15, 16]



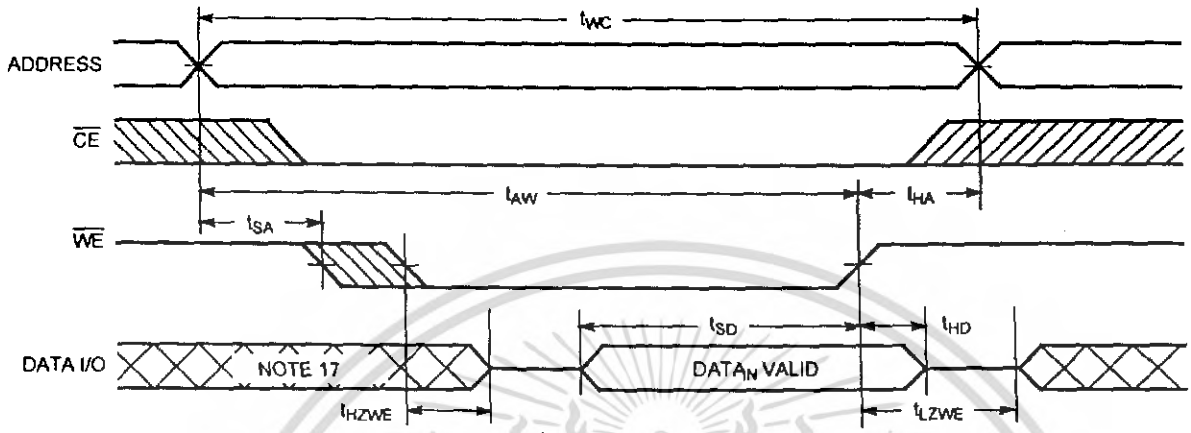
Notes:

14. Address valid prior to or coincident with $\overline{CE}$ transition LOW.
15. Data I/O is high impedance if $\overline{OE} = V_{IH}$.
16. If $\overline{CE}$ goes HIGH simultaneously with $\overline{WE}$ HIGH, the output remains in a high-impedance state.
17. During this period, the I/Os are in output state and input signals should not be applied.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Waveforms (continued)

Write Cycle No. 3 ($\overline{WE}$ Controlled, $\overline{OE}$ LOW) [11, 16]



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

www.dalsemi.com

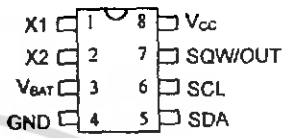
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode with oscillator running
- Optional industrial temperature range -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Recognized by Underwriters Laboratory

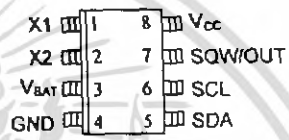
ORDERING INFORMATION

DS1307	8-Pin DIP
DS1307Z	8-Pin SOIC (150 mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

PIN ASSIGNMENT



DS1307 8-Pin DIP (300 mil)



DS1307Z 8-Pin SOIC (150 mil)

PIN DESCRIPTION

V_{CC}	- Primary Power Supply
X1, X2	- 32.768 kHz Crystal Connection
V_{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square wave/Output Driver

DESCRIPTION

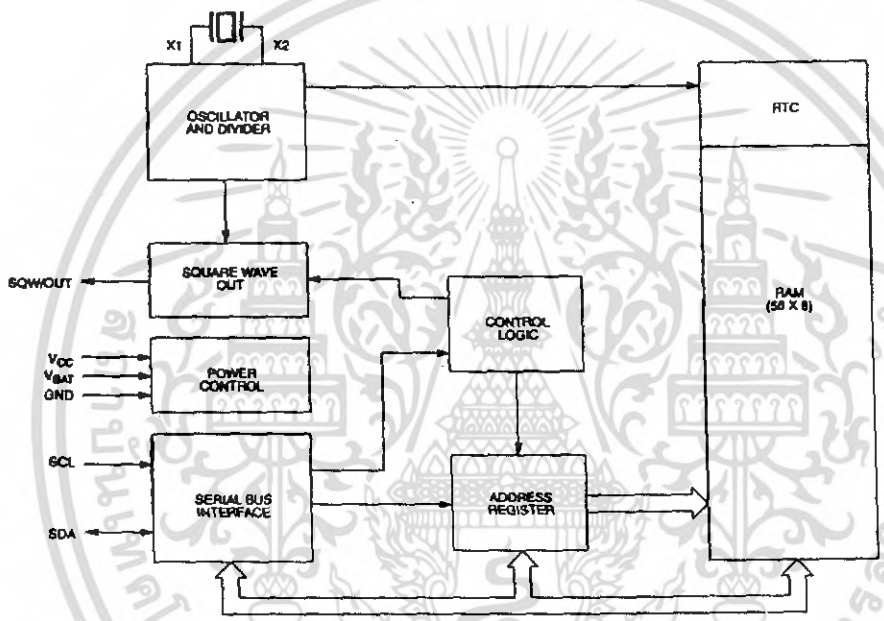
The DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2-wire bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit which detects power failures and automatically switches to the battery supply.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT} the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock.

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC}, GND - DC power is provided to the device on these pins. V_{CC} is the +5 volt input. When 5 volts is applied within normal limits, the device is fully accessible and data can be written and read. When a 3-volt battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the Timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} - Battery input for any standard 3-volt lithium cell or other energy source. Battery voltage must be held between 2.0 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48 mAhr or greater will back up the DS1307 for more than 10 years in the absence of power at 25 degrees C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCL (Serial Clock Input) - SCL is used to synchronize data movement on the serial interface.

SDA (Serial Data Input/Output) - SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

SQW/OUT (Square Wave/ Output Driver) - When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 kHz, 8 kHz, 32 kHz). The SQW/OUT pin is open drain which requires an external pullup resistor. SQW/OUT will operate with either Vcc or Vbat applied.

X1, X2 - Connections for a standard 32.768 kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

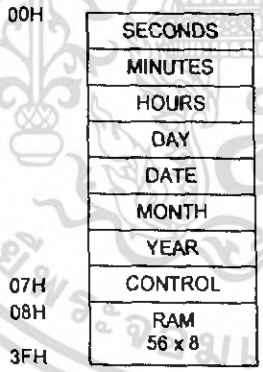
For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks." The DS1307 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

Please review Application Note 95, "Interfacing the DS1307 with a 8051-Compatible Microcontroller" for additional information.

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The real time clock registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The real time clock registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the Binary-Coded Decimal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

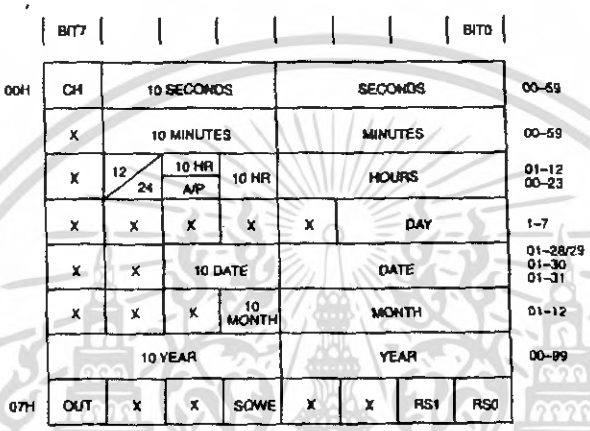
Please note that the initial power on state of all registers is not defined. Therefore it is important to enable the oscillator (CH bit=0) during initial configuration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.

DS1307 TIMEKEEPER REGISTERS Figure 3



CONTROL REGISTER

The DS1307 Control Register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	X	X	SQWE	X	X	RS1	RS0

OUT (Output control): This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE=0, the logic level on the SQW/OUT pin is 1 if OUT=1 and is 0 if OUT=0.

SQWE (Square Wave Enable): This bit, when set to a logic 1, will enable the oscillator output. The frequency of the square wave output depends upon the value of the RS0 and RS1 bits.

RS (Rate Select): These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

SQUAREWAVE OUTPUT FREQUENCY Table 1

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code MCS-51

```
#include"reg52.h"
#include<intrins.h>
#include<stdio.h>
#define LCD P1
#define DS1307_ID 0xD0
xdata unsigned char Memory[0x8000] _at_ 0x8000; // decleration
sbit RS = P3^2;
sbit E = P3^3;
sbit SW1 = P3^5;
sbit o = P2^4;
sbit oo = P2^3;
sbit SW2 = P3^4;
sbit SW3 = P3^0;
sbit SDA = P2^3;
sbit SCL = P2^4;
sbit led = P2^2;
unsigned char sec,min,hour,day,date,month,year,control;
unsigned char c,k,x1,x,s,a,w1,w2,w3,x1;
unsigned char e1,e2,e3,e4,e5;
unsigned char check1,check2,m1,m11,m2,m22,m3,m33,h1,h11,h2,h22,h3,h33;
unsigned int sh1,sh2,sh3,sm1,sm2,sm3 ;
unsigned char dd, mm, yy , hh ,mom,ss,udd, umm, uyy , uhh ,umom,uss ;
unsigned char dd1,dd2,dd3,hh1,hh2,hh3,mm1,mm2,mm3,phh1,phh2,phh3,pmm1,pmm2,pmm3;
unsigned char delete,AA,BB,CC,SC;
unsigned int  add,s1,s2,s3,z,check,i;

/*************************************Delay*********************************/
void delay()
{
 unsigned i,j;
  for (i=0;i<50;i++)
   for (j=0;j<2500;j++);
}
void delay1()
{
 unsigned i,j;
  for (i=0;i<50;i++)
   for (j=0;j<500;j++);
}

void delay_long(){
 unsigned i,j,k;
 for (i=0;i<5;i++)
   for (j=0;j<255;j++)
     for (k=0;k<255;k++);
}

// *********************************LCD*************************************
void InitLCD() {
 unsigned int i;

 for (i=0;i<5000;i++);

 LCD = 0x01; // clear screen
 RS = 0;
 E = 1; E = 0;
 for (i=0;i<200;i++);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 LCD = 0x3F; // set 8-bit mode
 RS = 0;
 E = 1; E = 0;
 for (i=0;i<200;i++);

 LCD = 0x0C; // display on
 RS = 0;
 E = 1; E = 0;
 for (i=0;i<200;i++);
}


void ClearLCD() {
 unsigned int i;
 LCD = 0x01; // clear screen
 RS = 0;
 E = 1; E = 0;
 for (i=0;i<500;i++);
}

void WriteLCD(unsigned char ch) {
 unsigned int i;
 LCD = ch;
 RS = 1;
 E = 1; E = 0;
 for (i=0;i<30;i++);
}

void PrintLCD(unsigned char stt[]) {
 unsigned char BB;
 for (BB=0;stt[BB]!=0;BB++)
  WriteLCD(stt[BB]);
}

void GotoLCD(unsigned char addr) {
 unsigned int i;
 LCD = addr | 0x80;
 RS = 0;
 E = 1; E = 0;
 for (i=0;i<30;i++);
}

/*************************************I2C*****************************************/

void i2c_delay(void)
{
 unsigned char i ;
 for(i=0;i<20;i++)
  _nop_ ();
}

void i2c_clk(void)
{
 i2c_delay();
 SCL = 1;
 i2c_delay();
 SCL = 0;
}

void i2c_start(void)
{
  if(SCL)
  SCL = 0;
  SDA = 1;
  SCL = 1;
  i2c_delay();
  SDA = 0;
  i2c_delay();
  SCL = 0;
}

void i2c_stop(void)
{
 if(SCL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  SCL = 0;
  SDA = 0;
  i2c_delay();
  SCL = 1;
  i2c_delay();
  SDA = 1;
}


bit i2c_wrdata(unsigned char dat)
{
  bit data_bit;
  unsigned char i;
  for(i=0;i<8;i++)
  {
     data_bit = dat & 0x80;
     SDA = data_bit;
     i2c_clk();
     dat = dat<<1;
  }
  SDA = 1;
  i2c_delay();
  SCL = 1;
  i2c_delay();
  data_bit = SDA;
  SCL = 0;
  i2c_delay();
  return(data_bit);
}


unsigned char i2c_rddata(void)
{
      bit rd_bit;
      unsigned char i,dat;
      dat = 0x00;
      for(i=0;i<8;i++)
      {
            i2c_delay();
            SCL = 1;
            i2c_delay();
            rd_bit = SDA;
            dat = dat<<1;
            dat = dat | rd_bit;
            SCL = 0;
      }
      SDA = 1;
      i2c_delay();
      i2c_clk();
      SCL = 1;
      return(dat);
}
/*****************************DS1307*****************************/
void send_to_lcd(unsigned char value)
{
   unsigned char buf = 0;
   buf = value & 0xF0;
   buf = (buf>>4)|(0x30);
   WriteLCD(buf);
   buf = value & 0x0F;
   buf = buf | 0x30;
   WriteLCD(buf);
}


unsigned char DS1307_rd(unsigned char addr)
{
  unsigned char ret;
  i2c_start();
  i2c_wrdata(DS1307_ID);
  i2c_wrdata(addr);
  i2c_start();
  i2c_wrdata(DS1307_ID+1);
  ret = i2c_rddata();
  i2c_stop();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  return(ret);
}

void DS1307_wrtime(unsigned char hh, mm, ss)
{
   i2c_start();
   i2c_wrdata(DS1307_ID);
   i2c_wrdata(0x00);
   i2c_wrdata(ss);
   i2c_wrdata(mm);
   i2c_wrdata(hh);
   i2c_stop();
}

void DS1307_wrdate(unsigned char dd, mom, yy)
{
   i2c_start();
   i2c_wrdata(DS1307_ID);
   i2c_wrdata(0x04);
   i2c_wrdata(dd);
   i2c_wrdata(mom);
   i2c_wrdata(yy);
   i2c_stop();
}


/*********************************Run ************************/
void Run()
{     ClearLCD();
      GotoLCD(0x00);
      PrintLCD("..RECEIVE DATA..");
      delay();
      delay();
      delay();

      GotoLCD(0x00);
      PrintLCD("..PLEASE..WAIT..");
      delay1();

      GotoLCD(0x40);
      PrintLCD("*");
      delay1();

      GotoLCD(0x40);
      PrintLCD("**");
      delay1();

      GotoLCD(0x40);
      PrintLCD("***");
      delay1();


      GotoLCD(0x40);
      PrintLCD("****");
      delay1();


      GotoLCD(0x40);
      PrintLCD("*****");
      delay1();


      GotoLCD(0x40);
      PrintLCD("******");
      delay1();


      GotoLCD(0x40);
      PrintLCD("*******");
      delay1();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GotoLCD(0x40);
    PrintLCD("********");
    delay1();


    GotoLCD(0x40);
    PrintLCD("*********");
    delay1();


    GotoLCD(0x40);
    PrintLCD("**********");
    delay1();


    GotoLCD(0x40);
    PrintLCD("***********");
    delay1();


    GotoLCD(0x40);
    PrintLCD("************");
    delay1();



    GotoLCD(0x40);
    PrintLCD("*************");
    delay1();


    GotoLCD(0x40);
    PrintLCD("**************");
    delay1();



    GotoLCD(0x40);
    PrintLCD("***************");
    delay1();


    GotoLCD(0x40);
    PrintLCD("****************");
    delay1();
}
/************************SEND CODE AND RECEIVE DATA**********************/
void Send_Receive1()
{  unsigned char dat = 0;
   unsigned char First,Second;
   TMOD = 0x21;
   SCON = 0x50;
   TH1 = 0xFD;
       TL1 = 0XFD;
   TF1 = 0;
   TI = 0;
   RI = 0;
   TR1 = 1;
////////////Send/////////////////////////////////
        delay();
        dat = 0xAA;
        SBUF = dat;
        while(~TI);
        TI = 0;

////////////Receive/////////////
         while(~RI);
         RI = 0;
         First = SBUF;
         delay();
         while(~RI);
         RI = 0;
         Second = SBUF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        add++;
        Memory[add]  First;

        add++;
        Memory[add]  Second;
}
/*********************************************************************/

void Send_Receive2()
{  unsigned char dat  0;
  unsigned char First,Second;
  TMOD  0x21;
  SCON  0x50;
  TH1  0xFD;
      TL1  0XFD;
  TF1  0;
  TI  0;
  RI  0;
  TR1  1;
//////////// Send ////////////////
        delay();
        dat  0xBB;
        SBUF  dat;
        while(!TI);
        TI  0;

/////// Receive ////////
        while(!RI);
        RI  0;
        First  SBUF;

        delay();
        while(!RI);
        RI  0;
        Second  SBUF;

        add++;
        Memory[add]  First;

        add++;
        Memory[add]  Second;

}
/*********************************************************************/

void Send_Receive3()
{  unsigned char dat  0;
  unsigned char First,Second;
  TMOD  0x21;
  SCON  0x50;
  TH1  0xFD;
      TL1  0XFD;
  TF1  0;
  TI  0;
  RI  0;
  TR1  1;
//////////// Send ////////////////
        delay();
        dat  0xCC;
        SBUF  dat;
        while(!TI);
        TI  0;

/////// Receive ////////
        while(!RI);
        RI  0;
        First  SBUF;

        delay();
        while(!RI);
        RI  0;
        Second  SBUF;

        add++;
        Memory[add]  First;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        add++;
        Memory[add] = Second;


}
/**********************************Form_Time ************************************/
void Form_Time()
{
    /********************** Sensor 1 ***************************/
     if(hh1 >= 0x24)
     {hh1 = hh1 & 0x0F;}
     if(mm1 >= 0x60)
     {mm1 = mm1 & 0x0F;}


    /********************** Sensor 2 ***************************/
     if(hh2 >= 0x24)
     {hh2 = hh2 & 0x0F;}
     if(mm2 >= 0x60)
     {mm2 = mm2 & 0x0F;}

    /********************** Sensor 3 ***************************/
     if(hh3 >= 0x24)
     {hh3 = hh3 & 0x0F;}
     if(mm3 >= 0x60)
     {mm3 = mm3 & 0x0F;}

}
/**********************************Retrive_Data_Sensor1...code AA ***********************/

void Retrive_Data_Sensor1()
{
     ClearLCD();
     GotoLCD(0x00);
     PrintLCD("Receive Data S_1");
     delay();
     delay();
     Run();
     add++;
     Memory[add] = 0xAA;/*******************code S1*********************/
     ////////////////////////////////
     Send_Receive1();

     ////////////////////////////////
     mm1 = min+pmm1;
     hh1 = hour+phh1;

}

/**********************************Retrive_Data_Sensor2...code BB ***********************/

void Retrive_Data_Sensor2()
{
     ClearLCD();
     GotoLCD(0x00);
     PrintLCD("Receive Data S_2");
     delay();
     delay();
     Run();
     add++;
     Memory[add] = 0xBB;/*******************code S2*********************/
     ////////////////////////////////
      Send_Receive2();
     ////////////////////////////////
     mm2 = min+pmm2;
     hh2 = hour+phh2;

}

/**********************************Retrive_Data_Sensor3...code CC ***********************/

void Retrive_Data_Sensor3()
{
     ClearLCD();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GotoLCD(0x00);
    PrintLCD("Receive Data S_3");
    delay();
    delay();
    Run();
    add++;
    Memory[add] = 0xCC;/*****************code S3*******************/
    ////////////////////////////////
     Send_Receive3();
    ////////////////////////////////
    mm3 = mm3 + pmm3;
    hh3 = hh3 + phh3;

}

/********************************* Time ********************************************/
void ser_datetime(void)
{
  ClearLCD();
  GotoLCD(0x00);
  PrintLCD("T:");
}

void display_datetime(void)
{  dd = 0x00;
  mm = 0x00;
  yy = 0x00;
  hh = 0x00;
  mom = 0x00;
  ss = 0x00;
  DS1307_wrtime(hh,mm,0x00);
  DS1307_wrdate(dd,mom,yy);
  sec = DS1307_rd(0x00);
  min = DS1307_rd(0x01);
  hour = DS1307_rd(0x02);
  day = DS1307_rd(0x03);
  date = DS1307_rd(0x04);
  month = DS1307_rd(0x05);
  year = DS1307_rd(0x06);
  GotoLCD(0x01);
  WriteLCD(':');
  send_to_lcd(date);
  WriteLCD('/');
  send_to_lcd(month);
  WriteLCD('/');
  send_to_lcd(year);
  GotoLCD(0x0B);
  send_to_lcd(hour);
  WriteLCD(':');
  send_to_lcd(min);
  GotoLCD(0x40);
  PrintLCD("SW1:Ent SW2:Next");
}
void Show_datetime()
{

  sec = DS1307_rd(0x00);
  min = DS1307_rd(0x01);
  hour = DS1307_rd(0x02);
  day = DS1307_rd(0x03);
  date = DS1307_rd(0x04);
  month = DS1307_rd(0x05);
  year = DS1307_rd(0x06);
  GotoLCD(0x00);
  WriteLCD('T');
  GotoLCD(0x01);
  WriteLCD(':');
  send_to_lcd(date);
  WriteLCD('/');
  send_to_lcd(month);
  WriteLCD('/');
  send_to_lcd(year);
  GotoLCD(0x0B);
  send_to_lcd(hour);
  WriteLCD(':');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   send_to_lcd(min);
   GotoLCD(0x40);
   send_to_lcd(sec);
}

void display_datetime1(void)
{
   DS1307_wrdate( dd, mom, yy);
   date = DS1307_rd(0x04);
   GotoLCD(0x00);
   PrintLCD("T");
   GotoLCD(0x01);
   WriteLCD(':');
   send_to_lcd(date);
   GotoLCD(0x40);
   PrintLCD("SW1:Ent SW2:Next");
}

void display_datetime2(void)
{
   DS1307_wrdate( dd, mom, yy);
   month = DS1307_rd(0x05);
   GotoLCD(0x00);
   PrintLCD("T");
   GotoLCD(0x01);
   WriteLCD(':');
   send_to_lcd(date);
   WriteLCD('/');
   send_to_lcd(month);
   GotoLCD(0x40);
   PrintLCD("SW1:Ent SW2:Next");
}

void display_datetime3(void)
{
   DS1307_wrdate(dd, mom, yy);
   year = DS1307_rd(0x06);
   GotoLCD(0x00);
   PrintLCD("T");
   GotoLCD(0x01);
   WriteLCD(':');
   send_to_lcd(date);
   WriteLCD('/');
   send_to_lcd(month);
      WriteLCD('/');
   send_to_lcd(year);
   GotoLCD(0x40);
   PrintLCD("SW1:Ent SW2:Next");
}

void display_datetime4(void)
{
   DS1307_wrtime( hh, mm,0x00);
   hour = DS1307_rd(0x02);
   GotoLCD(0x00);
   PrintLCD("T");
   GotoLCD(0x01);
   WriteLCD(':');
   send_to_lcd(date);
   WriteLCD('/');
   send_to_lcd(month);
      WriteLCD('/');
   send_to_lcd(year);
   GotoLCD(0x0B);
   send_to_lcd(hour);
      GotoLCD(0x40);
   PrintLCD("SW1:Ent SW2:Next");
}

void display_datetime5(void)
{
  DS1307_wrtime( hh, mm,0x00);
  min = DS1307_rd(0x01);
  GotoLCD(0x00);
  PrintLCD("T");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  GotoLCD(0x01);
  WriteLCD(':');
  send_to_lcd(date);
  WriteLCD('/');
  send_to_lcd(month);
       WriteLCD('/');
  send_to_lcd(year);
  GotoLCD(0x0B);
  send_to_lcd(hour);
       WriteLCD(':');
  send_to_lcd(min);
       GotoLCD(0x40);
  PrintLCD("SW1:Ent SW2:Next");
}
void display_datetime6(void)
{
   DS1307_wrdate( dd, mom, yy);
   date = DS1307_rd(0x04);
   GotoLCD(0x00);
   PrintLCD("TIME");
   GotoLCD(0x06);
   WriteLCD(':');
   GotoLCD(0x08);
   send_to_lcd(date);
   GotoLCD(0x40);
   PrintLCD("SW1:Ent SW2:Next");
}

void display_datetime7(void)
{
   DS1307_wrtime( hh, mm,0x00);
   hour = DS1307_rd(0x02);
   GotoLCD(0x00);
   PrintLCD("SAMPLING");
   GotoLCD(0x09);
   WriteLCD(':');
   GotoLCD(0x0B);
   send_to_lcd(hour);
        GotoLCD(0x40);
   PrintLCD("SW1:Ent SW2:Next");
}
void display_datetime8(void)
{
  DS1307_wrtime( hh, mm,0x00);
  min = DS1307_rd(0x01);
  GotoLCD(0x00);
  PrintLCD("SAMPLING");
  GotoLCD(0x09);
  WriteLCD(':');
  GotoLCD(0x0B);
  send_to_lcd(hour);
  WriteLCD('/');
  send_to_lcd(min);
       GotoLCD(0x40);
  PrintLCD("SW1:Ent SW2:Next");
}
/*****************************Sampling*************************************/
void Sampling()
      {
      GotoLCD(0x00);
      PrintLCD("SAMPLING : HH/MM");
 GotoLCD(0x40);
      PrintLCD("SW1:Ent SW2:Next");

      }
/********************************** Set Data Sampling ***********************************/
void Set_Data_Sampling()
{    Sampling();
     delay();
     delay();
         ClearLCD();
   dd = 0x00;
   mm = 0x00;
   yy = 0x00;
   hh = 0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  mom = 0x00;
  ss = 0x00;
  c4 = 1;
     x1 = 0;

 /*************************************/
while(SW1 == 1)
      {   display_datetime7();
     if(SW2==0)
       {      delay();

                                        x1 = 1;
            if(c4 == 1)
            hh = 0x01;
            if(c4 == 2)
            hh = 0x03;
            if(c4 == 3)
            hh = 0x06;
            if(c4 == 4)
            {hh = 0x12;
            c4 = 0;}
                                        c4 = c4 + 1;

            if(x == 1)
            {hh1 = uhh+hh;
             phh1 = hh;
                                              sh1 = 1;

             Memory[add] = hh;
            }

            if(x == 2)
            {hh2 = uhh+hh;
             phh2 = hh;
             sh2 = 1;

             Memory[add] = hh;
             }

            if(x == 3)
            {hh3   uhh+hh;
             phh3 = hh;
                                              sh3 = 1;

             Memory[add] = hh;
            }


       }


 }
  dd = 0x00;
  mm = 0x00;
  yy = 0x00;
  hh = 0x00;
  mom = 0x00;
  ss = 0x00;
  c5 = 1;
     delay();
     ClearLCD();
if(x1 == 0)
{
 while(SW1 == 1)
      {   display_datetime8();

     if(SW2==0)
       {      delay();
            if(c5 == 1)
            mm = 0x15;
            if(c5 == 2)
            {mm = 0x30;
            c5 = 0;}
                                        c5 = c5 + 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(x == 1)
            {mm1 = umm+mm;
             pmm1 = mm;
             sm1 = 1;
             Memory[add] = mm;}

            if(x == 2)
            {mm2 = umm+mm;
             pmm2 = mm;
             sm2 = 1;
             Memory[add] = mm;}

            if(x == 3)
            {mm3 = umm+mm;
             pmm3 = mm;
             sm3 = 1;
             Memory[add] = mm;}
            }

 }

 }

 delay();
     ClearLCD();

}
/********************************* Set Data ***********************************/
void Set_Data(){
   set_datetime();
   display_datetime();
       delay();
   delay();
       ClearLCD();
   dd = 0x00;
   mm = 0x00;
   yy = 0x00;
   hh = 0x00;
   mom = 0x00;
   ss = 0x00;
   i = 1;

 while(SW1 == 1)
 {  display_datetime1();

   if(SW2 == 0)
      {delay();
      i = 0;
      dd++;
            if(dd == 0x0A)
            dd = 0x10;
            if(dd == 0x1A)
            dd = 0x20;
            if(dd == 0x2A)
            dd = 0x30;
            if(dd == 0x32)
            dd = 0x01;
      delay1();
      udd = dd;
      delay1();
      Memory[add] = dd;

       }
}

if(i == 1)
{   delay();
Memory[add] = 0x00;}

add = add + 1;
i = 1;
     delay();
     ClearLCD();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(SW1 == 1)
    {  display_datetime2();
  if(SW2==0)
     { delay();
       i = 0;
       mom++;
                    if(mom == 0x0A)
                    mom = 0x10;
                    if(mom == 0x13)
                    mom = 0x01;
       delay1();
       umom = mom;
       delay1();
       Memory[add] = mom;

         }
}

if(i == 1)
{    delay();
Memory[add] = 0x00;}

add = add + 1;
i = 1;
     delay();
     ClearLCD();



while(SW1 == 1)
    {  display_datetime3();
   if(SW2==0)
     { delay();
       i = 0;
          yy++;
                                  if(yy == 0x0A)
                                  yy = 0x10;
                                  if(yy == 0x1A)
                                  yy = 0x20;
                                  if(yy == 0x2A)
                                  yy = 0x30;
                                  if(yy == 0x3A)
                                  yy = 0x40;
                                  if(yy == 0x4A)
                                  yy = 0x50;
                                  if(yy == 0x5A)
                                  yy = 0x60;
                                  if(yy == 0x6A)
                                  yy = 0x70;
                                  if(yy == 0x7A)
                                  yy = 0x80;
                                  if(yy == 0x8A)
                                  yy = 0x90;
                                  if(yy == 0x9A)
                                  yy = 0x00;
          delay1();
          uyy = yy;
          delay1();
          Memory[add] = yy;

            }
}

if(i == 1)
{    delay();
Memory[add] = 0x00;}

add = add + 1;
i = 1;
     delay();
     ClearLCD();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while(SW1 == 1)
    {   display_datetime4();
    if(SW2==0)
      {     delay();
            i = 0;
            hh++;
                                        if(hh == 0x0A)
                                        hh = 0x10;
                                        if(hh == 0x1A)
                                        hh = 0x20;
                                        if(hh == 0x24)
                                        hh = 0x00;
            delay1();
            uhh = hh;
            delay1();
            Memory[add] = hh;

      }
}

if(i == 1)
{   delay();
Memory[add] = 0x00;}

add = add + 1;
i = 1;
    delay();
    ClearLCD();



while(SW1 == 1)
    {   display_datetime5();

    if(SW2==0)
      {     delay();
            i = 0;
            mm++;
                                        if(mm == 0x0A)
                                        mm = 0x10;
                                        if(mm == 0x1A)
                                        mm = 0x20;
                                        if(mm == 0x2A)
                                        mm = 0x30;
                                        if(mm == 0x3A)
                                        mm = 0x40;
                                        if(mm == 0x4A)
                                        mm = 0x50;
                                        if(mm == 0x5A)
                                        mm = 0x00;
            delay1();
            umm = mm;
            delay1();
            Memory[add] = mm;
            ss = 0x00;

            }
}

if(i == 1)
{   delay();
Memory[add] = 0x00;}

delay();
    ClearLCD();
}
/*********************************Check_data ************************************/
void Check_data()
{     h1 = check1 & 0xF0;
      h11 = check1 & 0x0F;
      m1 = check2 & 0xF0;
      m11 = check2 & 0x0F;
      if(h1 == 0xA0)
          h1 = 0x00;
      if(h1 == 0xB0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    h1 = 0x00;
if(h1 == 0xC0)
    h1 = 0x00;
if(h1 == 0xD0)
    h1 = 0x00;
if(h1 == 0xE0)
    h1 = 0x00;
if(h1 == 0xF0)
    h1 = 0x00;

if(h11 == 0x0A)
    h11 = 0x10;
if(h11 == 0x0B)
    h11 = 0x11;
if(h11 == 0x0C)
    h11 = 0x12;
if(h11 == 0x0D)
    h11 = 0x13;
if(h11 == 0x0E)
    h11 = 0x14;
if(h11 == 0x0F)
    h11 = 0x15;

if(m1 == 0xA0)
    m1 = 0x00;
if(m1 == 0xB0)
    m1 = 0x00;
if(m1 == 0xC0)
    m1 = 0x00;
if(m1 == 0xD0)
    m1 = 0x00;
if(m1 == 0xE0)
    m1 = 0x00;
if(m1 == 0xF0)
    m1 = 0x00;

if(m11 == 0x0A)
    m11 = 0x10;
if(m11 == 0x0B)
    m11 = 0x11;
if(m11 == 0x0C)
    m11 = 0x12;
if(m11 == 0x0D)
    m11 = 0x13;
if(m11 == 0x0E)
    m11 = 0x14;
if(m11 == 0x0F)
    m11 = 0x15;

check1 = h1 + h11;
check2 = m1 + m11;
}
/*****Finish*****/
void Finish()
{
  ClearLCD();
  GotoLCD(0x00);
  PrintLCD(" STOP OPERATION ");
  add = add + 1;
        Memory[add] = 0xFF;//code FF is finished to retrive DATA
  delay();
  delay();
  delay();
  delay();
  ClearLCD();
  GotoLCD(0x00);
  PrintLCD(" STOP COMPLETE!!");
  GotoLCD(0x40);
  PrintLCD(" PRESS..RESTART ");
  while(1);
}
/*****Sent_data()*****/
void Sent_data()
{
      unsigned char p;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  ClearLCD();
  GotoLCD(0x00);
  PrintLCD("..SENDING DATA..");
  delay();
  delay();
  delay();
  ClearLCD();
  GotoLCD(0x00);
  PrintLCD("..SENDING DATA..");
  GotoLCD(0x40);
  PrintLCD("..PLEASE..WAIT..");


  TMOD = 0x21;
  SCON = 0x50;
  TH1 = 0xFD;
  TL1 = 0xFD;
  //TF1 = 0
  RI = 0;
  TI = 1;
  TR1= 1;

i = 0x8000;
p = 0;
while(p != 0xFF)

  { delay();
    delay();
    delay1();
    p = Memory[i];
    i = i+1;
    switch(p)
    {unsigned char st[2];
    case 0x00 : printf("00");
    break;
    case 0x01 : printf("01");
    break;
    case 0x02 : printf("02");
    break;
    case 0x03 : printf("03");
    break;
    case 0x04 : printf("04");
    break;
    case 0x05 : printf("05");
    break;
    case 0x06 : printf("06");
    break;
    case 0x07 : printf("07");
    break;
    case 0x08 : printf("08");
    break;
    case 0x09 : printf("09");
    break;
    case 0x0A : printf("0A");
    break;
    case 0x0B : printf("0B");
    break;
    case 0x0C : printf("0C");
    break;
    case 0x0D : printf("0D");
    break;
    case 0x0E : printf("0E");
    break;
    case 0x0F : printf("0F");
    break;
    case 0xAA : printf("AA");
    break;
    case 0xAB : printf("AA");
    break;
    case 0xBB : printf("BB");
    break;
    case 0xBC : printf("BB");
    break;
    case 0xCC : printf("CC");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    break;
    case 0xCD : printf("CC");
    break;
    case 0xE6 : printf("00");
    break;
    default   : sprintf(stt,"%X",p);
            printf(stt);
    }
  }
 ////////////////////////////////////////////////////////////////////
   Memory[0x8000] = 0xFF;
   ClearLCD();
   GotoLCD(0x00);
   PrintLCD("SENDING COMPLETE");
   GotoLCD(0x40);
   PrintLCD(" PRESS..RESTART ");

while(1);
}

/*************************Operate*************************************/
void Operate()
{
      add = add + 1;
 delay();
      Memory[add] = 0xDD;////////code DD is start to retrive DATA/////////////
 delay();
      while(SW2 == 1)
 {
     delay();
     ClearLCD();
     Show_datetime();
     GotoLCD(0x40);
        PrintLCD("Stand by Channal");
     delay_long();
     ClearLCD();
        GotoLCD(0x00);
         PrintLCD(" Press hold SW2 ");
     GotoLCD(0x40);
         PrintLCD("to stop operate.");
     delay_long();

      if(s1 == 1)
  {
            check1 = hh1;
            check2 = mm1;
            Check_data();
            hh1 = check1;
            mm1 = check2;
            Form_Time();
  }

 if(s1 == 1)
  {
            if(sh1 == 1)
                    {if(hour == hh1)
                    {Retrive_Data_Sensor1();}
   }

            if(sm1 == 1)
                    {if(min == mm1)
                    {Retrive_Data_Sensor1();}
        }
  }


     if(s2 == 1)
  {
            check1 = hh2;
            check2 = mm2;
            Check_data();
            hh2 = check1;
            mm2 = check2;
            Form_Time();
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(s2 == 1)
  {
          if(sh2 == 1)
                  {if(hour == hh2)
                  {Retrive_Data_Sensor2();}
   }

          if(sm2 == 1)
                  {if(min == mm2)
                  {Retrive_Data_Sensor2();}
   }
  }


    if(s3 == 1)
  {
  check1 = hh3;
          check2 = mm3;
          Check_data();
          hh3 = check1;
          mm3 = check2;
          Form_Time();
  }

  if(s3 == 1)
  {
          if(sh3 == 1)
          {if(hour == hh3)
          {Retrive_Data_Sensor3();}
   }

          if(sm3 == 1)
          {if(min == mm3)
          {Retrive_Data_Sensor3();}
   }
  }

}

}
/*************************Select_Ch 1 2 3 ********************************/
void Select_Ch1()
{     GotoLCD(0x03);
 PrintLCD("Channel_1");
      GotoLCD(0x40);
 PrintLCD("SW1:Set SW2:Next");
}

void Select_Ch2()
{     GotoLCD(0x03);
 PrintLCD("Channel_2");
      GotoLCD(0x40);
 PrintLCD("SW1:Set SW2:Next");
}

void Select_Ch3()
{     GotoLCD(0x03);
 PrintLCD("Channel_3");
      GotoLCD(0x40);
 PrintLCD("SW1:Set SW2:Next");
}
/*****************************Select_Ch*********************************/
void Select_Ch_3()
{     dd = 0x00;
  mm = 0x00;
  yy = 0x00;
  hh = 0x00;
  mom = 0x00;
  ss = 0x00;
      delay();
      ClearLCD();
      while(SW2 == 1)
          {           z = 3;
                      Select_Ch3();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if(SW1 == 0)
                                {add = add + 1;
        delay();
        Memory[add] = 0xCC;
        delay();
                                add = add + 1;

    z = 3;
    k = 0;
    x = 3;
    while(k == 0)
                                {s3 = 1;
        Set_Data_Sampling();
        k = 1;
                                }
     z = 3;
                    delay();
                    ClearLCD();
                    DS1307_wrtime(uhh,umm,0x00);
                    DS1307_wrdate(udd,umom,uyy);
                    Operate();
      }

  }
        delay();
        ClearLCD();
            DS1307_wrtime(uhh,umm,0x00);
    DS1307_wrdate(udd,umom,uyy);
        Operate();

}
/*****************************************************/
void Select_Ch_2()
{     dd = 0x00;
  mm = 0x00;
  yy = 0x00;
  hh = 0x00;
  mom = 0x00;
  ss = 0x00;
      delay();
      ClearLCD();
      while(SW2 == 1)
          {           z = 2;
                      Select_Ch2();
                      if(SW1 ==0)
                      {add = add + 1;
        delay();
        Memory[add] = 0xBB;
        delay();
                                add = add + 1;


    z = 2;
    k = 0;
    x = 2;
    while(k == 0)
                                    { s2 = 1;
                    Set_Data_Sampling();
                    k = 1;}
    z = 2;
     Select_Ch_3();
                    }
  }

 Select_Ch_3();
}
/*****************************************************/
void Select_Ch()
{  dd = 0x00;
  mm = 0x00;
  yy = 0x00;
  hh = 0x00;
  mom = 0x00;
  ss = 0x00;
  x = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while(SW2 == 1)
        {                       z = 1;
                                Select_Ch1();
                if(SW1 == 0)
                                {add = add + 1;
        delay();
        Memory[add] = 0xAA;
        delay();
                                add = add + 1;


    z = 0;
    k = 0;
    x = 1;
          while(k == 0)
        { s1 = 1;
                                Set_Data_Sampling();
        k = 1;
                                }
    z = 1;
    Select_Ch_2();
          }
          }
          Select_Ch_2();
  }



/*********************************Main*************************************************/
void main()
{

  a = 0;
  x1 = 1;
  z = 0;
  w1 = 0;
  w2 = 0;
  w3 = 0;
  s1 = 0;
  s2 = 0;
  s3 = 0;
     sh1 = 0;
     sh2 = 0;
     sh3 = 0;
     sm1 = 0;
     sm2 = 0;
     sm3 = 0;
 add = 0;
  InitLCD();
  PrintLCD("   DATA LOGGER");
  delay();
  delay();
  delay();
  i = 0x7FFF;
  i = i+1;

//////////////////////////
  if(Memory[0x8000] == 0xEE)
  {
   i = 0;
   ClearLCD();
   GotoLCD(0x00);
   PrintLCD(" SW1 send data");
   GotoLCD(0x40);
   PrintLCD(" SW2 set data");
  }
//////////////////////////////////
  if(Memory[0x8000] != 0xEE)
  {
   i = 1;
   ClearLCD();
   GotoLCD(0x00);
   PrintLCD(" START OPERATION");
   GotoLCD(0x40);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    PrintLCD(" SW2 set data");
   }

/*******************Set time******************/
 while(1)
 {
    if(SW2==0)
        {    add = 0x8000;
     Memory[0x8000] = 0xEE;
     add = add + 1;
     delay();
     ClearLCD();
     Set_Data();
 /*****************Set channel*****************/

              Select_Ch();
/*****************Operate**************************/
     delay();
     DS1307_wrtime(uhh,umm,0x00);
     DS1307_wrdate(udd,umoth,uyy);

     delay();


     Operate();
     Finish();

      }

    if(i == 0)
    {
      if(SW1==0)
       {
        delay();
        Sent_data();
       }
    }
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Program Data Logger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form 1

```
Dim Day As String
Dim Month As String
Dim Year As String
Dim Hour As String
Dim Minute As String
Dim check As String

Dim i As Integer
Dim n As Integer

Dim RC(1 To 57536) As String
Dim Sams1 As String
Dim Sams2 As String
Dim Sams3 As String

Dim Count0 As Integer
Dim Count1 As Integer
Dim Count2 As Integer
Dim Count3 As Integer

Dim Data_H As String
Dim Data_L As String

Dim Day1 As Integer
Dim Month1 As Integer
Dim Year1 As Integer
Dim Hour1 As Integer
Dim Minute1 As Integer

Dim Day2 As Integer
Dim Month2 As Integer
Dim Year2 As Integer
Dim Hour2 As Integer
Dim Minute2 As Integer

Dim Day3 As Integer
Dim Month3 As Integer
Dim Year3 As Integer
Dim Hour3 As Integer
Dim Minute3 As Integer

Dim Sams_I1 As Integer
Dim Sams_I2 As Integer
Dim Sams_I3 As Integer

''''''''''''''''''''''''''''''''''''



Private Sub Command1_Click()
Data1.Recordset.MoveLast
Sam1 = "0"
Date1 = "0"
Time1 = "0"
AA = "0"
Data1.Recordset.OpenRecordset
Data1.Recordset.MovePrevious
While Sam1 <> "0"
Data1.Recordset.Delete
Data1.Recordset.MovePrevious

While i <> 1000
    While j <> 500
    j = j + 1
    Wend
    i = i + 1
    j = 0
Wend
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
i = 0
Wend
Data1.Recordset.MoveLast
End Sub

Private Sub Command2_Click()
Data3.Recordset.MoveLast
Sam3 = "0"
Date3 = "0"
Time3 = "0"
CC = "0"
Data3.Recordset.OpenRecordset
Data3.Recordset.MovePrevious
While Sam3 <> "0"
Data3.Recordset.Delete
Data3.Recordset.MovePrevious

While i <> 1000
      While i <> 500
      j = j + 1
      Wend
      i = i + 1
      j = 0
Wend
i = 0
Wend
Data3.Recordset.MoveLast
End Sub

Private Sub Command3_Click()
i = 0
'On Error Resume Next
'msc.DTREnable = False
'msc.DTREnable = True
'msc.InputLen = 0
msc.InputLen = 2
'''''''''''''''''''''''''START V2''''''''''''''''''''''''''''''''''''''''''''''''''''

While i = 0
Receive.Enabled = True
Data = msc.Input
Receive.Enabled = False

   If "" = Data Then
      MsgBox "Please send data from Data Logger", vbOKOnly + vbExclamation, "Don't receive Data"
      Exit Sub
   End If

   If "EE" = Data Then
      i = 1
   Else
      MsgBox "Please send data from Data Logger again", vbOKOnly + vbExclamation, "Data Error"
      Exit Sub
   End If

Wend
RC(i) = Data
While "FF" <> Data '''''''Check FF
If i = 57537 Then
      MsgBox "Please send data from Data Logger again", vbOKOnly + vbExclamation, "Data Error"
      Exit Sub
End If

Receive.Enabled = True
Data = msc.Input
Receive.Enabled = False
i = i + 1
RC(i) = Data
Wend
''''''''''''''''''''''''''''
i = 2
Day = RC(i)
i = i + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Month = RC(i)
i = i + 1
Year = RC(i)
i = i + 1
Hour = RC(i)
i = i + 1
Minute = RC(i)
i = i + 1
Date.Text = Day & "/" & Month & "/" & Year & "  " & Hour & "." & Minute

Day1 = Day
Month1 = Month
Year1 = Year
Hour1 = Hour
Minute1 = Minute

Day2 = Day
Month2 = Month
Year2 = Year
Hour2 = Hour
Minute2 = Minute

Day3 = Day
Month3 = Month
Year3 = Year
Hour3 = Hour
Minute3 = Minute
Sams1 = "0"
Sams2 = "0"
Sams3 = "0"

If RC(i) = "AA" Then
Sensor1.Text = "Use"
i = i + 1
Sams1 = RC(i)
Sams_I1 = Sams1
  If CInt(Sams1) > 14 Then
    Sampling1.Text = Sams1 & " minutes"
  Else
    Sampling1.Text = Sams1 & " Hours"
  End If
i = i + 1
End If

If RC(i) = "BB" Then
Sensor2.Text = "Use"
i = i + 1
Sams2 = RC(i)
Sams_I2 = Sams2
  If CInt(Sams2) > 14 Then
    Sampling2.Text = Sams2 & " minutes"
  Else
    Sampling2.Text = Sams2 & " Hours"
  End If
i = i + 1
End If

If RC(i) = "CC" Then
Sensor3.Text = "Use"
i = i + 1
Sams3 = RC(i)
Sams_I3 = Sams3
  If CInt(Sams3) > 14 Then
    Sampling3.Text = Sams3 & " minutes"
  Else
    Sampling3.Text = Sams3 & " Hours"
  End If
i = i + 1
End If

If Sams1 = "0" Then
Sensor1.Text = "None"
Sampling1.Text = "None"
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Sams2 = "0" Then
Sensor2.Text = "None"
Sampling2.Text = "None"
End If

If Sams3 = "0" Then
Sensor3.Text = "None"
Sampling3.Text = "None"
End If
Count0 = 0
Count1 = 0
Count2 = 0
Count3 = 0

If RC(i) = "DD" Then
i = i + 1
While RC(i) <> "FF"

   Select Case RC(i)
Case "AA"
                '''''''''''''''
                i = i + 1
                Data_H = RC(i)
                i = i + 1
                Data_L = RC(i)
                i = i + 1
                AA = Data_H & "." & Data_L
                '''''''''''''''
                Count1 = Count1 + 1
                Sam1 = CStr(Count1)
                '''''''''''''''
                If Sams_11 > 14 Then
                  Minute1 = Minute1 + Sams_11
                  If Minute1 > 59 Then
                  Minute1 = Minute1 - 60
                  Hour1 = Hour1 + 1
                    If Hour1 > 23 Then
                    Hour1 = Hour1 - 24
                    Day1 = Day1 + 1
                      Select Case Month1
                            Case 1, 3, 5, 7, 8, 10
                              If Day1 > 31 Then
                                    Day1 = Day1 - 31
                                    Month1 = Month1 + 1
                              End If
                            Case 2
                              If Day1 > 28 Then
                                    Day1 = Day1 - 28
                                    Month1 = Month1 + 1
                              End If
                            Case 4, 6, 9, 11
                              If Day1 > 30 Then
                                    Day1 = Day1 - 30
                                    Month1 = Month1 + 1
                              End If
                            Case 12
                              If Day1 > 31 Then
                                    Day1 = Day1 - 31
                                    Month1 = 1
                                    Year1 = Year1 + 1
                              End If
                      End Select
                    End If
                  End If
                Else
                  Hour1 = Hour1 + Sams_11
                    If Hour1 > 23 Then
                    Hour1 = Hour1 - 24
                    Day1 = Day1 + 1
                      Select Case Month1
                            Case 1, 3, 5, 7, 8, 10
                              If Day1 > 31 Then
                                    Day1 = Day1 - 31
                                    Month1 = Month1 + 1
                              End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Case 2
                      If Day1 > 28 Then
                           Day1 = Day1 - 28
                           Month1 = Month1 + 1
                      End If
                    Case 4, 6, 9, 11
                      If Day1 > 30 Then
                           Day1 = Day1 - 30
                           Month1 = Month1 + 1
                      End If
                    Case 12
                      If Day1 > 31 Then
                           Day1 = Day1 - 31
                           Month1 = 1
                           Year1 = Year1 + 1
                      End If
                  End Select
                End If
              End If
              '''''''''''''''''''

            Minute = Minute1
            Year = Year1

            If Len(Minute) = 1 Then
            Minute = "0" + Minute
            End If

            If Len(Year) = 1 Then
            Year = "0" + Year
            End If

            Date1 = CStr(Day1) & "/" & CStr(Month1) & "/" & Year
            Time1 = CStr(Hour1) & "." & Minute
            Data1.Recordset.AddNew
          '''''''''''''
Case "BB"
            '''''''''''''
            i = i + 1
            Data_H = RC(i)
            i = i + 1
            Data_L = RC(i)
            i = i - 1
            BB = Data_H & "." & Data_L
            '''''''''''''''

            Count2 = Count2 + 1
            Sam2 = CStr(Count2)
            '''''''''''''''''

            If Sams_12 > 14 Then
              Minute2 = Minute2 + Sams_12
              If Minute2 > 59 Then
                Minute2 = Minute2 - 60
                Hour2 = Hour2 + 1

              If Hour2 > 23 Then
                Hour2 = Hour2 - 24
                Day2 = Day2 + 1

              Select Case Month2
                    Case 1, 3, 5, 7, 8, 10
                      If Day2 > 31 Then
                           Day2 = Day2 - 31
                           Month2 = Month2 + 1
                      End If
                    Case 2
                      If Day2 > 28 Then
                           Day2 = Day2 - 28
                           Month2 = Month2 + 1
                      End If
                    Case 4, 6, 9, 11
                      If Day2 > 30 Then
                           Day2 = Day2 - 30
                           Month2 = Month2 + 1
                      End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Case 12
                      If Day2 > 31 Then
                          Day2 = Day2 - 31
                          Month2 = 1
                          Year2 = Year2 + 1
                      End If
                End Select
                End If
                End If
              Else
                Hour2 = Hour2 + Sams_I2
                If Hour2 > 23 Then
                  Hour2 = Hour2 - 24
                  Day2 = Day2 + 1
                Select Case Month2
                    Case 1, 3, 5, 7, 8, 10
                      If Day2 > 31 Then
                          Day2 = Day2 - 31
                          Month2 = Month2 + 1
                      End If
                    Case 2
                      If Day2 > 28 Then
                          Day2 = Day2 - 28
                          Month2 = Month2 + 1
                      End If
                    Case 4, 6, 9, 11
                      If Day2 > 30 Then
                          Day2 = Day2 - 30
                          Month2 = Month2 + 1
                      End If
                    Case 12
                      If Day2 > 31 Then
                          Day2 = Day2 - 31
                          Month2 = 1
                          Year2 = Year2 + 1
                      End If
                End Select
                End If
              End If

              '''''''''''''''''''''''''

            Minute = Minute2
            Year = Year2

            If Len(Minute) = 1 Then
            Minute = "0" + Minute
            End If

            If Len(Year) = 1 Then
            Year = "0" + Year
            End If

            Date2 = CStr(Day2) & "/" & CStr(Month2) & "/" & Year
            Time2 = CStr(Hour2) & "." & Minute
            Data2.Recordset.AddNew

              '''''''''''''''''''''''''

Case "CC"
              ''''''''''''''''
              i = i + 1
              Data_H = RC(i)
              i = i + 1
              Data_L = RC(i)
              i = i + 1
              CC = Data_H & "." & Data_L
              ''''''''''''''''
              Count3 = Count1 + 3
              Sam3 = CStr(Count3)
              ''''''''''''''''
              If Sams_I3 > 14 Then
                Minute3 = Minute3 + Sams_I3
                If Minute3 > 59 Then
                Minute3 = Minute3 - 60
                Hour3 = Hour3 + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                If Hour3 > 23 Then
                Hour3 = Hour3 - 24
                Day3 = Day3 + 1
                  Select Case Month3
                      Case 1, 3, 5, 7, 8, 10
                        If Day3 > 31 Then
                            Day3 = Day1 - 31
                            Month3 = Month3 + 1
                        End If
                      Case 2
                        If Day3 > 28 Then
                            Day3 = Day3 - 28
                            Month3 = Month3 + 1
                        End If
                      Case 4, 6, 9, 11
                        If Day3 > 30 Then
                            Day3 = Day3 - 30
                            Month3 = Month3 + 1
                        End If
                      Case 12
                        If Day3 > 31 Then
                            Day3 = Day3 - 31
                            Month3 = 1
                            Year3 = Year3 + 1
                        End If
                  End Select
                End If
              End If
            Else
              Hour3 = Hour3 + Sams 13
                If Hour3 > 23 Then
                Hour3 = Hour3 - 24
                Day3 = Day3 + 1
                  Select Case Month3
                      Case 1, 3, 5, 7, 8, 10
                        If Day3 > 31 Then
                            Day3 = Day3 - 31
                            Month3 = Month3 + 1
                        End If
                      Case 2
                        If Day3 > 28 Then
                            Day3 = Day3 - 28
                            Month3 = Month3 + 1
                        End If
                      Case 4, 6, 9, 11
                        If Day3 > 30 Then
                            Day3 = Day3 - 30
                            Month3 = Month3 + 1
                        End If
                      Case 12
                        If Day3 > 31 Then
                            Day3 = Day3 - 31
                            Month3 = 3
                            Year3 = Year3 + 1
                        End If
                  End Select
                End If
              End If
              '''''''''''''''''''''''''
            Minute = Minute3
            Year = Year3

            If Len(Minute) = 1 Then
            Minute = "0" + Minute
            End If

            If Len(Year) = 1 Then
            Year = "0" + Year
            End If

            Date3 = CStr(Day3) & "/" & CStr(Month3) & "/" & Year
            Time3 = CStr(Hour3) & ":" & Minute
            Data3.Recordset.AddNew
              '''''''''''''''''''''''''
Case "00"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Count0 = Count0 - 1
            If Count0 - 10 Then
            MsgBox "Data Error!! , Maybe sending data error or data in Data Logger error ", vbOKOnly + vbExclamation,
"Data Error!!"
            Exit Sub
            End If
            '''''''''''''''''''''''''''
Case "DD"
            i = i + 1
            '''''''''''''''''''''''''''
Case Else
            MsgBox "Data Error!! , Please sand data again", vbCritical + vbOKOnly, "Data Error!!"
            Exit Sub

End Select
Wend

Else
MsgBox "Please send data from Data Logger again", vbOKOnly + vbExclamation, "Data Error"
End If

''''''''''''''''''''END V2''''''''''''''''''''''''''''''''''''''''''''


End Sub





Private Sub Command4_Click()
Form8.Show
End Sub

Private Sub Command5_Click()
Data2.Recordset.MoveLast
Sam2 = "0"
Date2 = "0"
Time2 = "0"
BB = "0"
Data2.Recordset.OpenRecordset
Data2.Recordset.MovePrevious
While Sam2 <> "0"
Data2.Recordset.Delete
Data2.Recordset.MovePrevious

While i <> 1000
     While j <> 500
     j = j + 1
     Wend
     i = i + 1
     j = 0
Wend
i = 0
Wend
Data2.Recordset.MoveLast

End Sub


Private Sub Command6_Click()
''''''''''''''''''''''''''''''''''''''''''''''''''''''
msc.CommPort = 1
msc.Settings = "9600,n,8,1"
msc.PortOpen = True
Timer2.Interval = 100
Timer2.Enabled = True
Command6.Enabled = False

''''''''''''''''''''''''''''''''''''''''''''''''''''''

End Sub

Private Sub Form_Activate()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timer1.Enabled = True
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Form1.BackColor = QBColor(3)

Label1.BackColor = QBColor(3)
Label2.BackColor = QBColor(3)
Label5.BackColor = QBColor(3)
Label6.BackColor = QBColor(3)
Label7.BackColor = QBColor(3)
Label8.BackColor = QBColor(3)
Label9.BackColor = QBColor(3)
Label10.BackColor = QBColor(3)


Table_Sensor1.BackColor = QBColor(11)
Table_Sensor2.BackColor = QBColor(11)
Table_Sensor3.BackColor = QBColor(11)

Data1.BackColor = QBColor(11)
Data2.BackColor = QBColor(11)
Data3.BackColor = QBColor(11)

Sensor1.BackColor = QBColor(11)
Sensor2.BackColor = QBColor(11)
Sensor3.BackColor = QBColor(11)
Date.BackColor = QBColor(11)
Sampling1.BackColor = QBColor(11)
Sampling2.BackColor = QBColor(11)
Sampling3.BackColor = QBColor(11)

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Data1.Recordset.MoveLast
Data2.Recordset.MoveLast
Data3.Recordset.MoveLast

End Sub




Private Sub Receive_Timer()
On Error Resume Next
msc.DTREnable = False
msc.DTREnable = True
End Sub


Private Sub Timer1_Timer()


Label5.Left = Label5.Left - 15

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If Label5.Left = -Label5.Width Then
Label5.Left = Width
End If


End Sub


Private Sub Plot_S1_Click()
Form1.Hide
Form2.Show
End Sub

Private Sub Plot_S2_Click()
Form1.Hide
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form3.Show
End Sub

Private Sub Plot_S3_Click()
Form1.Hide
Form4.Show
End Sub
```

Form2

```
Dim i As Integer
Dim b As Integer
Dim n As Integer
Dim check As Integer
Dim Start As Integer
Dim Finish As Integer
Dim a As String
Dim D(1000000) As String
Dim s(1000000) As String

Private Sub Command1_Click()
Form2.Hide
Form1.Show
End Sub

Private Sub Command2_Click()

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If check = 1 Then
For i = Start To Finish
If i = Start Then
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
Else
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
End If
Next i
End If

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Data1.Recordset.MoveFirst

For i = 0 To n
D(i) = Data1.Recordset.Data.Value
s(i) = Data1.Recordset.Sampling.Value
Data1.Recordset.MoveNext
Next i
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If n = 0 Then
MsgBox "Error!!! No Data , Cann't to plot graph", vbCritical, "No Data!!!"
Exit Sub
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Start = 0
Finish = 0

While Start = 0
check = 1
Start = InputBox("Please Input value of sampling that want to start", "Value of sampling that want to start", "1")
   If Start = 0 Then
   MsgBox "Error value = 0 , Please Insert value's start of sampling again ", vbCritical, "Error value"
   End If

   While Start > n
      If Start > n Then
      MsgBox "Error!!! value > number of sampling , Please Insert value's start of sampling again ", vbCritical
      End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Start = InputBox("Please Input value of sampling that want to start", "Value of sampling that want to start", "1")
        Wend
    Wend
    a = Str(n)

    While Finish < Start
    Finish = InputBox("Please Input value of sampling that want to finish", "Value of sampling that want to finish", a)
        If Finish < Start Then
            MsgBox "Error!!! value's finish < value's start , Please Insert value's finish of sampling again ", vbCritical,
"Error value"
        End If

    While Finish > n
        If Finish > n Then
            MsgBox "Error!!! value > number of sampling , Please Insert value's finish of sampling again ", vbCritical,
"Error value"
        End If
      Finish = InputBox("Please Input value of sampling that want to finish", "Value of sampling that want to finish", a)
    Wend
    Wend

    '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    For i = Start To Finish
    If i = Start Then
    Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
    Else
    Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbRed
    End If
    Next i
    End Sub

    Private Sub Command3_Click()

    Form5.Show

    End Sub



    Private Sub Command5_Click()
    Name_Folder = Dir.List(Dir.ListIndex)
    Data1.DatabaseName = Name_Folder & "\" & "S1_97new.mdb"
    path = Name_Folder & "\" & "S1_97new.mdb"
    Data1.Refresh
    DBGrid1.Refresh
    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    Data1.Recordset.MoveLast
    check = Data1.Recordset.Sampling.Value
    n = 1
    While "0" <> check
    Data1.Recordset.MovePrevious
    check = Data1.Recordset.Sampling.Value
    n = n + 1
    Wend

    n = n - 1
    b = n / 10
    b = b * 10
    If b <> 0 Then
      b = n - b
      If b <> 0 Then
            b = n / 10
            b = b + 1
            b = (b * 10) / 10
      Else
          b = n / 10
          b = (b * 10) / 10
      End If
    Else
    b = n / 10
    b = 1
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Picture1.Scale (-1, -12)-(13, 1)
        Picture1.BackColor = vbBlack
        Picture1.ForeColor = QBColor(15)
        Picture1.Line (0, 0)-(10, 0)
        Picture1.Line (0, -10)-(0, 0)
        For X = 0 To 10
            Picture1.Line (X, 0)-(X, 0.3)
             Picture1.CurrentX = X - 0.2
             Picture1.Print X * b
        Next X
        For Y = 0 To -10 Step -1
             Picture1.Line (-0.2, Y)-(0, Y)
             Picture1.CurrentX = -0.9
             Picture1.CurrentY = Y - 0.3
            Picture1.Print Y * -10
        Next Y
            Picture1.CurrentX = 10.5
             Picture1.CurrentY = -0.3
             Picture1.Print "Sampling"
            Picture1.CurrentX = -0.2
             Picture1.CurrentY = -11
             Picture1.Print "Degree Celcius"
             Picture1.CurrentX = 5
             Picture1.CurrentY = -11
             Picture1.Print "Temperature Sensor"

End Sub

Private Sub Drive1_Change()

End Sub

Private Sub Form_Activate()

Data1.BackColor = QBColor(11)
DBGrid1.BackColor = QBColor(11)
Drive.BackColor = QBColor(11)
Dir.BackColor = QBColor(11)
Form2.BackColor = QBColor(3)
Label5.BackColor = QBColor(3)
Timer1.Enabled = True
''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Data1.Recordset.MoveLast
check = Data1.Recordset.Sampling.Value
n = 1
While "0" <> check
Data1.Recordset.MovePrevious
check = Data1.Recordset.Sampling.Value
n = n + 1
Wend

n = n - 1
b = n / 10
b = b * 10
If b <> 0 Then
   b = n - b
   If b <> 0 Then
      b = n / 10
      b = b + 1
      b = (b * 10) / 10
   Else
      b = n / 10
      b = (b * 10) / 10
   End If
Else
b = n / 10
b = 1
End If


         Picture1.Scale (-1, -12)-(13, 1)
         Picture1.BackColor = vbBlack
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Picture1.ForeColor = QBColor(15)
        Picture1.Line (0, 0)-(10, 0)
        Picture1.Line (0, -10)-(0, 0)
        For X = 0 To 10
            Picture1.Line (X, 0)-(X, 0.3)
             Picture1.CurrentX = X - 0.2
             Picture1.Print X * b
        Next X
        For Y = 0 To -10 Step -1
             Picture1.Line (-0.2, Y)-(0, Y)
             Picture1.CurrentX = -0.9
             Picture1.CurrentY = Y - 0.3
            Picture1.Print Y * -10
        Next Y
            Picture1.CurrentX = 10.5
             Picture1.CurrentY = -0.3
             Picture1.Print "Sampling"
            Picture1.CurrentX = -0.2
             Picture1.CurrentY = -11
             Picture1.Print "Degree Celcius"
             Picture1.CurrentX = 5
             Picture1.CurrentY = -11
             Picture1.Print "Temperature Sensor"
End Sub

Private Sub Label5_Click()
MsgBox FormVar.var1.Text
End Sub

Private Sub Timer1_Timer()
Label5.Left = Label5.Left - 15

If Label5.Left = -Label5.Width Then
   Label5.Left = Width
End If
End Sub


Private Sub Drive_Change()
Dir.path = Drive.Drive
End Sub
```

Form3

```
Dim i As Integer
Dim b As Integer
Dim n As Integer
Dim check As Integer
Dim Start As Integer
Dim Finish As Integer
Dim a As String
Dim D(1000000) As String
Dim s(1000000) As String
Private Sub Command1_Click()
Form3.Hide
Form1.Show

End Sub

Private Sub Command2_Click() ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If check = 1 Then
For i = Start To Finish
If i = Start Then
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
Else
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
End If
Next i
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Data2.Recordset.MoveFirst

For i = 0 To n
D(i) = Data2.Recordset.Data.Value
s(i) = Data2.Recordset.Sampling.Value
Data2.Recordset.MoveNext
Next i

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If n = 0 Then
MsgBox "Error!!!  No Data , Cann't to plot graph", vbCritical, "No Data!!!"
Exit Sub
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Start = 0
Finish = 0

While Start = 0
check = 1
Start = InputBox("Please Input value of sampling that want to start", "Value of sampling that want to start", "1")
   If Start = 0 Then
   MsgBox "Error value = 0 , Please Insert value's start of sampling again ", vbCritical
   End If

   While Start > n
      If Start > n Then
      MsgBox "Error!!! value > number of sampling , Please Insert value's start of sampling again ", vbCritical, "Error
value"
      End If
   Start = InputBox("Please Input value of sampling that want to start", "Value of sampling that want to start", "1")
   Wend
Wend
a = Str(n)

While Finish < Start
Finish = InputBox("Please Input value of sampling that want to finish", "Value of sampling that want to finish", a)
      If Finish < Start Then
         MsgBox "Error!!! value's finish < value's start , Please Insert value's finish of sampling again ", vbCritical,
"Error value"
      End If

While Finish > n
      If Finish > n Then
         MsgBox "Error!!! value > number of sampling , Please Insert value's finish of sampling again ", vbCritical,
"Error value"
      End If
      Finish = InputBox("Please Input value of sampling that want to finish", "Value of sampling that want to finish",
a)
Wend
Wend
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
For i = Start To Finish
If i = Start Then
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
Else
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbRed
End If
Next i
End Sub

Private Sub Command3_Click()
Form6.Show

End Sub

Private Sub Command5_Click()
Name_Folder = Dir.List(Dir.ListIndex)
Data2.DatabaseName = Name_Folder & "\" & "S2_97new.mdb"
path = Name_Folder & "\" & "S2_97new.mdb"
Data2.Refresh
DBGrid1.Refresh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Data2.Recordset.MoveLast
check = Data2.Recordset.Sampling.Value
n = 1
While "0" <> check
Data2.Recordset.MovePrevious
check = Data2.Recordset.Sampling.Value
n = n + 1
Wend

n = n - 1

If n >= 10 Then
b = n / 10
Else
b = 0
End If

b = b * 10
If b <> 0 Then
   b = n - b
   If b <> 0 Then
        b = n / 10
        b = b + 1
        b = (b * 10) / 10
   Else
      b = n / 10
      b = (b * 10) / 10
   End If
Else
b = n / 10
b = 1
End If

       Picture1.Scale (-1, -12)-(13, 1)
      Picture1.BackColor = vbBlack
      Picture1.ForeColor = QBColor(15)
      Picture1.Line (0, 0)-(10, 0)
      Picture1.Line (0, -10)-(0, 0)
      For X = 0 To 10
           Picture1.Line (X, 0)-(X, 0.3)
            Picture1.CurrentX = X - 0.2
            Picture1.Print X * b
      Next X
      For Y = 0 To -10 Step -1
            Picture1.Line (-0.2, Y)-(0, Y)
            Picture1.CurrentX = -0.9
            Picture1.CurrentY = Y - 0.3
           Picture1.Print Y * -10
      Next Y
          Picture1.CurrentX = 10.5
            Picture1.CurrentY = -0.3
            Picture1.Print "Sampling"
           Picture1.CurrentX = -0.2
            Picture1.CurrentY = -11
            Picture1.Print "%"
            Picture1.CurrentX = 5
            Picture1.CurrentY = -11
            Picture1.Print "Soil Humidity Sensor"
End Sub

Private Sub Drive_Change()
Dir.path = Drive.Drive
End Sub

Private Sub Form_Activate()

Data2.BackColor = QBColor(11)
DBGrid1.BackColor = QBColor(11)
Drive.BackColor = QBColor(11)
Dir.BackColor = QBColor(11)
Form3.BackColor = QBColor(3)
Label5.BackColor = QBColor(3)
Timer1.Enabled = True
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Data2.Recordset.MoveLast
check = Data2.Recordset.Sampling_Value
n = 1
While "0" <> check
Data2.Recordset.MovePrevious
check = Data2.Recordset.Sampling_Value
n = n + 1
Wend

n = n - 1
b = n / 10
b = b * 10
If b <> 0 Then
b = n / 10
b = (b * 10) / 10
Else
b = n / 10
b = 1
End If


    Picture1.Scale (-1, -12)-(13, 1)
    Picture1.BackColor = vbBlack
    Picture1.ForeColor = QBColor(15)
    Picture1.Line (0, 0)-(10, 0)
    Picture1.Line (0, -10)-(0, 0)
    For X = 0 To 10
        Picture1.Line (X, 0)-(X, 0.3)
        Picture1.CurrentX = X - 0.2
        Picture1.Print X * b
    Next X
    For Y = 0 To -10 Step -1
        Picture1.Line (-0.2, Y)-(0, Y)
        Picture1.CurrentX = -0.9
        Picture1.CurrentY = Y - 0.3
        Picture1.Print Y * -10
    Next Y
        Picture1.CurrentX = 10.5
        Picture1.CurrentY = -0.3
        Picture1.Print "Sampling"
        Picture1.CurrentX = -0.2
        Picture1.CurrentY = -11
        Picture1.Print "%"
        Picture1.CurrentX = 5
        Picture1.CurrentY = -11
        Picture1.Print "Soil Humidity Sensor"
End Sub


Private Sub Timer1_Timer()
Label5.Left = Label5.Left - 15

If Label5.Left = -Label5.Width Then
   Label5.Left = Width
End If
End Sub
```

Form4

```
Dim i As Integer
Dim b As Integer
Dim n As Integer
Dim check As Integer
Dim Start As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim Finish As Integer
Dim a As String
Dim D(1000000) As String
Dim s(1000000) As String
Private Sub Command1_Click()
Form4.Hide
Form1.Show
End Sub

Private Sub Command2_Click()
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If check = 1 Then
For i = Start To Finish
If i = Start Then
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
Else
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
End If
Next i
End If

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Data3.Recordset.MoveFirst

For i = 0 To n
D(i) = Data3.Recordset.Data.Value
s(i) = Data3.Recordset.Sampling.Value
Data3.Recordset.MoveNext
Next i

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If n = 0 Then
MsgBox "Error!!! No Data , Cann't to plot graph", vbCritical, "No Data!!!"
Exit Sub
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Start = 0
Finish = 0

While Start = 0
check = 1
   Start = InputBox("Please Input value of sampling that want to start", "Value of sampling that want to start", "1")

   If Start = 0 Then
      MsgBox "Error!!! value = 0 , Please Insert value's start of sampling again ", vbCritical, "Error value"
   End If

   While Start > n
      If Start > n Then
      MsgBox "Error!!! value > number of sampling , Please Insert value's start of sampling again ", vbCritical, "Error
value"
      End If
   Start = InputBox("Please Input value of sampling that want to start", "Value of sampling that want to start", "1")
   Wend
Wend

a = Str(n)

While Finish < Start
Finish = InputBox("Please Input value of sampling that want to finish", "Value of sampling that want to finish", a)
   If Finish < Start Then
      MsgBox "Error!!! value's finish < value's start , Please Insert value's finish of sampling again ", vbCritical,
"Error value"
   End If

   While Finish > n
      If Finish > n Then
      MsgBox "Error!!! value > number of sampling , Please Insert value's finish of sampling again ", vbCritical,
"Error value"
      End If
   Finish = InputBox("Please Input value of sampling that want to finish", "Value of sampling that want to finish", a)
   Wend
Wend

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For i = Start To Finish
If i = Start Then
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbBlack
Else
Picture1.Line (s(i - 1) / b, -(D(i - 1) / 10))-(s(i) / b, -(D(i) / 10)), vbRed
End If
Next i
End Sub


Private Sub Command3_Click()
Form7.Show

End Sub

Private Sub Command5_Click()
Name_Folder = Dir.List(Dir.ListIndex)
Data3.DatabaseName = Name_Folder & "\" & "S3_97new.mdb"
path = Name_Folder & "\" & "S3_97new.mdb"
Data3.Refresh
DBGrid1.Refresh
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Data3.Recordset.MoveLast
check = Data3.Recordset.Sampling.Value
n = 1
While "0" <> check
Data3.Recordset.MovePrevious
check = Data3.Recordset.Sampling.Value
n = n + 1
Wend

n = n - 1
b = n / 10
b = b * 10
If b <> 0 Then
   b = n - b
   If b <> 0 Then
        b = n / 10
        b = b + 1
        b = (b * 10) / 10
   Else
      b = n / 10
      b = (b * 10) / 10
   End If
Else
b = n / 10
b = 1
End If

      Picture1.Scale (-1, -12)-(13, 1)
      Picture1.BackColor = vbBlack
      Picture1.ForeColor = QBColor(15)
      Picture1.Line (0, 0)-(10, 0)
      Picture1.Line (0, -10)-(0, 0)
      For X = 0 To 10
            Picture1.Line (X, 0)-(X, 0.3)
             Picture1.CurrentX = X - 0.2
             Picture1.Print X * b
      Next X
      For Y = 0 To -10 Step -1
             Picture1.Line (-0.2, Y)-(0, Y)
             Picture1.CurrentX = -0.9
             Picture1.CurrentY = Y - 0.3
            Picture1.Print Y * -10
      Next Y
           Picture1.CurrentX = 10.5
             Picture1.CurrentY = -0.3
             Picture1.Print "Sampling"
            Picture1.CurrentX = -0.2
             Picture1.CurrentY = -11
             Picture1.Print "Lux"
             Picture1.CurrentX = 5
             Picture1.CurrentY = -11
             Picture1.Print "Light Intensity Sensor"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub Drive_Change()
Dir.path = Drive.Drive
End Sub

Private Sub Form_Activate()

Data3.BackColor = QBColor(11)
DBGrid1.BackColor = QBColor(11)
Drive.BackColor = QBColor(11)
Dir.BackColor = QBColor(11)
Form4.BackColor = QBColor(3)
Label5.BackColor = QBColor(3)
Timer1.Enabled = True
'''''''''''''''''''''''''''''''''''''''''''''''''

'''''''''''''''''''''''''''''''''''''''''''''''''
Data3.Recordset.MoveLast
check = Data3.Recordset.Sampling.Value
n = 1
While "0" <> check
Data3.Recordset.MovePrevious
check = Data3.Recordset.Sampling.Value
n = n + 1
Wend

n = n - 1
b = n / 10
b = b * 10
If b <> 0 Then
b = n / 10
b = (b * 10) / 10
Else
b = n / 10
b = 1
End If

    Picture1.Scale (-1, -12)-(13, 1)
    Picture1.BackColor = vbBlack
    Picture1.ForeColor = QBColor(15)
    Picture1.Line (0, 0)-(10, 0)
    Picture1.Line (0, -10)-(0, 0)
    For X = 0 To 10
        Picture1.Line (X, 0)-(X, 0.3)
         Picture1.CurrentX = X - 0.2
         Picture1.Print X * b
    Next X
    For Y = 0 To -10 Step -1
         Picture1.Line (-0.2, Y)-(0, Y)
         Picture1.CurrentX = -0.9
         Picture1.CurrentY = Y - 0.3
        Picture1.Print Y * -10
    Next Y
        Picture1.CurrentX = 10.5
         Picture1.CurrentY = -0.3
         Picture1.Print "Sampling"
        Picture1.CurrentX = -0.2
         Picture1.CurrentY = -11
         Picture1.Print "Lux"
         Picture1.CurrentX = 5
         Picture1.CurrentY = -11
         Picture1.Print "Light Intensity Sensor"
End Sub



Private Sub Timer1_Timer()
Label5.Left = Label5.Left - 15

If Label5.Left = -Label5.Width Then
   Label5.Left = Width
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
End Sub
```

Form5

```
Private Sub Command1_Click()
Command1.Visible = False
Command2.Visible = False
Command3.Visible = False
With Printer '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
     .ScaleMode = 5
     .CurrentX = 1
     .CurrentY = 1
     .FontName = "MS Sans Serif"
  End With
  Printer.FontBold = False
  Printer.FontSize = 10
  Form5.PrintForm
  Printer.EndDoc ''
  Form5.Hide
End Sub

Private Sub Command2_Click()
Form5.Hide
End Sub

Private Sub Command3_Click()
Data1.DatabaseName = path
Data1.Refresh
DBGrid1.Refresh
End Sub

Private Sub Form_Load()
Dim W As Integer, H As Integer
Form5.BackColor = QBColor(3)
Data1.BackColor = QBColor(11)
DBGrid1.BackColor = QBColor(11)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
W = 0
H = 0
W = Me.Width
H = Me.Height
Me.Width = 1
Me.Height = 1
Me.Show

Do While True
  DoEvents
  If Not Me.Width >= W Then
     Me.Width = Me.Width + 40
  Else
     Me.Width = W
  End If

  If Not Me.Height >= H Then
    Me.Height = Me.Height + 50
  Else
     Me.Height = H
  End If

  Me.Top = (Screen.Height - Me.Height) / 45
  Me.Left = (Screen.Width - Me.Width) / 45
  If Me.Height >= H And Me.Width >= W Then
     Exit Do
  End If
Loop
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form6

```
Private Sub Command1_Click()
Command1.Visible = False
Command2.Visible = False
Command3.Visible = False
With Printer ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
      .ScaleMode = 5
      .CurrentX = 1
      .CurrentY = 1
      .FontName = "MS Sans Serif"
   End With
   Printer.FontBold = False
   Printer.FontSize = 10
   Form6.PrintForm
   Printer.EndDoc ''''
   Form6.Hide
End Sub

Private Sub Command2_Click()
Form6.Hide
End Sub

Private Sub Command3_Click()
Data2.DatabaseName = path
Data2.Refresh
DBGrid2.Refresh
End Sub

Private Sub Form_Load()
Dim W As Integer, H As Integer
Form6.BackColor = QBColor(3)
Data2.BackColor = QBColor(11)
DBGrid2.BackColor = QBColor(11)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
W = Me.Width
H = Me.Height
Me.Width = 1
Me.Height = 1
Me.Show

Do While True
   DoEvents
   If Not Me.Width >= W Then
      Me.Width = Me.Width + 40
   Else
      Me.Width = W
   End If

   If Not Me.Height >= H Then
     Me.Height = Me.Height + 50
   Else
      Me.Height = H
   End If

   Me.Top = (Screen.Height - Me.Height) / 45
   Me.Left = (Screen.Width - Me.Width) / 45
   If Me.Height >= H And Me.Width >= W Then
      Exit Do
   End If
Loop
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form7

```
Private Sub Command1_Click()
Command1.Visible = False
Command2.Visible = False
Command3.Visible = False
With Printer ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
      .ScaleMode = 5
      .CurrentX = 1
      .CurrentY = 1
      .FontName = "MS Sans Serif"
   End With
   Printer.FontBold = False
   Printer.FontSize = 10
   Form7.PrintForm
   Printer.EndDoc ''
   Form7.Hide
End Sub

Private Sub Command2_Click()
Form7.Hide
End Sub

Private Sub Command3_Click()
Data3.DatabaseName = path
Data3.Refresh
DBGrid3.Refresh
End Sub

Private Sub Form_Load()
Dim W As Integer, H As Integer
Form7.BackColor = QBColor(3)
Data3.BackColor = QBColor(11)
DBGrid3.BackColor = QBColor(11)
''''''''''''''''''''''''''''''''''''''''''''''''''''
W = Me.Width
H = Me.Height
Me.Width = 1
Me.Height = 1
Me.Show

Do While True
   DoEvents
   If Not Me.Width >= W Then
      Me.Width = Me.Width + 40
   Else
      Me.Width = W
   End If

   If Not Me.Height >= H Then
     Me.Height = Me.Height + 50
   Else
      Me.Height = H
   End If

   Me.Top = (Screen.Height - Me.Height) / 45
   Me.Left = (Screen.Width - Me.Width) / 45
   If Me.Height >= H And Me.Width >= W Then
      Exit Do
   End If
Loop
''''''''''''''''''''''''''''''''''''''''''''''''''''''

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form8

```
Dim W As Integer, H As Integer

Private Sub Command1_Click()
Dim Folder As New FileSystemObject
Dim Name_Folder As String
Name_Folder = InputBox("Please Input name's folder", "Create Folder", "My New Folder")
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If Folder.FolderExists(Dir.List(Dir.ListIndex) & "\" & Name_Folder) Then
        MsgBox "This name has had already", vbCritical, "Error name"
        Exit Sub
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Folder.CreateFolder (Dir.List(Dir.ListIndex) & "\" & Name_Folder)
Name_Folder = Dir.List(Dir.ListIndex) & "\" & Name_Folder
Dir.Refresh
Folder.CopyFile "C:\Documents and Settings\All Users\Documents\â»Ãà'÷»Õ4
âÍ&ÍèÍ'\VB SUKSAI\Database\*.mdb", Name_Folder
Dir.Refresh
End Sub

Private Sub Command2_Click()
Form8.Hide
End Sub


Private Sub Form_Load()
Dim W As Integer, H As Integer

Drive.BackColor = QBColor(11)
Dir.BackColor = QBColor(11)
Frame.BackColor = QBColor(11)

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

W = Me.Width
H = Me.Height
Me.Width = 1
Me.Height = 1
Me.Show

Do While True
  DoEvents
  If Not Me.Width >= W Then
    Me.Width = Me.Width + 40
  Else
    Me.Width = W
  End If

  If Not Me.Height >= H Then
    Me.Height = Me.Height + 50
  Else
    Me.Height = H
  End If

  Me.Top = (Screen.Height - Me.Height) / 2
  Me.Left = (Screen.Width - Me.Width) / 2
  If Me.Height >= H And Me.Width >= W Then
    Exit Do
  End If
Loop

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. นคร ภักดีชาติ, ธีรบูลย์ หล่อวิเชียรรุ่ง, ชัยวัฒน์ ลิ้มพรจิตวิไล.
"ปฏิบัติการไมโครคอนโทรลเลอร์ MSC-51 ด้วยภาษาซี"
บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

2. อภิชาติ ภู่พลับ.
"สนุกกับการประยุกต์ใช้ Visual Basic"
บริษัท อินโฟเพรส

3. อภิชาติ ภู่พลับ.
"เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic"
บริษัท อินโฟเพรส

4. สัจจะ จรัสรุ่งรวีวร
"คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.0"
บริษัท อินโฟเพรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้