

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องบันทึกข้อมูลชนิดพกพา

PORTABLE DATA STORAGE



โดย

นางสาวหฤทัย ศรีสวัสดิ์

นายอนุรักษ์ สุมิพันธ์

นายอนุรัตน์ กองแก้ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน 32575
วัน, เดือน, ปี 8 พ.ค. 2542

สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกข้อมูลชนิดพกพา
PORTABLE DATA STORAGE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องบันทึกข้อมูลชนิดพกพา

PORTABLE DATA STORAGE

ผู้จัดทำ

1. นางสาวหุทัย ศรีสวัสดิ์ 39013067
2. นายอนุรักษ์ สุมิพันธ์ 39013068
3. นายอนุรัตน์ กองแก้ว 39013069


(รศ. สมยศ จุณณะปิยะ)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกข้อมูลชนิดพกพา

PORTABLE DATA STORAGE

โดย นางสาวหฤทัย ศรีสวัสดิ์ 39013067

นายอนรรักษ์ สุมิพันธ์ 39013068

นายอนรรคน์ กองแก้ว 39013069

อาจารย์ที่ปรึกษา รศ. สมยศ จุณณะปิยะ

บทคัดย่อ

โครงการนี้เป็นการสร้างเครื่องบันทึกข้อมูลขนาดเล็ก สามารถพกพาได้ ประกอบด้วย ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 เป็นส่วนควบคุมการทำงาน, คีย์บอร์ด, ชุดแสดงผล, LCD และชุดส่งข้อมูลอนุกรมไปยังพอร์ทอนุกรมของเครื่อง ไมโครคอมพิวเตอร์ ซึ่งเครื่องบันทึกข้อมูลชนิดพกพานี้ ถูกออกแบบเพื่อใช้ในการบันทึกข้อมูลการใช้ไฟฟ้า และน้ำประปาจากมิเตอร์ และส่งข้อมูลเข้าไมโครคอมพิวเตอร์ ซึ่งมีการทำฐานข้อมูลเพื่อประโยชน์ในการคิดราคาค่าไฟฟ้า หรือค่าน้ำประปา และจัดพิมพ์ใบเสร็จรับเงินได้

ABSTRACT

This project is made small recorder machine which easy to use. The main part is microcontroller MCS-51 which use for controlling operation. The other component are 4x4 matrix keyboard, the LCD for display output, transmission serial data module transmitting asynchronous data to serial port of microcomputer.

THE PORTABLE DATA STORAGE is designed for recording electrical consumption of watt-hour meter and water consumption of water meter. Latter it will load this data to microcomputer which perform database for calculating payment of electrical or water consumption. Finally, it's designed to be able to print a bill.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์การศึกษา	1
1.3 ประโยชน์ที่ได้รับ	1
1.4 ขอบเขตของโครงการ	1
บทที่ 2 ทฤษฎีและหลักการของโครงการ	2
2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	2
2.2 การต่อคีย์บอร์ด	3
2.3 DOT MATRIX LCD MODULE	5
2.4 การเชื่อมต่อหน่วยความจำข้อมูลภายนอก (RAM)	9
2.5 การรับส่งข้อมูลอนุกรม	11
2.6 การสร้างฐานเวลาให้กับ MCS-51	13
2.7 การเขียนโปรแกรมด้วยภาษาแอสเซมบลี	15
บทที่ 3 การคำนวณและการสร้าง	23
3.1 Block Diagram	23
3.2 การคำนวณ Address	24
3.3 ลักษณะของตัวเครื่องบันทึกข้อมูล	26
บทที่ 4 การทดลองและผลการทดลอง	27
4.1 ผลการทดลองของส่วนเครื่องบันทึกข้อมูลชนิดทกทา	27
4.2 ผลการทดลอง โปรแกรมส่วนรับข้อมูลและแสดงผล	36
บทที่ 5 บทวิจารณ์และบทสรุป	44
5.1 บทสรุป	44
5.2 ประโยชน์ที่ได้รับจากการทำโครงการ	44
5.3 แนวทางการพัฒนาโครงการ	44
ภาคผนวก ก	45
ภาคผนวก ข	49
ภาคผนวก ค	67
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

สารบัญรูปภาพ

รูปที่ 2.1 แสดงการจัดวางขาของ 8051	2
รูปที่ 2.2 ภาพสัญญาณการเปลี่ยนแปลงจากการเกิดการกระเพื่อมของหน้าสัมผัสสวิตช์	4
รูปที่ 2.3 แสดง Single keyboard	4
รูปที่ 2.4 แสดง Matrix Keyboard	4
รูปที่ 2.5 แสดงการต่อ LCD MODULE แบบ MEMORY MAP	7
รูปที่ 2.6 แสดงการต่อ LCD MODULE แบบ I/O PORT	8
รูปที่ 2.7 ภาพเวลาแสดงลำดับเหตุการณ์การอ่านข้อมูลจากหน่วยความจำ RAM	9
รูปที่ 2.8 ภาพเวลาแสดงลำดับเหตุการณ์การเขียนข้อมูลลงในหน่วยความจำ RAM	10
รูปที่ 2.9 ภาพแสดงการติดต่อกับหน่วยความจำ RAM ของ 8051	11
รูปที่ 2.10 หน้าจอของเดสก์ท็อปโดยรวม	15
รูปที่ 2.11 หน้าต่างหลัก	16
รูปที่ 2.12 สปีดบาร์ (Speed bar)	17
รูปที่ 2.13 หน้าต่างฟอร์ม (Form)	17
รูปที่ 2.14 หน้าต่างยูนิต (Unit)	18
รูปที่ 2.15 หน้าต่าง Object Inspector	19
รูปที่ 2.16 หน้าต่าง Save Unit As	20
รูปที่ 2.17 หน้าต่าง Save Project As	20
รูปที่ 3.1 แสดง Block Diagram ของเครื่องบันทึกข้อมูล	23
รูปที่ 3.2 แสดงการจัด MEMORY ของเครื่องบันทึกข้อมูล	24
รูปที่ 3.3 ลักษณะของตัวเครื่องบันทึกข้อมูล	26
รูปที่ 4.1 แสดงการเข้าสู่โปรแกรมแสดงผล	36
รูปที่ 4.2 แสดงส่วนที่ให้ใส่รหัสผ่าน	36
รูปที่ 4.3 แสดงหน้าจอในการเลือกฟังก์ชันการทำงาน	37
รูปที่ 4.4 แสดงตารางรับข้อมูลไฟฟ้า	38
รูปที่ 4.5 แสดงตารางรับข้อมูลประปา	38
รูปที่ 4.6 แสดงการจัดการฐานข้อมูลไฟฟ้า	39
รูปที่ 4.7 แสดงการจัดการฐานข้อมูลประปา	40
รูปที่ 4.8 แสดงส่วนการค้นหา	41
รูปที่ 4.9 แสดงส่วนการขอลูฮอด	41
รูปที่ 4.10 แสดงกรณีที่ใช้ข้อมูลผิด	41

สารบัญรูปภาพ(ต่อ)

รูปที่ 4.11 แสดง MENU การสั่งพิมพ์ใบแจ้งหนี้	42
รูปที่ 4.12 การตั้งค่าเครื่องพิมพ์	42
รูปที่ 4.13 แสดงการเปลี่ยนรหัส	43
รูปที่ 4.14 แสดงการป้อนรหัสใหม่	43
รูปที่ 4.15 แสดงการจบการทำงาน	43



สารบัญตาราง

ตารางที่ 2.1 ขาดัญญาณของ LCD MODULE	6
ตารางที่ 2.2 ชุดคำสั่งควบคุมและการแสดงข้อความ	8



บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ในการเก็บข้อมูลแบบตัวเลขในลักษณะต่างๆ เช่น ในการเก็บค่าไฟฟ้าหรือน้ำประปานั้นเมื่อก่อนอาศัยการจดบันทึกลงบนกระดาษ หลังจากนั้น จึงนำข้อมูลที่ได้นำไปให้พนักงานที่มีหน้าที่คีย์ข้อมูลลงคอมพิวเตอร์ ซึ่งวิธีนี้มีข้อเสียคือ เสียเวลา และมีขั้นตอนในการทำงานซับซ้อนกัน จึงได้มีการพัฒนาเครื่องบันทึกข้อมูลชนิดพกพาขึ้น โดยข้อดีของเครื่องนี้คือ เป็นการคีย์ข้อมูลเข้าไปในเรื่องครั้งเดียวนั้นคือ เวลาในช่วงการไปจดบันทึกค่าของไฟฟ้าหรือประปาตามบ้านแล้วบันทึกเข้าไปในเครื่อง และเมื่อได้ข้อมูลแล้วนำข้อมูลไปเก็บรวบรวมไว้ในฐานข้อมูลเพื่อนำไปคิดคำนวณหาค่าไฟฟ้าและประปา

1.2 วัตถุประสงค์การศึกษา

1.2.1 ศึกษาการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 และการต่อร่วมกับอุปกรณ์อื่น ๆ

1.2.2 จัดทำเครื่องบันทึกข้อมูลชนิดพกพาที่มีความสามารถในการรับ และ ส่งข้อมูล แบบอนุกรมได้อย่างมีประสิทธิภาพ

1.3 ประโยชน์ที่ได้รับ

1.3.1 เข้าใจการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

1.3.2 สามารถพัฒนาระบบงานให้มีความรวดเร็ว,แน่นอน และมีประสิทธิภาพมากยิ่งขึ้น

1.4 ขอบเขตของโครงการ

1.4.1 ออกแบบและสร้างเครื่องบันทึกข้อมูลชนิดพกพาที่มีขนาดเล็ก สามารถนำไปใช้งานได้จริง

1.4.2 เครื่องบันทึกข้อมูลชนิดพกพาที่มีความสามารถในการบันทึกข้อมูลจำนวนมิเตอร์ไฟฟ้า 850 มิเตอร์ จำนวนมิเตอร์น้ำประปา 850 มิเตอร์

1.4.3 เครื่องบันทึกข้อมูลชนิดพกพาจะส่งข้อมูลอนุกรมแบบอะซิงโครนัสกับเครื่องไมโครคอมพิวเตอร์ ทางพอร์ตอนุกรม โดยใช้มาตรฐาน RS-232C

บทที่ 2

ทฤษฎีและหลักการของโครงการงาน

2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์แบบชิปเดี่ยวตระกูล MCS-51 นี้ผลิตโดยบริษัทอินเทล มีอยู่ด้วยกันหลายเบอร์ ลักษณะงานที่เหมาะสมกับการนำไมโครคอนโทรลเลอร์ไปใช้งาน มักจะเป็นงานประยุกต์ที่เกี่ยวข้องกับการควบคุม หรือจัดการสัญญาณอินพุต / เอาท์พุทของวงจรถอนิกส์และวงจรถิศจิตต่าง ๆ คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

-ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว

-มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์ สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์

-มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไปมี 256 ไบต์

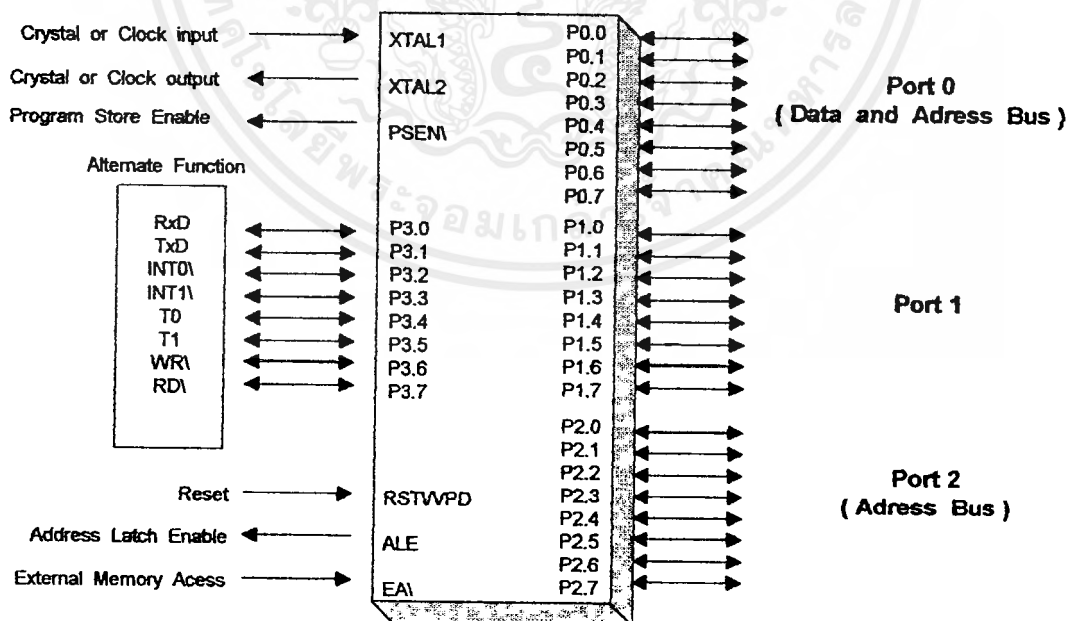
-หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์

-มีไทม์เมอร์ เคาน์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด

-รับอินเทอร์รัพท์ได้ 6 แหล่ง 5 เวกเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวกเตอร์

-มีพอร์ทรับส่งข้อมูลอนุกรม (UART) 2 พอร์ท แบบ Full Duplex เลือกรูปได้ 4 โหมด

-มีคำสั่งในการทำ AND,OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต



รูปที่ 2.1 แสดงการจัดวางขาของ 8051

ไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา แต่ละขา สัญญาจะมีหน้าที่ที่ ระบุชัดเจนตามสัญลักษณ์ที่อยู่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขา สัญญาที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง

8051 มีวงจรออสซิลเลเตอร์อยู่ภายใน สำหรับการสร้างพัลส์ของสัญญาณนาฬิกา ซึ่งจะนำไปเป็น ฐานเวลา หรือการกำหนดจังหวะการทำงานของหน่วยการทำงานทั้งหมดให้สอดคล้องกัน

2.2 การต่อคีย์บอร์ด

โดยปกติการรับสัญญาณอินพุตจากกลุ่มของสวิตช์ (คีย์บอร์ด) มักใช้เส้นสัญญาณเดียวในการ จัดการ แต่อย่างไรก็ตามหากว่าจำนวนของสวิตช์มีมากเกินไปเกินกว่า 8 ตัวแล้ว ก็ต้องใช้เส้นสัญญาณจากพอร์ท มากกว่าหนึ่งพอร์ทเข้าทำหน้าที่ดูแล

หลักในการคำนวณหาจำนวนเส้นสัญญาณที่ต้องใช้คือ หากเมตริกซ์ขนาด $M \times N$ ก็จะใช้เส้น สัญญาณจำนวนเพียง $M+N$ เส้น

หลักการของโปรแกรมในการตรวจสอบสภาพการกดสวิตช์ จะใช้หลักการ Keyboard Scan โดย เริ่มต้นต้องทำให้สถานะลอจิกของทุกเส้นสัญญาณในแนวคอลัมน์เป็นค่า 1 เสียก่อน จากนั้นจึงจะเริ่ม คำนวณค่าลอจิกของเส้นสัญญาณในแถวแรก (Row 0) เป็นลอจิก 0 เมื่อได้อ่านค่าข้อมูลของสัญญาณ ในแถวคอลัมน์เข้ามาตรวจสอบ หากพบว่ามีเส้นสัญญาณใดมีค่าลอจิกค่า 0 แสดงว่ามีการกดสวิตช์ ที่ตั้งอยู่ ระหว่างคอลัมน์นั้นกับเส้นสัญญาณในแนวแถวแรก ทำให้ช่องสัญญาณทั้งสองเชื่อมต่อเข้าด้วยกัน หากว่า ไม่มีเส้นสัญญาณใดเป็นลอจิกต่ำลงแสดงว่าไม่มีการกดสวิตช์ใด ๆ ใน แถวแรกเลย จากนั้นก็ดำเนินการ ซ้ำเช่นนี้กับเส้นสัญญาณในแถวอื่น ๆ อีกในรอบถัดไป จนกระทั่งครบทุกแถว

ในการนำไปใช้งานจะต้องเพิ่ม โปรแกรมย่อยหน่วงเวลา (Time Delay) ตักระยะหนึ่ง ก่อนที่จะ กลับมาวนรอบการอ่านสถานะสวิตช์ในรอบใหม่ เนื่องจากว่าในสถานะความเป็นจริงแล้ว ลักษณะสมบัติ เริงกลของอุปกรณ์สวิตช์ ภายหลังจากการกดสวิตช์หนึ่งครั้ง ก็จะมีสภาพการกระเพื่อมของหน้าสัมผัส (Contact Debounce) ทำให้มีลักษณะเหมือนการเปิดและปิดหน้าสัมผัสหลาย ๆ ครั้ง ซึ่งจะมีผลทำให้ โปรแกรมดำเนินการผิดพลาดได้ เนื่องจากมองเห็นว่ามีการกดสวิตช์หลายครั้ง ดังนั้นจึงต้องทำการหน่วง เวลาโปรแกรมสักกระยะหนึ่ง เพื่อเป็นการบรรเทาปัญหาของการกระเพื่อม (Debounce) โดยใช้กระบวนการทาง คำนวณค่าได้แวย์

086310⁰๙๙

2.3 DOT MATRIX LCD MODULE

อุปกรณ์ในปัจจุบันนี้ในส่วนแสดงผลนั้นจะใช้ LCD เสียเป็นส่วนใหญ่ว่าจะเป็นเครื่องเล่น VEDIO , เครื่องถ่ายเอกสาร , เครื่องมือวัดคุมต่างๆ , เครื่องคอมพิวเตอร์ เราพอจะแบ่ง DOT MATRIX LCD MODULE นี้ออกได้เป็นพวกๆ ดังนี้

1. CHARACTER LCD MODULE
2. GRAPHIC LCD MODULE
3. SEGMENT DISPLAY TYPE LCD MODULE

โดยในแต่ละแบบนี้ก็มีส่วนประกอบใหญ่ๆ แบ่งได้เป็น

1. DOT MATRIX LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการเปิดและปิดตัวเองกับแสงก็คือ ส่วนของที่เป็นตัวกระจกบรรจุผลึก

2. DRIVE เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่งโดยมีเบอร์ที่นิยมใช้ใน LCD MODULE เช่น HD44100H , MSM5259

3. CONTROLLER เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD MODUDE ให้ทำงานแสดงผลต่างๆ เช่น การลบจอภาพ , การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ IC ที่นิยมใช้กันคือ HD4478 ซึ่งจะใช้แบบ CHARACTER LCD MODULE เป็นส่วนใหญ่ และเบอร์ IC HD61830 จะใช้ในแบบ GRAPHIC LCD MODULE

ในการศึกษาการทำงานและใช้งาน LCD MODULE นั้นไม่ใช่เรื่องยากเลยถ้าเราสามารถทำความเข้าใจในส่วนของ CONTROLLER ได้ก็เพียงพอแล้วและโดยมาก LCD MODULE ในแต่ละบริษัทแล้วจะใช้ตัว CONTROLLER ที่มีหลักการทำงานเหมือนกันเป็นส่วนใหญ่และใน LCD MODULE แต่ละขนาดจำนวนตัวอักษรหรือจำนวนบรรทัดก็มีหลักการทำงานแบบเดียวกันทั้งหมด IC ที่นิยมใช้มากที่สุดตัวหนึ่งที่เป็น CONTROLLER LCD ก็คือ เบอร์ HD44780 โดยรูปแบบการทำงานของมันได้เป็นมาตรฐานให้กับ CONTROLLER LCD ตัวอื่นๆ ด้วย

HD44780 เป็นไอซี LSI ตัวหนึ่งใช้ควบคุม LCD โดยแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่างๆ ตัวมันเองสามารถต่อใช้งานแบบ 4 BIT หรือ 8 BIT ก็ได้ โดยถ้าเราต่อแบบ 4 BIT จะต่อใช้งานที่ DB7-DB4 เท่านั้น โดยข้อมูลครั้งแรกที่ส่งนั้น HD44780 จะถือเป็นข้อมูลที่ส่งมานั้นเป็นข้อมูล 4 BIT ล่าง

คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปได้ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วย DOT MATRIX ขนาด 5x8 DOT
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะ แบบ MEMORY MAP และ

แบบผ่าน 8255 PORT ซึ่งจะใช้ขาสัญญาณทั้งหมด 14 PIN (ขั้วต่อ 16 PIN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การใช้งานง่ายและสะดวก ระบบ ไมโครคอนโทรลเลอร์เพียงส่งข้อมูลให้กับ LCD MODULE เท่านั้นข้อความที่จะปรากฏบนแผงแสดง และจะค้างไว้ตลอดทำให้ไม่ต้องเสียเวลาของระบบ

5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR, DISPLAY, HOME, CURSOR, ON OFF, BLANK, CHARACTER และอื่นๆ อีก

6. สามารถแสดงผลเป็นภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว

7. กินกระแสต่ำและมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์ เท่านั้น

ตารางที่ 2.1 ขาสัญญาของ LCD MODULE

PIN	SYMBOL	LEVEL	FUNCTION
1	Vss	-	0 V GND
2	Vcc	-	+5 V POWER SUPPLY
3	Vee	-	+ V FOR LIQUID CRYSTAL DRIVE
4	RS	H/L	REGISTER SELECT H: DATA INPUT L: INTRUCTION INPUT
5	R/W	H/L	H: DATA READ L: DATA WRITE
6	E	H	ENABLE SIGNAL (L->H)
7	DB 0	H/L	DATA BUS BIT 0
8	DB 1	H/L	DATA BUS BIT 1
9	DB 2	H/L	DATA BUS BIT 2
10	DB 3	H/L	DATA BUS BIT 3
11	DB 4	H/L	DATA BUS BIT 4
12	DB 5	H/L	DATA BUS BIT 5
13	DB 6	H/L	DATA BUS BIT 6
14	DB 7	H/L	DATA BUS BIT 7

2.3.1 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์

LCD MODULE จะต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะดังได้กล่าวไปแล้ว ซึ่งทั้งสองแบบนี้จะมีข้อดีและข้อเสียแตกต่างกันไป โดยแต่ละแบบจะมีหลักการดังนี้

2.3.1.1 การต่อแบบ MEMORY MAP

1. สามารถต่อเข้ากับ CHIP เบอร์ต่างๆ ไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครคอนโทรลเลอร์มองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที

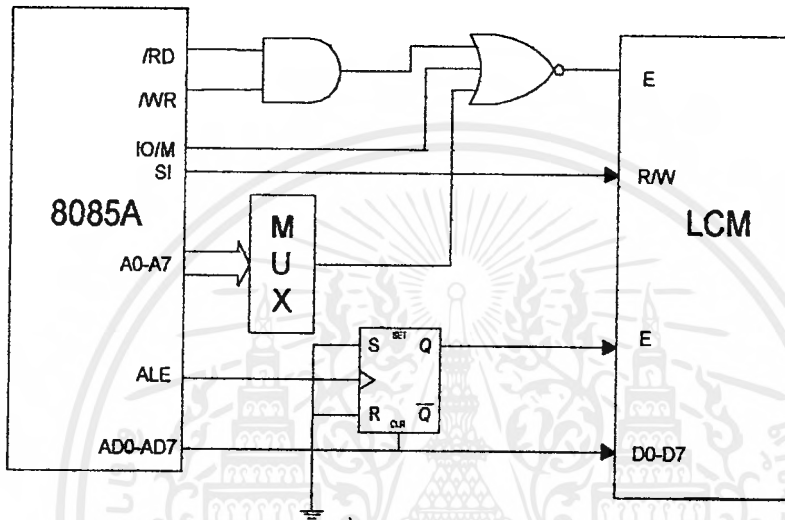
2. ผู้ใช้สามารถอ่านและเขียนข้อมูลจาก LCD MODULE ได้ ทำให้มองเสมือนว่าเป็น MEMORY BUFFER ไปในตัว

3. เนื่องจากอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้

4. ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น

5. ทำให้กินเนื้อที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการ DECODE ละเอียดพอสมควร

6. การจัดหาสัญญาณจะต้องเป็นไปตามแบบของ CHIP



รูปที่ 2.5 แสดงการต่อ LCD MODULE แบบ MEMORY MAP

2.3.1.2 การต่อแบบ I/O PORT

1. สามารถต่อเข้ากับ I/O PORT ใดๆ ก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับข้อกำหนดของ LCD MODULE

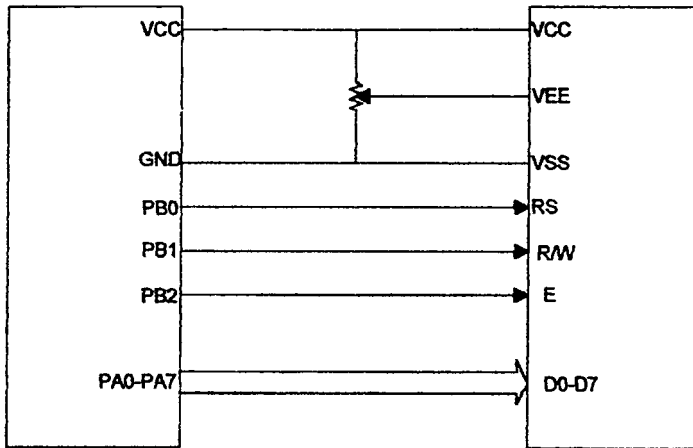
2. ผู้ใช้จะเขียนข้อมูลให้ LCD MODULE ได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนด MEMORY ส่วนหนึ่งให้เสมือน BUFFER ให้กับ LCD MODULE

3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเอง เพื่อรอให้ LCD MODULE กระทำขบวนการต่างๆ

4. ใช้ได้กับบอร์ดทุกๆ ไปที่มี PORT

5. ไม่เปลืองส่วนของ MEMORY ในการใช้งาน

6. การจัดหาสัญญาณกระทำได้อย่างอิสระ



รูปที่ 2.6 แสดงการต่อ LCD MODULE แบบ I/O PORT

2.3.2 จุดคำสั่งควบคุมและการแสดงข้อความ

ขั้วสัญญาณ VEE มีไว้สำหรับกำหนดความเข้มของตัวอักษร โดยถ้าต่อกับ GND จะมีความเข้มสูงสุด แต่ถ้าต่อกับ VCC จะมีความเข้มต่ำสุด ปกติ LCD รุ่นธรรมดาจะต่อกับ GND ไว้เลยก็ได้ ไม่ต้องใส่ VR ให้สั่นเปลือง แต่ถ้าเป็นรุ่น STN (มุมมองกว้าง) ให้ใช้ R 2K ต่อลง GND อีกที เพื่อให้ความเข้มมีความเหมาะสม การเขียนหรืออ่านข้อมูลกับ LCD MODULE ก็คือการกำหนดคุณสมบัติต่างๆ ในการใช้งานของ LCD ตามจุดคำสั่งควบคุม และรวมไปถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางต่อไปนี้

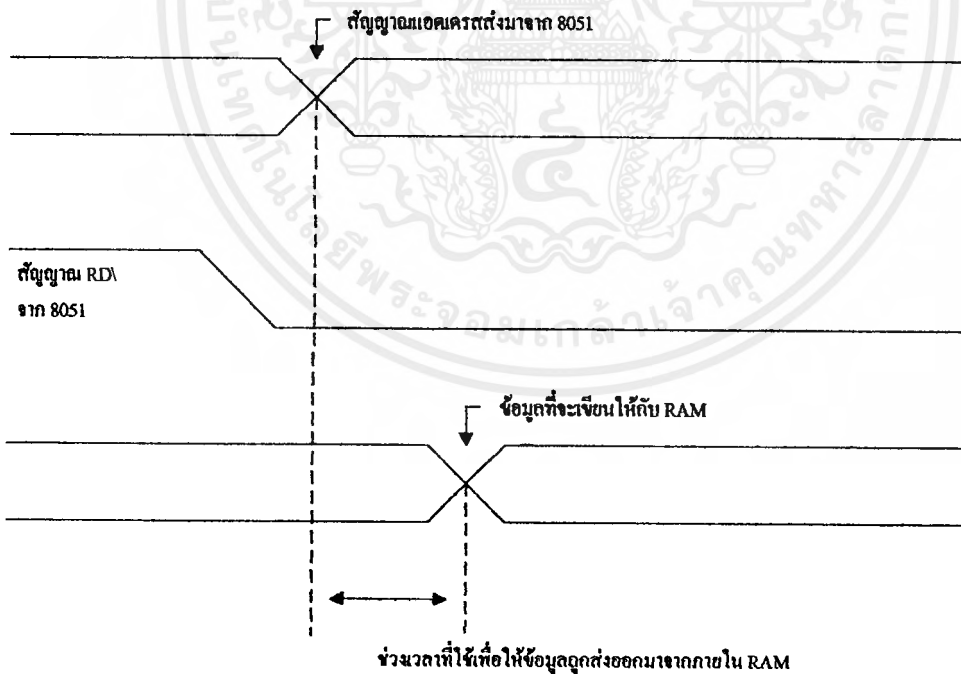
ตารางที่ 2.2 จุดคำสั่งควบคุมและการแสดงข้อความ

INSTRUCTION	RS	R/W	DATA BIT								EXE. TIME	
			7	6	5	4	3	2	1	0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B		40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*		40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*		40
SETCGRAM ADD.	0	0	0	1	CGRAM ADD.						40	
SET DDRAM ADD.	0	0	1	DDRAM ADD.						40		
BUSY,ADD. READ	0	1	BF	ADDRESS						40		
CGRAM,DDRAM.WR	1	0	WRITE DATA						40			
CGRAM,DDRAM RD	1	1	READ DATA						40			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเชื่อมต่อหน่วยความจำข้อมูลภายนอก (RAM)

การใช้หน่วยความจำข้อมูลภายนอกเป็นวิธีการแก้ปัญหาอย่างหนึ่ง ในกรณีที่มีความต้องการหน่วยความจำสำหรับการเก็บข้อมูลภายใน ซึ่งมีขนาดเพียง 128 หรือ 256 ไบต์เท่านั้น บางครั้งการใช้หน่วยความจำข้อมูลภายนอกยังเหมาะกับการประยุกต์บางอย่างที่จำเป็น ต้องมีการเก็บสำรองข้อมูลบางอย่างไว้ไม่ให้สูญหาย แม้ว่าจะไม่มีการจ่ายไฟฟ้าให้แก่ระบบ สามารถทำได้โดยการใช้ไอซีหน่วยความจำ RAM พร้อมแบตเตอรี่สำรองประเภทลิเทียมหรือนิเกิล-แคดเมียมเป็นตัวเก็บข้อมูลเหล่านี้ไว้แทน อย่างไรก็ตามไม่ว่าสาเหตุการนำไอซีหน่วยความจำภายนอกมาใช้งานจะเป็นอะไรจะมีผลทำให้พอร์ทอินพุต/เอาต์พุตข้อมูลของ 8051 ถูกนำไปใช้เพื่อติดต่อกับหน่วยความจำเหล่านี้แทน ดังนั้นอาจจำเป็นต้องมีการใช้วงจรประกอบอื่น ๆ เพื่อชดเชยความสามารถเหล่านั้นของ 8051 แทนกระบวนการนำข้อมูลเข้าไปจัดเก็บไว้ในหน่วยความจำ RAM แบบสแตติก จะเรียกว่า การเขียน (Write) ไปยังหน่วยความจำ และเมื่อมีการนำข้อมูลที่จัดเก็บไว้นั้นออกมา ก็จะเรียกว่า การอ่าน (Read) จากหน่วยความจำ ซึ่งขบวนการทั้งสองนี้จะต้องกระทำภายในช่วงรอบเวลาที่กำหนดไว้เท่านั้น สำหรับลักษณะสมบัติของหน่วยความจำ RAM มีความคล้ายคลึงกับหน่วยความจำ EPROM มาก รวมทั้งประเภทของสัญญาณติดต่อก็คือ บัสแอดเดรส บัสข้อมูล และกลุ่มสัญญาณควบคุมทั้งหมดเพียงแต่มีการเพิ่มสัญญาณ WR เพื่อระบุถึงการนำข้อมูลจากบัสข้อมูลลงเก็บ ไปยังหน่วยความจำตำแหน่งที่ได้ระบุแอดเดรสมาเท่านั้น

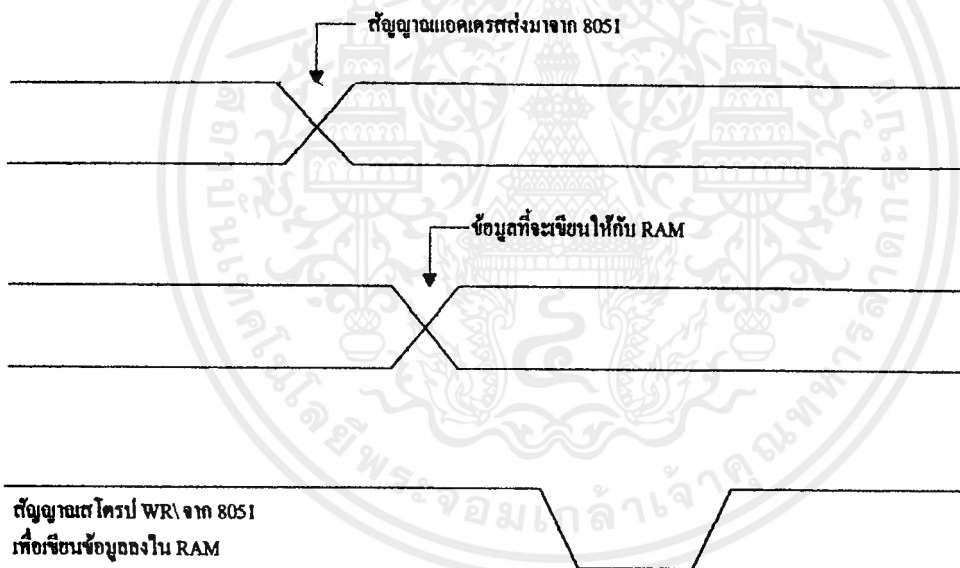


รูปที่ 2.7 ภาพเวลาแสดงลำดับเหตุการณ์การอ่านข้อมูลจากหน่วยความจำ RAM

จากแผนภาพเวลาของสัญญาณ แสดงให้เห็นถึงลำดับเหตุการณ์เมื่อมีการอ่านข้อมูลจากหน่วยความจำ RAM ดังนี้

1. ไมโครคอนโทรลเลอร์จะต้องเริ่มด้วยการส่งค่าแอดเดรสของหน่วยความจำที่ต้องการออกมาทางบัสแอดเดรส
2. สัญญาณ OE จะต้องถูกเปลี่ยนให้เป็นระดับลอจิกต่ำเพื่อระบุว่าต้องการอ่านข้อมูลจากหน่วยความจำ
3. หลังจากนั้น ไมโครคอนโทรลเลอร์จะหยุดรอในราวช่วงระยะเวลาหนึ่ง เรียกว่า Read Access Time เพื่อให้วงจรภายใน RAM ทำการถอดรหัสแอดเดรสและอ่านข้อมูล
4. หลังจากสิ้นสุดเวลาในข้อ 3 แล้ว ข้อมูลจะถูกส่งออกมาทางขาสัญญาณของบัสข้อมูลและไมโครคอนโทรลเลอร์สามารถอ่านข้อมูลนี้ไปประมวลผลต่อไป

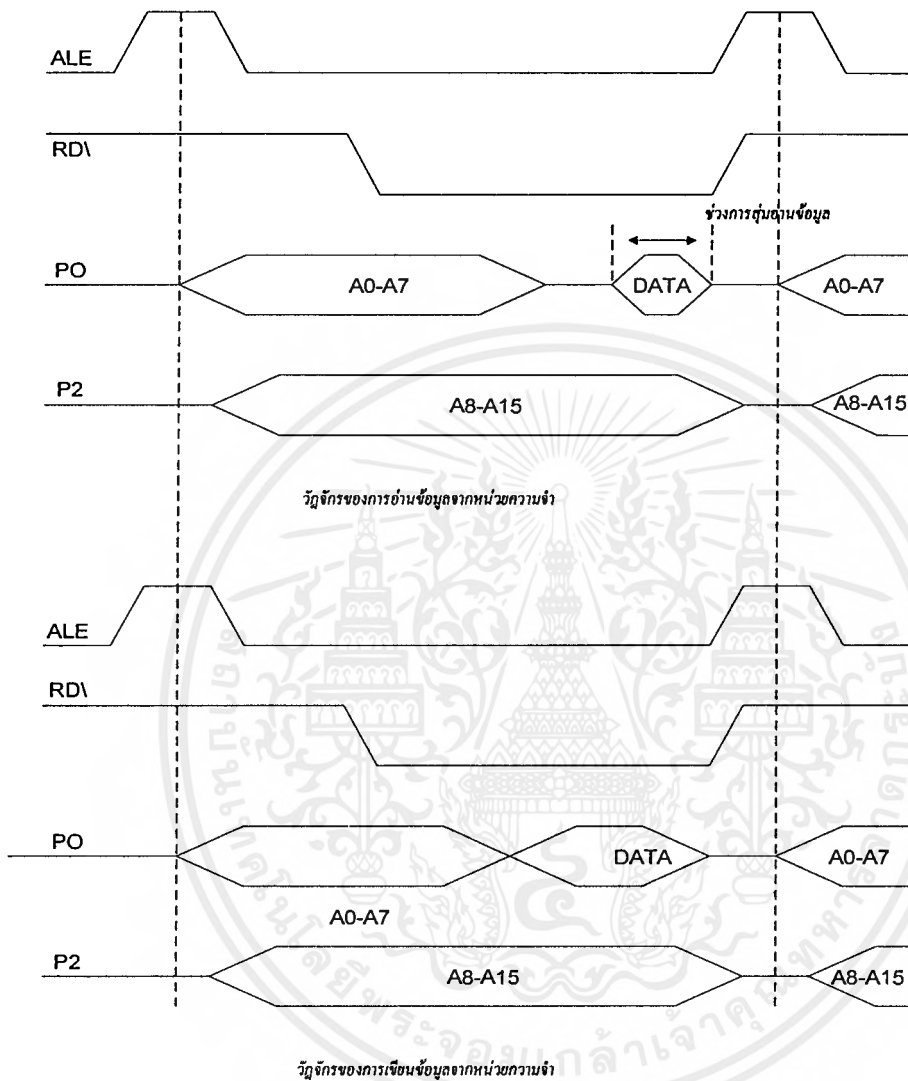
สำหรับแผนภาพเวลาของสัญญาณในรูปต่อไป นี้ แสดงให้เห็นถึงลำดับเหตุการณ์ เมื่อมีการเขียนข้อมูลในหน่วยความจำ RAM ดังนี้



รูปที่ 2.8 ภาพเวลาแสดงลำดับเหตุการณ์การเขียนข้อมูลลงในหน่วยความจำ RAM

1. ช่วงเริ่มต้น ไมโครคอนโทรลเลอร์จะต้องส่งค่าแอดเดรสของหน่วยความจำที่ต้องการออกมาทางบัสแอดเดรส
2. เวลาถัดไปในบัสข้อมูลจะเป็นข้อมูลที่ต้องการเขียนลงในหน่วยความจำ
3. ไมโครคอนโทรลเลอร์จะต้องทำการหยุดรอในช่วงระยะเวลาหนึ่งเช่นกัน เรียกว่า Write Access time เพื่อให้วงจรภายใน RAM ทำการถอดรหัสแอดเดรส และเตรียมการเขียนข้อมูล

4. เมื่อสิ้นสุดเวลาที่หยุดรอ ไมโครคอนโทรลเลอร์จะต้องขับสัญญาณ WR ให้เป็นระดับลอจิกต่ำหรือพัลส์ เพื่อให้ข้อมูลภายในบัสข้อมูลถูกนำไปยังตำแหน่งที่ต้องการภายในหน่วยความจำ RAM ต่อไป



รูปที่ 2.9 ภาพแสดงการติดต่อกับหน่วยความจำ RAM ของ 8051

ในรูปที่ 2.9 แสดงให้เห็นถึงแผนภาพเวลาของสัญญาณที่เกี่ยวข้องดังนี้ เมื่อเริ่มการติดต่อกับหน่วยความจำข้อมูลภายนอก จะเห็นว่ามีคำสั่งของแอดเดรสไบต์ต่ำ (A0-A7) ของตำแหน่งหน่วยความจำที่ต้องการออกมายังพอร์ต 0 ในราวก่อนช่วงเวลาขอบขาลงของสัญญาณ ALE ซึ่งจะนำไปใช้ในการควบคุมไอซีแลตซ์ เพื่อทำการค้างค่าแอดเดรสไบต์ต่ำนี้ไว้ ดังนั้นในเวลาต่อมาพอร์ต 0 ก็ว่าง และพร้อมที่จะนำไปใช้ในฐานะของบัสข้อมูล ส่วนค่าของแอดเดรสไบต์สูง (A8-A15) จะถูกส่งออกมาทางพอร์ต 2 ราวช่วงเวลาประมาณกึ่งกลางระหว่างสัญญาณ ALE เป็นระดับลอจิกสูง เมื่อ 8051 ทำการขับสัญญาณ RD หรือ WR ให้เป็นระดับลอจิกต่ำก็จะให้ผลทำให้เกิดการส่งและรับข้อมูลระหว่างหน่วยความจำ RAM กับบัสข้อมูลขึ้นได้

2.5 การรับส่งข้อมูลอนุกรม (UART) ของ 8051 และ 8052

พอร์ทสื่อสารอนุกรมของ 8051,8052 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางค่านส่งใช้ขา TxD (พอร์ท 3.1) ทางค่านรับใช้ขา RxD (พอร์ท 3.0) SBUF ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรม

พอร์ทสื่อสารอนุกรมของ 8051 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกบิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมดของพอร์ทสื่อสารอนุกรมนี้นี้

1. โหมด 0

พอร์ทสื่อสารอนุกรม 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต D0 ออกไปก่อนทางขา RxD และไม่มีการส่ง start bit แต่จะส่ง shift clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Clock)

2. โหมด 1

พอร์ทสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCIB และอัตราโอเวอร์โพล์ของ Timer

Baud Rate Mode 1 = $K * \text{Oscillator Freq.} / 32 * 12 * [256 - (TH1)]$ สำหรับ 8031,8032 โดยใช้ Timer1

Baud Rate Mode 1 = $\text{Oscillator Freq.} / 32 * [65536 - (RCAP2H, RCAP2L)]$ สำหรับ 8032,8052 โดยใช้ Timer 2

3. โหมด 2

พอร์ทสื่อสารอนุกรม 11 บิต ใช้ข้อมูล 9 บิต 1 start bit และ 1 stop bit (TB8 นิยมนำมาใช้ส่ง Parity bit) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU Clock โดยขึ้นกับบิต SMOD ใน PCON

Baud Rate Mode 2 = $1/32 \text{ Osc Freq.}$ เมื่อ SMOD = 1

Baud Rate Mode 2 = $1/64 \text{ Osc Freq.}$ เมื่อ SMOD = 0

4. โหมด 3

พอร์ทสื่อสารอนุกรมแบบ 11 bit UART โดยส่งข้อมูล 9 บิต 1 start และ 1 stop bit เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1 สำหรับ 8051 หรือขึ้นกับ อัตราโอเวอร์โพล์ของ Timer สำหรับ 80C154D

Baud Rate Mode 1 = $K * \text{Oscillator Freq.} / 32 * 12 * [256 - (TH1)]$ สำหรับ 8031,8032 โดยใช้ Timer1

Baud Rate Mode 1 = $\text{Oscillator Freq.} / 32 * [65536 - (RCAP2H, RCAP2L)]$ สำหรับ 8032,8052 ,80154,80154D โดยใช้ Timer 2

2.6 การสร้างฐานเวลาให้กับ MCS-51

การสร้างฐานเวลาให้แก่ไมโครคอนโทรลเลอร์มี 2 วิธีด้วยกันคือ

1. ใช้ชิปที่ทำหน้าที่สร้างฐานเวลาเพื่อส่งค่าปัจจุบันในขณะที่ใด ๆ ให้แก่ไมโครคอนโทรลเลอร์
2. ใช้วงจรกำเนิดความถี่ที่มีค่าคงที่ป้อนให้แก่เคาน์เตอร์ของไทม์เมอร์ 0 หรือ ไทม์เมอร์ 1 เพื่อให้ไทม์เมอร์ทั้งสองอินเตอร์รัปต์ซีทียูในช่วงเวลาตามที่กำหนดจากวงจรภายนอกได้

โดยการใช้งานไทม์เมอร์ หรือ เคาน์เตอร์นั้น ต้องคำนวณความเวลาที่ถูกต้อง และ ต้องมีการตั้งค่าใหม่ทุกครั้งเมื่อเริ่มทำงาน หากการทำงานเกิดผิดพลาดทางเวลาจะผิดพลาดไปด้วยเพราะใช้รีจิสเตอร์ภายใน MCS-51 เป็นตัวกำหนดฐานเวลา ดังนั้นในบางครั้ง การออกแบบระบบควบคุมจำเป็นต้องมีวงจรทำหน้าที่สร้างฐานเวลาหรือวงจรสร้างสัญญาณความถี่ที่คงที่ เพื่อเป็นอินพุตให้แก่เคาน์เตอร์ของไทม์เมอร์ใน MCS-51 โดยการสร้างฐานเวลาให้แก่ MCS-51 ทั้งสองวิธีต้องมีการทำงานที่เป็นอิสระแยกจากไมโครคอนโทรลเลอร์ตลอดเวลา จากสองวิธีที่กล่าวมาแล้ว คือ ในส่วนของวิธีแรกจะใช้ชิปทำหน้าที่สร้างฐานเวลาจริงให้ระบบ (RTC หรือ Real Time Clock) ข้อดีที่ได้คือ จะมีไทม์เมอร์และเคาน์เตอร์เหลือไว้ใช้งานอย่างอื่นอีกด้วย วิธีที่สอง จะใช้วงจรสร้างสัญญาณความถี่ที่คงที่ เพื่อเป็นอินพุตให้แก่เคาน์เตอร์ใน MCS-51 โดยต้องใช้งานที่มีความเที่ยงตรงสูง

วิธีการสร้างฐานเวลาจริงให้แก่ MCS-51 (ใช้ชิป RTC)

วิธีการที่จะสร้างฐานเวลานั้นจะใช้ชิป RTC ซึ่งใช้เบอร์ DS1202 “Serial Timekeeper Chip” ซึ่งชิปเบอร์นี้ทำได้แค่ให้ข้อมูลที่เป็นเวลาขณะใดให้แก่ไมโครคอนโทรลเลอร์เท่านั้น แต่มีการทำงานที่เที่ยงตรงสูงมาก สามารถนำมาต่อร่วมกับระบบเพื่อบอกเวลาให้แก่ไมโครคอนโทรลเลอร์ได้สะดวก เพราะใช้จำนวนสายในการติดต่อระหว่างชิปกับไมโครคอนโทรลเลอร์เพียง 3 เส้นเท่านั้น โดยชิปตัวนี้ใช้การติดต่อข้อมูลแบบอนุกรม

คุณสมบัติของชิป RTC มีดังนี้

1. ทำหน้าที่นับ วินาที นาที ชั่วโมง วันที่ของเดือน เดือน ปี รวมทั้งคำนวณปีอธิกสุรทินให้เองโดยอัตโนมัติ
2. มีหน่วยความจำขนาด 24 ไบต์ สำหรับเก็บข้อมูลทั่วไป
3. ใช้การติดต่อแบบอนุกรม จึงใช้จำนวนสายในการเชื่อมต่อเพียง 3 เส้น
4. ให้แรงดันไฟฟ้าเพียง 2.0 ถึง 5.5 โวลต์ และใช้กระแสเพียง 300 นาโนแอมแปร์ ที่ระดับแรงดัน 2.0 โวลต์
5. การโอนย้ายข้อมูลสามารถทำได้แบบครั้งละ 1 ไบต์ (Single Byte) หรือ ลั่งละหลาย ๆ ไบต์ (Burst Mode)
6. ระดับสัญญาณ TTLcompatible (VCC = 5 โวลต์)
7. ช่วงอุณหภูมิการใช้งานกว้างมาก ระหว่าง -40องศา+ และ - 88 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของชิป RTC

ชิป RTC เบอร์ DS1202 “Serial Time Keeper Chip” มี Real Time Clock/Calendar และ Static RAM ขนาด 24 ไบต์ ใช้สายเพียง 3 เส้นในการเชื่อมต่อกับ ไมโครคอนโทรลเลอร์เพื่อรับส่งข้อมูลเกี่ยวกับเวลา ข้อมูลที่ชิป RTC DS1202 มีให้ประกอบด้วย

- วินาที
- นาฬิกา
- ชั่วโมง
- วันที่
- เดือน
- ปี

วันที่วันสุดท้ายของเดือน จะถูกปรับโดยอัตโนมัติสำหรับเดือนที่มีจำนวนวันน้อยกว่า 31 วัน และมีการคำนวณจำนวนวันของเดือนกุมภาพันธ์ในปีอธิกสุรทินให้เอง ข้อมูลที่ส่งให้แก่ไมโครคอนโทรลเลอร์สามารถเลือกรูปแบบได้ทั้งแบบ 24 ชั่วโมง (0.00-23.59 นาฬิกา) หรือแบบ 12 ชั่วโมง (0.00-12.00 นาฬิกา) โดยมีข้อมูลเพิ่มเพื่อบอกให้ทราบว่าเป็นเวลาในช่วงกลางวันหรือกลางคืน)

การเชื่อมต่อชิป RTC DS1202 มีนาฬิกาหรือเวลาที่เดินอยู่ตลอดเวลา รวมทั้งมีหน่วยความจำจำนวนหนึ่ง ดังนั้นในการติดต่อกับชิป RTC DS1202 ผู้ใช้สามารถเลือกได้ว่าต้องการข้อมูลจากนาฬิกาหรือจากหน่วยความจำภายในชิป (CLOCK/RAM) การรับส่งข้อมูลสามารถกระทำได้ทั้งแบบทีละไบต์หรือรับส่งกัน คราวละหลายไบต์ นอกจากนี้ RTC DS1202 ยังถูกออกแบบให้ใช้พลังงานน้อยมาก และสิ้นเปลืองพลังงานจากแบตเตอรี่น้อยที่สุดเพื่อความสะดวกในการสำรองพลังงาน โดยชิปตัวนี้สามารถเก็บรักษาข้อมูลในหน่วยความจำและเวลาที่เดินอยู่ตลอดเวลาได้ที่กำลังไฟที่น้อยกว่า 1 ไมโครวัตต์

ในการรับหรือส่งข้อมูลใด ๆ ให้แก่ RTC DS1202 ไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ที่เพียงการติดต่อกับจะต้องส่งข้อมูลที่เป็นคำสั่งควบคุมการติดต่อซึ่งมีขนาด 8 บิตเสียก่อน โดยเริ่มต้นด้วยการให้ขา RST มีสถานะเป็น 1 (อยู่ในช่วงการติดต่อ) จากนั้นส่งข้อมูลจำนวน 8 บิตเข้าไปใน shift register ของ RTC ข้อมูลขนาด 8 บิต จะประกอบด้วยคำสั่งในการควบคุมชิป RTC และตำแหน่งของหน่วยความจำที่ต้องการติดต่อ (address/command byte) ในแต่ละครั้ง การรับข้อมูลจากไมโครโปรเซสเซอร์แต่ละบิตจะกระทำที่ช่วงขอบขาขึ้นของสัญญาณที่ขา SCLK (serial clock)

ภายในชิป RTC เบอร์ DS1202 ประกอบด้วยหน่วยความจำขนาด 24 ไบต์ และรีจิสเตอร์ที่ทำหน้าที่เก็บเวลาของชิปในขณะปัจจุบันจำนวน 8 ตัว รีจิสเตอร์ทั้ง 8 ตัวนี้สามารถเข้าถึงได้เสมือนเป็นหน่วยความจำตำแหน่งหนึ่ง ดังนั้นต่อไปจะมองว่าชิป RTC เบอร์นี้มีหน่วยความจำรวมทั้งสิ้น 32 ตำแหน่ง โดยประกอบขึ้นจากรีจิสเตอร์ 8 ตำแหน่ง และหน่วยความจำ 24 ตำแหน่ง ข้อมูลขนาด 8 บิตแรก (address/command byte) จะระบุตำแหน่งหน่วยความจำที่ต้องการติดต่อ (ทั้งตำแหน่งของหน่วยความจำทั่วไปและตำแหน่งของรีจิสเตอร์สำหรับเก็บเวลา) และบอกว่าเป็นการเขียนหรืออ่านข้อมูลจากหน่วยความจำตำแหน่งนั้น ๆ รวมทั้งระบุว่าการรับส่งข้อมูลเป็นแบบครึ่งบิต 1 ไบต์ หรือครึ่งละ หลาย ๆ

ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

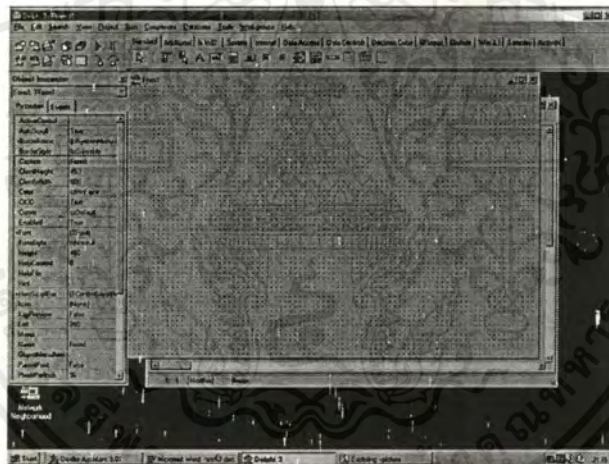
หลังจากมีสัญญาณนาฬิกาเกิดขึ้น 8 ครั้ง (สัญญาณ SCLK) ในระหว่างการเขียนข้อมูล 8 บิตแรก เข้าไปใน shift register สัญญาณนาฬิกาต่อไปที่จะเกิดขึ้นจะเป็นการนำข้อมูลออกจากชิป RTC สำหรับการอ่านข้อมูลหรือนำข้อมูลเข้าไปยังชิป RTC สำหรับการเขียนข้อมูล จำนวนของสัญญาณนาฬิกาทั้งหมดที่เกิดขึ้นในการติดต่อครั้งหนึ่ง ๆ จึงเท่ากับ 8×8 ในการส่งข้อมูลครั้งละ 1 ไบต์ single byte mode หรือ 8 บวกรวมที่สุด $192 (8 \times 24)$ สำหรับการส่งข้อมูลครั้งละหลาย ๆ ไบต์

2.7 การเขียนโปรแกรมด้วยภาษาเคลไฟ

2.7.1 หน้าจอของเคลไฟ

หน้าจอของเคลไฟแบ่งออกได้เป็น 4 ส่วนใหญ่ๆ ดังรูปที่ 2.10คือ

1. หน้าต่างหลัก
2. หน้าต่างฟอร์ม (Form)
3. หน้าต่างยูนิท (Unit)
4. หน้าต่างออบเจกต์อินสเปกเตอร์ (Object Inspector)



รูปที่ 2.10 หน้าจอของเคลไฟโดยรวม

1. หน้าต่างหลัก

จากรูปที่ 2.11 หน้าต่างหลักประกอบด้วย

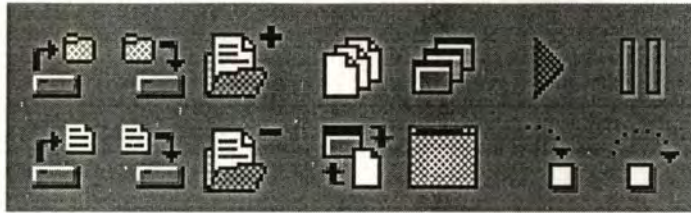
1. ไตเติลบาร์ (Title bar) จะแสดงรายชื่อของชิ้นงาน (Application) แต่ละชิ้นงาน
2. เมนูบาร์ (Menu bar) แสดงรายการตั้งแต่ File จนถึง Help
3. ก่อร่างคอมโปเนนต์ (Component) จะเป็นที่เก็บคอมโปเนนต์ต่างๆ ที่จะไปวางไว้บนฟอร์ม
4. สปีดบาร์ (Speed Bar) คือ ส่วนของปุ่มที่ใช้ในการเรียกการทำงานดังรูปที่ 2.12 ประกอบไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.1 Open Project คือ ปุ่มที่ใช้เปิดโปรเจกต์ใหม่ที่สร้างไว้แล้ว
- 4.2 Save File คือ ปุ่มที่ใช้เปิดไฟล์ใหม่ที่สร้างไว้แล้ว
- 4.3 Save All คือ ปุ่มที่ใช้เซฟโปรเจกต์ของงานที่กำลังทำอยู่
- 4.4 Save File คือ ปุ่มที่ใช้เซฟเฉพาะบางไฟล์
- 4.5 Add File To Project คือ ปุ่มที่ใช้ในการนำยูนิท (ไฟล์นามสกุล PAS) ของโปรเจกต์อื่นเข้ามาทำงานร่วมกับโปรเจกต์ที่เรากำลังทำงานอยู่
- 4.6 Remove File To Project คือ ปุ่มที่ใช้ในการนำยูนิทของโปรเจกต์ที่กำลังใช้งานอยู่ออกไป
- 4.7 Select Unit From List คือ ปุ่มที่ใช้เลือกยูนิทที่มีอยู่แล้วใน โปรเจกต์ขึ้นมาใช้งาน
- 4.8 Toggle From/Unit คือ ปุ่มที่ใช้ในการสลับกันทำงานระหว่างหน้าต่างฟอร์มกับหน้าต่างยูนิท
- 4.9 Select Form From List คือ ปุ่มที่ใช้เลือกฟอร์มที่มีอยู่แล้วใน โปรเจกต์ขึ้นมาใช้งาน
- 4.10 New Form คือ ปุ่มที่ใช้เรียกฟอร์มใหม่ขึ้นมาใช้งานร่วมกับโปรเจกต์ที่กำลังใช้งานอยู่
- 4.11 Run คือ ปุ่มที่ทำให้ตัว Project compile เริ่มทำงาน
- 4.12 Trace into คือปุ่มที่ใช้แสดงผลการทำงานของโปรเจกต์ทีละขั้นตอน
- 4.13 Pause คือ ปุ่มที่ทำให้การทำงานของโปรเจกต์หยุดชั่วคราว
- 4.14 Step Over คือ ปุ่มที่ใช้ในการข้ามการแสดงผลบางช่วงไป



รูปที่ 2.11 หน้าต่างหลัก



4.1	4.3	4.5	4.7	4.9	4.11	4.13
4.2	4.4	4.6	4.8	4.10	4.12	4.14

รูปที่ 2.12 สปีดบาร์ (Speed bar)

2. หน้าต่างฟอร์ม

หน้าต่างฟอร์มแสดงดังรูปที่ 2.13 เป็นหน้าต่างที่มีไว้สำหรับแสดงผลในการทำงานของตัวโปรแกรม เวลาใช้ให้นำคอมพิวเตอร์ไปเน็ตที่ต้องการใช้ลงไปวาง แล้วกำหนด Properties กับ Events ในออบเจกต์ อินสเปคเตอร์



รูปที่ 2.13 หน้าต่างฟอร์ม (Form)

3. หน้าต่างยูนิต

หน้าต่างยูนิตแสดงดังรูปที่ 2.14 เป็นหน้าต่างที่มีไว้สำหรับใส่เงื่อนไขในการทำงานของโปรแกรม โดยผ่านทาง Events ของออบเจกต์อินสเปคเตอร์ หรือจะใส่เงื่อนไขโดยตรงก็ได้

```

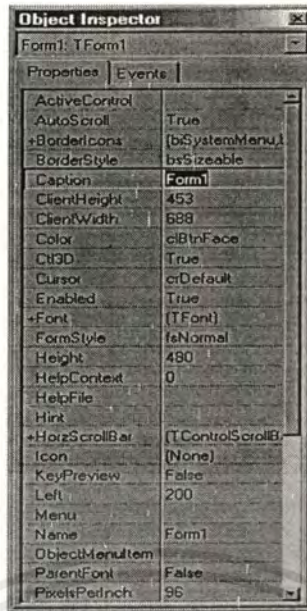
Unit1.pas
Unit
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
type
  TForm1 = class(TForm)
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
  ($R *.DFM)
  
```

รูปที่ 2.14 หน้าต่างยูนิท (Unit)

4. หน้าต่างออบเจกต์อินสเปกเตอร์

หน้าต่างออบเจกต์อินสเปกเตอร์ดังแสดงในรูปที่ 2.15 จะประกอบด้วย

1. Properties จะเป็นตัวกำหนดการทำงานของคอมโปเนนต์ต่างๆ
2. Events จะเป็นตัวผ่านจากคอมโปเนนต์ไปกำหนดเงื่อนไขต่างๆ ในยูนิท เช่น
 - OnClick จะทำงานก็ต่อเมื่อเราเลือกด้วยเมาส์หรือด้วยคีย์บอร์ด
 - OnMouseDown จะทำงานก็ต่อเมื่อเราคลิกเมาส์ลง
 - OnMouseUp จะทำงานก็ต่อเมื่อเราคลิกเมาส์ขึ้น
 - OnKeyDown จะทำงานโดยการกดคีย์ลงจะไปทำงานที่ key code ที่ กำหนดไว้ในเงื่อนไขของ Events นั้นๆ ในโปรแกรม



รูปที่ 2.15 หน้าต่าง Object Inspector

2.7.2 เริ่มต้นใช้งาน

การเริ่มต้นใช้งานจริงนี้จะเริ่มอธิบายตั้งแต่การเปิดใช้งานโปรแกรมเดลไฟ, การเปิดโปรเจกต์ใหม่, การวางคอมโปเนนต์, การกำหนดพร็อพเพอร์ตี้, การกำหนดอีเวนต์, การกำหนดเงื่อนไข, การเซฟไฟล์, การคอมไพล์, การเปิดโปรเจกต์เก่าขึ้นมาแก้ไข และการนำโปรเจกต์ไปใช้งานจริงหลังจากคอมไพล์แล้ว

1. การเปิดโปรแกรมเดลไฟขึ้นมาใช้งาน

เลือกไปที่ Start แล้วเลือกไปที่ Programs / Borland Delphi 3 ก็จะแสดงหน้าจอของโปรแกรมเดลไฟ 3 ออกมา

2. การเปิดโปรเจกต์ใหม่และการใช้งาน

1. ไปที่เมนูบาร์ เลือก File / New Application

2. กำหนดชื่อ Project ใหม่ให้เหมาะสมกับการใช้งาน

2.1 ไปที่เมนูบาร์ เลือก File / Save Project As หรือจะใช้เมาส์ไปคลิกที่ปุ่ม Save All บนสปีดบาร์

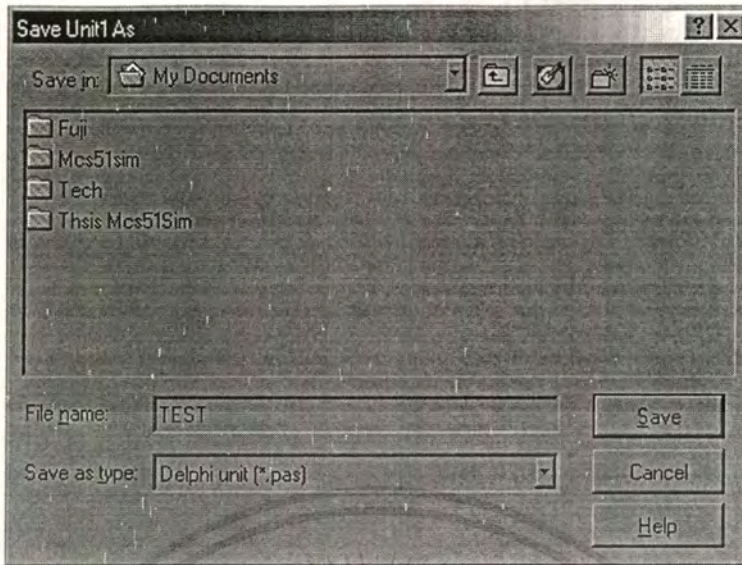
2.2 จะมีหน้าต่าง Save Unit As แสดงขึ้นมาดังรูปที่ 2.16 เป็นหน้าต่างที่มีไว้ให้เซฟไฟล์ยูนิท ให้พิมพ์ชื่อไฟล์ยูนิทที่เราต้องการลงในช่อง File name

2.3 คลิกไปที่ Save

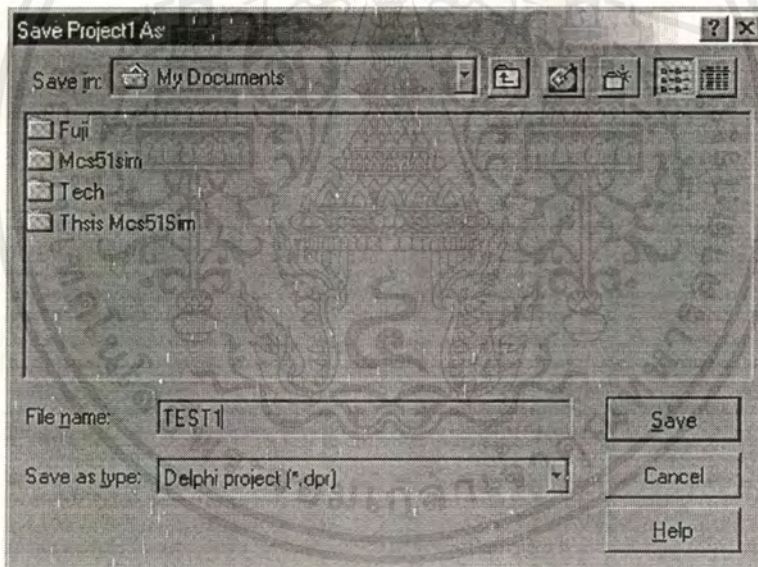
2.4 จะมีหน้าต่าง Save Project As แสดงขึ้นมา เป็นหน้าต่างที่มีไว้ให้เซฟไฟล์โปรเจกต์ ให้พิมพ์ชื่อไฟล์โปรเจกต์ที่เราต้องการลงในช่อง File name

2.5 คลิกไปที่ Save

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 หน้าต่าง Save Unit As



รูปที่ 2.17 หน้าต่าง Save Project As

2.7.3 ตัวอย่างการเริ่มต้นใช้งาน

ตัวอย่างการกำหนดชื่อโปรเจกต์ใหม่

1. PROGRAM PNEW {TEST1.dpr}
2. Unit PNEW1 {TEST.pas}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายตัวอย่างการกำหนดชื่อโปรเจกต์ใหม่

บรรทัดที่ 1 คือการตั้งชื่อโปรเจกต์ใหม่ โดยให้ชื่อว่า TEST1 แล้วเซฟเป็นสกุล dpr

บรรทัดที่ 2 คือการตั้งชื่อยูนิคใหม่ โดยให้ชื่อว่า TEST แล้วเซฟเป็นสกุล pas ทั้งสองบรรทัดข้างบนนี้จะทำได้ โดยการเลือกไปที่สปีคบาร์แล้วเลือกที่ Save All หรือเลือกที่เมนู File / Save All จะแสดงหน้าต่าง Save Unit1 As ดังในรูปที่ 2.16 ให้พิมพ์ TEST ลงในช่อง File name แล้วกด Save หลังจากนั้นจะมีหน้าต่าง Save Project1 As แสดงขึ้นมา ดังในรูปที่ 2.17 ให้พิมพ์ TEST1 ลงในช่อง File name แล้วกด Save

3. จัดการกำหนดคอมโปเนนต์ที่ต้องการ

ตัวอย่างการวางคอมโปเนนต์

Components

1. Form == Form1
 - Caption = Pnew
2. Standard :: Button == Button1
 - Caption = OK
3. Standard :: Label == Label1
 - Caption = Hi

อธิบายตัวอย่างการวางคอมโปเนนต์

ข้อ 1 เป็นการให้ Caption ของ Form1 เป็น Pnew โดยผ่านทาง Properties

ข้อ 2 เป็นการนำคอมโปเนนต์ Button จากกล่องคอมโปเนนต์ Standard มาวางบน Form1 จะได้ Button1 แล้วให้ค่า Properties ของ Button1 ให้ Caption เป็น OK

ข้อ 3 เป็นการนำคอมโปเนนต์ Label จากกล่องคอมโปเนนต์ Standard มาวางบน Form1 จะได้ Label1 แล้วให้ค่า Properties ของ Label1 ให้ Caption เป็น Hi

4. เมื่อกำหนดคอมโปเนนต์ตามตัวอย่างได้แล้วก็จะได้ฟอร์มออกมาจัดการใส่เงื่อนไข โดยผ่าน

Events

ตัวอย่างการใส่เงื่อนไข โดยผ่าน Events

Events OnClick ของ Button1

- 1: procedure TForm1.Button1Click(Sender: TObject)
- 2: begin
- 3: Label1.Caption := Hello;
- 4: end;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายตัวอย่างการใส่เงื่อนไขโดยผ่าน Events

การที่เราจะเข้าถึง Events ตามในตัวอย่างนี้ได้ เราต้องเข้าไปกำหนด Events (Object) ที่ชื่อว่า Button1 โดยทำดังต่อไปนี้

1. ใช้เมาส์เลือกไปที่วัตถุ Button1
2. ใช้เมาส์เลือกไปที่ Events ในออบเจกต์อินสเปคเตอร์
3. ใช้เมาส์ดับเบิลคลิกไปที่ OnClick
4. และหน้าต่างยูนิคก็จะแสดงขึ้นมา
5. แล้วใส่เงื่อนไขตามบรรทัดที่ 3 ในตัวอย่างลงไป (Label.Caption := Hello;)

5. ทำการคอมไพล์ (COMPILE) ให้อยู่ในสกุล exe

5.1 ขั้นตอนในการคอมไพล์มี 2 วิธี คือ

5.1.1 เลือกไปที่เมนู Project / Compile หรือกด Ctrl+F9

5.1.2 เลือกไปที่เมนู Run / Run หรือกด F9 แต่วิธีที่ง่ายที่สุด คือใช้เมาส์เลือกไปที่ปุ่ม Run

5.2 เมื่อเข้าใจขั้นตอนการคอมไพล์แล้วก็ทดลองนำโปรแกรม PNEW มาทำดู จะได้ผล

5.3 เมื่อต้องการจะปิดผลการรัน ก็ให้ใช้เมาส์เลือกไปที่ปุ่ม X ตรงมุมขวาบนของหน้าต่างที่กำลังรันอยู่

6. ทดลองเปิดโปรเจกต์ PNEW แล้วเปิดดูใหม่โดย

6.1 เลือกไปที่เมนู File / Open หรือเลือกไปที่ปุ่ม Open Project

6.2 ใช้เมาส์เลือกไปที่ชื่อโปรแกรมที่ต้องการจะเปิด

6.3 แล้วเลือกปุ่ม Open โปรแกรมก็จะเปิดออกมาพร้อมที่จะรับการแก้ไข

7. การนำโปรแกรมที่คอมไพล์แล้วมาทำเป็น SHORTCUT เพื่อที่จะใช้งาน ได้โดยมีขั้นตอนดังต่อไปนี้

7.1 ใช้เมาส์คลิกขวา แล้วเลือกไปที่ New / Shortcut ก็จะแสดงหน้าต่าง Create Shortcut ขึ้นมา

7.2 ให้เลือกไปที่ปุ่ม Browse ไฟล์ที่เราต้องการที่จะทำเป็น Shortcut

7.3 เมื่อเลือกได้แล้วให้ไปเลือกที่ปุ่ม Open จะกลับมาที่หน้าต่าง Create Shortcut

7.4 เลือกไปที่ Next และ Finish ก็จะได้ Shortcut

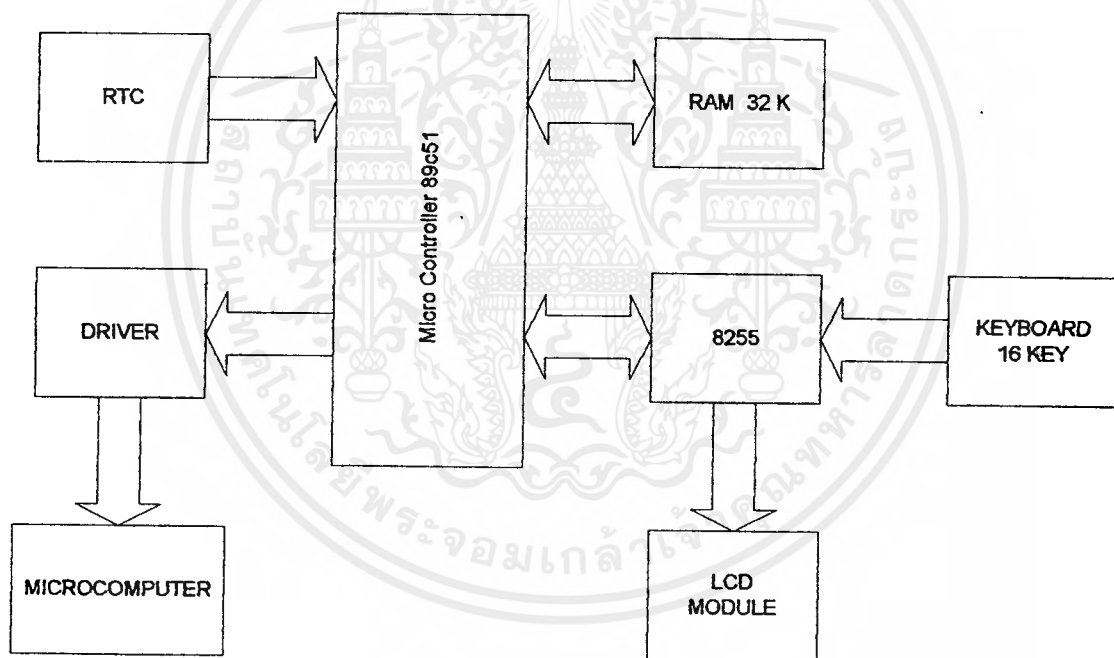
บทที่ 3

การคำนวณและการสร้าง

โครงการนี้เป็น การสร้างเครื่องทบทวนนิคทททททท ซึ่งใช้ในการเก็บข้อมูลแบบตัวเลขในลักษณะต่างๆ เช่น ในการเก็บค่าไฟฟ้าหรือน้ำประปานั้นเมื่อก่อนอาศัยการจดบันทึกลงบนกระดาษ หลังจากนั้นจึงนำข้อมูลที่ได้อ่านไปให้พนักงานที่มีหน้าที่คีย์ข้อมูลลงคอมพิวเตอร์ ซึ่งวิธีนี้มีข้อเสียคือ เสียเวลาและมีขั้นตอนในการทำงานซับซ้อนกัน จึงได้มีการพัฒนาเครื่องบันทึกข้อมูลชนิดททททททขึ้น โดยข้อดีของเครื่องนี้คือ เป็นการคีย์ข้อมูลเข้าไปในเรื่องครั้งเดียวนั่นคือ เวลาในช่วงการไปจดบันทึกค่าของไฟฟ้าหรือประปาตามบ้านแล้วบันทึกเข้าไปในเครื่อง และเมื่อได้ข้อมูลแล้วนำข้อมูลไปเก็บรวบรวมไว้ในฐานข้อมูลเพื่อนำไป คำนวณหาค่าไฟฟ้าและประปา

3.1 Block Diagram

โดยโครงสร้างของเครื่องอธิบายตาม Block Diagram ดังนี้



รูปที่ 3.1 แสดง Block Diagram ของเครื่องบันทึกข้อมูล

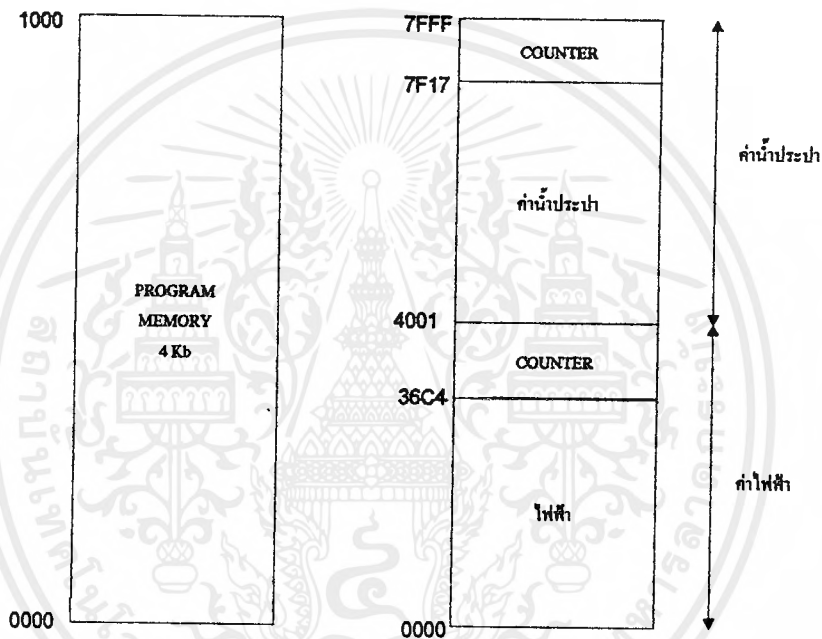
1. การรับข้อมูล คือรหัสตัวตัวเลขจะถูกป้อนเข้าคีย์บอร์ด 8255 รับมาแล้วส่งไปยัง Microcontroller ไปประมวลผลแล้วเก็บไว้ในแรม ขนาด 32 Kbyte และ LCD จะเป็นตัวแสดงผลการรับข้อมูล
2. ในการรับข้อมูลทุกครั้งจะใช้ RTC เป็นตัวสร้างฐานเวลา ซึ่งเป็นประโยชน์ในการตัดยอดค่าไปไฟฟ้าหรือค่าประปาประจำเดือน

8. การเคลื่อนย้ายข้อมูลจะมีสายส่งที่ต่อเข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ เพื่อใช้ในการเคลื่อนย้ายข้อมูลสู่คอมพิวเตอร์ โดยใช้มาตรฐานการส่ง RS-232

3.2 การคำนวณ Address

ข้อมูลรหัสตัวเลขที่เก็บไว้นั้นจะมี 3 แบบ

1. เป็นหมายเลขของมิเตอร์ไฟฟ้าหรือประปา
2. เป็นจำนวนการใช้ไฟฟ้าหรือประปา
3. วัน/เดือน/ปี



รูปที่ 3.2 แสดงการจัด MEMORY ของเครื่องบันทึกข้อมูล

สามารถคำนวณการใช้หน่วยความจำใน DATA MEMORY ได้ดังนี้

$$\text{RAM } 32 \text{ K} = 32 * 1024 = 32,768 \text{ Byte}$$

แบ่งเป็น 2 ส่วน คือ ค่าไฟฟ้า และ ค่าน้ำประปา

$$32,768 / 2 = 16,384 \text{ ไบต์}$$

ค่าไฟฟ้า

- 1 เรคอร์ด ประกอบด้วย
1. วัน/เดือน/ปี = 6 ไบต์
 2. หมายเลขมิเตอร์ = 6 ไบต์
 3. จำนวนการใช้ = 4 ไบต์
 4. ไบต์แรกของเรคคอร์ด = 1 ไบต์
 5. ไบต์สุดท้ายของเรคคอร์ด = 1 ไบต์

1 เรคอร์ด = 18 ไบต์

บันทึกจำนวนมิเตอร์ของไฟฟ้าได้ = $3624H/16D = 850$ มิเตอร์

ค่าน้ำประปา

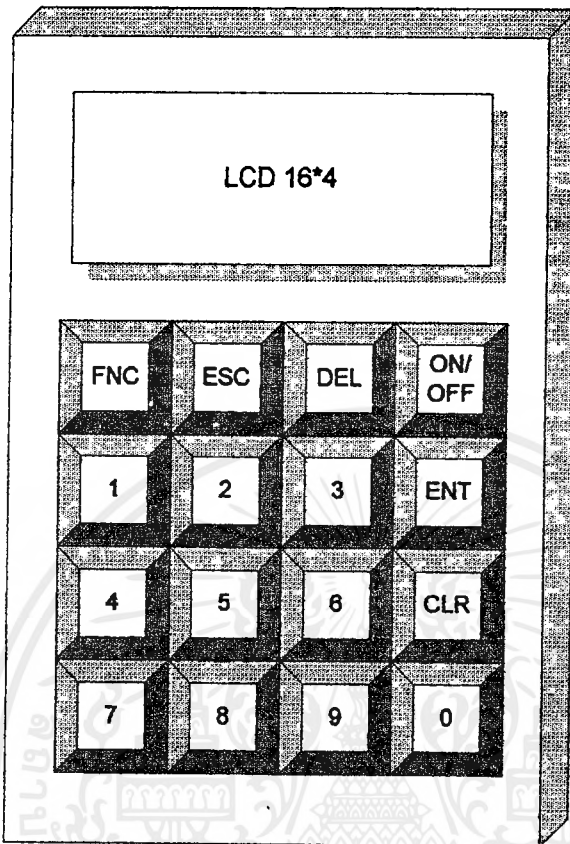
- 1 เรคอร์ด ประกอบด้วย
1. วัน/เดือน/ปี = 6 ไบต์
 2. หมายเลขมิเตอร์ = 7 ไบต์
 3. จำนวนการใช้ = 4 ไบต์
 4. ไบต์แรกของเรคคอร์ด = 1 ไบต์
 5. ไบต์สุดท้ายของเรคคอร์ด = 1 ไบต์

1 เรคอร์ด = 19 ไบต์

ให้แอดเรสเริ่มต้นของน้ำประปาที่ 4001H สิ้นสุดที่ 7FFFH

เพราะฉะนั้นบันทึกจำนวนมิเตอร์ของน้ำประปาได้ = $(7FFFH-4001H)/19D = 850$ มิเตอร์

3.3 ลักษณะของตัวเครื่องบันทึกข้อมูล



รูปที่ 3.3 ลักษณะของตัวเครื่องบันทึกข้อมูล

หน้าที่ของปุ่มต่างๆ ในเครื่องบันทึกข้อมูล

1. จอ LCD ขนาด 16 คีย์อักษร 4 บรรทัด
2. Function มี 3 Fn คือ -1. ELECTRICAL POWER บันทึกค่าไฟฟ้า
-2. WATER บันทึกค่าน้ำ
-3. SEND สั่งให้ส่งข้อมูล
3. DEL ใช้ในการลบข้อมูลที่ผิด
4. ON/OFF ใช้ในการเปิดปิดเครื่อง
5. ENT ยืนยันการส่ง
6. CLR ใช้เคลียร์ข้อมูลทั้งหมด เมื่อต้องการจะเก็บครั้งต่อไป
7. ปุ่มตัวเลข 0-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลองของส่วนเครื่องบันทึกข้อมูลภาคทฤษฎี

เมื่อทำการรันโปรแกรม หน้าจอ LCD จะแสดงข้อความดังต่อไปนี้

1. หน้าจอแสดงข้อความเริ่มต้นการทำงาน โดยมีชื่อเครื่อง,ผู้จัดทำ (LOGO)

PROJECT
3T/2
PORTABLE
DATA STORAGE

2. เครื่องจะแสดงวัน/เดือน/ปี ให้โดยอัตโนมัติ และ ตามความถูกต้อง

DD/MM/YY
11/10/41
[ENT] CORRECT
[CLR] INCORRECT

-ถ้าถูกต้องให้กดปุ่ม ENT

-ถ้าไม่ถูกต้องให้กดปุ่ม CLR เพื่อ SET วัน/เดือน/ปี ที่ถูกต้อง

กรณีกดปุ่ม CLR จะปรากฏหน้าจอดังต่อไปนี้

SET DD/MM/YY
DAY
MONTH
YEAR

กรณีกดปุ่ม ENT จะปรากฏหน้าจอ MAIN MENU เพื่อเลือก FUNCTION การใช้งานดังต่อไปนี้

3. หน้าจอ MAIN MENU

```

SELECT FUNCTION
1. ELECTRICAL
2. WATER
3. SEND

```

ให้กดปุ่ม FUNCTION 1 หรือ 2 หรือ 3 จะปรากฏหน้าจอดังต่อไปนี้

```

FUNCTION 1 หรือ 2 หรือ 3

```

4. กรณีเลือก FN1 หน้าจอจะปรากฏดังต่อไปนี้

```

*****
ELECTRICAL
SYSTEM
*****

```

รอประมาณ 3 วินาที หน้าจอจะแสดงข้อความดังต่อไปนี้

```

ELECTRICAL
[ENT] ENT
[CLR] CLEAR
[ESC] MAIN MENU

```

กดปุ่ม ENT เพื่อทำการป้อนข้อมูล

กดปุ่ม CLR เพื่อ CLEAR MEMORY ทั้งหมด โดยทำให้ MEMORY ตั้งแต่แอดเดรส 0000H-4000H เป็น 0000

กดปุ่ม ESC เพื่อต้องการกลับ MAIN MENU

4.1 กดปุ่ม ENT จะปรากฏหน้าจอให้ป้อนข้อมูลดังนี้

```

NUMBER
POWER

```

โดย NUMBER คือ ให้บันทึกหมายเลขมิเตอร์ไฟฟ้า จำนวน 6 หลัก

POWER คือ ให้บันทึกจำนวนหน่วยการใช้ไฟฟ้า จำนวน 4 หลัก

และในระหว่างการป้อนข้อมูลสามารถกดปุ่ม DEL เพื่อทำการลบตัวเลขที่ผิดพลาดได้ทีละหลัก
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการบันทึกข้อมูลจนครบแล้ว จะเข้าสู่หน้าจอ SUB MENU ดังนี้

PLEASE SELECT	
[OFF]	END PROGRAM
[ESC]	MAIN MENU
[ENT]	CONTINUE

กดปุ่ม OFF เพื่อจบการทำงาน

กดปุ่ม ESC เพื่อกลับไปสู่หน้าจอ MAIN MENU ในกรณีต้องการเลือกใช้ฟังก์ชันอื่น ๆ

กดปุ่ม ENT เพื่อป้อนข้อมูลรคคอร์ดต่อไปได้ทันที

4.2 กดปุ่ม CLR จะปรากฏหน้าจอต่อไปนี้ ค้างไว้ประมาณ 2 วินาที

CLEAR ALL DATA
PLEASE WAIT

และจะ ไปสู่หน้าจอดังนี้

CLEAR DATA
COMPLEAT

4.3 กดปุ่ม ESC เพื่อต้องการกลับ MAIN MENU

5. กรณีเลือก FN2 หน้าจอจะแสดงข้อความดังนี้

WATER
SYSTEM

รอประมาณ 3 วินาที หน้าจอจะแสดงข้อความดังต่อไปนี้

WATER SYSTEM	
[ENT]	ENT
[CLR]	CLEAR
[ESC]	MAIN MENU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กดปุ่ม ENT เพื่อทำการป้อนข้อมูล

กดปุ่ม CLR เพื่อ CLEAR MEMORY ทั้งหมด โดยทำให้ MEMORY ตั้งแต่แอดเดรส 4000H-7FFFH เป็น 0000

กดปุ่ม ESC เพื่อต้องการกลับ MAIN MENU

5.1 กดปุ่ม ENT จะปรากฏหน้าจอให้ป้อนข้อมูลดังนี้

NUMBER
POWER

โดย NUMBER คือ ให้บันทึกหมายเลขมิเตอร์ไฟฟ้า จำนวน 7 หลัก

POWER คือ ให้บันทึกจำนวนหน่วยการใช้ไฟฟ้า จำนวน 4 หลัก

และในระหว่างการป้อนข้อมูลสามารถกดปุ่ม DEL เพื่อทำการลบตัวเลขที่ผิดพลาดได้ที่ละหลัก เมื่อทำการบันทึกข้อมูลจนครบแล้ว จะเข้าสู่หน้าจอ SUB MENU ดังนี้

PLEASE SELECT
[OFF] END PROGRAM
[ESC] MAIN MENU
[ENT] CONTINUE

กดปุ่ม OFF เพื่อจบการทำงาน

กดปุ่ม ESC เพื่อกลับ ไปสู่หน้าจอ MAIN MENU ในกรณีต้องการเลือกใช้ฟังก์ชันอื่น ๆ

กดปุ่ม ENT เพื่อป้อนข้อมูลรคอร์ดต่อไปได้ทันที

5.2 กดปุ่ม CLR จะปรากฏหน้าจอต่อไปนี้ ค้างไว้ประมาณ 2 วินาที

CLEAR ALL DATA
PLEASE WAIT

และจะ ไปสู่หน้าจอดังนี้

CLEAR DATA
COMPLEAT

5.3 กดปุ่ม ESC เพื่อต้องการกลับ MAIN MENU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กรณีเลือก FN3 หน้าจอจะแสดงข้อความดังนี้

<p style="text-align: center;">SELECT MODE SEND</p> <p>1. ELECTRICAL</p> <p>2. WATER</p> <p>3. SEND</p>
--

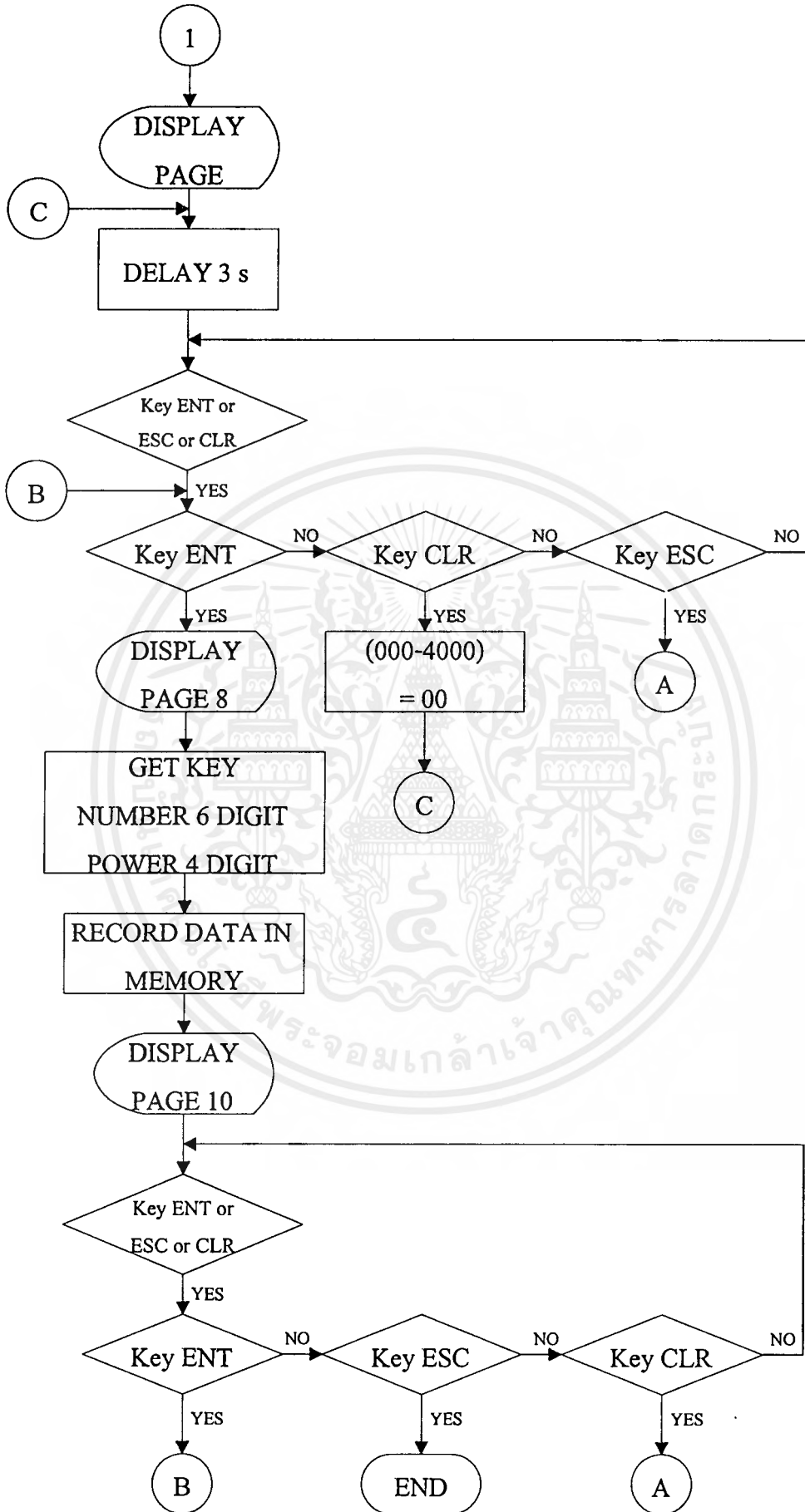
กดปุ่ม 1 เพื่อส่งข้อมูลไฟฟ้า

กดปุ่ม 2 เพื่อส่งข้อมูลน้ำประปา

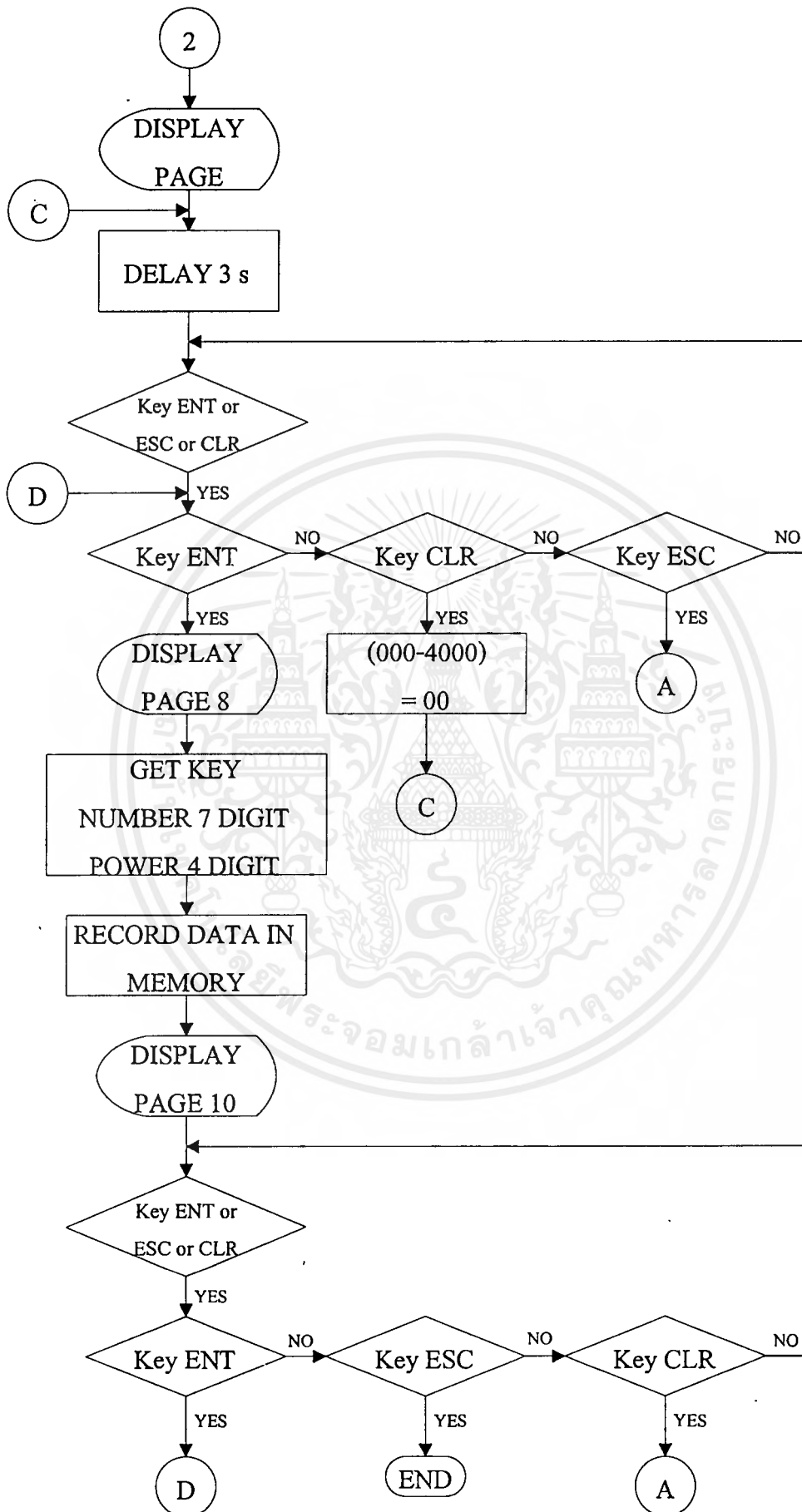
กดปุ่ม 3 กลับสู่ MAIN MENU

<p style="text-align: center;">SEND DATA</p> <p style="text-align: center;">PLEASE WAIT</p>

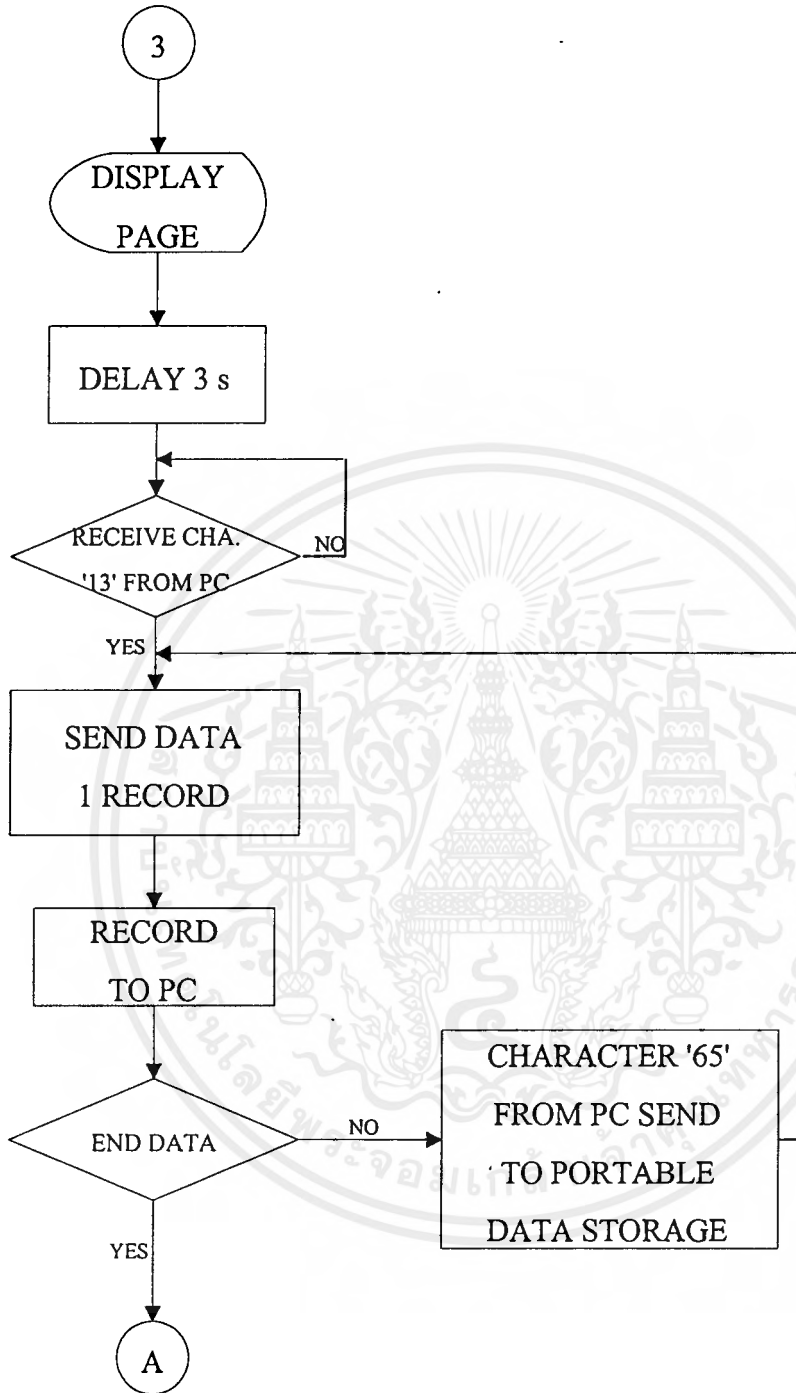
เมื่อส่งเสร็จเรียบร้อยแล้ว ก็จะกลับสู่หน้าจอเมนูหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

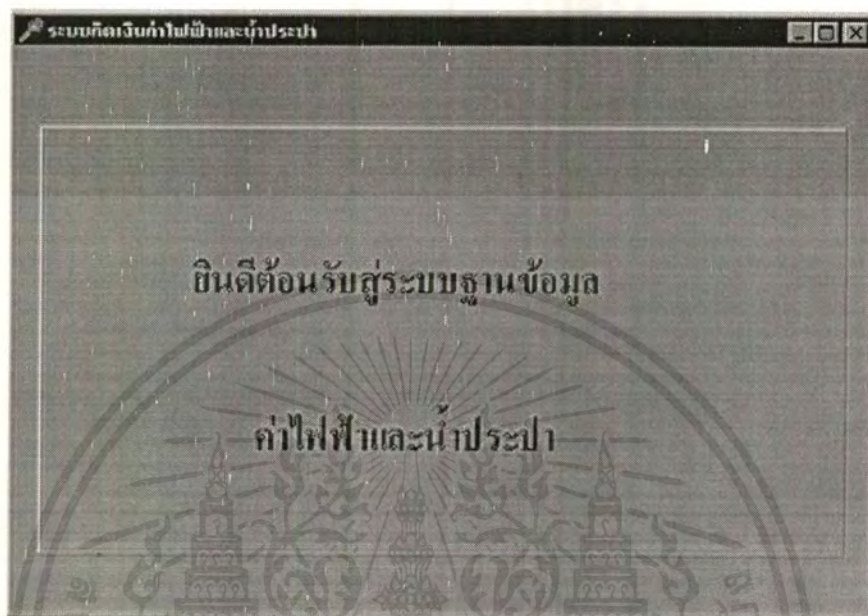


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองโปรแกรมส่วนรับข้อมูลและแสดงผล

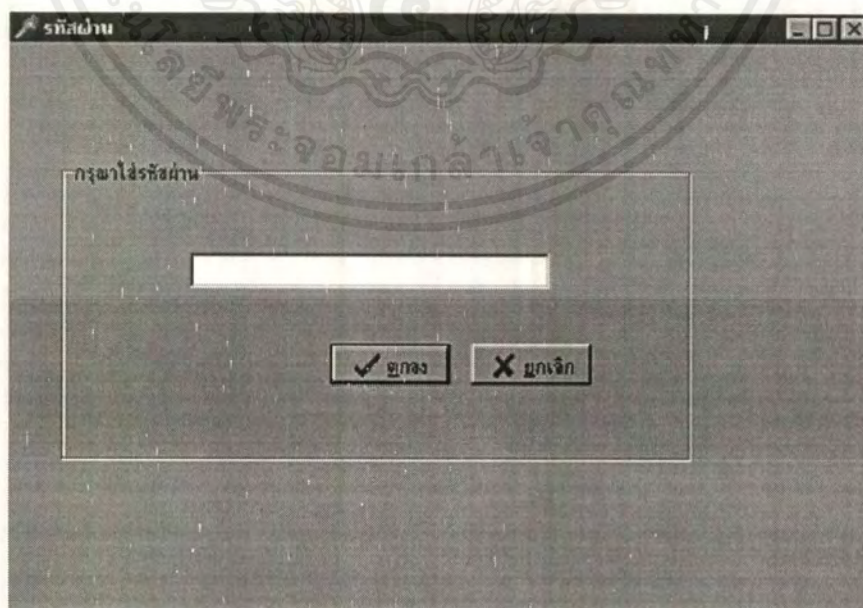
ลำดับขั้นตอนการทดลองส่วนของโปรแกรมแสดงผล

1. เมื่อทำการรันโปรแกรม ที่หน้าจอกอมพิวเตอร์ จะแสดงดังรูปที่ 4.1



รูปที่ 4.1 แสดงการเข้าสู่โปรแกรมแสดงผล

2. เมื่อประมาณ 3 วินาทีโปรแกรมจะให้ใส่รหัสผ่าน ซึ่งแสดงดังรูปที่ 4.2



รูปที่ 4.2 แสดงส่วนที่ให้ใส่รหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อใส่รหัสผ่านจะปรากฏหน้าจอเลือกฟังก์ชันที่จะทำ จะแสดงดังรูปที่ 4.3



รูปที่ 4.3 แสดงหน้าจอในการเลือกฟังก์ชันการทำงาน
จะประกอบด้วยปุ่มต่างๆ ดังนี้

1. รับข้อมูลไฟฟ้า
2. รับข้อมูลประปา
3. ข้อมูลผู้ใช้ไฟฟ้า
4. ข้อมูลผู้ใช้ประปา
5. พิมพ์ใบแจ้งหนี้
6. เปลี่ยนรหัสผ่าน
7. จบการทำงาน

3.1 กรณีเลือกรับข้อมูล ไฟฟ้า หมายถึงการรับข้อมูลไฟฟ้าจากเครื่องบันทึกข้อมูล แสดง
 ดังรูปที่ 4.4

Electricno	Electricunit	Day1	Month1	Year1
000002	400	8	3	42
000003	500	8	3	42
000001	325	8	4	42
000002	425	8	4	42
000003	525	8	4	42

รูปที่ 4.4 แสดงตารางรับข้อมูลไฟฟ้า

3.2 กรณีเลือกรับข้อมูลประปา หมายถึงการรับข้อมูลประปาจากเครื่องบันทึกข้อมูล
 แสดงดังรูปที่ 4.5

Waterno	Waterunit	Day1	Month1	Year1
0000002	2100	8	3	42
0000003	3100	8	3	42
0000001	1125	8	4	42
0000002	2125	8	4	42
0000003	3125	8	4	42

รูปที่ 4.5 แสดงตารางรับข้อมูลประปา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 กรณีเลือกการจัดการฐานข้อมูลไฟฟ้า จะเป็นการเพิ่มหรือลบข้อมูลของผู้ใช้ไฟฟ้า ซึ่งจะแสดงดังรูปที่ 4.6

รูปที่ 4.6 แสดงการจัดการฐานข้อมูลไฟฟ้า

จะประกอบด้วยปุ่มต่างๆ ดังนี้

1. เพิ่ม ทำหน้าที่เพิ่มจำนวนเรคคอร์ด
2. ลบ ทำหน้าที่ลบเรคคอร์ดที่เพิ่มเข้าไปใหม่
3. ยกเลิก ทำหน้าที่ยกเลิกเรคคอร์ดที่มีอยู่
4. ค้นหา ทำหน้าที่ค้นหาข้อมูลต่างๆของผู้ใช้ไฟฟ้า
5. ดูยอดเงิน ทำหน้าที่ดูยอดเงินของแต่ละเรคคอร์ด
6. เมนูหลัก ทำหน้าที่กลับสู่เมนูหลัก

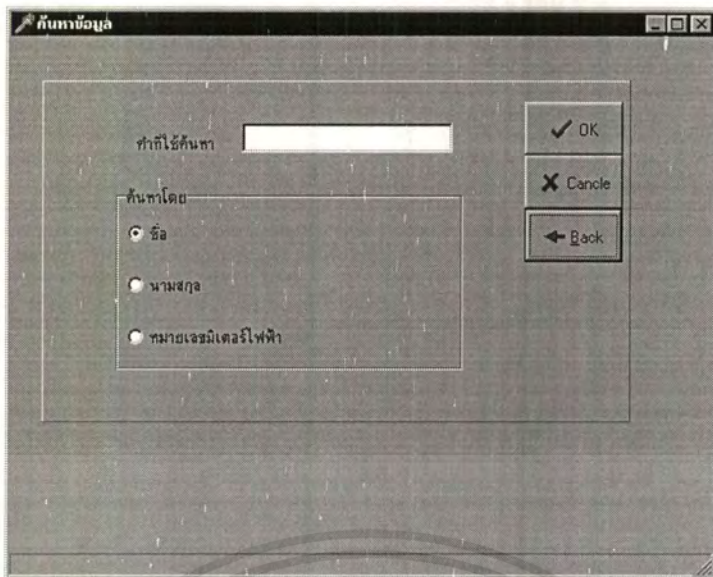
3.4 กรณีเลือกการจัดการฐานข้อมูลประปา จะเป็นการเพิ่มหรือลบข้อมูลของผู้ใช้น้ำประปา ซึ่งจะแสดงดังรูปที่ 4.7

รูปที่ 4.7 แสดงการจัดการฐานข้อมูลไฟฟ้า

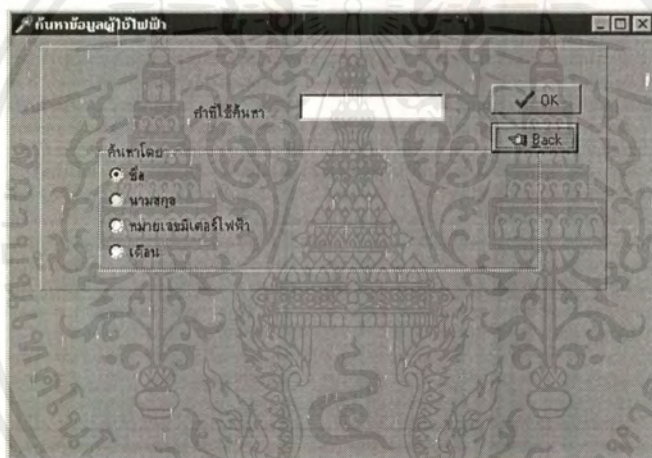
จะประกอบด้วยปุ่มต่างๆ ดังนี้

1. เพิ่ม ทำหน้าที่เพิ่มจำนวนเรคคอร์ด
2. บันทึก ทำหน้าที่บันทึกเรคคอร์ดที่เพิ่มเข้าไปใหม่
3. ยกเลิก ทำหน้าที่ยกเลิกเรคคอร์ดที่มีอยู่
4. ค้นหา ทำหน้าที่ค้นหาข้อมูลต่างๆของผู้ใช้ไฟฟ้า
5. ขูดเงิน ทำหน้าที่ขูดเงินของแต่ละเรคคอร์ด
6. เมนูหลัก ทำหน้าที่กลับสู่เมนูหลัก

ในส่วนของการจัดการฐานข้อมูลของไฟฟ้าและประปา จะมีส่วนของการค้นหาและการขูดเงิน ซึ่งจะแสดงดังรูปที่ 4.8 และรูปที่ 4.9 ตามลำดับ

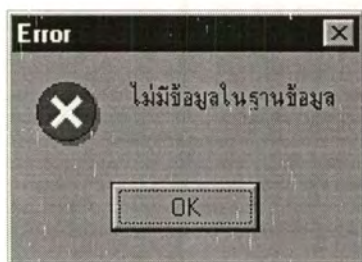


รูปที่ 4.8 แสดงส่วนการค้นหา



รูปที่ 4.9 แสดงส่วนการขอดูยอด

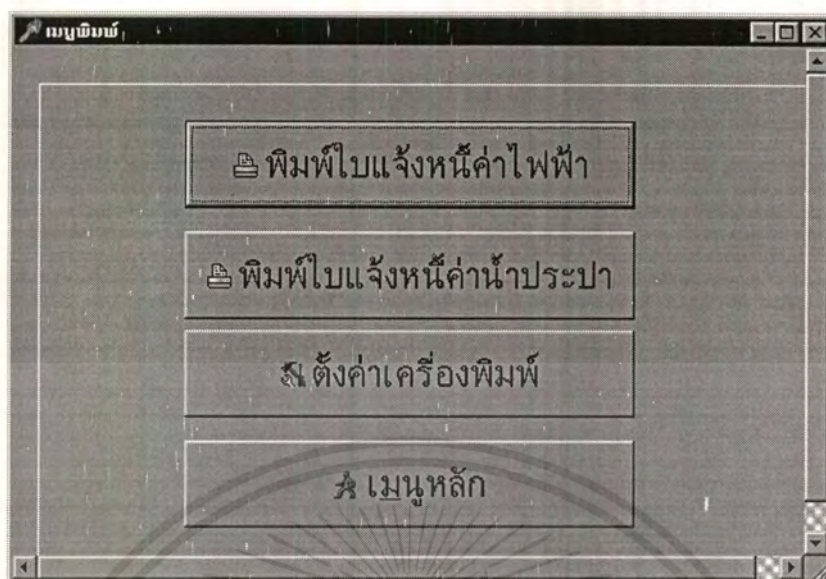
ในกรณีที่ได้ข้อมูลผิด โปรแกรมจะแสดงดังรูปที่ 4.10



รูปที่ 4.10 แสดงกรณีที่ได้ข้อมูลผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

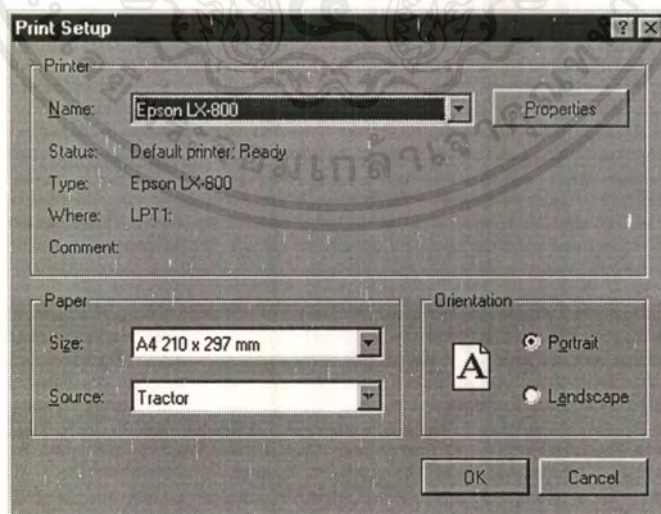
3.5 กรณีเลือกพิมพ์ใบแจ้งหนี้ ซึ่งจะแสดงดังรูปที่ 4.11



รูปที่ 4.11 แสดง MENU การสั่งพิมพ์ใบแจ้งหนี้

จะประกอบด้วยปุ่มต่างๆ ต่อไปนี้

1. พิมพ์ใบแจ้งหนี้ค่าไฟฟ้า
2. พิมพ์ใบแจ้งหนี้ค่าน้ำประปา
3. ตั้งค่าเครื่องพิมพ์ จะมีหน้าจอตั้งที่ 4.12

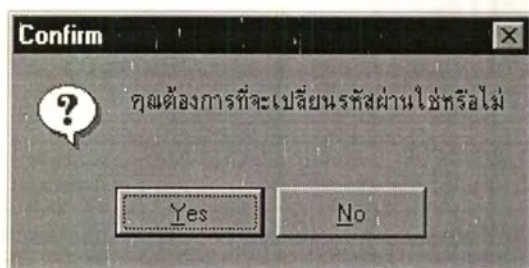


รูปที่ 4.12 การตั้งค่าเครื่องพิมพ์

4. เมนูหลัก เพื่อกลับสู่เมนูหลัก

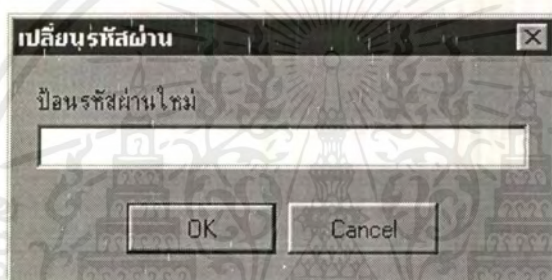
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 กรณีเลือกเปลี่ยนรหัสผ่าน ซึ่งจะแสดงดังรูปที่ 4.13



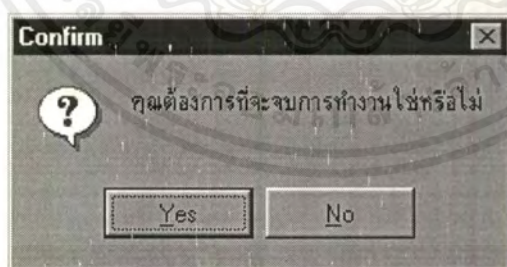
รูปที่ 4.13 แสดงการเปลี่ยนรหัส

ถ้าต้องการเปลี่ยนก็ตอบ YES หรือถ้าต้องการยกเลิกการเปลี่ยนแปลงก็ตอบ NO ในกรณีที่ตอบ YES จะปรากฏจอตงรูปที่ 4.14 และให้ใส่รหัสที่ต้องการเปลี่ยน



รูปที่ 4.14 แสดงการป้อนรหัสใหม่

3.7 กรณีเลือกจบการทำงาน ซึ่งจะแสดงดังรูปที่ 4.15



รูปที่ 4.15 แสดงการจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และบทสรุป

5.1 บทสรุป

ในการสร้างเครื่องบันทึกข้อมูลชนิดพกพา นี้ ในช่วงแรกได้ใช้เครื่อง JAZZ-31 ในการพัฒนาโปรแกรม ซึ่งมีความสะดวกในการพัฒนาโปรแกรมมาก แต่จะมีปัญหาในขั้นตอนการทดลองฮาร์ดแวร์ตามที่ได้ออกแบบไว้ การแก้ปัญหา จะใช้ EMULATOR เป็นเครื่องมือในการทดลองฮาร์ดแวร์ตามที่ได้ออกแบบไว้ ซึ่งได้ใช้ EPROM เบอร์ 27256 ในขั้นตอนสำเร็จจริง ๆ นั้น จะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 89C51 มาใช้ในการควบคุมการทำงาน เนื่องจากจะมีหน่วยความจำพิเศษ ประเภท Non-Volatile (NV RAM) มีแบตเตอรี่แบคอัพภายในตัวเอง ทำให้สามารถตัดแหล่งจ่ายออกไปจากตัว NV RAM โดยที่ข้อมูลที่บรรจุอยู่ใน NV RAM จะไม่สูญหาย อีกทั้งยังกินกำลังไฟต่ำ และยังสามารถลดการใช้ EPROM ลง เนื่องจาก 89C51 สามารถอัปเดตโปรแกรมลงไปในตัวมันได้

สำหรับการพัฒนาโปรแกรมที่ใช้ในการรับข้อมูล และ นำข้อมูลเหล่านั้นมาจัดการ โดยฐานข้อมูล จะใช้โปรแกรม DELPHI ในเขียนโปรแกรมส่วนนี้

5.2 ประโยชน์ที่ได้รับจากการทำโครงการ

ประโยชน์ที่ได้รับจากโครงการมี ดังนี้

1. มีความรู้ความเข้าใจในเกี่ยวกับการเขียน โปรแกรม ในแบบ Delphi มากขึ้น
2. มีความรู้ความเข้าใจในเกี่ยวกับการใช้งาน MCS-51 มากขึ้น
3. รู้จักการวางแผน ในการปฏิบัติงาน ให้เป็นระบบ
4. รู้จักหน้าที่ ความรับผิดชอบ ของแต่ละฝ่ายแต่ละคน
5. การรู้จักแก้ไขปัญหาต่างๆ ได้ด้วยตนเอง หรือขอคำปรึกษาจากอาจารย์ที่ปรึกษา
6. สามารถบริหารเวลา ได้อย่างเหมาะสม
7. ส่งเสริมการคิดที่มีระบบ ระเบียบ
8. เป็นแนวทางนำไปสู่การศึกษา ค้นคว้า วิจัย ในระดับการศึกษาที่สูงกว่าต่อไป

5.3 แนวทางการพัฒนาโครงการ

แนวทางต่อไปที่ควรมีการพัฒนาให้มีประสิทธิภาพสูงขึ้น มีดังนี้

1. สามารถทำให้ขนาดของเครื่องบันทึกข้อมูลมีขนาดเล็กและเบาลงได้
2. สามารถพัฒนาโปรแกรมควบคุม MCS-51 ให้มีฟังก์ชันการใช้งานให้มากกว่านี้ได้
3. สามารถพัฒนาโปรแกรมรับข้อมูลและแสดงผลให้มีฟังก์ชันการใช้งานให้มากกว่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
อัตราค่าไฟฟ้าและค่าประปา

อัตราค่าไฟฟ้า

ประเภทที่ 1 บ้านอยู่อาศัย

ลักษณะการใช้ สำหรับการใช้ไฟฟ้าในบ้านเรือนที่อยู่อาศัย วัดและโบสถ์ของศาสนาต่างๆ ตลอดจนบริเวณที่เกี่ยวข้อง โดยต่อผ่านเครื่องวัดหน่วยไฟฟ้าเครื่องเดียว

1.1 ปริมาณการใช้พลังงานไฟฟ้าไม่เกิน 150 หน่วยต่อเดือน

อัตรารายเดือน : ค่าพลังงานไฟฟ้า

5 หน่วย (กิโวลต์ชั่วโมง) แรก (หน่วยที่ 1-5)	เป็นเงิน	4.96	บาท
10 หน่วยต่อไป (หน่วยที่ 6-15)	หน่วยละ	0.7124	บาท
10 หน่วยต่อไป (หน่วยที่ 16-25)	หน่วยละ	0.8993	บาท
10 หน่วยต่อไป (หน่วยที่ 26-35)	หน่วยละ	1.1516	บาท
65 หน่วยต่อไป (หน่วยที่ 36-100)	หน่วยละ	1.5348	บาท
50 หน่วยต่อไป (หน่วยที่ 101-150)	หน่วยละ	1.6282	บาท
250 หน่วยต่อไป (หน่วยที่ 151-400)	หน่วยละ	2.1329	บาท
เกินกว่า 400 หน่วย (หน่วยที่ 401 เป็นต้นไป)	หน่วยละ	2.4226	บาท

ค่าไฟฟ้าน้ำต่ำสุด : เดือนละ 4.67 บาท

1.2 ปริมาณการใช้พลังงานไฟฟ้าเกินกว่า 150 หน่วยต่อเดือน

อัตรารายเดือน : ค่าพลังงานไฟฟ้า

35 หน่วย (กิโวลต์ชั่วโมง) แรก (หน่วยที่ 1-35)	เป็นเงิน	85.21	บาท
115 หน่วยต่อไป (หน่วยที่ 36-150)	หน่วยละ	1.1236	บาท
250 หน่วยต่อไป (หน่วยที่ 151-400)	หน่วยละ	2.1329	บาท
เกินกว่า 400 หน่วย (หน่วยที่ 401 เป็นต้นไป)	หน่วยละ	2.4226	บาท

ค่าไฟฟ้าน้ำต่ำสุด : เดือนละ 83.18 บาท

หมายเหตุ

1. ผู้ใช้ไฟฟ้าที่ติดตั้งเครื่องวัดหน่วยไฟฟ้า ขนาดไม่เกิน 5 แอมแปร์ 220 โวลท์ 1 เฟส 2 สาย จะถูกจัดให้อยู่ในประเภทที่ 1 ข้อ 1.1 แต่ถ้ามีการใช้พลังงานไฟฟ้าเกินกว่า 150 หน่วยต่อเดือนติดต่อกัน 3 เดือน ในเดือนถัดไปจะจัดเข้าอยู่ในประเภทที่ 1 ข้อ 1.2 และถ้ามีการใช้พลังงานไฟฟ้าไม่เกิน 150 หน่วยต่อเดือนติดต่อกัน 3 เดือน ในเดือนถัดไปจะจัดเข้าอยู่ในประเภทที่ 1 ข้อ 1.1 ตามเดิม

2. ผู้ใช้ไฟฟ้าที่ติดตั้งเครื่องวัดหน่วยไฟฟ้า ขนาดเกินกว่า 5 แอมแปร์ 220 โวลท์ 1 เฟส 2 สาย จะถูกจัดให้อยู่ในประเภทที่ 1 ข้อ 1.2 ตลอดไป

3. สถานที่ที่ใช้ในการประกอบศาสนกิจ ตลอดจนบริเวณที่เกี่ยวข้องสามารถเลือกใช้อัตราค่าไฟฟ้าประเภทที่ 3 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทที่ 2 กิจการขนาดเล็ก

ลักษณะการใช้ สำหรับการไฟฟ้าเพื่อประกอบธุรกิจ ธุรกิจรวมทั้งที่อยู่อาศัย อุตสาหกรรม และหน่วยงานรัฐวิสาหกิจ หรืออื่นๆ ตลอดจนบริเวณที่เกี่ยวข้อง ซึ่งมีความต้องการพลังงานไฟฟ้าเฉลี่ยใน 15 นาที สูงสุดต่ำกว่า 30 กิโลวัตต์ โดยต่อผ่านเครื่องวัดหน่วยไฟฟ้าเครื่องเดียว

อัตรารายเดือน : ค่าพลังงานไฟฟ้า

35 หน่วย (กิโลวัตต์ชั่วโมง) แรก (หน่วยที่ 1-35)	เป็นเงิน	89.89	บาท
115 หน่วยต่อไป (หน่วยที่ 36-150)	หน่วยละ	1.1236	บาท
250 หน่วยต่อไป (หน่วยที่ 151-400)	หน่วยละ	2.1329	บาท
เกินกว่า 400 หน่วย (หน่วยที่ 401 เป็นต้นไป)	หน่วยละ	2.4226	บาท
ค่าไฟฟ้าค่าสุด		87.85	บาท

ประเภทที่ 3 ส่วนราชการและองค์กรที่ไม่แสวงหากำไร

ลักษณะการใช้ สำหรับการไฟฟ้าของส่วนราชการ หน่วยงานตามกฎหมายว่าด้วยระเบียบบริหารราชการส่วนท้องถิ่น ตลอดจนบริเวณที่เกี่ยวข้อง ซึ่งมีปริมาณการใช้พลังงานไฟฟ้าเฉลี่ย 3 เดือนไม่เกิน 250,000 หน่วยต่อเดือน และองค์กรที่ไม่ใช่ส่วนราชการแต่มีวัตถุประสงค์ในการให้บริการโดยไม่คิดค่าตอบแทน รวมถึงสถานที่ที่ใช้ในการประกอบศาสนกิจ ตลอดจนบริเวณที่เกี่ยวข้อง แต่ไม่รวมถึงหน่วยงานรัฐวิสาหกิจ สถานทูต สถานที่ทำการเกี่ยวกับกิจการของต่างชาติ และสถานที่ทำการขององค์กรระหว่างประเทศ โดยต่อผ่านเครื่องวัดหน่วยไฟฟ้าเครื่องเดียว

อัตรารายเดือน

3.1 ระดับแรงดันไฟฟ้า 69 กิโลโวลท์ขึ้นไป

ค่าพลังงานไฟฟ้า :

10,000 หน่วย (กิโลวัตต์ชั่วโมง) แรก (หน่วยที่ 1-10,000)	เป็นเงิน	14,413.31	บาท
เกินกว่า 10,000 หน่วยขึ้นไป (หน่วยที่ 10,001 เป็นต้นไป)	หน่วยละ	1.4413	บาท

3.2 ระดับแรงดันไฟฟ้า 12-24 กิโลโวลท์ขึ้นไป

ค่าพลังงานไฟฟ้า :

300 หน่วย (กิโลวัตต์ชั่วโมง) แรก (หน่วยที่ 1-300)	เป็นเงิน	480.06	บาท
เกินกว่า 300 หน่วยขึ้นไป (หน่วยที่ 301 เป็นต้นไป)	หน่วยละ	1.6002	บาท

3.3 ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลท์

ค่าพลังงานไฟฟ้า :

10 หน่วย (กิโลวัตต์ชั่วโมง) แรก (หน่วยที่ 1-10)	เป็นเงิน	18.06	บาท
เกินกว่า 10 หน่วยขึ้นไป (หน่วยที่ 11 เป็นต้นไป)	หน่วยละ	1.8058	บาท

ค่าไฟฟ้าล่าสุด :

ระดับแรงดันไฟฟ้า 69 กิโลโวลท์ขึ้นไป	เดือนละ	13,831.78	บาท
ระดับแรงดันไฟฟ้า 12-24 กิโลโวลท์	เดือนละ	462.62	บาท
ระดับแรงดันไฟฟ้าต่ำกว่า 12 กิโลโวลท์	เดือนละ	17.48	บาท

หมายเหตุ

1. สถานที่ที่ใช้ในการประกอบศาสนกิจ ตลอดจนบริเวณที่เกี่ยวข้อง ที่จัดเข้าอยู่ในประเภทที่ 1 สามารถเลือกใช้อัตราค่าไฟฟ้าประเภทที่ 3 นี้ได้

ประเภทที่ 4 สูบน้ำเพื่อการเกษตร

ลักษณะการใช้ สำหรับการใช้ไฟฟ้ากับเครื่องสูบน้ำเพื่อการเกษตรของส่วนราชการ กลุ่มเกษตรกรที่ทางราชการรับรอง หรือ สหกรณ์เพื่อการเกษตร โดยต่อผ่านเครื่องวัดหน่วยไฟฟ้าเครื่องเดียว

อัตรารายเดือน : ค่าพลังงานไฟฟ้า

100 หน่วย (กิโลวัตต์ชั่วโมง) แรก (หน่วยที่ 1-100)	เป็นเงิน	115.16	บาท
เกินกว่า 100 หน่วยขึ้นไป (หน่วยที่ 101 เป็นต้นไป)	หน่วยละ	1.1516	บาท

ค่าไฟฟ้าล่าสุด : เดือนละ 109.35 บาท

หมายเหตุ

ผู้ใช้ไฟฟ้าในอัตราประเภทที่ 4 จะต้องมาติดต่อก่อนเพื่อทำสัญญากับการ ไฟฟ้านครหลวงก่อน

ข้อกำหนดเกี่ยวกับอัตราค่าไฟฟ้า

1. อัตราค่าไฟฟ้าข้างต้น ไม่รวมภาษีมูลค่าเพิ่ม
2. ค่าไฟฟ้าที่เรียกเก็บในแต่ละเดือน คือ ค่าไฟฟ้าตามอัตราค่าไฟฟ้าและค่าตัวประกอบการปรับอัตราค่าไฟฟ้าโดยอัตโนมัติ (Ft) ซึ่งเป็นค่าใช้จ่ายที่ไม่อยู่ในความควบคุมของการไฟฟ้า เช่น ราคาเชื้อเพลิงที่ใช้ในการผลิตไฟฟ้าเปลี่ยนแปลงไป รวมภาษีมูลค่าเพิ่ม โดยแสดงจำนวนเงินไว้ในใบเสร็จรับเงินค่าไฟฟ้าด้วย

อัตราค่าน้ำประปา

ประเภทที่ 1 : ที่พักอาศัย		ประเภทที่ 2 : ธุรกิจ รัฐวิสาหกิจ ราชการและอื่นๆ		ประเภทที่ 3 : อุตสาหกรรม	
ปริมาณน้ำใช้ (ม ³)	ราคาค่าน้ำ (บาท/ม ³)	ปริมาณน้ำใช้ (ม ³)	ราคาค่าน้ำ (บาท/ม ³)	ปริมาณน้ำใช้ (ม ³)	ราคาค่าน้ำ (บาท/ม ³)
0-30	เหมา 40 บาท	0-10	เหมา 80 บาท	0-10	เหมา 80 บาท
31-40	8.53	11-20	9.20	11-20	9.20
41-50	8.85	21-30	9.45	21-30	9.45
51-60	9.18	31-40	11.71	31-40	11.71
61-70	9.50	41-50	12.04	41-50	12.04
71-80	9.83	51-60	12.36	51-60	12.36
81-90	11.00	61-80	12.69	61-80	13.69
91-100	11.32	81-100	13.01	81-100	13.01
101-120	11.65	101-120	13.34	101-120	13.34
121-160	11.97	121-160	13.66	121-160	13.66
161-200	12.30	161-200	13.99	161-200	13.99
201 ขึ้นไป	12.95	201 ขึ้นไป	14.31	201-2000	14.18
				2001-4000	13.92
				4001-6000	13.40
				6001-10000	12.75
				10001-20000	12.10
				20001-30000	11.45
				30001-40000	10.80
				40001-50000	10.15
				50001 ขึ้นไป	9.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมควบคุมการทำงานของ MCS-51

```

;*****
;PROGRAM FOR PORTABLE DATA STORAGE
;BY KMITL 3T/2
; 1.MISS HARUTHAI SEESAWAT 39013067
; 2.MR. ANURAK SUMIPAN 39013068
; 3.MR. ANURAT KONGKEAW 39013069
;*****

ORG 8100H

PORTA EQU 0Fc00H ;LCD DATA (PA0-PA7)
PORTB EQU 0Fc01H ;LCD CONTROL (PB0-PB2)
PORTC EQU 0Fc02H
PORTP EQU 0Fc03H ;8255 CONTROL PORT

ELECCOUNT EQU 3ffcH
WATERCOUNT EQU 7ffcH
ELECSTART EQU 0000H
WATERSTART EQU 4000H

RTCDAT EQU P1.4
RTCRST EQU P1.6
RTCCLK EQU P1.5
SOUNDB EQU P1.2

DATEBUFF0 EQU 1AH
DATEBUFF1 EQU 1BH
DATEBUFF2 EQU 1CH
MONTHBUFF1 EQU 1DH
MONTHBUFF2 EQU 1EH
YEARBUFF1 EQU 1FH
YEARBUFF2 EQU 20H

KEYBUFF0 EQU 20H ;buffer of key
KEYBUFF1 EQU 21H
KEYBUFF2 EQU 22H
KEYBUFF3 EQU 23H
KEYBUFF4 EQU 24H
KEYBUFF5 EQU 25H
KEYBUFF6 EQU 26H
KEYBUFF7 EQU 27H
KEYBUFF8 EQU 28H
KEYBUFF9 EQU 29H
KEYBUFF10 EQU 2AH
KEYBUFF11 EQU 2BH

KEYDOWNBUFF EQU 2CH
KEYUPBUFF EQU 2DH
; ***** MAIN *****
MAIN:
MOV R2,#15H
ACALL DELAY
MOV DPTR,#PORTP ;SET 8255 CONTROL PORT
MOV A,#86H
MOVX @DPTR,A
ACALL CLR_LCD
MOV SP,#30H
ACALL INITLCD
ACALL PBEEP
MOV R7,#00H

LOGO: MOV DPTR,#PAGE1
ACALL DISPLAY
MOV R2,#0AH
ACALL DELAY

MOV DPTR,#PAGE2
ACALL DISPLAY
MOV R2,#0AH

```

```

                ACALL    DELAY
RDDATE:        ACALL    SHOWRTC
                ACALL    KEY_CORRECT

MAINMENU:     ACALL    CLR_LCD
                MOV     DPTR,#PAGE3      ;DISPLAY MAINMENU
                ACALL    DISPLAY
                MOV     R2,#05H
                ACALL    DELAY
                MOV     R7,#00H

CH_FUNC:      ACALL    KEY_FUNC

```

```

;*****SUBPROGRAM
; ***** DPADD SUB *****
; DPTR = DPTR + R2,R3
; IN  = DPTR,R2,R3
; OUT = DPTR
; REG = A,DPTR

```

```

DPADD:        MOV     A,DPL
                ADD     A,R3
                MOV     DPL,A
                MOV     A,DPH
                ADDC    A,R2
                MOV     DPH,A
                RET

```

```

; ***** DPSUB SUB *****
; DPTR = DPTR - R2,R3
; IN  = DPTR,R2,R3
; OUT = DPTR
; REG = A,DPTR

```

```

DPSUB:       CLR     C
                MOV     A,DPL
                SUBB    A,R3
                MOV     DPL,A
                MOV     A,DPH
                SUBB    A,R2
                MOV     DPH,A
                RET

```

```

;*****MULTIPLY SUB*****
;IN=B,A
;OUT=R2,R3

```

```

MUL:         MUL     AB
                MOV     R3,A
                MOV     R2,B
                RET

```

```

MUL19:       MOV     B,#19
                ACALL    MUL
                RET

```

```

MUL18:       MOV     B,#18
                ACALL    MUL
                RET

```

```

;*****
;GET VALUE FROM KEYBOARD 4*4

```

```

KEY:         MOV     R5,#4
                MOV     R1,#0F7H      ;FIRST SET
INKEY:       MOV     DPTR,#PORTC
                MOV     A,R1
                MOVX    @DPTR,A
SCAN:        MOVX    A,@DPTR      ;READ KEY
                MOV     R3,A
                ANL     A,#0F0H
                CJNE    A,#0F0h,KEYDOWN
                MOV     A,R1
                RR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R1,A
DJNZ R5,INKEY
SJMP KEY

KEYDOWN: MOV KEYDOWNBUFF,A
ACALL KBDL
MOV DPTR,#PORTC
MOVX A,@DPTR
MOV R3,A
ANL A,#0F0H
CJNE A,KEYDOWNBUFF,SCAN

BACK: MOVX A,@DPTR
MOV R3,A
ANL A,#0F0H
MOV KEYUPBUFF,A
ACALL KBDL
MOVX A,@DPTR
MOV R3,A
ANL A,#0F0H
CJNE A,KEYUPBUFF,BACK

RELEASE: MOV A,KEYDOWNBUFF
ACALL CHK_KEY
RET

CHK_KEY: ANL A,#0F0H
ROW1: CJNE A,#70H,ROW2
MOV DPTR,#TB_KEY
JMP COL

ROW2: CJNE A,#0B0H,ROW3
MOV DPTR,#TB_KEY+4
JMP COL

ROW3: CJNE A,#0D0H,ROW4
MOV DPTR,#TB_KEY+8
JMP COL

ROW4: CJNE A,#0E0H,SCAN
MOV DPTR,#TB_KEY+12
JMP COL
RET

;*****
;CHECK COLUMN

COL: MOV A,R3
ANL A,#0FH
COL1: CJNE A,#07H,COL2
RET

COL2: CJNE A,#0BH,COL3
INC DPTR
RET

COL3: CJNE A,#0DH,COL4
INC DPTR
INC DPTR
RET

COL4: CJNE A,#0EH,COL
INC DPTR
INC DPTR
INC DPTR
RET

;*****
;DELAY 20 mS. FOR PRESS AND RELEASE KEY

KBDL: MOV R3,#03H
DEL2: MOV R4,#00H
DEL1: DJNZ R4,DEL1
R3,DEL2
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****
;READ DATE MONTH YEAR FROM RTC(DS 1202)
```

```
READRTC:      MOV     R2,#8EH
               MOV     R3,#00H
               ACALL   RTCWR

               MOV     R2,#87H
               ACALL   RTCRD
               ACALL   HTOA
               MOV     DATEBUFF1,R2
               MOV     DATEBUFF2,R3

               MOV     R2,#89H
               ACALL   RTCRD
               ACALL   HTOA
               MOV     MONTHBUFF1,R2
               MOV     MONTHBUFF2,R3

               MOV     R2,#8DH
               ACALL   RTCRD
               ACALL   HTOA
               MOV     YEARBUFF1,R2
               MOV     YEARBUFF2,R3
               RET

SHOWRTC:      MOV     DPTR,#PAGEDATE
               ACALL   DISPLAY

               ACALL   READRTC
               MOV     A,DATEBUFF1
               ACALL   CKVALUE
               MOV     A,#0C3H
               ACALL   LCDLDS2

               MOV     A,DATEBUFF2
               ACALL   CKVALUE
               MOV     A,#0C4H
               ACALL   LCDLDS2

               MOV     A,MONTHBUFF1
               ACALL   CKVALUE
               MOV     A,#0C6H
               ACALL   LCDLDS2

               MOV     A,MONTHBUFF2
               ACALL   CKVALUE
               MOV     A,#0C7H
               ACALL   LCDLDS2

               MOV     A,YEARBUFF1
               ACALL   CKVALUE
               MOV     A,#0C9H
               ACALL   LCDLDS2

               MOV     A,YEARBUFF2
               ACALL   CKVALUE
               MOV     A,#0CAH
               ACALL   LCDLDS2
               RET

CKVALUE:      CJNE    A,#30H,CK1
               MOV     DPTR,#TB_KEY
               RET

CK1:          CJNE    A,#31H,CK2
               MOV     DPTR,#TB_KEY+11
               RET

CK2:          CJNE    A,#32H,CK3
               MOV     DPTR,#TB_KEY+10
               RET

CK3:          CJNE    A,#33H,CK4
```

```

MOV DPTR,#TB_KEY+9
RET
CK4: CJNE A,#34H,CK5
MOV DPTR,#TB_KEY+7
RET
CK5: CJNE A,#35H,CK6
MOV DPTR,#TB_KEY+6
RET
CK6: CJNE A,#36H,CK7
MOV DPTR,#TB_KEY+5
RET
CK7: CJNE A,#37H,CK8
MOV DPTR,#TB_KEY+3
RET
CK8: CJNE A,#38H,CK9
MOV DPTR,#TB_KEY+2
RET
CK9: CJNE A,#39H,OUT
MOV DPTR,#TB_KEY+1
OUT: RET

```

```

;*****
; CHANG HEX TO ASCII
;IN = A
;OUT = R2,R3

```

```

HTOA: PUSH ACC
SWAP A
ACALL HTOAS
MOV R2,A
POP ACC
ACALL HTOAS
MOV R3,A
RET

```

```

HTOAS: ANL A,#0FH
CJNE A,#0AH,$+3
JNC HTOAS1
ORL A,#30H
RET
HTOAS1: SUBB A,#9
ORL A,#40H
RET

```

```

; ***** ATOH SUB *****
; ASCII TO HEX CONVERT
; IN = R2,R3 30H,41H
; OUT = A 0AH
; REG = A,R2

```

```

ATOH: MOV A,R2
ACALL ATOHS
SWAP A
MOV R2,A
MOV A,R3
ACALL ATOHS
ORL A,R2
RET

```

```

ATOHS: CJNE A,#'A',$+3
JC ATOHS1
ADD A,#9
ATOHS1: ANL A,#0FH
RET

```

```

RTCRD: CLR RTCCLK ;CLK=0
ACALL RTCDL
SETB RTCRST ;RST=1
ACALL RTCDL

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,R2 ;COMMAND
ACALL RTCWRC

RTCWR1: MOV R4,#8 ;READ DATA
CLR A
CLR RTCCLK
ACALL RTCDL
MOV C,RTCDAT
RRC A
SETB RTCCLK
ACALL RTCDL
DJNZ R4,RTCWR1
MOV R3,A
CLR RTCRST ;RST=0
ACALL RTCDL
RET

; ***** RTCWR SUB *****
; WRITE SINGLE BYTE TO DS1202
; IN = R2 COMMAND
; R3 DATA
; REG = A,R2,R3,R4,R5

RTCWR: CLR RTCCLK ;CLK=0
ACALL RTCDL
SETB RTCRST ;RST=1
ACALL RTCDL
MOV A,R2 ;COMMAND
ACALL RTCWRC
MOV A,R3 ;DATA
ACALL RTCWRC
CLR RTCRST ;RST=0
ACALL RTCDL
RET

RTCWRC: MOV R4,#8 ;WRITE COMMAND/DATA
RTCWR1: RRC A
MOV RTCDAT,C
SETB RTCCLK ;RISING EDGE CLOCK
ACALL RTCDL
CLR RTCCLK
ACALL RTCDL
DJNZ R4,RTCWRC1
RET

RTCDL: MOV R5,#4 ;DELAY
DJNZ R5,$
RET

;*****
;INITIAL LCD

INITLCD: MOV A,#3EH ;FUNCTION SET
ACALL LCDWI
MOV A,#10H ;CURSOR/DISPLAY
ACALL LCDWI
MOV A,#0EH ;DISPLAY CONTROL
ACALL LCDWI
MOV A,#06H ;MODE SET
ACALL LCDWI
MOV A,#01H ;CLEAR
ACALL LCDWI
RET

DELAY: MOV R3,#00H
DELAY1: MOV R4,#00H
DJNZ R4,$
DJNZ R3,DELAY1
DJNZ R2,DELAY
RET

;*****
;SHOW LCD

DISPLAY: MOV A,#80H ;SET ADDRESS LINE 1
ACALL LCDLDS
MOV A,#0C0H ;SET ADDRESS LINE 2
ACALL LCDLDS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MOV     A,#90H           ;SET ADDRESS LINE 3
        ACALL  LCDLDS
        MOV     A,#0D0H        ;SET ADDRESS LINE 4
        ACALL  LCDLDS
        RET

LCDLDS:  PUSH   DPH             ;LOAD SUB.
        PUSH   DPL
        ACALL  LCDWI          ;WRITE ADDRESS
        POP    DPL
        POP    DPH
LCDLDS1: MOV     R2,#16          ;16 CHAR.
        CLR    A
        MOVC  A,@A+DPTR       ;MOVC FOR CODE / MOVX FOR DATA
        PUSH  DPH
        PUSH  DPL
        ACALL LCDWD           ;WRITE DATA
        POP   DPL
        POP   DPH
        INC   DPTR
        DJNZ  R2,LCDLDS1
        RET

LCDWI:   MOV     DPTR,#PORTA    ;DATA TO PORTA
        MOVX  @DPTR,A
        MOV   DPTR,#PORTB     ;PORTB READ MODIFY WRITE
        MOVX  A,@DPTR
        CLR   ACC.0            ;RS=0
        CLR   ACC.1            ;R/W=0
        CLR   ACC.2            ;ENABLE=0
        MOVX  @DPTR,A
        SETB  ACC.2            ;ENABLE=1
        MOVX  @DPTR,A
        CLR   ACC.2            ;ENABLE=0
        MOVX  @DPTR,A
        MOV   A,#0             ;DELAY
LCDWI1:  DEC    A
        JNZ   LCDWI1
        RET

LCDWD:   MOV     DPTR,#PORTA    ;DATA TO PORTA
        MOVX  @DPTR,A
        MOV   DPTR,#PORTB     ;PORTB READ MODIFY WRITE
        MOVX  A,@DPTR
        SETB  ACC.0            ;RS=1
        CLR   ACC.1            ;R/W=0
        CLR   ACC.2            ;ENABLE=0
        MOVX  @DPTR,A
        SETB  ACC.2            ;ENABLE=1
        MOVX  @DPTR,A
        CLR   ACC.2            ;ENABLE=0
        MOVX  @DPTR,A
        MOV   A,#0             ;DELAY
LCDWD1:  DEC    A
        JNZ   LCDWD1
        RET

;*****
;SHOW INTEGER THAT IS DATA ON LCD

SHOW_CHA: ADD     A,R7          ; CURSOR IS INCREMENT
          ACALL  LCDLDS2
          MOV    A,B

CH_DEL:   CJNE   A,#020H,RET2   ; IF DEL CURSOR IS DECREMENT
          DEC    R7
RET2:     RET

SHOWCURSOR: MOV    DPTR,#EMPTY
          ACALL  LCDLDS2
          RET

SHOW_FN:  MOV    A,#0CDH
          ACALL  LCDLDS2
          RET

;*****
;CLEAR LCD

```

```

;*****
CLR_LCD:      MOV     DFTR,#CLEAR
              ACALL  DISPLAY
              RET

;*****
LCDLDS2:     PUSH  DPH           ;LOAD SUB.
              PUSH  DPL
              ACALL LCDWI       ;WRITE ADDRESS
              POP   DPL
              POP   DPH
              MOV   R2,#01H     ;1 CHARECTER
              ACALL LCDLDS1
              RET

; ***** SOUND SUB *****

SOUND:      MOV   R5,#0           ;END FLAG
              MOV   R4,#80H      ;DELAY CONSTANT
SOUND1:     CALL  SOUNDS
              CJNE  R5,#1,SOUND1
              RET

SOUNDS:     SETB  SOUNDB         ;OUT 1
              CALL SOUNDX
              CLR  SOUNDB         ;OUT 0
CALL  SOUNDX
              RET

SOUNDX:     MOV   A,R2           ;FREQUENCY DELAY
SOUNDX1:    CALL  SOUNDY
              DEC  A
              JNZ  SOUNDX1

SOUNDY:     DJNZ  R4,SOUNDY1     ;LENGTH COUNT DOWN
              MOV  R4,#80H
              DJNZ R3,SOUNDY1
              MOV  R5,#1
SOUNDY1:    RET

KBEEP:      MOV   R2,#50H        ;KEY BEEP
              MOV   R3,#10H
              CALL  SOUND
              RET

HBEEP:      MOV   R2,#10H        ;SUCCESS BEEP
              MOV   R3,#40H
              CALL  SOUND
              RET

LBEEP:      MOV   R2,#90H        ;WARNING BEEP
              MOV   R3,#90H
              CALL  SOUND
              RET

EBEEP:      MOV   R2,#90H        ;ERROR BEEP
              MOV   R3,#00H
              CALL  SOUND
              MOV   R2,#0C0H
              MOV   R3,#00H
              CALL  SOUND
              RET

RBEEP:      MOV   R2,#4          ;READY BEEP
RBEEP1:     MOV   A,R2
              PUSH  ACC           ;-[
              MOV   R2,#70H
              MOV   R3,#15H
              CALL  SOUND
              MOV   R2,#00H
              MOV   R3,#16H
              CALL  DMSEC
              POP   ACC           ;-]

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R2,A
DJNZ R2,RBEEP1
RET

PBEEP: MOV R6,#30H ;POWER-UP BEEP
MOV R7,#38H
PBEEP2: MOV A,R7
MOV R2,A
MOV R3,#02H
CALL SOUND
MOV R2,#00H
MOV R3,#08H
CALL DMSEC
DEC R7
DJNZ R6,PBEEP2
RET

; ***** DMSEC SUB *****
; DELAY 1/1000 SECOND
; IN = R2,R3
; REG = A,R2,R3,R4

DMSEC: MOV R4,#230 ;1 MSEC LOOP
DMSEC1: NOP
NOP
DJNZ R4,DMSEC1
DJNZ R3,DMSEC
MOV A,R2
CJNE A,#0,DMSEC2
RET
DMSEC2: DEC R2
SJMP DMSEC

;*****SUBPROGRAM SENDDATA TO COMPUTER****

SETIX: MOV SCON,#050H ; MODE1 8BIT DATA 1STARTBIT 1STOP BIT
MOV TMOD,#20H ; MODE2
MOV TH1,#0FDH ; 9600 BAUD RATE
SETB TR1 ;START TIMER1

MOV DPTR,#PAGE15
ACALL DISPLAY

CHECKF3: ACALL KEY
CLR A
MOVC A,@A+DPTR
CJNE A,#31H,KEY2

MOV DPTR,#PAGE12
ACALL DISPLAY
MOV R2,#10H
ACALL DELAY

MOV DPTR,#ELECSTART
ACALL SEND
RET

KEY2: CJNE A,#32H,KEYESC
MOV DPTR,#PAGE12
ACALL DISPLAY
MOV R2,#10H
ACALL DELAY

MOV DPTR,#WATERSTART
ACALL SEND
RET

KEYESC: CJNE A,#3CH,CHECKF3
ACALL MAINMENU
RET

SEND: ACALL RBYTE ;WAIT RX OK
CJNE A,#13,SEND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GO:          INC    DPTR
            MOVX   A,@DPTR

STARTBYTE:  CJNE   A,#0FEH,GO      ;CHECK START
DATA BYTE:  INC    DPTR
            MOVX   A,@DPTR

            CJNE   A,#0FFH,TRANS  ;CHECK END
            INC   DPTR
            MOVX   A,@DPTR
            CJNE   A,#00H,ENDBYTE
            MOV    A,#66
            ACALL  SBYTE

            MOV    DPTR,#PAGE16
            ACALL  DISPLAY
            MOV    R2,#0AH
            ACALL  DELAY
            AJMP   MAINMENU
            RET

TRANS:      ACALL  SBYTE
            SJMP  DATA BYTE

ENDBYTE:    CJNE   A,#0FEH,ERROR  ;NEXT RECORD
            MOV    A,#67
            ACALL  SBYTE

REC:        ACALL  RBYTE
            CJNE   A,#65,REC      ;13H - ENTER
            SJMP  DATA BYTE

ERROR:      MOV    A,#65          ;SHOW "A" IF ERROR
            ACALL  SBYTE
            JMP    END1

; ***** SBYTE SUB *****
; SEND BYTE
; IN = A
; REG = NO

SBYTE:      MOV    SBUF,A
            JNB   TI,$          ;WAIT FOR SEND OK
            CLR   TI
            RET

; ***** RBYTE SUB *****
; READ ONE BYTE
; OUT = A
; REG = A

RBYTE:      JNB   RI,$          ;WAIT FOR RECEIVE OK
            CLR   RI
            MOV   A,SBUF
            RET

;*****
;GET INIEGER KEY FORM KEYBOARD
;OUT R7 (R7 COUNT)

GET_CH:     MOV    R2,#0ZH
            ACALL  DELAY

GET_CH2:    ACALL  KEY
            MOV    A,#00H
            MOVC  A,@A+DPTR
;           MOV    R5,A
            MOV    B,A
            SUBB  A,#3AH
            JNC   GET_CH2
            CJNE  A,#0E6H,RET1   ;20H-3AH = E6H

KEY_DEL:    DEC    R7
RET1:       RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;WRITE NUMBER OF METER1&METER2 TO BUFFER

WRBUFF1:  MOV     A,#KEYBUFF0
WRBUFF:   ADDC    A,R7
          MOV     R0,A
;         MOV     A,R5
          MOV     A,B
          MOV     @R0,A      ;ADDRESS 19-23H FOR KEY BUFF.
          RET

WRBUFF2:  MOV     A,#DATEBUFF0
          ACALL   WRBUFF
          RET

```

```

;*****
;CHECK DD/MM/YY
;IF CORECT PRESS ENTER
;IF INCORRECT PRESS CLR FOR SET DD/MM/YY

```

```

KEY_CORRECT:  ACALL   KEY
              CLR     A
              MOVC    A,@A+DPTR
              CJNE    A,#3EH,KEY_INCORECT
              RET

```

```

KEY_INCORECT: CJNE    A,#3FH,KEY_CORRECT
              ACALL   SEIDATE
              ACALL   SHOWRTC
              MOV     R7,#00H
              ACALL   KEY_CORRECT
              RET

```

```

;*****
;SHOW NUMBER OF CHOICE

```

```

SHOWFNC:     PUSH    DPH
              PUSH    DPL
              ACALL   RBEEP
              MOV     DPTR,#PAGE9      ; DISPLAY FUNCTION
              ACALL   DISPLAY
              MOV     R2,#03H
              ACALL   DELAY

              POP     DPL
              POP     DPH
              ACALL   SHOW_FN
              MOV     R2,#05H
              ACALL   DELAY
              RET

```

```

;*****
;CHECK CHOICE OF FUNCTION

```

```

KEY_FUNC:    ACALL   HBEEP
              ACALL   KEY
              CLR     A
              MOVC    A,@A+DPTR
              CJNE    A,#31H,KEY_F2
              ACALL   SHOWFNC

```

```

;*****
;CHOOSE FUNCTION 1 (ELECTRICAL SYSTEM)

```

```

KEY_F1:     ACALL   CLR_LCD
              MOV     DPTR,#PAGE4      ; DISPLAY ELECTRICAL SYSTEM
              ACALL   DISPLAY
              MOV     R2,#05H
              ACALL   DELAY
              ACALL   SUBFNC1

```

```

DATA1:     ACALL   METER1
              ACALL   MEMFN1
              ACALL   LOOP
              ACALL   HBEEP
              MOV     R7,#00H
              ACALL   CHOOSE1

```

RET

:CHOOSE FUNCTION 2 (WATER SYSTEM)

KEY_F2: CJNE A,#32H,KEY_F3
ACALL SHOWFNC
MOV R2,#02H
ACALL DELAY

WATERSYS: MOV DPTR,#PAGE5
ACALL DISPLAY

MOV R2,#05H
ACALL DELAY

DATA2: ACALL SUBFNC2
ACALL METER2
ACALL MEMFN2
ACALL LOOP
ACALL CHOOSE2
RET

:CHOOSE FUNCTION 3 (SEND DATA)

KEY_F3: CJNE A,#33H,KEY_FUNC
ACALL SHOWFNC
MOV R2,#03H
ACALL DELAY
ACALL SETIX
RET

SUBFNC1: ACALL KEY
CLR A
MOVC A,@A+DPTR
KEYENT1: CJNE A,#3EH,KEYF1CLR
ACALL LBEEP
RET

KEYF1CLR: CJNE A,#3FH,KEYF1ESC
AJMP CLRELEC
RET

KEYF1ESC: CJNE A,#3CH,SUBFNC1
AJMP MAINMENU
RET

SUBFNC2: ACALL KEY
CLR A
MOVC A,@A+DPTR
KEYENT2: CJNE A,#3EH,KEYF2CLR
ACALL LBEEP
RET

KEYF2CLR: CJNE A,#3FH,KEYF1ESC
AJMP CLRWATER
RET

:CLEAR ALL DATA IN RAM (ELECTRICAL)

CLRELEC: MOV DPTR,#ELECSTART
MOV R5,#OFFH
MOV R6,#40H
ACALL CLR
AJMP KEY_F1
RET

CLRWATER: MOV DPTR,#WATERSTART

```

MOV     R5,#0FFH
MOV     R6,#40H
ACALL  CLR
AJMP   WATERSYS
RET

CLR:    MOV     A,#00H
CLR2:   MOVX   @DPTR,A
        INC    DPTR
        DJNZ  R5,CLR2
        DJNZ  R6,CLR2

        MOV     DPTR,#PAGE11
        ACALL  DISPLAY
        MOV     R2,#0AH
        ACALL  DELAY
        MOV     DPTR,#PAGE14
        ACALL  DISPLAY
        MOV     R2,#0AH
        ACALL  DELAY

        RET

;*****
;SET DD/MM/YY

SETDATE: MOV     DPTR,#PAGE7 ; DAY MONTH AND YEAR
        ACALL  DISPLAY
        MOV     R2,#02H
        ACALL  DELAY
        ACALL  DAY
        MOV     R2,#02H

        ACALL  DELAY

        MOV     R2,DATEBUFF1
        MOV     R3,DATEBUFF2
        ACALL  ATOH
        MOV     R2,#86H ;WRITE DATE
        MOV     R3,A
        ACALL  RTCWR

        MOV     R2,MONTHBUFF1
        MOV     R3,MONTHBUFF2
        ACALL  ATOH
        MOV     R2,#88H
        MOV     R3,A
        ACALL  RTCWR

        MOV     R2,YEARBUFF1
        MOV     R3,YEARBUFF2
        ACALL  ATOH
        MOV     R2,#8CH
        MOV     R3,A
        ACALL  RTCWR
        RET

;*****
;GET DATA OF ELECTRICAL SYSTEM

METER1: MOV     DPTR,#PAGE8
        ACALL  DISPLAY
        MOV     A,#0C8H
        ACALL  SHOWCURSOR

KEYMETER: ACALL  KEYMETER1
        CJNE  R7,#06H,KEYMETER

        MOV     A,#98H
        ACALL  SHOWCURSOR

KEYPOWER: ACALL  KEYPOWER1
        CJNE  R7,#0AH,KEYPOWER
        RET

KEYMETER1: ACALL  GET_CH

```

```

                ACALL  WRBUFF 1
                MOV    A, #0C8H
                ACALL  SHOW_CHA
                ACALL  KBEEP
                RET

KEYPOWER1:    ACALL  GET_CH
                ACALL  WRBUFF 1
                MOV    A, #092H
                ACALL  SHOW_CHA
                ACALL  KBEEP
                RET

;*****
;GET DATA OF WATER SYSTEM

METER2:      MOV    DPTR, #PAGE8
                ACALL  DISPLAY
                MOV    A, #0C8H
                ACALL  SHOWCURSOR
KEYMETER2:   ACALL  KEYMETER1
                CJNE   R7, #07H, KEYMETER2

                MOV    A, #99H
                ACALL  SHOWCURSOR

POWER2:      ACALL  KEYPOWER1
                CJNE   R7, #0BH, POWER2
                RET

LOOP:        ACALL  KEY
                CLR    A
                MOVC  A, @A+DPTR
                CJNE  A, #3EH, LOOP
                RET

;*****
;WRITE NUMBER OF METER1(ELECTRICAL) TO RAM

MEMFN1:      ACALL  INC_COUNT1
                ACALL  MUL18
                MOV    DPTR, #ELECSTART
                ACALL  DPADD
                MOV    A, #0FFH                ;STOP VALUE
                MOVX  @DPTR, A
                ACALL  UNIT1
                ACALL  READRTC
                ACALL  MEMDATE
                DEC   DPL                    ;START VALUE
                MOV   A, #0FEH
                MOVX  @DPTR, A

                RET

;*****
;WRITE NUMBER OF METER2(WATER) TO RAM

MEMFN2:      ACALL  INC_COUNT2
                ACALL  MUL19
                MOV    DPTR, #WATERSTART
                ACALL  DPADD
                MOV    A, #0FFH                ;STOP VALUE
                MOVX  @DPTR, A
                ACALL  UNIT2
                ACALL  READRTC
                ACALL  MEMDATE
                DEC   DPL                    ;START VALUE
                MOV   A, #0FEH
                MOVX  @DPTR, A

                RET

;*****
;WRITE DATA TO RAM (SUB PROGRAM)

UNIT2:       MOV    A, KEYBUFF11
                DEC   DPL

```

```

MOVX    @DPTR,A

UNIT1:  MOV    A,KEYBUFF10
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF9
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF8
        DEC    DPL
        MOVX   @DPTR,A

MEMNO2: MOV    A,KEYBUFF7
        DEC    DPL
        MOVX   @DPTR,A

MEMNO1: MOV    A,KEYBUFF6
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF5
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF4
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF3
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF2
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,KEYBUFF1
        DEC    DPL
        MOVX   @DPTR,A
        RET

MEMDATE: MOV    A,YEARBUFF2    ;YEARBUFF 2
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,YEARBUFF1    ;YEARBUFF 1
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,MONTHBUFF2
        DEC    DPL

        MOVX   @DPTR,A

        MOV    A,MONTHBUFF1
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,DATEBUFF2
        DEC    DPL
        MOVX   @DPTR,A

        MOV    A,DATEBUFF1
        DEC    DPL
        MOVX   @DPTR,A
        RET
;*****
;INCREMENT DPTR(AT ADDRESS COUNTER)

INC_COUNT1: ACALL  COUNT1
            INC    A
            MOVX   @DPTR,A
            RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COUNT1:      MOV      DPTR,#ELECCOUNT
               MOVX    A,@DPTR
               RET

INC_COUNT2:   ACALL   COUNT2
               INC     A
               MOVX    @DPTR,A
               RET

COUNT2:      MOV      DPTR,#WATERCOUNT
               MOVX    A,@DPTR
               RET

;*****
;SET DATE/MONTH/YEAR TO DS1202

DAY:          MOV      A,#0C6H
               ACALL   SHOWCURSOR
DAY1:         ACALL   GET_CH
               ACALL   WRBUFF2
               MOV     A,#0C7H
               ACALL   SHOW_CHA
               CJNE   R7,#02H,DAY1

MONTH:        MOV      A,#097H
               ACALL   SHOWCURSOR
MONTH1:       ACALL   GET_CH
               ACALL   WRBUFF2
               MOV     A,#095H
               ACALL   SHOW_CHA
               CJNE   R7,#04H,MONTH1

YEAR:         MOV      A,#0D7H
               ACALL   SHOWCURSOR
YEAR1:        ACALL   GET_CH
               ACALL   WRBUFF2
               MOV     A,#0D3H
               ACALL   SHOW_CHA
               CJNE   R7,#06H,YEAR1
               RET

;*****
;SELCT ESC,ENT,ON/OFF

CHOOSE1:     ACALL   SUBELEC
               RET

CHOOSE2:     ACALL   SUBWATER
               RET

SUBELEC:      MOV      DPTR,#PAGE10 ;SELECT ESC, ENT, ON/OFF
               ACALL   DISPLAY
               MOV     R2,#05H
               ACALL   DELAY

KEY_AGAIN1:  ACALL   KEY
               CLR     A
               MOVC   A,@A+DPTR

KEY_ESC1:    CJNE   A,#3CH,ON_OFF1
               ACALL   HBEEP
               AJMP   MAINMENU
               RET

ON_OFF1:     CJNE   A,#3AH,KEY_ENT1
               ACALL   EBEEP
               JMP    END1
               RET

KEY_ENT1:    CJNE   A,#3EH,KEY_AGAIN1
               ACALL   HBEEP
               MOV     R7,#00H
               AJMP   DATA1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET

SUBWATER:  MOV    DPTR,#PAGE10 ;SELECT ESC, ENT, ON/OFF
           ACALL  DISPLAY
           MOV    R2,#05H
           ACALL  DELAY

KEY_AGAIN2: ACALL  KEY
           CLR    A
           MOVC   A,@A+DPTR

KEY_ESC2:  CJNE   A,#3CH,ON_OFF2
           ACALL  HBEEP
           AJMP  MAINMENU
           RET

ON_OFF2:   CJNE   A,#3AH,KEY_ENT2
           ACALL  EBEEP
           JMP   END1
           RET

KEY_ENT2:  CJNE   A,#3EH,KEY_AGAIN2
           ACALL  HBEEP
           MOV    R7,#00H
           AJMP  DATA2
           RET

```

```

;*****
;DISPLAY

```

```

PAGE1:    DB    "    PROJECT    "
           DB    "    3I/2    "
           DB    "    PORTABLE    "
           DB    "    DATA STORAGE    "

PAGE2:    DB    "    BY    "
           DB    "    39013067    "
           DB    "    39013068    "
           DB    "    39013069    "

PAGE3:    DB    "    SELECT FUNCTION"
           DB    "    1. ELECTRICAL "
           DB    "    2. WATER    "
           DB    "    3. SEND    "

PAGE4:    DB    "    ELECTRICAL    "
           DB    "[ENT]  ENTER    "
           DB    "[CLR]  CLEAR    "
           DB    "[ESC]  MAIN MENU"

PAGE5:    DB    "    WATER SYSTEM    "
           DB    "[ENT]  ENTER    "
           DB    "[CLR]  CLEAR    "
           DB    "[ESC]  MAIN MENU"

PAGE7:    DB    "    SET DD/MM/YY    "
           DB    "    DAY    "
           DB    "    MONTH    "
           DB    "    YEAR    "

PAGE8:    DB    "*****"
           DB    "    NUMBER    "
           DB    "    POWER    "
           DB    "*****"

PAGE9:    DB    "    "
           DB    "    FUNCTION    "
           DB    "    "
           DB    "    "

PAGE10:   DB    "    PLEASE SELECT    "
           DB    "[OFF]END PROGRAM"
           DB    "[ESC]MAIN MENU    "
           DB    "[ENT]CONTINEUE    "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PAGE11:      DB      "*****"
              DB      "  CLEAR ALL DATA"
              DB      "  PLEASE WAIT  "
              DB      "*****"

PAGE12:      DB      "*****"
              DB      "  SEND DATA  "
              DB      "  PLEASE WAIT  "
              DB      "*****"

PAGE13:      DB      "CLEAR ALL DATA ?"
              DB      "*****"
              DB      "[ENT] YES  "
              DB      "[CLR] NO   "

PAGE14:      DB      "*****"
              DB      "  CLEAR DATA  "
              DB      "  COMPLETE    "
              DB      "*****"

PAGE15:      DB      "SELECT MODE SEND"
              DB      "[1] ELECTRICAL "
              DB      "[2] WATER      "
              DB      "[ESC] MAIN MENU "

PAGE16:      DB      "*****"
              DB      "  SEND DATA  "
              DB      "  COMPLETE    "
              DB      "*****"

TB_KEY:      DB      30H,39H,38H,37H      :0,9,8,7
              DB      3FH,36H,35H,34H      :CLR,6,5,4
              DB      3EH,33H,32H,31H      :ENT,3,2,1
              DB      3AH,20H,3BH,3CH      :ON/OFF,DEL,EMPTY,ESC

PAGEDATE:    DB      "  DD/MM/YY  "
              DB      "  /  "
              DB      "[ENT] CORECT "
              DB      "[CLR] INCORECT "

CLEAR:       DB      "  "
              DB      "  "
              DB      "  "
              DB      "  "

EMPTY:       DB      20H

END1:        END

```

ภาคผนวก ค

โปรแกรม DELPHI ที่ใช้เป็นส่วนแสดงผล

```

unit portable1;

interface

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

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses portable2;

{$R *.DFM}

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  form1.hide;
  form2.show;
  timer1.enabled:=false;
end;

end.

```

```

unit portable2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons, Db, DBTables;

type
  TForm2 = class(TForm)
    GroupBox1: TGroupBox;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Table1: TTable;
    DataSource1: TDataSource;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public
  { Public declarations }
end;

var
  Form2: TForm2;

implementation

uses portable1, portable3;

{$R *.DFM}

procedure TForm2.BitBtn1Click(Sender: TObject);
var s:string;
    sflag1,sflag2:boolean;
begin
  table1.open;
  table1.first;

  s:=table1.fieldbyname('password1').asString;
  if edit1.text=s then sflag1:=true;
  if edit1.text='haruthai' then sflag2:=true;

  if sflag1 or sflag2 then
    begin
      form2.hide;
      form3.show;
    end
  else
    begin
      messagedlg('รหัสผ่านไม่ถูกต้อง กรุณาใส่รหัสผ่านใหม่',mterror,[mbok],0);
      edit1.setfocus;
      edit1.text:='';
    end;
end;

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
  application.terminate;
end;

end.

unit portable3;

interface

uses

  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Menus, Buttons, ExtCtrls, Db, DBTables, Grids, Calendar;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    BitBtn1: TBitBtn;
    Table1: TTable;
    DataSource1: TDataSource;
    Table2: TTable;
    DataSource2: TDataSource;

```

```

Table3: TTable;
DataSource3: TDataSource;
Table4: TTable;
DataSource4: TDataSource;
Table6: TTable;
DataSource6: TDataSource;
Table2Electricunit: TFloatField;
Table2Day1: TFloatField;
Table2Month1: TFloatField;
Table2Year1: TFloatField;
Calendar1: TCalendar;
Table7: TTable;
DataSource7: TDataSource;
BitBtn5: TBitBtn;
Edit1: TEdit;
Table8: TTable;
DataSource8: TDataSource;
BitBtn6: TBitBtn;
Table2oldelectricunit: TIntegerField;
Table2totalelectricunit: TIntegerField;
Table2name: TStringField;
Table2surname: TStringField;
Table2addr1: TStringField;
Table2province: TStringField;
Table2zip: TStringField;
Table2telephone: TStringField;
Table5: TTable;
Table9: TTable;
Table10: TTable;
Table5Waterno: TStringField;
Table5Day1: TFloatField;
Table5Month1: TFloatField;
Table5Year1: TFloatField;
Table5oldwaterunit: TIntegerField;
Table5totalwaterunit: TIntegerField;
Table5name: TStringField;
Table5surname: TStringField;
Table5addr1: TStringField;
Table5province: TStringField;
Table5zip: TStringField;
Table5telephone: TStringField;
Table5typewater: TFloatField;
Table11: TTable;
Table5waterpay: TCurrencyField;
Table2Electricno: TStringField;
Table2typeelectric: TFloatField;
Table2electricpay: TCurrencyField;
Table5Waterunit: TFloatField;
BitBtn7: TBitBtn;
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Table2CalcFields(DataSet: TDataSet);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure Table5CalcFields(DataSet: TDataSet);
procedure BitBtn7Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
  Form3: TForm3;

implementation

uses portable4, portable5, portable7, portable8 ,portable2, portable12,
  portable14;

{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    form3.hide;
    form4.show;
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
begin
    form3.hide;
    form8.show;
end;

procedure TForm3.BitBtn4Click(Sender: TObject);
begin
    table2.open;
    table4.open;
    table4.edit;

while not table2.eof do
begin
    table4.append;
    table4.fieldbyname('name').asString:=table2.fieldbyname('name').asString;

    table4.fieldbyname('surname').asString:=table2.fieldbyname('surname').asString;
    table4.fieldbyname('addr1').asString:=table2.fieldbyname('addr1').asString;

    table4.fieldbyname('province').asString:=table2.fieldbyname('province').asString;
;
    table4.fieldbyname('zip').asString:=table2.fieldbyname('zip').asString;

    table4.fieldbyname('telephone').asString:=table2.fieldbyname('telephone').asstring;

    table4.fieldbyname('month1').asinteger:=table2.fieldbyname('month1').asinteger;
    table4.fieldbyname('year1').asinteger:=table2.fieldbyname('year1').asinteger;
    table4.fieldbyname('day1').asinteger:=table2.fieldbyname('day1').asinteger;

    table4.fieldbyname('electricno').asString:=table2.fieldbyname('electricno').asString;
    table4.fieldbyname('oldelectricunit').asinteger:=table2.fieldbyname
('oldelectricunit').asinteger;
    table4.fieldbyname('thiselectricunit').asinteger:=table2.fieldbyname
('electricunit').asinteger;
    table4.fieldbyname('totalelectricunit').asinteger:=table2.fieldbyname
('totalelectricunit').asinteger;
    table4.fieldbyname('electricpay').ascurrency:=table2.fieldbyname
('electricpay').ascurrency;
    table4.fieldbyname('electrictype').asfloat:=table2.fieldbyname
('typeelectric').asfloat;
    table4.next;
    table2.next;
end;

    table5.open;
    table11.open;
    table11.edit;

while not table5.eof do
begin
    table11.append;
    table11.fieldbyname('name').asString:=table5.fieldbyname('name').asString;

    table11.fieldbyname('surname').asString:=table5.fieldbyname('surname').asString;
    table11.fieldbyname('addr1').asString:=table5.fieldbyname('addr1').asString;

    table11.fieldbyname('province').asString:=table5.fieldbyname('province').asString;
;
    table11.fieldbyname('zip').asString:=table5.fieldbyname('zip').asString;

    table11.fieldbyname('telephone').asString:=table5.fieldbyname('telephone').asstring;

    table11.fieldbyname('waterno').asString:=table5.fieldbyname('waterno').asString;
    table11.fieldbyname('oldwaterunit').asinteger:=table5.fieldbyname
('oldwaterunit').asinteger;
    table11.fieldbyname('thiswaterunit').asinteger:=table5.fieldbyname
('waterunit').asinteger;

```

```

    table11.fieldbyname('totalwaterunit').asinteger:=table5.fieldbyname
('totalwaterunit').asinteger;
    table11.fieldbyname('waterpay').ascurrency:=table5.fieldbyname
('waterpay').ascurrency;

table11.fieldbyname('month1').asinteger:=table5.fieldbyname('month1').asinteger;

table11.fieldbyname('year1').asinteger:=table5.fieldbyname('year1').asinteger;
    table11.fieldbyname('day1').asinteger:=table5.fieldbyname('day1').asinteger;

table11.fieldbyname('watertype').asfloat:=table5.fieldbyname('typewater').asfloat;
t;
    table11.next;
    table5.next;
end;

table5.first;
table6.open;
table6.edit;
while not table5.eof do
begin

if table5.fieldbyname('month1').asinteger=calendar1.month then
begin
    table6.append;
    table6.fieldbyname('name').asString:=table5.fieldbyname('name').asString;

table6.fieldbyname('surname').asString:=table5.fieldbyname('surname').asString;
    table6.fieldbyname('addr1').asString:=table5.fieldbyname('addr1').asString;

table6.fieldbyname('province').asString:=table5.fieldbyname('province').asString;
;
    table6.fieldbyname('zip').asString:=table5.fieldbyname('zip').asString;

table6.fieldbyname('telephone').asString:=table5.fieldbyname('telephone').asstri
ng;

table6.fieldbyname('waterno').asString:=table5.fieldbyname('waterno').asString;
    table6.fieldbyname('oldwaterunit').asinteger:=table5.fieldbyname
('oldwaterunit').asinteger;
    table6.fieldbyname('thiswaterunit').asinteger:=table5.fieldbyname
('waterunit').asinteger;
    table6.fieldbyname('totalwaterunit').asinteger:=table5.fieldbyname
('totalwaterunit').asinteger;

table6.fieldbyname('waterpay').ascurrency:=table5.fieldbyname('waterpay').ascurr
ency;

table6.fieldbyname('month1').asinteger:=table5.fieldbyname('month1').asinteger;
    table6.fieldbyname('year1').asinteger:=table5.fieldbyname('year1').asinteger;
    table6.fieldbyname('day1').asinteger:=table5.fieldbyname('day1').asinteger;

table6.fieldbyname('watertype').asfloat:=table5.fieldbyname('typewater').asfloat
;
    table6.next;
    table5.next;
end;
end;

table2.first;
table7.open;
table7.edit;
while not table2.eof do
begin

if table2.fieldbyname('month1').asinteger=calendar1.month then
begin
    table7.append;
    table7.fieldbyname('name').asString:=table2.fieldbyname('name').asString;

table7.fieldbyname('surname').asString:=table2.fieldbyname('surname').asString;
    table7.fieldbyname('addr1').asString:=table2.fieldbyname('addr1').asString;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

table7.fieldbyname('province').asString:=table2.fieldbyname('province').asString;
table7.fieldbyname('zip').asString:=table2.fieldbyname('zip').asString;
table7.fieldbyname('telephone').asString:=table2.fieldbyname('telephone').asString;
table7.fieldbyname('electricno').asString:=table2.fieldbyname('electricno').asString;
table7.fieldbyname('oldelectricunit').asinteger:=table2.fieldbyname('oldelectricunit').asinteger;
table7.fieldbyname('thiselectricunit').asinteger:=table2.fieldbyname('electricunit').asinteger;
table7.fieldbyname('totalelectricunit').asinteger:=table2.fieldbyname('totalelectricunit').asinteger;
table7.fieldbyname('electricpay').ascurrency:=table2.fieldbyname('electricpay').ascurrency;
table7.fieldbyname('month1').asinteger:=table2.fieldbyname('month1').asinteger;
table7.fieldbyname('year1').asinteger:=table2.fieldbyname('year1').asinteger;
table7.fieldbyname('day1').asinteger:=table2.fieldbyname('day1').asinteger;
table7.fieldbyname('electrictype').asfloat:=table2.fieldbyname('typeelectric').asfloat;
table7.next;
table2.next;
end;
end;

form3.hide;
form7.show;
table1.close;
table2.close;

end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
if
messagedlg('คุณต้องการที่จะจบการทำงานหรือไม่',mtconfirmation,[mbytes,mbno],0)-mryes
then application.terminate;
end;

procedure waterpay(watertype:variant;totalwater:integer;var paywater:real);
var pay1,tax:real;
total2,total3,total4,total5,total6,total7,total8:integer;
total9,total10,total11,total12:integer;
begin
total2:=-1;
total3:=-1;
total4:=-1;
total5:=-1;
total6:=-1;
total7:=-1;
total8:=-1;
total9:=-1;
total10:=-1;
total11:=-1;
total12:=-1;

if (watertype=2.00) then
begin
if totalwater<=10 then pay1:=80
else total2:=totalwater-10;

if (total2<=10)and (total2>0) then pay1:=80+(total2*9.20)
else total3:=total2-10;

if (total3<=10) and (total3>0)then pay1:=-80+(10*9.20)+(total3*9.45)
else total4:=total3-10;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (total4<=10) and (total4>0) then pay1:=80+(10*9.20)+(10*9.45)+
(total4*11.71)
    else total5:=total4-10;

    if (total5<=10) and (total5>0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(total5*12.04)
    else total6:=total5-10;

    if (total6<=10) and (total6>0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(10*12.04)+(total6*12.36)
    else total7:=total6-10;

    if (total7<=10) and (total7>=0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(10*12.04)+(10*12.36)+(total7*12.69)
    else total8:=total7-20;

    if (total8<=10) and (total8>=0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(10*12.04)+(10*12.36)+(20*12.69)+(total8*13.01)
    else total9:=total8-20;

    if (total9<=20) and (total9>=0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(10*12.04)+(10*12.36)+(20*12.69)+(20*13.01)+(total9*13.34)
    else total10:=total9-40;

    if (total10<=40) and (total10>=0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(10*12.04)+(10*12.36)+(20*12.69)+(20*13.01)+(20*13.34)+
(total10*13.66)
    else total11:=total10-40;

    if (total11<=40) and (total11>=0) then pay1:=80+(10*9.20)+(10*9.45)+
(10*11.71)+(10*12.04)+(10*12.36)+(20*12.69)+(20*13.01)+(20*13.34)+(40*13.66)+
(total11*13.99)
    else total12:=total11-40;

    if (total12>=0) then pay1:=80+(10*9.20)+(10*9.45)+(10*11.71)+(10*12.04)+
(10*12.36)+(20*12.69)+(20*13.01)+(20*13.34)+(40*13.66)+(40*13.99)+
(total12*14.31);

end

else if (watertype=1.00) then
begin
    if totalwater<=30 then
    begin
        pay1:= totalwater*7;
        if pay1 < 40 then pay1:=40;
    end

    else total2:=totalwater-30;

    if (total2<=10)and (total2>=0) then pay1:=210+(total2*8.53)
    else total3:=total2-10;

    if (total3<=10) and (total3>=0)then pay1:=210+(10*8.53)+(total3*8.85)
    else total4:=total3-10;

    if (total4<=10) and (total4>=0) then pay1:=210+(10*8.53)+(10*8.85)+
(total4*9.18)
    else total5:=total4-10;

    if (total5<=10) and (total5>=0) then pay1:=210+(10*8.53)+(10*8.85)+
(10*9.18)+(total5*9.50)
    else total6:=total5-10;

    if (total6<=10) and (total6>=0) then pay1:=210+(10*8.53)+(10*8.85)+
(10*9.18)+(10*9.50)+(total6*9.83)
    else total7:=total6-10;

    if (total7<=10) and (total7>=0) then pay1:=210+(10*8.53)+(10*8.85)+
(10*9.18)+(10*9.50)+(10*9.83)+(total7*11)
    else total8:=total7-10;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (total8<=10) and (total8>=0) then pay1:=-210+(85.3)+(88.5)+(91.8)+(95)+
(98.3)+(110)+(total8*11.32)
    else total9:=-total8-20;

    if (total9<=20) and (total9>=0) then pay1:=-210+(85.3)+(88.5)+(91.8)+(95)+
(98.3)+(110)+(113.2) +(total9*11.65)
    else total10:=-total9-40;

    if (total10<=40) and (total10>=0) then pay1:=-210+(85.3)+(88.5)+(91.8)+
(95)+(98.3)+(110)+(113.2) +(20*11.65)+(total10*11.97)
    else total11:=-total10-40;

    if (total11<=40) and (total11>=0) then pay1:=-210+(85.3)+(88.5)+(91.8)+
(95)+(98.3)+(110)+(113.2) +(20*11.65)+(40*11.97)+(total11*12.30)
    else total12:=-total11-40;

    if (total12>=0) then pay1:=-210+(85.3)+(88.5)+(91.8)+(95)+(98.3)+(110)+
(113.2) +(20*11.65)+(40*11.97)+(40*12.30)+(total12*12.95);

    end
    else showmessage('not correct water');

    tax:=(pay1)/10;          // tax 10%
    paywater:=pay1+tax;
end;

procedure electricpay(electrictype:variant;totalelectric:integer;var pay:real);
var pay1,payft,tax,ft:real;
    total2,total3,total4,total5,total6,total7,total8:integer;

begin
    total2:=-1;
    total3:=-1;
    total4:=-1;
    total5:=-1;
    total6:=-1;
    total7:=-1;
    total8:=-1;

    if (electrictype=1.2) then
        begin
            if totalelectric<=35 then pay1:=85.21
            else total2:=-totalelectric-35;
            if (total2<=115) and (total2>=0) then pay1:=85.21+(total2*1.1236)
            else total3:=-total2-115;
            if (total3<=250) and (total3>=0) then pay1:=85.21+(115*1.1236)+
(total3*2.1329)
            else total4:=-total3-250;
            if total4>=0 then pay1:=85.21+(115*1.1236)+(250*2.1329)+(total4*2.4226);

        end

    else if (electrictype=1.1) then
        begin
            if totalelectric<=5 then pay1:=-4.96
            else total2:=-totalelectric-5;

            if (total2<=10)and (total2>=0) then pay1:=4.96+(total2*0.7124)
            else total3:=-total2-10;

            if (total3<=10) and (total3>=0)then pay1:=4.96+(10*0.7124)+(total3*0.8993)
            else total4:=-total3-10;

            if (total4<=10) and (total4>=0) then pay1:=4.96+(10*0.7124)+(10*0.8993)+
(total4*1.1516)
            else total5:=-total4-10;

            if (total5<=65) and (total5>=0) then pay1:=4.96+(10*0.7124)+(10*0.8993)+
(10*1.1516)+(total5*1.5348)
            else total6:=-total5-65;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (total6<=50) and (total6>=0) then pay1:=4.96+(10*0.7124)+(10*0.8993)+
(10*1.1516)+(65*1.5348)+(total6*1.6282)
      else total7:=total6-50;

    if (total7<=250) and (total7>=0) then pay1:=4.96+(10*0.7124)+(10*0.8993)+
(10*1.1516)+(65*1.5348)+(50*1.6282)+(total7*2.1329)
      else total8:=total7-250;

    if total8>=0 then pay1:=4.96+(10*0.7124)+(10*0.8993)+(10*1.1516)+
(65*1.5348)+(50*1.6282)+(250*2.1329)+(total8*2.4226);
    end

  else if electrictype=2 then
    begin
      if totalelectric<=35 then pay1:=89.89
        else total2:=totalelectric-35;
      if (total2<=115) and (total2>=0) then pay1:=89.89+(total2*1.1236)
        else total3:=total2-115;
      if (total3<=250) and (total3>=0) then pay1:=89.89+(115*1.1236)+
(total3*2.1329)
        else total4:=total3-250;
      if total4>=0 then pay1:=89.89+(115*1.1236)+(250*2.1329)+(total4*2.4226);
    end
  else showmessage('not correct electric');

  ft:=55.77/100; //factor
  payft:=totalelectric*ft;
  tax:=(pay1+payft)/10; // tax 10%
  pay:=pay1+payft+tax;
end;

procedure TForm3.Table2CalcFields(DataSet: TDataSet);
var
  totalelectric:integer;
  thisunit,lastunit:integer;
  payelectric:real;
  typeelectric:variant;
begin
  thisunit:=table2.fieldbyname('electricunit').asinteger;
  lastunit:= table2.fieldbyname('oldelectricunit').asinteger;
  table2.FieldName('totalelectricunit').asinteger:=thisunit-lastunit;
  totalelectric:=table2.fieldbyname('totalelectricunit').asinteger;
  typeelectric:=table2.fieldbyname('typeelectric').asvariant;
  electricpay(typeelectric,totalelectric,payelectric);
  table2.FieldName('electricpay').asfloat:=payelectric;
end;

procedure TForm3.BitBtn5Click(Sender: TObject);
var InputString: string;
begin
  if messagedlg('คุณต้องการที่จะเปลี่ยนรหัสผ่านใช่หรือไม่',mtconfirmation,[mbyes,mbno],0)=mryes
  then
    begin
      InputString:= InputBox('เปลี่ยนรหัสผ่าน','ป้อนรหัสผ่านใหม่','');
      edit1.text:=inputstring;
      if edit1.text='' then exit;
      InputString:= InputBox('เปลี่ยนรหัสผ่าน','ป้อนรหัสผ่านใหม่อีกครั้ง','');
      if inputstring=edit1.text then
        begin
          table8.Open;
          table8.Edit;

          table8.insert;
          table8.fieldbyname('password1').asstring:=inputstring;

          table8.last;
          table8.delete;

          messagedlg('การเปลี่ยนรหัสผ่านเรียบร้อยแล้ว',mtinformation,[mbok],0);
        end;
    end;
end;

```

```

        end
    else
        begin
            showMessage('ป้อนรหัสผ่านครั้งใหม่ไม่ถูกต้อง');
        end
    end;

end;

end;
procedure TForm3.BitBtn6Click(Sender: TObject);
begin
    form3.hide;
    form12.show;
end;

procedure TForm3.Table5CalcFields(DataSet: TDataSet);
var
    totalwater:integer;
    thisunit,lastunit:integer;
    paywater:real;
    watertype:variant;
begin
    thisunit:=table5.fieldbyname('waterunit').asinteger;
    lastunit:= table5.fieldbyname('oldwaterunit').asinteger;
    table5.FieldByName('totalwaterunit').asinteger:=thisunit-lastunit;
    totalwater:=table5.fieldbyname('totalwaterunit').asinteger;
    watertype:=table5.fieldbyname('typewater').asvariant;
    waterpay(watertype,totalwater,paywater);
    table5.FieldByName('waterpay').asfloat:=paywater;
end;

procedure TForm3.BitBtn7Click(Sender: TObject);
begin
    form3.hide;
    form14.show;
end;

end.

unit portable4;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, ComDrv32, ExtCtrls, Db, DBTables, Grids, DBGrids, Calendar,
    DBCtrls, Gauges, ComCtrls, Buttons;

type
    TForm4 = class(TForm)
        CommPortDriver1: TCommPortDriver;
        Timer1: TTimer;
        Table1: TTable;
        DataSource1: TDataSource;
        Calendar1: TCalendar;
        Table2: TTable;
        DataSource2: TDataSource;
        Table3: TTable;
        DataSource3: TDataSource;
        BitBtn1: TBitBtn;
        Connect: TBitBtn;
        GroupBox1: TGroupBox;
        DBGrid1: TDBGrid;
        procedure ConnectClick(Sender: TObject);
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเขียนเพื่อใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure DisconnectClick(Sender: TObject);
procedure CommPortDriver1ReceiveData(Sender: TObject; DataPtr: Pointer;
  DataSize: Integer);
procedure Timer1Timer(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
private
  procedure ApplyCommSettings;
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;

implementation

uses portable2, portable3;

{$R *.DFM}

procedure TForm4.ApplyCommSettings;

var wasConnected: boolean;
begin
  wasConnected := CommPortDriver1.Connected;
  // This change needs CommPortDriver not connected
  if wasConnected then
    //DisconnectClick( nil );
    CommPortDriver1.ComPort := TComPortNumber((pnCOM1));
  //Reconnect
  if wasConnected then
    ConnectClick( nil );

end;

procedure TForm4.ConnectClick(Sender: TObject);

type TArraystr=array[1..19] of string;
var s:String;
    data:tarraystr;

begin
  data[17]:='0';
  //Set BaudRate
  CommPortDriver1.ComPortSpeed :=br9600;
  //Define Data Bit Format.
  CommPortDriver1.ComPortDataBits :=db8BITS;
  //Define parity bit
  CommPortDriver1.ComPortParity := ptNONE;
  CommPortDriver1.ComPortHwHandshaking := hhNone;
  CommPortDriver1.ComPortSwHandshaking := shNone;
  // Apply settings to comport.
  ApplyCommSettings;
  //If Connect function return true.
  if CommPortDriver1.Connect then
    begin
    //Connect.Enabled := false;
    // Disconnect.Enabled := true;
    end
  else // Error! on connect.
    begin
    MessageBeep( 0 );
    end;

  if CommPortDriver1.Connected then
    begin
    //Send started character ('Enter') to Mcs-51
    s :=#13;
    CommPortDriver1.SendData( pchar(s), length(s) );
    CommPortDriver1.SendData( @s, 1 );
    end;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm4.CommPortDriver1ReceiveData(Sender: TObject;
  DataPtr: Pointer; DataSize: Integer);

type TArraystr=array[1..19] of string;
var p: pchar;
    i:integer;
    s: string;
    a:string;

    oldday,oldmonth,oldyear:string;
    newday,newmonth,newyear:integer;
    oldelectricno,oldelectricunit:string;
    newelectricno:string;
    newelectricunit:integer;
    data:TArraystr;

begin
  i:=1;
  p := DataPtr;
  while DataSize > 0 do
    begin
      a:=(p^);
      data[i]:=a;
      inc(i);
      s := s + p^;
      dec( DataSize );
      inc( p );
    end;

    table1.open;
    Table1.insert;
    oldday:=data[1]+data[2];
    newday:=strtoint(oldday);

    oldmonth:=data[3]+data[4];
    newmonth:=strtoint(oldmonth);

    oldyear:=data[5]+data[6];
    newyear:=strtoint(oldyear);

    oldelectricno:=data[7]+data[8]+data[9]+data[10]+data[11]+data[12];
    newelectricno:=(oldelectricno);

    oldelectricunit:=data[13]+data[14]+data[15]+data[16];
    newelectricunit:=strtoint(oldelectricunit);

    Table1.FieldByName('day1').asinteger :=newday;
    Table1.FieldByName('month1').asinteger :=newmonth;
    Table1.FieldByName('year1').asinteger :=newyear;
    Table1.FieldByName('electricno').asstring :=newelectricno;
    Table1.FieldByName('electricunit').asinteger :=newelectricunit;
    Table1.Next;

    if data[17]='B' then
      begin
        timer1.enabled:=false;
        showmessage('รับข้อมูลเรียบร้อยแล้ว');

        end
      else
        begin
          Timer1.Enabled:=True;
        end ;
    end;

  procedure TForm4.DisconnectClick(Sender: TObject);
  begin
    //CommPortDriver1.Disconnect;
    //Disconnect.Enabled := false;
    // Connect.Enabled := true;
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm4.Timer1Timer(Sender: TObject);
var s:String;
begin
  //Send started character ('A') to Mcs-51
  s :=#65;
  CommPortDriver1.SendData( pchar(s), length(s) );
  CommPortDriver1.SendData( @s, 1 );
end;

procedure TForm4.BitBtn1Click(Sender: TObject);
var i,a,lastmonth:integer;
begin
  lastmonth:=calendar1.month-1;
  if lastmonth=0 then lastmonth:=12;

  table2.open;
  table2.first;
  i:=table2.recordcount;
  a:=0;

  while a<i do
  begin
    table2.delete;
    table2.next;
    a:=a+1;
  end;

  table3.open;
  table3.first;
  i:=table3.recordcount;
  a:=0;

  while a<i do
  begin
    table3.delete;
    table3.next;
    a:=a+1;
  end;

  table1.open;
  table1.setkey;
  table1.first;
  i:=table1.recordcount;
  a:=0;

  while a<i do
  begin
    table2.edit;
    table2.insert;
    table3.edit;
    table3.insert;
    if table1.fieldbyname('month1').asinteger=calendar1.Month then
      begin
        table1.gotokey;
        table2.FieldName('electricno').asString:=table1.fieldbyname
('electricno').asString;
        table2.FieldName('electricunit').asinteger:=table1.fieldbyname
('electricunit').asinteger;
        table2.FieldName('day1').asinteger:=table1.fieldbyname
('day1').asinteger;
        table2.FieldName('month1').asinteger:=table1.fieldbyname
('month1').asinteger;
        table2.FieldName('year1').asinteger:=table1.fieldbyname
('year1').asinteger;
        table1.next;
        table2.next;
        a:=a+1;
      end
    else if table1.fieldbyname('month').asinteger=lastmonth then
      begin
        table1.gotokey;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        table3.FieldName('electricno').asString:=table1.fieldbyname
('electricno').asString;
        table3.FieldName('electricunit').asinteger:=table1.fieldbyname
('electricunit').asinteger;
        table3.FieldName('day1').asinteger:=table1.fieldbyname
('day1').asinteger;
        table3.FieldName('month1').asinteger:=table1.fieldbyname
('month1').asinteger;
        table3.FieldName('year1').asinteger:=table1.fieldbyname
('year1').asinteger;
        table1.next;
        table3.next;
        a:=a+1;
        end ;
end ;

connect.enabled:=true;

{
    table1.first;
    i:=table1.recordcount;
    a:=0;

    while a<i do
    begin
        table1.delete;
        table1.next;
        a:=a+1;
    end; }

form4.hide;
form3.show;
end;

end.

unit portable5;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Grids, DBGrids, Db, DBTables, ExtCtrls, DBCtrls, StdCtrls, Mask, Buttons,
    ComCtrls, quickrpt, QrCtrls;

type
    TForm5 = class(TForm)
        StatusBar1: TStatusBar;
        QuickRep1: TQuickRep;
        QuickRep2: TQuickRep;
        DetailBand1: TQRBand;
        PageHeaderBand1: TQRBand;
        QRDBText1: TQRDBText;
        QRDBText2: TQRDBText;
        QRDBText3: TQRDBText;
        QRDBText4: TQRDBText;
        QRDBText5: TQRDBText;
        QRDBText6: TQRDBText;
        QRDBText7: TQRDBText;
        QRDBText8: TQRDBText;
        QRDBText9: TQRDBText;
        QRDBText10: TQRDBText;
        QRDBText11: TQRDBText;
        QRDBText12: TQRDBText;
        QRDBText13: TQRDBText;
        QRDBText14: TQRDBText;
        QRSysData1: TQRSysData;
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form5: TForm5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

implementation

uses portable3, portable6, portable7;

{SR *.DFM}

end.

unit portable6;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls, Buttons, Db, DBTables, Grids, DBGrids, Mask, DBCtrls, ComCtrls;

type

TForm6 = class(TForm)

 Panell: TPanel;

 RadioGroup1: TRadioGroup;

 BitBtn1: TBitBtn;

 BitBtn2: TBitBtn;

 BitBtn3: TBitBtn;

 Edit1: TEdit;

 Label1: TLabel;

 DataSource1: TDataSource;

 Table1: TTable;

 Panel2: TPanel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Label6: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 DBEdit1: TDBEdit;

 DBEdit3: TDBEdit;

 DBEdit2: TDBEdit;

 DBEdit4: TDBEdit;

 DBEdit5: TDBEdit;

 DBEdit6: TDBEdit;

 DBEdit8: TDBEdit;

 StatusBar1: TStatusBar;

 BitBtn5: TBitBtn;

 BitBtn7: TBitBtn;

 BitBtn8: TBitBtn;

 BitBtn9: TBitBtn;

 BitBtn10: TBitBtn;

 DBEdit9: TDBEdit;

 Label10: TLabel;

 BitBtn4: TBitBtn;

 procedure BitBtn1Click(Sender: TObject);

 procedure FormCreate(Sender: TObject);

 procedure BitBtn2Click(Sender: TObject);

 procedure BitBtn5Click(Sender: TObject);

 procedure BitBtn8Click(Sender: TObject);

 procedure BitBtn9Click(Sender: TObject);

 procedure BitBtn10Click(Sender: TObject);

 procedure FormClose(Sender: TObject; var Action: TCloseAction);

 procedure BitBtn7Click(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form6: TForm6;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

implementation

uses portable5, portable7, portable8, portable3;

{SR *.DFM}

```

procedure TForm6.BitBtn1Click(Sender: TObject);
type sseart=(stfirstname,stlastname,stelectricno,stwaterno);
var s:string;
    st:ssseart;
    sflag1,sflag2,sflag3,sflag4:boolean;
begin
    panel2.visible:=false;

    s:=edit1.text;
    if s='' then
    begin
        messagedlg('กรุณาใส่ข้อความที่ต้องการค้นหา',mterror,[mbok].0);
        edit1.setfocus;
    end
    else
    begin
        if radiogroup1.itemindex=0 then
        begin
            table1.indexname='nameindex';
            st:=stfirstname;
        end
        else if radiogroup1.itemindex=1 then
        begin
            table1.indexname='surnameindex';
            st:=stlastname;
        end
        else if radiogroup1.itemindex=2 then
        begin
            table1.indexname='electricnoindex';
            st:=stelectricno;
        end
        else if radiogroup1.itemindex=3 then
        begin
            table1.indexname='waternoindex';
            st:=stwaterno;
        end;

        table1.open;
        table1.refresh;

        table1.first;
        table1.edit;

        table1.setkey;

        case st of
            stfirstname:table1.fieldbyname('name').asstring:=s;
            stlastname:table1.fieldbyname('surname').asstring:=s;
            stelectricno:table1.fieldbyname('electricno').asstring:=s;
            stwaterno:table1.fieldbyname('waterno').asstring:=s;
        end;

        table1.gotokey;
        if s=table1.fieldbyname('name').asstring then sflag1:=true;
        if s=table1.fieldbyname('surname').asstring then sflag2:=true;
        if s=inttostr(table1.fieldbyname('electricno').asinteger) then
sflag3:=true;
        if s=inttostr(table1.fieldbyname('waterno').asinteger) then
sflag4:=true;

        if sflag1 OR sflag2 or sflag3 or sflag4 then
        begin
            table1.gotokey;
            panel1.visible:=false;
            panel2.visible:=true;
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
begin
    messagedlg('ไม่มีข้อมูลในฐานข้อมูล',mterror,[mbok],0);
    panel1.visible:=true;
    panel2.visible:=false;
    edit1.setfocus;
    edit1.text:='';
end;

end;
// table1.close;
end;

procedure TForm6.FormCreate(Sender: TObject);
begin
    panel2.visible:=false;
    panel1.visible:=true;
    // bitbtn5.visible:=false;
    // bitbtn6.visible:=false;
end;

procedure TForm6.BitBtn2Click(Sender: TObject);
begin
    panel2.visible:=false;
    panel1.visible:=true;
    edit1.setfocus;
    edit1.clear;
end;

procedure TForm6.BitBtn5Click(Sender: TObject);
begin
    dbedit1.readonly:=false;
    dbedit2.readonly:=false;
    dbedit3.readonly:=false;
    dbedit4.readonly:=false;
    dbedit5.readonly:=false;
    dbedit6.readonly:=false;
    dbedit7.readonly:=false;
    dbedit8.readonly:=false;
    statusBar1.simpletext:='ใหม่คนแก้ไขข้อมูล';
    dbedit1.setfocus;
end;

procedure TForm6.BitBtn8Click(Sender: TObject);
begin
    form6.hide;
    form8.show;
end;

procedure TForm6.BitBtn9Click(Sender: TObject);
var sflag1,sflag2,sflag3,sflag4:boolean;
begin
    table1.open;
    table1.next;

    if table1.fieldbyname('name').asstring = edit1.text then sflag1:=true;
    if table1.fieldbyname('surname').asstring = edit1.text then sflag2:=true;
    if intostr(table1.fieldbyname('electricno').asinteger) = edit1.text then
sflag3:=true;
    if intostr(table1.fieldbyname('waterno').asinteger) = edit1.text then
sflag4:=true;

    if sflag1 and sflag2 or sflag3 or sflag4 then
        exit

    else
    begin
        messagedlg('ข้อมูลสุดท้าย',mtwarning,[mbok],0);
        bitbtn9.enabled:=false;
    end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;

procedure TForm6.BitBtn10Click(Sender: TObject);
begin
  bitbtn9.enabled:=true;
  panel2.visible:=false;
  panel1.visible:=true;
  edit1.text:='';
  edit1.setfocus;

end;

procedure TForm6.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  panel2.visible:=false;
end;

procedure TForm6.BitBtn7Click(Sender: TObject);
var sflag1,sflag2,sflag3,sflag4:boolean;
begin
  table1.open;
  table1.prior;

  if table1.fieldbyname('name').asstring = edit1.text then sflag1:=true;
  if table1.fieldbyname('surname').asstring = edit1.text then sflag2:=true;
  if intostr(table1.fieldbyname('electricno').asinteger) = edit1.text then
sflag3:=true;
  if intostr(table1.fieldbyname('waterno').asinteger) = edit1.text then
sflag4:=true;

  if sflag1 and sflag2 or sflag3 or sflag4 then
    exit

  else
    begin
      messagedlg('ข้อมูลสุดท้าย',mtwarning,[mbok].0);
      bitbtn9.enabled:=true;
      end;

end;

end.

```

```
unit portable7;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, Mask, DBCtrls, Db, DBTables, Qrctrls, quickrpt, ExtCtrls,
Grids, DBGrids, Buttons, qrprntr, Calendar;
```

```
type
```

```
TForm7 = class(TForm)
  Panel1: TPanel;
  BitBtn2: TBitBtn;
  BitBtn4: TBitBtn;
  BitBtn1: TBitBtn;
  Table1: TTable;
  DataSource1: TDataSource;
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
private
  { Private declarations }
public
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    { Public declarations }
end;

var
    Form7: TForm7;

implementation

uses portable5, portable9, portable3;

{$R *.DFM}

procedure TForm7.BitBtn1Click(Sender: TObject);
var i:integer;
    s:string;
begin
    form9.QuickRep1.print;
    i:=form3.table6.recordcount;
    s:=inttostr(i);
    showmessage('พิมพ์ข้อมูลทั้งสิ้น '+s+' คน');

    showmessage('คุณารชลักรู ก้าลังจัดการกรพิมพ์');
end;

procedure TForm7.BitBtn4Click(Sender: TObject);
var s:string;
    i,count:integer;
begin
    form3.table6.open;
    form3.table6.edit;
    i:=form3.table6.recordcount;
    s:=inttostr(i);
    showmessage('table6 have '+s+'before delete');
    form3.table6.first;
    count:=0;

    while count<i do
    begin
        form3.table6.delete;
        form3.table6.next;
        count:=count+1;
    end;

    form3.table6.open;
    showmessage('table6 have '+inttostr(form3.table6.recordcount)+'after delete');
    //table1.open;
    //showmessage('table1 have '+inttostr(table1.recordcount)+'after delete');
    form7.hide;
    form3.show;
end;

end.

unit portable8;

interface

uses

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    Db, DBTables, StdCtrls, Buttons, ComCtrls;

type
    TForm8 = class(TForm)
        GroupBox1: TGroupBox;
        Edit1: TEdit;
        Edit2: TEdit;
        Edit3: TEdit;
        Edit4: TEdit;
        Edit5: TEdit;
        Edit6: TEdit;
        Edit7: TEdit;
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
Table1: TTable;
DataSource1: TDataSource;
StatusBar1: TStatusBar;
Label9: TLabel;
Edit9: TEdit;
BitBtn6: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
  Form8: TForm8;

implementation

uses portable5, portable6, portable3 ,portable11;

{$R *.DFM}

procedure TForm8.BitBtn1Click(Sender: TObject);
begin
  groupbox1.enabled:=true;
  edit1.setfocus;
  table1.open;
  statusBar1.simpletext:='โหมดเพิ่มข้อมูลผู้ใช้';

end;

procedure TForm8.BitBtn2Click(Sender: TObject);

var sflag1,sflag2,sflag3,sflag4:boolean;
begin
  if edit7.text ='' then sflag1:=true;

  if length(edit7.text)<>6 then sflag3:=true;

  if sflag1 and sflag2 then
  begin
  table1.close;
  messagedlg('กรุณานำป้อนหมายเลขมิเตอร์ไฟฟ้าและน้ำประปา',mterror,[mbok],0);
  edit7.setfocus;
  end
  else
  begin
  if sflag3 and sflag4 then
  begin
  messagedlg('กรุณานำป้อนหมายเลขมิเตอร์ไฟฟ้า 6 หลัก มิเตอร์น้ำประปา 7 หลัก',mterror,[mbok],0);
  exit;
  end
  else
  table1.open;
  table1.insert;
  table1.FieldName('name').asString:=edit1.text;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผูกพันให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

table1.FieldName('surname').asString:=edit2.text;
table1.FieldName('addr1').asString:=edit3.text;
table1.FieldName('province').asString:=edit4.text;
table1.FieldName('zip').asString:=edit5.text;
table1.FieldName('telephone').asString:=edit6.text;
table1.FieldName('electricno').asinteger:=strtoint(edit7.text);

table1.FieldName('typeelectric').asfloat:=strtofloat(edit9.text);

table1.next;
table1.close;
showmessage('บันทึกเรียบร้อยแล้ว');
edit1.text:='';
edit2.text:='';
edit3.text:='';
edit4.text:='';
edit5.text:='';
edit6.text:='';
edit7.text:='';

edit9.text:='';

```

```

end;
table1.close;
end;

```

```

procedure TForm8.BitBtn3Click(Sender: TObject);
begin
    edit1.text:='';
    edit2.text:='';
    edit3.text:='';
    edit4.text:='';
    edit5.text:='';
    edit6.text:='';
    edit7.text:='';
    edit9.text:='';
    edit1.SetFocus;

```

```

end;

```

```

procedure TForm8.BitBtn4Click(Sender: TObject);
begin
    form5.hide;
    form6.show;

```

```

end;

```

```

procedure TForm8.BitBtn5Click(Sender: TObject);
begin
    form8.hide;
    form3.show;

```

```

end;

```

```

procedure TForm8.BitBtn6Click(Sender: TObject);
begin
    form8.hide;
    form11.show;

```

```

end;

```

```

end.

```

```

unit portable9;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Db, ExtCtrls, quickrpt, DBTables, StdCtrls, Buttons, Qrctrls, Grids,

Calendar;

type

```
TForm9 = class(TForm)
  QuickRep1: TQuickRep;
  DetailBand1: TQRBand;
  QRDBText1: TQRDBText;
  QRDBText2: TQRDBText;
  QRDBText3: TQRDBText;
  QRDBText4: TQRDBText;
  QRDBText5: TQRDBText;
  QRDBText6: TQRDBText;
  QRDBText7: TQRDBText;
  procedure QuickRep1BeforePrint(Sender: TQuickRep;
    var PrintReport: Boolean);
```

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

```
var
  Form9: TForm9;
```

implementation

uses portable3;

```
{$R *.DFM}
```

```
procedure TForm9.QuickRep1BeforePrint(Sender: TQuickRep;
  var PrintReport: Boolean);
```

```
var s:string;
```

```
    i:integer;
```

```
begin
```

```
    i:=form3.table6.recordcount;
```

```
    s:=inttostr(i);
```

```
    showmessage('พิมพ์ทั้งสิ้น '+s+' รายการ');
```

```
end;
```

```
end.
```

```
unit portable10;
```

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

ExtCtrls, quickrpt, Qrctrls;

type

```
TForm10 = class(TForm)
```

```
  QuickRep1: TQuickRep;
```

```
  DetailBand1: TQRBand;
```

```
  QRDBText1: TQRDBText;
```

```
  QRDBText2: TQRDBText;
```

```
  QRDBText3: TQRDBText;
```

```
  QRDBText4: TQRDBText;
```

```
  QRDBText5: TQRDBText;
```

```
  QRDBText6: TQRDBText;
```

```
  QRDBText7: TQRDBText;
```

```
  QRDBText8: TQRDBText;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    QRDBText9: TQRDBText;
    QRDBText10: TQRDBText;
    PageHeaderBand1: TQRBand;
    QRSysData1: TQRSysData;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form10: TForm10;

implementation
uses portable3;

{$R *.DFM}

end.

unit portable11;

interface

uses

  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Db, DBTables, ExtCtrls, DBCtrls, Grids, DBGrids, Mask;

type
  TForm11 = class(TForm)
    Query1: TQuery;
    DataSource1: TDataSource;
    Panel1: TPanel;
    RadioGroup1: TRadioGroup;
    Edit1: TEdit;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    Query2: TQuery;
    Query3: TQuery;
    DataSource2: TDataSource;
    DataSource3: TDataSource;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    DBGrid3: TDBGrid;
    Query5: TQuery;
    DBGrid5: TDBGrid;
    DataSource5: TDataSource;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form11: TForm11;

implementation
uses portable8;

{$R *.DFM}

procedure TForm11.BitBtn1Click(Sender: TObject);
begin
if radiogroup1.itemindex=0 then
begin
  with query1 do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end.

unit portable12;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Db, DBTables, ExtCtrls, ComDrv32, StdCtrls, Buttons, Grids, Calendar,
DBGrids;

type

```
TForm12 = class(TForm)
  connect: TBitBtn;
  BitBtn2: TBitBtn;
  CommPortDriver1: TCommPortDriver;
  Timer1: TTimer;
  Table1: TTable;
  DataSource1: TDataSource;
  Table2: TTable;
  DataSource2: TDataSource;
  Table3: TTable;
  DataSource3: TDataSource;
  Calendar1: TCalendar;
  GroupBox1: TGroupBox;
  DBGrid1: TDBGrid;
  procedure connectClick(Sender: TObject);
  procedure CommPortDriver1ReceiveData(Sender: TObject; DataPtr: Pointer;
    DataSize: Integer);
  procedure Timer1Timer(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
private
  procedure ApplyCommSettings;
  { Private declarations }
public
  { Public declarations }
end;
```

var

Form12: TForm12;

implementation

uses portable4, portable3;

{SR *.DFM}

```
procedure TForm12.ApplyCommSettings;
var wasConnected: boolean;
begin
  wasConnected := CommPortDriver1.Connected;
  // This change needs CommPortDriver not connected
  if wasConnected then
    //DisconnectClick( nil );
  CommPortDriver1.ComPort := TComPortNumber((pnCOM1));
  //Reconnect
  if wasConnected then
    ConnectClick( nil );
end;
```

```
procedure TForm12.connectClick(Sender: TObject);
```

```
type TArraystr=array[1..19] of string;
```

```
var s:String;
    data:tarraystr;
```

```
begin
```

```
  data[18]:='-0';
```

```
  //Set BaudRate
```

```
  CommPortDriver1.ComPortSpeed :=br9600;
```

```
  //Define Data Bit Format.
```

```
  CommPortDriver1.ComPortDataBits :=db8BITS;
```

```
  //Define parity bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CommPortDriver1.ComPortParity := ptNONE;
CommPortDriver1.ComPortHwHandshaking := hhNone;
CommPortDriver1.ComPortSwHandshaking := shNone;
// Apply settings to comport.
ApplyCommSettings;
//If Connect function return true.
if CommPortDriver1.Connect then
begin
//Connect.Enabled := false;
// Disconnect.Enabled := true;
end
else // Error! on connect.
begin
MessageBeep( 0 );
end;

if CommPortDriver1.Connected then
begin
//Send started character ('Enter') to Mcs-51
s :=#13;
CommPortDriver1.SendData( pchar(s), length(s) );
CommPortDriver1.SendData( @s, 1 );
end;
end;

procedure TForm12.CommPortDriver1ReceiveData(Sender: TObject;
DataPtr: Pointer; DataSize: Integer);

type
TArraystr=array[1..19] of string;
var p: pchar;
i:integer;
s: string;
a,oldwaterno,oldwaterunit:string;
oldday,oldmonth,oldyear:string;
newday,newmonth,newyear:integer;
newwaterno,newelecno:string;
newelecunit,newwaterunit:integer;
data:TArraystr;

begin
i:=1;
p := DataPtr;
while DataSize > 0 do
begin
a:=(p^);
data[i]:=a;
inc(i);
s := s + p^;
dec( DataSize );
inc( p );
end;

table1.open;
Table1.insert;
oldday:=data[1]+data[2];
newday:=strtoint(oldday);
oldmonth:=data[3]+data[4];
newmonth:=strtoint(oldmonth);
oldyear:=data[5]+data[6];
newyear:=strtoint(oldyear);
oldwaterno:=data[7]+data[8]+data[9]+data[10]+data[11]+data[12]+data[13];
newwaterno:=(oldwaterno);
oldwaterunit:=data[14]+data[15]+data[16]+data[17];
newwaterunit:=strtoint(oldwaterunit);

Table1.FieldName('day1').asinteger :=newday;
Table1.FieldName('month1').asinteger :=newmonth;
Table1.FieldName('year1').asinteger :=newyear;
Table1.FieldName('waterno').asstring :=newwaterno;
Table1.FieldName('waterunit').asinteger :=newwaterunit;
Table1.Next;

if data[18]='B' then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    timer1.enabled:=false;
    showmessage('รับข้อมูลเรียบร้อยแล้ว');

    end
else
begin

    Timer1.Enabled:=True;
end ;

end;

procedure TForm12.Timer1Timer(Sender: TObject);

var s:String;
begin
    //Send started character ('A') to Mcs-51
    s :=#65;
    CommPortDriver1.SendData( pchar(s), length(s) );
    CommPortDriver1.SendData( @s, 1 );
end;

procedure TForm12.BitBtn2Click(Sender: TObject);
var i,a,lastmonth:integer;
begin

    lastmonth:=calendar1.month-1;
    if lastmonth=0 then lastmonth:=12;

    table2.open;
    table2.first;
    i:=table2.recordcount;
    a:=0;

    while a<i do
    begin
        table2.delete;
        table2.next;
        a:=a+1;
    end;

    table3.open;
    table3.first;
    i:=table3.recordcount;
    a:=0;

    while a<i do
    begin
        table3.delete;
        table3.next;
        a:=a+1;
    end;

    table1.open;
    table1.setkey;
    table1.first;
    i:=table1.recordcount;
    a:=0;

    while a<i do
    begin
        table2.edit;
        table2.insert;
        table3.edit;
        table3.insert;
        if table1.fieldbyname('month1').asinteger=calendar1.Month then
        begin
            table1.gotokey;
            table2.fieldbyname('waterno').asstring:=table1.fieldbyname
('waterno').asstring;
            table2.FieldName('waterunit').asinteger:=table1.fieldbyname
('waterunit').asinteger;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        table2.FieldName('day1').asinteger:=table1.fieldbyname
('day1').asinteger;
        table2.FieldName('month1').asinteger:=table1.fieldbyname
('month1').asinteger;
        table2.FieldName('year1').asinteger:=table1.fieldbyname
('year1').asinteger;
        table1.next;
        table2.next;
        a:=a+1;
        end
    else if table1.fieldbyname('month1').asinteger=lastmonth then
begin
    table1.gotokey;
    table3.fieldbyname('waterno').asString:=table1.fieldbyname
('waterno').asString;
    table3.FieldName('waterunit').asinteger:=table1.fieldbyname
('waterunit').asinteger;
    table3.FieldName('day1').asinteger:=table1.fieldbyname
('day1').asinteger;
    table3.FieldName('month1').asinteger:=table1.fieldbyname
('month1').asinteger;
    table3.FieldName('year1').asinteger:=table1.fieldbyname
('year1').asinteger;
    table1.next;
    table3.next;
    a:=a+1;
    end ;
end ;

```

```
//connect.enabled:=true;
```

```

{
    table1.first;
    i:=table1.recordcount;
    a:=0;

    while a<i do
begin
    table1.delete;
    table1.next;
    a:=a+1;
end;
}

```

```

form12.hide;
form3.show;
end;

```

```
end.
```

```
unit portable13;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Db, DBTables, Mask, DBCtrls, StdCtrls, ExtCtrls, Buttons, ComCtrls;
```

```
type
```

```

TForm13 = class(TForm)
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    RadioGroup1: TRadioGroup;
    Edit1: TEdit;
    Label1: TLabel;
    BitBtn2: TBitBtn;
    BitBtn6: TBitBtn;
    Panel2: TPanel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;

```

```

DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
BitBtn5: TBitBtn;
BitBtn9: TBitBtn;
BitBtn7: TBitBtn;
BitBtn10: TBitBtn;
Table1: TTable;
DataSource1: TDataSource;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
DBEdit8: TDBEdit;
Label9: TLabel;
StatusBar1: TStatusBar;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form13: TForm13;

implementation

uses portable6, portable8, portable14;

{$R *.DFM}

procedure TForm13.BitBtn1Click(Sender: TObject);

type sseart=(stfirstname,stlastname,stwaterno);
var s:string;
    st:sseart;
    sflag1,sflag2,sflag4:boolean;
begin

  panel2.visible:=false;

  s:=edit1.text;
  if s='' then
  begin
    messagedlg('กรุณาใส่ข้อความที่ต้องการค้นหา',mterror,[mbok],0);
    edit1.setfocus;
  end

  else
  begin

    if radiogroup1.itemindex=0 then
    begin
      table1.indexname:='nameindex';
      st:=stfirstname;
    end
    else if radiogroup1.itemindex=1 then
    begin
      table1.indexname:='surnameindex';
      st:=stlastname;
    end
    else if radiogroup1.itemindex=2 then
    begin
      table1.indexname:='waternoindex';
      st:=stwaterno;
    end
  end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    end;

    table1.open;
    table1.refresh;

    table1.first;
    table1.edit;

    table1.setkey;

    case st of
    stfirstname:table1.fieldbyname('name').asstring:=s;
    stlastname:table1.fieldbyname('surname').asstring:=s;

    stwaterno:table1.fieldbyname('waterno').asstring:=s;
    end;

    table1.gotokey;
    if s=table1.fieldbyname('name').asstring then sflag1:=true;
    if s=table1.fieldbyname('surname').asstring then sflag2:=true;

    if s=table1.fieldbyname('waterno').asstring then sflag4:=true;

    if sflag1 OR sflag2 OR sflag4 then
    begin
        table1.gotokey;
        panel1.visible:=false;
        panel2.visible:=true;
    end

    else
    begin
        messagedlg('ไม่มีข้อมูลในฐานข้อมูล',mterror,[mbok],0);
        panel1.visible:=true;
        panel2.visible:=false;
        edit1.setfocus;
        edit1.text:='';
    end;
end;

end;
end;

procedure TForm13.BitBtn9Click(Sender: TObject);
var sflag1,sflag2,sflag4:boolean;
begin
    table1.open;
    table1.next;

    if table1.fieldbyname('name').asstring = edit1.text then sflag1:=true;
    if table1.fieldbyname('surname').asstring = edit1.text then sflag2:=true;

    if intostr(table1.fieldbyname('waterno').asinteger) = edit1.text then
    sflag4:=true;

    if sflag1 and sflag2 or sflag4 then
    exit

    else
    begin
        messagedlg('ข้อมูลสุดท้าย',mtwarning,[mbok],0);
        bitbtn9.enabled:=false;
        end;
end;

end;

procedure TForm13.BitBtn7Click(Sender: TObject);
var sflag1,sflag2,sflag4:boolean;
begin
    table1.open;
    table1.prior;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if table1.fieldbyname('name').asString = edit1.text then sflag1:=true;
if table1.fieldbyname('surname').asString = edit1.text then sflag2:=true;

if intostr(table1.fieldbyname('waterno').asinteger) = edit1.text then
sflag4:=true;

if sflag1 and sflag2 or sflag4 then
exit

else
begin
messagedlg('ข้อมูลสุดท้าย',mtwarning,[mbok],0);
bitbtn9.enabled:=true;
end;

end;

procedure TForm13.BitBtn10Click(Sender: TObject);
begin
bitbtn9.enabled:=true;
panel2.visible:=false;
panel1.visible:=true;
edit1.text:='';
edit1.setfocus;

end;

procedure TForm13.BitBtn2Click(Sender: TObject);
begin
panel2.visible:=false;
panel1.visible:=true;
edit1.setfocus;
edit1.clear;

end;

procedure TForm13.BitBtn8Click(Sender: TObject);
begin
form13.hide;
form14.show;

end;

procedure TForm13.BitBtn5Click(Sender: TObject);
begin
dbedit1.readonly:=false;
dbedit2.readonly:=false;
dbedit3.readonly:=false;
dbedit4.readonly:=false;
dbedit5.readonly:=false;
dbedit6.readonly:=false;
dbedit7.readonly:=false;
dbedit8.readonly:=false;
statusbar1.simpletext:='โหมดแก้ไขข้อมูล';
dbedit1.setfocus;

end;

procedure TForm13.FormCreate(Sender: TObject);
begin
panel2.visible:=false;

end;

end.

unit portable14;

interface

uses

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

ComCtrls, Db, DBTables, StdCtrls, Buttons;

type

TForm14 = class(TForm)

 GroupBox1: TGroupBox;

 Edit1: TEdit;

 Edit2: TEdit;

 Edit3: TEdit;

 Edit4: TEdit;

 Edit5: TEdit;

 Edit6: TEdit;

 Edit7: TEdit;

 Edit9: TEdit;

 BitBtn1: TBitBtn;

 BitBtn2: TBitBtn;

 BitBtn3: TBitBtn;

 BitBtn4: TBitBtn;

 BitBtn5: TBitBtn;

 BitBtn6: TBitBtn;

 Table1: TTable;

 DataSource1: TDataSource;

 StatusBar1: TStatusBar;

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Label4: TLabel;

 Label5: TLabel;

 Label6: TLabel;

 Label7: TLabel;

 Label8: TLabel;

 procedure BitBtn1Click(Sender: TObject);

 procedure BitBtn2Click(Sender: TObject);

 procedure BitBtn3Click(Sender: TObject);

 procedure BitBtn4Click(Sender: TObject);

 procedure BitBtn5Click(Sender: TObject);

 procedure BitBtn6Click(Sender: TObject);

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form14: TForm14;

implementation

 uses portable13, portable3, portable15;

 {\$R *.DFM}

 procedure TForm14.BitBtn1Click(Sender: TObject);

 begin

 groupBox1.enabled:=true;

 edit1.setfocus;

 table1.open;

 statusbar1.simpletext:='ใหม่ตเพิ่มข้อมูลผู้ใช้';

 end;

 procedure TForm14.BitBtn2Click(Sender: TObject);

 var sflag1,sflag2,sflag3,sflag4:boolean;

 begin

 if edit7.text ='' then sflag1:=true;

 if length(edit7.text)<>6 then sflag3:=true;

 if sflag1 and sflag2 then

 begin

 table1.close;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

messagedlg ('กรุณาป้อนหมายเลขมิเตอร์ไฟฟ้าและน้ำประปา', mterror, [mbok], 0);
edit7.setfocus;
end
else
begin
if sflag3 and sflag4 then
begin
    messagedlg ('กรุณาป้อนหมายเลขมิเตอร์ไฟฟ้า 6 หลัก มิเตอร์น้ำประปา 7 หลัก', mterror, [mbok], 0);
    exit;
end
else
table1.open;
table1.insert;
table1.FieldName('name').asString:=edit1.text;

table1.FieldName('surname').asString:=edit2.text;
table1.FieldName('addr1').asString:=edit3.text;
table1.FieldName('province').asString:=edit4.text;
table1.FieldName('zip').asString:=edit5.text;
table1.FieldName('telephone').asString:=edit6.text;
table1.FieldName('electricno').asinteger:=strtoint(edit7.text);

table1.next;
table1.close;
showmessage('บันทึกเรียบร้อยแล้ว');
edit1.text:='';
edit2.text:='';
edit3.text:='';
edit4.text:='';
edit5.text:='';
edit6.text:='';
edit7.text:='';
edit9.text:='';

end;
table1.close;
end;

procedure TForm14.BitBtn3Click(Sender: TObject);
begin
    edit1.text:='';
    edit2.text:='';
    edit3.text:='';
    edit4.text:='';
    edit5.text:='';
    edit6.text:='';
    edit7.text:='';
    edit9.text:='';
    edit1.SetFocus;

end;

procedure TForm14.BitBtn4Click(Sender: TObject);
begin
    form14.hide;
    form13.show;

end;

procedure TForm14.BitBtn5Click(Sender: TObject);
begin
    form14.hide;
    form15.show;

end;

procedure TForm14.BitBtn6Click(Sender: TObject);
begin
    form14.hide;
    form3.show;

end;

end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

program portableproject;
uses
  Forms,
  portable1 in 'portable1.pas' {Form1},
  portable2 in 'portable2.pas' {Form2},
  portable3 in 'portable3.pas' {Form3},
  portable4 in 'portable4.pas' {Form4},
  portable5 in 'portable5.pas' {Form5},
  portable6 in 'portable6.pas' {Form6},
  portable7 in 'portable7.pas' {Form7},
  portable8 in 'portable8.pas' {Form8},
  portable9 in 'portable9.pas' {Form9},
  portable10 in 'portable10.pas' {Form10},
  portable11 in 'portable11.pas' {Form11},
  portable12 in 'portable12.pas' {Form12},
  portable13 in 'portable13.pas' {Form13},
  portable14 in 'portable14.pas' {Form14},
  portable15 in 'portable15.pas' {Form15};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm2, Form2);
  Application.CreateForm(TForm3, Form3);
  Application.CreateForm(TForm4, Form4);
  Application.CreateForm(TForm5, Form5);
  Application.CreateForm(TForm6, Form6);
  Application.CreateForm(TForm7, Form7);
  Application.CreateForm(TForm8, Form8);
  Application.CreateForm(TForm9, Form9);
  Application.CreateForm(TForm10, Form10);
  Application.CreateForm(TForm11, Form11);
  Application.CreateForm(TForm12, Form12);
  Application.CreateForm(TForm13, Form13);
  Application.CreateForm(TForm14, Form14);
  Application.CreateForm(TForm15, Form15);
  Application.Run;
end.

```

กิตติกรรมประกาศ

เนื่องด้วยโครงการนี้จะสำเร็จได้นั้นจำเป็นอย่างยิ่งที่จะต้องได้รับความช่วยเหลือจากท่านอาจารย์ในภาควิชา และบุคคลอื่นๆ ที่ผู้จัดทำได้รับความช่วยเหลือและข้อเสนอแนะต่างๆ ที่เป็นประโยชน์ต่อผู้จัดทำในการจัดทำโครงการนี้ขึ้นมา อาจารย์ สมยศ จุณณะปิยะ อาจารย์ที่ปรึกษา รวมถึงอาจารย์ทุกท่านที่ได้คอยประสิทธิ์ประสาทวิชาให้ และที่ขาดไม่ได้ขอขอบพระคุณเป็นอย่างยิ่ง บิดา และ มารดา ในทุกๆ สิ่งทุกอย่างที่ทำให้เรามาถึงจุดนี้ได้ และท้ายสุดขอขอบคุณเพื่อนๆ ที่คอยให้ข่าวสารและกำลังใจมาโดยตลอด

คณะผู้จัดทำ

1 เมษายน 2542



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ประเมษฐ์ ประนยานันท์ และ ปิยพงศ์ เผ่าวงษ์ , คู่มือและการประยุกต์ใช้งานไมโคร คอนโทรลเลอร์ MCS-51, บริษัท ซีเอ็ดดูเคชั่น, กรุงเทพมหานคร, 2536
2. สมยศ จุณณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พิมพ์ครั้งที่ 2, กรุงเทพมหานคร, 2541
3. ไกรวุฒิ โรจน์ประเสริฐสุด, เข้าใจ/สร้าง/เล่น ไมโครโปรเซสเซอร์ 2 (MCS-51) 80C535, บริษัทซีเอ็ดดูเคชั่น, กรุงเทพมหานคร, 2539



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Microprocessor Supervisory Circuits

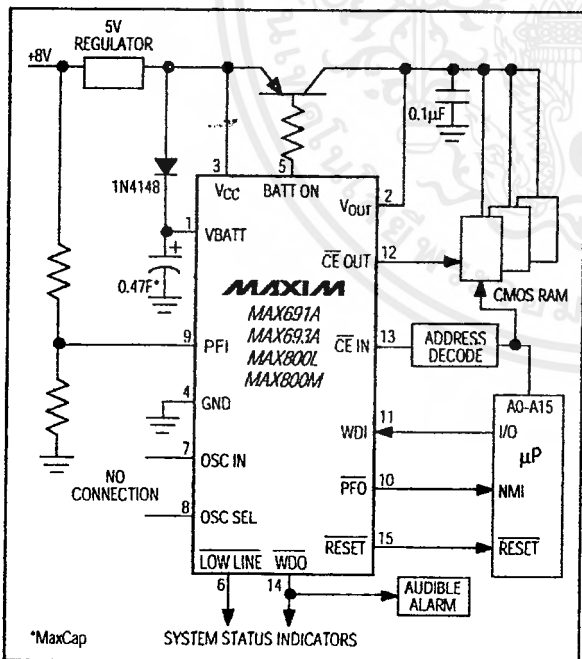
General Description

The MAX691A/MAX693A/MAX800L/MAX800M microprocessor (μP) supervisory circuits are pin-compatible upgrades to the MAX691, MAX693, and MAX695. They improve performance with 30 μA supply current, 200ms typ reset active delay on power-up, and 6ns chip-enable propagation delay. Features include write protection of CMOS RAM or EEPROM, separate watchdog outputs, backup-battery switchover, and a \overline{RESET} output that is valid with V_{CC} down to 1V. The MAX691A/MAX800L have a 4.65V typical reset-threshold voltage, and the MAX693A/MAX800M's reset threshold is 4.4V typical. The MAX800L/MAX800M guarantee power-fail accuracies to $\pm 2\%$.

Applications

- Computers
- Controllers
- Intelligent Instruments
- Automotive Systems
- Critical μP Power Monitoring

Typical Operating Circuit



Features

- ◆ 200ms Power-OK/Reset Timeout Period
- ◆ 1 μA Standby Current, 30 μA Operating Current
- ◆ On-Board Gating of Chip-Enable Signals, 10ns Max Delay
- ◆ MaxCap™ or SuperCap™ Compatible
- ◆ Guaranteed \overline{RESET} Assertion to $V_{CC} = +1V$
- ◆ Voltage Monitor for Power-Fail or Low-Battery Warning
- ◆ Power-Fail Accuracy Guaranteed to $\pm 2\%$ (MAX800L/M)
- ◆ Available in 16-Pin Narrow SO and Plastic DIP Packages

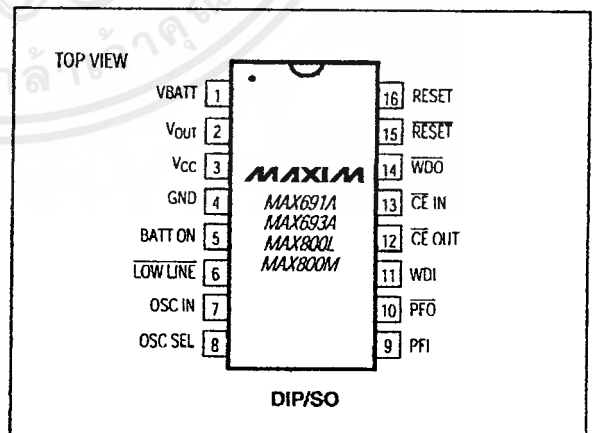
Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX691ACPE	0°C to +70°C	16 Plastic DIP
MAX691ACSE	0°C to +70°C	16 Narrow SO
MAX691ACWE	0°C to +70°C	16 Wide SO
MAX691AC/D	0°C to +70°C	Dice*
MAX691AEPE	-40°C to +85°C	16 Plastic DIP
MAX691AESE	-40°C to +85°C	16 Narrow SO
MAX691AEWE	-40°C to +85°C	16 Wide SO
MAX691ACJE	-40°C to +85°C	16 CCDIP
MAX691AMJE	-55°C to +125°C	16 CERDIP

Ordering Information continued on last page.

* Dice are specified at $T_A = +25^\circ C$, DC parameters only.

Pin Configuration



SuperCap is a registered trademark of Baknor Industries. MaxCap is a registered trademark of The Carborundum Corp.



Maxim Integrated Products 1

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800. For small orders, phone 408-737-7600 ext. 3468.

MAX691A/MAX693A/MAX800L/MAX800M

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

ABSOLUTE MAXIMUM RATINGS

Terminal Voltage (with respect to GND)

V _{CC}	-0.3V to +6V
VBATT	-0.3V to +6V
All Other Inputs	-0.3V to (V _{OUT} + 0.3V)
Input Current	
V _{CC} Peak	1.0A
V _{CC} Continuous	250mA
VBATT Peak	250mA
VBATT Continuous	25mA
GND, BATT ON	100mA
All Other Outputs	25mA

Continuous Power Dissipation (T_A = +70°C)

Plastic DIP (derate 10.53mW/°C above +70°C)	842mW
Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Wide SO (derate 9.52mW/°C above +70°C)	762mW
CERDIP (derate 10.00mW/°C above +70°C)	800mW
Operating Temperature Ranges	
MAX69_AC_/_MAX800_C_	0°C to +70°C
MAX69_AE_/_MAX800_E_	-40°C to +85°C
MAX69_AMJE	-55°C to +125°C
Storage Temperature Range	-65°C to +160°C
Lead Temperature (soldering, 10sec)	+300°C

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

ELECTRICAL CHARACTERISTICS

(MAX691A, MAX800L: V_{CC} = +4.75V to +5.5V, MAX693A, MAX800M: V_{CC} = +4.5V to +5.5V, VBATT = 2.8V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Operating Voltage Range, V _{CC} , VBATT (Note 1)			0		5.5	V
V _{OUT} Output	V _{CC} = 4.5V	I _{OUT} = 25mA		V _{CC} - 0.02	V _{CC} - 0.05	V
		I _{OUT} = 250mA	MAX69_AC	V _{CC} - 0.2	V _{CC} - 0.3	
			MAX69_AE, MAX800_C/E	V _{CC} - 0.2	V _{CC} - 0.35	
			MAX69_A/M		V _{CC} - 0.40	
I _{OUT} = 210mA	MAX69_AC/AE, MAX800_C/E	V _{CC} - 0.17	V _{CC} - 0.3V			
V _{CC} -to-V _{OUT} On-Resistance	V _{CC} = 4.5V	MAX69_AC, MAX800_C		0.8	1.2	Ω
		MAX69_AE, MAX800_E		0.8	1.4	
		MAX69_A/M		0.8	1.6	
V _{OUT} in Battery-Backup Mode	VBATT = 4.5V, I _{OUT} = 20mA		VBATT - 0.3			V
	VBATT = 2.8V, I _{OUT} = 10mA		VBATT - 0.25			
	VBATT = 2.0V, I _{OUT} = 5mA		VBATT - 0.15			
VBATT-to-V _{OUT} On-Resistance	VBATT = 4.5V				15	Ω
	VBATT = 2.8V				25	
	VBATT = 2.0V				30	
Supply Current in Normal Operating Mode (Excludes I _{OUT})	V _{CC} > VBATT - 1V			30	100	μA
Supply Current in Battery-Backup Mode (Excludes I _{OUT}) (Note 2)	V _{CC} < VBATT - 1.2V VBATT = 2.8V	T _A = +25°C		0.04	1	μA
		T _A = T _{MIN} + T _{MIN}			5	
VBATT Standby Current (Note 3)	VBATT + 0.2V ≤ V _{CC}	T _A = +25°C	-0.1		0.02	μA
		T _A = T _{MIN} + T _{MIN}	-1.0		0.02	
Battery Switchover Threshold	Power-up			VBATT + 0.3		V
	Power-down			VBATT - 0.3		

Microprocessor Supervisory Circuits

ELECTRICAL CHARACTERISTICS (continued)

(MAX691A, MAX800L: $V_{CC} = +4.75V$ to $+5.5V$, MAX693A, MAX800M: $V_{CC} = +4.5V$ to $+5.5V$, $V_{BATT} = 2.8V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Battery Switchover Hysteresis			60		mV
BATT ON Output Low Voltage	$I_{SINK} = 3.2mA$		0.1	0.4	V
	$I_{SINK} = 25mA$		0.7	1.5	
BATT ON Output Short-Circuit Current	Sink current		60		mA
	Source current	1	15	100	μA
RESET AND WATCHDOG TIMER					
Reset Threshold Voltage	MAX691A, MAX800L	4.50	4.65	4.75	V
	MAX693A, MAX800M	4.25	4.40	4.50	
	MAX800L, $T_A = +25^\circ C$, V_{CC} falling	4.55		4.70	
	MAX800M, $T_A = +25^\circ C$, V_{CC} falling	4.30		4.45	
Reset Threshold Hysteresis			15		mV
V_{CC} to RESET Delay	Power-down		80		μs
LOW LINE-to-RESET Delay			800		ns
Reset Active Timeout Period, Internal Oscillator	Power-up	140	200	280	ms
Reset Active Timeout Period, External Clock (Note 4)	Power-up		2048		Clock Cycles
Watchdog Timeout Period, Internal Oscillator	Long period	1.0	1.6	2.25	sec
	Short period	70	100	140	
Watchdog Timeout Period, External Clock (Note 4)	Long period		4096		Clock Cycles
	Short period		1024		
Minimum Watchdog Input Pulse Width	$V_{IL} = 0.8V$, $V_{IH} = 0.75 \times V_{CC}$	100			ns
RESET Output Voltage	$I_{SINK} = 50\mu A$, $V_{CC} = 1V$, $V_{BATT} = 0V$, V_{CC} falling		0.004	0.3	V
	$I_{SINK} = 3.2mA$, $V_{CC} = 4.25V$		0.1	0.4	
	$I_{SOURCE} = 1.6mA$, $V_{CC} = 5V$	3.5			
RESET Output Short-Circuit Current	Output source current		7	20	mA
RESET Output Voltage Low (Note 5)	$I_{SINK} = 3.2mA$	0.1	0.4		V
LOW LINE Output Voltage	$I_{SINK} = 3.2mA$, $V_{CC} = 4.25V$			0.4	V
	$I_{SOURCE} = 1\mu A$, $V_{CC} = 5V$	3.5			
LOW LINE Output Short-Circuit Current	Output source current	1	15	100	μA
WDO Output Voltage	$I_{SINK} = 3.2mA$			0.4	V
	$I_{SOURCE} = 500\mu A$, $V_{CC} = 5V$	3.5			
WDO Output Short-Circuit Current	Output source current		3	10	mA
WDI Threshold Voltage (Note 6)	V_{IH}	$0.75 \times V_{CC}$			V
	V_{IL}			0.8	
WDI Input Current	WDI = 0V	-50	-10		μA
	WDI = V_{OUT}		20	50	

Microprocessor Supervisory Circuits

MAX691A/MAX693A/MAX800L/MAX800M

ELECTRICAL CHARACTERISTICS (continued)

(MAX691A, MAX800L: $V_{CC} = +4.75V$ to $+5.5V$, MAX693A, MAX800M: $V_{CC} = +4.5V$ to $+5.5V$, $V_{BATT} = 2.8V$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
POWER-FAIL COMPARATOR					
PFI Input Threshold	MAX69_AC/AE/AM, $V_{CC} = 5V$	1.2	1.25	1.3	V
	MAX800_C/E, $V_{CC} = 5V$	1.225	1.25	1.275	
PFI Leakage Current			± 0.01	± 25	nA
\overline{PFO} Output Voltage	$I_{SINK} = 3.2mA$			0.4	V
	$I_{SOURCE} = 1\mu A$, $V_{CC} = 5V$	3.5			
\overline{PFO} Output Short-Circuit Current	Output source current	1	15	100	μA
PFI-to- \overline{PFO} Delay	$V_{IN} = -20mV$, $V_{OD} = 15mV$		25		μs
	$V_{IN} = 20mV$, $V_{OD} = 15mV$		60		
CHIP-ENABLE GATING					
\overline{CE} IN Leakage Current	Disable mode		± 0.005	± 1	μA
\overline{CE} IN-to- \overline{CE} OUT Resistance (Note 7)	Enable mode		75	150	Ω
\overline{CE} OUT Short-Circuit Current (Reset Active)	Disable mode, \overline{CE} OUT = 0V	0.1	0.75	2.0	mA
\overline{CE} IN-to- \overline{CE} OUT Propagation Delay (Note 8)	50 Ω source impedance driver, $C_{LOAD} = 50pF$		6	10	ns
\overline{CE} OUT Output Voltage High (Reset Active)	$V_{CC} = 5V$, $I_{OUT} = -100\mu A$	3.5			V
	$V_{CC} = 0V$, $V_{BATT} = 2.8V$, $I_{OUT} = 1\mu A$	2.7			
RESET-to- \overline{CE} OUT Delay	Power-down		12		μs
INTERNAL OSCILLATOR					
OSC IN Leakage Current	OSC SEL = 0V		0.10	± 5	μA
OSC IN Input Pull-Up Current	OSC SEL = V_{OUT} or floating, OSC IN = 0V		10	100	μA
OSC SEL Input Pull-Up Current	OSC SEL = 0V		10	100	μA
OSC IN Frequency Range	OSC SEL = 0V		50		kHz
OSC IN External Oscillator Threshold Voltage	V_{IH}	$V_{OUT} - 0.3$	$V_{OUT} - 0.6$		V
	V_{IL}		3.65	2.00	
OSC IN Frequency with External Capacitor	OSC SEL = 0V, $C_{OSC} = 47pF$		100		kHz

Note 1: Either V_{CC} or V_{BATT} can go to 0V, if the other is greater than 2.0V.

Note 2: The supply current drawn by the MAX691A/MAX800L/MAX800M from the battery excluding I_{OUT} typically goes to 10 μA when $(V_{BATT} - 1V) < V_{CC} < V_{BATT}$. In most applications, this is a brief period as V_{CC} falls through this region.

Note 3: "+" = battery-discharging current, "-" = battery-charging current.

Note 4: Although presented as typical values, the number of clock cycles for the reset and watchdog timeout periods are fixed and do not vary with process or temperature.

Note 5: RESET is an open-drain output and sinks current only.

Note 6: WDI is internally connected to a voltage divider between V_{OUT} and GND. If unconnected, WDI is driven to 1.6V (typ), disabling the watchdog function.

Note 7: The chip-enable resistance is tested with $V_{CC} = +4.75V$ for the MAX691A/MAX800L and $V_{CC} = +4.5V$ for the MAX693A/MAX800M. \overline{CE} IN = \overline{CE} OUT = $V_{CC} / 2$.

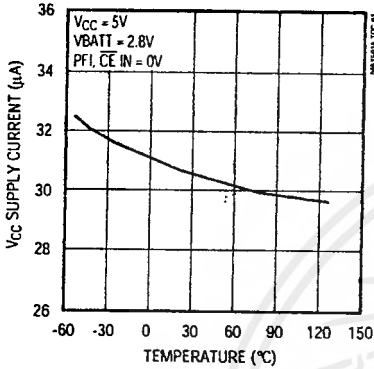
Note 8: The chip-enable propagation delay is measured from the 50% point at \overline{CE} IN to the 50% point at \overline{CE} OUT.

Microprocessor Supervisory Circuits

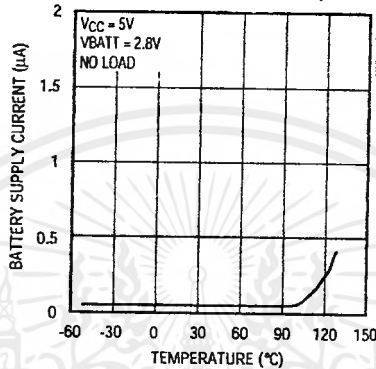
Typical Operating Characteristics

($T_A = +25^\circ\text{C}$, unless otherwise noted.)

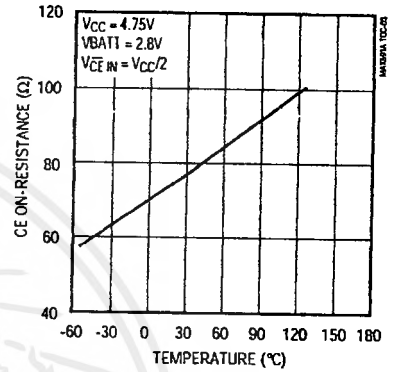
V_{CC} SUPPLY CURRENT vs. TEMPERATURE (NORMAL OPERATING MODE)



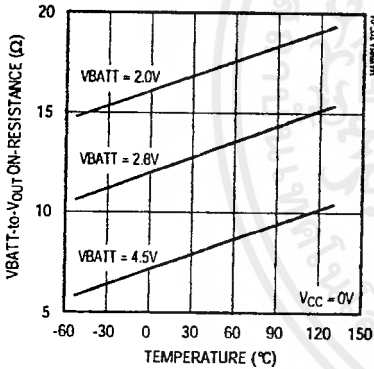
BATTERY SUPPLY CURRENT vs. TEMPERATURE (BATTERY-BACKUP MODE)



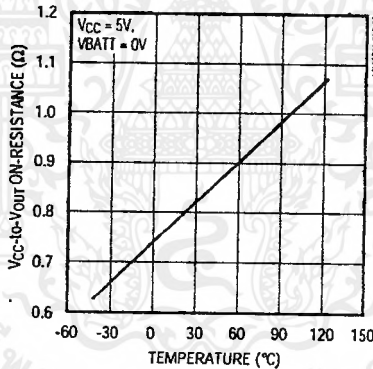
CHIP-ENABLE ON-RESISTANCE vs. TEMPERATURE



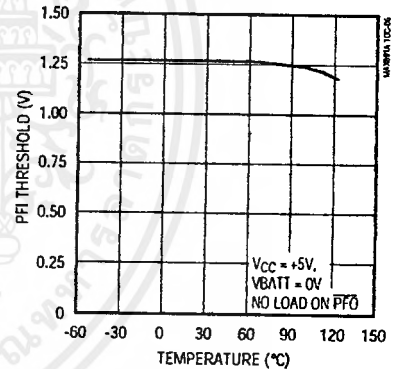
VBATT to V_{OUT} ON-RESISTANCE vs. TEMPERATURE



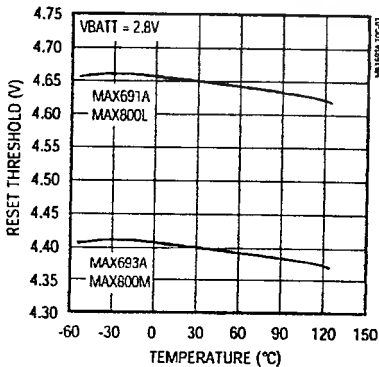
V_{CC} to V_{OUT} ON-RESISTANCE vs. TEMPERATURE



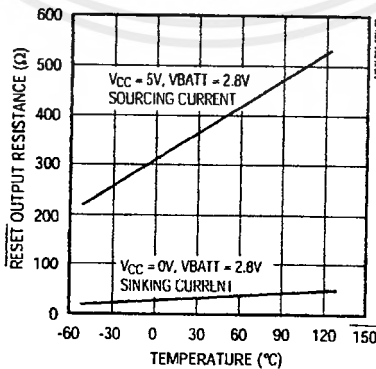
PFI THRESHOLD vs. TEMPERATURE



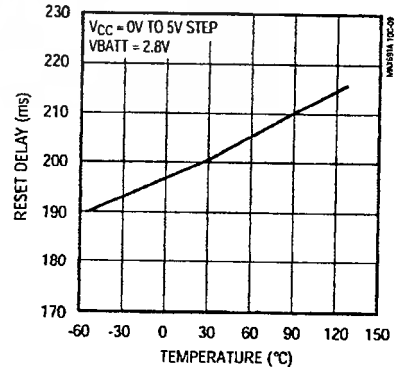
RESET THRESHOLD vs. TEMPERATURE



RESET OUTPUT RESISTANCE vs. TEMPERATURE



RESET DELAY vs. TEMPERATURE

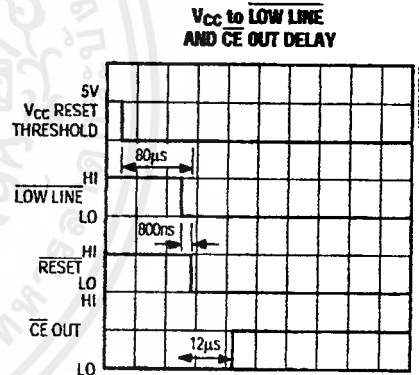
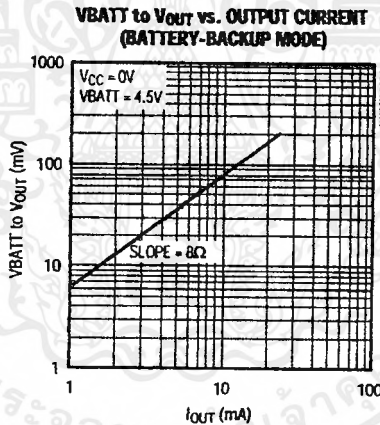
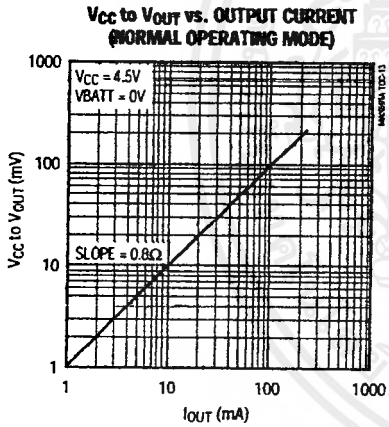
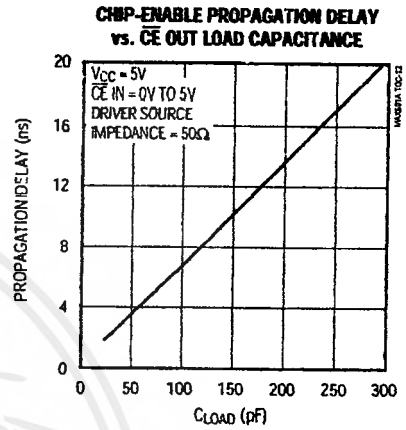
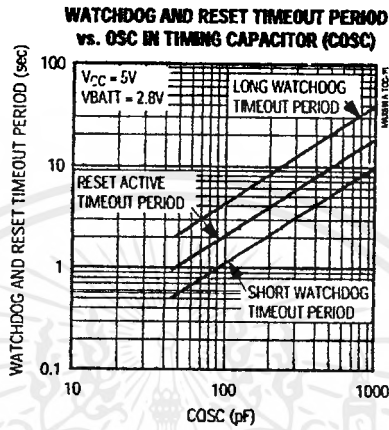
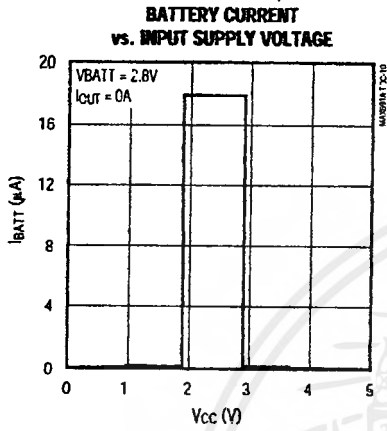


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

Typical Operating Characteristics (continued)

($T_A = +25^\circ\text{C}$, unless otherwise noted.)



MAX691A/MAX693A/MAX800L/MAX800M

Microprocessor Supervisory Circuits

Pin Description

PIN	NAME	FUNCTION
1	VBATT	Battery-Backup Input. Connect to external battery or capacitor and charging circuit. If backup battery is not used, connect to GND.
2	VOUT	Output Supply Voltage. When VCC is greater than VBATT and above the reset threshold, VOUT connects to VCC. When VCC falls below VBATT and is below the reset threshold, VOUT connects to VBATT. Connect a 0.1µF capacitor from VOUT to GND. Connect VOUT to VCC if no backup battery is used.
3	VCC	Input Supply Voltage, 5V input.
4	GND	Ground. 0V reference for all signals.
5	BATT ON	Battery On Output. When VOUT switches to VBATT, BATT ON goes high. When VOUT switches to VCC, BATT ON goes low. Connect the base of a PNP through a current-limiting resistor to BATT ON for VOUT current requirements greater than 250mA.
6	LOW LINE	LOW LINE output goes low when VCC falls below the reset threshold. It returns high as soon as VCC rises above the reset threshold.
7	OSC IN	External Oscillator Input. When OSC SEL is unconnected or driven high, a 10µA pull-up connects from VOUT to OSC IN, the internal oscillator sets the reset and watchdog timeout periods, and OSC IN selects between fast and slow watchdog timeout periods. When OSC SEL is driven low, the reset and watchdog timeout periods may be set either by a capacitor from OSC IN to ground or by an external clock at OSC IN (Figure 3).
8	OSC SEL	Oscillator Select. When OSC SEL is unconnected or driven high, the internal oscillator sets the reset delay and watchdog timeout period. When OSC SEL is low, the external oscillator input (OSC IN) is enabled (Table 1). OSC SEL has a 10µA internal pull-up.
9	PFI	Power-Fail Input. This is the noninverting input to the power-fail comparator. When PFI is less than 1.25V, PFO goes low. When PFI is not used, connect PFI to GND or VOUT.
10	PFO	Power-Fail Output. This is the output of the power-fail comparator. PFO goes low when PFI is less than 1.25V. This is an uncommitted comparator, and has no effect on any other internal circuitry.
11	WDI	Watchdog Input. WDI is a three-level input. If WDI remains either high or low for longer than the watchdog timeout period, WDO goes low and reset is asserted for the reset timeout period. WDO remains low until the next transition at WDI. Leaving WDI unconnected disables the watchdog function. WDI connects to an internal voltage divider between VOUT and GND, which sets it to mid-supply when left unconnected.
12	CE OUT	Chip-Enable Output. CE OUT goes low only when CE IN is low and VCC is above the reset threshold. If CE IN is low when reset is asserted, CE OUT will stay low for 15µs or until CE IN goes high, whichever occurs first.
13	CE IN	Chip-Enable Input. The input to chip-enable gating circuit. If CE IN is not used, connect CE IN to GND or VOUT.
14	WDO	Watchdog Output. If WDI remains high or low longer than the watchdog timeout period, WDO goes low and reset is asserted for the reset timeout period. WDO returns high on the next transition at WDI. WDO remains high if WDI is unconnected.
15	RESET	RESET Output goes low whenever VCC falls below the reset threshold. RESET will remain low typically for 200ms after VCC crosses the reset threshold on power-up.
16	RESET	RESET is an active-high output. It is open drain, and the inverse of RESET.

Detailed Description

RESET and RESET Outputs

The MAX691A/MAX693A/MAX800L/MAX800M's RESET and RESET outputs ensure that the µP (with reset inputs asserted either high or low) powers up in a known state, and prevents code-execution errors during power-down or brownout conditions.

The RESET output is active low, and typically sinks 3.2mA at 0.1V saturation voltage in its active state. When deasserted, RESET sources 1.6mA at typically VOUT - 0.5V. RESET output is open drain, active high, and typically sinks 3.2mA with a saturation voltage of 0.1V. When no backup battery is used, RESET output is

guaranteed to be valid down to VCC = 1V, and an external 10kΩ pull-down resistor on RESET insures that it will be valid with VCC down to GND (Figure 1). As VCC goes below 1V, the gate drive to the RESET output switch reduces accordingly, increasing the RDS(ON) and the saturation voltage. The 10kΩ pull-down resistor insures the parallel combination of switch plus resistor is around 10kΩ and the output saturation voltage is below 0.4V while sinking 40µA. When using a 10kΩ external pull-down resistor, the high state for RESET output with VCC = 4.75V will be 4.5V typical. For battery voltages ≥ 2V connected to VBATT, RESET and RESET remain valid for VCC from 0V to 5.5V.

MAXIM

Microprocessor Supervisory Circuits

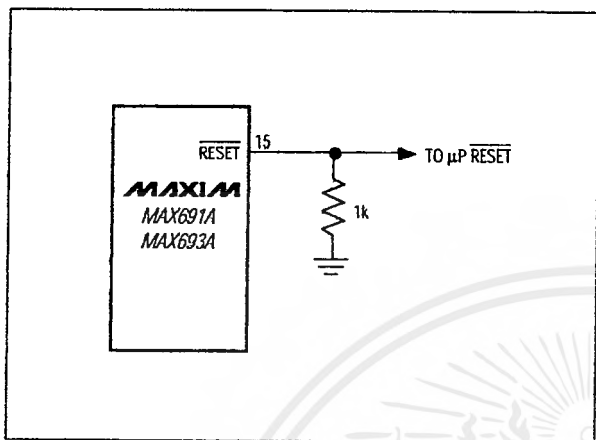


Figure 1. Adding an external pull-down resistor ensures $\overline{\text{RESET}}$ is valid with V_{CC} down to GND.

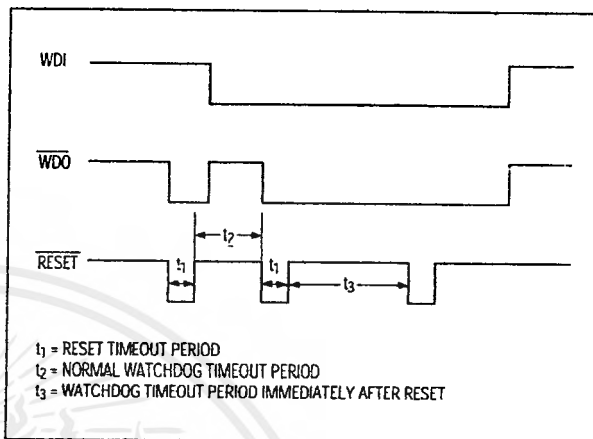


Figure 2. Watchdog Timeout Period and Reset Active Time

$\overline{\text{RESET}}$ and $\overline{\text{RESET}}$ are asserted when V_{CC} falls below the reset threshold (4.65V for the MAX691A/MAX800L, 4.4V for the MAX693A/MAX800M) and remain asserted for 200ms typ after V_{CC} rises above the reset threshold on power-up (Figure 5). The devices' battery-switchover comparator does not affect reset assertion. However, both reset outputs are asserted in battery-backup mode since V_{CC} must be below the reset threshold to enter this mode.

Watchdog Function

The watchdog monitors μP activity via the Watchdog Input (WDI). If the μP becomes inactive, $\overline{\text{RESET}}$ and $\overline{\text{RESET}}$ are asserted. To use the watchdog function, connect WDI to a bus line or μP I/O line. If WDI remains high or low for longer than the watchdog timeout period (1.6sec nominal), $\overline{\text{WDO}}$, $\overline{\text{RESET}}$, and $\overline{\text{RESET}}$ are asserted (see *RESET and $\overline{\text{RESET}}$ Outputs* section, and the *Watchdog Output* discussion on this page).

Watchdog Input

A change of state (high to low, low to high, or a minimum 100ns pulse) at the WDI during the watchdog period resets the watchdog timer. The watchdog default timeout is 1.6sec.

To disable the watchdog function, leave WDI floating. An internal resistor network (100k Ω equivalent impedance at WDI) biases WDI to approximately 1.6V. Internal comparators detect this level and disable the watchdog timer. When V_{CC} is below the reset threshold, the watchdog function is disabled and WDI is disconnected from its internal resistor network, thus becoming high impedance.

Watchdog Output

The Watchdog Output ($\overline{\text{WDO}}$) remains high if there is a transition or pulse at WDI during the watchdog timeout period. The watchdog function is disabled and $\overline{\text{WDO}}$ is a logic high when V_{CC} is below the reset threshold, battery-backup mode is enabled, or WDI is an open circuit. In watchdog mode, if no transition occurs at WDI during the watchdog timeout period, $\overline{\text{RESET}}$ and $\overline{\text{RESET}}$ are asserted for the reset timeout period (200ms typical). $\overline{\text{WDO}}$ goes low and remains low until the next transition at WDI (Figure 2). If WDI is held high or low indefinitely, $\overline{\text{RESET}}$ and $\overline{\text{RESET}}$ will generate 200ms pulses every 1.6sec. $\overline{\text{WDO}}$ has a 2 x TTL output characteristic.

Selecting an Alternative Watchdog and Reset Timeout Period

The OSC SEL and OSC IN inputs control the watchdog and reset timeout periods. Floating OSC SEL and OSC IN or tying them both to V_{OUT} selects the nominal 1.6sec watchdog timeout period and 200ms reset timeout period. Connecting OSC IN to GND and floating or connecting OSC SEL to V_{OUT} selects the 100ms normal watchdog timeout delay and 1.6sec delay immediately after reset. The reset timeout delay remains 200ms (Figure 2). Select alternative timeout periods by connecting OSC SEL to GND and connecting a capacitor between OSC IN and GND, or by externally driving OSC IN (Table 1 and Figure 3). OSC IN is internally connected to a $\pm 100\text{nA}$ (typ) current source that charges and discharges the timing capacitor to create the oscillator frequency, which sets the reset and watchdog timeout periods (see *Connecting a Timing Capacitor at OSC IN* in the *Applications Information* section).

Microprocessor Supervisory Circuits

Table 1. Reset Pulse Width and Watchdog Timeout Selections

OSC SEL	OSC IN	Watchdog Timeout Period		Reset Timeout Period
		Normal	Immediately After Reset	
Low	External Clock Input	1024 clks	4096 clks	2048 clks
Low	External Capacitor	(600/47pF x C)ms	(2.4/47pF x C)sec	(1200/47pF x C)ms
Floating	Low	100ms	1.6sec	200ms
Floating	Floating	1.6sec	1.6sec	200ms

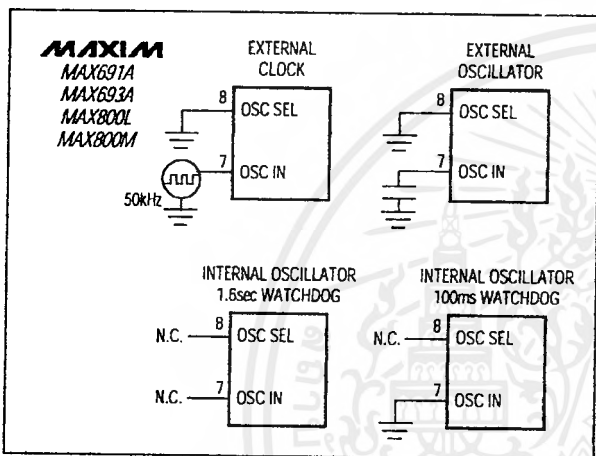


Figure 3. Oscillator Circuits

Chip-Enable Signal Gating

The MAX691A/MAX693A/MAX800L/MAX800M provide internal gating of chip-enable (CE) signals to prevent erroneous data from being written to CMOS RAM in the event of a power failure. During normal operation, the CE gate is enabled and passes all CE transitions. When reset is asserted, this path becomes disabled, preventing erroneous data from corrupting the CMOS RAM. All these parts use a series transmission gate from \overline{CE} IN to \overline{CE} OUT (Figure 4).

The 10ns max CE propagation delay from \overline{CE} IN to \overline{CE} OUT enables the parts to be used with most μ Ps.

Chip-Enable Input

The Chip-Enable Input (\overline{CE} IN) is high impedance (disabled mode) while RESET and \overline{RESET} are asserted.

During a power-down sequence where V_{CC} falls below the reset threshold or a watchdog fault, \overline{CE} IN assumes a high-impedance state when the voltage at \overline{CE} IN goes high or 15 μ s after reset is asserted, whichever occurs first (Figure 5).

During a power-up sequence, \overline{CE} IN remains high impedance, regardless of \overline{CE} IN activity, until reset is deasserted following the reset timeout period.

In the high-impedance mode, the leakage currents into this terminal are $\pm 1\mu$ A max over temperature. In the low-impedance mode, the impedance of \overline{CE} IN appears as a 75 Ω resistor in series with the load at \overline{CE} OUT.

The propagation delay through the CE transmission gate depends on both the source impedance of the drive to \overline{CE} IN and the capacitive loading on the Chip-Enable Output (\overline{CE} OUT) (see Chip-Enable Propagation Delay vs. \overline{CE} OUT Load Capacitance in the *Typical Operating Characteristics*). The CE propagation delay is production tested from the 50% point of \overline{CE} IN to the 50% point of \overline{CE} OUT using a 50 Ω driver and 50pF of load capacitance (Figure 6). For minimum propagation delay, minimize the capacitive load at \overline{CE} OUT, and use a low output-impedance driver.

Chip-Enable Output

In the enabled mode, the impedance of \overline{CE} OUT is equivalent to 75 Ω in series with the source driving \overline{CE} IN. In the disabled mode, the 75 Ω transmission gate is off and \overline{CE} OUT is actively pulled to V_{OUT} . This source turns off when the transmission gate is enabled.

LOW LINE Output

LOW LINE is the buffered output of the reset threshold comparator. LOW LINE typically sinks 3.2mA at 0.1V. For normal operation (V_{CC} above the LOW LINE threshold), LOW LINE is pulled to V_{OUT} .

Power-Fail Comparator

The power-fail comparator is an uncommitted comparator that has no effect on the other functions of the IC. Common uses include low-battery indication (Figure 7), and early power-fail warning (see *Typical Operating Circuit*).

Power-Fail Input

Power Fail Input (PFI) is the input to the power-fail comparator. It has a guaranteed input leakage of ± 25 nA max over temperature. The typical comparator delay is 25 μ s from V_{IL} to V_{OL} (power failing), and 60 μ s from V_{IH} to V_{OH} (power being restored). If PFI is not used, connect it to ground.

MAX691A/MAX693A/MAX800L/MAX800M

Microprocessor Supervisory Circuits

MAX691A/MAX693A/MAX800L/MAX800M

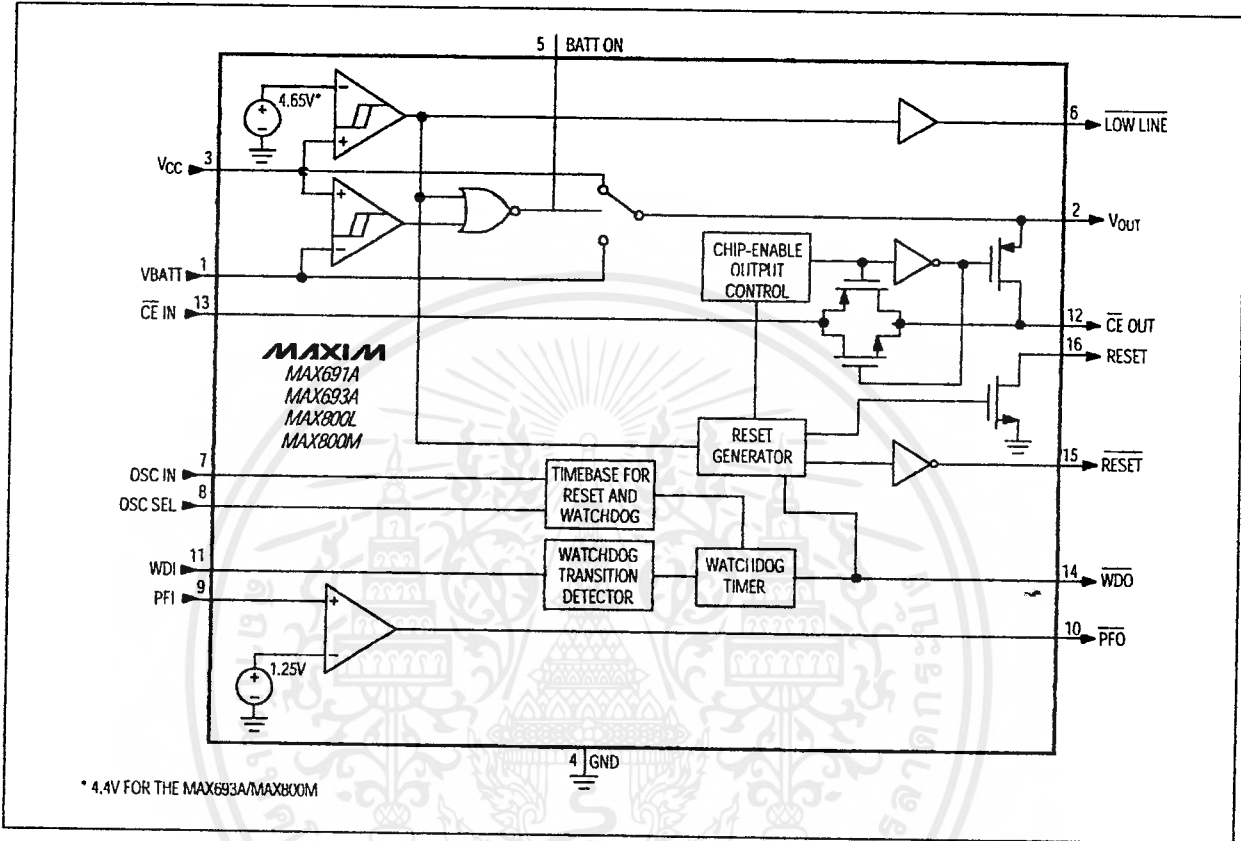


Figure 4. MAX691A/MAX693A/MAX800L/MAX800M Block Diagram

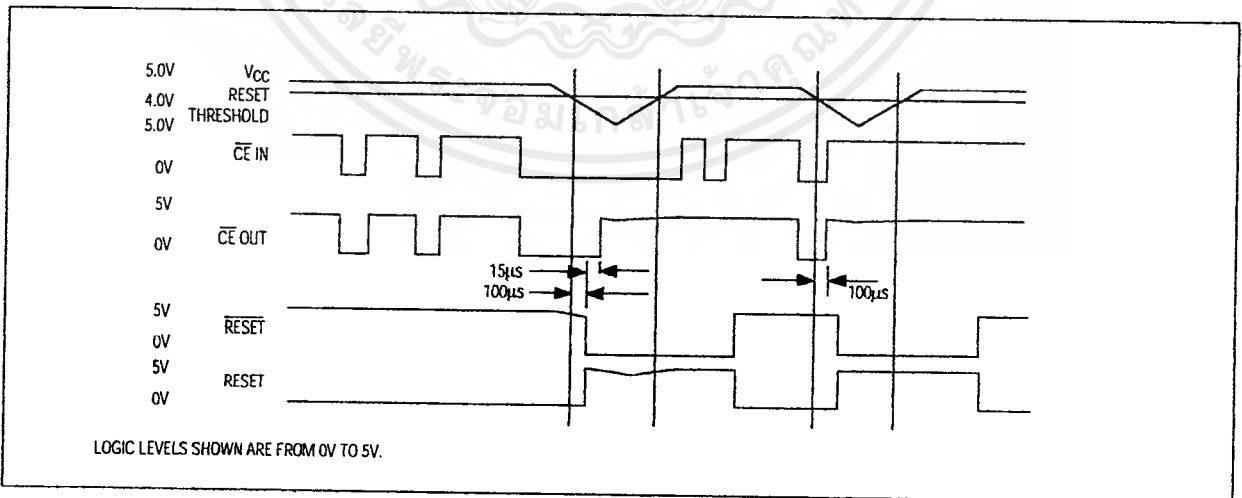


Figure 5. Reset and Chip-Enable Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microprocessor Supervisory Circuits

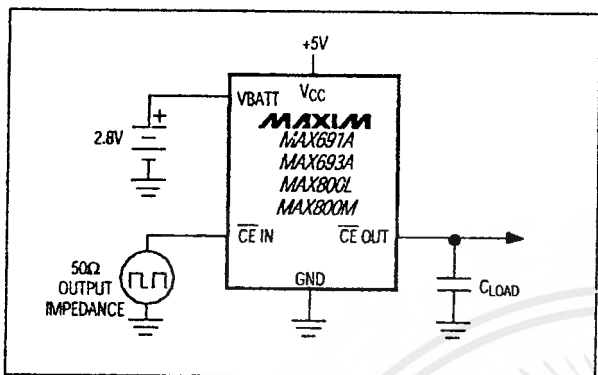


Figure 6. CE Propagation Delay Test Circuit

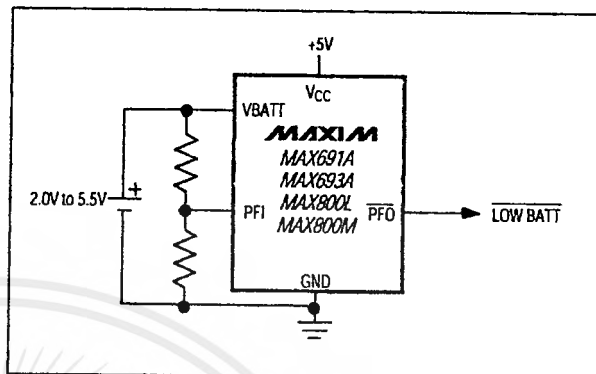


Figure 7. Low-Battery Indicator

Table 2. Input and Output Status in Battery-Backup Mode

PIN	NAME	STATUS
1	VBATT	Supply current is 1 μ A max.
2	VOUT	VOUT is connected to VBATT through an internal PMOS switch.
3	VCC	Battery switchover comparator; monitors VCC for active switchover.
4	GND	GND 0V, 0V reference for all signals.
5	BATT ON	Logic high. The open-circuit output is equal to VOUT.
6	LOWLINE	Logic low*
7	OSC IN	OSC IN is ignored.
8	OSC SEL	OSC SEL is ignored.
9	PFI	The power-fail comparator remains active in the battery-backup mode for VCC \geq VBATT - 1.2V typ.
10	PFO	The power-fail comparator remains active in the battery-backup mode for VCC \geq VBATT - 1.2V typ. Below this voltage, PFO is forced low.
11	WDI	Watchdog is ignored.
12	CE OUT	Logic high. The open-circuit voltage is equal to VOUT.
13	CE IN	High impedance
14	WDO	Logic high. The open-circuit voltage is equal to VOUT.
15	RESET	Logic low*
16	RESET	High impedance*

* VCC must be below the reset threshold to enter battery-backup mode.

Power-Fail Output

The Power-Fail Output (PFO) goes low when PFI goes below 1.25V. It typically sinks 3.2mA with a saturation voltage of 0.1V. With PFI above 1.25V, PFO is actively pulled to VOUT.

Battery-Backup Mode

Two conditions are required to switch to battery-backup mode: 1) VCC must be below the reset threshold, and 2) VCC must be below VBATT. Table 2 lists the status of the inputs and outputs in battery-backup mode.

Battery On Output

The Battery On (BATT ON) output indicates the status of the internal VCC/battery-switchover comparator, which controls the internal VCC and VBATT switches. For VCC greater than VBATT (ignoring the small hysteresis effect), BATT ON typically sinks 3.2mA at 0.1V saturation voltage. In battery-backup mode, this terminal sources approximately 10 μ A from VOUT. Use BATT ON to indicate battery-switchover status or to supply base drive to an external pass transistor for higher-current applications (see *Typical Operating Circuit*).

Input Supply Voltage

The Input Supply Voltage (VCC) should be a regulated 5V. VCC connects to VOUT via a parallel diode and a large PMOS switch. The switch carries the entire current load for currents less than 250mA. The parallel diode carries any current in excess of 250mA. Both the switch and the diode have impedances less than 1 Ω each. The maximum continuous current is 250mA, but power-on transients may reach a maximum of 1A.

Microprocessor Supervisory Circuits

Battery-Backup Input

The Battery-Backup Input (VBATT) is similar to the V_{CC} input except the PMOS switch and parallel diode are much smaller. Accordingly, the on-resistances of the diode and the switch are each approximately 10Ω . Continuous current should be limited to 25mA and peak currents (only during power-up) limited to 250mA. The reverse leakage of this input is less than $1\mu A$ over temperature and supply voltage (Figure 8).

Output Supply Voltage

The Output Supply Voltage (V_{OUT}) pin is internally connected to the substrate of the IC and supplies current to the external system and internal circuitry. All open-circuit outputs will, for example, assume the V_{OUT} voltage in their high states rather than the V_{CC} voltage. At the maximum source current of 250mA, V_{OUT} will typically be 200mV below V_{CC} . Decouple this terminal with a $0.1\mu F$ capacitor.

Applications Information

The MAX691A/MAX693A/MAX800L/MAX800M are not short-circuit protected. Shorting V_{OUT} to ground, other than power-up transients such as charging a decoupling capacitor, destroys the device.

All open-circuit outputs swing between V_{OUT} and GND rather than V_{CC} and GND.

If long leads connect to the chip inputs, insure that these leads are free from ringing and other conditions that would forward bias the chip's protection diodes.

There are three distinct modes of operation:

- 1) Normal operating mode with all circuitry powered up. Typical supply current from V_{CC} is $35\mu A$ while only leakage currents flow from the battery.
- 2) Battery-backup mode where V_{CC} is typically within 0.7V below VBATT. All circuitry is powered up and the supply current from the battery is typically less than $60\mu A$.
- 3) Battery-backup mode where V_{CC} is less than VBATT by at least 0.7V. VBATT supply current is $1\mu A$ max.

Using SuperCap™ or MaxCap™ with the MAX691A/MAX693A/MAX800L/MAX800M

VBATT has the same operating voltage range as V_{CC} , and the battery switchover threshold voltages are typically $\pm 30mV$ centered at VBATT, allowing use of a SuperCap and a simple charging circuit as a backup source (Figure 9).

If V_{CC} is above the reset threshold and VBATT is 0.5V above V_{CC} , current flows to V_{OUT} and V_{CC} from VBATT until the voltage at VBATT is less than 0.5V above V_{CC} . For example, with a SuperCap connected to VBATT and through a diode to V_{CC} , if V_{CC} quickly changes from 5.4V to 4.9V, the capacitor discharges through V_{OUT} and V_{CC} until VBATT reaches 5.1V typ. Leakage current through the SuperCap charging diode and the internal power diode eventually discharges the SuperCap to V_{CC} . Also, if V_{CC} and VBATT start from 0.1V above the reset threshold and power is lost at V_{CC} , the SuperCap on VBATT discharges through V_{CC} until VBATT reaches the reset threshold; then the battery-backup mode is initiated and the current through V_{CC} goes to zero.

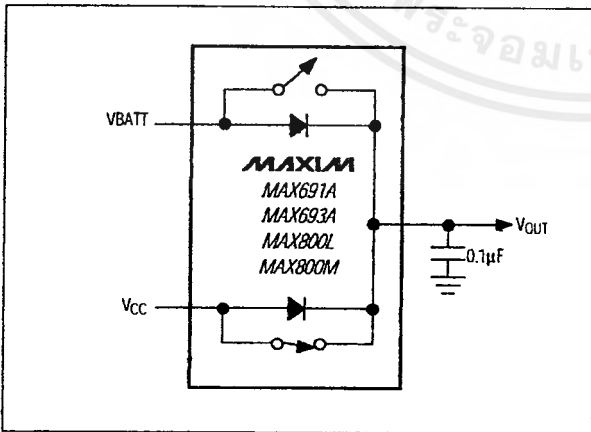


Figure 8. V_{CC} and VBATT to V_{OUT} Switch

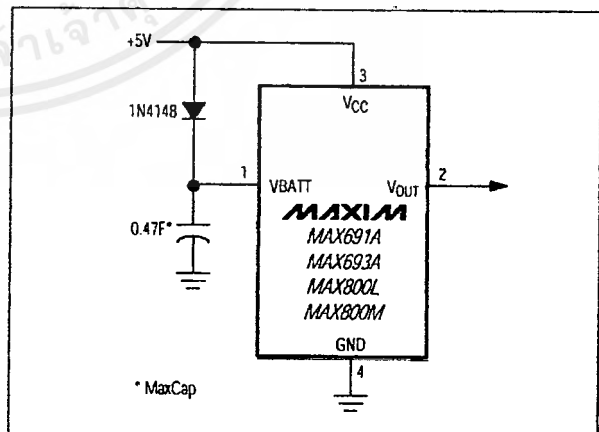


Figure 9. SuperCap or MaxCap on VBATT

Microprocessor Supervisory Circuits

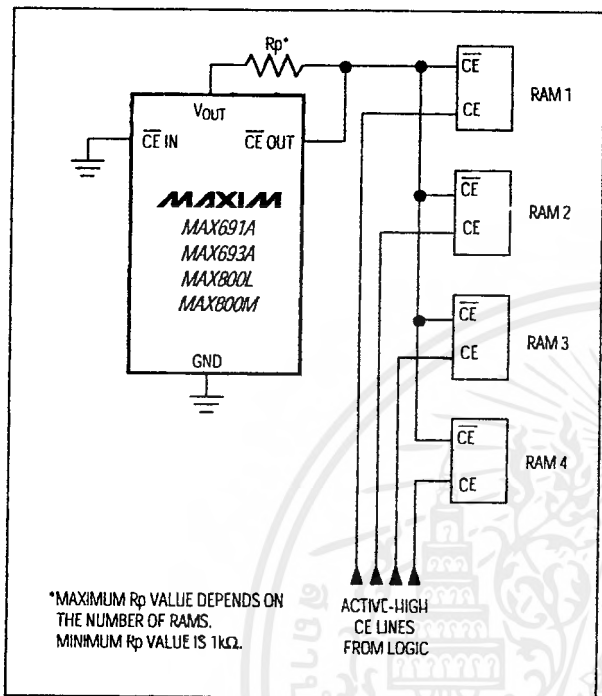


Figure 10. Alternate CE Gating

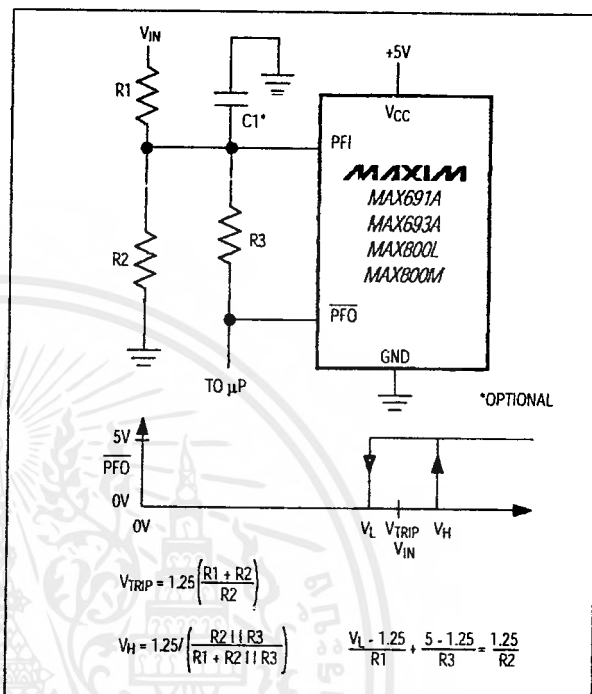


Figure 11. Adding Hysteresis to the Power-Fail Comparator

Using Separate Power Supplies for VBATT and VCC

If using separate power supplies for V_{CC} and V_{BATT} , V_{BATT} must be less than 0.3V above V_{CC} when V_{CC} is above the reset threshold. As described in the previous section, if V_{BATT} exceeds this limit and power is lost at V_{CC} , current flows continuously from V_{BATT} to V_{CC} via the V_{BATT} -to- V_{OUT} diode and the V_{OUT} -to- V_{CC} switch until the circuit is broken (Figure 8).

Alternate Chip-Enable Gating

Using memory devices with both CE and \overline{CE} inputs allows the CE loop to be bypassed. To do this, connect \overline{CE} IN to ground, pull up \overline{CE} OUT to V_{OUT} , and connect \overline{CE} OUT to the CE input of each memory device (Figure 10). The CE input of each part then connects directly to the chip-select logic, which does not have to be gated.

Adding Hysteresis to the Power-Fail Comparator

Hysteresis adds a noise margin to the power-fail comparator and prevents repeated triggering of PFO when V_{IN} is near the power-fail comparator trip point. Figure 11 shows how to add hysteresis to the power-fail com-

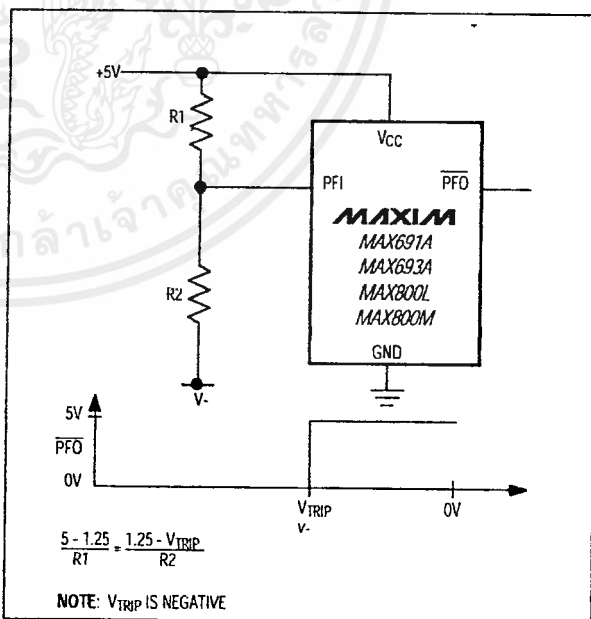


Figure 12. Monitoring a Negative Voltage

Microprocessor Supervisory Circuits

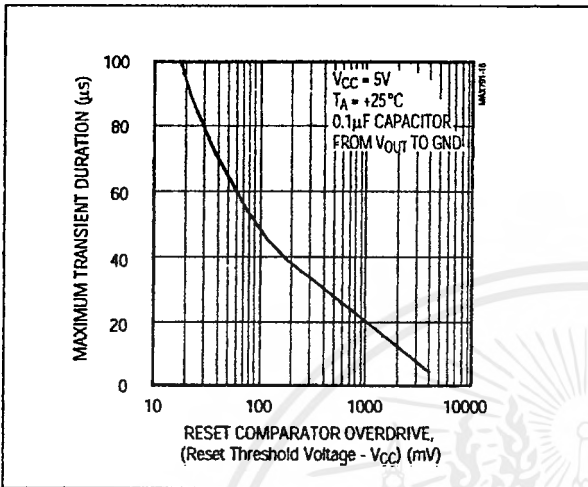


Figure 13. Maximum Transient Duration without Causing a Reset Pulse vs. Reset Comparator Overdrive

parator. Select the ratio of R1 and R2 such that PFI sees 1.25V when V_{IN} falls to the desired trip point (V_{TRIP}). Resistor R3 adds hysteresis. It will typically be an order of magnitude greater than R1 or R2. The current through R1 and R2 should be at least $1\mu A$ to ensure that the 25nA (max) PFI input current does not shift the trip point. R3 should be larger than $10k\Omega$ to prevent it from loading down the PFO pin. Capacitor C1 adds noise rejection.

Monitoring a Negative Voltage

The power-fail comparator can be used to monitor a negative supply voltage using Figure 12's circuit. When the negative supply is valid, \overline{PFO} is low. When the negative supply voltage drops, \overline{PFO} goes high. This circuit's accuracy is affected by the PFI threshold tolerance, the V_{CC} voltage, and resistors R1 and R2.

Backup-Battery Replacement

The backup battery may be disconnected while V_{CC} is above the reset threshold. No precautions are necessary to avoid spurious reset pulses.

Negative-Going V_{CC} Transients

While issuing resets to the μP during power-up, power-down, and brownout conditions, these supervisors are relatively immune to short-duration, negative-going V_{CC} transients (glitches). It is usually undesirable to reset the μP when V_{CC} experiences only small glitches.

Figure 13 shows maximum transient duration vs. reset-comparator overdrive, for which reset pulses are **not** generated. The graph was produced using negative-going V_{CC} pulses, starting at 5V and ending below the reset threshold by the magnitude indicated (reset comparator overdrive). The graph shows the maximum pulse width a negative-going V_{CC} transient may typically have without causing a reset pulse to be issued. As the amplitude of the transient increases (i.e., goes farther below the reset threshold), the maximum allowable pulse width decreases. Typically, a V_{CC} transient that goes 100mV below the reset threshold and lasts for 40 μs or less will not cause a reset pulse to be issued.

A 100nF bypass capacitor mounted close to the V_{CC} pin provides additional transient immunity.

Connecting a Timing Capacitor at OSC IN

When OSC SEL is connected to ground, OSC IN disconnects from its internal $10\mu A$ (typ) pull-up and is internally connected to a $\pm 100nA$ current source. When a capacitor is connected from OSC IN to ground (to select alternative reset and watchdog timeout periods), the current source charges and discharges the timing capacitor to create the oscillator that controls the reset and watchdog timeout period. To prevent timing errors or oscillator start-up problems, minimize external current leakage sources at this pin, and locate the capacitor as close to OSC IN as possible. The sum of PC-board leakage plus OSC capacitor leakage must be small compared to $\pm 100nA$.

Microprocessor Supervisory Circuits

Maximum V_{CC} Fall Time

The V_{CC} fall time is limited by the propagation delay of the battery switchover comparator and should not exceed 0.03V/μs. A standard rule of thumb for filter capacitance on most regulators is on the order of 100μF per amp of current. When the power supply is shut off or the main battery is disconnected, the associated initial V_{CC} fall rate is just the inverse or 1A/100μF = 0.01V/μs. The V_{CC} fall rate decreases with time as V_{CC} falls exponentially, which more than satisfies the maximum fall-time requirement.

Watchdog Software Considerations

A way to help the watchdog timer keep a closer watch on software execution involves setting and resetting the watchdog input at different points in the program, rather than "pulsing" the watchdog input high-low-high or low-high-low. This technique avoids a "stuck" loop where the watchdog timer continues to be reset within the loop, keeping the watchdog from timing out. Figure 14 shows an example flow diagram where the I/O driving the watchdog input is set high at the beginning of the program, set low at the beginning of every subroutine or loop, then set high again when the program returns to the beginning. If the program should "hang" in any subroutine, the I/O is continually set low and the watchdog timer is allowed to time out, causing a reset or interrupt to be issued.

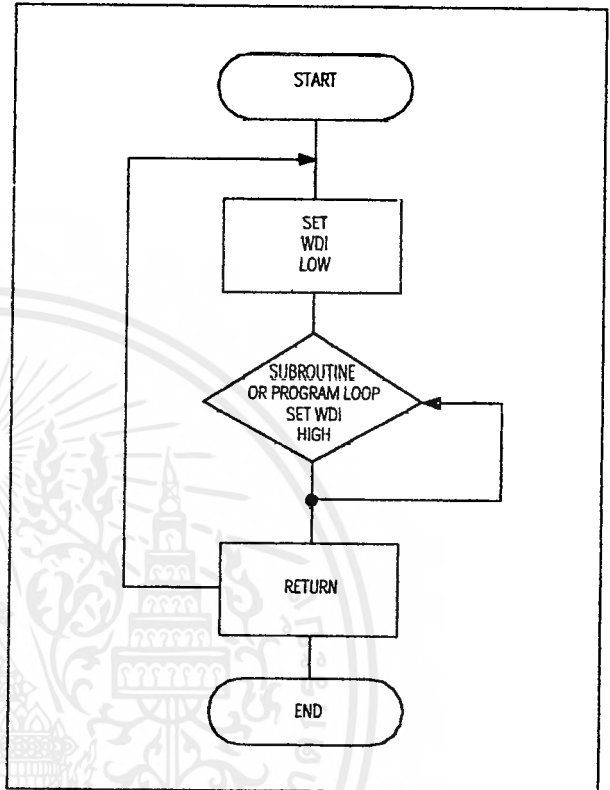


Figure 14. Watchdog Flow Diagram

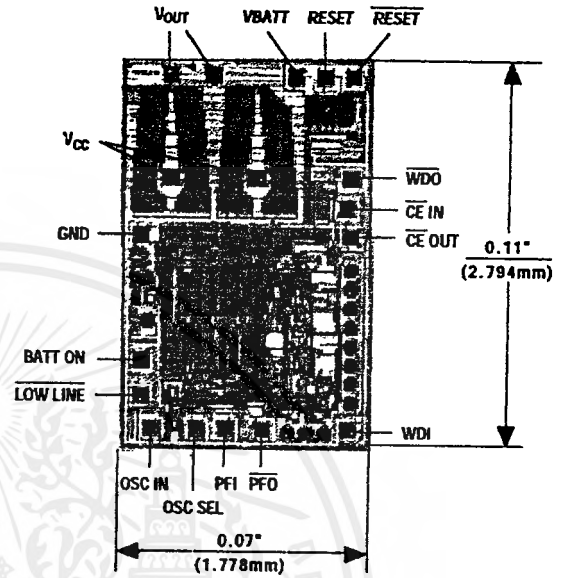
Microprocessor Supervisory Circuits

Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX693ACPE	0°C to +70°C	16 Plastic DIP
MAX693ACSE	0°C to +70°C	16 Narrow SO
MAX693ACWE	0°C to +70°C	16 Wide SO
MAX693AC/D	0°C to +70°C	Dice*
MAX693AEPE	-40°C to +85°C	16 Plastic SO
MAX693AESE	-40°C to +85°C	16 Narrow SO
MAX693AEWE	-40°C to +85°C	16 Wide SO
MAX693AEJE	-40°C to +85°C	16 CERDIP
MAX693AMJE	-55°C to +125°C	16 CERDIP
MAX800LCPE	0°C to +70°C	16 Plastic DIP
MAX800LCSE	0°C to +70°C	16 Narrow SO
MAX800LEPE	-40°C to +85°C	16 Plastic DIP
MAX800LESE	-40°C to +85°C	16 Narrow SO
MAX800MCPE	0°C to +70°C	16 Plastic DIP
MAX800MCSE	0°C to +70°C	16 Narrow SO
MAX800MEPE	-40°C to +85°C	16 Plastic DIP
MAX800MESE	-40°C to +85°C	16 Narrow SO

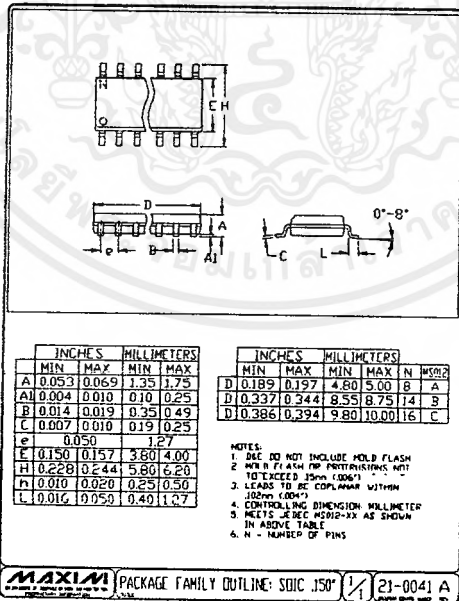
* Dice are specified at $T_A = +25^\circ\text{C}$, DC parameters only.

Chip Topography



TRANSISTOR COUNT: 729
SUBSTRATE CONNECTED TO VOUT

Package Information



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

16 **Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600**

© 1996 Maxim Integrated Products

Printed USA

MAXIM is a registered trademark of Maxim Integrated Products.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. 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
Multi-Drop 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.26 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 (μ F)	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	8/10	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

MAXIM

Maxim Integrated Products 1

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800.
For small orders, phone 408-737-7600 ext. 3468.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
T _{IN}	-0.3V to (V _{CC} - 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN}	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
T _{OUT} (Note 1)	±15V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
Output Voltages		16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
T _{OUT}	±15V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	Operating Temperature Ranges	
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2_AC_, MAX2_C_	0°C to +70°C
Continuous Power Dissipation (T _A = +70°C)		MAX2_AE_, MAX2_E_	-40°C to +85°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	MAX2_AM_, MAX2_M_	-55°C to +125°C
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	889mW	Storage Temperature Range	-65°C to +160°C
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW	Lead Temperature (soldering, 10sec)	+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, SHDN or V_{CC} = 0V.

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

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1-C4 = 0.1µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	Normal operation			5	40	µA
	SHDN = 0V, MAX222/242, shutdown			±0.01	±1	
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V, MAX222/242			±0.01	±10	µA
	V _{CC} = SHDN = 0V, V _{OUT} = ±15V			±0.01	±10	
Data Rate	All except MAX220, normal operation			200	116	kbits/ sec
	MAX220			22	20	
Transmitter Output Resistance	V _{CC} = V+ = V- = 0V, V _{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V _{CC} = 5V	All except MAX243 R2 _{IN}	0.8	1.3		V
		MAX243 R2 _{IN} (Note 2)	-3			
RS-232 Input Threshold High	V _{CC} = 5V	All except MAX243 R2 _{IN}		1.8	2.4	V
		MAX243 R2 _{IN} (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V _{CC} = 5V, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Sinking V _{OUT} = V _{CC}		10	30		
TTL/CMOS Output Leakage Current	SHDN = V _{CC} or EN = V _{CC} (SHDN = 0V for MAX222), 0V ≤ V _{OUT} ≤ V _{CC}			±0.05	±10	µA

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

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ± 10%, C₁–C₄ = 0.1μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

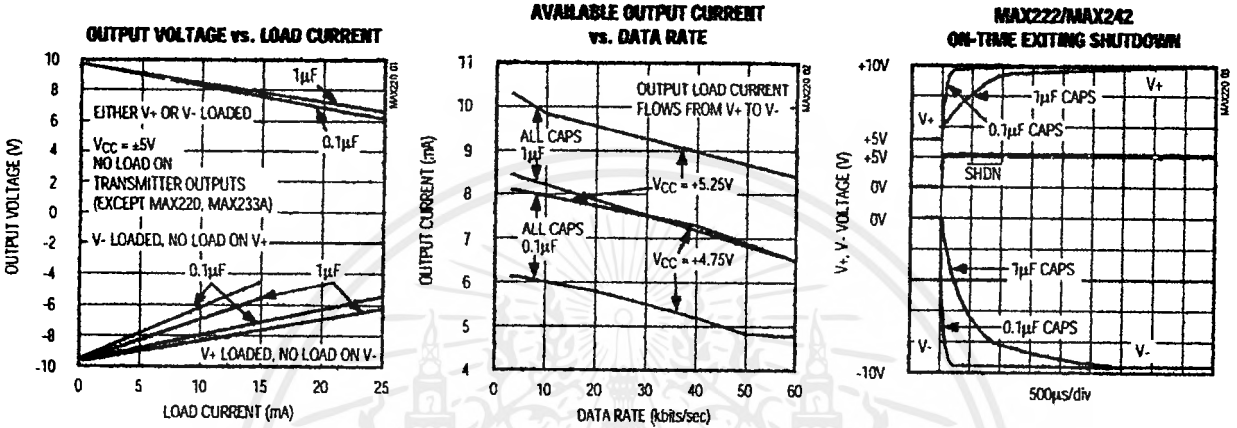
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
EN Input Threshold Low	MAX242			1.4	0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V _{CC} Supply Current (SHDN = V _{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T _A = +25°C		0.1	10	μA
		T _A = 0°C to +70°C		2	50	
		T _A = -40°C to +85°C		2	50	
		T _A = -55°C to +125°C		35	100	
SHDN Input Leakage Current	MAX222/242				±1	μA
SHDN Threshold Low	MAX222/242			1.4	0.8	V
SHDN Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C _L = 50pF to 2500pF, R _L = 3kΩ to 7kΩ, V _{CC} = 5V, T _A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/μs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	t _{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	μs
		MAX220		4	10	
	t _{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TLL (normal operation), Figure 2	t _{PHLR}	MAX222/232A/233A/242/243		0.5	1	μs
		MAX220		0.6	3	
	t _{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (shutdown), Figure 2	t _{PHLS}	MAX242		0.5	10	μs
	t _{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t _{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t _{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time (SHDN goes high), Figure 4	t _{ET}	MAX222/242, 0.1μF caps (includes charge-pump start-up)		250		μs
Transmitter-Output Disable Time (SHDN goes low), Figure 4	t _{DT}	MAX222/242, 0.1μF caps		600		ns
Transmitter + to - Propagation Delay Difference (normal operation)	t _{PHLT} - t _{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (normal operation)	t _{PHLR} - t _{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

Note 2: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is ≥ 0V or is floating.

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

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



DATA SHEET

For a complete data sheet, please also download:

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

74HC/HCT373

Octal D-type transparent latch;
3-state

Product specification
File under Integrated Circuits, IC06

September 1993

Philips
Semiconductors



PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Octal D-type transparent latch; 3-state

74HC/HCT373

FEATURES

- 3-state non-inverting outputs for bus oriented applications
- Common 3-state output enable input
- Functionally identical to the "563", "573" and "533"
- Output capability: bus driver
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT373 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT373 are octal D-type transparent latches featuring separate D-type inputs for each latch and 3-state outputs for bus oriented applications. A latch enable (LE)

input and an output enable (\overline{OE}) input are common to all latches.

The "373" consists of eight D-type transparent latches with 3-state true outputs. When LE is HIGH, data at the D_n inputs enters the latches. In this condition the latches are transparent, i.e. a latch output will change state each time its corresponding D-input changes.

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

The "373" is functionally identical to the "533", "563" and "573", but the "563" and "533" have inverted outputs and the "563" and "573" have a different pin arrangement.

QUICK REFERENCE DATA

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

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PHL} / t _{PLH}	propagation delay	C _L = 15 pF; V _{CC} = 5 V			
	D _n to Q _n		12	14	ns
	LE to Q _n		15	13	ns
C _I	input capacitance		3.5	3.5	pF
C _{PD}	power dissipation capacitance per latch	notes 1 and 2	45	41	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

∑ (C_L × V_{CC}² × f_o) = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}. For HCT the condition is V_I = GND to V_{CC} - 1.5 V

ORDERING INFORMATION

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

Octal D-type transparent latch; 3-state

74HC/HCT373

PIN DESCRIPTION

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

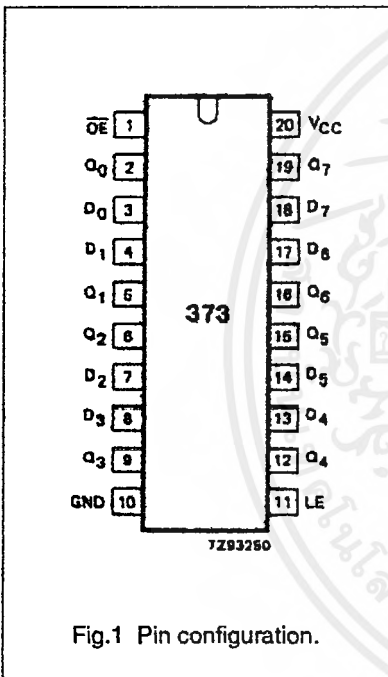


Fig.1 Pin configuration.

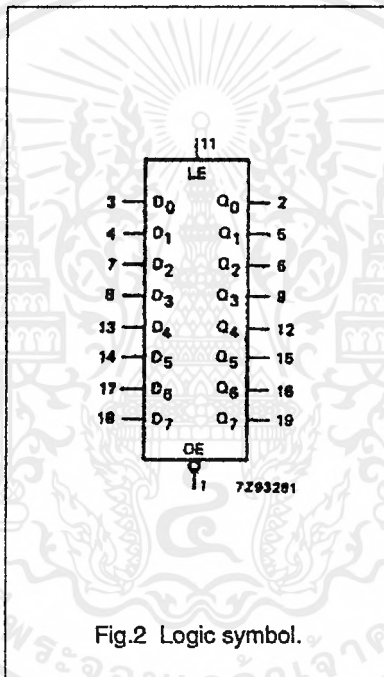


Fig.2 Logic symbol.

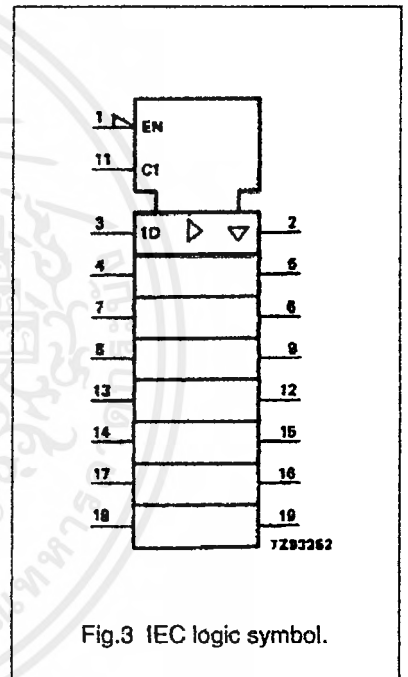


Fig.3 IEC logic symbol.

Octal D-type transparent latch; 3-state

74HC/HCT373

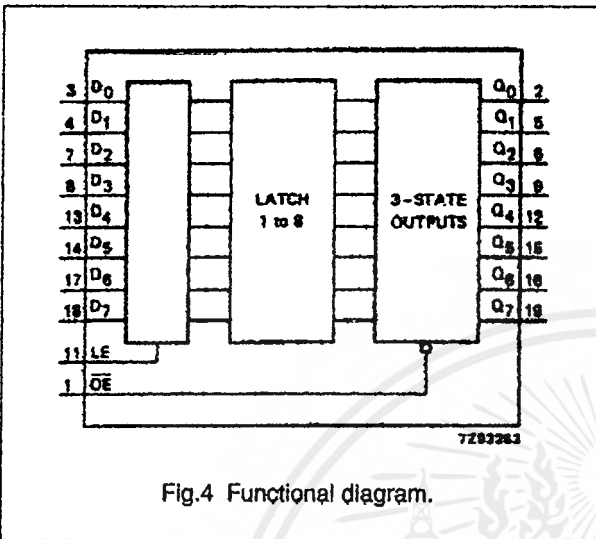


Fig.4 Functional diagram.

FUNCTION TABLE

OPERATING MODES	INPUTS			INTERNAL LATCHES	OUTPUTS Q ₀ to Q ₇
	OE	LE	D _n		
enable and read register (transparent mode)	L	H	L	L	L
	L	H	H	H	H
latch and read register	L	L	l	L	L
	L	L	h	H	H
latch register and disable outputs	H	X	X	X	Z
	H	X	X	X	Z

Notes

- H = HIGH voltage level
h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition
L = LOW voltage level
l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition
X = don't care
Z = high impedance OFF-state

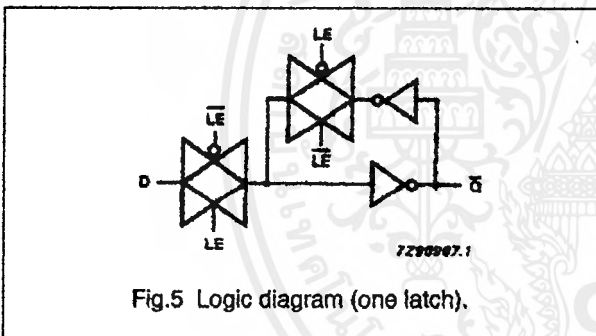


Fig.5 Logic diagram (one latch).

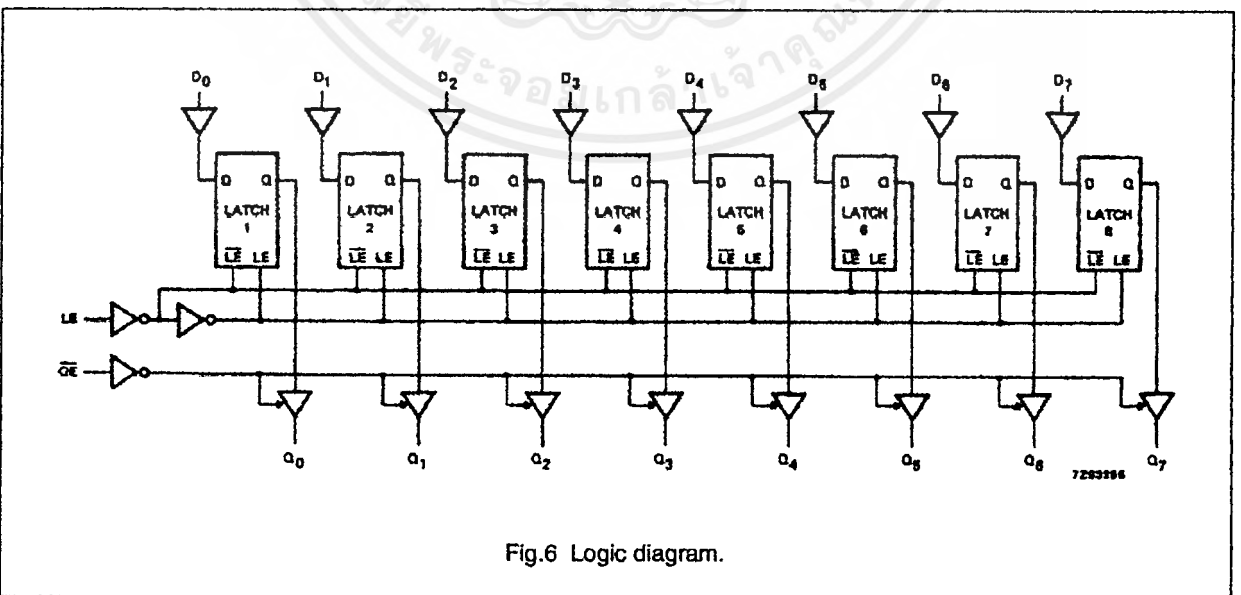


Fig.6 Logic diagram.

Octal D-type transparent latch; 3-state

74HC/HCT373

DC CHARACTERISTICS FOR 74HC

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

Output capability: bus driver

I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

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

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HC							V _{CC} (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t _{PHL} /t _{PLH}	propagation delay D _n to Q _n		41 15 12	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.7
t _{PHL} /t _{PLH}	propagation delay LE to Q _n		50 18 14	175 35 30		220 44 37		265 53 45	ns	2.0 4.5 6.0	Fig.8
t _{PZH} /t _{PZL}	3-state output enable time OE to Q _n		44 16 13	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.9
t _{PHZ} /t _{PLZ}	3-state output disable time OE to Q _n		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.9
t _{THL} /t _{TLH}	output transition time		14 5 4	60 12 10		75 15 13		90 18 15	ns	2.0 4.5 6.0	Fig.7
t _w	LE pulse width HIGH	80 16 14	17 6 5		100 20 17		120 24 20		ns	2.0 4.5 6.0	Fig.8
t _{su}	set-up time D _n to LE	50 10 9	14 5 4		65 13 11		75 15 13		ns	2.0 4.5 6.0	Fig.10
t _h	hold time D _n to LE	5 5 5	-8 -3 -2		5 5 5		5 5 5		ns	2.0 4.5 6.0	Fig.10

Octal D-type transparent latch; 3-state

74HC/HCT373

DC CHARACTERISTICS FOR 74HCT

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

Output capability: bus driver

I_{CC} category: MSI

Note to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications.

To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
D _n	0.30
LE	1.50
\overline{OE}	1.00

AC CHARACTERISTICS FOR 74HCT

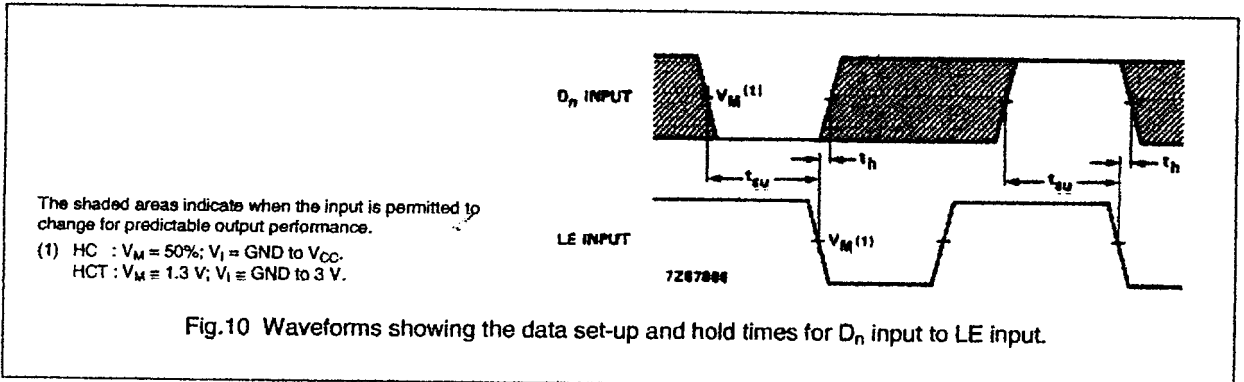
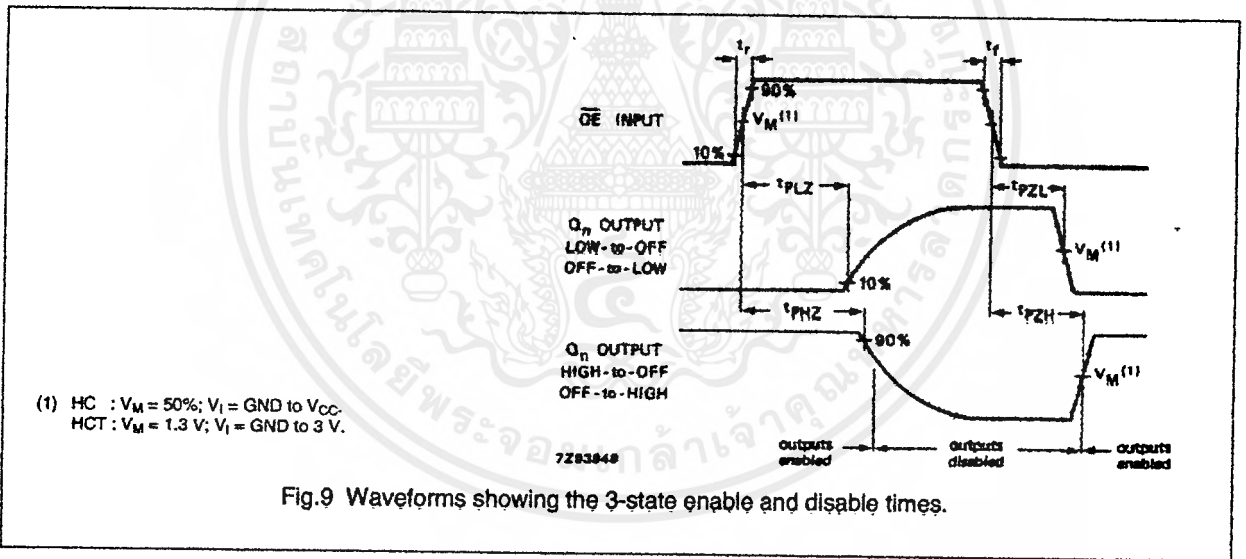
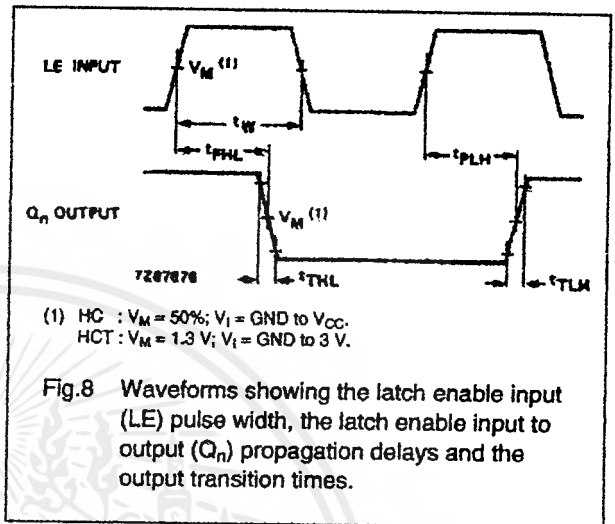
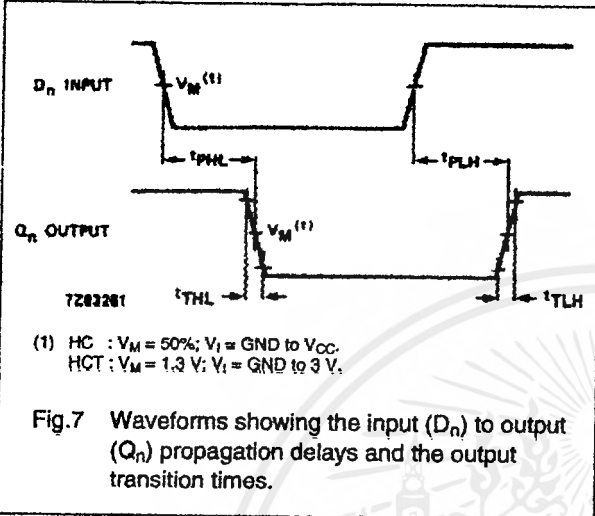
GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HCT							V _{CC} (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t _{PHL} / t _{PLH}	propagation delay D _n to Q _n		17	30		38		45	ns	4.5	Fig.7
t _{PHL} / t _{PLH}	propagation delay LE to Q _n		16	32		40		48	ns	4.5	Fig.8
t _{PZL} / t _{PZL}	3-state output enable time \overline{OE} to Q _n		19	32		40		48	ns	4.5	Fig.9
t _{PHZ} / t _{PLZ}	3-state output disable time \overline{OE} to Q _n		18	30		38		45	ns	4.5	Fig.9
t _{THL} / t _{TLH}	output transition time		5	12		15		18	ns	4.5	Fig.7
t _w	LE pulse width HIGH	16	4		20		24		ns	4.5	Fig.8
t _{su}	set-up time D _n to LE	12	6		15		18		ns	4.5	Fig.10
t _h	hold time D _n to LE	4	-1		4		4		ns	4.5	Fig.10

Octal D-type transparent latch; 3-state

74HC/HCT373

AC WAVEFORMS



Octal D-type transparent latch; 3-state

74HC/HCT373

PACKAGE OUTLINES

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



DATA SHEET

For a complete data sheet, please also download:

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

74HC/HCT138

3-to-8 line decoder/demultiplexer;
inverting

Product specification
File under Integrated Circuits, IC06

September 1993

Philips
Semiconductors



PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

FEATURES

- Demultiplexing capability
- Multiple input enable for easy expansion
- Ideal for memory chip select decoding
- Active LOW mutually exclusive outputs
- Output capability: standard
- I_{CC} category: MSI

The 74HC/HCT138 decoders accept three binary weighted address inputs (A₀, A₁, A₂) and when enabled, provide 8 mutually exclusive active LOW outputs (\bar{Y}_0 to \bar{Y}_7).

The "138" features three enable inputs: two active LOW (\bar{E}_1 and \bar{E}_2) and one active HIGH (E₃). Every output will be HIGH unless \bar{E}_1 and \bar{E}_2 are LOW and E₃ is HIGH.

This multiple enable function allows easy parallel expansion of the "138" to a 1-of-32 (5 lines to 32 lines) decoder with just four "138" ICs and one inverter.

GENERAL DESCRIPTION

The 74HC/HCT138 are high-speed SI-gate CMOS devices and are pin compatible with low power Schottky TTL (LS TTL). They are specified in compliance with JEDEC standard no. 7A.

The "138" can be used as an eight output demultiplexer by using one of the active LOW enable inputs as the data input and the remaining enable inputs as strobes. Unused enable inputs must be permanently tied to their appropriate active HIGH or LOW state.

The "138" is identical to the "238" but has inverting outputs.

QUICK REFERENCE DATA

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

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PHL} /t _{PLH}	propagation delay A _n to \bar{Y}_n	C _L = 15 pF; V _{CC} = 5 V			
t _{PHL} /t _{PLH}	E ₃ to \bar{Y}_n		12	17	ns
	\bar{E}_n to \bar{Y}_n		14	19	ns
C _I	input capacitance		3.5	3.5	pF
C _{PD}	power dissipation capacitance per package	notes 1 and 2	67	67	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

∑ (C_L × V_{CC}² × f_o) = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}
For HCT the condition is V_I = GND to V_{CC} - 1.5 V

ORDERING INFORMATION

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

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 2, 3	A_0 to A_2	address inputs
4, 5	\bar{E}_1, \bar{E}_2	enable inputs (active LOW)
6	E_3	enable input (active HIGH)
8	GND	ground (0 V)
15, 14, 13, 12, 11, 10, 9, 7	\bar{Y}_0 to \bar{Y}_7	outputs (active LOW)
16	V_{CC}	positive supply voltage

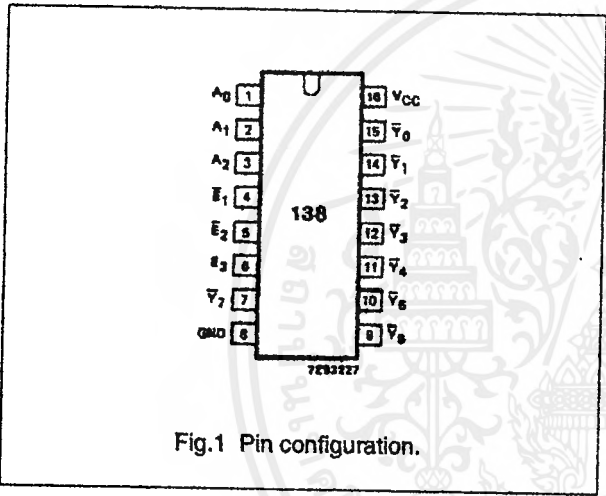


Fig.1 Pin configuration.

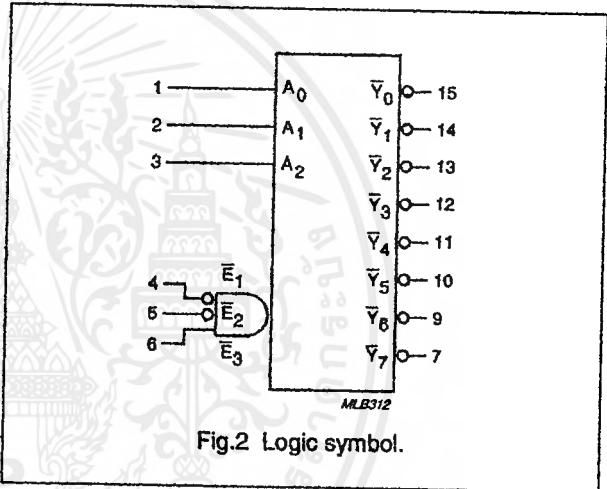


Fig.2 Logic symbol.

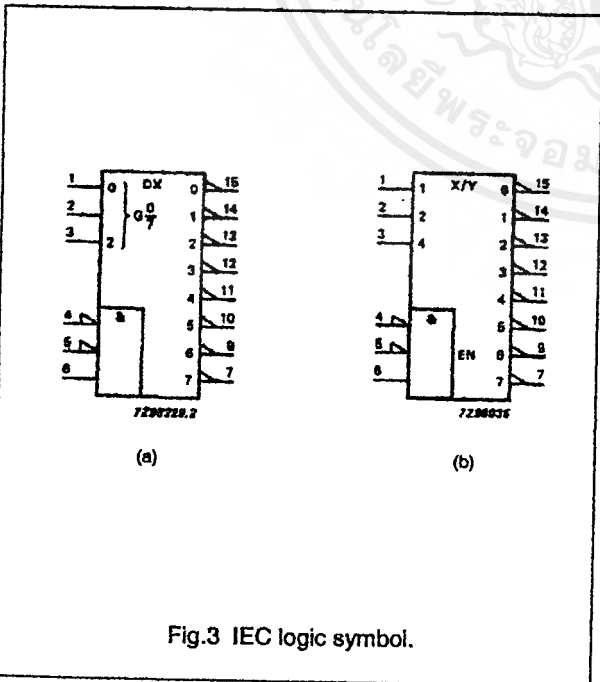


Fig.3 IEC logic symbol.

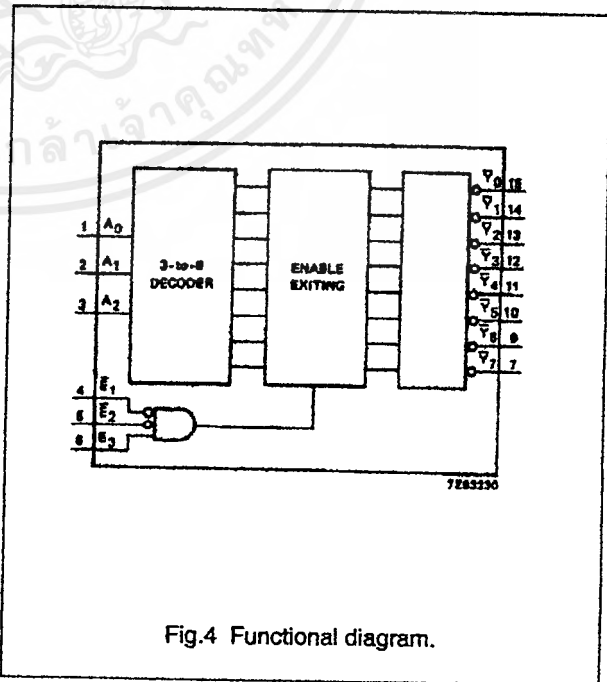


Fig.4 Functional diagram.

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

DC CHARACTERISTICS FOR 74HC

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

Output capability: standard

I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

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

SYMBOL	PARAMETER	T _{amb} (°C)						UNIT	TEST CONDITIONS		
		74HC							V _{CC} (V)	WAVEFORMS	
		+25			-40 to +85		-40 to +125				
		min.	typ.	max.	min.	max.	min.				max.
t _{PHL} / t _{PLH}	propagation delay A _n to \bar{Y}_n		41 15 12	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
t _{PHL} / t _{PLH}	propagation delay E ₃ to \bar{Y}_n		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.6
t _{PHL} / t _{PLH}	propagation delay \bar{E}_n to \bar{Y}_n		47 17 14	150 30 26		190 38 33		225 45 38	ns	2.0 4.5 6.0	Fig.7
t _{THL} / t _{TLH}	output transition time		19 7 6	75 15 13		95 19 16		110 22 19	ns	2.0 4.5 6.0	Figs 6 and 7

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

DC CHARACTERISTICS FOR 74HCT

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

Output capability: standard

I_{CC} category: MSI

Note to HCT types

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

INPUT	UNIT LOAD COEFFICIENT
A _n	1.50
\bar{E}_n	1.25
E ₃	1.00

AC CHARACTERISTICS FOR 74HCT

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

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HCT									V _{CC} (V)	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} /t _{PLH}	propagation delay A _n to \bar{Y}_n		20	35		44		53	ns	4.5	Fig.6	
t _{PHL} /t _{PLH}	propagation delay E ₃ to \bar{Y}_n		18	40		50		60	ns	4.5	Fig.6	
t _{PHL} /t _{PLH}	propagation delay \bar{E}_n to \bar{Y}_n		19	40		50		60	ns	4.5	Fig.7	
t _{THL} /t _{TLH}	output transition time		7	15		19		22	ns	4.5	Figs 6 and 7	

3-to-8 line decoder/demultiplexer; inverting

74HC/HCT138

AC WAVEFORMS

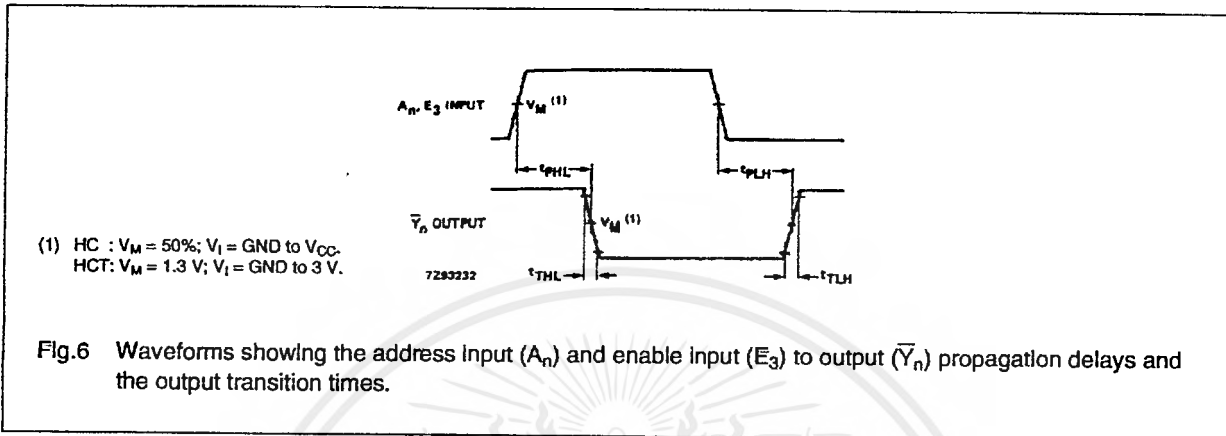


Fig.6 Waveforms showing the address input (A_n) and enable input (E_3) to output (\bar{Y}_n) propagation delays and the output transition times.

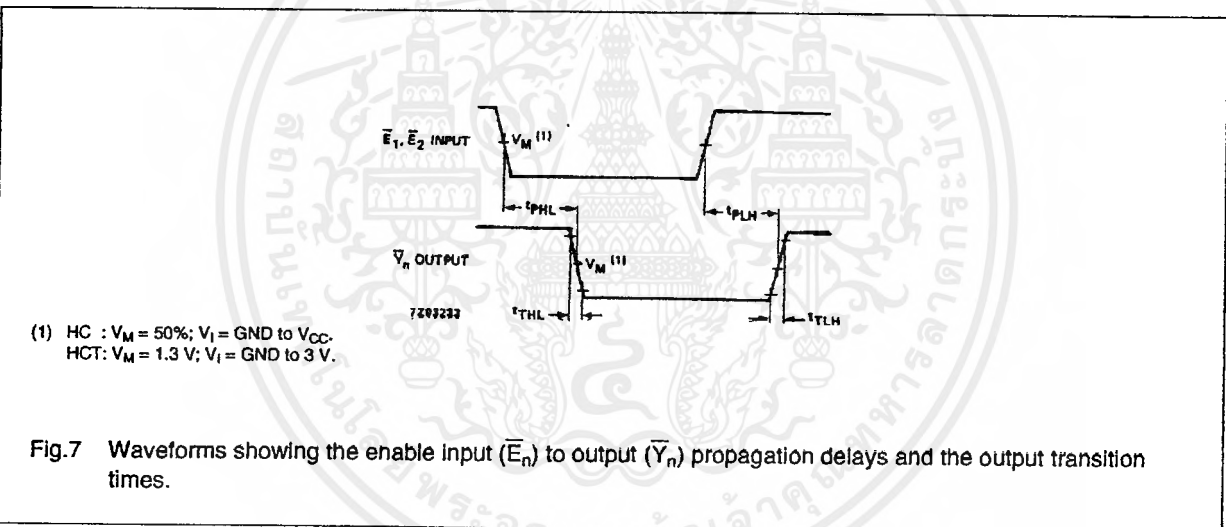


Fig.7 Waveforms showing the enable input (\bar{E}_n) to output (\bar{Y}_n) propagation delays and the output transition times.

PACKAGE OUTLINES

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