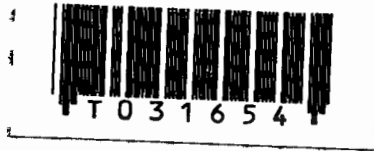


สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องแปลงสัญญาณอนาลอกเป็นดิจิทัลส่งผ่านเครื่องพิมพ์



โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาฟิสิกส์ประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

พ.ศ. 2540

พ.ศ. 2540

เลขหมู่.....

เลขทะเบียน..... 31654

วัน, เดือน, ปี 19 พ.ค. 2541

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRINTING A/D CONVERTER



MR. SANCHAYOT SAKULJITTACHAROEN
MR. ANANTACHAI NAIJIT

**A Special Project Submitted in Partial Fulfillment of the
Requirement for the Bachelor Degree of Science**

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1997

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ

เครื่องแปลงสัญญาณเอนาลอกเป็นดิจิทัลส่งผ่าน
เครื่องพิมพ์

โดย

นายสัญญาเขต

สกุลจิตตเจริญ

นายอนันตชัย

นัยจิตร

ภาควิชา

ฟิสิกส์ประยุกต์

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์วิชิต

ศิริโชติ

อาจารย์ชนภรณ์

ลีลาวัฒนานนท์

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อนุมัติให้นับโครงการพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

(รองศาสตราจารย์สุรพล รักวิจัย)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

คณะกรรมการโครงการพิเศษ

(ผู้ช่วยศาสตราจารย์วิชิต ศิริโชติ)

ประธานกรรมการ

(อาจารย์บัณฑิต คำรงค์ศักดิ์)

กรรมการ

(อาจารย์ชนภรณ์ ลีลาวัฒนานนท์)

กรรมการ

ภักธีรา กิตติเดชาชาญ

(อาจารย์ภักธีรา กิตติเดชาชาญ)

กรรมการ

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงการพิเศษ

เครื่องแปลงสัญญาณแอนาลอกเป็นดิจิตอลส่งผ่าน
เครื่องพิมพ์

โดย

นายสัญญาชิต

สกุลจิตตเจริญ

นายอนันตชัย

นัยจิตร

ภาควิชา

ฟิสิกส์ประยุกต์

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์วิชิต

ศิริโชค

อาจารย์ธนภรณ์

ธิดาวัดนามนท์

ปีการศึกษา

2540

บทคัดย่อ

โครงการนี้ได้มีการพัฒนาการเชื่อมโดยตรงระหว่างตัวแปลงข้อมูลแอนาลอกเป็นดิจิตอลกับเครื่องพิมพ์แบบเซนทรอนิกส์ ซึ่งตัวแปลงข้อมูลนี้ประกอบไปด้วยวงจร 2 วงจร วงจรแรกเป็นวงจรแปลงข้อมูลแอนาลอกเป็นดิจิตอลแบบสไลด์ไคโดยใช้ไอซี CA3162 ซึ่งถูกควบคุมโดยไมโครคอมพิวเตอร์ 89C2051 วงจรส่วนที่สองเป็นรีจิสเตอร์บัฟเฟอร์ที่มีการเชื่อมต่อแบบเซนทรอนิกส์และรีเลย์ไทม์คล็อกได้ถูกควบคุมโดยไมโครคอมพิวเตอร์ 89C51 วงจรส่วนแรกได้ให้อินพุตกระแสตรง 8 ช่อง ตัวแปลงข้อมูลได้มีช่วงโวลต์เตจอินพุตตั้งแต่ -10mV ถึง $+100\text{mV}$ ซึ่งมีความละเอียดถึง 10 บิต บัฟเฟอร์ของตัวแปลงข้อมูลได้ทริกเกอร์ทุกๆ 5 นาที จากนั้นตัวแปลงข้อมูลจึงส่งข้อมูลแบบอนุกรมไปยังรีจิสเตอร์บัฟเฟอร์ เมื่อบัฟเฟอร์ได้จัดการกระบวนการทางข้อมูลแล้วจึงส่งข้อมูลไปยังเครื่องพิมพ์ ซึ่งเครื่องพิมพ์สามารถพิมพ์ได้ทั้งข้อมูลแบบเทกซ์และแบบกราฟ ประโยชน์ของเครื่องแปลงสัญญาณแอนาลอกเป็นดิจิตอลส่งผ่านเครื่องพิมพ์สามารถใช้สำหรับการวัดปริมาณทางฟิสิกส์โดยที่ไม่ต้องใช้ พีซี

Special Project Title	Printing A/D Converter
Advisor	Mr.Sanchayot Sakujittacharoen Mr.Anantachai Naijit
Spacial Project Advisors	Asst.Prof.Wichit Sirichote MissThanapron Leelawattananon
Department	Applied Physics Ladkrabang
Academic Year	1997

Abstract

An analog-to-digital converter connecting directly to a centronic printer has been developed. The converter consists of two circuits. The first is a dual slope integrating A/D, CA3162 controlled by a single chip microcomputer, 89C2051. The second circuit is a buffer register having centronic interfacing and real time clock controlled by an MPU, 89C51. The front-end provides 8 channels single-end DC input. The converter is capable of providing input voltage ranging from -10mV to +1000mV with the resolution of 10-Bit. The buffer triggers the converter with a specified time duration, say every 5 min, then the converter sends a serial data to the buffer register. The processed data performed by the buffer is then sent to a printer. Additionally, the reported can be printed both in text data and graph. Application of the Printing A/D converter can be used for the measurement of physical quantity where a PC is not available.

กิติกรรมประกาศ

โครงการพิเศษชิ้นนี้เป็นผลสำเร็จลงได้ ต้องขอขอบคุณ บิดา มารดา ซึ่งคอยให้กำลังใจ และให้ความอุปการะให้ได้รับการศึกษาจนถึงระดับอุดมศึกษา อาจารย์วิชิต ศิริโชติ ผู้ให้คำปรึกษาและจัดหาอุปกรณ์ในการทำโครงการ คุณเสรี รักวงษ์ และคุณเนพชล สุภรศรี ผู้อุปการะที่เครื่องพิมพ์ คณะกรรมการทุกท่านที่ช่วยตรวจและแก้ไขรายงานที่เป็นฉบับสมบูรณ์นี้และสุดท้ายความช่วยเหลือที่ได้รับจากภาควิชา ที่ให้อุปกรณ์และเครื่องมือวัดต่างๆจนจะทำการทดลอง ขอขอบคุณกลุ่มบุคคลเหล่านี้เป็นอย่างมาก



สารบัญ

หน้า

บทคัดย่อปัญหาพิเศษภาษาไทย	ก
บทคัดย่อปัญหาพิเศษภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ก
สารบัญ	ง
สารบัญตาราง	ณ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ที่มาของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการดำเนินงาน	1
1.4 ประโยชน์ที่ได้รับ	2
บทที่ 2 ทฤษฎีเบื้องต้น	3
2.1 ไมโครคอนโทรลเลอร์เบอร์ 89C51	3
2.2 ไมโครคอนโทรลเลอร์เบอร์ 89C2051	7
บทที่ 3 อุปกรณ์เชื่อมต่อและการอินเตอร์เฟซ	9
3.1 การแปลงสัญญาณอนาลอกเป็นดิจิตอล	9
3.2 พอร์ตเครื่องพิมพ์	15
3.3 นาฬิกาฐานเวลา	18
3.4 จอภาพแสดงผล LCD	21
บทที่ 4 การออกแบบและการทำงานของวงจร	31
4.1 การทำงานของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล	31
4.2-การทำงานของวงจรรับข้อมูลส่งผ่านเครื่องพิมพ์	32
บทที่ 5 การทดลองและผลการทดลอง	37
บทที่ 6 สรุปและข้อเสนอแนะ	43
6.1 บทสรุป	43
6.2 ข้อเสนอแนะ	43

ภาคผนวก ก. โปรแกรมควบคุมสัญญาณอนาล็อกเป็นดิจิทัล

ภาคผนวก ข. โปรแกรมควบคุมเครื่อง Printing A/D Converter

ภาคผนวก ค. Data Sheet

เอกสารอ้างอิง

ประวัติผู้เขียน



สารบัญตาราง

	หน้า
ตารางที่ 2.1 ฟังก์ชันของพอร์ตP3	6
ตารางที่ 3.2.1 แสดงขาของเครื่องพิมพ์แบบขนาน	15
ตารางที่ 3.2.2 แสดงขาของเครื่องพิมพ์แบบอนุกรม	16
ตารางที่ 3.2.3 แสดงขาของเครื่องพิมพ์แบบเซนทรอนิกส์	17
ตารางที่ 3.4.1 แสดงคำสั่งของ LCD MODULE	24



สารบัญรูป

	หน้า
รูปที่ 3.1.1 วงจร A/D Converter แบบสไลป์คู้	10
ก) แสดงบล็อกไดอะแกรม	10
ข) เอาท์พุทของวงจรอินทิเกรเตอร์เมื่อเทียบกับเวลา	10
รูปที่ 3.1.2 ก) แสดงวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลแบบสไลป์คู้พร้อมส่วนออโค-ซีโร	11
ข) แสดงชนิดของสัญญาณแรงดันไฟฟ้าที่เอาท์พุทของโอปแอมป์ A_2 สำหรับสัญญาณอินพุทเป็นลบ และสัญญาณอ้างอิงเป็นบวก	11
รูปที่ 3.1.3 แสดงการขจัดสัญญาณในโหมดคอนเวอร์ชันของวงจรสัญญาณอนาลอกเป็นสัญญาณดิจิตอลแบบสไลป์คู้อินทิเกรชันพร้อมด้วยส่วนออโค-ซีโร ($t_2 = T$)	14
รูปที่ 3.3.1 แผนภาพหน่วยการทำงานภายในไอซี DS 1202.	18
รูปที่ 3.3.2 แผนภาพสัญญาณเวลาของการติดต่อแบบ Single-byte transfer	20
รูปที่ 3.3.3 วงจรอย่างง่ายเฉพาะส่วนวงจรอินเตอร์เฟสไอซี DS 1202	20
รูปที่ 3.4.1 การกระทำการเขียน	22
รูปที่ 3.4.2 การกระทำการอ่านอนุกรม DMC (DMC Series)	22
รูปที่ 3.4.3 แสดงรูปแบบของตัวอักษร	30
รูปที่ 4.1 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล	31
รูปที่ 4.2.1 วงจรรับข้อมูลส่งผ่านเครื่องพิมพ์	33
รูปที่ 4.2.2 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับจอแสดงผล LCD	34
รูปที่ 4.2.3 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ real time clock DS1202	34
รูปที่ 4.2.4 สัญญาณการสื่อสารระหว่างไมโครคอนโทรลเลอร์กับเครื่องพิมพ์	35
รูปที่ 4.2.5 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับเครื่องพิมพ์.	35
รูปที่ 4.2.6 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับสวิทช์	36
รูปที่ 4.2.7 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับลำโพง	37

บทที่ 1
บทนำ

1.1 ที่มาของปัญหา

ในปัจจุบันนี้เรามีความต้องการตรวจวัดปริมาณทางฟิสิกส์ (Physical Quantity) เพื่อนำมาทำการทดลอง, วิจัย, ตรวจสอบ ซึ่งปริมาณทางฟิสิกส์ที่เราจะนำมาตรวจสอบนั้นมีด้วยกันมากมาย เช่น การวัดค่ากระแส(current sensor), การวัดอุณหภูมิ(temperature sensor), การวัดแรงดันไฟฟ้า(voltage sensor) ภายใต้สภาวะแวดล้อมต่างๆ

เนื่องจากเราไม่สามารถที่จะคอยเฝ้าตรวจการได้อย่างตลอดเวลา ด้วยเหตุนี้เราจึงต้องอาศัยระบบการสื่อสารแบบใช้สายระยะใกล้เชื่อมต่อกับเครื่องพิมพ์เพื่อให้เครื่องพิมพ์แสดงข้อมูลที่ได้จากการตรวจวัด และเพื่อความสะดวกในการเคลื่อนย้ายและติดตั้งในสถานที่ต่างๆเราจึงได้ทำการออกแบบระบบดังกล่าวให้มีขนาดเล็กกะทัดรัดพกพาสะดวก

1.2 จุดประสงค์ของโครงการ

1. เพื่อศึกษาปริมาณทางฟิสิกส์และระบบตรวจวัดปริมาณทางฟิสิกส์
2. เพื่อศึกษาระบบ Telemetry
3. เพื่อศึกษาการออกแบบวงจรและเชื่อมต่อปริมาณทางฟิสิกส์กับไมโครคอนโทรลเลอร์
4. เพื่อศึกษาหลักการสื่อสารเพื่อใช้ในการติดต่อกับไมโครคอนโทรลเลอร์
5. เพื่อให้รู้จักการแก้ปัญหาจากสิ่งที่ศึกษา

1.3 ขอบเขตการดำเนินงาน

โครงการในส่วนแรกนี้ เราจะทำการสร้างและพัฒนาตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลส่งออกเครื่องพิมพ์ โดยอาศัยหลักการสื่อสารแบบใช้สายในระยะใกล้(RS-232C) ส่งข้อมูลไปให้เครื่องพิมพ์ทำการพิมพ์ข้อมูลออกมาพร้อมทั้งออกแบบให้มีการแสดงผลข้อมูลแบบตามเวลาจริง(Real Time)

ในส่วนที่สองเราจะทดลองเชื่อมต่อปริมาณทางฟิสิกส์เข้ากับตัววัดปริมาณทางฟิสิกส์ เพื่อทดสอบและพัฒนาซอฟต์แวร์สำหรับไมโครคอมพิวเตอร์ให้เหมาะสมกับปริมาณที่จะนำมาวัด ซึ่งปริมาณทางฟิสิกส์ที่จะนำมาตรวจสอบนั้น ได้แก่ การวัดอุณหภูมิตามจุดต่างๆ

ในส่วนสุดท้ายเราจะทำการพัฒนาให้ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลส่งออกเครื่องพิมพ์สามารถที่จะแสดงผลเป็นลักษณะเป็นกราฟโดยเป็นกราฟระหว่างปริมาณทางฟิสิกส์กับเวลา

1.4 ประโยชน์ที่ได้รับ

1. สามารถตรวจวัดปริมาณทางฟิสิกส์ต่างๆ ได้อย่างถูกต้อง
2. สามารถตรวจวัดได้ตามเวลาจริง(Real Time)
3. สามารถเชื่อมต่อกับเครื่องพิมพ์ได้



บทที่ 2

ทฤษฎีเบื้องต้น

2.1 ไมโครคอนโทรลเลอร์เบอร์ 89C51

2.2.1 คุณสมบัติพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ 89C51

1. เป็นผลิตภัณฑ์ที่ใช้แทนไมโครคอนโทรลเลอร์ตระกูล MCS-51TM ได้
2. มีหน่วยความจำเป็นแบบ Flash memory ขนาด 4 กิโลไบต์ สามารถทำการ โปรแกรมซ้ำได้
3. ใช้ไฟเลี้ยงได้ตั้งแต่ 2.7 โวลต์ ถึง 6 โวลต์
4. ออสซิลเลเตอร์สามารถป้อนได้ตั้งแต่ 0 เฮิรตซ์ ถึง 20 กิโลเฮิรตซ์
5. สามารถล๊อคโปรแกรมได้สามระดับ
6. หน่วยความจำแรมภายในขนาด 128 ไบต์
7. พอร์ตอินพุต/เอาต์พุตที่สามารถโปรแกรมได้ 32 บิต
8. คำนับและตัวตั้งเวลาขนาด 16 บิต 2 ตัว
9. มีแหล่งอินเตอร์รัพท์ 5 แหล่ง
10. มีพอร์ตอนุกรมสามารถโปรแกรมได้
11. มีโหมด Low Power Idle และ Power Down

AT89C51 เป็นไมโครคอมพิวเตอร์ขนาด 8 บิต ทำจาก CMOS ใช้ไฟเลี้ยงต่ำ และมีหน่วยความจำขนาด 4 กิโลไบต์ เป็นแบบ Flash Programmable and Erasable Read Only Memory (PEROM). AT89C51 สร้างด้วยเทคโนโลยี Atmel's high density nonvolatile memory และสามารถใส่คำสั่ง MCS-51TM กับขาที่ต่อออกมาตามมาตรฐานทางอุตสาหกรรมได้ หน่วยความจำโปรแกรมสามารถโปรแกรมซ้ำได้ เมื่อรวมซีพียู 8 บิตกับ Flash ในชิป แล้ว AT89C51 เป็นไมโครคอมพิวเตอร์ที่มีความสามารถสูงซึ่งมีประโยชน์มากมายในการควบคุม

AT89C51 มีลักษณะเฉพาะตามมาตรฐานดังต่อไปนี้ มี Flash 4 กิโลไบต์, แรม 128 ไบต์, อินพุต/เอาต์พุต 32 บิต, ตัวนับและตัวตั้งเวลา 16 บิต 2 ตัว, แหล่งกำเนิดอินเตอร์รัพท์ 5 แหล่ง, พอร์ต อณุกรมแบบ full duplex และมีวงจรนาฬิกาจับวงจรออสซิลเลเตอร์ในชิป นอกจากนี้แล้ว AT89C51 ได้

ถูกออกแบบให้เป็นแบบสแตติกลอจิกสำหรับการทำงานความถี่ที่ต่ำลงไปถึงศูนย์ และสนับสนุนการเลือกโหมด การรักษากำลัง โหมดIDLE จะหยุดการทำงานของ CPU ขณะที่RAM , ตัวนับและตัวตั้งเวลา , พอร์ตอนุกรม และระบบอินเทอร์รัพท์ ยังคงทำงานต่อไป POWER DOWN MODE จะรักษาข้อมูลที่อยู่ใน RAM แต่จะหยุดออกซิเลเตอร์ ทำให้ชิพตัวอื่น ๆ หยุดทำงานจนกระทั่งทำการรีเซตฮาร์ดแวร์

การอธิบายขาของ AT89C51

V_{cc}

ไฟเลี้ยง(Supply voltage)

GND

กราวด์

PORT 0

พอร์ต0 เป็นพอร์ตอินพุต/เอาต์พุตสองทิศทางขนาด 8 บิตแบบ open drain พอร์ตเอาต์พุตแต่ละขาสามารถซิงค์อินพุต TTL ได้ 8 ตัว

เมื่อ 1 ได้ถูกเขียนไปที่พอร์ต 0 สามารถใช้เป็นอินพุตอิมพีแดนซ์สูงๆได้ ดังนั้นพอร์ต 0 สามารถถูกสร้างเป็นบัสแอสแตเรส/ข้อมูลไบท์ต่ำระหว่างการเข้าถึงหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลภายนอก ในโหมดนี้ P0 มีพุลอัพภายใน ดังนั้นพอร์ต 0 ได้รับรหัสข้อมูล(code bytes)ระหว่าง Flash Programming และเอาต์พุตรหัสข้อมูลระหว่างการเปลี่ยนแปลงโปรแกรม

PORT 1

พอร์ต 1 เป็นพอร์ตอินพุต/เอาต์พุตสองทิศทางขนาด 8 บิตโดยมีการพุลอัพภายใน บัฟเฟอร์เอาต์พุตของพอร์ต 1 สามารถซิงค์/ซอร์สอินพุต TTL ได้ 4 ตัว เมื่อ 1 ได้ถูกเขียนไปที่ขาพอร์ต1 แล้วมันได้ถูกดึงให้สูงขึ้นโดยพุลอัพภายในและสามารถถูกใช้เป็นอินพุต ขณะที่อินพุต ขาพอร์ต 1 ซึ่งถูกดึงให้ต่ำลงจะให้กระแสซอร์ส(I_{OL})เพราะว่าพุลอัพภายใน ดังนั้นพอร์ต 1 ได้รับไบท์แอสแตเรสไบท์ต่ำระหว่าง Flash Programming และการเปลี่ยนแปลงโปรแกรม

PORT2

พอร์ต 2 เป็นพอร์ตอินพุท/เอาต์พุตสองทิศทางขนาด 8 บิตโดยมีการพูลอัพ บัฟเฟอร์เอาต์พุตของพอร์ต 2 สามารถซิงค์/ซอร์สอินพุท TTL ได้ 4 ตัว เมื่อ 1 ได้ถูกเขียนไปที่ขาพอร์ต 1 แล้วมันได้ถูกดึงให้สูงขึ้นโดยพูลอัพภายในและสามารถถูกใช้เป็นอินพุท ขณะที่พอร์ต 1 ขาพอร์ต 1 ซึ่งถูกดึงให้ต่ำลงจะให้กระแสชอร์ต(I_{IL})เพราะว่าพูลอัพภายใน

พอร์ต 2 ปล่องไบท์แอสเครสไบท์สูงระหว่างการนำจากหน่วยความจำโปรแกรมภายนอกและระหว่างการถึงหน่วยความจำข้อมูลภายนอกซึ่งใช้แอสเครส 16 บิต(MOVX @ DPTR) ในที่นี้พอร์ต 2 ใช้พูลอัพเมื่อปล่อง 1 ระหว่างการเข้าถึงหน่วยความจำข้อมูลภายนอกซึ่งใช้แอสเครส 8 บิต(MOVX @ RI) พอร์ต 2 ได้ปล่องข้อมูลที่อยู่ใน P2 Special Function Register ดังนั้นพอร์ต 2 ได้รับบิตแอสเครสไบท์สูงและสัญญาณควบคุมบางสัญญาณระหว่าง Flash Programming และการเปลี่ยนแปลง

PORT2

พอร์ต 2 เป็นพอร์ตอินพุท/เอาต์พุตสองทิศทางขนาด 8 บิตโดยมีการพูลอัพ บัฟเฟอร์เอาต์พุตของพอร์ต 2 สามารถซิงค์/ซอร์สอินพุท TTL ได้ 4 ตัว เมื่อ 1 ได้ถูกเขียนไปที่ขาพอร์ต 1 แล้วมันได้ถูกดึงให้สูงขึ้นโดยพูลอัพภายในและสามารถถูกใช้เป็นอินพุท ขณะที่พอร์ต 1 ขาพอร์ต 1 ซึ่งถูกดึงให้ต่ำลงจะให้กระแสชอร์ต(I_{IL})เพราะว่าพูลอัพภายใน

พอร์ต 2 ปล่องไบท์แอสเครสไบท์สูงระหว่างการนำจากหน่วยความจำโปรแกรมภายนอกและระหว่างการถึงหน่วยความจำข้อมูลภายนอกซึ่งใช้แอสเครส 16 บิต(MOVX @ DPTR) ในที่นี้พอร์ต 2 ใช้พูลอัพเมื่อปล่อง 1 ระหว่างการเข้าถึงหน่วยความจำข้อมูลภายนอกซึ่งใช้แอสเครส 8 บิต(MOVX @ RI) พอร์ต 2 ได้ปล่องข้อมูลที่อยู่ใน P2 Special Function Register ดังนั้นพอร์ต 2 ได้รับบิตแอสเครสไบท์สูงและสัญญาณควบคุมบางสัญญาณระหว่าง Flash Programming และการเปลี่ยนแปลง

PORT 3

พอร์ต 3 เป็นพอร์ตอินพุท/เอาต์พุตสองทิศทางขนาด 8 บิตโดยมีการพูลอัพ บัฟเฟอร์เอาต์พุตของพอร์ต 3 สามารถซิงค์/ซอร์สอินพุท TTL ได้ 4 ตัว เมื่อ 1 ได้ถูกเขียนไปที่ขาพอร์ต 1 แล้วมันได้ถูกดึงให้สูงขึ้นโดยพูลอัพภายในและสามารถถูกใช้เป็นอินพุท ขณะที่พอร์ต 1 ขาพอร์ต 1 ซึ่งถูกดึงให้ต่ำลงจะให้กระแสชอร์ต(I_{IL})เพราะว่าพูลอัพภายใน

ดังนั้นพอร์ต 3 ได้รักษาฟังก์ชันลักษณะพิเศษต่างๆของ AT89C51 ดังรายละเอียดด้านล่างนี้

ตารางที่ 2.1 ฟังก์ชันของพอร์ตP3

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

ดังนั้นพอร์ต 3 ได้รับสัญญาณควบคุมบางสัญญาณสำหรับ Flash Programming และการเปลี่ยนแปลงการโปรแกรม

RST

อินพุทรีเซต ให้ high บนขาสำหรับ 2 แมกซ์ซีไอเกิดขณะที่ออสซิลเลเตอร์กำลังทำการรีเซตอุปกรณ์

ALE/PROG

Address Latch Enable เอาท์พุทพัลส์สำหรับการ latch ไบต์ค่าแอสเซสระหว่างการเข้าถึงหน่วยความจำภายนอก ดังนั้นขาเป็น Program pulse input (PROG) ระหว่าง Flash Programming โดยการทำงานปกติ ALE ได้ปล่อยที่อัตรา 1/6 ของความถี่ออสซิลเลเตอร์ และอาจจะใช้สำหรับการตั้งเวลาหรือการนับ อย่างไรก็ตามพัลส์ ALE 1 พัลส์ได้ถูกข้ามระหว่างการเข้าถึงหน่วยความจำข้อมูลภายนอกแต่ละครั้ง

การทำงานของ ALE จะหยุดโดยการเซตบิต 0 ของ SFR ที่ตำแหน่ง 8EH เมื่อบิตเซต ALE จะทำงานระหว่างคำสั่ง MOVX และ MOVC ในลักษณะอื่นๆ ขาจะถูกดึงให้สูงขึ้นเล็กน้อย การเซตบิต ALE ที่ไม่ทำงานจะไม่มีผลถ้าไมโครคอนโทรเลอร์อยู่ในโหมดการทำงานภายนอก

PSEN

Program Store Enable เป็นการอ่านสโตรบให้กับหน่วยความจำโปรแกรมภายนอก เมื่อ AT89C51 กำลังกระตุ้นรหัสจากหน่วยความจำโปรแกรมภายนอก และ PSEN ได้ถูกกระทำเป็นคู่ในแต่ละแมกซ์ซีไอเกิด แล้วจะได้ว่าการทำงานของ PSEN ได้ถูกข้ามระหว่างการเข้าถึงหน่วยความจำข้อมูลภายนอกในแต่ละครั้ง

EA/V_{pp}

External Access Enable EA ต้องถูกต่อลงกราวด์ในการอนุญาตให้อุปกรณ์ไปนำรหัสจากตำแหน่งหน่วยความจำโปรแกรมภายนอกเริ่มตั้งแต่ 0000H ขึ้นไปถึง FFFFH อย่างไรก็ตามถ้าได้โปรแกรมล๊อกรบิท 1 EA จะถูกเลขชี้ไปเป็นรีเซต

EA จะถูกต่อไปยัง V_{pp} สำหรับการกระทำโปรแกรมภายใน ดังนั้นขานี้ได้รับโวลต์เตจการอนุญาตการโปรแกรม 12 โวลต์(V_{pp})ระหว่าง Flash Programming

XTAL1

อินพุตไปยังการสับเปลี่ยนตัวขยายออสซิลเลเตอร์และอินพุตไปยังวงจรการทำงานของตัวนับภายใน

XTAL2

เอาต์พุตจากการสับเปลี่ยนตัวขยายออสซิลเลเตอร์

2.2 ไมโครคอนโทรลเลอร์เบอร์ 89C2051

- 1.เป็นผลิตภัณฑ์ที่ใช้แทนไมโครคอนโทรลเลอร์ตระกูล MCS-51™ ได้
- 2.มีหน่วยความจำเป็นแบบ Flash memory ขนาด 2 กิโลไบต์ สามารถทำการโปรแกรมซ้ำได้
- 3.ใช้ไฟเลี้ยงได้ตั้งแต่ 2.7 โวลต์ ถึง 6 โวลต์
- 4.ออสซิลเลเตอร์สามารถป้อนได้ตั้งแต่ 0 เฮิร์ตซ์ ถึง 24 กิโลเฮิร์ตซ์
- 5.สามารถล๊อคโปรแกรมได้สองระดับ
- 6.หน่วยความจำแรมภายในขนาด 128 ไบต์
- 7.พอร์ทอินพุต/เอาต์พุตที่สามารถโปรแกรมได้ 15 บิต
- 8.ตัวนับและตัวตั้งเวลาขนาด 16 บิต 2 ตัว
- 9.มีแหล่งอินเตอร์รัพท์ 5 แหล่ง
- 10.มีพอร์ทอนุกรม UART สามารถโปรแกรมได้
- 11.เอาต์พุตขับ LED ได้โดยตรง
- 12.มีวงจรเปรียบเทียบอนาล็อก
- 13.มีโหมด Low Power Idle และ Power Down

ส่วนคุณสมบัติอื่นๆของ AT89C2051 จะคล้ายกับคุณสมบัติของ AT89C51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

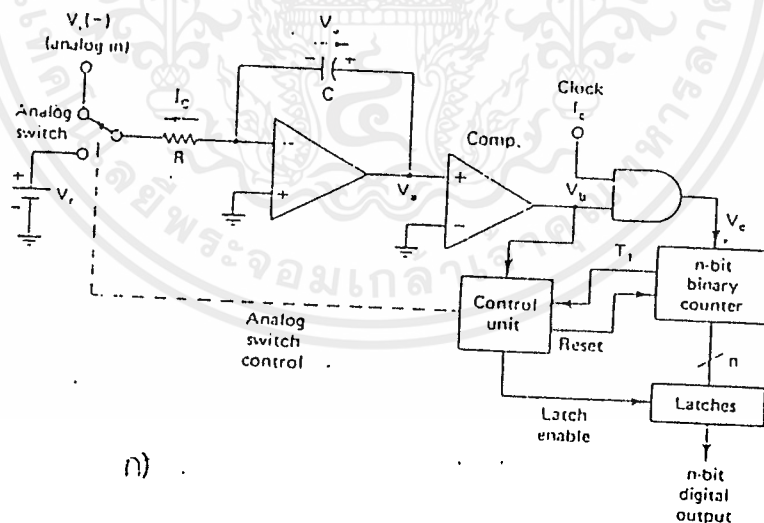
อุปกรณ์เชื่อมต่อและการอินเทอร์เฟสแบบขนาน

3.1 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบสโลปคู่ (Dual-Slope A/D converter)

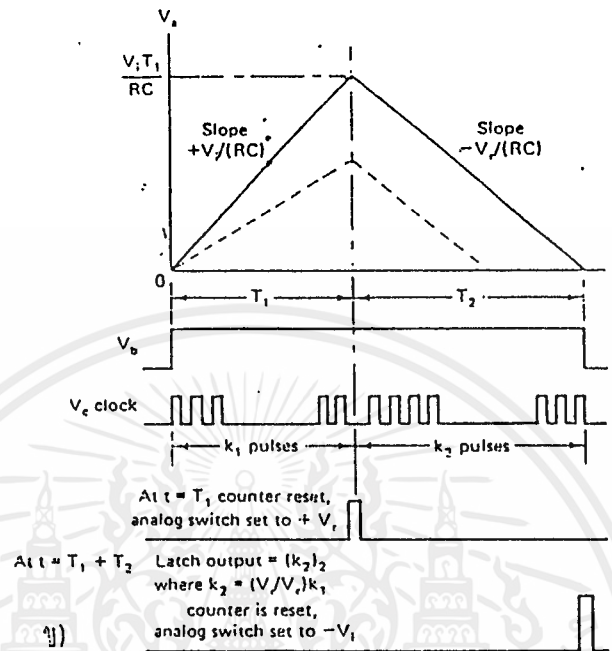
วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบสโลปคู่ อินทิเกรชันนี้ ข้อมูลที่แปลงได้เป็นจำนวนบิต หรือแบบ BCD การแปลงสัญญาณแบบสโลปคู่จะไม่เปรียบเทียบแรงดันที่ต้องการตรงๆ แต่จะนำไปเปรียบเทียบกับค่าแรงดันอ้างอิงค่าหนึ่ง โดยใช้เวลาเป็นตัวกำหนดการทำงาน การทำงานจะเริ่มต้นจาก สวิตช์ S_1 รับอินพุตเข้ามาสู่วงจรอินทิเกรชันด้วยเวลาคงที่ T_1 เมื่อครบเวลา T_1 สวิตช์ S_1 จะถูกสลับกลับมาที่แรงดันอ้างอิงที่มีศักย์ที่ตรงข้ามกับแรงดันอินพุต วงจรนับจะเริ่มนับเวลาไปจนกระทั่งเอาท์พุทของวงจรอินทิเกรเตอร์ โดยที่ T_1 เป็นเวลาที่กำหนดคงที่ และ T_2 เป็นเวลาแปรตามแรงดันอินพุต

ซึ่งความสัมพันธ์

$$T_2 = T_1 * V_{in} / V_{ref} \quad (3.1)$$



n)

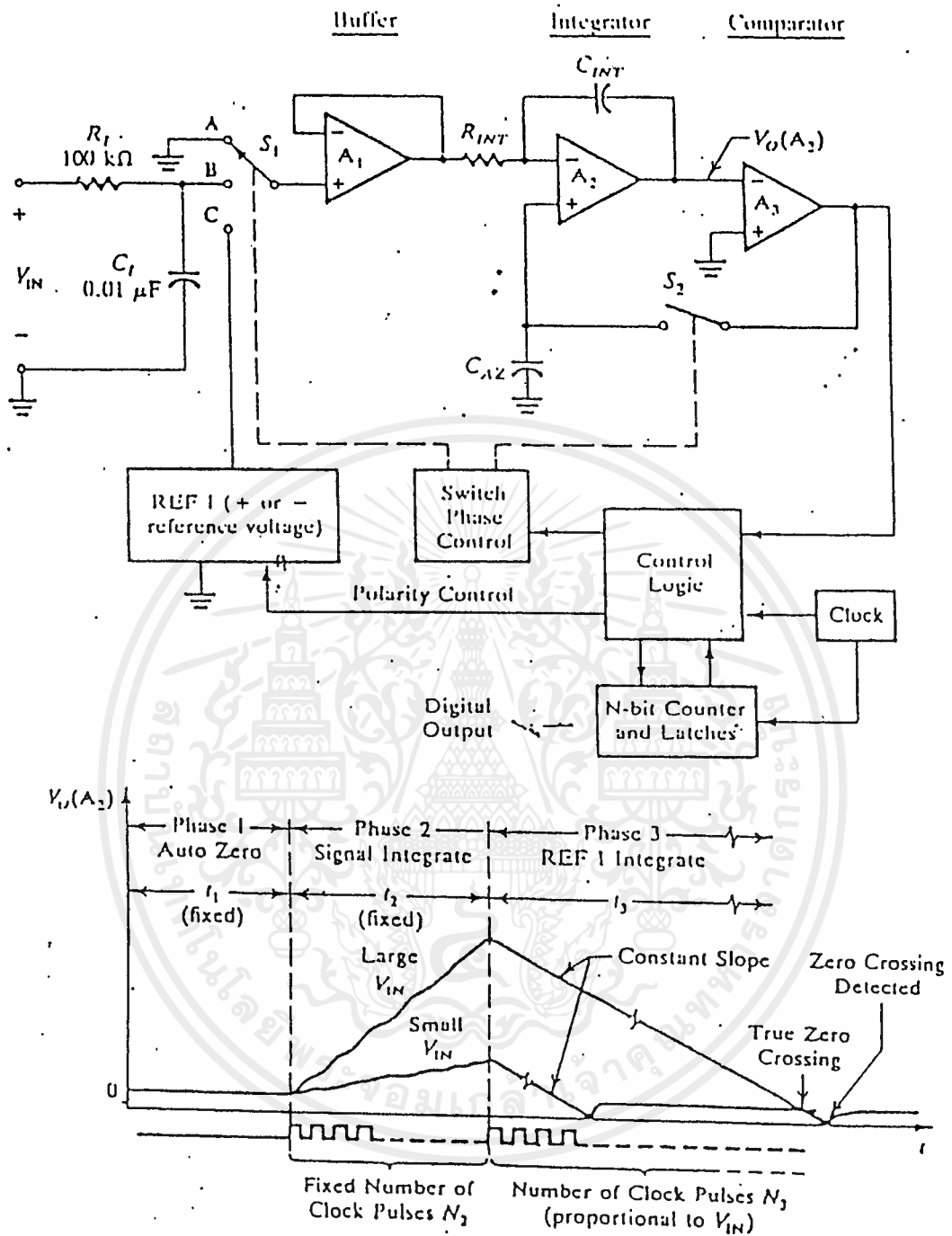


รูปที่ 3.1.1 วงจร A/D Converter แบบสโลปคู่

ก) แสดงบล็อกไดอะแกรม

ข) เอาท์พุทของวงจรอินทิเกรเตอร์เมื่อเทียบกับเวลา

ผลของการนับด้วยวงจรมับในเวลา T_2 จะอยู่ในรูปของจำนวนพัลส์ต่อโวลต์ ส่วนเอาต์พุทนั้นจะขึ้นอยู่กับการนับรวมแต่ละตัว ส่วนแสดงผลจะแสดงผลของการนับออกเป็นค่าโวลต์ หรือเป็นจำนวนบิต ถ้าผลการนับแสดงออกเป็นโวลต์จะอยู่ในลักษณะของจำนวนหลัก เช่น สามหลักครึ่ง สี่หลักครึ่ง ส่วนข้อมูลที่เป็นจำนวนบิต หรือเลขฐานสองจะเป็น 8 บิต , 12 บิต



รูปที่ 3.1.2 ก) แสดงวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลแบบสโลปคู่ พร้อมส่วนออสโคป

ข) แสดงชนิดของสัญญาณแรงดันไฟฟ้าที่เอาต์พุตของแอมป์ A_2 สำหรับ สัญญาณอินพุตเป็นลบ และสัญญาณอ้างอิงเป็นบวก

จากวงจรสโลปคู่ที่ได้กล่าวมาแล้วได้มีการพัฒนางจรดังกล่าวโดยเพิ่มส่วนออโต-ซีโร (Auto-Zero)

ซึ่งใช้ออปแอมป์เป็นตัวเปรียบเทียบกับแรงดันแบบซีโร-ครอสซึ่ง มีการชาร์จสัญญาณจากตัวเก็บประจุในวงจรอินทิเกรเตอร์ มีส่วนที่เป็นสัญญาณนาฬิกา และส่วนที่เป็นวงจรรีบ ส่วนที่มีการเพิ่มเติมเข้าไปใหม่ คืออนาล็อกสวิตช์และวงจรออโต-ซีโร เพื่อลดแรงดันออฟเซตคิซีซีที่เกิดจากวงจรอินทิเกรเตอร์ และวงจรเปรียบเทียบกับแรงดัน วงจรแปลงสัญญาณแบบนี้สามารถแปลงสัญญาณมีค่าความถูกต้องถึงหกหลักครึ่ง (20 บิต) แต่ใช้เวลานานถึง 100 มิลลิวินาที

จากรูปที่ 3.1.2 ก) และ ข) ในส่วนของไซเกิลแรก (เฟส 1) เริ่มจากส่วนอินพุทของวงจรขยายสัญญาณแรงดันไฟฟ้าแบบเป็นบัพเฟอร์ต่อกับกราวด์ (สวิตช์ S_1 อยู่ที่ตำแหน่ง A) ขณะที่สวิตช์ S_2 จะปิดวงจรเพื่อชาร์จสัญญาณเข้าตัวเก็บประจุออโต-ซีโร C_z ด้วยเหตุนี้ผลการคำนวณค่าแรงดันออฟเซตคิซีซีที่ผิดพลาดของออปแอมป์ A_1 และ A_2 และวงจรเปรียบเทียบกับแรงดัน A_2 จะถูกเก็บไว้ใน C_z เนื่องจากวงจรเปรียบเทียบกับแรงดัน A_1 รวมอยู่ในลูป จะแก้ไขแรงดันออฟเซตที่เป็นสัญญาณรบกวนที่เกิดขึ้นเองในวงจร โดยปกติจะมีค่าประมาณ 10 ไมโครโวลต์ เวลา t ที่กำหนดค่าไว้แล้วเป็นเวลาที่ใช้ในโหมดออโต-ซีโร แต่ในกรณีอื่นอาจจะถูกเซตให้มีค่ามากกว่า 20 เท่าของเวลาการชาร์จ ที่มีผลต่อการรวมสัญญาณที่เลื่อนไปเทอมอื่นๆ

ส่วนเริ่มของเฟสที่สอง ลูปของออโต-ซีโรจะเปิดออก (สวิตช์ S_2 เปิดออก) และสวิตช์อินพุท S_1 จะต่อเข้ากับจุด B (ต่อกับสัญญาณอินพุท V_{in}) วงจรอินทิเกรเตอร์ A_2 จะอินทิเกรตสัญญาณอินพุทในเวลา t_2 ที่กำหนดค่าไว้แล้ว ในขณะเดียวกัน วงจรบัพเฟอร์ที่จุดเริ่มต้นของเฟสสอง และจะมีค่าเป็นไปตามจำนวนเลขที่ถูกกำหนดไว้ของพัลส์ของสัญญาณนาฬิกา ที่จุดสุดท้ายของ t_2 อยู่ในตำแหน่งจุด C ซึ่งถูกกำหนดไว้กับแหล่งจ่ายสัญญาณแรงดันอ้างอิง จะมีลักษณะสัญญาณที่ตรงข้ามกับสัญญาณอินพุท ด้วยเหตุนี้ระหว่างที่อยู่ในเฟสสอง ความสำคัญของการคำนวณ V_{in} จะมีค่าที่เซตค่าแรงดันอ้างอิง ให้ความสำคัญเดียวกัน

เนื่องจากแรงดันอ้างอิงได้ถูกกำหนดค่าไว้แล้ว สโลปของสัญญาณเอาต์พุทของออปแอมป์ A_2 จะมีผลคือนิเกรตสัญญาณอินพุทกลับไปยังศูนย์ตามรูป 3.1.1 สิ่งสำคัญมากที่ใช้ในส่วน of แรงดันอ้างอิง ต้องมีค่ามากกว่าค่ามากที่สุดของสัญญาณอินพุท วงจรเปรียบเทียบกับแรงดัน A_1 เป็นแบบ ซีโร-ครอสซึ่ง สัญญาณอินพุทจนจบสามเฟส สัญญาณอินพุทจนจบสามเฟสและหยุดวงจรนับ และรีเซตที่จุดเริ่มต้นของเฟสสาม ด้วยเหตุนี้ สัญญาณแรงดันไฟฟ้าที่เปลี่ยนแปลงไปที่เอาต์พุทของออปแอมป์ A_2 เป็นจุดสุดท้ายของ t_2 มีค่า

$$V_{0(2)}(t=t_2) = -[V_{in}/(R_{INT} \times C_{INT})] \dots\dots\dots(3.2)$$

และที่จุดสุดท้ายของ t_2

$$V_{0(2)}(t=t_2) = -[V_{REF}/(R_{INT} \times C_{INT}) \times t_2] \dots\dots\dots(3.3)$$

เนื่องจากค่าแรงดันไฟฟ้าที่เปลี่ยนแปลงเท่ากัน, ค่าเวลา t_2 มีค่า (หรือค่า N_2 คือ จำนวนพัลส์ทั้งหมดในระหว่าง t_2)

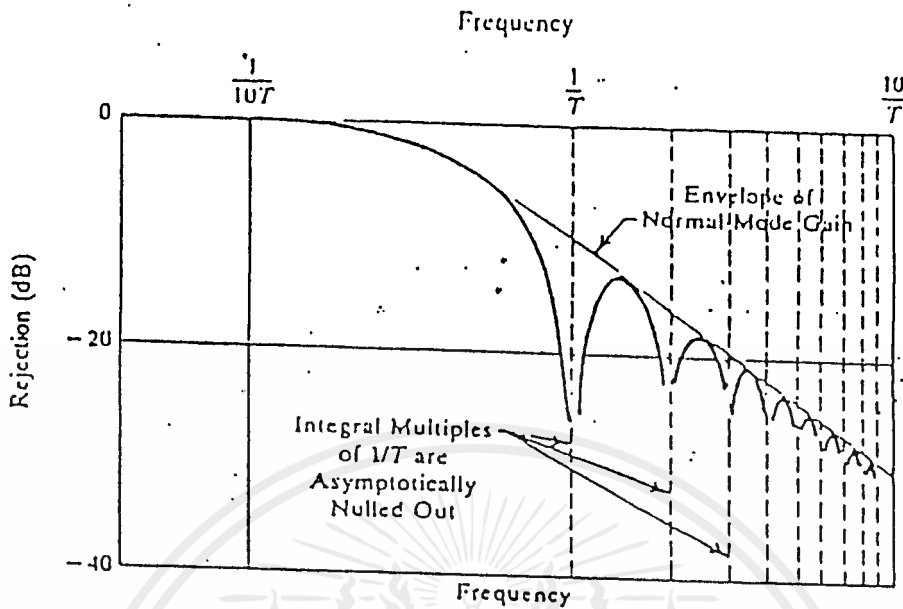
$$t_2 = V_{in} \times (t_2 / V_{REF}) \dots\dots\dots(3.4)$$

ค่าในเทอมของจำนวนพัลส์

$$N_2 = V_{in} \times (N_2 / V_{REF}) \dots\dots\dots(3.5)$$

จากสมการจะพบว่าความถูกต้องของการแปลงข้อมูลไม่ได้ขึ้นอยู่กับค่าความถูกต้องของแรงดันอ้างอิง และความถูกต้องของสัญญาณนาฬิกาเท่านั้น แต่ยังขึ้นกับเสถียรภาพของตัวเก็บประจุในวงจรอินทิเกรเตอร์ C_{int} ที่เปลี่ยนแปลง ยกเว้นค่าที่ไม่เปลี่ยนแปลงในระหว่างการแปลงข้อมูลในแต่ละไซเคิล

ข้อดีของวงจรแปลงสัญญาณแบบนี้ การอินทิเกรตสัญญาณคาบเวลา t_2 สามารถเซตให้เท่ากับค่าจำนวนเต็มของสัญญาณอินพุต ที่มีความถี่ผสมกันหลายความถี่ จะถูกอินทิเกรตเพื่อให้ได้ค่าเฉลี่ยเป็นศูนย์ที่เอาท์พุทของวงจรอินทิเกรเตอร์ การขจัดนี้เรียกว่า โหมดนอร์มัล (normal mode or line frequency rejection) แสดงในรูป 3.1.3



รูปที่ 3.1.3 แสดงการขจัดสัญญาณในโหมดคอนอร์มอลของวงจรสัญญาณอนาลอกเป็นสัญญาณดิจิทัลแบบสโลปคู่อินทิเกรชันพร้อมด้วยส่วนออร์โต-ซีโร ($l_2 = T$)

แม้ว่าวงจรนี้จะสามารถแปลงสัญญาณอนาลอกในย่านความถี่ต่ำได้ดี แต่ก็ยังมีข้อผิดพลาดที่เกิดจากกระแสรั่วที่สวิตช์ S_2 มีค่าประมาณน้อยกว่า 10 พิโคแอมป์ ซึ่งส่งผลไปลดค่าแรงดันไฟฟ้าที่ถูกเก็บไว้ใน C_2 และ C_{in} และปัญหาที่เกิดจากสวิตช์ S_1 และสวิตช์ S_2 ก็ยังมีผลต่อออปแอมป์ A_1 และ A_2 ทำให้เกิดความไม่เป็นเชิงเส้นที่เอาต์พุต ซึ่งเหมือนกับผลของค่าสโล-เรท ในออปแอมป์ A_1 ขณะที่สวิตช์ S_1 ได้จากแหล่งจ่ายแรงดันอ้างอิงกราวด์ ตัวเก็บประจุ C_{in} และ C_2 จะมีค่าการดักกลืนไดอิเล็กตริกเกิดขึ้นแต่เราสามารถทำให้ลดลงได้โดยการใช้ตัวเก็บประจุแบบโพลีสไตลีน โพลีโพรพิลีน นอกจากนี้ยังมีการดีเลย์ (Delay) ในระหว่างการทำซีโร-ครอสซึ่งที่เอาต์พุตของออปแอมป์ A_2 กับการทำซีโร-ครอสซึ่ง โดยส่วนควบคุมลอจิก ซึ่งจะมีผลให้เกิดความแตกต่างในจำนวนของการนับ N_1 เท่ากับค่าความผิดพลาดส่วนที่เหลือใน ออร์โต-ซีโร จึงจำเป็นต้องเก็บผลรวมของค่าความผิดพลาดทั้งหมดให้มีค่าน้อยกว่า $1/2$ LSB สำหรับการแปลงข้อมูลเอาต์พุต

3.2 พอร์ตเครื่องพิมพ์

คอนเนกเตอร์ของพอร์ตเครื่องพิมพ์

มี 3 รูปแบบคือ พอร์ตขนาน(parallel port), พอร์ตอนุกรม(serial port) และคอนเนกเตอร์แบบเซนทรอนิกส์(cen-tronics printer connector)

พอร์ตเครื่องพิมพ์แบบขนาน

จะติดตั้งอยู่ที่คอมพิวเตอร์ไอบีเอ็มพีซี XT/AT และคอมแพคทีเบิลเพื่อเชื่อมต่อกับเครื่องพิมพ์ใช้คอนเนกเตอร์ชนิด D ขนาด 25 ขา มีรายละเอียดการจัดขาดังนี้

ตารางที่ 3.2.1 แสดงขาของเครื่องพิมพ์แบบขนาน

ขาที่	ลักษณะการทำงาน	คำอธิบาย
1	Input / Output	Strobe(STROBE)
2	Input / Output	Data Bit 0 (PRTD0)
3	Input / Output	Data Bit 1 (PRTD1)
4	Input / Output	Data Bit 2 (PRTD2)
5	Input / Output	Data Bit 3 (PRTD3)
6	Input / Output	Data Bit 4 (PRTD4)
7	Input / Output	Data Bit 5 (PRTD5)
8	Input / Output	Data Bit 6 (PRTD6)
9	Input / Output	Data Bit 7 (PRTD7)
10	Input	Acknowledgc(ACK)
11	Input	Busy
12	Input	Paper End / Empty (PE)
13	Input	Select (SLCT)
14	Output	Auto Feed (AUTOFDXT)
15	Output	Error / Oscillator (ERR)
16	Output	Initialize Printer (INIT)
17	Output	Select Input (SLCTIN)
18-25	GND	Ground

พอร์ตเครื่องพิมพ์แบบอนุกรม

จะมีลักษณะคล้ายกับพอร์ต RS_232C จะมีความแตกต่างในบางขาสัญญาณได้แก่ขา 17 (TTY Transmit data : TTY-TXD) และขา 25(TTY-Receive data : TTY-RXD) ถ้าหากในการทำงานไม่ได้อ้างอิงขาสัญญาณทั้งสองขานี้สามารถใช้พอร์ต RS-232 แทนได้ เพราะระดับสัญญาณเหมือนกัน

ใช้คอนเน็คเตอร์ชนิด D ขนาด 25 ขา มีรายละเอียดการจัดขาดังนี้ ตารางที่ 3.2.2 แสดงขาของเครื่องพิมพ์แบบอนุกรม

ขาที่	ลักษณะการทำงาน	คำอธิบาย
1	GND	Earth Ground
2	0	Transmit Data (TXD)
3	1	Receivce Data (RXD)
6	1	Data Set Ready(DSR)
7	GND	Signal Ground (GND)
8	1	Data Carrier Detect (DCD)
11	0	Reverse Channel
17	0	TTY Transmit Data (TTY-TXD)
20	0	Data Terminal Ready (DTR)
23	0	TTY Receivce Data Return
24	1	TTY Transmis,ot Data Return
25	1	TTY Receive Data (TTY-RXD)
4 , 5 , 9 , 10 , 12 , 13 , 14 , 15 , 16 , 18 , 19 , 21 , 22 Not Connected		

คอนเน็กเตอร์เครื่องพิมพ์แบบเซนทรอนิกส์

จะติดตั้งอยู่ที่เครื่องพิมพ์เพื่อรับข้อมูลจากคอมพิวเตอร์ ใช้คอนเน็กเตอร์ชนิดแอมเฟนิอล (amphenol) ขนาด 36 ขา การใช้งานกับคอนเน็กเตอร์แบบนี้ต้องใช้สายมัลติคอร์ขนาด 36 เส้น มีข้อดีคือ เกิดการข้ามช่องสัญญาณต่ำ และมีการรบกวนจากภายนอกน้อยที่สุดเมื่อเทียบกับแบบอื่นๆ มีรายละเอียดการจัดขาดังนี้

ตารางที่ 3.2.3 แสดงขาของเครื่องพิมพ์แบบเซนทรอนิกส์

ขาที่	ลักษณะการทำงาน	ค่าอธิบาย	ขาที่	ลักษณะการทำงาน	ค่าอธิบาย
1	Input	Strobe	19	GND	Strobe Twisted-Pair Return
2	Input	Data Bit 0	20	GND	Data Bit 0 Twisted-Pair Return
3	Input	Data Bit 1	21	GND	Data Bit 1 Twisted-Pair Return
4	Input	Data Bit 2	22	GND	Data Bit 2 Twisted-Pair Return
5	Input	Data Bit 3	23	GND	Data Bit 3 Twisted-Pair Return
6	Input	Data Bit 4	24	GND	Data Bit 4 Twisted-Pair Return
7	Input	Data Bit 5	25	GND	Data Bit 5 Twisted-Pair Return
8	Input	Data Bit 6	26	GND	Data Bit 6 Twisted-Pair Return
9	Input	Data Bit 7	27	GND	Data Bit 7 Twisted-Pair Return
10	Output	Acknowledge (ACKN)	28	GND	Acknowledge Twisted-Pair Return
11	Output	Busy	29	GND	Busy Twisted-Pair Return
12	Output	Paper End / Empty (PE)	30	GND	Paper End/Empty Twisted-Pair Return
13	Output	Logic High (5 Volts)	31	Input	Initialize (INIT)
14	Input	Auto Feed XT	32	Output	Error (EER)
15	N.A.	Not Used	33	GND	Signal Ground
16	GND	Logic Ground	34	N.A.	Not Used
17	GND	Earth Ground	35	Output	Logic High (+5 Volts)
18	N.A.	Not Used	36	Input	Select Input (SLCT IN)

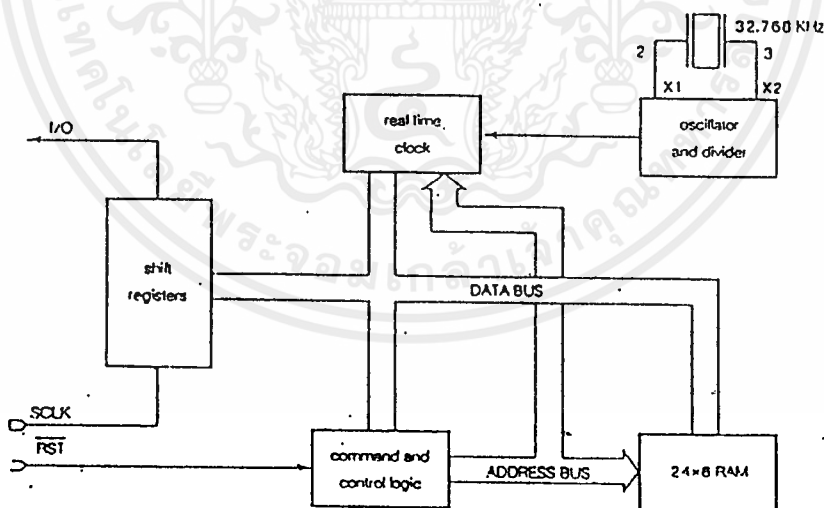
สำหรับขา 13 และ 35 สามารถใช้เป็นขาสัญญาณ สำหรับตรวจจับเครื่องพิมพ์ (printer-detect lines) หรือขาสำหรับต่อแหล่งจ่ายกระแสต่ำ (low-current power sources)

3.3 นาฬิกาบอกเวลา

ระบบควบคุมที่มีระบบเวลาเข้ามาเกี่ยวข้อง เช่น ระบบการบอกเวลาชั่วโมง นาที และ วินาที ของระบบนาฬิกา หรือบอกวัน เดือน ปี เป็นต้น การดำเนินการเหล่านี้จำเป็นต้องมีวงจรฐานเวลาที่ถูกต้องแม่นยำ เพื่อนำมาปรับปรุงค่าเหล่านี้ตลอดเวลา และหากว่าให้ไมโครคอนโทรลเลอร์ มาดูแลงานเหล่านี้ด้วยแล้ว ก็จะสิ้นเปลืองเวลาอย่างมาก ดังนั้นในทางปฏิบัติจึงนิยมนำไอซีประเภท Real time clock (RTC) มาใช้งานโดยตรง เนื่องจากมีวงจรด้านเวลาอยู่ภายใน และใช้อุปกรณ์ประกอบเพียงคริสตอลเท่านั้น สามารถดำเนินการจัดการได้อัตโนมัติ เสมือนกับเป็นระบบนาฬิกา และ ปฏิทิน(Calendar) ให้กับไมโครคอนโทรลเลอร์

การใช้งาน DS 1202 เบื้องต้น

ไอซี DS 1202 ของบริษัท Dallas Semiconductor ทำหน้าที่เป็นนาฬิกาและปฏิทิน (Real time clock/calendar) ซึ่งจะให้ข้อมูลเกี่ยวกับเวลา เช่น วินาที นาที ชั่วโมง (ทั้งรูปแบบ 24 ชั่วโมง / 12 ชั่วโมง พร้อมระบุค่า AM/PM) และวัน เดือน ปี ๖ ทั้งเดือนที่มีน้อยกว่า 31 วัน และการปรับวันในระบบอธิกสุรทิน) ไมโครคอนโทรลเลอร์ที่ต้องการนำข้อมูลเหล่านี้ไปใช้งาน ทำได้โดยวิธีการรับ/ส่งข้อมูลแบบอนุกรม (ขาสัญญาณ I/O) โดยมีสัญญาณนาฬิกาเพื่อกำหนดจังหวะ(ขาสัญญาณ SCLK) ดังแสดงในรูปที่ 3.3.1



รูปที่ 3.3.1 แผนภาพหน่วยการทำงานภายในไอซี DS 1202

3.4 จอภาพและการแสดงผล

แนวทางการใช้งาน LCD MODULE

ปัจจุบัน LCD ได้ใช้สำหรับแสดงผลในเครื่องมือเครื่องใช้ต่างๆกันอย่างกว้างขวาง เนื่องจากมันกินกระแสต่ำ และสามารถแสดงผลเป็นตัวอักษร ตัวเลข หรือกราฟฟิคก็ได้ (เฉพาะรุ่น) แต่ในด้านวงจรซึ่งมีระบบการทำงานที่ซับซ้อน และจัดหาอุปกรณ์ได้ค่อนข้างยาก จึงได้มีผู้ผลิตผลิต LCD ออกมาเป็น LCD MODULE ซึ่งเป็น MODULE ที่ประกอบไปด้วยตัว LCD และวงจรควบคุม (เรียกว่า LCM.) ทำให้ผู้ใช้สามารถเชื่อมต่อเข้ากับระบบไมโครโด้ง่ายและสะดวกในการเขียนโปรแกรม รวมทั้ง LCD MODULE ยังมีจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม จึงทำให้มีการใช้แผงแสดงด้วย LCD MODULE กันมากขึ้น

LCD MODULE มีอยู่หลายรุ่น โดยในแต่ละรุ่นจะมีคุณสมบัติแตกต่างกัน ซึ่ง LCD MODULE แบ่งออกได้เป็น 2 กลุ่มหลักคือ แบบ DOT MATRIX และแบบ GRAPHIC โดยแบบ DOT MATRIX จะแสดงผลเป็นตัวอักษรขนาด 5*8 DOT และมีจำนวนตัวอักษรกับจำนวนบรรทัดแตกต่างกันในแต่ละรุ่น ส่วนแบบ GRAPHIC จะแสดงผลเป็นแบบ BIT-MAP คือจะสร้างเป็นภาพใดๆก็ได้ตามต้องการ แนวทางในการใช้งานทั้ง 2 แบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบ DOT MATRIX มากกว่า เนื่องจากมีราคาถูก และเพียงพอต่อการใช้งานเป็นส่วนใหญ่ และในที่นี้จะกล่าวเฉพาะการใช้งานแบบ DOT MATRIX เท่านั้น คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปเป็นข้อๆได้ดังนี้

1. มิให้เลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงเป็น DOT MATRIX ขนาด 5*8 DOT
3. สามารถต่อเข้ากับระบบไมโครโด้ 2 ลักษณะ คือแบบ MEMORY MAP (20-PIN LCD BUS) และแบบผ่าน 8255 PORT (26-PIN 8255 BUS) โดยกรณี 26-PIN 8255 BUS จะใช้แผ่น PCB (DMCAD) เป็นตัว ADAPTER ทำให้เป็น 8255 BUS อีกที
4. การใช้งานง่ายและสะดวก คือระบบไมโครโด้เพียงส่งข้อมูลให้กับ LCD MODULE ข้อความก็จะปรากฏบนแผงแสดง และจะคงไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโครโด้
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR DISPLAY , HOME CURSOR , ON OFF CURSOR , BLINK CHARACTER และอื่นๆ
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสน้อย และมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 v. เท่านั้น

ขาสัญญาณของ LCD MODULE

ภาพแสดงผังเวลาของขาสัญญาณ

TIMING CHART			時序			
Item (項目)	Symbol 符號	Measuring Condition 測試條件	(標準值) Standard Value			Unit 單位
			min	typ	max	
Enable Cycle Time (允許時間周期)	TcyE	Figs.1,2	1000	—	—	nS
Enable Pulse Width, High Level (允許脈沖寬度、高電平)	PW _{EH}	Figs.1,2	450	—	—	nS
Enable Rise and Decay Time (允許上升和下降時間)	t _{Er,Ed}	Figs.1,2	—	—	25	nS
Address Setup Time,RS,R/W—E 地址建立時間)	t _{AS}	Figs.1,2	140	—	—	nS
Data Delay Time (數據延遲時間)	t _{DD}	Fig.2	—	—	320	nS
Data Setup Time (數據設置時間)	t _{DS}	Fig.1	195	—	—	nS
Data Hold Time (數據保持時間)	t _H	Fig.1	10	—	—	nS
Data Hold Time	t _{DH}	Fig.2	20	—	—	nS
Address Hold Time(地址保持時間)	t _{AH}	Figs.1,2	10	—	—	nS

* V_{CC} = 5.0V ± 5%, T_a = 25°C

FIG.1 WRITE OPERATION (寫操作)

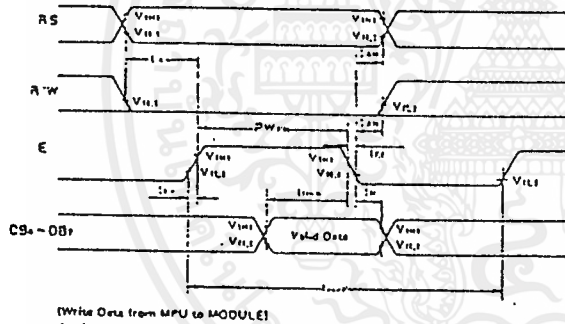
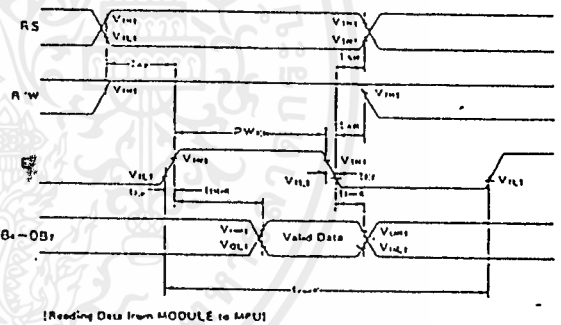


FIG.2 READ OPERATION (讀操作)



รูปที่ 3.4.1 การกระทำการเขียน

รูปที่ 3.4.2 การกระทำการอ่าน

อนุกรม DMC (DMC Series)

การเชื่อมต่อเข้ากับระบบไมโคร

LCD MODULE จะเชื่อมต่อเข้ากับระบบไมโครได้ 2 ลักษณะ คือแบบ MEMORY MAP โดยผ่าน LCD BUS ขนาด 20 PIN และแบบ I/O PORT โดยผ่าน 8255 BUS ขนาด 26 PIN ซึ่งทั้งสองแบบนี้จะมีข้อดีและข้อเสียแตกต่างกันไป โดยแต่ละแบบจะมีหลักการดังนี้

การต่อแบบ MEMORY MAP

1. สามารถต่อเข้ากับ CHIP เบอร์ต่างๆไปได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครมองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที
2. สามารถเขียนและอ่านข้อมูลจาก LCD MODULE ได้ ทำให้มองเสมือนว่าเป็น MEMORY BUFFER
3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้
4. ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น
5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการ DECODE ละเอียดพอสมควร
6. การจัดหาสัญญาณจะต้องเป็นไปตามแบบของ CHIP แต่ละเบอร์ด้วย

การต่อแบบ I/O PORT

1. สามารถต่อเข้ากับ I/O PORT ใดๆก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมาให้ตรงกับข้อกำหนดของ LCD MODULE
2. สามารถเขียนข้อมูลให้ LCD MODULE ได้อย่างเดียว ซึ่งควรจะกำหนด MEMORY ส่วนหนึ่งให้เป็นเสมือน BUFFER ให้กับ LCD MODULE อีกที
3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครเอง เพื่อรอให้ LCD MODULE กระทำกระบวนการต่างๆ
4. ใช้ได้กับบอร์ดต่างๆที่มี PORT
5. ไม่เปลืองส่วนของ MEMORY ในการใช้งาน
6. การจัดหาสัญญาณกระทำได้อย่างอิสระ

ชุดคำสั่งควบคุมและการแสดงข้อความ

การเขียนและการอ่านข้อมูลกับ LCD MODULE คือการกำหนดคุณสมบัติต่างๆในการใช้งานของ LCD ตามชุดคำสั่งควบคุม และรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางต่อไปนี้

ตารางที่ 3.4.1 แสดงคำสั่งของ LCD MODULE

INSTRUCTION	R S	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	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	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40	
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS						40		
BUSY,ADD. READ	0	1	BF	ADDRESS						0		
CGRAM,DDRAM WR	1	0	WRITE DATA						40			
CGRAM,DDRAM RD	1	1	READ DATA						40			

ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับ LCD MODULE จะแบ่งได้เป็น 2 ลักษณะคือ INSTRUCTION และ DATA โดยกำหนดด้วยขาสัญญาณ RS คือถ้า RS=0 หมายถึงส่งสัญญาณควบคุม(INSTRUCTION)

หรืออ่านค่า FLAG สภาพการทำงานของ LCD MODULE และถ้า RS=1 หมายถึงการเขียนหรืออ่าน DATA กับ LCD MODULE

2. หลักในการเขียนข้อมูลให้ LCD MODULE คือเมื่อมีการเขียนข้อมูลลงไปแล้ว ตัว LCD MODULE จะต้องใช้เวลาในการทำงานซึ่งขณะหนึ่ง (ตามค่า EXECUTE TIME ในตาราง) ซึ่งระบบไมโครสามารถตรวจสอบได้จาก BUSY FLAG (BF) และถ้าเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลอันดับต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบ I/O PORT คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครก็จะต้องใช้วิธีการหน่วงเวลาแทน

3. การเขียนข้อมูลให้กับ LCD MODULE นี้ สามารถทำได้ทั้งแบบ 8 BIT และ 4 BIT โดยกรณี 4 BIT จะใช้สายสัญญาณ DATA เพียง 4 เส้น คือ DB4-DB7 (ใช้สำหรับระบบไมโครแบบ 4 BIT หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 BIT เพียงแต่ให้เขียน 2 ครั้ง คือ DB4-DB7 ก่อน แล้วตามด้วย DB0-DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่ง FUNCTION SET ด้วย

4. DDRAM (DISPLAY DATA RAM) คือหน่วยความจำภายในตัว LCD MODULE ที่เป็น BUFFER ของข้อมูล โดยถ้าเขียนรหัส ASCII ใดๆ ลงไปในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่แสดงบนหน้าจอ

5. CGRAM (CHARACTER GENERATOR RAM) คือหน่วยความจำภายในตัว LCD MODULE สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้าง ADDRESS ได้ทั้งหมด 64 BYTE คือ 8 ตัวอักษร คูณกับ 8 ROW

รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

จะทำการเขียนตัวอักษร SPACE ลงใน DDRAM ทั้งหมด และกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะกลับไปยังตำแหน่งซ้ายบนสุดของจอภาพ

2. CURSOR AT HOME

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

สำหรับกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

3. ENTER MODE SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

I/D = 0 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ DECREMENT

I/D = 1 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ INCREMENT

S = 0 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะถูกเลื่อนไปทิศทางตามค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะอยู่กับที่ และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

4. DISPLAY ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D = 0 กำหนดให้ OFF DISPLAY

D = 1 กำหนดให้ ON DISPLAY

C = 0 กำหนดให้ OFF CURSOR

C = 1 กำหนดให้ ON CURSOR โดย CURSOR จะเป็นเส้นขีดได้ตัวอักษร

B = 0 กำหนดให้ ไม่มีการกระพริบที่ตำแหน่ง CURSOR

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่ง CURSOR (กระพริบเป็นรูปสี่เหลี่ยม)

5. DISPLAY SHIFT

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

S/C = 0 กำหนดให้เลื่อน CURSOR ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความบนแผงแสดงตามทิศทาง R/L ไป 1 COLUMN
(เลื่อนทุกบรรทัด)

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

6. FUNCTION SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 4 BIT

DL = 1 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 8 BIT

จะสังเกตเห็นว่า การกำหนดค่า DL นี้ สามารถกระทำได้ที่ DB4-DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 BIT ตั้งแต่แรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD MODULE มีการรับข้อมูลแบบ 4 BIT ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 DUTY และ 1/11 DUTY

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 DUTY

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5*7 DOTS

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5*10 DOTS (กรณีที่ LCD MODULE เป็นแบบ 5*7 อยู่แล้ว ก็จะไม่ผลอะไร)

7. SET CGRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนด ADDRESS ของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที

8. SET DDRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

สำหรับการกำหนด ADDRESS ของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที ตำแหน่งของ ADDRESS ในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนอักษรต่อบรรทัดไม่เท่ากัน ซึ่งแสดงดังภาพต่อไปนี้

9. BUSY FLAG AND ADDRESS READ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	ADDRESS						

สำหรับการอ่านค่า BF (BUSY FLAG) ซึ่งบอกถึงความพร้อมของ LCD MODULE ในการรับข้อมูล ถ้า BF = 0 หมายถึงว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ยังเป็นการอ่านค่า ADDRESS ของ CGRAM หรือ DDRAM ด้วย

การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

สำหรับการเขียนข้อมูลลงในหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มขึ้นหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้มีการกำหนด ADDRESS ที่ใด

2. READ DATA FROM DDRAM OR CGRAM

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มขึ้นหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้มีการกำหนด ADDRESS ที่ใด

แนวทางการเขียนโปรแกรมควบคุม

1. เมื่อจ่ายไฟเลี้ยงให้กับ LCD MODULE ครั้งแรก ภายในจะมีการ RESET ระบบโดยอัตโนมัติ ซึ่งจะใช้เวลา 10 มิลลิวินาที หลังจากทีระดับแรงไฟขึ้นถึง 4.5 โวลต์ แล้ว ทั้งนี้ระบบ RESET ดังกล่าวจะกระทำสิ่งต่างๆต่อไปนี้

- ทำการ CLEAR จอภาพทั้งหมด (CLEAR DISPLAY)
- กำหนดคุณสมบัติด้วยคำสั่ง FUNCTION SET คือ DL = 1 (ติดต่อกับระบบไมโครในแบบ 8 BIT) , N = 0 (แสดงข้อมูล 1 บรรทัด) , F = 0 (กำหนดตัวอักษรแบบ 5*7 DOTS)
- กำหนดคุณสมบัติด้วยคำสั่ง DISPLAY ON/OFF คือ D = 0 (ไม่แสดงข้อมูล) , C = 0 (CURSOR OFF) , B = 0 (BLINK OFF)
- กำหนดคุณสมบัติด้วยคำสั่ง ENTRY MODE SET คือ I/D = 1 (INCREMENT) , S = 0 (NO SHIFT)

การใช้งาน LCD MODULE ต้องรอให้ขบวนการ RESET ภายในทำงานเรียบร้อยแล้ว ซึ่งจะตรวจสอบได้ด้วย BF (BUSY FLAG) หรืออาจจะใช้การหน่วงเวลาก็ได้

2. การใช้งาน LCD MODULE จะเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่างๆรวมทั้งการอ่านหรือเขียนข้อมูล จะถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่ โดยปกติโปรแกรมจะ

ต้องกำหนดคุณสมบัติต่างๆที่ต้องการไว้ที่ส่วนต้นและจากนั้นก็จะเป็นการอ่านและเขียนข้อมูลลงใน DDRAM ซึ่งก็คือข้อความที่จะให้แสดงนั่นเอง

การโหลดอักษรที่ออกแบบเอง

LCD MODULE สามารถโหลดตัวอักษรที่ออกแบบเองได้ โดยกระทำได้เพียง 8 ตัวอักษร ซึ่งมีหลักการดังนี้ ให้ทำการโหลดรูปแบบที่กำหนดไว้แล้ว ลงในหน่วยความจำ CGRAM ADDRESS ที่อ้างถึง CGRAM จะใช้เพียง 6 BIT โดยอ้างได้ทั้งหมด 64 BYTE (คือ 8 ตัวอักษร คูณกับ 8 ROW) และการเรียกใช้ตัวอักษรที่ออกแบบเองนี้ จะใช้รหัส 00-07 (หรือ 08-0F ก็ได้) ขึ้นตอนในการโหลดตัวอักษรเป็นดังนี้

1. กำหนด ADDRESS ของ CGRAM ด้วยคำสั่ง SET CGRAM ทั้งนี้ถ้าโหลดอักษรตัวแรกก็จะใช้ ADDRESS ที่ตำแหน่ง 0
2. เขียนข้อมูลที่เป็นแบบของตัวอักษรด้วยคำสั่ง WRITE DATA โดยใช้ 8 BYTE ต่อหนึ่งตัวอักษร คือ เรียง ROW จากแถวบนลงล่าง และใช้ BIT ที่ 0-4 เท่านั้น การโหลดนี้จะทำได้ต่อเนื่องไปเรื่อยๆ โดยไม่ต้องกำหนด ADDRESS ใหม่
3. ตัวอักษรที่โหลดไปแล้ว สามารถเรียกใช้ได้ตลอดไปในขณะที่ยังมีไฟเลี้ยงอยู่ ซึ่งการโหลดนี้ให้กระทำที่ส่วนต้นของโปรแกรมได้เลย รูปแบบของตัวอักษรมีตัวอย่างดังนี้

	ADDRESS	DATA
	00	04
	01	0E
	02	1F
	03	1F
	04	0E
	05	04
	06	04
	07	1F

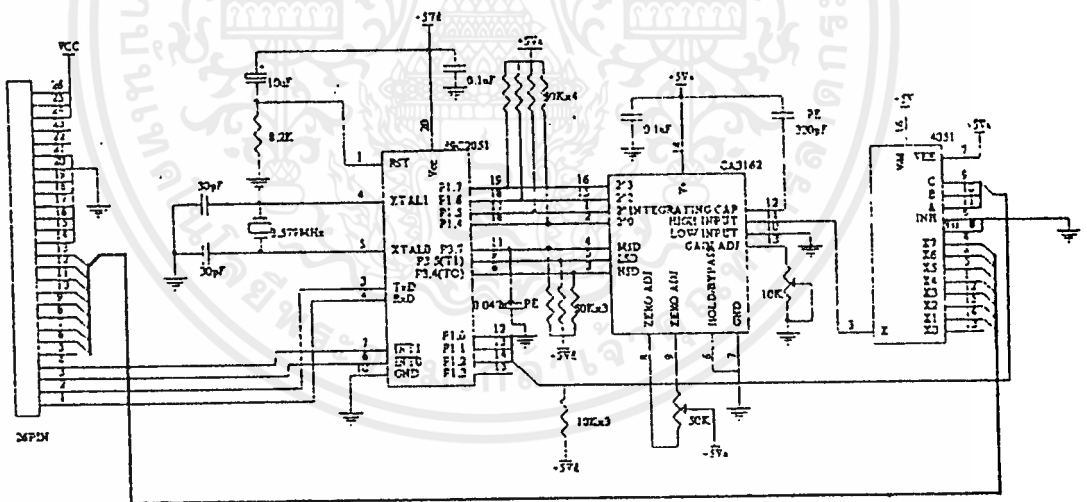
รูปที่ 3.4.3 แสดงรูปแบบของตัวอักษร

บทที่ 4

การออกแบบและการทำงานของวงจร

4.1 การทำงานของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล

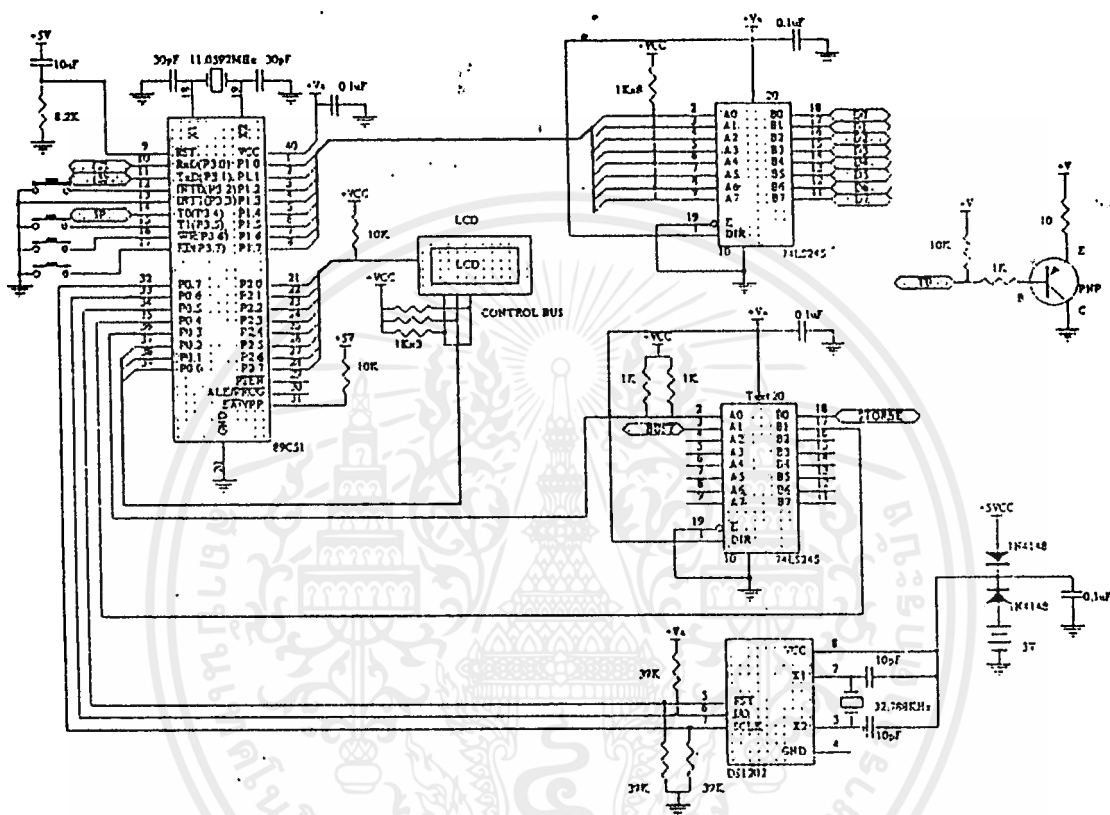
จากรูปที่ 4.1 เมื่อป้อนไฟเลี้ยงให้กับวงจร แล้วไอซีมัลติเพลกเซอร์ 4051 จะรับค่าข้อมูลทางไฟฟ้าเป็นโวลต์ตรงที่แปลงมาจากค่าปริมาณทางฟิสิกส์โดยตัวเซนเซอร์(ในโครงการนี้ใช้ไอซี LM35 ซึ่งทำหน้าที่แปลงค่าข้อมูลทางอุณหภูมิไปเป็นค่าข้อมูลทางไฟฟ้า) เข้ามาในแต่ละช่องของช่องรับข้อมูล(X0 ถึง X7) จากนั้นไมโครคอนโทรลเลอร์ 89C2051 จะส่งค่าแอดเดรสจากพอร์ท P1.0 ถึง P1.2 ออกไปยังไอซีมัลติเพรกเซอร์ 4051 เข้าไปทางขา A B และ C เพื่อเลือกช่องข้อมูลเรียงตามลำดับตั้งแต่ช่องข้อมูลที่ 0(X0) ถึง 7(X7) ซึ่งจะนำค่าข้อมูลที่เป็นแบบอนาลอกเหล่านี้ไปแปลงเป็นค่าข้อมูลทางดิจิตอลโดยไอซี A/D CA3162 และข้อมูลดิจิตอลที่ได้เหล่านี้จะถูกส่งแบบขนานไปยังไมโครคอนโทรลเลอร์ 89C2051 เพื่อเปลี่ยนการสื่อสารข้อมูลดิจิตอลแบบขนานไปเป็นแบบอนุกรม แล้วจึงส่งข้อมูลดิจิตอลเหล่านี้ออกไปยังส่วนของวงจรรับข้อมูลส่งผ่านเครื่องพิมพ์ทางขา TxD ของไมโครคอนโทรลเลอร์ 89C2051 แบบอนุกรมต่อไป



รูปที่ 4.1 วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล

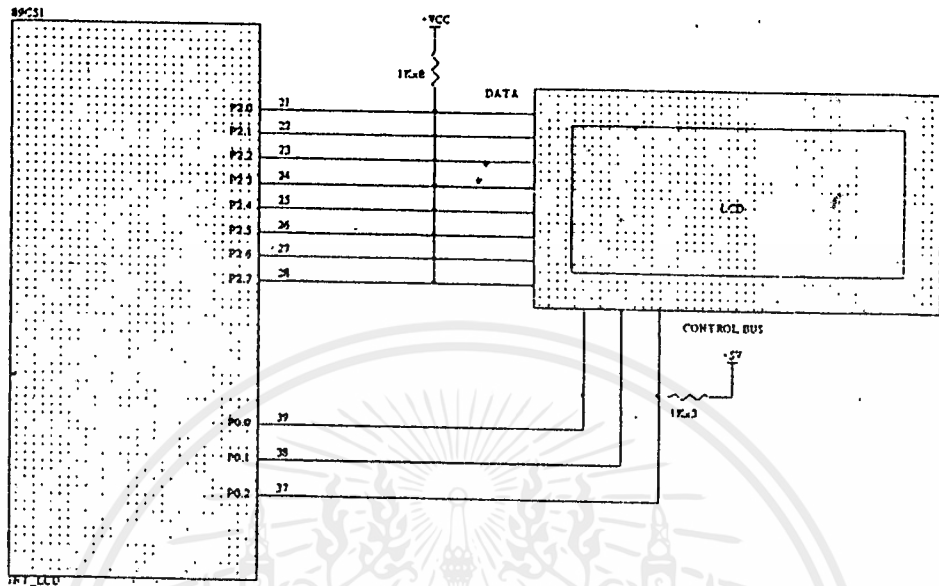
4.2 การทำงานของวงจรรับข้อมูลส่งผ่านเครื่องพิมพ์

จากรูปที่ 4.2.1 ในขณะที่มีการป้อนไฟเลี้ยงให้กับวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล วงจรรับข้อมูลส่งผ่านเครื่องพิมพ์ก็จะได้รับไฟเลี้ยงจากที่เดียวกันด้วย เมื่อวงจรส่วนนี้ได้รับไฟเลี้ยงแล้ว ไมโครคอนโทรลเลอร์ 89C51 จะส่งสัญญาณควบคุมออกไปทางพอร์ท P0.0 ถึง P0.2 เพื่ออนุญาตให้ LCD ทำงานและทุกครั้งที่ต้องการให้แสดงตัวสตริงบนจอ LCD จะต้องส่งข้อมูลที่เป็นค่าสตริงออกไปทางพอร์ท P2.0 ถึง P2.7 เสมอ(รูปที่ 4.2.2) ทุกครั้งที่ต้องการอ่านหรือให้แสดง วัน เดือน ปี และ/หรือเวลา ไมโครคอนโทรลเลอร์ 89C51 จะต้องส่งสัญญาณออกไปบอกตัวไอซีเรียลไทม์ DS1202 จากพอร์ท P0.6 ของ 89C51 ไปยังขา I/O ของ DS1202 จากนั้น DS1202 จึงจะส่งข้อมูล วัน เดือน ปี และ/หรือเวลาขณะนั้นไปให้ไมโครคอนโทรลเลอร์ 89C51 จากขา I/O ของ DS1202 ไปยังพอร์ท P0.6 ของ 89C51 ตามสัญญาณเคล็อกจากขา SCLK ของ DS1202 ในส่วนนี้ได้มีการใส่ถ่านแบกอัพไว้เพื่อป้องกันการสูญหายของข้อมูลขณะที่ไม่มีไฟเลี้ยง(รูปที่ 4.2.3) จากตอนที่ 4.1 ข้อมูลดิจิทัลที่ส่งออกมาแบบอนุกรมจากไมโครคอนโทรลเลอร์ 89C2051 เป็นข้อมูลดิจิทัลที่ถูกรับโดยไมโครคอนโทรลเลอร์ 89C51 ซึ่ง 89C51 จะเปลี่ยนข้อมูลดิจิทัลที่สื่อสารแบบอนุกรมไปเป็นแบบขนานออกไปทางพอร์ท P1 ผ่านไอซีบัฟเฟอร์ 74LS245 และส่งต่อไปยังเครื่องพิมพ์เพื่อพิมพ์เป็นข้อมูลออกมา(รูปที่ 4.2.4) และทุกครั้งที่จะส่งข้อมูลออกไปยังเครื่องพิมพ์นี้จะต้องตรวจสอบว่าเครื่องพิมพ์พร้อมที่จะทำงานหรือไม่โดยดูจากขาสัญญาณ busy ของเครื่องพิมพ์ ถ้าสัญญาณ busy เป็น low แสดงว่าเครื่องพิมพ์พร้อมที่จะรับข้อมูล และ 89C2051 ก็จะส่งข้อมูลออกไปรอที่สายบัสข้อมูล จากนั้นจึงส่งสัญญาณสตrobeออกไปให้เครื่องพิมพ์ แล้วเครื่องพิมพ์ก็จะรับข้อมูลจากสายบัสข้อมูลเข้าไปพิมพ์เป็นข้อมูลออกมา(รูปที่ 4.2.5) ในส่วนของสวิทซ์ได้มีการตรวจสอบการกดสวิทซ์อยู่ตลอดเวลาว่ามีการกดหรือไม่ ซึ่งหน้าที่ของสวิทซ์ได้แตกต่างกันไปแล้วแต่กรณีสามารถศึกษาได้จากหัวข้อผลการทดลองในบทที่ 5 (รูปที่ 4.2.6) และในการเริ่มป้อนไฟเลี้ยง หรือมีการกดสวิทซ์จะมีเสียงบี๊ฟเกิดขึ้นที่ลำโพงเล็กๆที่ต่อออกไปจากพอร์ท P3.4 ของ 89C51(รูปที่ 4.2.7)

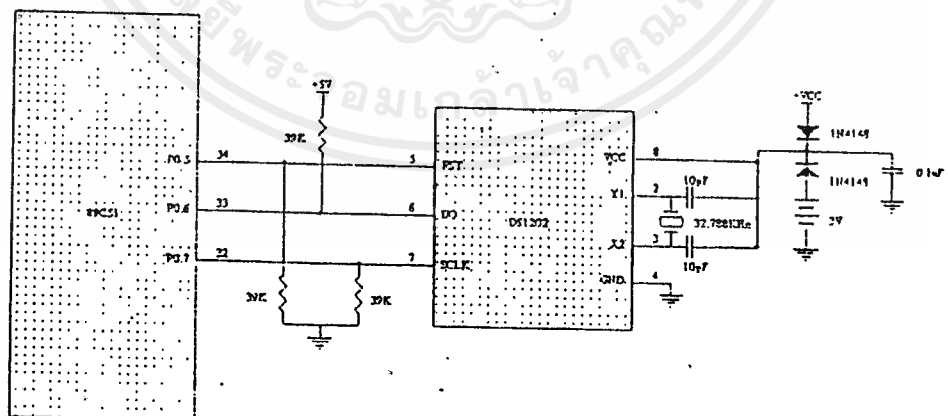


รูปที่ 4.2.1 วงจรรับข้อมูลส่งผ่านเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

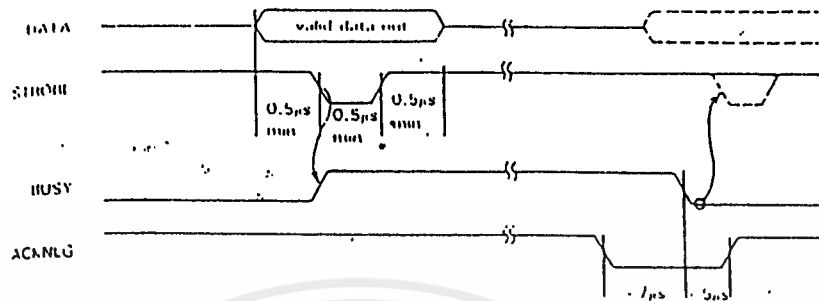


รูปที่ 4.2.2 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับจอแสดงผล LCD

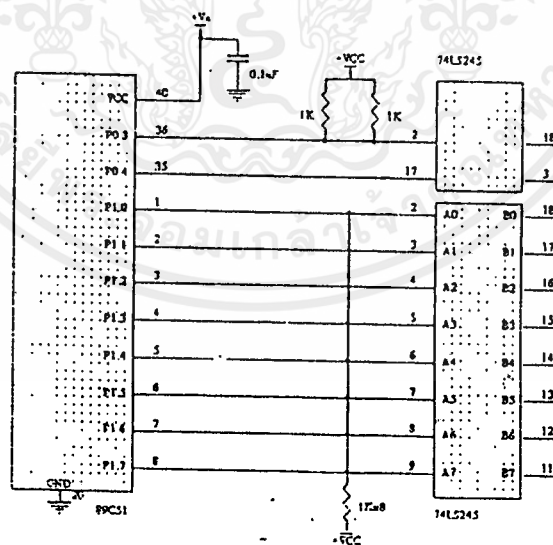


รูปที่ 4.2.3 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ real time clock DS1202

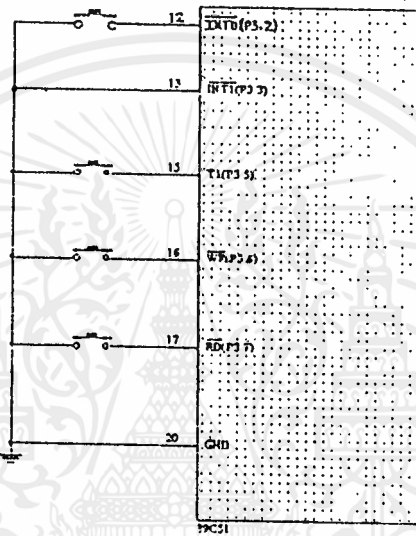
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



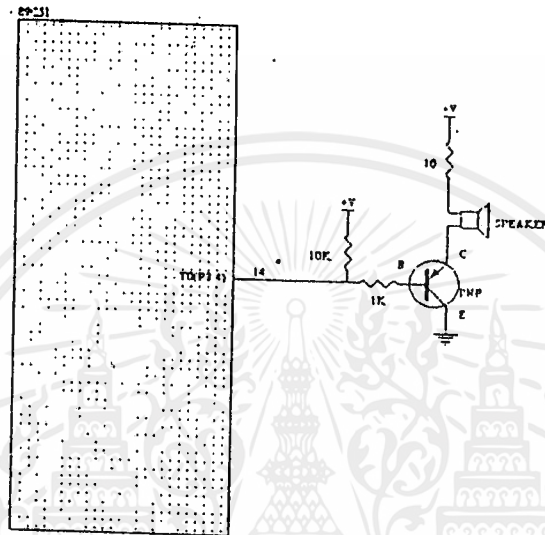
รูปที่ 4.2.4 สัญญาณการสื่อสารระหว่างไมโครคอนโทรลเลอร์กับเครื่องพิมพ์



รูปที่ 4.2.5 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับเครื่องพิมพ์



รูปที่ 4.2.6 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับสวิตช์



รูปที่ 4.2.7 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับลำโพง

บทที่ 5

การทดลองและผลการทดลอง

การทดลอง

เมื่อป้อนไฟเลี้ยงให้กับระบบแล้ว LED ทั้ง 2 ตัว ของระบบจะสว่างขึ้น และมีเสียงบี๊พจาก ลำโพงขนาดเล็กดังขึ้นกับข้อความปรากฏบนหน้าจอ LCD ว่า “prn a/d conv.” ซึ่งเป็นชื่อย่อของ โครงการนี้คือ PRINTING A/D CONVERTER ต่อไปบนหน้าจอ LCD จะแสดงข้อความ “OFF GRAPH” ซึ่งแสดงว่าในตอนเริ่มแรกระบบได้จัดรูปแบบที่จะพิมพ์ออกทางเครื่องพิมพ์ เป็นแบบที่ไม่ใช่กราฟ หรือเป็นเหมือนตารางข้อมูล ถ้าจะให้แสดงเป็นกราฟต้อง กดปุ่ม P3.6 จอ LCD จะแสดง “ON GRAPH ” และส่งเสียงบี๊พ แต่ถ้าต้องการให้แสดงเป็นตารางข้อมูลก็ต้องกดปุ่ม P3.5 จอ LCD จะแสดง “ OFF GRAPH” และส่งเสียงบี๊พ เมื่อเลือกรูปแบบที่ต้องการแสดงแล้วก็ต้องกดปุ่ม P3.7 ซึ่งขณะนี้ทำหน้าที่เป็นปุ่ม ENTER /RUN ต่อไประบบจะให้เลือกช่วงเวลาที่จะให้แสดง ข้อมูลว่าต้องการให้แสดงข้อมูลทุกๆ กี่นาทีกี่ โดยปุ่ม P3.6 คือทุกๆ 1 นาที P3.5คือทุกๆ 10 นาที และ P3.2 คือทุกๆ 30 นาที ซึ่งในตอนเริ่มแรกระบบจะตั้งไว้ที่ทุกๆ 1 นาที เมื่อเลือกช่วงเวลาแล้ว กดปุ่ม P3.7 หรือปุ่ม ENTER/RUN ระบบก็จะเริ่มพิมพ์ข้อมูลปัจจุบันออกทางเครื่องพิมพ์

ในการแสดงข้อมูลออกทางเครื่องพิมพ์ ทั้งที่เป็นแบบกราฟและแบบตาราง มีข้อมูลที่คล้ายกันคือ ด้านบนสุดจะเป็นชื่อโครงการกับชื่อผู้ทำโครงการ ต่อจากนั้นจะเป็น วันเดือนปี (DD/MM/YY) ชั่วโมง,นาทีกี่,วินาที (HH/MM/SS) และข้อมูล (DATA) โดยที่ CH0 ถึง CH7 คือ CHANNEL ที่ 0 ถึง 7 ค่าข้อมูลที่จัดนี้คือค่าอุณหภูมิ ซึ่งได้แสดงออกมาเป็นค่าโวลต์เตจ มีหน่วยเป็นมิลลิโวลต์ และมีค่าตั้งแต่ 000 ถึง 999 มิลลิโวลต์ ถ้าค่าข้อมูลเกิน 999 มิลลิโวลต์ จะถูกแสดงเป็น EEE หรือแสดงว่าเกิดโอเวอร์โฟลว์ขึ้น การอ่านค่าโวลต์เตจเป็นค่าอุณหภูมิอ่านได้ดังตัวอย่างคือ 296 มิลลิโวลต์ อ่านค่าได้เป็น 29.6 c และ 639 มิลลิโวลต์ อ่านค่าได้เป็น 63.9 c เป็นต้นในกรณีที่ให้แสดงเป็นกราฟจะมีข้อมูลกราฟเพิ่มขึ้นมาและใช้ channel เพียง 2 channel เท่านั้น แกนนอนของกราฟเป็นค่าโวลต์เตจ หรืออุณหภูมิ ส่วนแกนตั้งเป็นแกนเวลา และเครื่องหมายมาร์กแสดงค่าข้อมูลได้ใช้ค่าตัวเลขของแต่ละ channel เช่น channel 0 ใช้เลข 0 เป็นเครื่องหมายมาร์ก แสดงค่าข้อมูลเป็นต้น การเพิ่มขึ้นของค่าเวลาได้เพิ่มขึ้น ตามช่วงเวลาที่เราได้เลือกไว้ เช่น ทุกๆ 1 นาที เวลาจะเพิ่มขึ้นดังนี้

ขณะที่ระบบทำงานอยู่เราสามารถเลือกช่วงเวลาใหม่ โดยการกดปุ่ม P3.6 ทุกๆ 1 นาที P3.5 คือ ทุกๆ 10 นาที หรือ P3.2 คือทุกๆ 30 นาที นอกจากนี้แล้วยังสามารถเลือกโหมด ON GRAPH เมื่อกดปุ่ม P3.7 ระบบก็จะเปลี่ยนไปเป็นระบบ OFF GRAPH หรือ ถ้าปัจจุบันอยู่ที่ OFF GRAPH เมื่อกดปุ่ม P3.7 ระบบก็จะเปลี่ยนไปเป็นโหมด ON GRAPH เป็นต้นในการกดปุ่มใดปุ่มหนึ่ง หรือปุ่มรีเซ็ต ระบบจะเริ่มพิมพ์ ข้อมูลใหม่ตั้งแต่ ชื่อโครงการ จนถึงค่าของข้อมูลไปเรื่อยๆ ระบบจะหยุดทำงาน เมื่อเอาไฟเลี้ยงออกจากระบบเท่านั้น ซึ่งในการวัดอุณหภูมินี้ ได้ใช้เซนเซอร์วัดอุณหภูมิ LM35 ในการเปลี่ยนค่าอุณหภูมิไปเป็น พลังงานทางไฟฟ้า หรือโวลต์เตจ

ผลการทดลอง

ได้มีการใช้โครงการเครื่องแปลงสัญญาณอนาลอกเป็นดิจิตอลส่งผ่านเครื่องพิมพ์นี้ทำการทดลองวัดค่าอุณหภูมิที่ชั้น 3 ตึกวิทยาศาสตร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดยใช้ LM35 เป็นตัวเซนเซอร์วัดอุณหภูมิ 2 ตัว เพื่อเป็นตัวเปรียบเทียบกัน ช่วงเวลาที่ให้แสดงข้อมูลคือทุกๆ 30 นาที ซึ่งได้ผลการทดลองดังนี้

PRINTING A/D CONVERTER

PROGRAMMED BY

MR. ANANTACHAI NAIJIT

AND

MR. SANCHAYOT SAKULJITTACHAROEN

DD/MM/YY : 30/03/98

[illegible]

00:47:01	CH0	CH1	-	0
	275	276		
01:17:01	CH0	CH1	-	0
	275	275		
01:47:01	CH0	CH1	-	0
	276	276		
02:17:01	CH0	CH1	-	0
	274	275		
02:47:01	CH0	CH1	-	0
	273	274		
03:17:01	CH0	CH1	-	0
	273	273		
03:47:01	CH0	CH1	-	0
	272	272		
04:17:01	CH0	CH1	-	0
	270	271		
04:47:01	CH0	CH1	-	0
	270	271		
05:17:01	CH0	CH1	-	0
	268	268		
05:47:01	CH0	CH1	-	0
	260	261		
06:17:01	CH0	CH1	-	0
	263	263		
06:47:01	CH0	CH1	-	0
	265	266		
07:17:01	CH0	CH1	-	0
	269	271		
07:47:01	CH0	CH1	-	0
	273	274		
08:17:01	CH0	CH1	-	0
	283	284		
08:47:01	CH0	CH1	-	0
	304	304		
09:17:01	CH0	CH1	-	0
	310	311		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

09:47:01	CH0	CH1	-	๑
	305	306		
10:17:01	CH0	CH1	-	๑
	312	316		
10:47:01	CH0	CH1	-	๑
	313	316		
11:17:01	CH0	CH1	-	๑
	326	329		
11:47:01	CH0	CH1	-	๑
	317	319		
12:17:01	CH0	CH1	-	๑
	330	333		
12:47:01	CH0	CH1	-	๑
	335	338		
13:17:01	CH0	CH1	-	๑
	331	332		
13:47:01	CH0	CH1	-	๑
	333	335		
14:17:01	CH0	CH1	-	๑
	335	336		
14:47:01	CH0	CH1	-	๑
	320	321		
15:17:01	CH0	CH1	-	๑
	314	315		
15:47:01	CH0	CH1	-	๐1
	318	320		
16:17:01	CH0	CH1	-	๑
	323	325		
16:47:01	CH0	CH1	-	๑
	310	311		
17:17:01	CH0	CH1	-	๑
	297	299		
17:47:01	CH0	CH1	-	๑
	298	300		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการทดลอง

6.1 สรุป

โครงการ PRINTING A/D CONVERTER นี้สามารถนำไปใช้วัดปริมาณทางฟิสิกส์ต่างๆ ได้ เช่น อุณหภูมิ ความดัน และกระแสไฟฟ้า เป็นต้น เพียงแต่สามารถหาตัวเซนเซอร์ที่เหมาะสมมาใช้งานเท่านั้น ก็สามารถวัดค่าทางฟิสิกส์ต่างๆ ได้ ระบบนี้มี channel จัดข้อมูลถึง 8 channel โดยแต่ละ channel อาจนำไปวัดปริมาณทางฟิสิกส์ที่เหมือนกัน หรือต่างกันก็ได้ตามความต้องการของผู้ใช้ ซึ่งไม่จำเป็นต้องใช้ครบทั้ง 8 channel ก็ได้ และระบบที่มีขนาดเล็กนี้ทำให้สามารถพกพาเคลื่อนย้ายได้อย่างสะดวก ในการที่จะนำไปวัดปริมาณทางฟิสิกส์ในสถานที่ต่างๆ ในการวัดปริมาณทางฟิสิกส์ต่างๆ เหล่านี้ เราสามารถเก็บข้อมูลเหล่านี้เป็นฐานข้อมูล เพื่อใช้ในการอ้างอิงถึงในครั้งต่อไปได้ เพราะในการเก็บข้อมูลแต่ละครั้ง จะมีการแสดงวัน เวลา ขณะที่ทำการวัดด้วย ในขณะเดียวกันก็จะมีโหมดให้พรอดเป็นกราฟออกมา เพื่อแสดงแนวโน้มการเปลี่ยนแปลงของข้อมูลที่วัดได้ ซึ่งง่ายต่อการสังเกต การเปลี่ยนแปลงของข้อมูลมากกว่าแบบตารางข้อมูล แต่กราฟที่พรอดได้นี้ยังไม่ละเอียดมากนัก เนื่องจากแกนโวลต์เดจจะเพิ่มขึ้นทีละ 20 มิลลิโวลต์ เพราะเนื้อที่ของเครื่องพิมพ์มีขนาดเล็กเกินไป ถ้าต้องการแก้ไข ความละเอียดนี้ก็ต้องใช้เครื่องพิมพ์ที่มีขนาดใหญ่ และแก้ไขโปรแกรมที่ควบคุมระบบด้วย

6.2 ข้อเสนอแนะ

จากการทดลองที่ได้ให้ผลเป็นที่น่าพอใจ ทั้งนี้เนื่องจากการออกแบบวงจรที่มีเสถียรภาพ การเลือกอุปกรณ์ที่มีคุณภาพดี รวมทั้งเมื่อนำไปวัดปริมาณต่างๆ แล้วปรากฏว่ามีความแม่นยำสูง

ภาคผนวก ก.

โปรแกรมควบคุมสัญญาณอนาลอกเป็นดิจิทัล

; Solar Logger V1.0
;
; 14 March 1996
;
; 8 Channels Miniature Analog Sender
;
; 89C2051 + CA3162E ADC + 4051 MUX
;
; 2KB code memory

0000 CPU "8051.TBL"
0000 HOF "INT8"
;
;
;MCS-51 INTERNAL REGISTERS
;
00F0 = B: EQU 0F0H ;B REGISTER
00E0 = ACC: EQU 0E0H ;ACCUMULATOR
00D0 = PSW: EQU 0D0H ;PROGRAM STATUS WORD
00B8 = IPC: EQU 0B8H ;INTERRUPT PRIORITY
00B0 = P3: EQU 0B0H ;PORT 3
00A8 = IEC: EQU 0A8H ;INTERRUPT ENABLE
00A0 = P2: EQU 0A0H ;PORT 2
0099 = SBUF: EQU 99H ;SEND BUFFER
0098 = SCON: EQU 98H ;SERIAL CONTROL
0090 = P1: EQU 90H ;PORT 1
008D = TH1: EQU 8DH ;TIMER 1 HIGH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

008C = TH0: EQU 8CH ;TIMER 0 HIGH
008B = TL1: EQU 8BH ;TIMER 1 LOW
008A = TL0: EQU 8AH ;TIMER 0 LOW
0089 = TMOD: EQU 89H ;TIMER MODE
0088 = TCON: EQU 88H ;TIMER CONTROL
0087 = PCON: EQU 87H ;POWER CONTROL REGISTER
0083 = DPH: EQU 83H ;DATA POINTER HIGH
0082 = DPL: EQU 82H ;DATA POINTER LOW
0081 = SP: EQU 81H ;STACK POINTER
0080 = P0: EQU 80H ;PORT 0

```

``` ; ;MCS-51 INTERNAL BIT ADDRESSES ; ```

```

00D7 = CY: EQU 0D7H ;CARRY FLAG
00D6 = AC: EQU 0D6H ;AUXILIARY-CARRY FLAG
00D5 = F0: EQU 0D5H ;USER FLAG 0
00D4 = RS1: EQU 0D4H ;REGISTER SELECT MSB
00D3 = RS0: EQU 0D3H ;REGISTER SELECT LSB
00D2 = OV: EQU 0D2H ;OVERFLOW FLAG
00D0 = P: EQU 0D0H ;PARITY FLAG
00BC = PS: EQU 0BCH ;PRIORITY SERIAL PORT
00BB = PT1: EQU 0BBH ;PRIORITY TIMER 1
00BA = PX1: EQU 0BAH ;PRIORITY EXTERNAL 1
00B9 = PT0: EQU 0B9H ;PRIORITY TIMER 0
00B8 = PX0: EQU 0B8H ;PRIORITY EXTERNAL 0
00AF = EA: EQU 0AFH ;ENABLE ALL INTERRUPT
00AC = ES: EQU 0ACH ;ENABLE SERIAL INTERRUPT
00AB = ET1: EQU 0ABH ;ENABLE TIMER 1 INTERRUPT
00AA = EX1: EQU 0AAH ;ENABLE EXTERNAL 1 INTERR
00A9 = ET0: EQU 0A9H ;ENABLE TIMER 0 INTERRUPT
00A8 = EX0: EQU 0A8H ;ENABLE EXTERNAL 0 INTERR
009F = SM0: EQU 09FH ;SERIAL MODE 0

```

009E =	SM1:	EQU	09EH	;SERIAL MODE 1
009D =	SM2:	EQU	09DH	;SERIAL MODE 2
009C =	REN:	EQU	09CH	;SERIAL RECEPTION ENABLE
009B =	TB8:	EQU	09BH	;TRANSMITT BIT 8
009A =	RB8:	EQU	09AH	;RECEIVE BIT 8
0099 =	TI:	EQU	099H	;TRANSMIT INTERRUPT FLAG
0098 =	RI:	EQU	098H	;RECEIVE INTERRUPT FLAG
008F =	TF1:	EQU	08FH	;TIMER 1 OVERFLOW FLAG
008E =	TR1:	EQU	08EH	;TIMER 1 RUN CONTROL BIT
008D =	TF0:	EQU	08DH	;TIMER 0 OVERFLOW FLAG
008C =	TR0:	EQU	08CH	;TIMER 0 RUN CONTROL BIT
008B =	IE1:	EQU	08BH	;EXT INTERR. 1 EDGE FLAG
008A =	IT1:	EQU	08AH	;EXT INTERR. 1 TYPE FLAG
0089 =	IE0:	EQU	089H	;EXT INTERR. 0 EDGE FLAG
0088 =	IT0:	EQU	088H	;EXT INTERR. 0 TYPE FLAG

;

PORT 1 BIT ADDRESS

0090 =	P1.0:	EQU	090H	; START COMMAND
0091 =	P1.1:	EQU	091H	; select A-B only or A-B-C-D
0092 =	P1.2:	EQU	092H	;
0093 =	P1.3:	EQU	093H	;
0094 =	P1.4:	EQU	094H	; output soleniod #4 option
0095 =	P1.5:	EQU	095H	; output solenoid #3 option
0096 =	P1.6:	EQU	096H	; output solenoid #2 ---> A-C
0097 =	P1.7:	EQU	097H	; output solenoid #1 ---> A-B
00A0 =	p2.0:	equ	0a0h	; time base indicator
00A1 =	p2.1:	equ	0a1h	; clock out 50.000 Hz for calibration !
00A2 =	p2.2:	equ	0a2h	; tone output
00B0 =	p3.0:	equ	0b0h	
00B1 =	p3.1:	equ	0b1h	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

00B2 =    p3.2:    equ    0b2h
00B3 =    p3.3:    equ    0b3h
00B4 =    p3.4:    equ    0b4h
00B5 =    p3.5:    equ    0b5h
00B6 =    p3.6:    equ    0b6h
00B7 =    p3.7:    equ    0b7h

```

```

0048 =    SAVE_L:  EQU    048H    ; SAVE POINTER USE IN INTERRUPT
SERVICE

```

```

0049 =    SAVE_H:  EQU    049H    ; ROUTINE

```

```

0038 =    STATE:   EQU    038H    ; STATE BYTE 1 = TIME

```

```

2 = PROGRAM

```

```

; RAM BIT ADDRESS

```

```

0018 =    serial :    equ    18h    ; on/off send data

```

```

0019 =    sim:      equ    19h    ; simulate mode (speed-up test)

```

```

001A =    blink:    equ    1Ah

```

```

0021 =    output:   equ    21h    ; 16 bit output control 21h 22h

```

```

000B =    KEY:      EQU    0BH

```

```

0008 =    LED:      EQU    08H    ; INTERRUPT BLINK

```

```

000A =    XOFF_FLAG: EQU    0AH

```

```

000D =    ZEROING:  EQU    0DH

```

```

000E =    VALVE:    EQU    0EH

```

```

000F =    CYCLE_COUNT: EQU    0FH

```

```

0010 =    check_sum: equ    10h

```

```

;current time storage

```

0023 =	CHANNEL_MAX:	EQU	23H
0024 =	bcc:	equ	24h
0025 =	sec100:	equ	25h
0026 =	sec:	equ	26h
0027 =	min:	equ	27h
0028 =	hour:	equ	28h
0029 =	day:	equ	29h
002A =	MONTH:	EQU	2AH
002B =	YEAR:	EQU	2BH
002E =	warm_code:	equ	2Eh
002D =	light:	equ	2dh
002C =	econo:	equ	2ch
002F =	tl0adj:	equ	2fh
0033 =	pgm_buffer:	equ	33h
0030 =	msd:	equ	30h
0031 =	lsd:	equ	31h
0032 =	nsd:	equ	32h
0033 =	digit:	equ	33h
0034 =	kwhr:	equ	34h ; 000000 KWHR 34h, 35h, 36h
0037 =	channel:	equ	37h
0038 =	data_buffer:	equ	38h ;two bytes for each channel i.e., require 8x2=16

byte

; @38, @39 channel 0

; @40, @41 channel 1

; @42, @43 channel 2

; @44, @45 channel 3

; @46, @47 channel 4

; @48, @49 channel 5

; @50, @51 channel 6

; @52, @53 channel 7

004A = RECEIVE_BUFFER: EQU 4AH

1000 = TOP_FIFO: EQU 1000H ; TOP OF FIFO BUFFER

000D = CR: EQU 0DH

000A = LF: EQU 0AH

0010 = EOS: EQU 10H

0007 = BELL: EQU 07H

0000 ORG 000H

0000 02052C LJMP MAIN_TIMER

0003 org 003h

0003 0203DD ljmp service_int0

000B ORG 00BH

000B 02022F LJMP SERVICE_T0 ; TIMER COUNTER 0

INTERRUPT

0013 org 013h

0013 020415 ljmp service_int1

0100 ORG 0100H

;INIT_RS232C INITIALIZED 89C2051 SERIAL PORT

; BAUD RATE: 9600

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; DATA LENGTH: 8 BIT
;
; STOP BIT: 1 BIT
;
; PARITY: NO
;
; X-TAL: 3.579545 MHz

0100 758921 INIT: MOV TMOD,#00100001B
0103 759852 MOV SCON,#01010010B
0106 758780 MOV PCON,#10000000B ; smod = 1
0109 758DFE MOV TH1,#0FEH ;TIMER1 LOAD VALUE
010C D28E SETB TR1 ;START TIMER1
010E 22 RET

; SEND SEND A TO SERIAL PORT
;
; CHECK TRANSMITTER BUFFER BEFORE SEND
;
;
; ENTRY: A
;
; EXIT: NO
;

010F 3099FD SEND: JNB TI,SEND
0112 C299 CLR TI
0114 F599 MOV SBUF,A ;OK BUFFER EMPTY

SEND OUT...

0116 301004 jnb check_sum,no_check_sum
0119 2524 add a,bcc
011B F524 mov bcc,a

011D no_check_sum:

011D 22 RET

;RECIVES RECIVE BYTE FROM SERIAL PORT WITH TIME-OUT

FUNCTION

; ENTRY: NO

; EXIT: A = ASCII NUMBER OR STRING COMMAND

; A = FF TIME-OUT

```
011E 7F05  RECIVES:  MOV    R7,#05H
0120 7E00  AGAIN:      MOV    R6,#00H      ; TIME-OUT DELAY
0122 209807 GET_RI:    JB      RI,READY
0125 DEFB                      DJNZ   R6,GET_RI
0127 DFF7                      DJNZ   R7,AGAIN
0129 74FF                      MOV    A,#0FFH      ; TIME-OUT NO DATA
RECEPTEED
012B 22                      RET
012C C298  READY:      CLR     RI
012E E599                      MOV    A,SBUF
0130 22                      RET
```

;RECIVE RECIVE BYTE PUT TO A FROM SERIAL PORT

; WAIT UNTIL RECIVER BUFFER READY

; ENTRY: NO

; EXIT: A

```
0131 3098FD  RECIVE:  JNB      RI,RECIVE
0134 C298                      CLR     RI
0136 E599                      MOV    A,SBUF      ; GET IT
0138 12010F          LCALL   SEND      ; ECHO TO TERMINAL
013B 22                      RET
```

;SEND_PROMPT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

013C 120146  SEND_PROMPT:  LCALL  CR_LF
013F 900799                      MOV    DPTR,#PROMPT
0142 120756                      LCALL  SEND_STRING
0145 22                          RET

```

```

;CR_SEND SEND CR TO RS232

```

```

; ENTRY: NO

```

```

; EXIT: NO

```

```

;

```

```

0146 740D  CR_LF:  MOV    A,#CR
0148 12010F                      LCALL  SEND
014B 740A                      MOV    A,#LF
014D 12010F                      LCALL  SEND
0150 22                          RET

```

```

0013 =  XOFF:  EQU    13H

```

```

0011 =  XON:    EQU    11H

```

```

0000 =  BREAK: EQU    00H

```

```

000F =  EOF:    equ    0FH

```

```

;XOFF_SEND SEND XOFF TO RS232

```

```

; ENTRY: NO

```

```

; EXIT: NO

```

```

;

```

```

0151 7413  XOFF_SEND: MOV    A,#XOFF ;GET XOFF CHARACTER
0153 12010F                      LCALL  SEND
0156 22                          RET

```

```

;XON_SEND

```

```

;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
;
0157 7411    XON_SEND: MOV    A,#XON
0159 12010F                LCALL  SEND
015C 22                RET

```

;SPACE

```

015D 7420    SPACE:  MOV    A,#20H
015F 12010F                LCALL  SEND
0162 22                RET

```

;ASCII_BIN CONVERTS SINGLE ASCII CHARACTER TO SINGLE NIBBLE

```

;    BINARY
;    ENTRY: A ( 30-39 for 0-9, 41-46 for A-F, 61-66 for a-f)
;    EXIT: A
;    ignor lower case clr bit 5 before

```

```

0163 C3    ASCII_BIN:  CLR    C
0164 FE                MOV    R6,A
0165 9441                SUBB   A,#41H
0167 5005                JNC    ASCII_AF
0169 EE                MOV    A,R6
016A C3                CLR    C
016B 9430                SUBB   A,#30H
016D 22                RET
016E C3    ASCII_AF:   CLR    C
016F EE                MOV    A,R6
0170 9437                SUBB   A,#37H
0172 22                RET

```

;COMBINE DATA 2 BYTE TO SINGLE BYTE

```

;    ENTRY: R4= LOW NIBBLE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;          R5= HIGH NIBBLE
;          EXIT: A
;
0173 ED      COMBINE:      MOV      A,R5
0174 C4                      SWAP      A
0175 2C                      ADD       A,R4
0176 22                      RET

;GET_BYTE GET DATA FROM SERIAL PORT TWO BYTE SAVE TO
;          R5 AS A HIGH NIBBLE AND R4 AS A LOW NIBBLE
;          ENTRY: NO, serial port must be initialized before calling
;          EXIT: A
;
;
0177 120131  GET_BYTE: LCALL  RECIVE  ; FIRST READING MUST GO TO
R5 !!
017A 120163                      LCALL  ASCII_BIN
017D FD                      MOV      R5,A
017E 120131                      LCALL  RECIVE
0181 120163                      LCALL  ASCII_BIN
0184 FC                      MOV      R4,A
0185 120173                      LCALL  COMBINE
0188 22                      RET

```

;BIN_ASCII CONVERT BIN TO ASCII CODE

```

;          ENTRY: A
;          EXIT : A
;
0189      BIN_ASCII:
0189 540F      ANL      A,#0FH
018B FA      MOV      R2,A
018C C3      CLR      C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

018D 940A          SUBB  A,#0AH
018F 5004          JNC   ASCII_AF2
0191 EA           MOV   A,R2
0192 2430          ADD   A,#30H
0194 22           RET
0195 EA  ASCII_AF2: MOV   A,R2
0196 2437          ADD   A,#37H
0198 22           RET

```

;BYTE_ASCII CONVERT A TO ASCII IN R4, R5

```

0199 F5F0  BYTE_ASCII: MOV   B,A
019B 120189        LCALL  BIN_ASCII
019E FC          MOV   R4,A
019F E5F0        MOV   A,B
01A1 C4          SWAP  A
01A2 120189        LCALL  BIN_ASCII
01A5 FD          MOV   R5,A
01A6 22          RET

```

;SEND_ASCII SEND ASCII IN R5(HI) R4(LO) BYTE TO MONITOR

```

;   INPUT : A
;   OUTPUT : NONE

```

```

01A7 120199  SEND_ASCII: LCALL BYTE_ASCII
01AA ED          MOV   A,R5
01AB 12010F        LCALL SEND
01AE EC          MOV   A,R4
01AF 12010F        LCALL SEND
01B2 22          RET

```

;NIBBLE_SHIFT SHIFT FOUR DIGIT BCD NUMBER LEFT A NIBBLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; ENTRY : R0 POINTED TO LSD
; EXIT : NO

01B3 08 NIBBLE_SHIFT: INC R0
01B4 E6 MOV A,@R0
01B5 C4 SWAP A
01B6 54F0 ANL A,#0F0H
01B8 F9 MOV R1,A
01B9 18 DEC R0
01BA E6 MOV A,@R0
01BB C4 SWAP A
01BC FA MOV R2,A
01BD 540F ANL A,#0FH
01BF 29 ADD A,R1
01C0 08 INC R0
01C1 F6 MOV @R0,A
01C2 EA MOV A,R2
01C3 54F0 ANL A,#0F0H
01C5 18 DEC R0
01C6 F6 MOV @R0,A
01C7 22 RET

;SHIFT_16BIT SHIFT LEFT 1 BIT ADDRESS \$3D,\$3E

; \$3D LO BYTE
; \$3E HI BYTE
; ENTRY: NO
; EXIT: Carry Flag

01C8 C3 SHIFT_16BIT: CLR C ; {(\$3E,\$3D)} X 2
01C9 E53D MOV A,3DH
01CB 253D ADD A,3DH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

01CD F53D          MOV    3DH,A
01CF E53E          MOV    A,3EH
01D1 353E          ADDC   A,3EH
01D3 F53E          MOV    3EH,A
01D5 22            RET                ; CARRY GONE

```

;BIN_BCD CONVERT 16 BIT BINARY TO BCD

; SHIFT LEFT 1 BIT FROM MOST SIGNIFICANT BIT TO LEAST

SIGNIFICANT

; BIT , IF CARRY = 1 THEN BCD= (2^N + BCD)

;

; ENTRY: 16 BIT DATA

; \$3D = LO BYTE

; \$3E = HI BYTE

; EXIT: 6 DIGIT BCD

; \$36L = 0

; \$36H = 10

; \$37L = 100

; \$37H = 1000

; \$38L = 10000

; \$38H = 100000

```

01D6 753600  BIN_BCD: MOV    36H,#00 ; CLEAR BCD BEFORE

```

```

01D9 753700          MOV    37H,#00

```

```

01DC 753800          MOV    38H,#00

```

```

01DF 7900          MOV    R1,#00H

```

```

01E1 7F10          MOV    R7,#16D

```

```

01E3 1201C8  BIN_1:    LCALL  SHIFT_16BIT

```

```

01E6 5011          JNC     NEXT_BIT

```

; OK CARRY = 1 ADD BCD TO 2^N CONSTANT IN TABLE2

01E8 9001FF	MOV	DPTR,#TABLE2
01EB C3	CLR	C
01EC 7836	MOV	R0,#36H
01EE 7E03	MOV	R6,#03H
01F0 E9 BIN_2:	MOV	A,R1
01F1 93	MOVC	A,@A+DPTR
01F2 36	ADDC	A,@R0
01F3 D4	DAA	
01F4 F6	MOV	@R0,A
01F5 A3	INC	DPTR
01F6 08	INC	R0
01F7 DEF7	DJNZ	R6,BIN_2
01F9 09 NEXT_BIT:	INC	R1
01FA 09	INC	R1
01FB 09	INC	R1
01FC DFE5	DJNZ	R7,BIN_1
01FE 22	RET	
01FF 682703 TABLE2:	DFB	68H,27H,03H
0202 846301	DFB	84H,63H,01H
0205 928100	DFB	92H,81H,00H
0208 964000	DFB	96H,40H,00H
020B 482000	DFB	48H,20H,00H
020E 241000	DFB	24H,10H,00H
0211 120500	DFB	12H,05H,00H
0214 560200	DFB	56H,02H,00H
0217 280100	DFB	28H,01H,00H
021A 640000	DFB	64H,00H,00H
021D 320000	DFB	32H,00H,00H
0220 160000	DFB	16H,00H,00H
0223 080000	DFB	08H,00H,00H

0226 040000	DFB	04H,00H,00H
0229 020000	DFB	02H,00H,00H
022C 010000	DFB	01H,00H,00H

#####

```

;      TIMER 0 INTERRUPT SERVICE ROUTINE      #
;
;      enter to this routine every 1/10 Hz      #
;
;      update current time and date            #
;
;      check current day and time every 1 sec   #
;
;      if current time = set time the activates 8 output bit #
;
;                                              #

```

#####

```

022F C0D0      SERVICE_T0:  PUSH  PSW
0231 C0E0                      PUSH  ACC
0233 C0F0                      PUSH  B
0235 C082                      PUSH  DPL
0237 C083                      PUSH  DPH
0239 75D008     MOV    PSW,#00001000B ; SELECT REGISTER
BANK 1

```

; TIMER 0 SERVICE BODY

; cpl p3.5 ; check time base should be 10/2 Hz

```

023C 758C8B     MOV    TH0,#COUNT_H
023F 852F8A     MOV    TL0,tl0adj          ; get low byte
adjustable
0242 12030C     lcall   light_on
0245 0525     inc     sec100          ;

```

```

0247 E525          mov     a,sec100

0249 B40A27        cjne     a,#0AH,CHECK_SEC  ; update second every
10 counts

024C 1204AF        lcall    send_ca3162

024F 120146        lcall    cr_lf

0252 052C          inc      econo

0254 E52C          mov     a,econo

0256 B40507        cjne     a,#05h,skip_econo

0259 752C00        mov     econo,#00h

025C D21A          setb     blink

025E C2B7          clr      p3.7          ; turn on light every 5 second
                        ; since inc sec100 as binary

0260      skip_econo:

0260 120376        lcall    io_out

0263 301803        jnb      serial,no_send

0266 120335        LCALL    SEND_TIME

0269 752500        no_send: mov     sec100,#00h

026C E526          mov     a,sec

026E 2401          add     a,#01d

0270 D4            daa

0271 F526          mov     sec,a

0273      CHECK_SEC:

0273 E526          mov     a,sec

0275 B46010        cjne     a,#60H,CHECK_MIN

0278 301903        jnb      sim,min_send

027B 120335        lcall    send_time

027E      min_send:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

027E 752600          mov    sec,#00d
0281 E527            mov    a,min
0283 2401            add     a,#01d
0285 D4              daa
0286 F527            mov     min,a

0288      CHECK_MIN:
0288 E527            mov     a,min
028A B4600A          cjne    a,#60H,CHECK_HOUR
028D 752700          mov     min,#00d
0290 E528            mov     a,hour
0292 2401            add     a,#01d
0294 D4              daa
0295 F528            mov     hour,a

0297      CHECK_HOUR:
0297 E528            mov     a,hour
0299 B4240A          cjne    a,#24h,check_day
029C 752800          mov     hour,#00d
029F E529            mov     a,day
02A1 2401            add     a,#01d
02A3 D4              daa
02A4 F529            mov     day,a

```

```

02A6      check_day:
02A6 12039B          lcall    get_month_day
02A9 E529            mov     a,day
02AB B5F00A          cjne    a,b,check_month
02AE 752901          mov     day,#01d
02B1 E52A            mov     a,month
02B3 2401            add     a,#01d
02B5 D4              daa

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

02B6 F52A          mov     month,a

02B8      check_month:

02B8 E52A          mov     a,month
02BA B4130A        cjne     a,#13h,check_year
02BD 752A01        mov     month,#01d
02C0 E52B          mov     a,year
02C2 2401          add     a,#01d
02C4 D4           daa
02C5 F52B          mov     year,a

02C7      check_year:

02C7 D083          POP     DPH
02C9 D082          POP     DPL
02CB D0F0          POP     B
02CD D0E0          POP     ACC
02CF D0D0          POP     PSW

02D1 D28C          setb    tr0
02D3 32           RETI

```

;send_output send output (display) to terminal for simulate test

```

02D4      send_outputs:

02D4 12015D        lcall    space
02D7 12015D        lcall    space
02DA 7E01          mov     r6,#01d          ; tiny timer 8 bit I/O $21
02DC 9007EA        mov     dptr,#output_prompt
02DF 120756        lcall    send_string

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

02E2 7821          mov    r0,#output
02E4 7F08  outer:  mov    r7,#08d
02E6 E6           mov    a,@r0
02E7 F4           cpl     a                ; sink current drive circuit
02E8 F590         MOV     P1,A                ; send current output to port 1
02EA F4           cpl     a
02EB F5F0         mov     b,a
02ED E5F0  inner:  mov     a,b
02EF 33           rlc     a
02F0 F5F0         mov     b,a
02F2 400A         jc      on_contact
02F4 742D         mov     a,#"- "
02F6 12010F       lcall   send
02F9 12015D       lcall   space
02FC 8008         sjmp    next_bits

02FE             on_contact:
02FE 744F         mov     a,#"O"
0300 12010F       lcall   send
0303 12015D       lcall   space

0306 DFE5  next_bits: djnz   r7,inner
0308 08           inc     r0
0309 DED9         djnz   r6,outer
030B 22           ret

```

;light_on blink led shows time base is run

```

030C 301A0F  light_on: jnb    blink,led_on    ; if blink not set do not count down
030F E52D           mov    a,light
0311 04           inc     a
0312 F52D         mov     light,a

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0314 B40107          cjne  a,#01h,led_on
0317 752D00          mov   light,#00h
031A D2B7            setb  p3.7          ; off light
031C C21A            clr   blink        ; finished blink
031E 22             led_on:  ret

```

;send_tone send tone like beep beep to p2.2

```

031F 7F00    send_tone: mov  r7,#00h
0321 C2A2    tone0:  clr   p2.2
0323 7E50          mov   r6,#50h
0325 DEFE    tone1:  djnz  r6,tone1
0327 D2A2          setb  p2.2
0329 7E50          mov   r6,#50h

032B DEFE    tone2:  djnz  r6,tone2
032D DFF2          djnz  r7,tone0
032F 22             ret

```

;BCD_ASCII CONVERT BCD NUMBER TO ASCII CODE

```

;      ENTRY: A {LOW NIBBLE ONLY}
;      EXIT: A

```

```

0330 540F    BCD_ASCII: ANL   A,#0FH          ;GET ONLY LOW NIBBLE
0332 2430          ADD   A,#30H          ;CONVERT TO ASCII CODE
0334 22             RET

```

;send current time

```

0335          send_time:

```

0335 120146	LCALL	CR_LF
0338 E529	MOV	A, DAY
033A 1201A7	LCALL	SEND_ASCII
033D 742D	MOV	A, # "-"
033F 12010F	LCALL	SEND
0342 E52A	MOV	A, MONTH
0344 1201A7	LCALL	SEND_ASCII
0347 742D	MOV	A, # "-"
0349 12010F	LCALL	SEND
034C 7425	MOV	A, #25H
034E 1201A7	LCALL	SEND_ASCII
0351 E52B	MOV	A, YEAR
0353 1201A7	LCALL	SEND_ASCII
0356 12015D	LCALL	SPACE
0359 E528	MOV	A, HOUR
035B 1201A7	LCALL	SEND_ASCII
035E 743A	MOV	A, # ":"
0360 12010F	LCALL	SEND
0363 E527	MOV	A, MIN
0365 1201A7	LCALL	SEND_ASCII
0368 743A	MOV	A, # ":"
036A 12010F	LCALL	SEND
036D E526	MOV	A, SEC
036F 1201A7	LCALL	SEND_ASCII
0372 1202D4	lcall	send_outputs
0375 22	RET	

;io_out send output o1-o8 when current time = setting time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; scan setting time and compare current time
;
; if current time = set time then activate i/o
;
; els next set time till eof found

```

6

```

0376 7833    io_out:    mov     r0,#pgm_buffer
0378 E6      io0:      mov     a,@r0
0379 B40F01          cjne     a,#eof,io1
037C 22                      ret

```

```

037D 20B503  io1:      jb      p3.5,skip_day ; if p3.7 = 0 then skip day checking
0380 B52912          cjne     a,day,io4      ; check day

0383 08      skip_day: inc     r0
0384 E6              mov     a,@r0
0385 B5280E          cjne     a,hour,io2      ; check hour
0388 08              inc     r0
0389 E6              mov     a,@r0
038A B5270A          cjne     a,min,io3      ; check min
038D 08              inc     r0              ; current time = set time
038E E6              mov     a,@r0          ; get output
038F F521          mov     output,a        ; put to output byte
0391 F4              cpl     a              ; i.e., sink current is used for driving

```

LED

```

0392 F590          mov     pl,a            ; activate port 1 also
0394 22                      ret

```

```

0395 08      io4:      inc     r0
0396 08      io2:      inc     r0
0397 08      io3:      inc     r0
0398 08              inc     r0
0399 80DD          sjmp     io0

```

;GET_MONTH_DAY get day-end of each month, e.g. Jan=31, Feb=28, Mar=31,..

; ,etc.

; entry: current month 01,02,03,...,12

; exit: B = day-end

;

039B E52A get_month_day: mov a,month ; get current month

039D C3 clr c

039E 9410 subb a,#10h

03A0 4007 jc bcd_is_binary

03A2 E52A mov a,month

03A4 C3 clr c

03A5 9406 subb a,#06h

03A7 8002 sjmp convert_to_binary

03A9 E52A bcd_is_binary: mov a,month

03AB 14 convert_to_binary: dec a

03AC 9003B5 mov dptr,#day_end_table

03AF 93 movc a,@a+dptr

03B0 F5F0 mov b,a

03B2 05F0 inc b ; to compensate increment day

before

03B4 22 ret

03B5 31 day_end_table: dfb 31h ; Jan

03B6 28 dfb 28h ; Feb

03B7 31 dfb 31h ; Mar

03B8 30 dfb 30h ; Apr

03B9 31 dfb 31h ; May

03BA 30 dfb 30h ; Jun

03BB 31 dfb 31h ; Jul

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

03BC 31	dfb 31h ; Aug
03BD 30	dfb 30h ; Sep
03BE 31	dfb 31h ; Oct
03BF 30	dfb 30h ; Nov
03C0 31	dfb 31h ; Dec

;BUFFER_ASCII CONVERT BCD IN BUFFER START \$30 - \$35

; ENTRY: BCD NUMBER IN TRANSMIT BUFFER

; EXIT: ASCII CODE IN TRANSMIT BUFFER

03C1 7F06	BUFFER_ASCII: MOV	R7,#06H
03C3 7830		MOV R0,#30H
03C5 E6	NEXT_ASCII: MOV	A,@R0
03C6 120330		LCALL BCD_ASCII
03C9 F6		MOV @R0,A
03CA 08		INC R0
03CB DFF8		DJNZ R7,NEXT_ASCII
03CD 22		RET

;SEND_BUFFER SEND ASCII CODE IN TRANSMIT BUFFER TO SERIAL

PORT

; THEN DATA SEPERATOR /32/32

; ENTRY: ASCII CODE IN TRANSMIT BUFFER

; EXIT: NO

03CE 7F06	SEND_BUFFER:	MOV	R7,#06H
03D0 7835		MOV	R0,#35H
03D2 E6	NEXT_BYTE:	MOV	A,@R0
03D3 12010F		LCALL SEND	;SEND OUT TO SERIAL

PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
03D6 18          DEC      R0
03D7 DFF9          DJNZ     R7,NEXT_BYTE
03D9 120146        lcall     cr_lf
03DC 22           RET
```

#####

```

;
;      service_int0 read BCD data from CA3162E three digits
;      starting from MSD, LSD, and NSD as timing shown below
;
;      <-----> 2 ms
;      MSD  ----
;      NSD  ----
;      LSD  ----
;
;      rearrange into two bytes i.e., 0 MSD NSD LSD
;
;      integrates 5 consecutive bytes to form 5-point moving average
;
;      raw data filtering
```

```
03DD      service_int0:
03DD C0D0      PUSH  PSW
03DF C0E0      PUSH  ACC
03E1 C0F0      PUSH  B
03E3 C082      PUSH  DPL
03E5 C083      PUSH  DPH
03E7 75D010     MOV   PSW,#00010000B ; SELECT REGISTER BANK 2
```

```
03EA      service_body_int0:
03EA 1203F8     LCALL  SEND_PACKET
03ED D083      POP    DPH
03EF D082      POP    DPL
03F1 D0F0      POP    B
```

```

03F3 D0E0          POP      ACC
03F5 D0D0          POP      PSW
03F7 32            RETI

```

```

;#####

```

```

;SEND_PACKET

```

```

03F8      SEND_PACKET:

```

```

03F8 743A          mov      a,#"."
03FA 12010F        lcall     send
03FD 752400        mov      bcc,#00
0400 D210          setb     check_sum
0402 120451        lcall     scan_analog
0405 120441        lcall     send_kwhr
0408 C210          clr      check_sum
040A E524          mov      a,bcc
040C F4           cpl      a
040D 04           inc      a
040E 1201A7        lcall     send_ascii
0411 120146        lcall     cr_lf
0414 22           RET

```

```

;#####

```

```

;SERVICE_INT1 INCREMENT 6 DIGITS BCD COUNTER BY 1

```

```

;

```

```

0415      SERVICE_INT1:

```

```

0415 C0D0          PUSH     PSW
0417 C0E0          PUSH     ACC
0419 C0F0          PUSH     B
041B C082          PUSH     DPL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

041D C083	PUSH	DPH
041F 75D018	MOV	PSW,#00011000B ; SELECT REGISTER BANK 3
0422 C3	CLR	C
0423 7834	MOV	R0,#KWHR
0425 E6	MOV	A,@R0
0426 3401	ADDC	A,#01H
0428 D4	DAA	
0429 F6	MOV	@R0,A
042A 08	INC	R0
042B E6	MOV	A,@R0
042C 3400	ADDC	A,#00H
042E D4	DAA	
042F F6	MOV	@R0,A
0430 08	INC	R0
0431 E6	MOV	A,@R0
0432 3400	ADDC	A,#00H
0434 D4	DAA	
0435 F6	MOV	@R0,A
0436 D083	POP	DPH
0438 D082	POP	DPL
043A D0F0	POP	B
043C D0E0	POP	ACC
043E D0D0	POP	PSW
0440 32	RETI	

;SEND_KWHR

0441 E536	SEND_KWHR: MOV	A,KWHR+2
0443 1201A7	LCALL	SEND_ASCII
0446 E535	MOV	A,KWHR+1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0448 1201A7          LCALL    SEND_ASCII
044B E534            MOV      A,KWHR
044D 1201A7          LCALL    SEND_ASCII
0450 22              RET

```

;SCAN_ANALOG scan 8 channels analog input

```

0451      SCAN_ANALOG:
0451 E537      MOV      A,CHANNEL
0453 A2E0      mov      c,0E0h
0455 9290      mov      p1.0,c
0457 A2E1      mov      c,0E1h
0459 9291      mov      p1.1,c
045B A2E2      mov      c,0E2h
045D 9292      mov      p1.2,c
045F 12051B    lcall     delay_100ms

0462 30B7FD    wait_msd:
                        jnb     p3.7,wait_msd
0465      wait_msd_low:
0465 20B7FD      jb      p3.7,wait_msd_low
0468 120509      lcall     delay_1ms
046B 4390F0      orl      p1,#11110000b ; make p1.4-p1.7 high
046E E590      mov      a,p1
0470 54F0      anl      a,#11110000b
0472 C4        swap     a
0473 F530      mov      msd,a ; save MSD
0475 20B5FD    wait_lsd:
                        jb      p3.5,wait_lsd
0478 4390F0      orl      p1,#11110000b ; make p1.4-p1.7 high
047B E590      mov      a,p1
047D 54F0      anl      a,#11110000b

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

047F C4          swap    a
0480 F531          mov     lsd,a          ; save LSD
0482 20B4FD  wait_nsd:
                                jb      p3.4,wait_nsd
0485 4390F0          orl     p1,#11110000b  ; make p1.4-p1.7 high
0488 E590          mov     a,p1
048A 54F0          anl     a,#11110000b
048C C4          swap    a
048D F532          mov     nsd,a          ; save NSD

048F 1204C6          lcall   bcd_seven
0492 E530          mov     a,msd
0494 12010F          lcall   send
0497 E532          mov     a,nsd
0499 12010F          lcall   send
049C E531          mov     a,lsd
049E 12010F          lcall   send
                                ; lcall space

                                ; scan next channel

04A1 0537          inc     channel
04A3 E537          mov     a,channel
04A5 B523A9          cjne    a,CHANNEL_MAX,scan_analog ; SENDING LOOP
DEPEND ON THE
04A8 753700          mov     channel,#00h          ; NUMBER OF
MAXIMUM CHANNEL
04AB 7590F0          mov     p1,#11110000b          ; wait at channel 0
04AE 22          ret

```

;SEND_CA3162

04AF SEND_CA3162:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

04AF C2A8      clr      ex0
04B1 1204C6    LCALL    BCD_SEVEN
04B4 E530      MOV      A,MSD
04B6 12010F    LCALL    SEND
04B9 E532      MOV      A,NSD
04BB 12010F    LCALL    SEND
04BE E531      MOV      A,LSD
04C0 12010F    LCALL    SEND
04C3 D2A8      setb     ex0
04C5 22        RET

```

;BCD_SEVEN CONVERT MSD NSD LSD BCD DATA TO ASCII

REPRESENTATION

; ACCEPT 0A FOR NEGATIVE OVERRANGE CHANGE TO ---
; AND 0B FOR POSITIVE OVERRANGE CHANGE TO EEE

04C6 BCD_SEVEN:

```

04C6 E530      MOV      A,MSD
04C8 B40A05    CJNE     A,#0AH,OVER_MSD
04CB 75302D    MOV      MSD,#"-"
04CE 800C      SJMP     CHECK_NSD

```

04D0 B40B05 OVER_MSD: CJNE A,#0BH,BCD_MSD

04D3 753045 MOV MSD,#"E"

04D6 8004 SJMP CHECK_NSD

04D8 2430 BCD_MSD: ADD A,#30H

04DA F530 MOV MSD,A

04DC E532 CHECK_NSD: MOV A,NSD

04DE B40A05 CJNE A,#0AH,OVER_NSD

04E1 75322D MOV NSD,#"-"

04E4 800C SJMP CHECK_LSD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

04E6 B40B05  OVER_NSD: CJNE    A,#0BH,BCD_NSD
04E9 753245                      MOV     NSD,#"E"
04EC 8004                      SJMP     CHECK_LSD

04EE 2430  BCD_NSD:  ADD     A,#30H
04F0 F532                      MOV     NSD,A

04F2 E531  CHECK_LSD: MOV     A,LSD
04F4 B40A05                      CJNE    A,#0AH,OVER_LSD
04F7 75312D                      MOV     LSD,#"-"
04FA 800C                      SJMP     CHECK_FINISH

04FC B40B05  OVER_LSD: CJNE    A,#0BH,BCD_LSD
04FF 753145                      MOV     LSD,#"E"
0502 8004                      SJMP     CHECK_FINISH

0504 2430  BCD_LSD:  ADD     A,#30H
0506 F531                      MOV     LSD,A

0508 22      CHECK_FINISH: RET

0509 7B01  delay_1ms:  mov     r3,#01h      ; delay 1 ms
050B 7C95  delay_1mss: mov     r4,#095h      ; ORIGINAL 90H
050D DCFE                      djnz     r4,$
050F DBFA                      djnz     r3,delay_1mss
0511 22                      ret

0512 7B02  delay_2ms:  mov     r3,#02h      ; delay 2 ms
0514 7C95  delay_2mss: mov     r4,#095h      ; ORIGINAL 90H
0516 DCFE                      djnz     r4,$
0518 DBFA                      djnz     r3,delay_2mss

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
051A 22                ret-

051B 7BC8    delay_100ms:  mov    r3,#200d
051D 7C95    delay_100mss: mov    r4,#95h
051F DCFE                                djnz    r4,$
0521 DBFA                                djnz    r3,delay_100mss
0523 22                ret
```

;COMPARE TWO 16 BIT BINARY NUMBER
;
; ENTRY: R0 FIRST OPERAND
;
; R1 SECOND OPERAND
;
; EXIT: C=1 FIRST OPERAND < SECOND OPERAND
;
; C=0 FIRST OPERAND >= SECOND OPERAND

```
0524 C3    COMPARE:    CLR    C
0525 E6                                MOV    A,@R0
0526 97                                SUBB   A,@R1
0527 08                                INC    R0
0528 09                                INC    R1
0529 E6                                MOV    A,@R0
052A 97                                SUBB   A,@R1
052B 22                                RET
```

```
#####  
;#                                MAIN PROGRAM                                #  
#####
```

```
0040 =    CURRENT_TIME_L:    EQU    40H
```

```

0041 =    CURRENT_TIME_H:    EQU    41H
0042 =    CURRENT_PROG_L:    EQU    42H
0043 =    CURRENT_PROG_H:    EQU    43H

```

```

0092 =    COUNT_L:                EQU    92h    ;7AH    ; <--- find adjust in
case of oscillation

```

; frequency is not a

3.579545 MHz

```

008B =    COUNT_H:                EQU    8BH    ; interrupt every 3.579545
MHz/12*(65536-29830)

```

; = 10 Hz

```

052C    MAIN_TIMER:

```

```

052C 758160    MOV    SP,#60H                ; stack area start $60-$7F

```

```

052F 752D00    mov    light,#00h

```

```

0532 7590FF    mov    pl,#11111111b        ; off output

```

```

0535 C218      clr    serial

```

```

0537 C21A      clr    blink

```

```

0539 752500    mov    sec100,#00h

```

```

053C 752C00    mov    econo,#00h

```

```

053F 752F92    mov    tl0adj,#92h            ; for adjustment later via user

```

calibrate

```

0542 E52E      mov    a,warm_code

```

```

0544 B42502    cjne    a,#"%",cold_boot

```

```

0547 8027      sjmp    warm_boot

```

```

0549    cold_boot:

```

```

0549 752E25    mov    warm_code,#"% "

```

```

054C 75330F    mov    pgm_buffer,#eof        ; blank pgm at power up

```

```

054F 752600    MOV    SEC,#00D

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0552 752730      MOV      MIN,#30H
0555 752812      MOV      HOUR,#12H
0558 752929      MOV      DAY,#29H
055B 752A04      MOV      MONTH,#04H
055E 752B37      MOV      YEAR,#37H
0561 752100      mov      output,#00h
0564 753400      mov      kwhr,#00h
0567 753500      mov      kwhr+1,#00h
056A 753600      mov      kwhr+2,#00h
056D 752308      MOV      CHANNEL_MAX,#08H      ; maximum channel to be scan is
8 channel

```

```

0570      warm_boot:
0570 D2AF      SETB      EA      ; ENABLE ALL
INTERRUPT
0572 D2A9      SETB      ET0      ; ENABLE TIMER 0
INTERRUPT
0574 758901      MOV      TMOD,#00000001B      ; SET MODE 1 T0 (16 BIT
DIVIDER)
0577 120100      LCALL      INIT
057A D2B9      SETB      PT0      ; TIMER 0 HIGHEST PRIORITY
; MOV DPTR,#TITLE
; LCALL SEND_STRING
057C 758C8B      mov      th0,#count_h      ;start timer
057F 758A92      mov      tl0,#count_l
;setb tr0      ;start timer
0582 D28A      setb      it1      ; enable negative going trigger for
0584 D2AA      setb      ex1      ; external interrupt 0
0586 753700      mov      channel,#00h
0589 D2A8      setb      ex0      ; enable trigger read
058B D288      setb      it0      ; negative going

```

```

058D 80F    TEST10: sjmp $                                ; wait trigger read

058F 3098FB    jnb      ri,test10                            ; polls receive buffer and wait
command
0592 C298      clr      ri
0594 E599      mov      a,sbuf

0596 B4410E    cjne     a,#"A",skip_modem_init ; SKIP MODEM INITIALISED
STRING
0599 120131    lcall     recive                            ; SEARCH "AT" STRING
059C B45408    cjne     a,#"T",skip_modem_init

059F 120131    dummy_init:
                                lcall     recive
05A2 B40DFA    cjne     a,#cr,dummy_init      ; UNTIL END OF STRING
05A5 80E6      sjmp     test10

05A7          skip_modem_init:
05A7 4420      orl      a,#20h                ; accept upper & lower
05A9 B46F02    cjne     a,#"o",com10
05AC 804E      sjmp     off_serial

05AE B47402    com10:  cjne     a,#"t",com11
05B1 8050      sjmp     transmit_time

05B3 B47A02    com11:  cjne     a,#"z",com12
05B6 804F      sjmp     adjust_clock

05B8 B46502    com12:  cjne     a,#"e",com13
05BB 807D      sjmp     set_time

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

05BD B46403 com13: cjne a,#"d",com14

05C0 020658 ljmp set_date

05C3 B43F03 com14: cjne a,#"?",com15

05C6 020676 ljmp print_com

05C9 B47303 com15: cjne a,#"s",com16

05CC 020682 ljmp speed_up

05CF B47003 com16: cjne a,#"p",com17

05D2 020687 ljmp program_time

05D5 B47203 com17: cjne a,#"r",com18

05D8 0206EA ljmp read_pgm

05DB B46303 com18: cjne a,#"c",com19

05DE 020741 ljmp clear_io

05E1 B42F03 com19: cjne a,#"/",com20

05E4 020750 ljmp once

05E7 B46D03 com20: cjne a,#"m",com21

05EA 020622 ljmp set_channel_max

05ED com21:

05ED 12013C lcall send_prompt

05F0 02058D LJMP TEST10

05F3 758A92 RUN2: MOV TL0,#COUNT_L

05F6 758C8B MOV TH0,#COUNT_H

05F9 D28C SETB TR0 ; START TIMER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

05FB 22 RET

;off_serial stop transmit current date and time to terminal

```
05FC      off_serial:
05FC C218      clr      serial
05FE 12013C      lcall    SEND_PROMPT
0601 808A      sjmp     test10
```

;transmit_time start send current date and time to terminal

```
0603 D218      transmit_time: setb      serial
0605 8086      sjmp     test10
```

;adjust_clock change low byte of timer 0

```
0607 120146      adjust_clock: lcall    cr_lf
060A E52F      mov     a,tl0adj
060C 1201A7      lcall    send_ascii
060F 120146      lcall    cr_lf
0612 743F      mov     a,#"?"
0614 12010F      lcall    send
0617 120177      lcall    get_byte
061A F52F      mov     tl0adj,a
061C 12013C      lcall    send_prompt
061F 02058D      ljmp     test10
```

;set_channel_max change maximum channel to be scanned

```
0622      set_channel_max:
0622 120146      lcall    cr_lf
0625 E523      mov     a,channel_max
0627 1201A7      lcall    send_ascii
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

062A 120146      lcall    cr_lf
062D 743F        mov     a,#"?"
062F 120177      lcall    get_byte
0632 F523        mov     channel_max,a
0634 12013C      lcall    send_prompt
0637 02058D      ljmp     test10

```

;set_time enter current time

```

063A 9007F9      set_time: mov    dptr,#time_set
063D 120756      lcall    send_string
0640 120177      lcall    get_byte
0643 F528        mov     hour,a      ; save hour
0645 120177      lcall    get_byte
0648 F527        mov     min,a       ; save min
064A 120177      lcall    get_byte
064D F526        mov     sec,a       ; save sec
064F 120335      lcall    send_time
0652 12013C      lcall    send_prompt
0655 02058D      ljmp     test10

```

;set_date enter current date

```

0658 900811      set_date: mov    dptr,#date_set
065B 120756      lcall    send_string
065E 120177      lcall    get_byte
0661 F529        mov     day,a
0663 120177      lcall    get_byte
0666 F52A        mov     month,a
0668 120177      lcall    get_byte
066B F52B        mov     year,a
066D 120335      lcall    send_time

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0670 12013C      lcall    send_prompt
0673 02058D      ljmp     test10
```

```
;print_com shows command listing
```

```
0676 90079C      print_com: mov    dptr,#command
0679 120756              lcall    send_string
067C 12013C              lcall    send_prompt
067F 02058D              ljmp     test10
```

```
;speed_up simulate output at high speed
```

```
0682 B219      speed_up:  cpl     sim
0684 02058D      ljmp     test10
```

```
;program_time wait file from terminal write to on chip ram
```

```
;      start $30-$63 60 bytes or 20 line accepted
```

```
;      program format:
```

```
;      :01 05 2030 1 1 0 1 0 0 1 1crlf
```

```
;      :02 06 2120 1 0 0 0 1 1 0 1crlf
```

```
;      :00crlf
```

```
;      where
```

```
;      : initiate start of line
```

```
;      01 record 1,2,3,...
```

```
;      05 date
```

```
;      2030 time
```

```
;      1 1 0 1 0 0 1 1 digital output
```

```
;      crlf end of line flag
```

```
;      00 end of record flag
```

```
;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ; maximum program storage provides 7 days, i.e. each day require four
- ; byte, thus total byte is 28 bytes from \$30 - \$50

```

0687 C28C    program_time: clr      tr0                ; stop timer before so after
program
0689 9007CF                mov     dptr,#program    ; loading is finished actual time
need
068C 120756                lcall    send_string    ; to be adjusted again.
068F 120146                lcall    cr_lf
0692 7833                mov     r0,#pgm_buffer

0694 120131    pgm1:        lcall    recive
0697 B43AFA                cjne    a,#":",pgm1

069A 120177                lcall    get_byte    ; skip line

069D B40002                cjne    a,#00h,pgm2    ; end of file detected
06A0 8032                sjmp    stop_pgm

06A2                pgm2:
06A2 120131                lcall    recive    ; skip space
06A5 120177                lcall    get_byte    ; get date byte
06A8 F6                mov     @r0,a        ; save date

06A9 120131                lcall    recive    ; skip space
06AC 08                inc     r0

06AD 120177                lcall    get_byte    ; get hour byte
06B0 F6                mov     @r0,a        ; save hour

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

06B1 08          inc      r0

06B2 120177      lcall    get_byte

06B5 F6          mov      @r0,a          ; save min

06B6 08          inc      r0

06B7 7F08        mov      r7,#08d        ; get control I/O bit

06B9 120131      pgm3: lcall    recive          ; skip space

06BC 120131      lcall    recive
                  ; mov      b,a
                  ; lcall    xoff_send        ; pause string sending from
terminal
                  ; mov      a,b
06BF B43003      cjne     a,#"0",pgm4
06C2 C3          clr      c
06C3 8001        sjmp     pgm5

06C5 D3          pgm4: setb     c
06C6 E6          pgm5: mov      a,@r0
06C7 33          rlc      a
06C8 F6          mov      @r0,a
                  ;lcall    xon_send        ; ready to receive again

06C9 DFEE        djnz     r7,pgm3          ; loop 8 times

06CB 120131      pgm6: lcall    recive

06CE B40AFA      cjne     a,#1f,pgm6

06D1 08          inc      r0

06D2 80C0        sjmp     pgm1          ; read next line

06D4 760F        stop_pgm: mov    @r0,#eof    ; end of file

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

06D6 120131      lcall    recive
06D9 B40AF8      cjne     a,#lf,stop_pgm
06DC 9007BE      mov      dptr,#finish_pgm
06DF 120756      lcall    send_string
06E2 12013C      lcall    send_prompt

06E5 D28C        setb     tr0                ; resume timer interrupt again
06E7 02058D      ljmp     test10

```

```

;read_pgm read internal program for checking
;
;   reteive internal program send to terminal for checking
;
;   or changing current settings

```

```

06EA 120146  read_pgm:  lcall    cr_lf
06ED 7833          mov      r0,#pgm_buffer
06EF E6          read2:   mov      a,@r0
06F0 B40F08      cjne     a,#eof,read3 ; until eos (0FH) found
06F3 900829      mov      dptr,#end_of_file
06F6 120756      lcall    send_string
06F9 8040          sjmp     read4

```

```

06FB 1201A7  read3:   lcall    send_ascii ; send day
06FE 12015D          lcall    space
0701 08          inc      r0
0702 E6          mov      a,@r0                ; get hour
0703 1201A7      lcall    send_ascii ; send hour
0706 743A        mov      a,#":"
0708 12010F      lcall    send
070B 08          inc      r0
070C E6          mov      a,@r0
070D 1201A7      lcall    send_ascii ; send min

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0710 12015D      lcall    space

0713 9007EA      mov     dptr,#output_prompt

0716 120756      lcall    send_string

0719 08          inc     r0

071A E6          mov     a,@r0


071B 7F08        mov     r7,#08h

071D 33      read_io:  rlc     a

071E F5F0        mov     b,a

0720 4007        jc      send_on

0722 742D        mov     a,#"-

0724 12010F      lcall    send

0727 8005        sjmp    next_io_bit

0729 744F      send_on: mov     a,#"O"

072B 12010F      lcall    send

072E      next_io_bit:

072E 12015D      lcall    space

0731 E5F0        mov     a,b

0733 DFE8        djnz    r7,read_io


0735 120146      lcall    cr_lf

0738 08          inc     r0

0739 80B4        sjmp    read2


073B      read4:

073B 12013C      lcall    send_prompt

073E 02058D      ljmp    test10
```

;clear_io clear all 8 bit p1.0-p1.7

```
0741 752100  clear_io: mov    output,#00000000b
0744 7590FF                mov    p1,#11111111b    ; clear port1 also
0747 120335                lcall   send_time        ; display results
074A 12013C                lcall   send_prompt
074D 02058D                ljmp    test10
```

;once read current time once

```
0750 1203F8  once:    LCALL   SEND_PACKET
0753 02058D                ljmp    test10
```

;SEND_STRING SEND STRING CONSTANT TO TERMINAL

; ENTRY: DPTR

; EXIT: FOUND EOS

;

```
0756 E4      SEND_STRING: CLR    A
0757 93                        MOVC   A,@A+DPTR
0758 B41001                CJNE    A,#EOS,SEND_STRING1
075B 22                        RET
```

```
075C C082  SEND_STRING1: PUSH   DPL
```

```
075E C083                        PUSH   DPH
```

```
0760 12010F                LCALL   SEND
```

```
0763 D083                        POP     DPH
```

```
0765 D082                        POP     DPL
```

```
0767 A3                        INC     DPTR
```

```
0768 80EC                        SJMP    SEND_STRING
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; STRING CONSTANT AREA

076A 0D0A414E41TITLE: DFB cr,lf,"ANALOG SENDER",CR,LF
077B 4465736967 DFB "Designed by Wichit Sirichote"
0797 0D0A DFB CR,LF
0799 262010 PROMPT: DFB "& ",EOS
079C 0D0A command: dfb cr,lf
079E 2F20747269 dfb "/" trigger read", cr,lf
07AE 6D206D6178 dfb "m max channel", cr,lf
07BD 10 dfb eos

07BE 50726F6772finish_pgm: dfb "Program stored",cr,lf,eos
07CF 0D0A526561program: dfb cr,lf,"Ready to receive program",eos
07EA output_prompt:
07EA 4F55545055 dfb "OUTPUT [1..8] ",eos
07F9 0D0A456E74 time_set:dfb cr,lf,"Enter current time > ",eos
0811 0D0A456E74 date_set: dfb cr,lf,"Enter current date > ",eos
0829 454F4610 end_of_file: dfb "EOF",eos
082D 574841543F WHAT?: DFB "WHAT?",CR,LF,EOS

0000 END

00D6 AC	00E0 ACC	0607 ADJUST_CLOCK
0120 AGAIN	016E ASCII_AF	0195 ASCII_AF2
0163 ASCII_BIN	00F0 B	0024 BCC
0330 BCD_ASCII	03A9 BCD_IS_BINARY	0504 BCD_LSD
04D8 BCD_MSD	04EE BCD_NSD	04C6 BCD_SEVEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0007 BELL	01E3 BIN_1	01F0 BIN_2
0189 BIN_ASCII	01D6 BIN_BCD	001A BLINK
0000 BREAK	03C1 BUFFER_ASCII	0199 BYTE_ASCII
0037 CHANNEL	0023 CHANNEL_MAX	02A6 CHECK_DAY
0508 CHECK_FINISH	0297 CHECK_HOUR	04F2 CHECK_LSD
0288 CHECK_MIN	02B8 CHECK_MONTH	04DC CHECK_NSD
0273 CHECK_SEC	0010 CHECK_SUM	02C7 CHECK_YEAR
0741 CLEAR_IO	0549 COLD_BOOT	05AE COM10
05B3 COM11	05B8 COM12	05BD COM13
05C3 COM14	05C9 COM15	05CF COM16
05D5 COM17	05DB COM18	05E1 COM19
05E7 COM20	05ED COM21	0173 COMBINE
079C COMMAND	0524 COMPARE	03AB CONVERT_TO_BINARY
008B COUNT_H	0092 COUNT_L	000D CR
0146 CR_LF	0043 CURRENT_PROG_H	0042 CURRENT_PROG_L
0041 CURRENT_TIME_H	0040 CURRENT_TIME_L	00D7 CY
000F CYCLE_COUNT	0038 DATA_BUFFER	0811 DATE_SET
0029 DAY	03B5 DAY_END_TABLE	051B DELAY_100MS
051D DELAY_100MSS	0509 DELAY_1MS	050B DELAY_1MSS
0512 DELAY_2MS	0514 DELAY_2MSS	0033 DIGIT
0083 DPH	0082 DPL	059F DUMMY_INIT
00AF EA	002C ECONO	0829 END_OF_FILE
000F EOF	0010 EOS	00AC ES
00A9 ET0	00AB ET1	00A8 EX0
00AA EX1	00D5 F0	07BE FINISH_PGM
0177 GET_BYTE	039B GET_MONTH_DAY	0122 GET_RI
0028 HOUR	0089 IE0	008B IE1
00A8 IEC	0100 INIT	02ED INNER
0378 IO0	037D IO1	0396 IO2
0397 IO3	0395 IO4	0376 IO_OUT
00B8 IPC	0088 IT0	008A IT1
000B KEY	0034 KWHR	0008 LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

031E LED_ON	000A LF	002D LIGHT
030C LIGHT_ON	0031 LSD	052C MAIN_TIMER
0027 MIN	027E MIN_SEND	002A MONTH
0030 MSD	03C5 NEXT_ASCII	01F9 NEXT_BIT
0306 NEXT_BITS	03D2 NEXT_BYTE	072E NEXT_IO_BIT
01B3 NIBBLE_SHIFT	011D NO_CHECK_SUM	0269 NO_SEND
0032 NSD	05FC OFF_SERIAL	0750 ONCE
02FE ON_CONTACT	02E4 OUTER	0021 OUTPUT
07EA OUTPUT_PROMPT	00D2 OV	04FC OVER_LSD
04D0 OVER_MSD	04E6 OVER_NSD	00D0 P
0080 P0	0090 P1	0090 P1.0
0091 P1.1	0092 P1.2	0093 P1.3
0094 P1.4	0095 P1.5	0096 P1.6
0097 P1.7	00A0 P2	00A0 P2.0
00A1 P2.1	00A2 P2.2	00B0 P3
00B0 P3.0	00B1 P3.1	00B2 P3.2
00B3 P3.3	00B4 P3.4	00B5 P3.5
00B6 P3.6	00B7 P3.7	0087 PCON
0694 PGM1	06A2 PGM2	06B9 PGM3
06C5 PGM4	06C6 PGM5	06CB PGM6
0033 PGM_BUFFER	0676 PRINT_COM	07CF PROGRAM
0687 PROGRAM_TIME	0799 PROMPT	00BC PS
00D0 PSW	00B9 PT0	00BB PT1
00B8 PX0	00BA PX1	009A RB8
06EF READ2	06FB READ3	073B READ4
012C READY	071D READ_IO	06EA READ_PGM
004A RECEIVE_BUFFER	0131 RECIVE	011E RECIVES
009C REN	0098 RI	00D3 RS0
00D4 RS1	05F3 RUN2	0049 SAVE_H
0048 SAVE_L	0099 SBUF	0451 SCAN_ANALOG
0098 SCON	0026 SEC	0025 SEC100
010F SEND	01A7 SEND_ASCII	03CE SEND_BUFFER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

04AF SEND_CA3162	0441 SEND_KWHR	0729 SEND_ON
02D4 SEND_OUTPUTS	03F8 SEND_PACKET	013C SEND_PROMPT
0756 SEND_STRING	075C SEND_STRING1	0335 SEND_TIME
031F SEND_TONE	0018 SERIAL	03EA SERVICE_BODY_INT0
03DD SERVICE_INT0	0415 SERVICE_INT1	022F SERVICE_T0
0622 SET_CHANNEL_MAX	0658 SET_DATE	063A SET_TIME
01C8 SHIFT_16BIT	0019 SIM	0383 SKIP_DAY
0260 SKIP_ECONO	05A7 SKIP_MODEM_INIT	009F SM0
009E SM1	009D SM2	0081 SP
015D SPACE	0682 SPEED_UP	0038 STATE
06D4 STOP_PGM	01FF TABLE2	009B TB8
0088 TCON	058D TEST10	008D TF0
008F TF1	008C TH0	008D TH1
0099 TI	07F9 TIME_SET	076A TITLE
008A TL0	002F TLOADJ	008B TL1
0089 TMOD	0321 TONE0	0325 TONE1
032B TONE2	1000 TOP_FIFO	008C TR0
008E TR1	0603 TRANSMIT_TIME	000E VALVE
0475 WAIT_LSD	0462 WAIT_MSD	0465 WAIT_MSD_LOW
0482 WAIT_NSD	0570 WARM_BOOT	002E WARM_CODE
082D WHAT?	0013 XOFF	000A XOFF_FLAG
0151 XOFF_SEND	0011 XON	0157 XON_SEND
002B YEAR	000D ZEROING	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมเครื่อง Printing A/D Converter

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 1

ptd08.asm

1 ; Special Project Title Printing A/D Converter
2
3 ; Name Mr.Anantachai Naijit
4
5 ; Mr.Sanchayot Sakuljittacharoen
6
7 ; Special Project Advisor Asst. Prof.Wichit Sirichote
8
9 ; Asst. Prof.Tanaporn Leelawattananon
10
11 ; Department Applied Physics
12
13 ; Academic Year 1997
14
15 ; PROGRAM CONTROL DS1202+89C51+SOUND+LCD+SWICH
16 ; PROGRAMMER CHAI + SAN
17

0000 18 ORG 000H
0000 020726 19 LJMP MAIN
0086= 20 IO EQU P0.6 ;P3.5 , P2.5
0087= 21 SCLK EQU P0.7 ;P3.4 , P2.4
0085= 22 RST\ EQU P0.5 ;P3.7 , P2.3
23
0084= 24 BUSY EQU P0.4 ;P3.3 , P2.1
0083= 25 STROBE EQU P0.3 ;P3.2 , P2.0

26

0082=	27	RS	EQU	P0.2
0081=	28	RW	EQU	P0.1
0080=	29	EN	EQU	P0.0

30

0001=	31	SOH	EQU	01H
0004=	32	EOT	EQU	04H
000D=	33	CR	EQU	0DH
000A=	34	LF	EQU	0AH
0010=	35	EOS	EQU	10H

36

0017=	37	NOW_H_L	EQU	17H
0018=	38	GET_LCD	EQU	18H
0019=	39	SCAN_LCD	EQU	19H
001A=	40	GET_R2	EQU	1AH
001B=	41	GET_R4	EQU	1BH
001C=	42	PRN_S_H	EQU	1CH
001D=	43	PRN_S_L	EQU	1DH
001E=	44	NOW_S_H	EQU	1EH
001F=	45	NOW_S_L	EQU	1FH
0020=	46	TYPE	EQU	20H
0021=	47	ST	EQU	21H
0022=	48	CONTROL_MAIN	EQU	22H
0023=	49	MODE	EQU	23H
0028=	50	D	EQU	28H
0030=	51	NOW_H_H	EQU	30H
0031=	52	PRN_M_H	EQU	31H
0032=	53	PRN_M_L	EQU	32H
0033=	54	NOW_M_H	EQU	33H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0034=      55      NOW_M_L      EQU      34H
00B1=      56      TX1          EQU      P3.1      ;P3.1
0035=      57      DATA_BUFFER EQU      35H

0100      58              ORG      100H

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 2

ptd08.asm

```

59
60
61
62      ;////////////////// DEFINE RS232 PORT ////////////////////////
0100 759852 63 INT_RS232: MOV      SCON,#52H      ; SET SERIAL IN
MODE 1
0103 758DFD 64              MOV      TH1,#0FDH      ; SET BAUD RATE
9600
0106 758921 65              MOV      TMOD,#21H      ; SET TIMER1 IMN
MODE 2
0109 E587    66              MOV      A,PCON
010B 547F    67              ANL      A,#7FH
010D F587    68              MOV      PCON,A
010F D28E    69              SETB     TR1          ; INITIAL RS232
0111 22      70              RET
71
0112 758921 72 INT:      MOV      TMOD,#00100001B
0115 759852 73              MOV      SCON,#01010010B
0118 758780 74              MOV      PCON,#10000000B      ; SMOD = 1
011B 758DFE 75              MOV      TH1,#0FEH          ; TIMER1 LOAD
VALUE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

011E D28E 76 SETB TR1 ; START TIMER1
0120 22 77 RET
78

0121 3099FD 79 SEND: JNB TI,SEND
0124 C299 80 CLR TI
0126 F599 81 MOV SBUF,A
0128 22 82 RET
83
84

0129 3098FD 85 RECIVE: JNB RI,RECIVE
012C C298 86 CLR RI
012E E599 87 MOV A,SBUF
0130 22 88 RET
89
90
91

92 ; SEND_STRING:SEND STRING IN DATA TABLE TO PRINTER
PORT
93 ;
94

0131 E4 95 SEND_STRING: CLR A
0132 93 96 MOVC A,@A+DPTR
0133 B41001 97 CJNE A,#EOS,SEND_STR1
0136 22 98 RET
99
100
0137 101 SEND_STR1:
0137 12015A 102 CALL PRINT_STR
013A A3 103 INC DPTR
013B 80F4 104 SJMP SEND_STRING

```

105
106 ;//////////////////HEX TO ASCII //////////////////
107 ;INPUT :A
108 ;OUTPUT :R2,R3
109 ;USED REGISTER :A,R2,R3
110

```

```

013D C0E0 111 HTOA:    PUSH    ACC
013F C4    112          SWAP    A
0140 12014B 113          CALL    HTOAS
0143 FA    114          MOV     R2,A
0144 D0E0 115          POP     ACC
0146 12014B 116          CALL    HTOAS
0149 FB    117          MOV     R3,A

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 3
ptd08.asm

```

014A 22    118          RET
119
014B 540F 120 HTOAS:    ANL     A,#0FH
014D B40A00 121          CJNE    .A,#0AH,$+3
0150 5003 122          JNC      HTOAS1
0152 4430 123          ORL     A,#30H
0154 22    124          RET
125
0155 9409 126 HTOAS1:    SUBB    A,#9
0157 4440 127          ORL     A,#40H
0159 22    128          RET
129

```

```

130 ;*****

```

```

131 ;PRINT_STR: P2.0 = STROBE SIGNAL (INPUT PRINTER)
132 ;          P2.1 = BUSY SIGNAL (OUTPUT
PRINTER)

133 ;*****
015A      134      PRINT_STR:
          135
015A 2084FD 136          JB      BUSY,PRINT_STR
015D F590   137          MOV     P1,A
015F C283   138          CLR     STROBE
0161 D283   139          SETB    STROBE
0163 22     140          RET
          141
          142          ;////////// SEND DATA //////////
          143
          144
0164      145      SEND_DATA:
0164 7835   146          MOV     R0,#DATA_BUFFER
0166 120181 147          LCALL    GET_DATA
0169 7836   148          MOV     R0,#36H
016B 120192 149          LCALL    PRT_DATA
016E 90089E 150          MOV     DPTR,#TEXT1
0171 120131 151          LCALL    SEND_STRING
0174 D2B1   152          SETB    TX1
0176 22     153          RET
          154
0177      155      SEND_TX:
0177 D2B1   156          SETB    TX1
0179 00     157          NOP
017A 00     158          NOP
017B 00     159          NOP
017C C2B1   160          CLR     TX1
017E 00     161          NOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

017F 00	162	NOP
0180 22	163	RET
	164	
0181	165	GET_DATA:
0181 120177	166	LCALL SEND_TX
0184 120129	167	LCALL RECIVE
0187 B43A02	168	CJNE A,#:',GET
018A 7835	169	MOV R0,#35H
018C F6	170 GET:	MOV @R0,A
018D 08	171	INC R0
018E B856F0	172	CJNE R0,#56H,GET_DATA
0191 22	173	RET
	174	
0192	175	PRT_DATA:
0192 7903	176	MOV R1,#03H

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 4

ptd08.asm

0194 E6	177	SD: MOV A,@R0
0195 12015A	178	LCALL PRINT_STR
0198 08	179	INC R0
0199 D9F9	180	DJNZ R1,SD
019B 9009AF	181	MOV DPTR,#TEXT5
019E 120131	182	LCALL SEND_STRING
01A1 B84EEE	183	CJNE R0,#4EH,PRT_DATA
01A4 22	184	RET
	185	
	186	
	187	;***** SETTIME SUB. *****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	188	
01A5	189	SETTIME:
01A5 7E8E	190	MOV R6,#8EH ; WRITE PROTECTION
COMMAND		
01A7 7F00	191	MOV R7,#00H
01A9 1201DE	192	LCALL BYTEWR
	193	
01AC 7E80	194	MOV R6,#80H ;WRITE SECOND AND CLR
CHFLAG		
01AE 7F00	195	MOV R7,#00H ;SEC =00
01B0 1201DE	196	LCALL BYTEWR
	197	
01B3 7E82	198	MOV R6,#82H ;WRITE MINUTE
01B5 7F10	199	MOV R7,#10H ;MIN = 08
01B7 1201DE	200	LCALL BYTEWR
	201	
01BA 7E84	202	MOV R6,#84H ; WRITE HOUR
01BC 7F09	203	MOV R7,#09H ; HOUR = 12
01BE 1201DE	204	LCALL BYTEWR
	205	
01C1 7E86	206	MOV R6,#86H ;WRITE DATE
01C3 7F26	207	MOV R7,#26H ;DATE = 24
01C5 1201DE	208	LCALL BYTEWR
	209	
01C8 7E88	210	MOV R6,#88H ;WRITE MONTH
01CA 7F12	211	MOV R7,#12H ;MONTH = 12
01CC 1201DE	212	LCALL BYTEWR
	213	
01CF 7E8C	214	MOV R6,#8CH ; WRITE YEAR
01D1 7F97	215	MOV R7,#97H ; YEAR = 97
01D3 1201DE	216	LCALL BYTEWR
	217	

```

01D6 7E8E    218      MOV    'R6,#8EH    ;WRITE PROTECTION
"ACTIVE"
01D8 7F80    219      MOV    R7,#80H
01DA 1201DE  220      LCALL  BYTEWR
01DD 22      221      RET
222
223      ;***** BYTEWR SUB. *****
224      ; WRITE SINGLE BYTE TO STC
225      ; IN  = R6 COMMMAND
226      ;      = R7 DATA
227      ; REG = A,B,R6,R7
228
01DE C286    229      BYTEWR: CLR    IO    ; COMMAND BYTE
"WRITE"
01E0 120251  230      LCALL  DELAY
01E3 D285    231      SETB   RSTN    ; RST=1
01E5 120251  232      LCALL  DELAY
01E8 75F008  233      MOV    B,#8    ; SEND COMMAND
01EB C3      234      CLR    C
235

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 5

ptd08.asm

```

01EC EE      236  BYTEWR1: MOV    A,R6
01ED 13      237      RRC    A
01EE FE      238      MOV    R6,A
01EF 9286    239      MOV    IO,C
01F1 120246  240      LCALL  SCLKRW
01F4 D5F0F5  241      DJNZ   B,BYTEWR1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	242		
01F7 75F008	243	MOV B,#8	;SEND DATA BYTE
01FA C3	244	CLR C	
	245		
01FB EF	246	BYTEWR2: MOV A,R7	
01FC 13	247	RRC A	
01FD FF	248	MOV R7,A	
01FE 9286	249	MOV IO,C	
0200 120246	250	LCALL SCLKRW	
0203 D5F0F5	251	DJNZ B,BYTEWR2	
0206 C285	252	CLR RST\	
0208 120251	253	LCALL DELAY	
020B 22	254	RET	
	255		
	256	; BYTERD SUB.	
	257	;	
	258	;	
	259	;	
	260	;	
	261	;	
	262		
	263		
020C D286	264	BYTERD: SETB IO	;COMMAND BYTE "READ"
020E 120251	265	LCALL DELAY	
0211 D285	266	SETB RST\	
0213 120251	267	LCALL DELAY	
0216 75F008	268	MOV B,#8	;SEND COMMAND
0219 C3	269	CLR C	
	270		
021A EE	271	BYTERD1: MOV A,R6	
021B 13	272	RRC A	
021C FE	273	MOV R6,A	

```

021D 9286      274      MOV      IO,C
021F 12023B    275      LCALL   SCLKCOM
0222 D5F0F5    276      DJNZ     B,BYTERD1
                        277
0225 75F008    278      MOV      B,#8      ;RECIVE DATA BYTE
0228 7F00      279      MOV      R7,#0     ;CLEAR DATA BUFFER
                        280
022A 120246    281 BYTERD2:  LCALL   SCLKRW
022D EF        282      MOV      A,R7
022E A286      283      MOV      C,IO ;   READ SERIAL DATA
0230 13        284      RRC      A
0231 FF        285      MOV      R7,A
0232 D5F0F5    286      DJNZ     B,BYTERD2
0235 C285      287      CLR      RST\
0237 120251    288      LCALL   DELAY
023A 22        289      RET
                        290
291 ;***** SCLKCOM SUB. *****
292 ;
293 ;
294 ;

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 6

ptd08.asm

295 ;

296

```

023B C287      297 SCLKCOM:  CLR      SCLK
023D 120251    298      LCALL   DELAY
0240 D287      299      SETB     SCLK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0242 120251 300          LCALL  DELAY
0245 22      301          RET
                        302
                        303 ;***** SCLKRW
SUB.*****
                        304 ;
                        305 ;
                        306 ;
                        307
0246 D287    308 SCLKRW: SETB   SCLK
0248 120251  309          LCALL  DELAY
024B C287    310          CLR     SCLK
024D 120251  311          LCALL  DELAY
0250 22      312          RET
                        313
                        314 ;***** DELAY SUB.
*****
                        315 ;
                        316 ;
                        317
0251 7905    318 DELAY:  MOV    R1,#5H
0253 D9FE    319          DJNZ   R1,$
0255 22      320          RET
                        321
0256 7900    322 DELAY1: MOV    R1,#0
0258 7B00    323 DELAY2: MOV    R3,#0
025A DBFE    324          DJNZ   R3,$
025C D9FA    325          DJNZ   R1,DELAY2
025E DDF6    326          DJNZ   R5,DELAY1
0260 22      327          RET
                        328

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

329

330 ; GRAPH

331

332 ;

333 ;***** AXIS *****

0261 334 HAXIS:

0261 900B0D 335 MOV DPTR,#H_M_S ; STRING 'HH/MM/SS'

0264 120131 336 LCALL SEND_STRING

0267 900AE1 337 MOV DPTR,#GRAPH5 ; SRTING ' DATA '

026A 120131 338 LCALL SEND_STRING

026D 900A8E 339 MOV DPTR,#GRAPH1 ; STRING '++++++'

0270 120131 340 LCALL SEND_STRING

0273 22 341 RET

342

0274 343 VAXIS:

0274 C0E0 344 PUSH ACC

0276 C0D0 345 PUSH PSW

0278 75D008 346 MOV PSW,#00001000B ; REGISTER BANG 1

027B 1206C4 347 LCALL TDISP ; DISPLAY TIME

027E AD35 348 MOV R5,DATA_BUFFER ; R5 POINT ADDRESS

35H

0280 120181 349 LCALL GET_DATA

0283 1202A3 350 CALL CH_STR1 ; STRING 'CH0 CH1 '

0286 900AC5 351 MOV DPTR,#GRAPH2 ; STRING '-'

0289 120131 352 LCALL SEND_STRING

028C 1202B0 353 CALL PLOT_CH

```

028F 1202E0 354      CALL    CR_LF_SP
0292 1202BB 355      CALL    DATA1
0295 1202E0 356      CALL    CR_LF_SP
0298 90089E 357      MOV     DPTR,#TEXT1
029B 120131 358      LCALL   SEND_STRING
029E D0D0 359      POP     PSW
02A0 D0E0 360      POP     ACC
02A2 22 361          RET
362
363      ;***** AXIS . SUB
*****
364
02A3 900B17 365 CH_STR1: MOV     DPTR,#CHENNEL0
02A6 120131 366          LCALL   SEND_STRING
02A9 900B1C 367          MOV     DPTR,#CHENNEL1
02AC 120131 368          LCALL   SEND_STRING
02AF 22 369          RET
370
02B0 7800 371 PLOT_CH: MOV     R0,#00H      ; CHENNEL0
02B2 1202ED 372          CALL   PLOT_DATA ; KEEP DATA AND PLOT
DATA
02B5 7801 373          MOV     R0,#01H      ; CHENNEL1
02B7 1202FD 374          LCALL   MAIN_PLOT
02BA 22 375          RET
376
02BB E536 377 DATA1: MOV     A,36H      ; CHENNEL 0 (36H-38H)
02BD 12015A 378          LCALL   PRINT_STR
02C0 E537 379          MOV     A,37H
02C2 12015A 380          LCALL   PRINT_STR
02C5 E538 381          MOV     A,38H

```

```

02C7 12015A 382          LCALL  PRINT_STR
02CA 9009AF 383          MOV     DPTR,#TEXT5
02CD 120131 384          LCALL  SEND_STRING
02D0 E539   385          MOV     A,39H          ; CHENNEL 1 (39H-
3BH)
02D2 12015A 386          LCALL  PRINT_STR
02D5 E53A   387          MOV     A,3AH
02D7 12015A 388          LCALL  PRINT_STR
02DA E53B   389          MOV     A,3BH
02DC 12015A 390          LCALL  PRINT_STR
02DF 22     391          RET
                                392
02E0 90089E 393  CR_LF_SP: MOV     DPTR,#TEXT1
02E3 120131 394          LCALL  SEND_STRING
02E6 900AC7 395          MOV     DPTR,#GRAPH3
02E9 120131 396          LCALL  SEND_STRING
02EC 22     397          RET
                                398
02ED 1202FD 399  LOT_DATA: LCALL  MAIN_PLOT ; KEEP DATA AND PLOT
DATA
02F0 90089C 400          MOV     DPTR,#TEXT0          ; CARRIER
RETURN
02F3 120131 401          LCALL  SEND_STRING
02F6 900AEA 402          MOV     DPTR,#GRAPH6          ; SPACE 26
02F9 120131 403          LCALL  SEND_STRING
02FC 22     404          RET
                                405
                                406          ;
                                407          ;***** MAIN PLOT *****
                                408          ;
                                409
02FD          410  MAIN_PLOT:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

02FD 120388 411      LCALL  CHDISP
0300 BD000A 412      CJNE   R5,#00H,PL1      ; CHECK HIGH

```

CHENNEL R5

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 8

ptd08.asm

```

0303 7A01      413      MOV     R2,#01H
0305 7B00      414      MOV     R3,#00H
0307 1203DA    415      LCALL   SUB_R5
030A 020387    416      JMP      PLOT_STOP
030D BD010A    417 PL1:   CJNE   R5,#01H,PL2
0310 7A06      418      MOV     R2,#06H
0312 7B05      419      MOV     R3,#05H
0314 1203DA    420      LCALL   SUB_R5
0317 020387    421      JMP      PLOT_STOP
031A BD020A    422 PL2:   CJNE   R5,#02H,PL3
031D 7A0B      423      MOV     R2,#0BH
031F 7B0A      424      MOV     R3,#0AH
0321 1203DA    425      LCALL   SUB_R5
0324 020387    426      JMP      PLOT_STOP
0327 BD030A    427 PL3:   CJNE   R5,#03H,PL4
032A 7A10      428      MOV     R2,#10H
032C 7B0F      429      MOV     R3,#0FH
032E 1203DA    430      LCALL   SUB_R5
0331 020387    431      JMP      PLOT_STOP
0334 BD040A    432 PL4:   CJNE   R5,#04H,PL5
0337 7A15      433      MOV     R2,#15H
0339 7B14      434      MOV     R3,#14H
033B 1203DA    435      LCALL   SUB_R5

```

```

033E 020387 436 JMP PLOT_STOP
0341 BD050A 437 PL5: CJNE R5,#05H,PL6
0344 7A1A 438 MOV R2,#1AH
0346 7B19 439 MOV R3,#19H
0348 1203DA 440 LCALL SUB_R5
034B 020387 441 JMP PLOT_STOP
034E BD060A 442 PL6: CJNE R5,#06H,PL7
0351 7A1F 443 MOV R2,#1FH
0353 7B1E 444 MOV R3,#1EH
0355 1203DA 445 LCALL SUB_R5
0358 020387 446 JMP PLOT_STOP
035B BD070A 447 PL7: CJNE R5,#07H,PL8
035E 7A24 448 MOV R2,#24H
0360 7B23 449 MOV R3,#23H
0362 1203DA 450 LCALL SUB_R5
0365 020387 451 JMP PLOT_STOP
0368 BD080A 452 PL8: CJNE R5,#08H,PL9
036B 7A29 453 MOV R2,#29H
036D 7B28 454 MOV R3,#28H
036F 1203DA 455 LCALL SUB_R5
0372 020387 456 JMP PLOT_STOP
0375 BD090A 457 PL9: CJNE R5,#09H,PL0
0378 7A2E 458 MOV R2,#2EH
037A 7B2D 459 MOV R3,#2DH
037C 1203DA 460 LCALL SUB_R5
037F 020387 461 JMP PLOT_STOP
0382 7432 462 PL0: MOV A,#32H
0384 1203B5 463 LCALL PLDISP ;END CHECK HIGH CHENNEL

```

R5

```

0387 22 464 PLOT_STOP: RET
465
466 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

467 ;***** PLOT.SUB
*****
468 ;
469
0388 470 CHDISP:
0388 B80014 471 CJNE R0,#00H,CHDISP1

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 9

ptd08.asm

```

038B AD36 472 MOV R5,36H ;HIGH CHANNEL 0
038D ED 473 MOV A,R5
038E 9430 474 SUBB A,#48
0390 FD 475 MOV R5,A
0391 AE37 476 MOV R6,37H ;MEDIUM CHANNEL 0
0393 EE 477 MOV A,R6
0394 9430 478 SUBB A,#48
0396 FE 479 MOV R6,A
0397 AF38 480 MOV R7,38H ;LOW CHANNEL 0
0399 EF 481 MOV A,R7
039A 9430 482 SUBB A,#48
039C FF 483 MOV R7,A
039D 61B4 484 AJMP CH_END
039F B80112 485 CHDISP1: CJNE R0,#01H,CH_END
03A2 AD39 486 MOV R5,39H ;HIGH CHANNEL 1
03A4 ED 487 MOV A,R5
03A5 9430 488 SUBB A,#48
03A7 FD 489 MOV R5,A
03A8 AE3A 490 MOV R6,3AH ;MEDIUM CHANNEL 1
03AA EE 491 MOV A,R6

```

```

03AB 9430      492      SUBB    A,#48
03AD FE        493      MOV     R6,A
03AE AF3B      494      MOV     R7,3BH          ;LOW  CHANNEL 1
03B0 EF        495      MOV     A,R7
03B1 9430      496      SUBB    A,#48
03B3 FF        497      MOV     R7,A
03B4 22        498      CH_END:   RET
                                499      .
                                500
03B5           501      PLDISP:           ;DISPLAY SPACE, STAR

```

AND DATA

```

03B5 FC        502      MOV     R4,A
03B6 BC0002    503      CJNE    R4,#00H,SPACE1
03B9 8008      504      SJMP    STAR
03BB 9009AF    505 SPACE1:  MOV     DPTR,#TEXT5          ;DISPLAY SPACE
03BE 120131    506          LCALL   SEND_STRING
03C1 DCF8      507          DJNZ    R4,SPACE1
03C3 B80008    508 STAR:   CJNE    R0,#00,STAR0
03C6 900AFD    509          MOV     DPTR,#STAR_0          ;STAR 0
03C9 120131    510          LCALL   SEND_STRING
03CC 800B      511          SJMP    END_PLDISP
03CE B80108    512 STAR0:  CJNE    R0,#01,END_PLDISP
03D1 900AFF    513          MOV     DPTR,#STAR_1          ;STAR 1
03D4 120131    514          LCALL   SEND_STRING
03D7 8000      515          SJMP    END_PLDISP
03D9 22        516 END_PLDISP:  RET
                                517
                                518
                                519
03DA           520SUB_R5:           ;CHECK MEDIUM

```

CHENNEL R6

```

03DA BE0007    521      CJNE    R6,#00H,SUB00

```

```

03DD EB      522      MOV    A,R3
03DE 1203B5  523      LCALL  PLDISP
03E1 02043A  524      JMP     PASS01

                    525

03E4 BE0206  526 SUB00:  CJNE    R6,#02H,COMPARE01
03E7 EA      527      MOV    A,R2
03E8 1203B5  528      LCALL  PLDISP
03EB 813A    529      AJMP    PASS01
03ED 5006    530 COMPARE01: JNC     SUB01

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 10

ptd08.asm

```

03EF EB      531      MOV    A,R3
03F0 1203B5  532      LCALL  PLDISP
03F3 813A    533      AJMP    PASS01

                    534

03F5 BE0408  535 SUB01: CJNE    R6,#04H,COMPARE02
03F8 EA      536      MOV    A,R2
03F9 2401    537      ADD     A,#01H
03FB 1203B5  538      LCALL  PLDISP
03FE 813A    539      AJMP    PASS01
0400 5008    540 COMPARE02: JNC     SUB02
0402 EB      541      MOV    A,R3
0403 2401    542      ADD     A,#01H
0405 1203B5  543      LCALL  PLDISP
0408 813A    544      AJMP    PASS01

                    545

040A BE0608  546 SUB02: CJNE    R6,#06H,COMPARE03
040D EA      547      MOV    A,R2

```

040E 2402 548 ADD A,#02H
0410 1203B5 549 LCALL PLDISP
0413 813A 550 AJMP PASS01
0415 5008 551 COMPARE03: JNC SUB03
0417 EB 552 MOV A,R3
0418 2402 553 ADD A,#02H
041A 1203B5 554 LCALL PLDISP
041D 813A 555 AJMP PASS01

556

041F BE0808 557 SUB03: CJNE R6,#08H,COMPARE04

0422 EA 558 MOV A,R2

0423 2403 559 ADD A,#03H

0425 1203B5 560 LCALL PLDISP

0428 813A 561 AJMP PASS01

042A 5008 562 COMPARE04: JNC SUB04

042C EB 563 MOV A,R3

042D 2403 564 ADD A,#03H

042F 1203B5 565 LCALL PLDISP ;END CHECK MEDIU

CHENNEL R6

0432 813A 566 AJMP PASS01

567

0434 EA 568 SUB04: MOV A,R2

0435 2404 569 ADD A,#04H

0437 1203B5 570 LCALL PLDISP

571

043A 22 572 PASS01: RET

573

574

575

576

577 ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

578 ;***** CHECK TIME

579 ;

043B 580 RD_TIME:

043B 7E85 581 MOV R6,#85H ; READ HOUR

043D 12020C 582 LCALL BYTERD

0440 EF 583 MOV A,R7 ; UPPER HOUR

0441 C4 584 SWAP A

0442 540F 585 ANL A,#0FH

0444 F530 586 MOV NOW_H_H,A

0446 EF 587 MOV A,R7 ; LOWER HOUR

0447 540F 588 ANL A,#0FH

0449 F517 589 MOV NOW_H_L,A

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 11

ptd08.asm

044B 7E83 590 MOV R6,#83H ; READ MINUTE

044D 12020C 591 LCALL BYTERD

0450 EF 592 MOV A,R7 ; UPPER MINUTE

0451 C4 593 SWAP .A

0452 540F 594 ANL A,#0FH

0454 F533 595 MOV NOW_M_H,A ; KEEP HIGHER MINUTE IN

NOW_M_H

0456 EF 596 MOV A,R7 ; LOWER MINUTE

0457 540F 597 ANL A,#0FH

0459 F534 598 MOV NOW_M_L,A ; KEEP LOWER MINUTE IN

NOW_M_L

045B 7E81 599 MOV R6,#81H ; READ SECOND

045D 12020C 600 LCALL BYTERD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0460 EF      601      MOV      A,R7          ;  UPPER SECOND
0461 C4      602      SWAP     A
0462 540F    603      ANL      A,#0FH
0464 F51E    604      MOV      NOW_S_H,A    ;  KEEP UPPER SECOND AT
NOW_S_H
0466 EF      605      MOV      A,R7          ;  LOWER SECOND
0467 540F    606      ANL      A,#0FH
0469 F51F    607      MOV      NOW_S_L,A    ;  KEEP LOWER SECOND
AT NOW_S_L
046B 22      608      RET
                                609
                                610
046C          611      SCANKEY:
046C 20B619  612  SCAN0:  JB      P3.6,SCAN1
046F 751901  613      MOV      SCAN_LCD,#01H
0472 851918  614      MOV      GET_LCD,SCAN_LCD
0475 1205D5  615      CALL     MAIN_LCD
0478 751B00  616      MOV      GET_R4,#00H ;DISPLAY TIME EVERY N*10
MIN
047B 751A01  617      MOV      GET_R2,#01H ; DISPLAY TIME EVERY N
MIN
047E D2B6    618      SETB     P3.6
0480 D221    619      SETB     ST
0482 120564  620      LCALL     UBEEP
0485 0204BD  621      JMP      END_SCAN
0488          622      SCAN1:
0488 20B519  623      JB      P3.5,SCAN2
048B 751902  624      MOV      SCAN_LCD,#02H
048E 851918  625      MOV      GET_LCD,SCAN_LCD
0491 1205D5  626      CALL     MAIN_LCD
0494 751B01  627      MOV      GET_R4,#01H
0497 751A00  628      MOV      GET_R2,#00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

049A D2B5    629      SETB    P3.5
049C D221    630      SETB    ST
049E 120564  631      LCALL  UBEEP .
04A1 0204BD  632      JMP     END_SCAN
04A4          633      SCAN2:
04A4 20B216  634      JB      P3.2,END_SCAN
04A7 751903  635      MOV     SCAN_LCD,#03H
04AA 851918  636      MOV     GET_LCD,SCAN_LCD
04AD 1205D5  637      CALL   MAIN_LCD
04B0 751B03  638      MOV     GET_R4,#03H
04B3 751A00  639      MOV     GET_R2,#00H
04B6 D2B2    640      SETB    P3.2
04B8 D221    641      SETB    ST .
04BA 120564  642      LCALL  UBEEP
04BD 22      643END_SCAN:  RET
                644
                645
04BE 04      646ADD_TEN:  INC     A
04BF B40602  647      CJNE   A,#06H,TDEC_R4 ; IF PRINTTIME_H=60
THEN
    04C2 7400  648      MOV     A,#00H      ; PRINTTIME_H IS
EQUAL 00

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 12

ptd08.asm

```

04C4 DCF8    649TDEC_R4:  DJNZ   R4,ADD_TEN
04C6 22      650      RET
                651
                652

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

04C7 04      653 ADD_UNIT: INC  A

04C8 B40A02 654      CJNE  A,#0AH,DEC_R2 ; IF PRINTTIME_L=10
THEN
04CB 7400    655      MOV   A,#00H      ; PRINTTIME_L IS
EQUAL 00
04CD DAF8    656 DEC_R2:  DJNZ  R2,ADD_UNIT
04CF 22      657      RET
658
659
04D0 E532    660 COMP_M_S: MOV   A,PRN_M_L      ; COMP. =
COMPARE
04D2 8534F0 661      MOV   B,NOW_M_L      ; M = MINUTE
04D5 B5F04F 662      CJNE  A,B,PASSTIME ; L = LOWER
04D8 1204DC 663      CALL  COMP_SEC
04DB 22      664      RET
665
666
04DC E51C    667 COMP_SEC: MOV   A,PRN_S_H      ; S = SECOND
04DE 851EF0 668      MOV   B,NOW_S_H      ; H = HIGHER
04E1 B5F043 669      CJNE  A,B,PASSTIME
04E4 E51D    670      MOV   A,PRN_S_L
04E6 851FF0 671      MOV   B,NOW_S_L
04E9 B5F03B 672      CJNE  A,B,PASSTIME
04EC 1204F0 673      CALL  DISP_DATA
04EF 22      674      RET
675
676
04F0          677      DISP_DATA:
04F0 202016 678      JB     TYPE,GHDISP
04F3 120680 679      LCALL  D_M_Y      ;DISPLAY DATA
TYPE 1
04F6 C0E0    680      PUSH  ACC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

04F8 C0D0    681      PUSH    PSW
04FA 75D010  682      MOV     PSW,#00010000B
04FD 1206C4  683      LCALL   TDISP      ;
0500 D0D0    684      POP     PSW
0502 D0E0    685      POP     ACC
0504 120164  686      LCALL   SEND_DATA
0507 8003    687      SJMP    STOP_COMP
0509          688 GHDISP:
0509 120274  689      LCALL   VAXIS      ;DISPLAY DATA TYPE 2
050C 22      690 STOP_COMP:  RET
          691
          692
050D          693      CHK_TIME:
050D 913B    694      CALL   RD_TIME
050F 916C    695      CALL   SCANKEY
0511 AC1B    696      MOV     R4,GET_R4
0513 202111  697      JB      ST,PASSTIME
0516 BC000F  698      CJNE    R4,#00H,TEN_COL
0519 E532    699 UNIT_COL: MOV     A,PRN_M_L
051B AA1A    700      MOV     R2,GET_R2
051D 91C7    701      CALL   ADD_UNIT
051F 8534F0  702      MOV     B,NOW_M_L
0522 B5F002  703      CJNE    A,B,PASSTIME
0525 91DC    704      CALL   COMP_SEC
0527 22      705 PASSTIME:  RET
          706
0528 E531    707 TEN_COL:  MOV     A,PRN_M_H

```

```

052A AC1B 708      MOV    R4,GET_R4
052C 91BE 709      CALL   ADD_TEN
052E 8533F0 710     MOV    B,NOW_M_H
0531 B5F0F3 711     CJNE   A,B,PASSTIME
0534 AA1A 712      MOV    R2,GET_R2
0536 BA00E0 713     CJNE   R2,#00H,UNIT_COL
0539 91D0 714      CALL   COMP_M_S
053B 80EA 715      SJMP    PASSTIME

716
717 ;program sound speaker
718
00A0= 719 SNDCON EQU 0A0H
720
721
722 ; ***** SOUND SUB ***** 36
723 ; SOUND GENERATOR
724 ; IN = R2 FREQUENCY
725 ; = R3 LENGTH
726 ; REG = A,R2,R3,R4,R5,DPTR
727

053D 7D00 728      SOUND: MOV    R5,#0 ;END FLAG
053F 7CA0 729      MOV    R4,#SNDCON ;DELAY

```

CONSTANT

```

0541 120548 730      SOUND1: LCALL SOUNDS
0544 BD01FA 731      CJNE   R5,#1,SOUND1
0547 22 732          RET
733
0548 734      SOUNDS:
0548 D2B4 735      SETB   P3.4 ;OUT 1 (P3.4)
054A 120553 736      LCALL SOUNDX
054D C2B4 737      CLR    P3.4 ;OUT 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

054F 120553    738          LCALL SOUNDX
0552 22        739          RET
                                740
0553 EA        741    SOUNDX:  MOV    A,R2                ;FREQUENCY
DELAY
0554 12055B    742    SOUNDX1: LCALL SOUNDX
0557 14        743          DEC    A
0558 70FA      744          JNZ    SOUNDX1
055A 22        745          RET
                                746
055B DC06      747    SOUNDX: DJNZ  R4,SOUNDX1          ;LENGTH COUNT
DOWN
055D 7CA0      748          MOV    R4,#SNDCON
055F DB02      749          DJNZ  R3,SOUNDX1
0561 7D01      750          MOV    R5,#1
0563 22        751    SOUNDX1: RET
                                752
                                753    ; ***** nBEEP SUB *****
                                754    ; ANY BEEP SUBROUTINE
                                755    ; REG = A,R2,R3,R4,R5,DPTR
                                756
                                757
0564 7A0D      758    UBEEP:  MOV    R2,#0DH            ;USER BEEP (37)
0566 7B60      759          MOV    R3,#60H
0568 12053D    760          LCALL SOUND
056B 22        761          RET
                                762
                                763    ;
                                764    ;***** LCD . SUB
*****
                                765    ;

```

766 ;***** DELAY SUB

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 14

ptd08.asm

767 ;DELAY SUBOUTINE

768 ;IN = R2

769 ;REG = R2,R3,R4

770

056C 7B00 771 DELAY00: MOV R3,#0

056E 7C00 772 DELAY01: MOV R4,#0

0570 DCFE 773 DJNZ R4,\$

0572 DBFA 774 DJNZ R3,DELAY01

0574 DAF6 775 DJNZ R2,DELAY00

0576 22 776 RET

777

778 ;***** LCDDIS

779 ;LOAD DATA TO LCD_MODULE

780 ;IN = DPTR START BLOCK (40 BYTE)

781 ;REG = A,R2,DPTR

782

0577 7480 783 LCDLD: MOV A,#80H ;SET ADDRESS LINE

1

0579 120582 784 LCALL LCDLDS

057C 74C0 785 MOV A,#0C0H ;SET ADDRESS

LINE 2

057E 120582 786 LCALL LCDLDS

0581 22 787 RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

788

0582 C083 789 LCDLDS: PUSH DPH ;LOAD SUB

0584 C082 790 PUSH DPL

0586 1205A0 791 LCALL LCDWI ;WRITE ADDRESS

0589 D082 792 POP DPL

058B D083 793 POP DPH

794

058D 7A08 795 MOV R2,#8 ;8 CHAR.

058F E4 796 LCDLDS1: CLR A ;MOVC FOR

CODE /MOV FOR DATA

0590 93 797 MOVC A,@A+DPTR

0591 C083 798 PUSH DPH

0593 C082 799 PUSH DPL

0595 1205B3 800 LCALL LCDWD ;WRITE DATA

0598 D082 801 POP DPL

059A D083 802 POP DPH

059C A3 803 INC DPTR

059D DAF0 804 DJNZ R2,LCDLDS1

059F 22 805 RET

806

807 ;***** LCD *****

808 ; LCD WRITE INSTRUCTION (RS = 0))

809 ; IN = A

810 ; REG = A,DPTR

811

05A0 F5A0 812 LCDWI: MOV P2,A ;DATA TO PORT 2

813 ;CAUTOIN NO YET CORRECT

05A2 C282 814 CLR RS

05A4 C281 815 CLR RW

05A6 C280 816 CLR EN

05A8 D280 817 SETB EN

05AA 00 818 NOP

```

05AB C280      819          CLR    EN
05AD 7400      820          MOV    A,#0          ;DELAY
05AF 14        821 LCDWI1: DEC    A
05B0 70FD      822          JNZ    LCDWI1
05B2 22        823          RET
                        824          ;***** LCDWD
*****
                        825          ; LCD WRITE DATA (RS=1)

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 15

ptd08.asm

```

                        826          ;IN  = A
                        827          ;REG = A,DPTR
                        828
                        829
05B3 F5A0      830 LCDWD: MOV    P2,A
05B5 D282      831          SETB   RS
05B7 C281      832          CLR    RW
05B9 C280      833          CLR    EN
05BB D280      834          SETB   EN
05BD C280      835          CLR    EN
05BF 7400      836          MOV    A,#0
                        837
05C1 14        838 LCDWD1: DEC    A
05C2 70FD      839          JNZ    LCDWD1
05C4 22        840          RET
                        841
                        842

```

```

843 ;***** INITIALIZE LCD DISPLAY
*****
844 ;
845 ;
846
05C5 847 INT_LCD:
848 ; MOV R2,#20H
849 ; LCALL DELAY00
05C5 7438 850 MOV A,#00111000B ;FUNCTION SET
05C7 1205A0 851 LCALL LCDWI
05CA 740E 852 MOV A,#00001110B ;DISPLAY ON/OFF
05CC 1205A0 853 LCALL LCDWI
05CF 7401 854 MOV A,#01H ;CLEAR
05D1 1205A0 855 LCALL LCDWI
05D4 22 856 RET
857
858 ;
859 ;***** MAIN LCD
*****
860 ;
05D5 861 MAIN_LCD:
05D5 C0E0 862 PUSH ACC
05D7 C0D0 863 PUSH PSW
05D9 75D010 864 MOV PSW,#00010000B ; REGISTER

```

BANG 2

```

05DC AE19 865 MOV R6,SCAN_LCD
05DE 7902 866 MOV R1,#02H
05E0 1205C5 867 LOOP_LCD: LCALL INT_LCD
05E3 BE0006 868 CJNE R6,#00H,SCAN_LCD0
05E6 900620 869 MOV DPTR,#MAIN00
05E9 020616 870 JMP SCAN_LCD5
05EC BE0106 871 SCAN_LCD0: CJNE R6,#01H,SCAN_LCD1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

05EF 900630 872          MOV  DPTR,#MAIN01
05F2 020616 873          JMP   SCAN_LCD5
05F5 BE0206 874  SCAN_LCD1: CJNE  R6,#02H,SCAN_LCD2
05F8 900640 875          MOV  DPTR,#MAIN02
05FB 020616 876          JMP   SCAN_LCD5
05FE BE0306 877  SCAN_LCD2: CJNE  R6,#03H,SCAN_LCD3
0601 900650 878          MOV  DPTR,#MAIN03
0604 020616 879          JMP   SCAN_LCD5
0607 BE0406 880  SCAN_LCD3: CJNE  R6,#04H,SCAN_LCD4
060A 900660 881          MOV  DPTR,#MAIN04
060D 020616 882          JMP   SCAN_LCD5
0610 BE0503 883  SCAN_LCD4: CJNE  R6,#05H,SCAN_LCD5
0613 900670 884          MOV  DPTR,#MAIN05

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 16

ptd08.asm

```

0616 120577 885  SCAN_LCD5:  LCALL  LCDLD
0619 D9C5    886          DJNZ   R1,LOOP_LCD
061B D0D0    887          POP    PSW
061D D0E0    888          POP    ACC
061F 22      889          RET
                        890

```

```

0620 20205052 891 MAIN00:  DB   ' PRN A/D CONV. '

```

```

0624 4E20412F

```

```

0628 4420434F

```

```

062C 4E562E20

```

```

0630 20202020 892          MAIN01: DB   ' 1 MINUTE '

```

```

0634 3120204D

```

```

0638 494E5554

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

063C 45202020

0640 20202031 893 MAIN02: DB ' 10 MINUTES '

0644 3020204D

0648 494E5554

064C 45532020

0650 20202033 894MAIN03: DB ' 30 MINUTES '

0654 3020204D

0658 494E5554

065C 45532020

0660 20204F46 895MAIN04: DB ' OFF GRAPHS '

0664 46202047

0668 52415048

066C 53202020

0670 2020204F 896MAIN05: DB ' ON GRAPHS '

0674 4E202047

0678 52415048

067C 53202020

897

898 ;***** TIME DISPLAY SUB

899 ;AT TX RS 232

900

0680 9009B5 901 D_M_Y: MOV DPTR,#TEXT7

0683 120131 902 LCALL SEND_STRING

0686 7E87 903 MOV R6,#87H ; READ

DATE

0688 12020C 904 LCALL BYTERD

068B 1206B0 905 CALL DISP

068E 9009AD 906 MOV DPTR,#TEXT4

0691 120131 907 LCALL SEND_STRING

0694 7E89 908 MOV R6,#89H ; READ

MONTH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0696 12020C 909          LCALL BYTERD
0699 1206B0 910          CALL  DISP
069C 9009AD 911          MOV   DPTR,#TEXT4
069F 120131 912          LCALL SEND_STRING
06A2 7E8D   913          MOV   R6,#8DH          ; READ YEAR
06A4 12020C 914          LCALL BYTERD
06A7 1206B0 915          CALL  DISP
06AA 9009AF 916          MOV   DPTR,#TEXT5
06AD 3131   917          CALL  SEND_STRING
06AF 22     918          RET
                        919
06B0 EF     920DISP: MOV   A,R7          ; UPPER
06B1 C4     921          SWAP  A
06B2 540F   922          ANL   A,#0FH
06B4 12013D 923          LCALL  HTOA
06B7 12015A 924          LCALL  PRINT_STR
06BA EF     925          MOV   . A,R7          ; LOWER

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 17

ptd08.asm

```

06BB 540F   926          ANL   A,#0FH
06BD 12013D 927          LCALL  HTOA
06C0 12015A 928          LCALL  PRINT_STR
06C3 22     929          RET   .
                        930
06C4        931          TDISP:
06C4 7E85   932          MOV   R6,#85H          ; READ HOUR
06C6 12020C 933          LCALL  BYTERD
06C9 EF     934          MOV   A,R7          ; UPPER HOUR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

06CA C4	935	SWAP	A	
06CB 540F	936	ANL	A,#0FH	
06CD 12013D	937	LCALL	HTOA	
06D0 12015A	938	LCALL	PRINT_STR	
06D3 EF	939	MOV	A,R7	; LOWER HOUR
06D4 540F	940	ANL	A,#0FH	
06D6 12013D	941	LCALL	HTOA	
06D9 12015A	942	LCALL	PRINT_STR	
06DC 90089A	943	MOV	DPTR,#TEXT	
06DF 120131	944	LCALL	SEND_STRING	
06E2	945	MIN:		
06E2 7E83	946	MOV	R6,#83H	; READ MINUTE
06E4 12020C	947	LCALL	BYTERD	
06E7 EF	948	MOV	A,R7	; UPPER MINUTE
06E8 C4	949	SWAP	A	
06E9 540F	950	ANL	A,#0FH	
06EB F531	951	MOV PRN_M_H,A ; STROGE HIGHMINUTE AT		
PRINTTIME				
06ED 12013D	952	LCALL	HTOA	
06F0 12015A	953	LCALL	PRINT_STR	
06F3 EF	954	MOV	A,R7	; LOWER MINUTE
06F4 540F	955	ANL	A,#0FH	
06F6 F532	956	MOV	PRN_M_L,A	; STROGE
LOWERMINUTE AT PRINTTIME				
06F8 12013D	957	LCALL	HTOA	
06FB 12015A	958	LCALL	PRINT_STR	
06FE 90089A	959	MOV	DPTR,#TEXT	
0701 120131	960	LCALL	SEND_STRING	
	961			
0704 7E81	962	MOV	R6,#81H	; READ SECOND
0706 12020C	963	LCALL	BYTERD	
0709 EF	964	MOV	A,R7	; UPPER SECOND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

070A C4      965      SWAP  A
070B 540F    966      ANL   A,#0FH
070D F51C    967      MOV   PRN_S_H,A  ; STROGE UPPERSECOND AT

```

PRINTTIME

```

070F 12013D  968      LCALL HTOA
0712 12015A  969      LCALL PRINT_STR
0715 EF      970      MOV   A,R7          ; LOWER SECOND
0716 540F    971      ANL   A,#0FH
0718 F51D    972      MOV   PRN_S_L,A  ; STROGE LOWERSECOND AT

```

PRINTTIME

```

071A 12013D  973      LCALL HTOA
071D 12015A  974      LCALL PRINT_STR
0720 9009AF  975      MOV   DPTR,#TEXT5
0723 3131    976      CALL  SEND_STRING
0725 22      977      RET

```

978

979

980

981

982

983 ; MAIN PROGRAM

984

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 18

ptd08.asm

985

986

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

987

```

0726      988      MAIN:
0726 5151    989      CALL  DELAY
0728 758160  990      MOV   SP,#60H   ; STACK AREA STRAT $60-7F
072B 3100    991      CALL  INT_RS232
072D D283    992      SETB   STROBE
                993      ;   CLR   RST\
                994      ;   SETB  SCLK
                995      ;   LCALL SETTIME
072F B164    996      CALL  UBEEP
0731 751900  997      MOV   SCAN_LCD,#00H   ; " PRN A/D CONV. "
0734 B1D5    998      CALL  MAIN_LCD
0736 7A05    999      MOV   R2,#5
0738 B16C   1000     CALL  DELAY00
073A 020742 1001     JMP   M1
073D 20B509 1002     MODE1: JB    P3.5,MODE2   ; P3.5 = 0 = OFF

```

GRAPHS

```

0740 B164    1003     CALL  UBEEP
0742 751904  1004 M1:   MOV   SCAN_LCD,#04H   ; " OFF GRAPHS "
0745 B1D5    1005     CALL  MAIN_LCD
0747 C223    1006     CLR   MODE
0749 20B609  1007     MODE2: JB    P3.6,END_MODE   ; P3.6 = 0 = ON

```

GRAPHS

```

074C B164    1008     CALL  UBEEP
074E 751905  1009     MOV   SCAN_LCD,#05H   ; " ON GRAPHS "
0751 B1D5    1010     CALL  MAIN_LCD
0753 D223    1011     SETB  MODE
0755 20B7E5  1012     END_MODE: JB    P3.7,MODE1   ; P3.7 = 0 =

```

ENTER,RUN

```

0758 B164    1013     CALL  UBEEP
075A 7A05    1014     MOV   R2,#5
075C B16C    1015     CALL  DELAY00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

075E 202341 1016          JB      MODE,MAIN_RTC2
                                1017
0761          1018      MAIN_RTC1:
0761 020769 1019          JMP      T01
0764 20B60D 1020 TIME01:  JB      P3.6,TIME02
0767 B164    1021          CALL    UBEEP
0769 751901 1022 T01:     MOV     SCAN_LCD,#01H      ; " 1 MINUTE "
076C B1D5    1023          CALL    MAIN_LCD
076E 751B00 1024          MOV     GET_R4,#00H
0771 751A01 1025          MOV     GET_R2,#01H
0774 20B50D 1026 TIME02:  JB      P3.5,TIME03
0777 B164    1027          CALL    UBEEP
0779 751902 1028          MOV     SCAN_LCD,#02H      ; " 10 MINUTES "
077C B1D5    1029          CALL    MAIN_LCD
077E 751B01 1030          MOV     GET_R4,#01H
0781 751A00 1031          MOV     GET_R2,#00H
0784 20B20D 1032 TIME03:  JB      P3.2,END_TIME
0787 B164    1033          CALL    UBEEP
0789 751903 1034          MOV     SCAN_LCD,#03H      ; " 30 MINUTES "
078C B1D5    1035          CALL    MAIN_LCD
078E 751B03 1036          MOV     GET_R4,#03H
0791 751A00 1037          MOV     GET_R2,#00H
0794 20B7CD 1038 END_TIME: JB      P3.7,TIME01
0797 B164    1039          CALL    UBEEP
0799 7A05    1040          MOV     R2,#5
079B B16C    1041          CALL    DELAY00
079D D222    1042          SETB    CONTROL_MAIN
079F 0207FE 1043          JMP     RTC01

```

ptd08.asm

```

                                1044
07A2          1045      MAIN_RTC2:
07A2 0207AA 1046                JMP      T04
07A5 20B60D 1047      TIME04: JB      P3.6,TIME05
07A8 B164     1048                CALL    UBEEP
07AA 751901 1049      T04:      MOV     SCAN_LCD,#01H      ; " 1 MINUTE
"
07AD B1D5     1050                CALL    MAIN_LCD
07AF 751B00 1051                MOV     GET_R4,#00H
07B2 751A01 1052                MOV     GET_R2,#01H
07B5 20B50D 1053      TIME05: JB      P3.5,TIME06
07B8 B164     1054                CALL    UBEEP
07BA 751902 1055                MOV     SCAN_LCD,#02H      ; " 10
MINUTES "
07BD B1D5     1056                CALL    MAIN_LCD
07BF 751B01 1057                MOV     GET_R4,#01H
07C2 751A00 1058                MOV     GET_R2,#00H
07C5 20B20D 1059TIME06: JB      P3.2,END_TIME1
07C8 B164     1060                CALL    UBEEP
07CA 751903 1061                MOV     SCAN_LCD,#03H      ; " 30
MINUTES "
07CD B1D5     1062                CALL    MAIN_LCD
07CF 751B03 1063                MOV     GET_R4,#03H
07D2 751A00 1064                MOV     GET_R2,#00H
07D5 20B7CD 1065      END_TIME1: JB      P3.7,TIME04
07D8 B164     1066                CALL    UBEEP
07DA 7A05     1067                MOV     R2,#5
07DC B16C     1068                CALL    DELAY00
07DE C222     1069                CLR     CONTROL_MAIN
07E0 020859 1070                JMP     RTC02
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1071
07E3      1072      MAIN_RTC01:
07E3 B164  1073          CALL  UBEEP
07E5 751900 1074          MOV   SCAN_LCD,#00H          ; " PRN
A/D CONV. "
07E8 B1D5  1075          CALL  MAIN_LCD
07EA 7A05  1076          MOV   R2,#5
07EC B16C  1077          CALL  DELAY00
07EE 751904 1078          MOV   SCAN_LCD,#04H          ; " OFF
GRAPHS "
07F1 B1D5  1079          CALL  MAIN_LCD
07F3 7A05  1080          MOV   R2,#5
07F5 B16C  1081          CALL  DELAY00
07F7 851819 1082          MOV   SCAN_LCD,GET_LCD      ; " XX
MINUTES "
07FA B1D5  1083          CALL  MAIN_LCD
07FC D222  1084          SETB   CONTROL_MAIN
07FE      1085      RTC01:
07FE 9008A6 1086          MOV   DPTR,#TEXT3          ; TOPIC
0801 120131 1087          CALL  SEND_STRING
0804 90094F 1088          MOV   DPTR,#TEXT8
0807 120131 1089          CALL  SEND_STRING
080A C220  1090          CLR    TYPE          ;
CHOOSE MAIN1
080C C221  1091          CLR    ST
080E 120680 1092          LCALL  D_M_Y
0811 1206C4 1093          LCALL  TDISP
0814 120164 1094          LCALL  SEND_DATA
0817      1095      MAIN1:
0817 12050D 1096          LCALL  CHK_TIME
081A E530  1097          MOV   A,NOW_H_H
081C B4020C 1098          CJNE   A,#2,IN_DAY1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

081F F528      1099      MOV      D,A
0821 8528F0     1100      MOV      B,D
0824 95F0       1101      SUBB     A,B
0826 B40202     1102      CJNE     A,#2,IN_DAY1

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 20

ptd08.asm

```

0829 80B8      1103      JMP      MAIN_RTC01
082B           1104      IN_DAY1:
082B 2021B5     1105      JB       ST,MAIN_RTC01
082E 30B702     1106      JNB      P3.7,CHANGE_MAIN
0831 80E4       1107      SJMP     MAIN1
                1108
0833 202202     1109      CHANGE_MAIN: JB      CONTROL_MAIN,MAIN_RTC02
0836 80AB       1110      JMP      MAIN_RTC01
                1111
0838           1112      MAIN_RTC02:
0838 120564     1113      CALL     UBEEP
083B 751900     1114      MOV      SCAN_LCD,#00H      ; " PRN A/D
CONV. "
083E 1205D5     1115      CALL     MAIN_LCD
0841 7A05       1116      MOV      R2,#5
0843 12056C     1117      CALL     DELAY00
0846 751905     1118      MOV      SCAN_LCD,#05H      ; " ON
GRAPHS "
0849 1205D5     1119      CALL     MAIN_LCD
084C 7A05       1120      MOV      R2,#5
084E 12056C     1121      CALL     DELAY00

```

```

0851 851819 1122          MOV     SCAN_LCD,GET_LCD    ; " XX
MINUTES "
0854 1205D5 1123          CALL    MAIN_LCD
0857 C222    1124          CLR      CONTROL_MAIN
0859          1125          RTC02:
0859 9008A6 1126          MOV     DPTR,#TEXT3          ; TOPIC
085C 120131 1127          CALL    SEND_STRING
085F 900AD1 1128          MOV     DPTR,#GRAPH4
0862 120131 1129          LCALL    SEND_STRING
0865 D220    1130          SETB     TYPE                ; CHOOSE MAIN2
0867 C221    1131          CLR      ST
0869 120680 1132          LCALL    D_M_Y
086C 9008A1 1133          MOV     DPTR,#TEXT2
086F 120131 1134          LCALL    SEND_STRING
0872 9009B8 1135          MOV     DPTR,#GRAPH0
0875 120131 1136          LCALL    SEND_STRING
0878 120261 1137          LCALL    HAXIS
087B 120274 1138          LCALL    VAXIS
087E          1139          MAIN2:
087E 12050D 1140          LCALL    CHK_TIME
0881 E530    1141          MOV     A,NOW_H_H
0883 B4020C 1142          CJNE     A,#2,IN_DAY2
0886 F528    1143          MOV     D,A
0888 8528F0 1144          MOV     B,D
088B 95F0    1145          SUBB     A,B
088D B4029B 1146          CJNE     A,#2,IN_DAY1
0890 80A6    1147          JMP      MAIN_RTC02
0892          1148          IN_DAY2:
0892 2021A3 1149          JB       ST,MAIN_RTC02
0895 30B79B 1150          JNB      P3.7,CHANGE_MAIN
0898 80E4    1151          SJMP     MAIN2

```

1152

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

089A 3A 1153 TEXT: DB ':'
089B 10 1154 DB EOS
089C 0D10 1155 TEXT0: DB CR,EOS
089E 0D0A 1156 TEXT1: DB CR,LF
08A0 10 1157 DB EOS
08A1 0D0A0D0A 1158 TEXT2: DB CR,LF,CR,LF
08A5 10 1159 DB EOS
1160
08A6 0D0A 1161 TEXT3: DB CR,LF

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 21

ptd08.asm

08A8 20205052 1162 DB ' PRINTING A/D CONVERTER
,CR,LF
08AC 494E5449
08B0 4E472020
08B4 412F4420
08B8 20434F4E
08BC 56455254
08C0 45522020
08C4 20200D0A
08C8 0D0A 1163 DB CR,LF
08CA 20202020 1164 DB ' PROGRAMMED BY
,CR,LF
08CE 20202050
08D2 524F4752
08D6 414D4D45
08DA 44204259
08DE 20202020

08E2 20202020

08E6 200D0A

08E9 0D0A 1165

DB CR,LF

08EB 20204D52 1166

DB ' MR. ANANTACHAI NAIJIT

','CR,LF

08EF 2E20414E

08F3 414E5441

08F7 43484149

08FB 204E4149

08FF 4A495420

0903 20202020

0907 200D0A

090A 20202020 1167

DB ' AND

','CR,LF

090E 20202020

0912 2020414E

0916 44202020

091A 20202020

091E 20202020

0922 20202020

0926 0D0A

0928 20204D52 1168

DB ' MR. SANCHAYOT SAKULJITTACHAROEN

','CR,LF

092C 2E205341

0930 4E434841

0934 594F5420

0938 53414B55

093C 4C4A4954

0940 54414348

0944 41524F45

0948 4E200D0A

094C 0D0A10 1169

DB CR,LF,EOS

094F 20202020 1171 TEXT8: DB ' DATA

,CR,LF

0953 20202020

0957 20202020

095B 20202020

095F 20202020

0963 20202020

0967 20202020

096B 20202020

096F 20444154

0973 41200D0A

0977 20204444 1172 DB ' DD/MM/YY HH/MM/SS CH0 CH1 CH2 CH3 CH4 CH5 CH6
CH7'

097B 2F4D4D2F

097F 59592048

0983 482F4D4D

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 22

ptd08.asm

0987 2F535320

098B 43483020

098F 43483120

0993 43483220

0997 43483320

099B 43483420

099F 43483520

09A3 43483620

09A7 434837

09AA 0D0A 1173 DB CR,LF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0A5C 30303030

0A60 30303030

0A64 30303030

0A68 30303030

0A6C 30303030

0A70 30303030

0A74 30303030

0A78 30303030

0A7C 30303030

0A80 30303030

0A84 30303030

0A88 3030450D

0A8C 0A

0A8D 10

1191

DB EOS

0A8E 202B2B2B

1192

GRAPH1:

DB

++++++',CR,LF

0A92 2B2B2B2B

0A96 2B2B2B2B

0A9A 2B2B2B2B

0A9E 2B2B2B2B

0AA2 2B2B2B2B

0AA6 2B2B2B2B

0AAA 2B2B2B2B

0AAE 2B2B2B2B

0AB2 2B2B2B2B

0AB6 2B2B2B2B

0ABA 2B2B2B2B

0ABE 2B2B2B2B

0AC2 0D0A

0AC4 10

1193

DB EOS

0AC5 2D

1194

GRAPH2:

DB

'

0AC6 10

1195

DB EOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0AC7 20202020 1196 GRAPH3: DB ' ,EOS
0ACB 20202020
0ACF 2010
0AD1 20204444 1197 GRAPH4: DB ' DD/MM/YY :'
0AD5 2F4D4D2F
0AD9 59592020
0ADD 203A20
0AE0 10 1198 DB EOS
0AE1 44415441 1199 GRAPH5: DB 'DATA,mV ',EOS
0AE5 2C6D5620
0AE9 10
0AEA 20202020 1200 GRAPH6: DB ' ,'
0AEE 20202020
0AF2 20202020
0AF6 20202020
0AFA 2020
0AFC 10 1201 DB EOS
0AFD 3010 1202 STAR_0: DB '0',EOS
0AFF 3110 1203 STAR_1: DB '1',EOS
0B01 3210 1204 STAR_2: DB '2',EOS
0B03 3310 1205 STAR_3: DB '3',EOS
0B05 3410 1206 STAR_4: DB '4',EOS

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 24

ptd08.asm

0B07 3510 1207 STAR_5: DB '5',EOS
0B09 3610 1208 STAR_6: DB '6',EOS
0B0B 3710 1209 STAR_7: DB '7',EOS
0B0D 48482F4D 1210 H_M_S: DB 'HH/MM/SS ',EOS

```

0B11 4D2F5353
0B15 2010
0B17 43483020 1211      CHENNEL0:      DB  'CH0 ',EOS
0B1B 10
0B1C 43483120 1212      CHENNEL1:      DB  'CH1 ',EOS
0B20 10
0B21 43483220 1213      CHENNEL2:      DB  'CH2 ',EOS
0B25 10
0B26 43483320 1214      CHENNEL3:      DB  'CH3 ',EOS
0B2A 10
0B2B 43483420 1215      CHENNEL4:      DB  'CH4 ',EOS
0B2F 10
0B30 43483520 1216      CHENNEL5:      DB  'CH5 ',EOS
0B34 10
0B35 43483620 1217      CHENNEL6:      DB  'CH6 ',EOS
0B39 10
0B3A 43483720 1218      CHENNEL7:      DB  'CH7 ',EOS
0B3E 10
1219
0000=      1220      END

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 1

ptd08.asm

```

add_ten = 04BE      646  649  709
add_unit = 04C7      653  656  701
busy = 0084         24   136
byterd = 020C       264  582  591  600  904  909  914  933  947  963
byterd1 = 021A      271  276
byterd2 = 022A      281  286

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bytewr = 01DE	229	192	196	200	204	208	212	216	220
bytewr1 = 01EC	236	241							
bytewr2 = 01FB	246	251							
ch_end = 03B4	498	484	485						
ch_str1 = 02A3	365	350							
change_main = 0833	1109	1106	1150						
chdisp = 0388	470	411							
chdisp1 = 039F	485	471							
channel0 = 0B17	1211	365							
channel1 = 0B1C	1212	367							
channel2 = 0B21	1213								
channel3 = 0B26	1214								
channel4 = 0B2B	1215								
channel5 = 0B30	1216								
channel6 = 0B35	1217								
channel7 = 0B3A	1218								
chk_time = 050D	693	1096	1140						
comp_m_s = 04D0	660	714							
comp_sec = 04DC	667	663	704						
compare01 = 03ED	530	526							
compare02 = 0400	540	535							
compare03 = 0415	551	546							
compare04 = 042A	562	557							
control_main = 0022	48	1042	1069	1084	1109	1124			
cr = 000D	33	1155	1156	1158	1161	1162	1163	1164	1165
	1167	1168	1169	1171	1173	1180	1186	1188	1190
	1192								
cr_lf_sp = 02E0	393	354	356						
d = 0028	50	1099	1100	1143	1144				
d_m_y = 0680	901	679	1092	1132					
data1 = 02BB	377	355							
data_buffer = 0035	57	146	348						
dec_r2 = 04CD	656	654							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

delay = 0251 318 230 232 253 265 267 288 298 300 309 311

989

delay00 = 056C 771 775 1000 1015 1041 1068 1077 1081 1117 1121

delay01 = 056E 772 774

delay1 = 0256 322 326

delay2 = 0258 323 325

disp = 06B0 920 905 910 915

disp_data = 04F0 677 673

en = 0080 29 816 817 819 833 834 835

end_mode = 0755 1012 1007

end_pldisp = 03D9 516 511 512 515

end_scan = 04BD 643 621 632 634

end_time = 0794 1038 1032

end_time1 = 07D5 1065 1059

eos = 0010 35 97 1154 1155 1157 1159 1169 1174 1177 1179

1181 1183 1191 1193 1195 1196 1198 1199 1201 1202 1203 1204 1205 1206 1207

1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218

eot = 0004 32

get = 018C 170 168

get_data = 0181 165 147 172 349

get_lcd = 0018 38 614 625 636 1082 1122

get_r2 = 001A 40 617 628 639 700 712 1025 1031 1037 1052 1058

1064

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 2

ptd08.asm

get_r4 = 001B 41 616 627 638 696 708 1024 1030 1036 1051 1057

1063

ghdisp = 0509 688 678

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

graph0 = 09B8    1185 1135
graph1 = 0A8E    1192 339
graph2 = 0AC5    1194 351
graph3 = 0AC7    1196 395
graph4 = 0AD1    1197 1128
graph5 = 0AE1    1199 337
graph6 = 0AEA    1200 402
h_m_s = 0B0D    1210 335
haxis = 0261    334 1137
htoa = 013D    111 923 927 937 941 952 957 968 973
htoas = 014B    120 113 116
htoas1 = 0155    126 122
in_day1 = 082B    1104 1098 1102 1146
in_day2 = 0892    1148 1142
int = 0112    72
int_lcd = 05C5    847 867
int_rs232 = 0100    63 991
io = 0086    20 229 239 249 264 274 283
lcdld = 0577    783 885
lcdlds = 0582    789 784 786
lcdlds1 = 058F    796 804
lcdwd = 05B3    830 800
lcdwd1 = 05C1    838 839
lcdwi = 05A0    812 791 851 853 855
lcdwi1 = 05AF    821 822
lf = 000A    34 1156 1158 1161 1162 1163 1164 1165 1166 1167
1168 1169 1171 1173 1180 1186 1188 1190 1192
loop_lcd = 05E0    867 886
m1 = 0742    1004 1001
main = 0726    988 19
main00 = 0620    891 869
main01 = 0630    892 872

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

main02 = 0640      893  875
main03 = 0650      894  878
main04 = 0660      895  881
main05 = 0670      896  884
main1 = 0817      1095 1107
main2 = 087E      1139 1151
main_lcd = 05D5    861  615  626  637  998 1005 1010 1023 1029 1035 1050
1056 1062 1075 1079 1083 1115 1119 1123
main_plot = 02FD    410  374  399
ain_rtc01 = 07E3    1072 1103 1105 1110
main_rtc02 = 0838    1112 1109 1147 1149
main_rtc1 = 0761    1018
main_rtc2 = 07A2    1045 1016
min = 06E2      945
mode = 0023      49 1006 1011 1016
mode1 = 073D    1002 1012
mode2 = 0749     1007 1002
now_h_h = 0030     51  586 1097 1141
now_h_l = 0017     37  589
now_m_h = 0033     54  595  710
now_m_l = 0034     55  598  661  702
now_s_h = 001E     44  604  668
now_s_l = 001F     45  607  671
pass01 = 043A     572  524  529  533  539  544  550  555  561  566
passtime = 0527    705  662  669  672  697  703  711  715

```

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

Page 3

ptd08.asm

```
pl0 = 0382      462  457
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pl1 = 030D 417 412
 pl2 = 031A 422 417
 pl3 = 0327 427 422
 pl4 = 0334 432 427
 pl5 = 0341 437 432
 pl6 = 034E 442 437
 pl7 = 035B 447 442
 pl8 = 0368 452 447
 pl9 = 0375 457 452
 pldisp = 03B5 501 463 523 528 532 538 543 549 554 560 565 570
 plot_ch = 02B0 371 353
 plot_data = 02ED 399 372
 plot_stop = 0387 464 416 421 426 431 436 441 446 451 456 461
 print_str = 015A 134 102 136 178 378 380 382 386 388 390 924
 928 938 942 953 958 969 974
 prn_m_h = 0031 52 707 951
 prn_m_l = 0032 53 660 699 956
 prn_s_h = 001C 42 667 967
 prn_s_l = 001D 43 670 972
 prt_data = 0192 175 149 183
 rd_time = 043B 580 694
 recive = 0129 85 167
 rs = 0082 27 814 831
 rst = 0085 22 231 252 266 287
 rtc01 = 07FE 1085 1043
 rtc02 = 0859 1125 1070
 rw = 0081 28 815 832
 scan0 = 046C 612
 scan1 = 0488 622 612
 scan2 = 04A4 633 623
 scan_lcd = 0019 39 613 614 624 625 635 636 865 997 1004 1009
 1022 1028 1034 1049 1055 1061 1074 1078 1082 1114 1118 1122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scan_lcd0 = 05EC    871  868
scan_lcd1 = 05F5    874  871
scan_lcd2 = 05FE    877  874
scan_lcd3 = 0607    880  877
scan_lcd4 = 0610    883  880
scan_lcd5 = 0616    885  870  873  876  879  882  883
scankey = 046C    611  695
sclk = 0087    21  297  299  308  310
sclkcom = 023B    297  275
sclkrw = 0246    308  240  250  281
sd = 0194    177  180
send = 0121    79
send_data = 0164    145  686  1094
send_str1 = 0137    101  97
send_string = 0131    95  104  151  182  336  338  340  352  358  366  368
384  394  396  401  403  506  510  514  902  907  912  917  944  960  976  1087
1089  1127  1129  1134  1136
send_tx = 0177    155  166
settime = 01A5    189
sndcon = 00A0    719  729  748
soh = 0001    31
sound = 053D    728  760
sound1 = 0541    730  731
sounds = 0548    734  730
soundx = 0553    741  736  738
soundx1 = 0554    742  744
soundy = 055B    747  742

```


text4 = 09AD	1176	906	911
text5 = 09AF	1178	181	383 505 ³ 916 975
text6 = 09B1	1180		
text7 = 09B5	1182	901	
text8 = 094F	1171	1088	
time01 = 0764	1020	1038	
time02 = 0774	1026	1020	
time03 = 0784	1032	1026	
time04 = 07A5	1047	1065	
time05 = 07B5	1053	1047	
time06 = 07C5	1059	1053	
tx1 = 00B1	56	152	156 160
type = 0020	46	678	1090 1130
ubeep = 0564	758	620	631 642 996 1003 1008 1013 1021 1027 1033
1039 1048 1054 1060	1066	1073	1113
unit_col = 0519	699	713	
vaxis = 0274	343	689	1138



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

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

Description

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

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

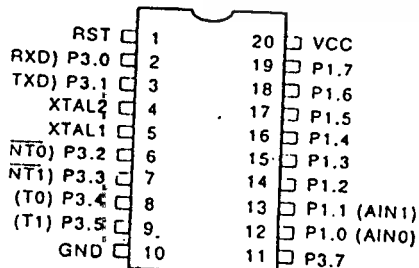
ATMEL

8-Bit Microcontroller with 2 Kbytes Flash

AT89C2051

Pin Configuration

PDIP/SOIC

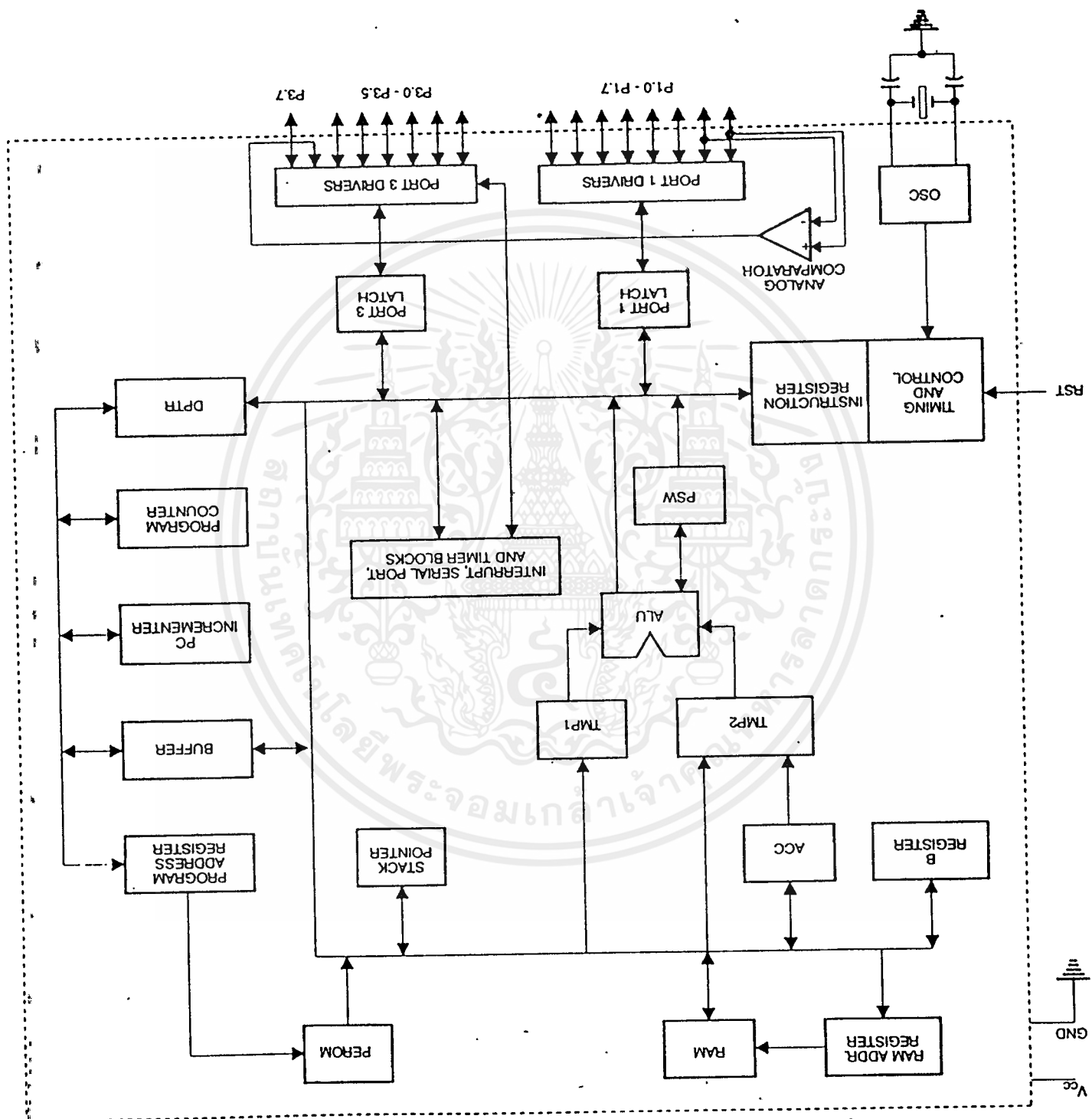


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า

หรือใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาต

ATMEL

อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Amel ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Vcc

Supply voltage.

GND

Ground.

Port 1

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

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

Port 3

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

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

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

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

RST

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

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

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

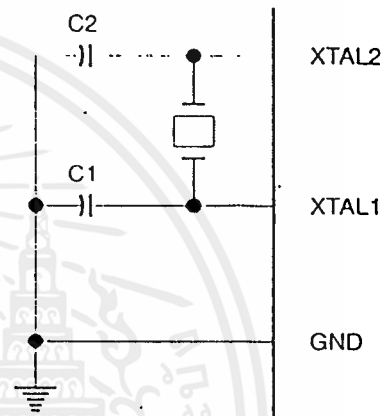
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

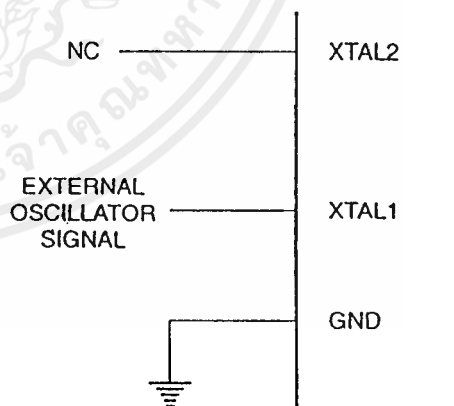
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF $\pm$ 10 pF for Crystals
= 40 pF $\pm$ 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Special Function Registers

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

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

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

Table 1. AT89C2051 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP X0000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 00000000							0AFH
0A0H								0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H		SP 00000111	DPL 00000000	DPH 00000000			PCON 00000000	87H

Restrictions on Certain Instructions

The AT89C2051 and AT89C1051 are economical and cost-effective members of Atmel's growing family of microcontrollers. They contain 2 Kbytes and 1 Kbyte of flash program memory, respectively. Both are fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program these devices.

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

Branching instructions:

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

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the AT89C2051). Violating the physical space limits may cause unknown program behavior.

JNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above ap-

plies. Again, violating the memory boundaries may cause erratic execution.

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

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory, while the AT89C1051 has 64 bytes. Thus, in the AT89C1051 the stack depth is limited to 64 bytes, the amount of available RAM. External DATA memory access is not supported in either device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

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

Program Memory Lock Bits

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

Lock Bit Protection Modes⁽¹⁾

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

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

Idle Mode

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

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

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

Power Down Mode

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

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

Programming The Flash

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

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

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

1. Power-up sequence:
Apply power between VCC and GND pins
Set RST and XTAL1 to GND
With all other pins floating, wait for greater than 10 milliseconds
 2. Set pin RST to 'H'
Set pin P3.2 to 'H'
 3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
- To Program and Verify the Array:
4. Apply data for Code byte at location 000H to P1.0 to P1.7
 5. Raise RST to 12V to enable programming.
 6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
 7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropriate levels. Output data can be read at the port P1 pins.
 8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
 9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2 Kbytes array or until the end of the object file is reached.
 10. Power-off sequence:
set XTAL1 to 'L'
set RST to 'L'
Float all other I/O pins
Turn Vcc power off

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

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

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

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

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

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

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

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

Programming Interface

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

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

Flash Programming Modes

Mode	RST	P3.2/ PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data ^(1,3)	12V		L	H	H	H
Read Code Data ⁽¹⁾	H	H	L	L	H	H
Write Lock Bit - 1	12V		H	H	H	H
Bit - 2	12V		H	H	L	L
Chip Erase	12V	 ⁽²⁾	H	L	L	L
Read Signature Byte	H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.

2. Chip Erase requires a 10 ms PROG pulse.

3. P3.1 is pulled Low during programming to indicate RDY/BSY.

Figure 3. Programming the Flash Memory

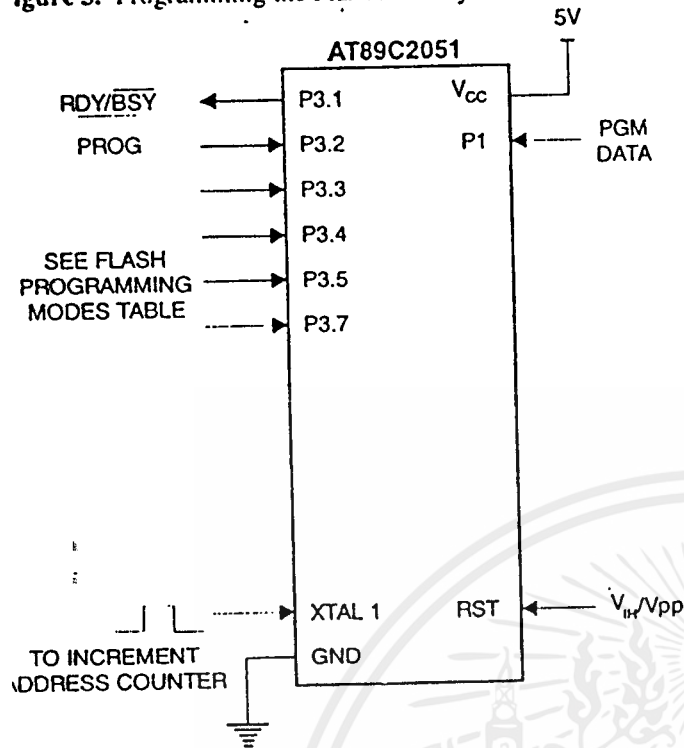
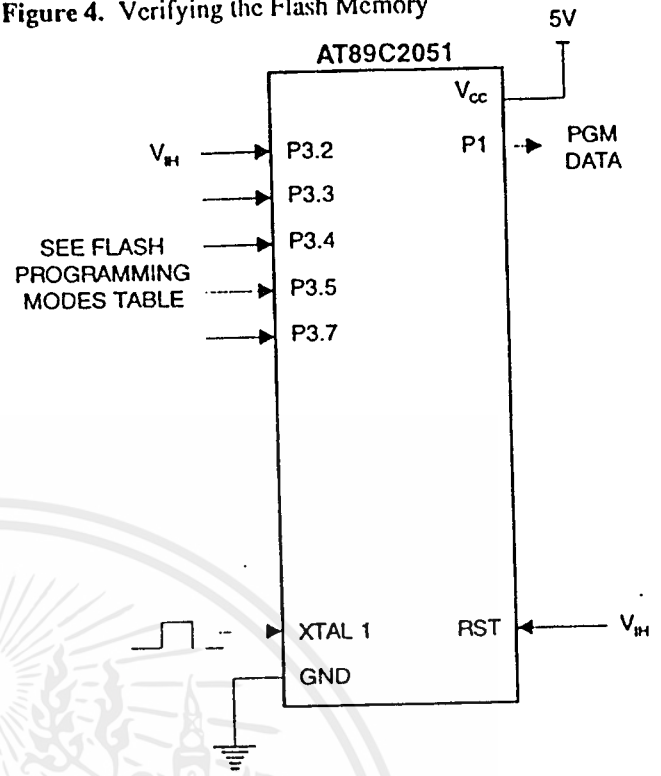


Figure 4. Verifying the Flash Memory



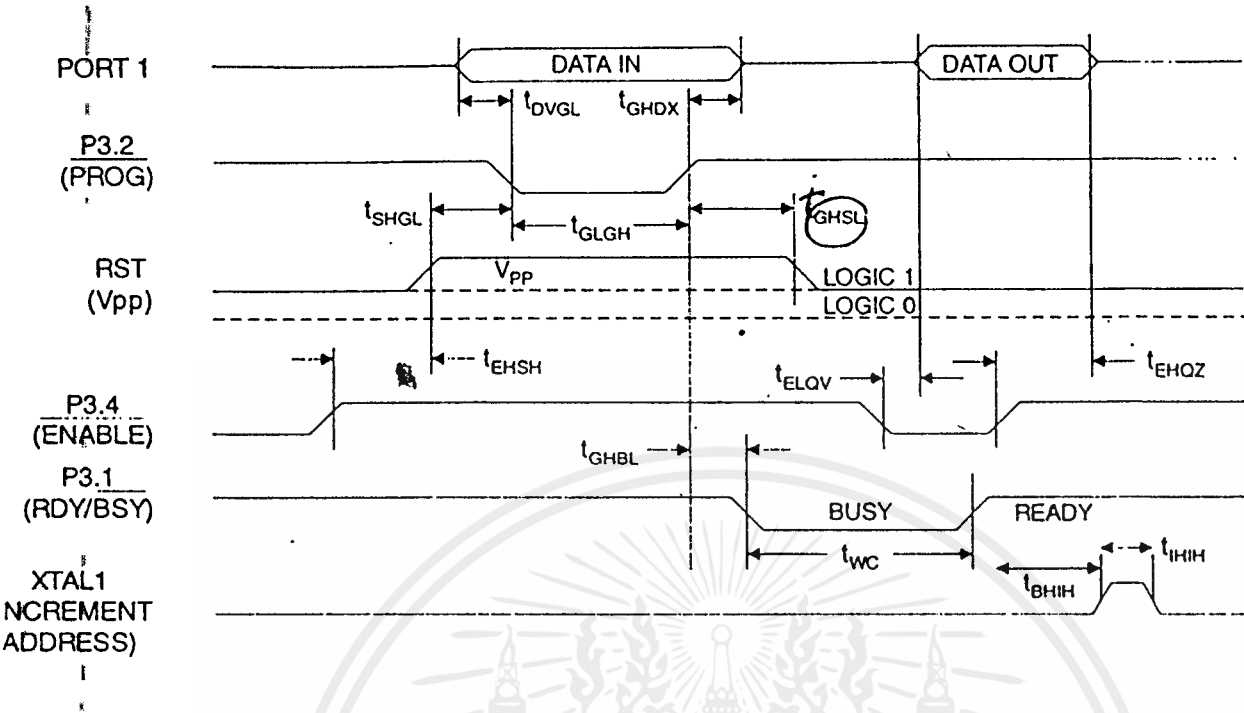
Flash Programming and Verification Characteristics

$T_A = 21^\circ\text{C to } 27^\circ\text{C}$, $V_{CC} = 5.0 \pm 10\%$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ash Programming and Verification Waveforms



bsolute Maximum Ratings*

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

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

D.C. Characteristics

$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7\text{ V}$ to 6.0 V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage		-0.5	$0.2 V_{CC}-0.1$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC}+0.9$	$V_{CC}+0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC}+0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1, 3)	$I_{OL} = 20\text{ mA}$, $V_{CC} = 5\text{ V}$ $I_{OL} = 10\text{ mA}$, $V_{CC} = 2.7\text{ V}$		0.5	V
V_{OH}	Output High Voltage (Ports 1, 3)	$I_{OH} = -80\text{ }\mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -30\text{ }\mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12\text{ }\mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	$V_{IN} = 0.45\text{ V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	$V_{IN} = 2\text{ V}$		-750	μA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		± 10	μA
V_{OS}	Comparator Input Offset Voltage	$V_{CC} = 5\text{ V}$		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{ID}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^{\circ}\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, $V_{CC} = 6\text{ V}/3\text{ V}$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6\text{ V}/3\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{ V}$ P1.0 & P1.1 = 0V or V_{CC} $V_{CC} = 3\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		100 20	μA μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 20 mA
Maximum total I_{OL} for all output pins: 80 mA

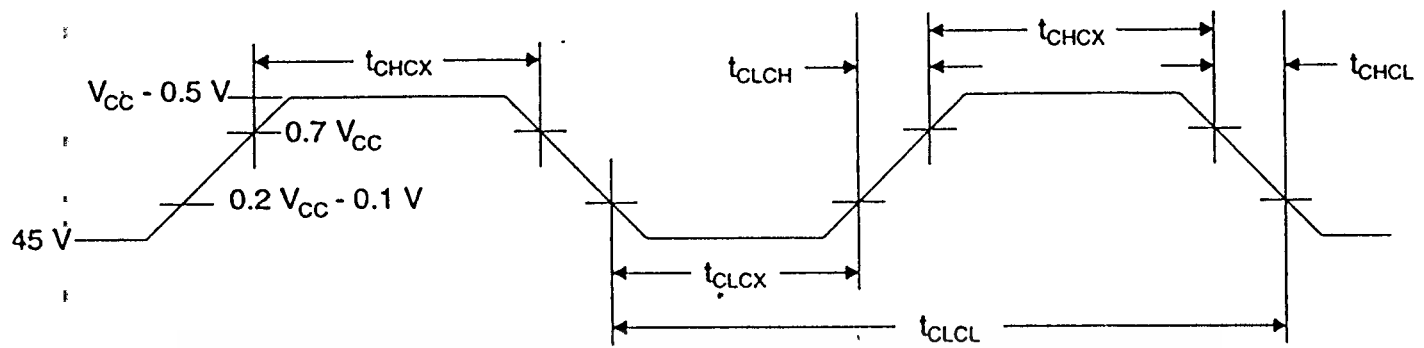
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power Down is 2 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Clock Drive Waveforms



External Clock Drive

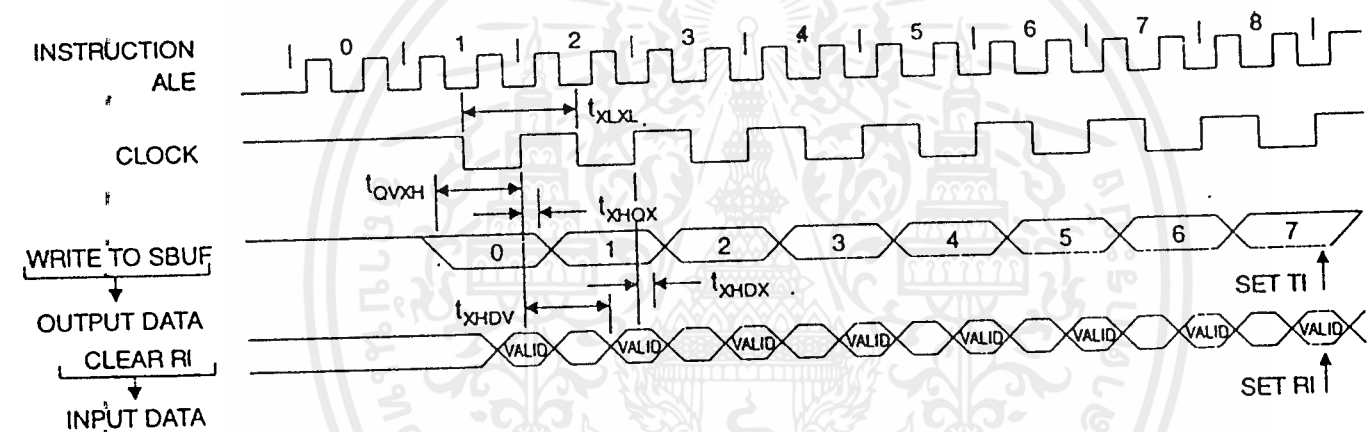
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

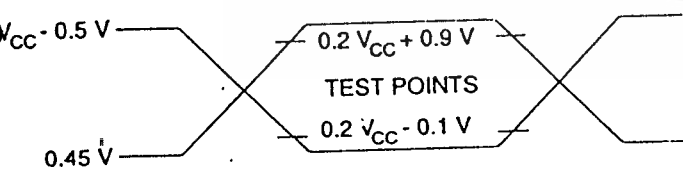
$V_{CC} = 5.0 V \pm 20\%$; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{CLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$ -133		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$ -33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$ -133	ns

Shift Register Mode Timing Waveforms

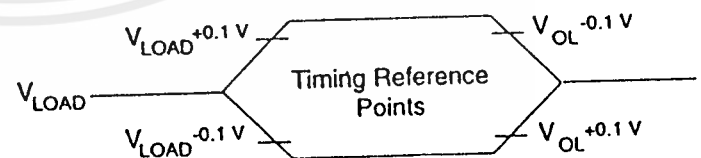


AC Testing Input/Output Waveforms⁽¹⁾



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

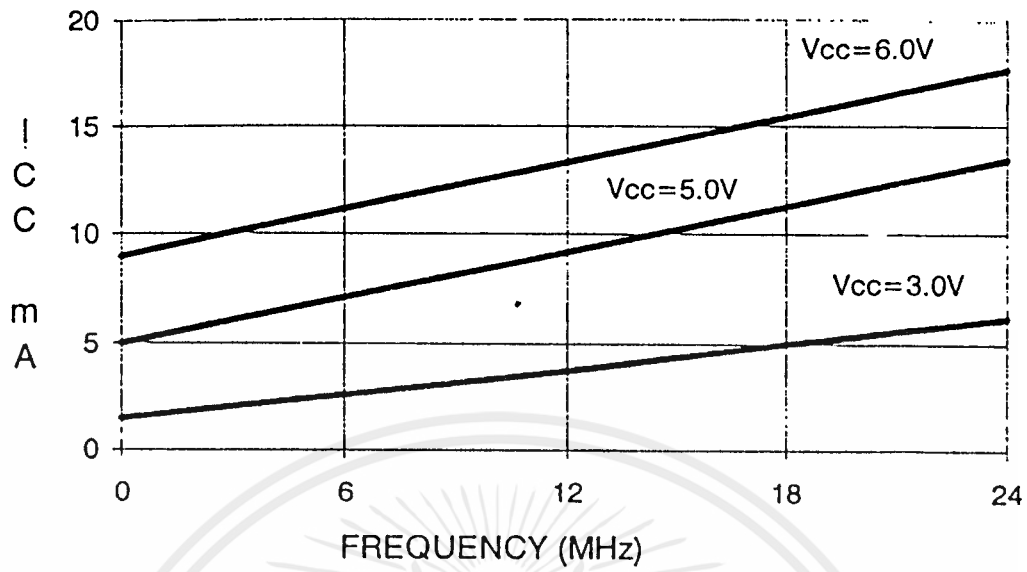
Float Waveforms⁽¹⁾



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

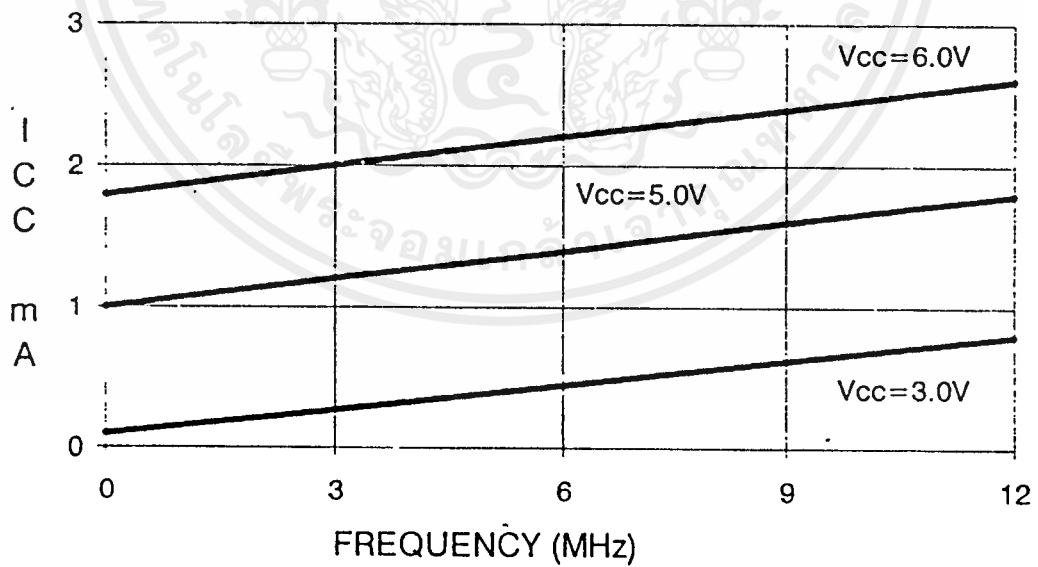
AT89C2051

TYPICAL ICC - ACTIVE (85°C)



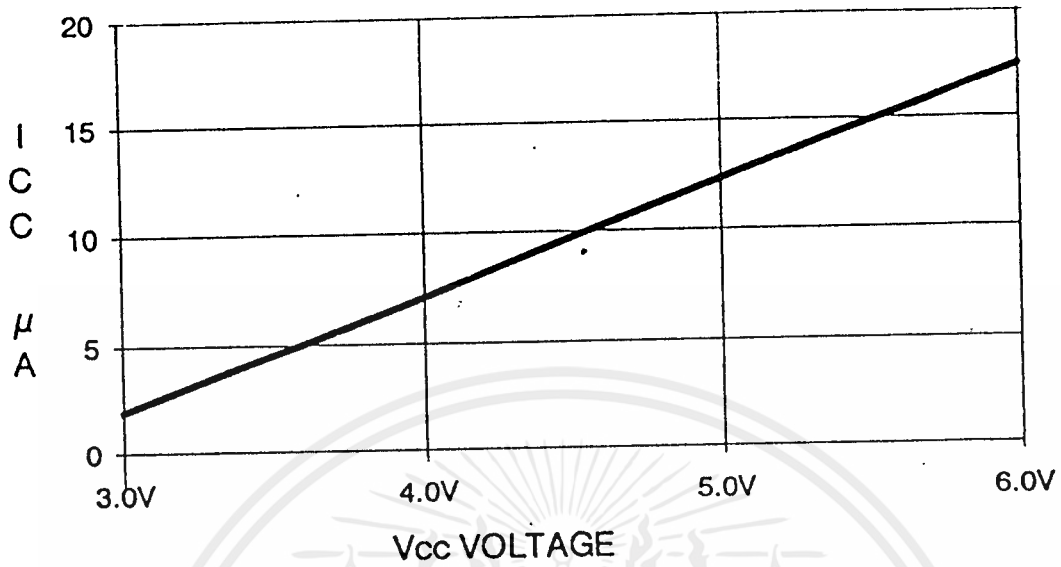
AT89C2051

TYPICAL ICC - IDLE (85°C)



AT89C2051

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



- ic:
1. XTAL1 tied to GND for ICC (power down).
 2. P1.0 and P1.1 = Vcc or GND.
 3. Lock bits programmed.

ering Information

eed Hz)	Power. Supply	Ordering Code	Package	Operation Range
2	2.7 V to 6.0 V	AT89C2051-12PC AT89C2051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-12PI AT89C2051-12SI	20P3 20S	Industrial (-40°C to 85°C)
4	4.0 V to 6.0 V	AT89C2051-24PC AT89C2051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-24PI AT89C2051-24SI	20P3 20S	Industrial (-40°C to 85°C)



Package Type

3	20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
Endurance: 1,000 Write/Erase Cycles
Data Retention: 10 Years
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Five Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

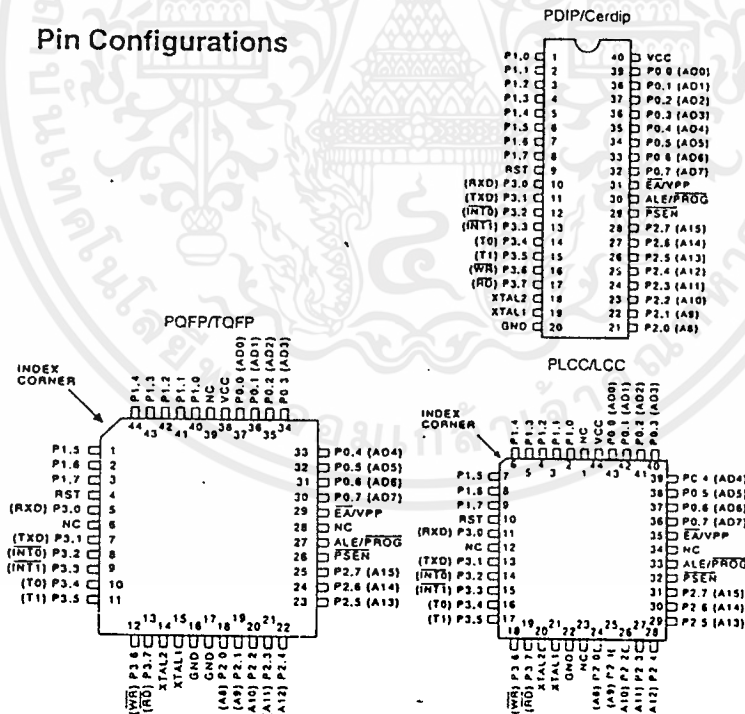
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer, with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five source two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is

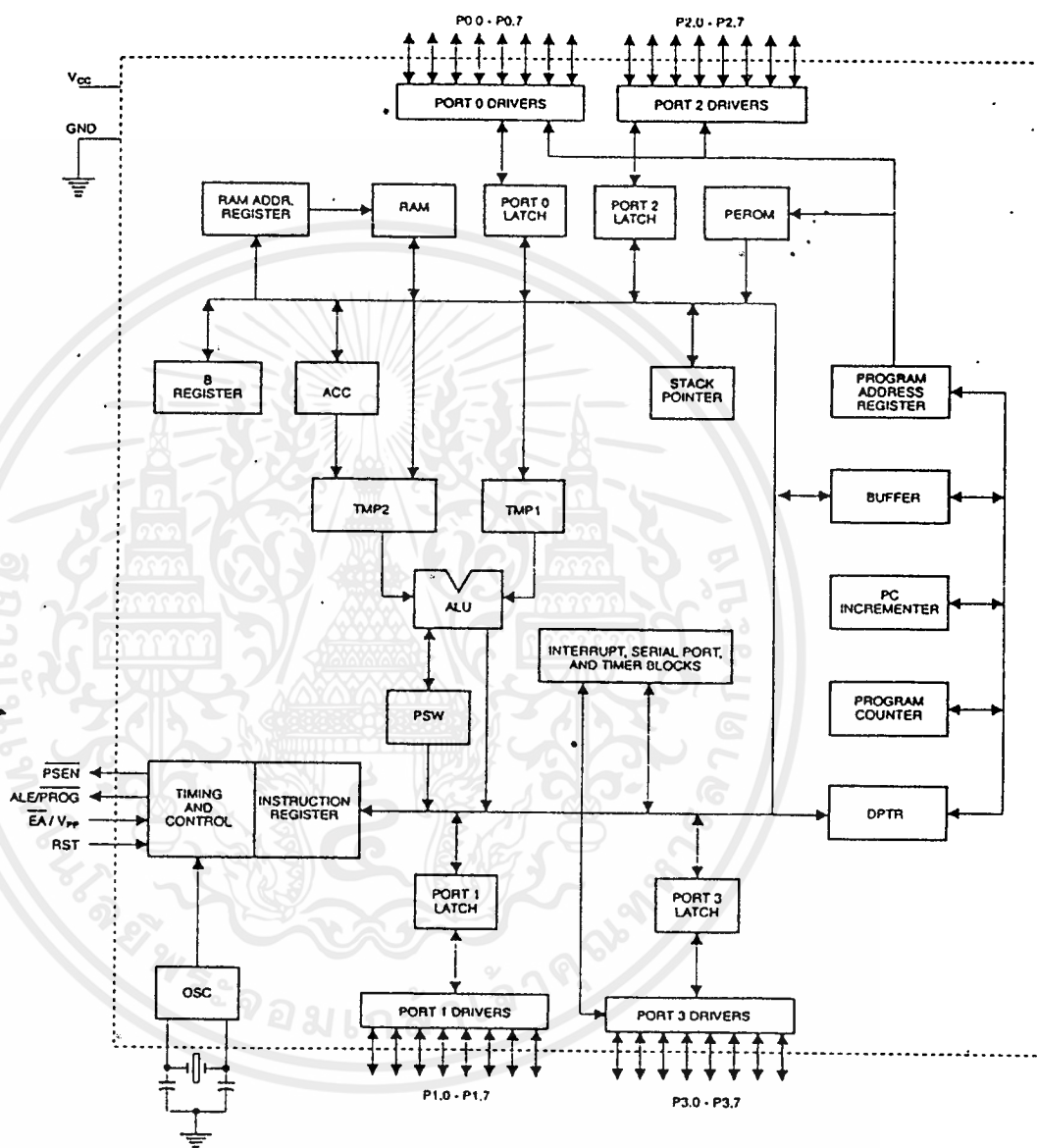
continued

Pin Configurations



8-Bit
Microcontroller
with 4 Kbytes
Flash

AT89C51



Description (Continued)

designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

Vcc

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s

are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

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

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/Vpp

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset.

EA should be strapped to VCC for internal program executions.

This pin also receives the 12-volt programming enable voltage (Vpp) during Flash programming, for parts that require 12-volt Vpp.

continued





Pin Description (Continued)

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

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

Idle Mode

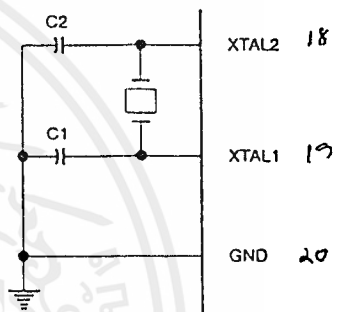
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

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

Power Down Mode

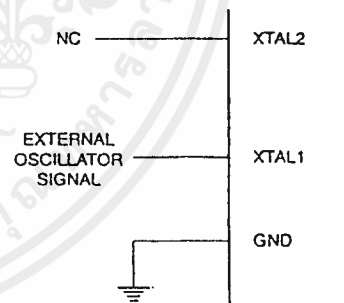
In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines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

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF $\pm$ 10 pF for Crystals
= 40 pF $\pm$ 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

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

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up

without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (VCC) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	Vpp = 12 V	Vpp = 5 V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.

4. Raise $\overline{EA}/V_{pp}$ to 12 V for the high-voltage programming mode.
5. Pulse ALE/ $\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/ $\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,





031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 51H indicates 89C51
- (032H) = FFH indicates 12 V programming
- (032H) = 05H indicates 5 V programming

Programming Interface

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

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

Flash Programming Modes

Mode	RST	PSEN	ALE/ PROG	EA/ V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V ⁽¹⁾	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock Bit - 1	H	L		H/12V	H	H	H	H
Bit - 2	H	L		H/12V	H	H	L	L
Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase	H	L	⁽²⁾	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

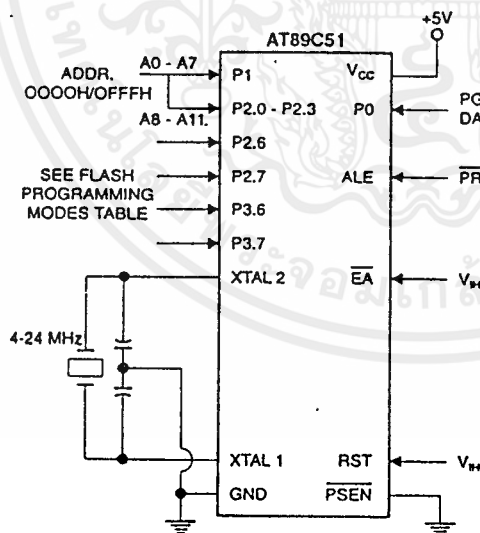
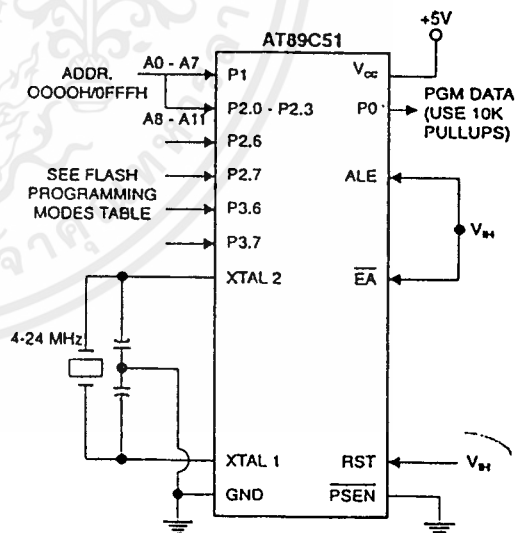


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

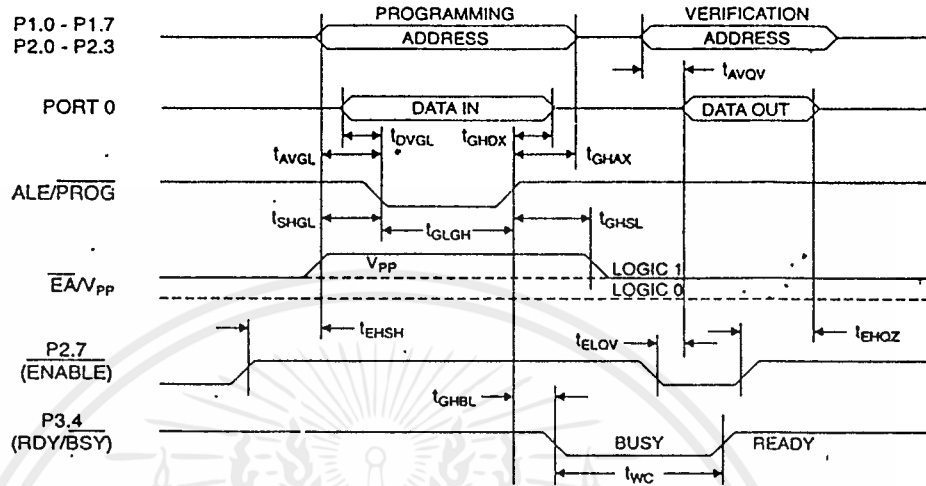
T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V _{PP} ⁽¹⁾	Programming Enable Voltage	11.5	12.5	V
I _{PP} ⁽¹⁾	Programming Enable Current*		1.0	mA
1/f _{CLCL}	Oscillator Frequency	4	24	MHz
t _{AVGL}	Address Setup to PROG Low	48t _{CLCL}		
t _{GHAX}	Address Hold After PROG	48t _{CLCL}		
t _{OVGL}	Data Setup to PROG Low	48t _{CLCL}		
t _{GHDX}	Data Hold After PROG	48t _{CLCL}		
t _{EHSH}	P2.7 (ENABLE) High to V _{PP}	48t _{CLCL}		
t _{SHGL}	V _{PP} Setup to PROG Low	10		μs
t _{GHSL} ⁽¹⁾	V _{PP} Hold After PROG	10		μs
t _{GLGH}	PROG Width	1	110	μs
t _{AVOV}	Address to Data Valid		48t _{CLCL}	
t _{ELOV}	ENABLE Low to Data Valid		48t _{CLCL}	
t _{EHQV}	Data Float After ENABLE	0	48t _{CLCL}	
t _{GHBL}	PROG High to BUSY Low		1.0	μs
t _{WC}	Byte Write Cycle Time		2.0	ms

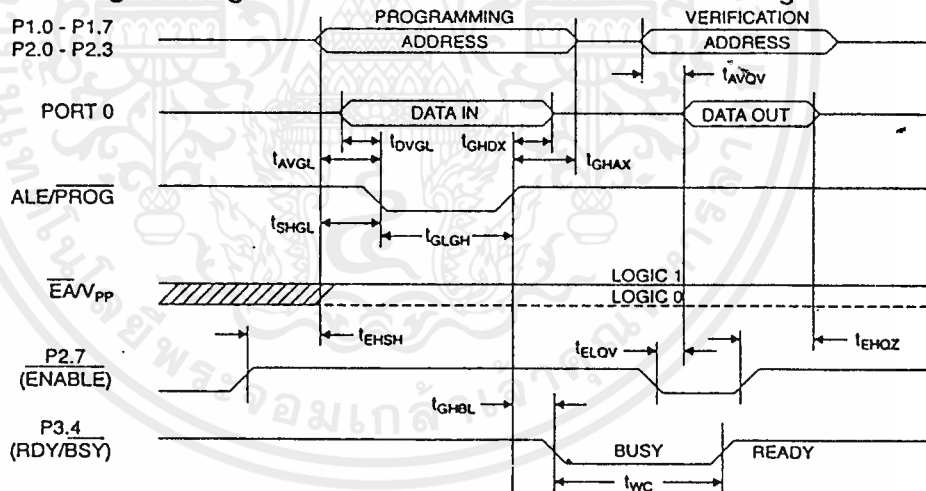
Note: 1. Only used in 12-volt programming mode.



Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage.....	6.6 V
DC Output Current.....	15.0 mA

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

D.C. Characteristics

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 5.0\text{ V} \pm 20\%$ (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6\text{ mA}$		0.45	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, $\overline{PSEN}$)	$I_{OL} = 3.2\text{ mA}$		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	$I_{OH} = -60\text{ }\mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -25\text{ }\mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10\text{ }\mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800\text{ }\mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -300\text{ }\mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80\text{ }\mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{ V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{ V}$		-650	μA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{ V}$		100	μA
		$V_{CC} = 3\text{ V}$		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port:

Port 0: 26 mA

Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power Down is 2 V.





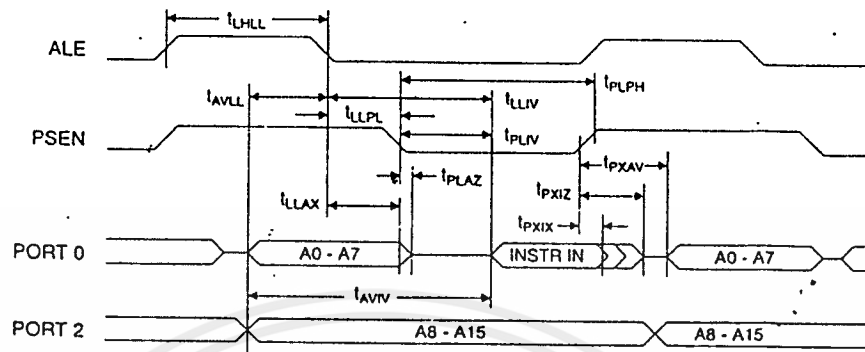
A.C. Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; Load Capacitance for all other outputs = 80 pF)

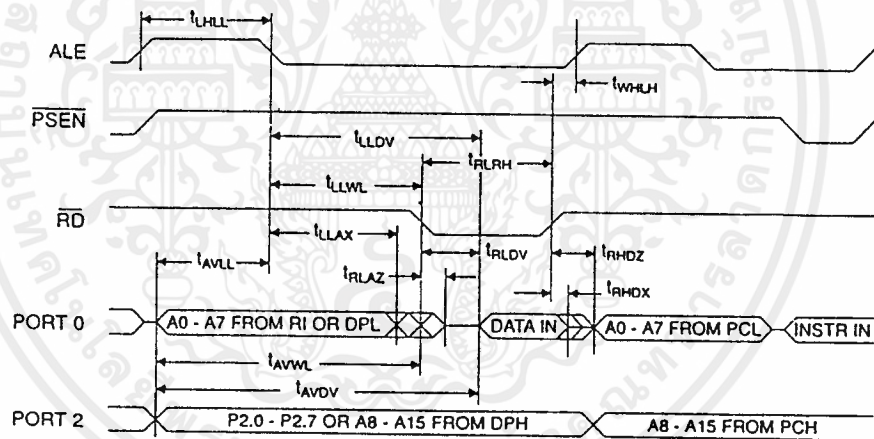
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
1/t _{CLCL}	Oscillator Frequency			0	24	MHz
t _{LHLL}	ALE Pulse Width	127		2t _{CLCL} -40		ns
t _{AVLL}	Address Valid to ALE Low	28		t _{CLCL} -13		ns
t _{LLAX}	Address Hold After ALE Low	48		t _{CLCL} -20		ns
t _{LLIV}	ALE Low to Valid Instruction In		233		4t _{CLCL} -65	ns
t _{LLPL}	ALE Low to PSEN Low	43		t _{CLCL} -13		ns
t _{LPLH}	PSEN Pulse Width	205		3t _{CLCL} -20		ns
t _{LPLV}	PSEN Low to Valid Instruction In		145		3t _{CLCL} -45	ns
t _{PIXI}	Input Instruction Hold After PSEN	0		0		ns
t _{PIXZ}	Input Instruction Float After PSEN		59		t _{CLCL} -10	ns
t _{PIXV}	PSEN to Address Valid	75		t _{CLCL} -8		ns
t _{AVIV}	Address to Valid Instruction In		312		5t _{CLCL} -55	ns
t _{PLAZ}	PSEN Low to Address Float		10		10	ns
t _{RLRH}	RD Pulse Width	400		6t _{CLCL} -100		ns
t _{WLWH}	WR Pulse Width	400		6t _{CLCL} -100		ns
t _{RLDV}	RD Low to Valid Data In		252		5t _{CLCL} -90	ns
t _{RHOX}	Data Hold After RD	0		0		ns
t _{RHOZ}	Data Float After RD		97		2t _{CLCL} -28	ns
t _{LLDV}	ALE Low to Valid Data In		517		8t _{CLCL} -150	ns
t _{AVDV}	Address to Valid Data In		585		9t _{CLCL} -165	ns
t _{LLWL}	ALE Low to RD or WR Low	200	300	3t _{CLCL} -50	3t _{CLCL} +50	ns
t _{AVWL}	Address to RD or WR Low	203		4t _{CLCL} -75		ns
t _{OVWX}	Data Valid to WR Transition	23		t _{CLCL} -20		ns
t _{OVWH}	Data Valid to WR High	433		7t _{CLCL} -120		ns
t _{WHQX}	Data Hold After WR	33		t _{CLCL} -20		ns
t _{RLAZ}	RD Low to Address Float		0		0	ns
t _{WLH}	RD or WR High to ALE High	43	123	t _{CLCL} -20	t _{CLCL} +25	ns

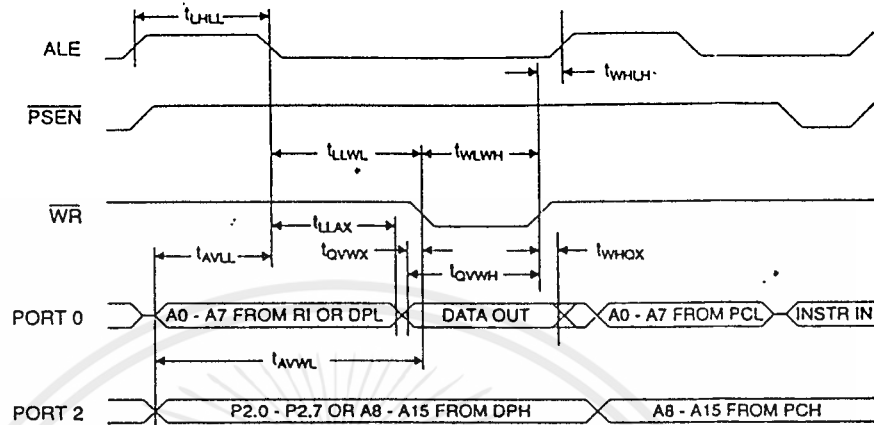
External Program Memory Read Cycle



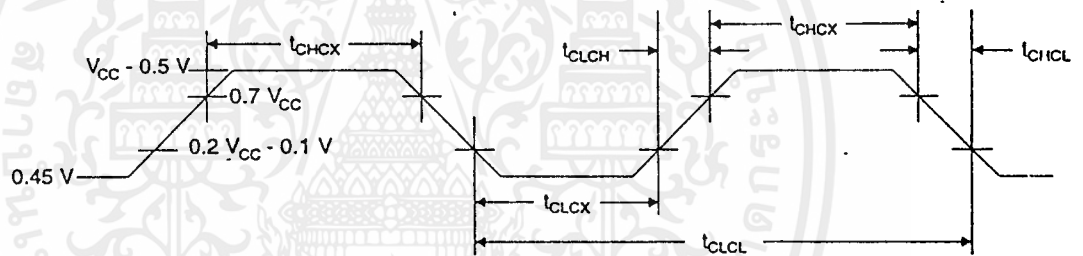
External Data Memory Read Cycle



External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

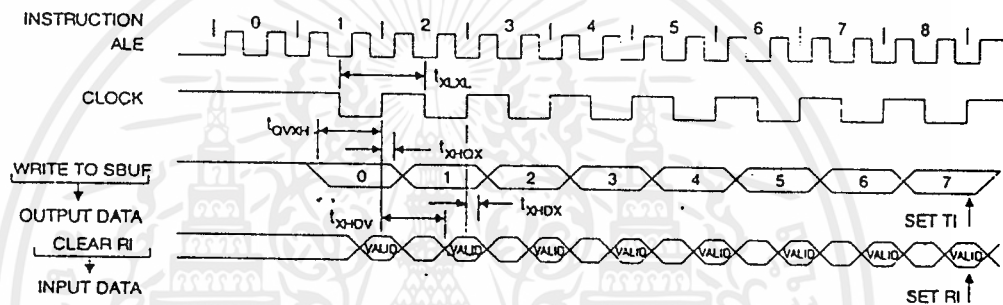
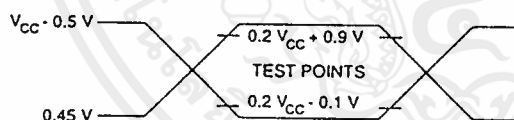
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

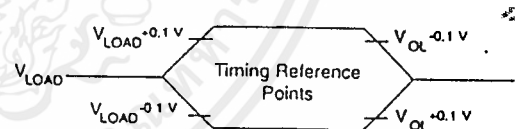
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t _{XLXL}	Serial Port Clock Cycle Time	1.0		12t _{CLCL}		μs
t _{OVXH}	Output Data Setup to Clock Rising Edge	700		10t _{CLCL} -133		ns
t _{XHOX}	Output Data Hold After Clock Rising Edge	50		2t _{CLCL} -33		ns
t _{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t _{XHDV}	Clock Rising Edge to Input Data Valid		700		10t _{CLCL} -133	ns

Shift Register Mode Timing Waveforms

AC Testing Input/Output Waveforms⁽¹⁾

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

Float Waveforms⁽¹⁾

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

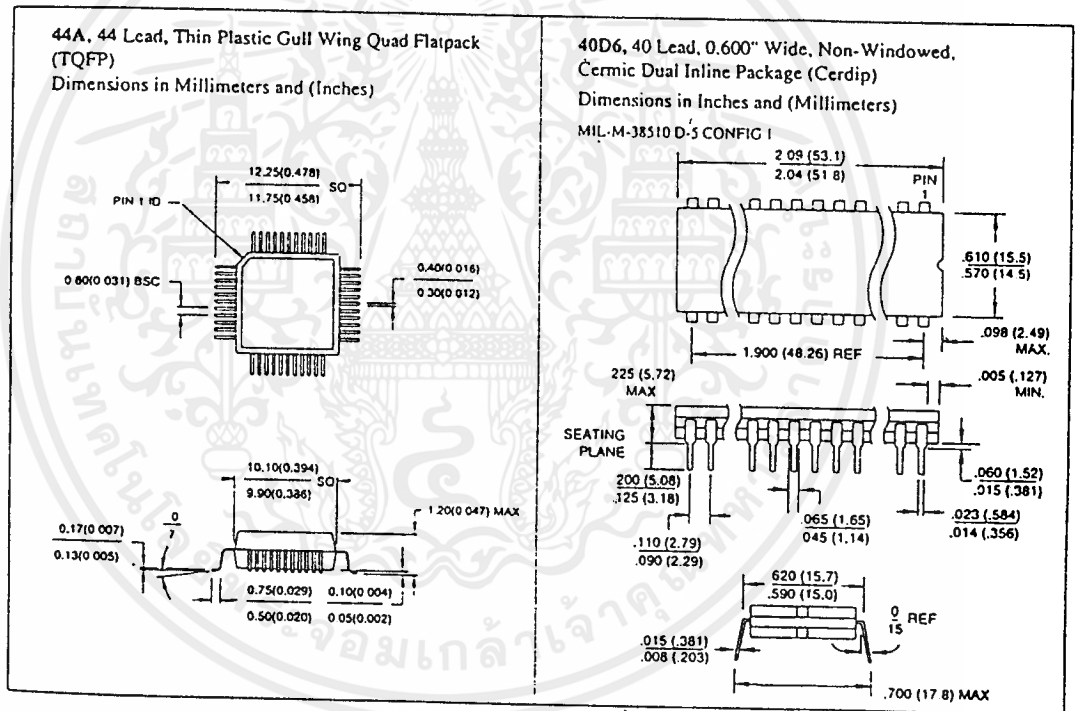
Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5 V $\pm$ 20%	AT89C51-12AC AT89C51-12JC AT89C51-12PC AT89C51-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-12AI AT89C51-12JI AT89C51-12PI AT89C51-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-12AA AT89C51-12JA AT89C51-12PA AT89C51-12QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
	5 V $\pm$ 10%	AT89C51-12DM AT89C51-12LM	40D6 44L	Military (-55°C to 125°C)
		AT89C51-12DM/883 AT89C51-12LM/883	40D6 44L	Military/883C Class B, Fully Compliant (-55°C to 125°C)
16	5 V $\pm$ 20%	AT89C51-16AC AT89C51-16JC AT89C51-16PC AT89C51-16QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-16AI AT89C51-16JI AT89C51-16PI AT89C51-16QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-16AA AT89C51-16JA AT89C51-16PA AT89C51-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
	5 V $\pm$ 20%	AT89C51-20AC AT89C51-20JC AT89C51-20PC AT89C51-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-20AI AT89C51-20JI AT89C51-20PI AT89C51-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-24AC AT89C51-24JC AT89C51-24PC AT89C51-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
24	5 V $\pm$ 20%	AT89C51-24AI AT89C51-24JI AT89C51-24PI AT89C51-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

Ordering Information

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TOFP)
40D6	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
44L	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

Packaging Information



เอกสารอ้างอิง

Daughrth, Kevin M. ,Analog-to-digital conversion : a practical approach ,pp.103-128
,Mcgraw-Hill, New York, 1994

Horowitz P.,and Hill wanfield, The Art of Electronics ,2nd ed.,pp.730-733, Cambridge
University press, New York, 1989

สุนทร วิฑูรพจน์, การใช้งานไมโครคอนโทรลเลอร์ตระกูล8051, บริษัท ซีเอ็ดดูเคชั่น
จำกัด(มหาชน), 2537



ประวัติผู้เขียน

นายสัตยชต สกฤจิตตเจริญ เกิดวันที่ 20 ตุลาคม พ.ศ. 2518 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นและระดับมัธยมศึกษาตอนปลายจากโรงเรียนเบญจมราชูทิศ จันทบุรี แล้วเข้าศึกษาต่อในระดับปริญญาตรีสาขาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และสำเร็จการศึกษาในปีการศึกษา 2540

นายอนันตชัย นัยจิตร เกิดวันที่ 12 มีนาคม พ.ศ. 2517 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นและระดับมัธยมศึกษาตอนปลายจากโรงเรียนพิษณุโลกพิทยาคม พิษณุโลก แล้วเข้าศึกษาต่อในระดับปริญญาตรีสาขาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และสำเร็จการศึกษาในปีการศึกษา 2540

