

ปริญญานิพนธ์

แบบจำลองระบบจอดรถ

PARKING LOT SYSTEM MODEL



นายเจตน์กาญจน์

อินทร์ม่วง

นายเด่นศักดิ์

อินตาคำ

นายวีระพล

ร้องครอง

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

ผท.

๖๖๕๒

๒๕๔๗

สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๔๗

โดย...เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
เลขทะเบียน...๕๙๔๖๔...
วัน,เดือน,ปี... ๕ ส.ย. ๒๕๔๗

๑๑๒๖๘๙๔๕
b.๑๑๒๖๘๙๔๕
๑๑๒๖๘๙๔๕

ปริญญานิพนธ์

เรื่อง แบบจำลองระบบจอดรถ
Parking Lot System Model

วัตถุประสงค์

1. เพื่อศึกษาการทำงานและหลักการใช้งานบัตรบาร์โค้ด
2. เพื่อออกแบบเครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ดและมีระบบคิดเงินแสดงผลออกทางจอแอลซีดี
3. เพื่อสร้างเครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ดและมีระบบคิดเงินแสดงผลออกทางจอแอลซีดี
4. เพื่อหาคุณภาพของเครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ดและมีระบบคิดเงินโดยแสดงผลออกทางจอแอลซีดี
5. เพื่อเป็นต้นแบบที่สามารถนำไปใช้งานได้จริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รู้ถึงหลักการทำงานและหลักการใช้งานบัตรบาร์โค้ด
2. ได้ต้นแบบเครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ด และมีระบบคิดเงินโดยแสดงผลออกทางจอแอลซีดี
3. ได้โครงสร้างของเครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ด และมีระบบคิดเงินโดยแสดงผลออกทางจอแอลซีดี
4. ได้คุณภาพของเครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ด และมีระบบคิดเงินโดยแสดงผลออกทางจอแอลซีดี
5. ได้เครื่องบันทึกทะเบียนรถที่หลักลงในบัตรบาร์โค้ด และมีระบบคิดเงินโดยแสดงผลออกทางจอแอลซีดี ไปใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	แบบจำลองระบบจอตกร	
นักศึกษา	นายเจตน์กาญจน์	อินทร์ม่วง
	นายเด่นศักดิ์	อินตาคำ
	นายวีระพล	ร้องครอง
อาจารย์ที่ปรึกษา	อาจารย์สุรพงษ์	สิริพงศ์ดี
อาจารย์ที่ปรึกษาร่วม	ผศ.วิสุทธิ	อิทธิพรธรรม
หลักสูตร	ครุศาสตรบัณฑิต	
สาขาวิชา	เทคโนโลยีการวัดคุมทางอุตสาหกรร	
ปีการศึกษา	2547	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบ และสร้างแบบจำลองระบบจอตกรแบบจำลองนี้แบ่งเป็น 2 ส่วนคือ ส่วนที่หนึ่งเป็นส่วนของการป้อนข้อมูลจากคีย์สวิตช์ และส่วนที่สองคือส่วนของภาคแสดงผลโดยใช้จอแอลซีดีในการแสดงผลซึ่งแบบจำลองระบบจอตกรนี้สามารถนำไปประยุกต์ใช้งานได้

Thesis Title	Parking Lot System Model	
Students	Mr.Jetagarn	Inmong
	Mr.Densak	Intakham
	Mr.Weerapol	Rongkrong
Advisor	Mr.Surapong	Siripongdee
Co-Advisor	Asst.Prof.Wisuit	Atipornatum
Education Level	Bachelor of Science in Industrial Education	
Program in	Industrial Instrument Technology	
Academic Year	2004	

ABSTRACT

This study presents the design and model of parking system that is divided into two parts. The first part is the process of data supply by the switch key and the second part is the output representation on LCD screen. From the study, this model of parking system can be applied practically.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริิณญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้ด้วยดีนั้นเนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่านคณะผู้จัดทำขอขอบคุณท่านอาจารย์ที่ปรึกษาทั้งสองท่านที่ได้ให้คำปรึกษา ให้คำแนะนำ แนวความคิด ความรู้ต่างๆ และการแก้ไขปัญหาในการจัดทำปริิณญานิพนธ์รวมทั้งขอขอบคุณท่านอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ได้ให้ความช่วยเหลือ และคำแนะนำนอกจากนี้ต้องขอขอบคุณรุ่นพี่ เพื่อนๆ ที่ช่วยแนะนำให้คำปรึกษาสุดท้ายขอขอบคุณบุคคลที่สำคัญ และระลึกถึงคือ บิดาและมารดา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 จิตความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 บาร์โค้ด (Barcode)	3
2.2.1 ความเป็นมาของบาร์โค้ด	3
2.2.2 หลักการของบาร์โค้ด	3
2.2.3 การอ่านบาร์โค้ด	4
2.2.4 ชนิดของบาร์โค้ด	5
2.3 โมดูล LCD	6
2.3.1 รายละเอียดเกี่ยวกับโมดูล LCD	6
2.3.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD	6
2.3.3 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16X2)	7
2.3.4 คำสั่งควบคุมโมดูล LCD	8
2.3.5 การเขียนคำสั่งและข้อมูลให้กับ LCD	11
2.4 ไมโครคอนโทรลเลอร์	12
2.5 RTC	15
2.5.1 RTC หรือ Real Time Clock	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.5.2 โครงสร้างและขาสัญญาณต่างๆ	18
2.5.3 การเชื่อมต่อ RTC เข้ากับไมโครคอนโทรลเลอร์	19
บทที่ 3 การออกแบบการสร้างและการทำงาน	20
3.1 กล่าวนำ	20
3.2 การออกแบบวงจร	21
3.2.1 ส่วนของวงจรรับข้อมูล	21
3.2.2 ส่วนแสดงผล	22
3.3 ซอฟต์แวร์	24
3.3.1 การรับข้อมูลมาแสดงผล	24
3.2.2 การแสดงผล	24
บทที่ 4 การทดลองและผลการทดลอง	25
4.1 การทดลองส่วนรับข้อมูล	25
4.1.1 การทดลองส่วนรับข้อมูลจากบัตรบาร์โค้ด	25
4.1.2 การทดลองส่วนรับข้อมูลจากคีย์เมตริกซ์สวิตช์	26
4.2 การทดลองส่วนของการแสดงผล	27
บทที่ 5 บทสรุป	29
5.1 สรุป	29
5.2 ปัญหาและแนวทางแก้ไข	29
5.3 แนวทางการพัฒนา	30
บรรณานุกรม	31
ภาคผนวก ก เครื่องต้นแบบ	32
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	34
ภาคผนวก ค รายการอุปกรณ์	38
ภาคผนวก ง แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม	42
ภาคผนวก จ คู่มือการใช้งาน	84
ภาคผนวก ฉ รายละเอียดและคุณสมบัติของอุปกรณ์	89

ประวัติผู้แต่ง เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
ค.1 รายการอุปกรณ์ของวงจรประมวลผล	39
ค.2 รายการอุปกรณ์ของวงจรคีย์เมตริกซ์สวิตช์	39
ค.3 รายการอุปกรณ์ของวงจรเวลา	40
ค.4 รายการอุปกรณ์ของวงจรแหล่งจ่ายแรงดัน	40
ค.5 รายการอุปกรณ์ของวงจรแสดงผล	41



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การแทนค่าเลขฐานสองของแถบต่างๆ	4
2.2 สัญลักษณ์ที่อ่านได้จากหัวอ่าน	5
2.3 ขาต่างๆ ของจอ LCD ขนาด 16 ตัวอักษร 2 บรรทัด	7
2.4 รายละเอียดของรูปแบบข้อมูลคำสั่ง	8
2.5 รายละเอียดของรูปแบบข้อมูลคำสั่ง	9
2.6 รายละเอียดของรูปแบบข้อมูลคำสั่ง	9
2.7 รายละเอียดของรูปแบบข้อมูลคำสั่ง	10
2.8 รายละเอียดของรูปแบบข้อมูลคำสั่ง	11
2.9 โครงสร้างพอร์ท (Port) 0 (บิต)	13
2.10 โครงสร้างของพอร์ท 1 (บิต)	14
2.11 โครงสร้างของพอร์ท 2 (บิต)	14
2.12 โครงสร้างของพอร์ท 3 (บิต)	15
2.13 ชิพ RTCเบอร์ DS1302	16
2.14 ชิพ RTC เบอร์ DS1307	17
2.15 โครงสร้างของชิพ RTC เบอร์ DS1302	18
2.16 การเชื่อมต่อ DS1302 เข้ากับ MCS-51	19
3.1 โครงสร้างการทำงานของแบบจำลองระบบจอตลอด	20
3.2 ชุดหัวอ่านรหัสบาร์โค้ด	21
3.3 รูปกายเมตริกซ์สวิตช์	22
3.4 รูป RTC	23
3.5 รูป LCD	24
4.1 วงจรส่วนรับข้อมูลจากบัตรรหัสบาร์โค้ด	25
4.2 วงจรกายเมตริกซ์สวิตช์	26
4.3 รูปวงจรการต่อแสดงผล	26
4.4 รูปผลการทดลองแอลซีดี	27
ก.1 ภาพค่าน้ำของแบบจำลองระบบจอตลอด	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.2 ภาพด้านข้างของแบบจำลองระบบจรวด	33
ข.1 วงจรแบบจำลองระบบจรวด	35
ข.2 แผ่นวงจรพิมพ์แบบจำลองระบบจรวดด้านหน้า	36
ข.3 แผ่นวงจรพิมพ์แบบจำลองระบบจรวดด้านหลัง	37
ง.1 ผังงานโปรแกรมของแบบจำลองระบบจรวด	43
จ.1 ส่วนประกอบและปุ่มควบคุมของตัวเครื่องแบบจำลองระบบจรวด	88



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันการจ่อครยนต์ จะใช้พนักงานในการเก็บค่าบริการจ่อครด ซึ่งทำให้เสียเวลาในการคิดค่าบริการ และมักจะประสบปัญหาการถูกโจรกรรมรยนต์ที่เกิดขึ้นอยู่เป็นประจำ เพราะเจ้าหน้าที่รักษาความปลอดภัยไม่สามารถที่จะทราบได้ว่าผู้ที่ขั้บรยนต์ออกมาใช้เจ้าของรยนต์ตัวจริงหรือไม่

จากปัญหาดังกล่าวถ้ามีอุปกรณ์ที่ช่วยอำนวยความสะดวกให้กับผู้ใช้บริการเพื่อให้ผู้ใช้บริการเกิดความปลอดภัยในทรัพย์สิน และอุ่นใจในการใช้บริการ โดยมีการบันทึกทะเบียนรยนต์ผ่านทางบัตรบาร์โค้ด และยังแสดงค่าบริการอีกด้วย จึงได้จัดทำแบบจำลองระบบจ่อครดขึ้นมาเพื่อช่วยอำนวยความสะดวก

จากปัญหาที่เกิดขึ้นในปัจจุบันในการคิดค่าจ่อครดนั้นพนักงานจะเป็นผู้คิดค่าบริการซึ่งอาจเกิดการผิดพลาดได้ พนักงานจึงได้ใช้เครื่องคำนวณเข้ามาช่วยแต่ก็ยังเสียเวลาอยู่ดี และพนักงานยังต้องคอยดูแลรถของลูกค้าที่มีจำนวนมากให้ปลอดภัยจากการถูกโจรกรรม การดูแลรยนต์อาจทำได้ไม่ทั่วถึงเนื่องจากสถานที่จ่อครดมีบริเวณกว้างผู้ประกอบการจึงต้องเพิ่มพนักงานดูแลรถของลูกค้าให้ทั่วถึงจากการถูกโจรกรรมแต่ก็ยังมีการโจรกรรมเกิดขึ้นเนื่องจากพนักงานดูแลไม่สามารถทราบได้ว่าผู้ที่ขั้บรยนต์ออกมาใช้เจ้าของตัวจริงหรือไม่

ดังนั้นผู้จัดทำจึงได้จัดทำแบบจำลองระบบจ่อครดที่มีคุณสมบัติบันทึกทะเบียนรถ 4 หลักลงในหน่วยความจำของไมโครคอนโทรลเลอร์เพื่อผู้ใช้บริการจะได้นำมายืนยันการเป็นเจ้าของรยนต์ มีระบบคิดเงินโดยแสดงผลออกทาง LCD ซึ่งคำนวณค่าจ่อครดได้ตามเวลาที่กำหนด และตั้งค่าใช้จ่ายในการจ่อครดได้จากการคีย์สวิตช์

1.2 ขีดความสามารถของโครงการ

โครงการนี้มีขีดความสามารถดังนี้

- 1) รับบัตรและบันทึกทะเบียนรถได้สี่หลัก
- 2) มีระบบคิดเงินโดยแสดงผลออกทางจอแอลซีดี
- 3) คำนวณค่าจ่อครดได้ตามเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตั้งค่าใช้จ่ายในการจ่อครถได้จากคิยส์วิทซ์

1.3 เนื้อหาโดยสังเขป

ในปริญญานิพนธ์นี้ ได้มีการรวบรวม ข้อมูลรายละเอียด ทั้งทฤษฎี หลักการทำงาน การออกแบบและการสร้างวงจรที่ใช้งานและจัดทำคู่มือการใช้งาน และจัดทำคู่มือการใช้งาน ซึ่งมีเนื้อหาโดยสังเขปของแต่ละบทดังนี้

บทที่ 1 บทนำกล่าวความเป็นมา และความสำคัญของปริญญานิพนธ์, จิตความสามารถของโครงการ และเนื้อหาโดยสังเขปของแต่ละบท

บทที่ 2 ทฤษฎี และหลักการของชุดหัวอ่านบัตรบาร์โค้ด, จอ LCD, MCS 51, RTC

บทที่ 3 การออกแบบการสร้างการทำงานในส่วนของฮาร์ดแวร์ดังนี้ คือ ไอซีเรกูเรเตอร์, การต่อ LCD เข้ากับ MCS 51, การต่อ RTC เข้ากับ MCS ส่วนของซอฟต์แวร์มีดังนี้ การโปรแกรม MCS 51 เพื่อทำงานร่วมกับบัตรบาร์โค้ด, RTC และแสดงผลออกทางจอ LCD

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 บทสรุปปัญหาแนวทางแก้ไขและพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

ในระบบการควบคุมการเข้าออกโดยใช้บัตรนักศึกษามาใช้งานเป็นบัตรสำหรับผ่านเข้าออกระบบโดยใช้ข้อมูลบนบัตรเป็นข้อมูลประจำของผู้ถือบัตรรวมถึงการกดรหัสผ่านซึ่งบัตรนักศึกษามีรูปแบบของข้อมูลบันทึกอยู่ในรหัสบาร์โค้ด

2.2 บาร์โค้ด

2.2.1 ความเป็นมาของบาร์โค้ด

สหรัฐอเมริกาได้ออกสิทธิบัตรรับรองบาร์โค้ดขึ้นเมื่อปี ค.ศ.1994 ในแบบที่เรียกว่า Circular bar code ต่อมาในปี ค.ศ.1960 ก็มีการรับรองบาร์โค้ดแบบนี้เรียกว่า Rail identification symbol หลังจากนั้นเป็นต้นมาเทคนิคของบาร์โค้ดรูปแบบต่างๆ ก็มีมากขึ้น และเริ่มใช้งานจริงจังเมื่อปี ค.ศ.1970 เมื่อคณะกรรมการบริหารด้านห้างสรรพสินค้าของสหรัฐอเมริกา ได้นำรหัสที่เรียกว่า UPC (Universal Product Code) ซึ่งเป็นรหัสที่ใช้กันมากในสินค้าออกเผยแพร่ และใช้กันแพร่หลายในสหรัฐอเมริกา และยุโรปตั้งแต่ปี ค.ศ. 1973 และปี ค.ศ. 1977 ตามลำดับ

การใช้งานด้านอื่นๆ เริ่มตั้งแต่ปี ค.ศ. 1980 เป็นต้นมาแต่คนส่วนใหญ่เริ่มคุ้นเคยกับบาร์โค้ดเป็นอย่างดีจากการรหัสสินค้า และการชำระเงินที่คอมพิวเตอร์รวมออกมาจากการอ่านบาร์โค้ดสินค้าเหล่านั้นจากความสะดวกเหล่านี้สามารถลดพนักงาน ณ จุดนี้ลงได้

ในปี ค.ศ.1981 มีห้างสรรพสินค้ามากกว่า 4,000 แห่งในสหรัฐอเมริกา และแคนาดาใช้บาร์โค้ดในธุรกิจนี้ นอกจากนี้ยังใช้กับกิจการอื่น เช่น ห้องสมุด บริการสุขภาพ งานบริการ การผลิตสินค้า เป็นต้น

2.2.2 หลักการของบาร์โค้ด

บาร์โค้ดเป็นการแทนข้อมูลที่เป็นรหัสฐานสอง (Binary Code) ในรูปแบบของแถบขาวดำที่มีความกว้างแคบแตกต่างกันวางเรียงขนานสลับกัน ด้วยจำนวนของแถบรหัสสลับกันไปขึ้นอยู่กับข้อมูลที่ต่างกัน และชนิดของบาร์โค้ดที่เลือกใช้ต่างกันด้วย

การเข้ารหัสของบาร์โค้ด แบ่งเป็น 2 วิธีการคือ

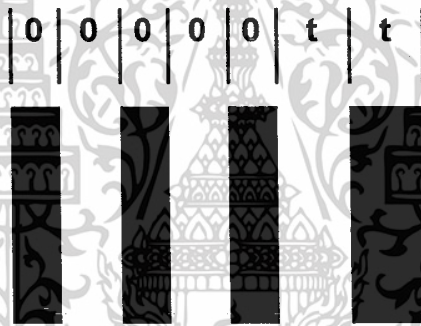
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. จะใช้สีของบาร์โค้ดนำมาเข้ารหัสโดยใช้แถบสีดำแทน “1” และแถบสีขาวหรือแถบว่างเป็น “0” ลักษณะเช่นนี้เรียกว่า “เดลต้าโค้ด” (Delta Code)

2. จะใช้ความกว้างของแถบรหัสนำมาเข้ารหัสโดยถ้าเป็นแถบกว้างจะเป็น “1” และแถบแคบจะเป็น “0” ลักษณะเช่นนี้เรียกว่า “วิทช์โค้ด” (Width Code) การเข้ารหัสชนิดนี้จะไม่สนใจสีของแถบเลย

แถบขาว-ดำ มีลักษณะ และชื่อที่ใช้คือ

- แถบสีดำที่มีความกว้างมากเรียกว่า Wide Bar
- เส้นหรือแถบสีดำที่มีความกว้างน้อยเรียกว่า Narrow Bar
- ช่องว่าง (แถบสีขาว) ที่มีความกว้างมากเรียกว่า Wide Space
- ช่องว่าง (เส้นสีดำ) ที่มีความกว้างน้อยเรียกว่า Narrow Space



รูปที่ 2.1 การแทนค่าเลขฐานสองของแถบต่างๆ

2.2.3 การอ่านบาร์โค้ด

ในการอ่านบาร์โค้ดใช้หลักการเปลี่ยนรหัสบาร์โค้ดให้เป็นรหัสแอสกี โดยอาศัยความแตกต่างกันระหว่างแถบเข้มกับพื้นที่ว่าง โดยพื้นที่ว่าง (ปกติจะเป็นสีขาวหรือสีอ่อน) จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบเข้ม (ซึ่งใช้สีดำหรือสีที่มีความเข้มมาก) หัวอ่าน (Bar Code Reader) จะประกอบไปด้วยตัวกำเนิดแสงที่ผ่านเลนส์ออกมา โดยถูกบังคับทิศทางให้มีจุดรวมแสงเล็กที่สุดกับตัวรับแสงที่มีความไวสูง ทั้งสองอย่างนี้จะบรรจุไว้ในหัวอ่านเดียวกันที่มีหลายรูปแบบ แต่แบบที่เป็นพื้นฐานที่สุดอยู่ในรูปคล้ายปากกาขนาดใหญ่ (Wand type)

หัวอ่านจะสแกนผ่านรหัสบาร์โค้ดในขณะที่ตัวกำเนิดแสงจะทำให้เกิดแสงส่องผ่านเลนส์ไปกระทบบนรหัสบาร์โค้ด และสะท้อนกลับจากแถบกลับไปยังตัวรับแสง (Photo sensor) ที่เกิดค่าความแตกต่างขึ้นตามหลักการสะท้อนกลับในแต่ละแถบ ทำให้เกิดสถานะลอจิก “1” และ “0” เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมดตลอดความกว้างของทุกแถบแล้ว จะตรงกับแพตเทิร์นที่ได้กำหนดไว้แล้วในหัวอ่านรหัสบาร์โค้ด จะใช้ตัวกำเนิดแสงสีแดงหรือสีขาวยกเว้นส่วนใหญ่จะใช้แสงสีแดง เนื่องจากแสงสีขาวต้องการพลังงาน และความเข้มของแสงสูงมากกว่าแสงสีแดง แสงสีแดงสามารถอ่านรหัสที่พิมพ์ด้วยสีต่างๆ ได้ทุกสียกเว้นรหัสที่พิมพ์ด้วยสีแดง

องค์ประกอบสำคัญสองประการที่จำเป็นอย่างมากในการอ่านบาร์โค้ดได้ถูกต้อง

ประการแรกคือ พื้นที่ภายในแถบ และช่องว่างจะต้องทำให้เกิดความแตกต่างของการสะท้อนกลับอย่างมาก (Contrast) เช่น แถบสีดำ และช่องว่างสีขาว เป็นต้น ซึ่งปกติความแตกต่างนี้จะต้องอยู่ในช่วงระหว่างอัตรา 80-90% ขึ้นไป

ประการที่สองคือ ความกว้างระหว่างแถบกว้างหรือช่องว่างต่อแถบแคบหรือช่องว่างแคบ จะเป็นอัตราส่วน 2 : 0.5, 2 : 1 และ 3 : 1

ตามหลักของบาร์โค้ดแล้วสัญญาณที่อ่านได้จากหัวอ่านจะไม่ขึ้นกับชนิดของบาร์โค้ดแต่จะขึ้นกับแถบขาว-ดำที่รู้ค่าคือ ถ้าเป็นแถบดำสัญญาณที่ได้จากหัวอ่านจะเป็น “1” ถ้าเป็นแถบขาวสัญญาณที่อ่านได้จะเป็น “0” ความกว้างของสัญญาณที่อ่านได้จะเท่ากับความกว้างของแถบขาว-ดำ



รูปที่ 2.2 สัญลักษณ์ที่อ่านได้จากหัวอ่าน

2.2.4 ชนิดของบาร์โค้ด

ปัจจุบันบาร์โค้ดที่นิยมใช้กันแพร่หลาย แบ่งได้เป็น

1. ชนิดรหัส 2 ใน 5 (2 of 5 code)
2. ชนิดรหัส 2 ใน 5 แบบสอดแทรก (interleaved 2 of 5)
3. ชนิดรหัส 3 ใน 9 (3 of 9 or 39 code)
4. ชนิดรหัส Codabar
5. ชนิดรหัส UPC (Universal Product Code)
6. ชนิดรหัส EAN (European Article Numbering)
7. ชนิดรหัส Code 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โมดูล LCD

2.3.1 รายละเอียดเกี่ยวกับโมดูล LCD

ในโมดูล LCD ประกอบไปด้วย 3 ส่วนหลักๆ คือ

1. ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็น โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD
2. ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิพควบคุมโดยเฉพาะชิพที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักษรส่วน HD61830 ใช้ควบคุมแบบกราฟิก
3. ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผล แสดงข้อมูลตามที่กำหนดชิพที่ใช้เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

2.3.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้าง และคำสั่งที่ใช้ในการควบคุมให้ดียิ่งขึ้น

1. บัฟเฟอร์อินพุต (Input) เอาต์พุต (Output) เป็นตัวที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอกเพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม
2. รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล
3. รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผลหรือนำข้อมูลไปสร้างตัวอักษรเพื่อเติมในแรมตัวเก็บอักษร
4. แรมเก็บข้อมูลแสดงผล (Display Data RAM : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรอม และแรมเก็บตัวอักษรเพื่อนำไปแสดงที่ตัวแสดงผล
5. รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านข้อมูลแล้วไปแสดงที่ตัวแสดงผลได้ซึ่งมีขนาด 7,200 บิตโดยจะถูกอ่านค่าของข้อมูลใน DDRAM

6. แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บตัวอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ในกรณีที่ตัวอักษรในรอมเก็บตัวอักษรไม่เพียงพอมีขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

512 บิต การเขียนและการอ่านค่า ไปใช้นั้นทำได้เช่นเดียวกับรอมเก็บตัวอักษรคือ เขียนข้อมูลลงใน
แรมเก็บข้อมูลแสดงผลแล้วตัวควบคุมจะมาอ่านค่าจากแรมเก็บตัวอักษรเอง

7. แฟล็ก Busy เป็นส่วนหนึ่งที่แจ้งสถานะ การทำงานของตัวควบคุมให้อุปกรณ์ภายนอก
ทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัว
ควบคุมต้องตรวจสอบสถานะของแฟล็ก Busy นี้เสียก่อน

2.3.3 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด

โมดูล LCD ขนาด 16×2 มีขาต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดการทำงานของแต่ละขามี
ดังนี้



รูปที่ 2.3 ขาต่างๆ ของจอ LCD ขนาด 16 ตัวอักษร 2 บรรทัด

ขา 1 : กราวด์ (Gnd)

ขา 2 : +5 โวลต์ (Volt)

ขา 3 : ปรับความสว่าง (Brightness)

ขา 4 : Rs

ขา 5 : R/W

ขา 6 : E

ขา 7-14 : D0-D7

1. ขา 1 (Vss) ใช้สำหรับต่อกราวด์

2. ขา 2 (Vdd) ต่อไฟเลี้ยง +5 โวลต์

3. ขา 3 (Vo) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ขา 4 (Rs) เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำกรประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่งแต่ถ้าขาเป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

5. ขา 5 (R/W) เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

6. ขา 6 (E) เป็นขาอินาเบิล LCD ให้ทำงาน

7. ขา 7-14 (D0-D7) เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก ขนาด 8 บิต 9 บิต

2.3.4 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุมแน่นอนว่าจะต้องกำหนดให้ขา RS และ R/W เป็น “0” แล้วเขียนคำสั่งควบคุม คำสั่งควบคุม โมดูล LCD ของชิพควบคุม HD44780 ที่สำคัญมีดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (Clear Display) มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างเข้าไปในแรมเก็บข้อมูลแสดงผลเมื่อตัวควบคุมเอ็กซ์คิวิต์คำสั่งนี้จะทำการกำหนดแอดเดรสของแรมเก็บข้อมูลแสดงผลเป็น “0” เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผลแล้วเซตบิต I/D ให้เป็น “1”

2. คำสั่ง Return Home ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์ที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผลแต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลงนั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry Mode Set) มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	1	I/D	S

รูปที่ 2.4 รายละเอียดของรูปแบบข้อมูลคำสั่ง

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูลถ้าหากบิต S เป็น “1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผลตัวเคอร์เซอร์จะอยู่กับที่แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้ายแต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่าเมื่อเขียนหรืออ่านข้อมูลแล้วทำให้แอดเดรสของแรมเก็บข้อมูลแสดงผลจะเพิ่มขึ้นแต่ถ้าเป็น “0” แอดเดรสจะลดลง

ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H และที่ใช้บ่อยคือ 06H หมายถึงกำหนดให้เมื่อเกิดข้อมูลใหม่เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของแรมเก็บข้อมูลแสดงผลเพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผลมีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	1	D	C	B

รูปที่ 2.5 รายละเอียดของรูปแบบข้อมูลคำสั่ง

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผลถ้าบิตนี้เป็น “1” จะเป็นการเปิดจอแสดงผลถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมตัวเคอร์เซอร์บนจอแสดงผลถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผลต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดเป็น “0” จะเป็นการปิดเคอร์เซอร์หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกระพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผลแต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผลแสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์ และข้อมูลตัวอักษรมีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	1	I/D	S

รูปที่ 2.6 รายละเอียดของรูปแบบข้อมูลคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการเลื่อนเคอร์เซอร์ และตัวอักษรบนจอแสดงผลขึ้นอยู่กับกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

6. คำสั่งกำหนดฟังก์ชันการทำงานมีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต 5	บิต 4	บิต 3	บิต 2	บิต1	บิต 0
0	0	1	DL	N	F	*	*

รูปที่ 2.7 รายละเอียดของรูปแบบข้อมูลคำสั่ง

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูลถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิตแต่ถ้าเป็น “1” จะเป็น 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผลถ้าเป็น “0” จะแสดงผลเป็น 1 บรรทัดถ้าเป็น “1” จะแสดง 2 บรรทัดในกรณีที่จอแสดงผลสามารถแสดง ได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัดก็กำหนดบิต N นี้ให้เป็น “1”

บิต F ใช้เลือกความละเอียดตัวอักษรให้การแสดงผลถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5×7 จด และถ้าเป็น “1” จะแสดงผลเป็นแบบ 5×10 จด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดโมดูล LCD ให้ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5×7 จด

7. คำสั่งเลือกแอดเดรสของแรมเก็บตัวอักษรเมื่อต้องการกำหนดแอดเดรสของแรมเก็บตัวอักษรต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของแรมเก็บตัวอักษรจะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้แรมเก็บตัวอักษรโดยแอดเดรสของแรมเก็บตัวอักษรอยู่ระหว่าง 00H-3FH

8. คำสั่งเลือกแอดเดรสของแรมเก็บข้อมูลแสดงผลใช้ในการเลือกแอดเดรสของแรมเก็บข้อมูลแสดงผลก่อนที่จะทำการอ่านหรือเขียนข้อมูลนั้นที่บิต 7 ต้องเป็น “1” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของแรมเก็บข้อมูลแสดงผลจะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนเอกสารนี้เป็นเอกสารทสวงนไวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรสยังขึ้นอยู่กับกำหนดยุคสถานะที่บิต N ด้วยหากบิต N เป็น “0” แอดเดรสของแรมเก็บข้อมูลแสดงผลจะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น “1” แอดเดรสของแรมเก็บข้อมูลแสดงผลจะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

9. คำสั่งอ่านแฟลก Busy และแอดเดรสมีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
BF	A	A	A	A	A	A	A

รูปที่ 2.8 รายละเอียดของรูปแบบข้อมูลคำสั่ง

เป็นคำสั่งที่ใช้อ่านแฟลก Busy (ในตารางจะกำหนดเป็น BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่งแต่ถ้าเป็น “1” แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลก ต้องกำหนดให้ขา R/W เป็น “1” ด้วยแต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของแรมเก็บตัวอักษร และแรมเก็บข้อมูลแสดงผลด้วยโดยบิต 0-6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.3.5 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องป้อนคำสั่งแล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูลที่ต้องการแสดงผล เนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้นคือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูลดังนั้นในการส่งคำสั่งและข้อมูล จึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “0” หมายความว่าข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่งในทางตรงกันข้ามหากขา RS ได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลในแรมเก็บตัวอักษร และแรมเก็บข้อมูลแสดงผลเริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรสจากนั้นกำหนดให้ขา RS เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น “1” ข้อมูลในขนาด 8 บิตหรือ 4 บิตก็จะปรากฏบนบัสข้อมูลโดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของแรมเก็บตัวอักษรหรือแรมเก็บข้อมูลแสดงผลตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูลเมื่อกำหนดแอดเดรส และป้อนลอจิก “1” ให้ขา RS แล้วต้องกำหนดให้ขา R/W เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงในแรมเก็บข้อมูลแสดงผลต่อไป

2.4 ไมโครคอนโทรลเลอร์

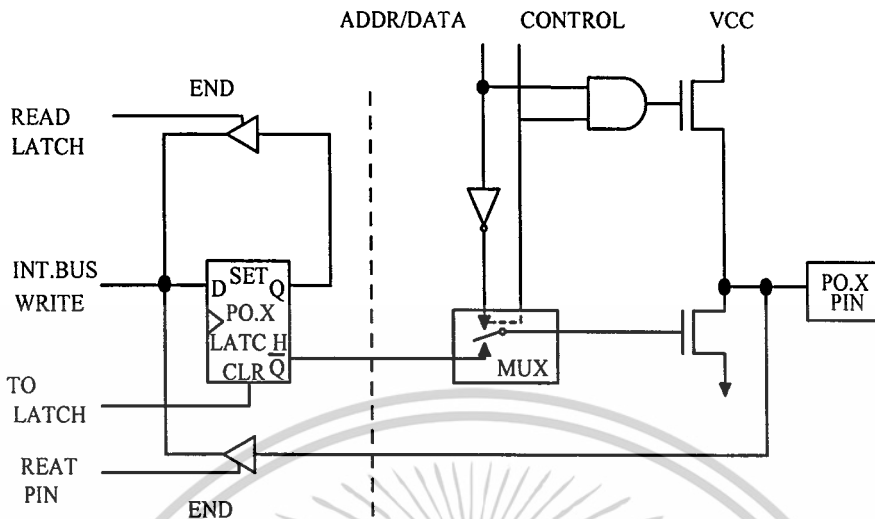
ซึ่งเกิดชิพไมโครคอนโทรลเลอร์ คือการเอาแรม (RAM) รอม (ROM) ไอโอพอร์ต (I/O Port), ไทมเมอร์ (Timer), ซีเรียลพอร์ต (Serial Port) มาใส่เข้าไปในชิพทำให้ขนาดเล็กลงมากมีผู้ผลิตออกมาหลายราย เช่น Intel, Atmel, Philips, Siemens, Dallas Semiconductor แต่ในที่นี้เราจะกล่าวถึงแค่ บริษัทฟิลิปส์

คุณสมบัติของไมโครคอนโทรลเลอร์ โดยบริษัทฟิลิปส์ทางบริษัทฟิลิปส์ได้พัฒนาออกมาหลายรุ่นโดยเพิ่ม A/D, D/A และขยายไอโอพอร์ตส่วนของโปรแกรมเมมโมรี่ก็มีทั้งแบบ Flash และ One-Time-Programmable (OTP) โดยพอร์ทของ 8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆ ดังนี้

1. ขา 40 (Vcc) +5 โวลต์

2. ขา 20 (Vss) กราวด์

3. ขา 32-39 (พอร์ท 0) มีทั้งหมด 8 บิต คือ P0.7-P0.0 มีโครงสร้างแบบ Open-Drain Bi-Directional ไม่มีความต้านทานภายในพูลอัพภายในดังแสดงในรูปที่ 2.11 โดยขา 32-39 (พอร์ท 0) มีทั้งหมด 8 บิต คือ P0.7-P0.0 ใช้งานได้ 2 หน้าที่ คือเป็นทางผ่านของแอดเดรสบัส และดาต้าบัส เมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือไอโอพอร์ต ถ้าต้องการให้ทำงานเป็นอินพุตพอร์ท ต้องส่งลอจิก “1” ไปยังพอร์ทนี้จะมีผลให้ Q ของ D-FF เป็น “0” ทำให้ Fet ตัวล่างมีสถานะ Off สัญญาณที่ใช้อ่านอินพุตพอร์ทเลขที่ โดยส่งสัญญาณ Read Latch ไปกระตุ้นที่ Tri-State Buffer ตัวบน และการอ่าน พอร์ท (pin) จะใช้สัญญาณ Read (pin) พอร์ทนี้ไม่สามารถพูลอัพภายใน



รูปที่ 2.9 โครงสร้างพอร์ต 0 (บิต)

4. ขา 1-8 (พอร์ต 1) มีทั้งหมด 8 บิตคือ P1.0-P1.7 มีโครงสร้างคล้ายพอร์ต 0 แต่จะใช้ความต้านทานพูลอัพภายในแทน (Internal Pull-up Resistor) มีโครงสร้างดังรูปที่ 2.4

5. ขา 21-28 (พอร์ต 2) มีทั้งหมด 8 บิต คือ P2.7-P2.0 มีโครงสร้างคล้ายพอร์ต 0 โดยมี Fet ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน (Internal Pull up) พอร์ตนี้ทำงาน 2 หน้าทีคือ สามารถใช้เป็นแอดเดรสขนาด 8 บิต A15-A8 และเป็นไอโอพอร์ตใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุตต้องใส่ลอจิก 1 มาที่พอร์ติก่อนเพื่อบังคับให้ Fet อยู่ในสถานะ off ดังรูปที่ 2.5

6. ขา 10-17 (พอร์ต 3) มีทั้งหมด 8 บิต คือ ขา (P3.7-P3.0) มีโครงสร้างคล้ายกับพอร์ต 1 ทำงานได้ 2 หน้าทีคือเป็นไอโอพอร์ต ถ้าจะโปรแกรมให้เป็นอินพุตพอร์ตต้องส่งลอจิก “1” เข้ามาที่พอร์ติก่อน และอีกหน้าที่หนึ่งก็คือใช้ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้า

7. ขา 30 (Ale) เป็นขาส่งสโตรบสำหรับใช้ในการแลตช์แอดเดรสไบต์ต่ำ (A7-A0) ที่ส่งออกมาจาก (พอร์ต 0) สัญญาณนี้จะแอดทีฟทุกๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล

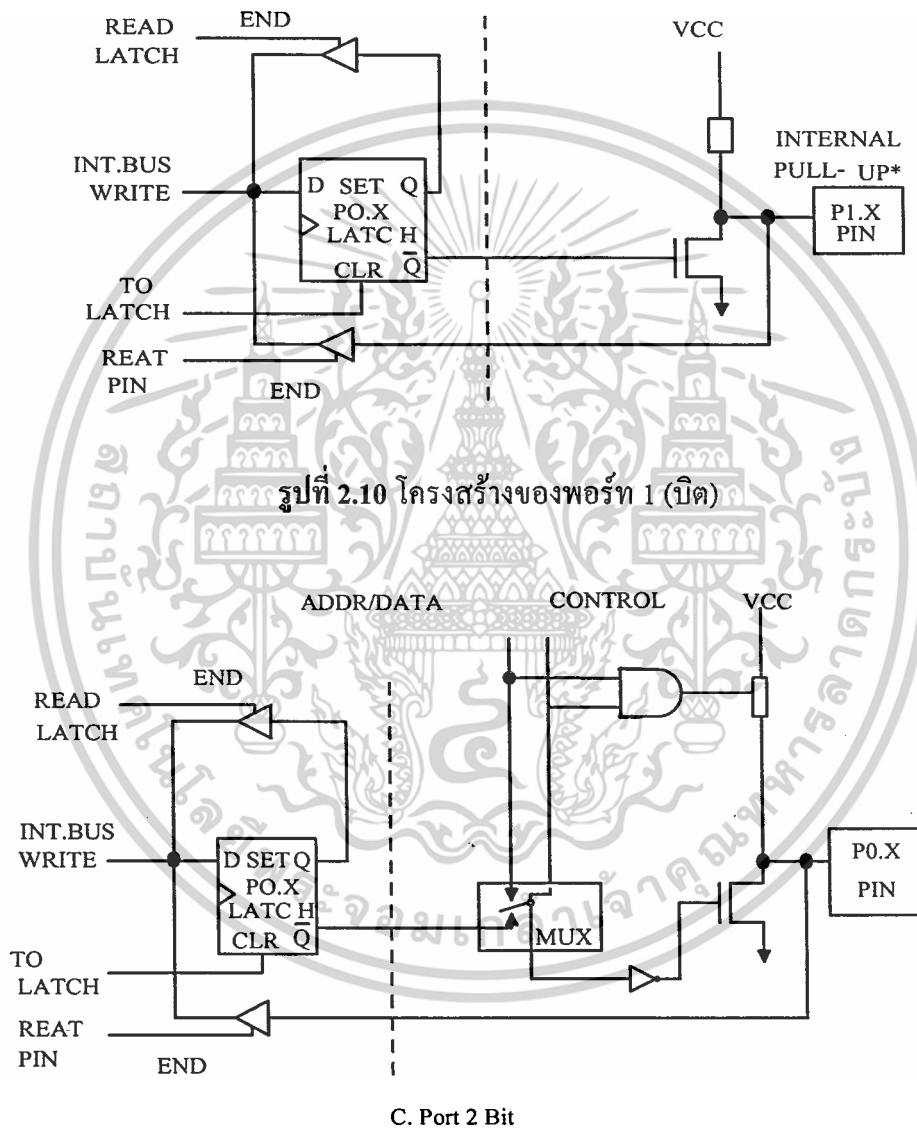
8. ขา 29 (PSEN) เป็นขาสโตรบที่ใช้สำหรับการอ่านข้อมูลจาก Program Memory ภายนอก สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขา

9. ขา 31 (EA) ใช้เลือกหน่วยความจำโปรแกรมภายนอก ป้อน “0” จะอ่านโปรแกรมจากภายนอกชิพ ป้อน “1” จะอ่านโปรแกรมจากภายในชิพ

10. ขา 9 (RST) ขารีเซ็ต (Reset) จะรีเซ็ตได้ก็ต่อเมื่อป้อนลอจิก “1” เข้าที่ขานี้นานอย่างน้อย 2 แมชชีนไซเคิล

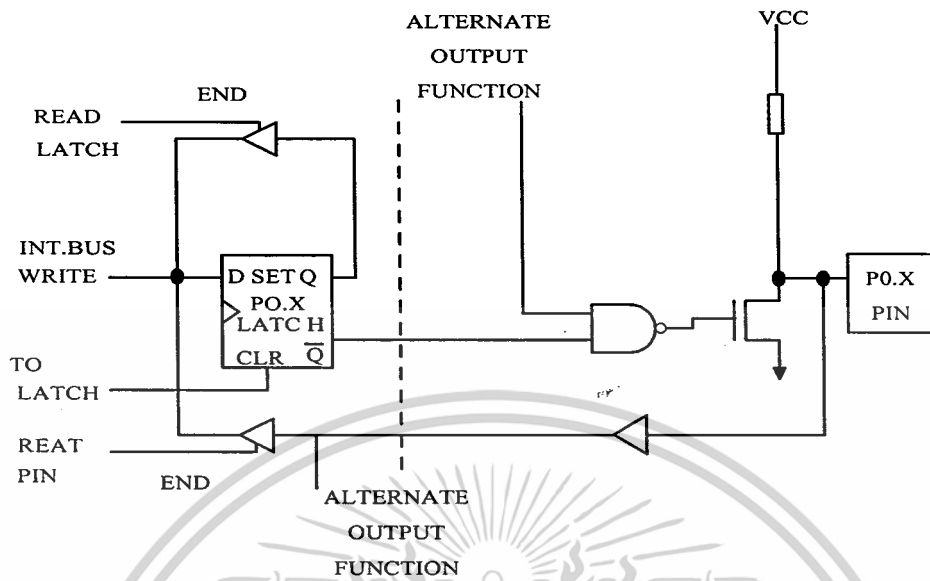
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. ขา 19 (XTAL1) ใช้ต่อคริสตัลภายนอกโดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์ภายใน
12. ขา 18 (XTAL2) ใช้ต่อคริสตัลภายนอกโดยเป็นเอาต์พุตของวงจรออสซิลเลเตอร์ภายใน



รูปที่ 2.11 โครงสร้างของ พอร์ต 2 (บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

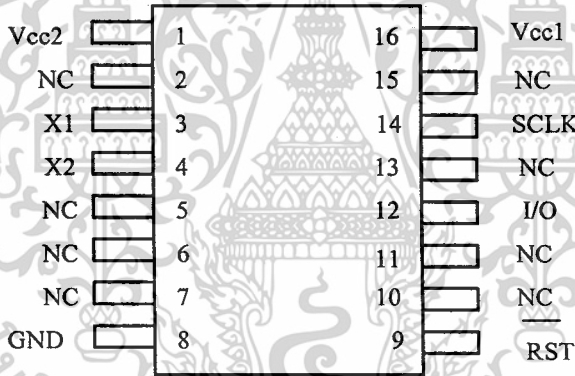
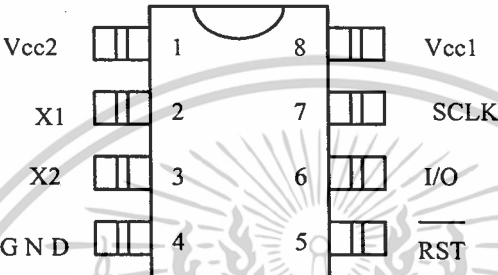
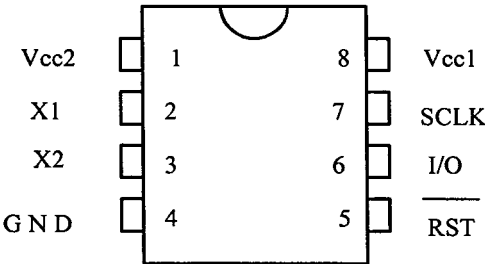


รูปที่ 2.12 โครงสร้างของพอร์ท 3 (บิต)

2.5 RTC

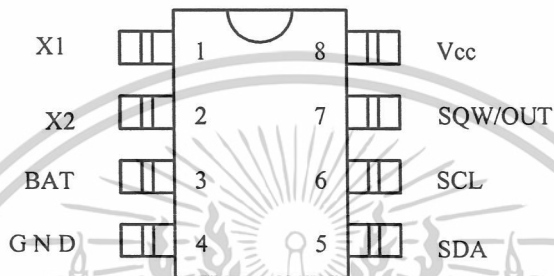
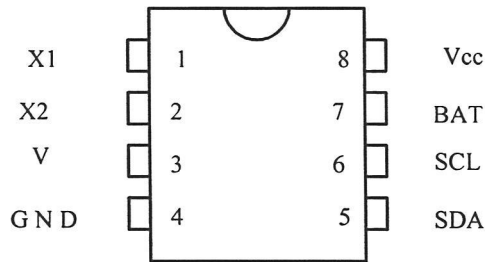
2.5.1 RTC หรือ Real Time Clock

เป็นชิพขนาด 8 ขา ที่ใช้เก็บเวลาได้ตั้งแต่ วินาที นาที ชั่วโมง วัน เดือน ปี และปีที่ เป็นปีอธิกสุรทิน (เดือนกุมภาพันธ์ ที่มี 28 หรือ 29 วัน) ซึ่งสามารถใช้งานได้ถึงปี ค.ศ. 2100 เมื่อโปรแกรม (ตั้งเวลา) ให้กับตัวชิพแล้วเวลาตัวชิพก็จะเดินต่อไปเรื่อยๆ ตราบใดที่ยังมีไฟเลี้ยงจ่ายให้กับตัวชิพอยู่ RTC ที่ใช้งานมีอยู่หลายเบอร์ด้วยกันเช่น DS1202, DS1302 และ DS1307 ซึ่งเป็นของบริษัท Dallas Semiconductor



รูปที่ 2.13 ชิพ RTC เบอร์ DS1302

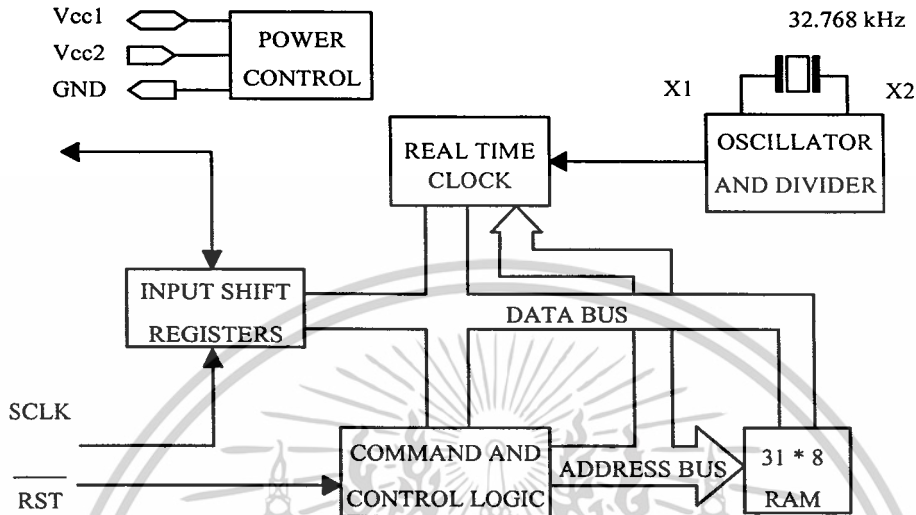
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ชิพ RTC เบอร์ DS1307

สำหรับชิพเบอร์ DS1202 และ DS1302 จะมีลักษณะคล้ายกันซึ่งมีขาที่เชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ 3 ขาด้วยกัน แตกต่างกันที่ขาที่สำหรับจ่ายไฟเลี้ยงคือ DS1202 จะไม่มีขาสำหรับต่อแบตเตอรี่สำรอง ถ้าต้องการให้มีแบตเตอรี่สำรองจะต้องต่ออุปกรณ์เสริมข้างนอก ซึ่งเบอร์ DS1302 จะสามารถต่อแบตเตอรี่สำรองเข้าที่ขา VCC1 และไฟเลี้ยงปกติเข้าที่ VCC2 ซึ่ง DS1202 และ DS1302 สามารถใช้แทนกันได้เพราะขา VCC1 ใน DS1302 ตรงกับขา VCC ใน DS1202 อยู่ที่ขา 8 ส่วนชิพ RTC เบอร์ DS1307 นั้นจะแตกต่างกันมากกว่า คือใช้การเชื่อมต่อกับไมโครคอนโทรลเลอร์แบบ I² Bus หรือใช้สายเพียง 2 เส้น แต่คุณสมบัติอื่นๆ ยังคงเหมือนกับเบอร์ DS1302

2.5.2 โครงสร้างและขาสัญญาณต่างๆ

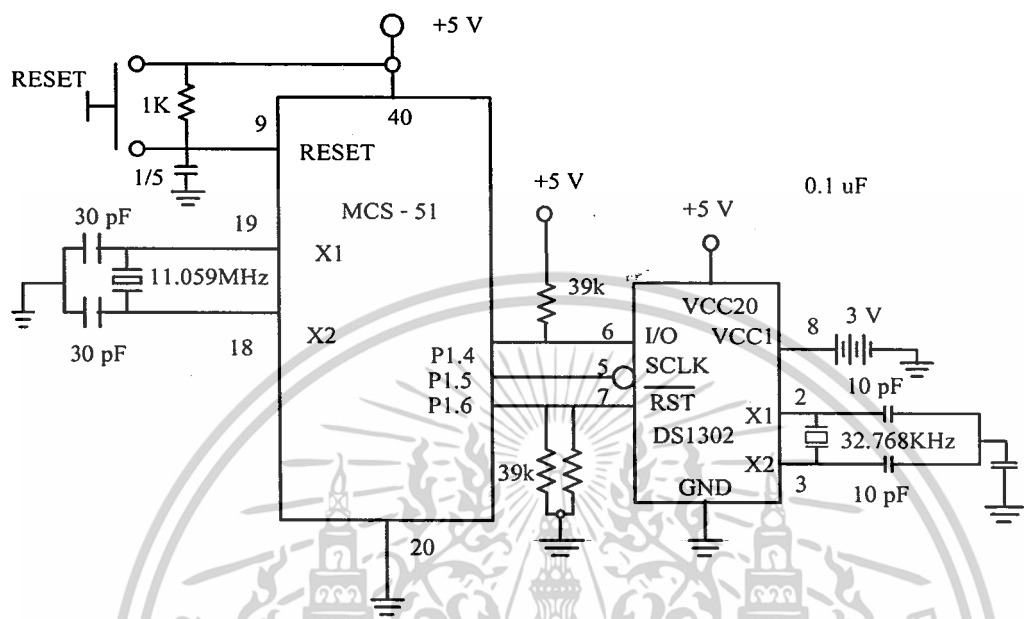


รูปที่ 2.15 โครงสร้างของชิพ RTC เบอร์ DS1302

- VCC1 : เป็นขาต่อแบตเตอรี่ Backup สำหรับเบอร์ DS1302 และ Trickle Charge สำหรับเบอร์ DS1302 เท่านั้น
- VCC2 : เป็นขาไฟเลี้ยงสำหรับเบอร์ DS1302 เท่านั้น (DS1202 ไม่ได้ใช้งานขานี้)
- I/O : เป็นขาสัญญาณข้อมูลที่ใช้ติดต่อกับไมโครคอนโทรลเลอร์
- SCLK : เป็นขาสัญญาณ Clock สำหรับอ่านหรือเขียนข้อมูลกับตัวชิพ RTC
- RST : เป็นขาสัญญาณเพื่อบอกกับตัวชิพ RTC เมื่อขานี้เป็น "1" ขา I/O ทำงาน ขานี้เป็น "0" ขา I/O อยู่ในสถานะ Tri-State
- X1-X2 : เป็นขาสำหรับต่อคริสตัล 32.768 กิโลเฮิร์ตซ์ (kHz) เพื่อเป็นสัญญาณนาฬิกาอ้างอิงของ RTC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 การเชื่อมต่อ RTC เข้ากับไมโครคอนโทรลเลอร์



รูปที่ 2.16 การเชื่อมต่อ DS1302 เข้ากับ MCS-51

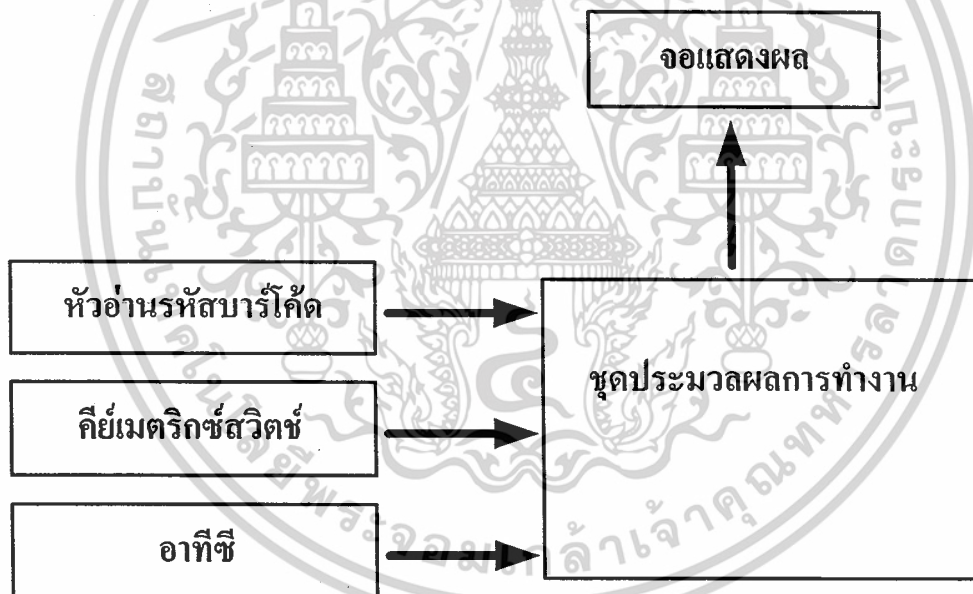
สำหรับการเชื่อมต่อ DS1302 เข้ากับตัว MCS-51 จะต่อเข้าทางพอร์ตไหนก็ได้ หรือต่อผ่าน 8255 ก็ได้ แต่ในที่นี้ต่อเข้าทาง พอร์ต 1 บิตที่ (4-6)

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

การออกแบบแบบจำลองระบบจอตกรประกอบด้วย 3 ส่วนคือ ส่วนของวงจรรับข้อมูล ส่วนประมวลผลและส่วนการแสดงผล ซึ่งส่วนของวงจรรับข้อมูลได้แก่ ส่วนรับข้อมูลหัวอ่านรหัสบาร์โค้ด วงจรรับข้อมูลจากคีย์เมตริกซ์สวิตช์ และ RTC ส่วนประมวลผลได้แก่ ไมโครคอนโทรลเลอร์ และส่วนแสดงผลซึ่งก็คือจอ LCD โดยการทำงานทั้งหมดได้แสดงโครงสร้างการทำงานของแบบจำลองระบบจอตกรในรูปที่ 3.1 ซึ่งมีการทำงานดังนี้



รูปที่ 3.1 โครงสร้างการทำงานของแบบจำลองระบบจอตกร

ส่วนของวงจรรับข้อมูลหัวอ่านรหัสบาร์โค้ด ส่วนของของวงจรรับข้อมูลจากคีย์เมตริกซ์สวิตช์ และอาทีซีจะต่ออยู่กับไมโครคอนโทรลเลอร์โดยไมโครคอนโทรลเลอร์จะทำหน้าที่เป็นศูนย์กลางของส่วนต่างๆ โดยส่วนป้อนข้อมูลจะทำการป้อนข้อมูลโดยเมื่อมีผู้ต้องการนำรหัสเข้าไปจอตกรในทีจอตกรจะนำบาร์โค้ดมาอ่านยังหัวอ่านรหัสบาร์โค้ดพนักงานทำการกรหัสป้ายทะเบียนรถเป็นเลขจำนวน 4 ตัวบนปุ่มกดรหัสจากนั้นไอซี 89C51 จะรับข้อมูลมาจากคีย์เมตริกซ์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิตช์เพื่อทำการประมวลผลโดยนำรหัสที่ได้ส่งไปเก็บยังบัตรบาร์โค้ดเมื่อผู้ใช้บริการต้องการนำรถออกนํ้าบัตรบาร์โค้ดมาให้พนักงานเพื่อให้พนักงานนำบัตรมาจุดผ่านชุดหัวอ่านบัตรจากนั้นไอซี 89C51 จะรับข้อมูลมาจากชุดหัวอ่าน และอาทิสเพื่อทำการประมวลผลแล้วแสดงผลออกที่ LCD ซึ่งในบทนี้จะกล่าวถึง การออกแบบ การสร้าง และการทำงานของระบบโดยแยกเป็นสอง ส่วนใหญ่ๆ คือ ส่วนวงจรและส่วนโปรแกรม

3.2 การออกแบบวงจร

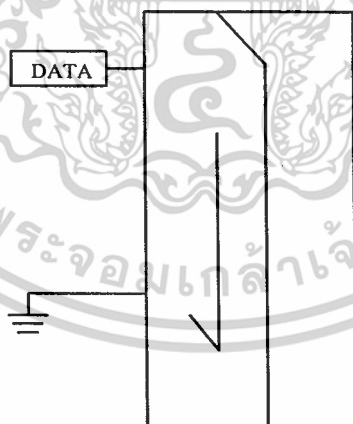
การออกแบบวงจรแบบจำลองระบบจอลครดจะแยกเป็นสองส่วนคือ ส่วนของวงจรรับข้อมูลและส่วนแสดงผล ซึ่งมีการทำงานดังนี้

3.2.1 ส่วนของวงจรรับข้อมูล

ในส่วนของวงจรรับข้อมูลประมวลผลทำหน้าที่ในการรับข้อมูลเพื่อทำการส่งข้อมูลให้แก่ไมโครคอนโทรลเลอร์เพื่อนำไปประมวลผล ซึ่งแบ่งได้เป็นสามส่วนคือ

1) ส่วนรับข้อมูลหัวอ่านรหัสบาร์โค้ด

ทำหน้าที่ในการรับข้อมูลจากการรูดบัตรรหัสบาร์โค้ด โดยผ่านชุดหัวอ่านรหัสบัตรบาร์โค้ด จากนั้นทำการส่งต่อให้แก่ไมโครคอนโทรลเลอร์เพื่อทำการประมวลผลต่อไป



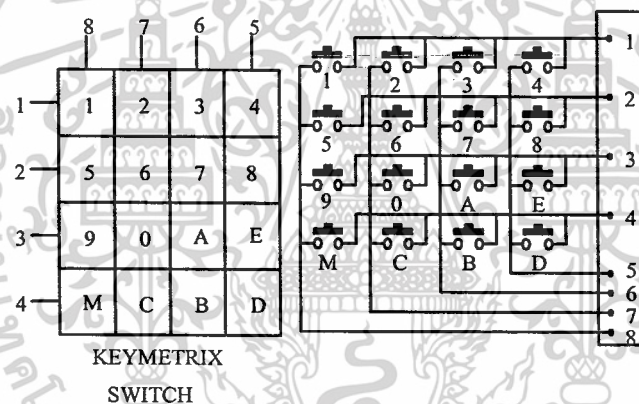
รูปที่ 3.2 ชุดหัวอ่านรหัสบาร์โค้ด

เมื่อมีผู้ต้องการที่จะนำรถเข้าไปจอดในที่จอดรถพนักงานจะทำการบันทึกทะเบียนรถ และทำการรูดรหัสบัตรบาร์โค้ด เพื่อทำการบันทึกรหัสบัตรบาร์โค้ด และทะเบียนรถลงในหน่วยความจำของไมโครคอนโทรลเลอร์ ซึ่งเมื่อต้องการนำรถออกจากที่จอดรถ ผู้จอดรถให้นำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัตรรหัสบาร์โค้ดที่ได้จากพนักงานในตอนเช้าจะส่งให้พนักงานทำการรูดรหัสบัตรบาร์โค้ด โดยรูดรหัสบัตรบาร์โค้ดผ่านชุดหัวอ่าน รหัสข้อมูลจากบัตรจะถูกส่งรหัสข้อมูลออกเส้นทางข้อมูล (Data Bus) สัญญาณข้อมูลที่ออกมาจากเอาต์พุตของชุดหัวอ่านรหัสบัตรบาร์โค้ดจะถูกส่งไปยังพอร์ตอินพุตของไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์จะทำการตรวจสอบการทำงานจากโปรแกรมต่อ เพื่อทำการประมวลผลจากนั้นส่งข้อมูลไปส่วนแสดงผล

2) วงจรรับข้อมูลคีย์เมตริกซ์สวิตช์

ในส่วนของวงจรรับข้อมูลคีย์เมตริกซ์สวิตช์จะทำหน้าที่ในการรับสัญญาณข้อมูลจากการกดข้อมูลของผู้ที่จะทำการจอร์คคือเลขทะเบียน โดยการกดเลขทะเบียนเมื่อสัญญาณข้อมูลที่ได้รับเข้ามานั้นจะเปลี่ยนเป็นสัญญาณรหัสฐานสอง 4 บิตเพื่อส่งไปยังพอร์ต 0 ของไอซี MCS-51 เพื่อนำไปประมวลผล



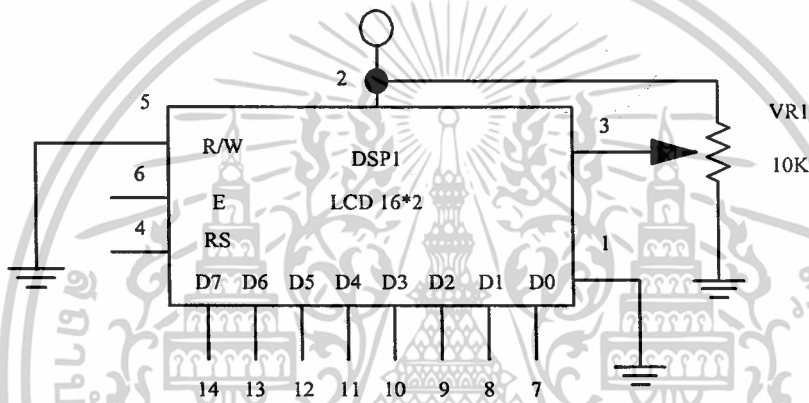
รูปที่ 3.3 รูปคีย์เมตริกซ์สวิตช์

คีย์เมตริกซ์สวิตช์มี 12 ปุ่มกด ประกอบด้วยตัวเลข 0-9 และอักษร A, B, C, D, E, M โดยแบ่งการเดินสายออกเป็นแวนอน และแวนดั่งของแต่ละปุ่มกดโดยการกดแต่ละครั้งจะได้เอาต์พุตทางแวนอน และแวนดั่งอย่างละหนึ่งค่าโดยเอาต์พุตจะให้รหัสไม่ซ้ำกัน 16 ค่าเมื่อผู้ใช้ทำการกดปุ่มใดปุ่มหนึ่งบนปุ่มกดรหัสข้อมูลรหัสจะถูกส่งไปให้อินพุตของ MCS-51 ซึ่งเป็นอุปกรณ์ประมวลผลของวงจรทั้งหมด และจะได้นำรหัสที่กดส่งไปแสดงผล

3.2.2 ส่วนแสดงผล

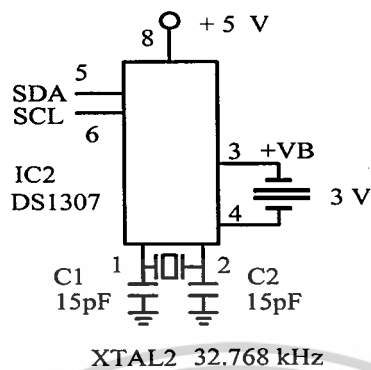
ในส่วนของการแสดงผลแสดงผล ซึ่งก็คือส่วนของจอ LCD โดยส่วนแสดงผลนี้จะรับข้อมูลจาก ชุดหัวอ่านรหัสบาร์โค้ด คีย์เมตริกซ์สวิตช์ และ RTC จะทำหน้าที่ในการแสดงผลที่ได้จากการประมวลผลของไมโครคอนโทรลเลอร์ จอ LCD จะมีขาทั้งสิ้น 14 ขา เมื่อได้รับข้อมูลมาจากเอกสารเป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์โดยที่ขา 4 (RS) จะเป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล และขา 5 (R/W) จะเป็น “1” จะเป็นการอ่านข้อมูล และขา 7 ถึงขา 14 (D0-D7) เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่างจอ LCD กับอุปกรณ์ภายนอกขนาด 8 บิต และขา RS ขา R/W และขา E จะใช้งานร่วมกัน ขา 1 (Vss) ต่อกราวด์ ขา 2 (Vdd) ต่อไฟเลี้ยง 5 โวลต์ให้ และยังสามารถปรับความเข้มได้โดยการต่อความต้านทานปรับค่าได้ 10 กิโลโอห์มเข้ากับขา 3 (Vo) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล ขา 6 (E) เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดูลจอ LCD ให้ทำงาน



รูปที่ 3.4 รูป LCD

RTC ทำหน้าที่สร้างฐานเวลาให้กับหน่วยประมวลผลโดย DS1307 จะมีโหมดการทำงาน 2 โหมดคือโหมดเขียนข้อมูล และโหมดอ่านข้อมูลในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน ส่วนโหมดการเขียนข้อมูลจะถูกใช้ก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่ และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไปโดย DS1307 จะติดต่อกับไมโครคอนโทรลเลอร์แบบ I²C โดย DS1307 มีทั้งหมด 8 ขาโดยขา 8 (Vcc) และกราวด์ (Gnd) ขา 4 ต่อกับไฟเลี้ยง +5 โวลต์ V_{BAT} ขา 3 ใช้ต่อกับแบตเตอรี่ 3 โวลต์ เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไปแม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ขา 5 (Sda) และขา 6 (Scl) เป็นขาที่เชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัสขา 7 (I²C Sqw Out) ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1 เฮิรตซ์ (Hz) 4.096 กิโลเฮิรตซ์ 8.192 กิโลเฮิรตซ์ และ 32 กิโลเฮิรตซ์ ในการใช้งานต้องต่อตัวต้านทาน 1 กิโลโอห์ม (kOhm) พูลอัพที่ขา 1 (X₁) และขา 2 (X₂) ใช้ต่อกับคริสตัลความถี่มาตรฐาน 32.768 กิโลเฮิรตซ์ เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่โดยไม่เสียค่าใช้จ่าย



รูปที่ 3.5 รูป RTC

3.3 ซอฟต์แวร์

ในส่วนนี้ของซอฟต์แวร์ จะเป็นการเขียนโปรแกรมเพื่อนำมาควบคุมการทำงานของแบบจำลองระบบจลนศาสตร์ โดยการใช้หลักการโดยแบบจำลองระบบจลนศาสตร์นี้จะใช้หลักการการทำงานของเครื่องอ่านบาร์โค้ด ซึ่งโปรแกรมหรือคำสั่งแสดงอยู่ในภาคผนวก

3.3.1 การรับข้อมูลมาประมวลผล

ในส่วนนี้ทำหน้าที่รับข้อมูลมาสองทางคือ การรับข้อมูลมาจากเครื่องอ่านบาร์โค้ด และคีย์เมตริกซ์สวิตช์เพื่อนำมาประมวลผลที่ไมโครคอนโทรลเลอร์ และส่งไปแสดงผลที่จอ LCD

3.3.2 การแสดงผล

ในส่วนนี้ทำหน้าที่เป็นตัวแสดงค่าที่ได้จากการทำการรู้ตัว และจากการป้อนรหัสข้อมูลจากคีย์เมตริกซ์สวิตช์มาแสดงที่จอ LCD โดยการเขียนโปรแกรมให้จอ LCD ทำการแสดงผลเพียงอย่างเดียว

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองส่วนรับข้อมูล

เพื่อให้ง่ายแก่การทดลอง และตรวจสอบการทำงานของระบบจึงได้แบ่งการทดลองออกเป็น 2 ส่วน คือ ส่วนแรกเป็นการทดลองส่วนรับข้อมูลจากบัตรบาร์โค้ด ส่วนรับข้อมูลจากคีย์เมตริกซ์สวิตช์ และส่วนที่สองเป็นการแสดงผลของแบบจำลองระบบจอร์ด

4.1.1 การทดลองส่วนรับข้อมูลจากบัตรบาร์โค้ด

1) ลำดับขั้นตอนการทดลอง

1. ทำการเตรียมแบบจำลองระบบจอร์ดให้พร้อมใช้งาน และต่อหัวอ่านบาร์โค้ดเข้ากับแบบจำลองระบบจอร์ด
2. ตรวจสอบความเรียบร้อย
3. จ่ายแรงดัน 5 โวลต์ ให้กับวงจร
4. นำบัตรบาร์โค้ดมาทำการรูดโดยรหัสที่ติดอยู่กับบัตรบาร์โค้ดคือ รหัส 113
5. เมื่อทำการรูดบัตรบาร์โค้ดสังเกตผลที่จอแอลซีดี



รูปที่ 4.1 บัตรบาร์โค้ด



รูปที่ 4.2 หน้าจอ LCD รหัสบาร์โค้ด

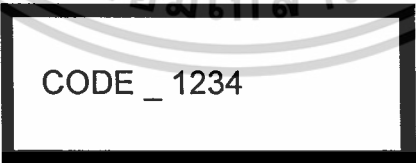
2) สรุปผลการทดลอง

จากการทดลองเมื่อเราทำการรูดบัตรรหัสบาร์โค้ดผ่านหัวอ่าน โดยที่รหัสที่ติดอยู่กับบัตรบาร์โค้ดคือ รหัส 113 และสังเกตผลที่จอแอลซีดี ปรากฏว่า จอแอลซีดีแสดงผลออกมาเป็น รหัส 113 เหมือนกับรหัสที่ติดอยู่กับบัตรบาร์โค้ด จึงสรุปได้ว่าชุดหัวอ่านบัตรบาร์โค้ดสามารถอ่านรหัสจากบัตรบาร์โค้ดได้ ดังนั้นจึงสามารถนำหัวอ่านบัตรบาร์โค้ดมาใช้งานในแบบจำลองระบบจอตลอดได้

4.1.2 การทดลองส่วนรับข้อมูลจากคีย์เมตริกซ์สวิตช์

1) ลำดับขั้นตอนการทดลอง

1. ทำการจัดเตรียมแบบจำลองระบบจอตลอดให้พร้อมใช้งาน พร้อมทั้งประกอบวงจรเข้ากับคีย์เมตริกซ์สวิตช์
2. ตรวจสอบความเรียบร้อย
3. จ่ายแรงดัน 5 โวลต์ให้กับวงจร
4. กดตัวเลขบนคีย์เมตริกซ์สวิตช์ ในการทดลองนี้เราจะกดเลข 1234
5. สังเกตผลจากจอแอลซีดี



รูปที่ 4.3 จอLCD รหัสเมื่อกดคีย์เมตริกสวิตช์

2) สรุปผลการทดลอง

จากการทดลองเมื่อเราทำการกดตัวเลขบนคีย์เมตริกซ์สวิตช์ ในที่นี้เราจะกดเลข 1234 เมื่อเราสังเกตผลที่หน้าจอแอลซีดี ปรากฏว่าจอแอลซีดีแสดงผลตัวเลข 1234 ตามที่เราได้ทำการคีย์ลงในคีย์เมตริกซ์สวิตช์ จึงสรุปได้ว่าคีย์เมตริกซ์สวิตช์สามารถทำการป้อนค่าเพื่อนำไปประมวลผล และแสดงที่หน้าจอแอลซีดีได้ตามที่เราต้องการ ดังนั้นจึงสามารถนำคีย์เมตริกซ์สวิตช์ มาใช้งานในแบบจำลองระบบจอร์นได้

4.2 การทดลองส่วนของการแสดงผล

ในส่วนที่สองของการทดลองจอ LCD จะเป็นการทดลองการแสดงผลของส่วนของวงจรรับข้อมูลจากไมโครคอนโทรลเลอร์ โดยในการทดลองนี้เราจะทำการทดลอง RTC ไปด้วยเนื่องจากจะใช้ร่วมกับไมโครคอนโทรลเลอร์ในการควบคุมให้จอ LCD แสดงผลการทดลองการแสดงผลส่วนรับข้อมูลจากไมโครคอนโทรลเลอร์

1) ลำดับขั้นตอนการทดลอง

1. ทำการเตรียมแบบจำลองระบบจอร์นที่เสร็จสมบูรณ์ให้พร้อมใช้งาน ซึ่งมีวงจรดังรูปที่ 4.3
2. ตรวจสอบความเรียบร้อย
3. จ่ายแรงดัน 5 โวลต์ ให้กับวงจร
4. สังเกตผลที่หน้าจอแอลซีดี



รูปที่ 4.4 รูปผลการทดลองแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ผลการทดลอง

จากการทดลอง เมื่อจ่ายแรงดัน 5 โวลต์ ให้กับแบบจำลองระบบจอตกรล และในที่นี้เราได้ทำการทดลองอาร์ทีซีไปด้วย ดังนั้นการแสดงผลเราจะแสดงผลออกมาเป็นนาฬิกา คือ บอกเวลาเป็นชั่วโมง และนาที จึงสรุปได้ว่าในส่วนของการประมวลผล และอาร์ทีซีสามารถนำมาใช้ร่วมกันได้ในการใช้งานในแบบจำลองระบบจอตกรล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป

5.1 สรุป

แบบจำลองระบบจอตกรประกอบด้วย 2 ส่วนคือ ส่วนรับข้อมูลประมวลผล และส่วนการแสดงผล ซึ่งจะเป็นการจำลองระบบจอตกร ในแบบจำลองนี้มีบาร์โค้ด คีย์เมตริกซ์สวิตช์ และ RTC เป็นตัวส่งข้อมูลให้ไมโครคอนโทรลเลอร์ประมวลผล และส่งให้ LCD ทำการแสดงผล แบบจำลองนี้จะช่วยให้พนักงานตรวจสอบรถที่นำมาจอดให้ตรวจสอบได้ง่ายขึ้น และมีความปลอดภัยต่อการโจรกรรมได้เหมาะกับบริษัทสำนักงานห้างต่างๆ ที่มีการจอตกรกันมากสามารถนำไปประยุกต์ใช้งานได้จริง

5.2 ปัญหาและแนวทางแก้ไข

จากการดำเนินการสร้าง และทดสอบโครงงานพบว่ามีปัญหาเกิดขึ้นหลายประการซึ่งสรุปได้ดังนี้

1. ปัญหา การหาอุปกรณ์บัตรแถบแม่เหล็กมีความยุ่งยากในเรื่องการทำบัตรแถบแม่เหล็กเนื่องจากมีร้านขายตัวเครื่องแต่ไม่รับทำตัวบัตร ซึ่งการทำบัตรมีราคาแพง และต้องสั่งทำเป็นจำนวนมาก

แนวทางแก้ไข หาอุปกรณ์ที่มีหลักการทำงานเหมือนกัน และสามารถใช้แทนกันได้โดยที่จิตความสามารถการทำงานเหมือนเดิม โดยเปลี่ยนเป็นบาร์โค้ดเนื่องจากการสั่งทำตัวบัตรง่ายมากสามารถสั่งทำจำนวนมากได้เพราะราคาถูก

2. ปัญหา การออกแบบวงจรต่างๆ ล่าช้าเนื่องจากไม่มีข้อมูลเกี่ยวกับการใช้งานของบาร์โค้ดที่เป็นหัวใจสำคัญของงาน

แนวทางแก้ไข ทำการศึกษาค้นหาข้อมูลจากอินเทอร์เน็ต และแหล่งข้อมูลหลายๆ แหล่ง

3. ปัญหา การออกแบบลายวงจรเกิดการผิดพลาดทำให้วงจรไม่ทำงานตามที่ต้องการ

แนวทางแก้ไข ทำการตรวจเช็คลายวงจร และทำการแก้ไขเพิ่มเติมส่วนที่ผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนา

1. ในการบันทึกทะเบียนรถเนื่องจากมีแค่ตัวเลข 4 หลักเมื่อเกิดมีทะเบียนรถที่มีตัวเลขซ้ำกันอาจเกิดการผิดพลาดในการทำงานได้ จึงควรพัฒนาให้สามารถบันทึกตัวอักษรของทะเบียนรถได้
2. เมื่อเกิดไฟดับข้อมูลที่ได้บันทึกไว้จะสูญหายเนื่องจากเป็นข้อมูลที่บันทึกไว้ในหน่วยความจำของไมโครคอนโทรลเลอร์ จึงควรพัฒนาให้ใช้หน่วยความจำภายนอกเป็นตัวบันทึกข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

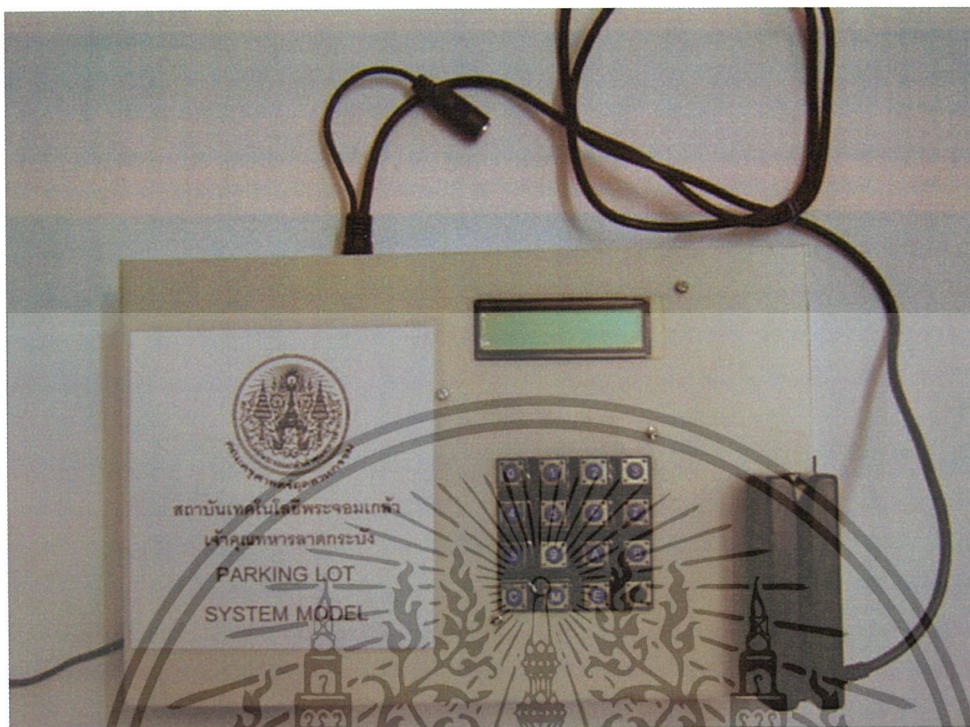
วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตรวิไล. เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์
 MCS-51. กรุงเทพฯ : อินโนเวติฟ เอ็ดดูเคชัน. 2546
 สมยศ จุณณะปิยะ. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51. พิมพ์ครั้งที่ 5. กรุงเทพฯ :
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2546



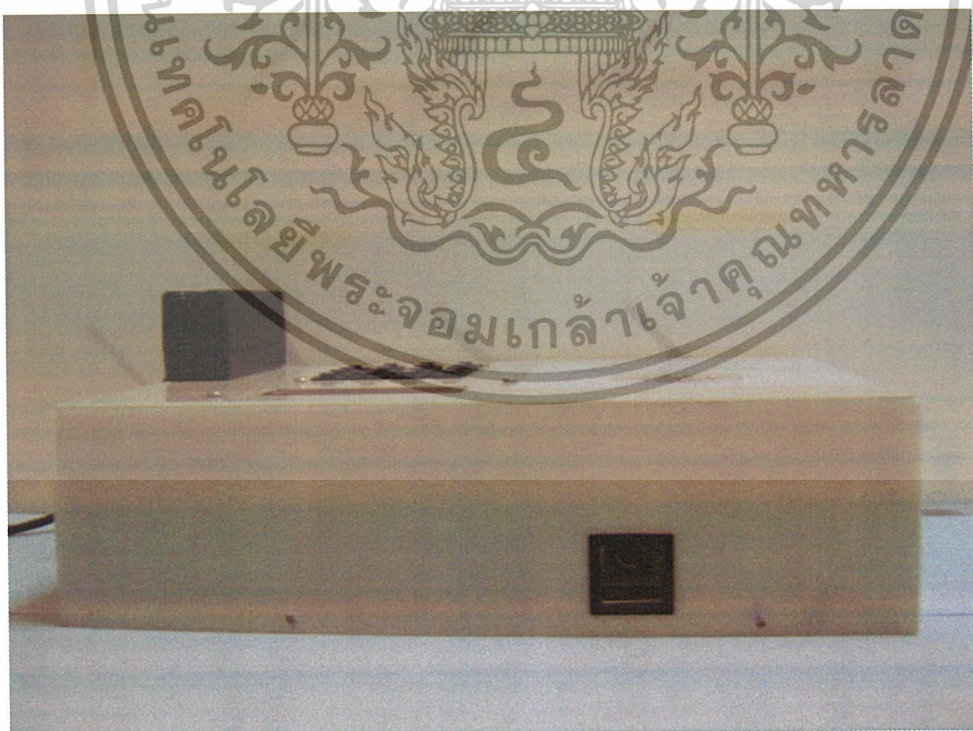
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 ภาพด้านหน้าของแบบจำลองระบบจอดรถ

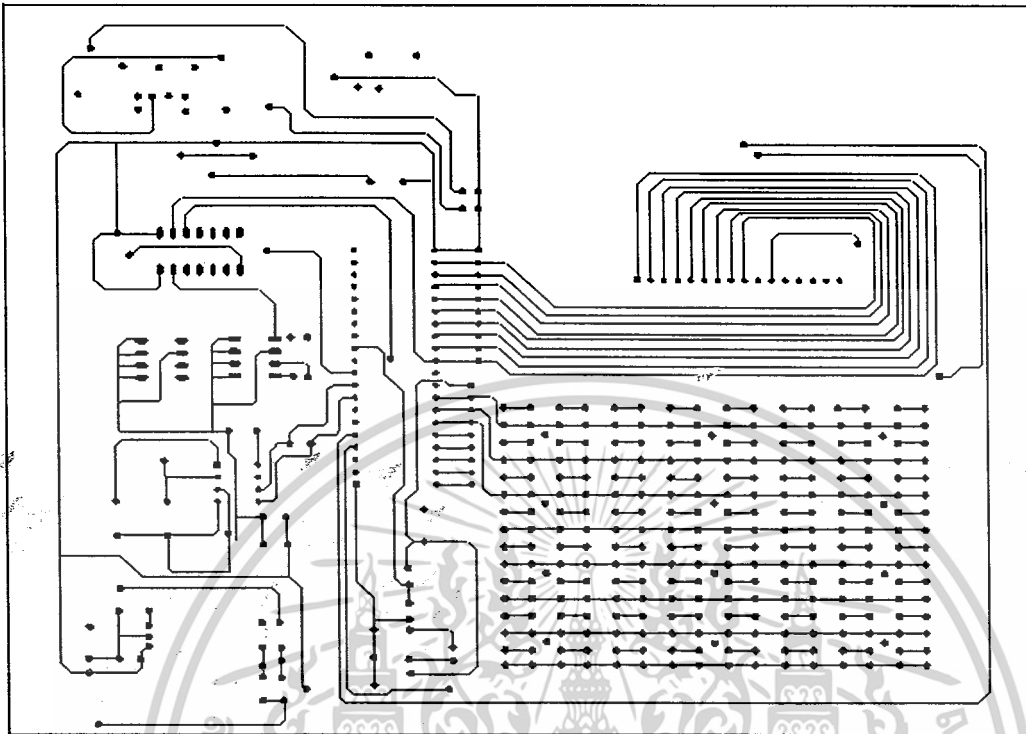


รูปที่ ก.2 ภาพด้านข้างของแบบจำลองระบบจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

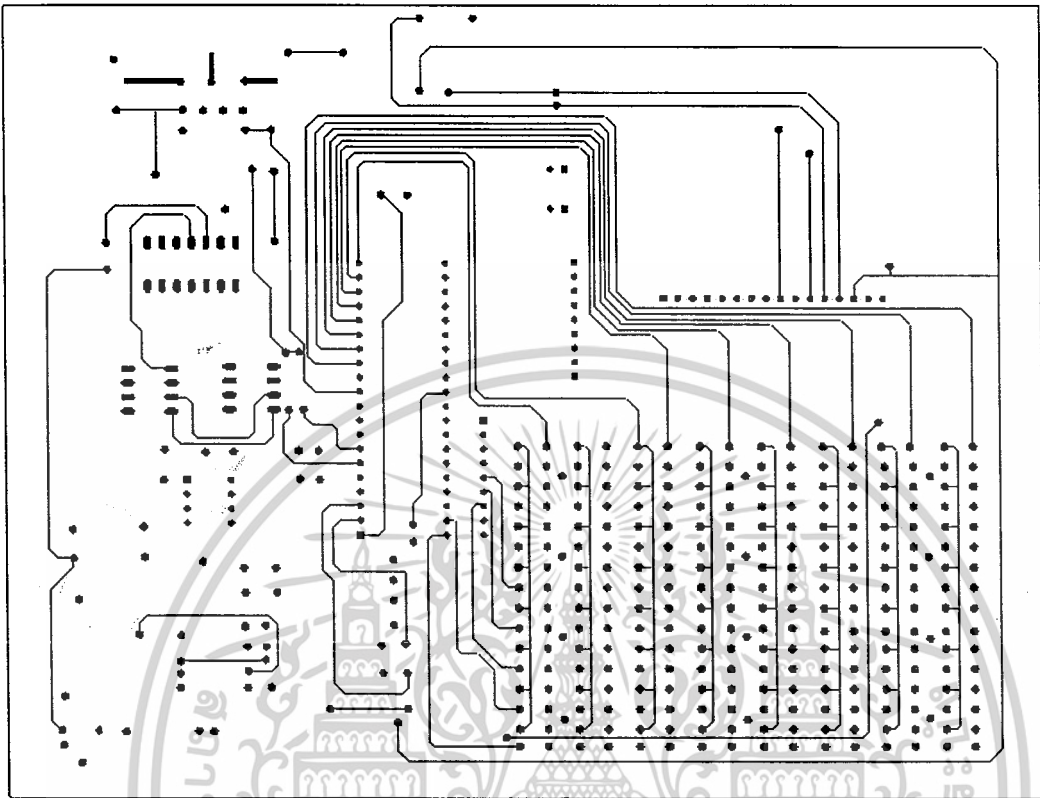


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 แผงวงจรพิมพ์แบบจำลองระบบจอกรถด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 แผ่นวงจรพิมพ์แบบจำลองระบบขอรถด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ค.1 รายการอุปกรณ์ของวงจรประมวลผล

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
ICL	MCS-51	1 ตัว
ตัวเก็บประจุ		
C1 , C2	33 pf	2 ตัว
C3	1 μ F 50 V	1 ตัว
C4	0.1 μ F 50 V	1 ตัว
ตัวความต้านทาน		
R1	10k 1/4 W	1 ตัว
อุปกรณ์อื่นๆ		
S1	สวิทช์	1 ตัว
X1	คริสตอล 11.0592	1 ตัว

ตาราง ค.2 รายการอุปกรณ์ของวงจรซีเอ็มเทริกซ์สวิทช์

ชื่ออุปกรณ์	รายละเอียด	จำนวน
ตัวเก็บประจุ		
C1	0.1 μ F 50 V	1 ตัว
ตัวความต้านทาน		
R1- R8	1 k Ω 1/4 W	8 ตัว
อุปกรณ์อื่นๆ		
S1 – S6	สวิทช์	64 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ค.3 รายการอุปกรณ์ของวงจรเวลา

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
ICL	DS 1307	1 ตัว
ตัวเก็บประจุ		
C1 , C2	15 pf	2 ตัว
C3	1 µF 50 V	1 ตัว
ตัวความต้านทาน		
R1 , R2	4.7 kΩ 1/4 W	2 ตัว
อุปกรณ์อื่นๆ		
E1	แบตเตอรี่ 3 V	1 ตัว
X1	คริสตอล 32.768 kHz	1 ตัว

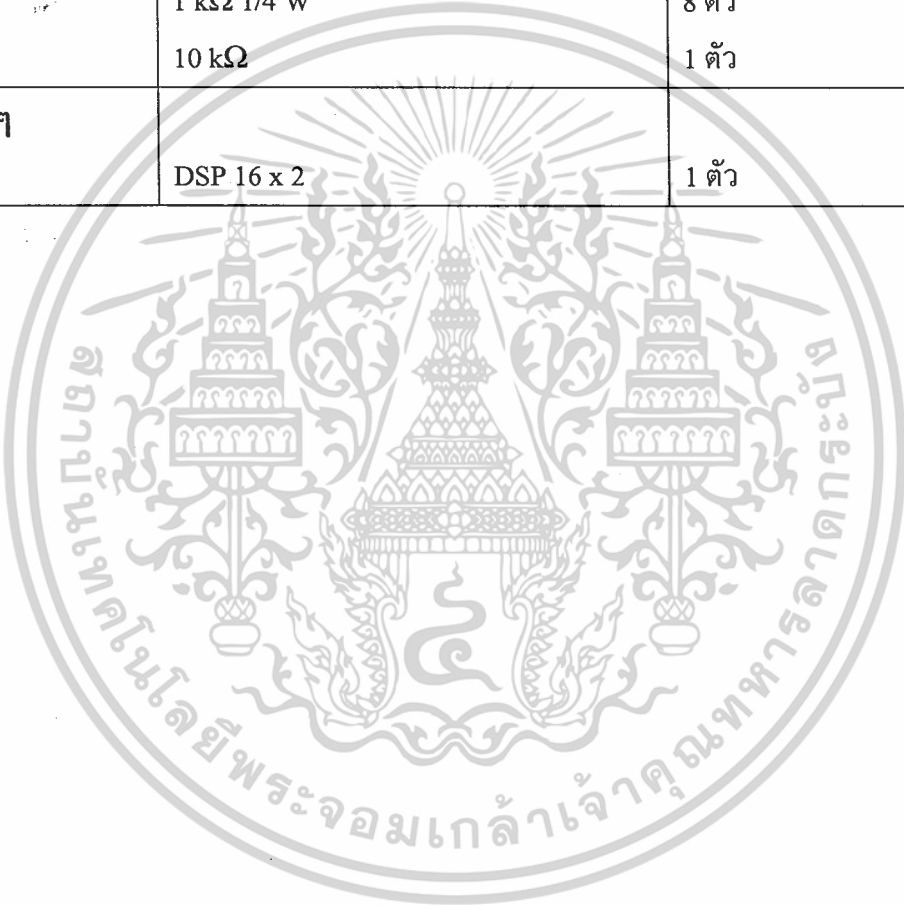
ตาราง ค.4 รายการอุปกรณ์ของวงจรแหล่งจ่ายแรงดัน

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	L700 50V	1 ตัว
อุปกรณ์กึ่งตัวนำ		
D1 – D4	IN 5401	4 ตัว
ตัวเก็บประจุ		
C1	2200 µF 50V	1 ตัว
C2	10p	1 ตัว
C3	10 µF 50 V	1 ตัว
อุปกรณ์อื่นๆ		
T1	หม้อแปลง 220 V 500 MA	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง ค.5 รายการอุปกรณ์ของวงจรแสดงผล

ชื่ออุปกรณ์	รายละเอียด	จำนวน
ตัวเก็บประจุ C1	0.1 μ F 50 V	1 ตัว
ตัวความต้านทาน R1 – R8 VR1	1 k Ω 1/4 W 10 k Ω	8 ตัว 1 ตัว
อุปกรณ์อื่นๆ LCD 1	DSP 16 x 2	1 ตัว



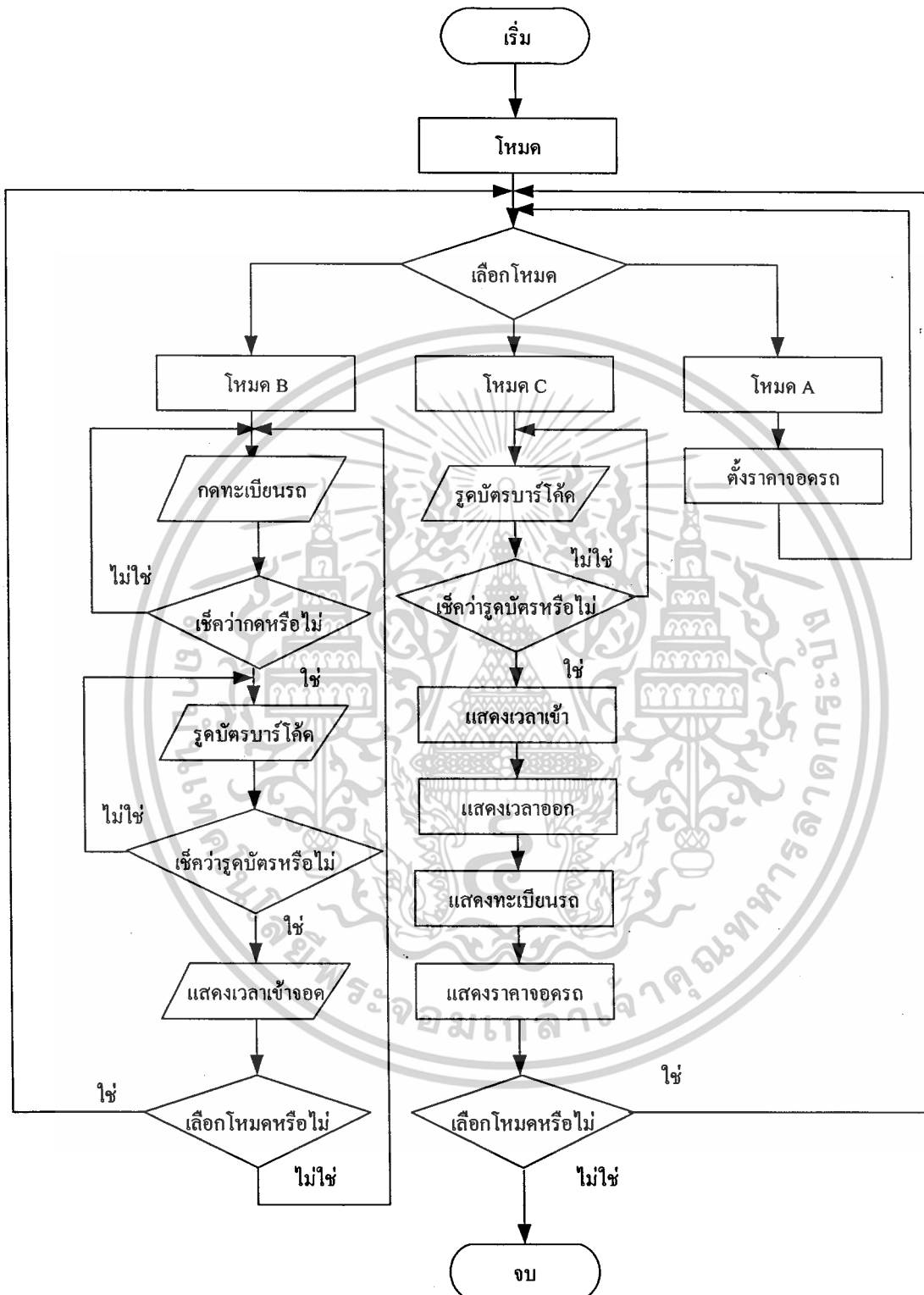
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ง

แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.1 ฟังก์ชันโปรแกรมของแบบจำลองระบบจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของแบบจำลองระบบจอตกรด

```
=====
;
=====PROGRAM BARCODE=====
;
;
;
;
;
;
;
;
;
;=====
```

```
LCD_ADDR EQU 030H ; For keep LCD Address
LCD_DATA EQU 031H ; For keep LCD Data
LCD_PTR EQU 032H ; For keep LCD 3 Char Pointer
LCD_ENBIT P3.6 ; LCD Module Enable (Active High : Level)
LCD_RS BIT P3.7 ; LCD Module Register Select

KEY_BUF EQU 7BH ;BUFFER KEY SCII CODE FROM KEYPRESS
KEEP_KEY EQU 7CH ;KEEP KEY SCII CODE 7CH - 7FH
BATH_HR EQU 39H ;KEEP BATH PER HOUR

;-----
SDA BIT P3.3 ; SDA I2C Bus
SCL BIT P3.2 ; SCL I2C Bus
I2C_ADDR EQU 033H ; For keep I2C Address
I2C_DATA EQU 034H ; For keep I2C Data
FLAG EQU 02FH ; User FLAG
I2C_ACK BIT FLAG.0 ; Define I2C Acknowledge as bit

SECONDS EQU 040H ; For keep Seconds
MINUTES EQU 041H ; For keep Minutes
HOURS EQU 042H ; For keep Hours
DAY EQU 043H ; For keep Day
DATE EQU 044H ; For keep Date
MONTH EQU 045H ; For keep Month
YEAR EQU 046H ; For keep Year
CONTROL EQU 047H ; For keep Control Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTC_ID	EQU	11010000B	; RTC Slave Address	
BAR_BUF	EQU	078H	; For Keep	Scan Key
KEEP_BAR	EQU	079H	; For Keep	CODE KEY
DATA_BAR	EQU	060H	; For Keep 60H TO 70H DATA BARCODE	
BATH	EQU	061H	;KEEP BATH/HR	
BAR_1	EQU	071H	;KEEP CODE BARCODE	
BAR_2	EQU	072H		
BAR_3	EQU	073H		
IRAM_BUF_1	EQU	030H		
IRAM_BUF_2	EQU	031H		
IRAM_BUF_3	EQU	032H		
DPH_BUF	EQU	07AH		
DPL_BUF	EQU	07BH		
NUM_REC	EQU	035H		
;DPH	EQU	83H		
;DPL	EQU	82H		
RESULT_SEC	EQU	036H		
MIN_BUF	EQU	037H		
HOURS_BUF	EQU	038H		
MUL_BUF	EQU	07EH		
ADD_BATH	EQU	07FH	;KEEP ADD BATH	
RE_S	EQU	48H	;KEEP BCD	
RE_F	EQU	49	;KEEP DIGIT FORWARD	
;-----				
ORG 0000H				
MOV BATH_HR, #10H				
MAIN: LCALL INIT_LCD ; Call LCD Initial subroutine				
LCALL LCD_CLR				

เอกสารนี้เป็นเอกสารเพื่อการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV  DPTR, #0000H
MOV  DPH_BUF, #01H
MOV  DPL_BUF, #00H
;MOV  BATH_HR, #10H      ;KEEP NUM BATH DEF
MOV  PSW, #18H
MOV  R1, #01H
MOV  R2, #00H
MOV  PSW, #00H

```

```

;-----CHECK MEN-----

```

```

CK_MENU:  LCALL LCD_CLR
          LCALL MENU
          MOV  A, KEY_BUF
          CJNE A, #65, MODE_OUT ;65 = 'A' EXCHANG
          LCALL EX_CH
          ;MOV  LCD_ADDR, #000H
          ;LCALL SET_ADDR_LCD

          ;MOV  A, BATH_HR
          ;MOV  LCD_DATA, A
          ;LCALL BCD2LCD

          ;LJMP $

```

```

;LCALL LCD_CLR
;LCALL WRCHAR_LCD
MOV  LCD_ADDR, #040H
LCALL SET_ADDR_LCD
;LCALL KEY
MOV  A, #'H'
MOV  LCD_DATA, A
LCALL WRCHAR_LCD
MOV  LCD_ADDR, #041H
LCALL SET_ADDR_LCD

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;LCALL KEY
MOV A,#61
MOV LCD_DATA,A
LCALL WRCHAR_LCD

MOV LCD_ADDR,#042H
LCALL SET_ADDR_LCD
;LCALL KEY
MOV A,KEY_BUF
MOV LCD_DATA,A
LCALL WRCHAR_LCD
MOV LCD_ADDR,#05H
LCALL SET_ADDR_LCD
MOV LCD_DATA,#95 ;95 = '/'
LCALL WRCHAR_LCD

LCALL DELAY_1s
;LCALL LCD_CLR

;LJMP CK_MENU

MODE_OUT: CJNE A,#67,MODE_IN ;67 = 'C' OUT MODE
           LJMP OUT_M

MODE_IN: CJNE A,#66,MODE_E ;66 = 'B' IN MODE
          LJMP IN_M

MODE_E:

           LJMP MAIN

;-----END MAIN-----

;-----MENU-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     LCD_ADDR, #00H
LCALL   SET_ADDR_LCD
MOV     LCD_DATA, #'M' ;DEF SCII TO LCD DATA
LCALL   WRCHAR_LCD
MOV     LCD_ADDR, #01H
LCALL   SET_ADDR_LCD
MOV     LCD_DATA, #'O' ;DEF SCII TO LCD DATA
LCALL   WRCHAR_LCD
MOV     LCD_ADDR, #02H
LCALL   SET_ADDR_LCD
MOV     LCD_DATA, #'D' ;DEF SCII TO LCD DATA
LCALL   WRCHAR_LCD
MOV     LCD_ADDR, #03H
LCALL   SET_ADDR_LCD
MOV     LCD_DATA, #'E' ;DEF SCII TO LCD DATA
LCALL   WRCHAR_LCD
MOV     LCD_ADDR, #040H
LCALL   SET_ADDR_LCD
MOV     LCD_ADDR, #04H
LCALL   SET_ADDR_LCD
MOV     LCD_DATA, #058 ;DEF SCII TO LCD DATA
LCALL   WRCHAR_LCD
MOV     LCD_ADDR, #05H
LCALL   SET_ADDR_LCD

```

```

LCALL   KEY
MOV     A, KEY_BUF
MOV     LCD_DATA, A ;DEF SCII TO LCD DATA
LCALL   WRCHAR_LCD
LCALL   DELAY_1s
;LCALL DELAY_1s
RET

```

;-----FUNCTION EXCHANG BATH PER HOUR-----

EX_CH: MOV LCD_ADDR, #040H

```

LCALL SET_ADDR_LCD
;LCALL KEY
MOV A,#'E'
MOV LCD_DATA,A ;DEF SCII TO LCD DATA
LCALL WRCHAR_LCD
MOV LCD_ADDR,#041H
LCALL SET_ADDR_LCD
MOV LCD_DATA,#58 ;DEF SCII TO LCD DATA
LCALL WRCHAR_LCD
MOV LCD_ADDR,#042H
LCALL SET_ADDR_LCD
LCALL KEY
MOV A,KEY_BUF
MOV LCD_DATA,A
LCALL WRCHAR_LCD
MOV A,KEY_BUF
CJNE A,#49,GET_2
MOV BATH_HR,#010H
RET

GET_2: CJNE A,#50,GET_3 ;50 = KEYPESS NUMBER 5
MOV BATH_HR,#020H ;BATH PER HOUR
RET
;LJMP IN_M

GET_3: CJNE A,#51,GET_4
MOV BATH_HR,#030H
RET

GET_4: CJNE A,#52,GET_5
MOV BATH_HR,#040H
RET

GET_5: CJNE A,#53,GET_6
MOV BATH_HR,#050H

```

RET

GET_6:

;LCALL DELAY_1s

LJMP MAIN

;-----OUT MODE FUNCTION-----

OUT_M1: ;MOV A,BATH_HR

;MOV LCD_DATA,A

;LCALL BCD2LCD

;LJMP \$

;

MOV LCD_ADDR,#000H

LCALL SET_ADDR_LCD

LCALL KEY

MOV A,KEY_BUF

MOV LCD_DATA,A

LCALL WRCHAR_LCD

MOV A,KEY_BUF

CJNE A,#49,GET_22

MOV BATH_HR,#010H

LJMP OUT_M1

GET_22: CJNE A,#50,GET_33 ;50 = KEYPESS NUMBER 5

MOV BATH_HR,#020H ;BATH PER HOUR

LJMP OUT_M1

GET_33: CJNE A,#51,GET_44

MOV BATH_HR,#030H

LJMP OUT_M1

GET_44: CJNE A,#52,GET_55

```
MOV BATH_HR,#040H
```

```
LJMP OUT_M1
```

```
GET_55: CJNE A,#53,GET_66
```

```
MOV BATH_HR,#050H
```

```
GET_66:
```

```
OUT_M1:MOV BATH_HR,#20H
```

```
LCALL LCD_CLR
```

```
MOV LCD_ADDR,#00H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA,'#C' ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD
```

```
MOV LCD_ADDR,#01H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA,'#O' ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD
```

```
MOV LCD_ADDR,#02H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA,'#D' ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD
```

```
MOV LCD_ADDR,#03H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA,'#E' ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD
```

```
MOV LCD_ADDR,#04H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA,'#.' ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
LCALL SCAN_BAR
LCALL RD_INBAR
;LJMP OUT_M
;-----

LCALL RTC_RD      ; Read RTC
MOV  LCD_ADDR, #040H
LCALL SET_ADDR_LCD
MOV  A,HOURS      ; Get Hour
ANL  A,#00110000B  ; Get x10 Digit
JZ   NWRITE_TIME_HN ; Check x10 Digit = 0 ?
SWAP A            ; If not => Write to LCD
ADD  A,#030H      ; Convert to ASCII
AJMP HWRITE_TIME_HH ;
NWRITE_TIME_HN:MOV A,#' ' ; x10 = 0 then Write space
HWRITE_TIME_HH:MOV LCD_DATA,A ; Write to LCD
LCALL WRCHAR_LCD ;
MOV  LCD_ADDR, #041H
LCALL SET_ADDR_LCD
MOV  A,HOURS      ; Write x1 Digit to LCD
ANL  A,#00001111B ;
ADD  A,#030H      ; Convert to ASCII
MOV  LCD_DATA,A ;
LCALL WRCHAR_LCD ;
MOV  LCD_ADDR, #043H
LCALL SET_ADDR_LCD
MOV  LCD_DATA,#':' ; Odd => Write ':'
LCALL WRCHAR_LCD ;
LJMP SWRITE_MINUTES ;
EWRITE_SPACE:;MOV LCD_DATA,#' ' ;
;ACALL WRCHAR_LCD ;
SWRITE_MINUTES:MOV LCD_DATA,MINUTES ; Write Minutes to LCD
LCALL BCD2LCD      ; Write from RTC BCD Data
LCALL DELAY_1s

```

MATH:

```

LCALL RTC_RD      ;HOURS KEEP HOUR NEW
MOV  A, MINUTES   ;MINUTES KEEP MIN NEW
MOV  PSW, #18H
SUBB A, R6        ;R6 KEEP MINUTES OLD
MOV  PSW, #00H
CJNE A, #00H, BATH_ADD ;VAL IN ACC IS MIN CHECK
LJMP BATH_0

```

```

BATH_0:  MOV  LCD_ADDR, #000H
         LCALL SET_ADDR_LCD
         MOV  A, HOURS      ;HOURS KEEP TIME NEW
         MOV  PSW, #18H
         SUBB A, R5        ;R5 KEEP TIME OLD
         MOV  PSW, #00H
         MOV  MUL_BUF, A
         ;MOV  A, #04H      ;TEST HOURS AT CAR KEEP MAX 09H/10H
         CJNE A, #00H, NXT_NUM2 ;IF A = 00H CHECK HOURS
         MOV  PSW, #18H
         MOV  R7, BATH_HR
         MOV  B, #00H
         MOV  PSW, #00H
         LJMP SHO_DATA

```

```

NXT_NUM2: ;MOV  A, MUL_BUF
          MOV  MUL_BUF, A
          MOV  LCD_DATA, A
          LCALL BCD2LCD
          LCALL DELAY_1s
          MOV  LCD_ADDR, #004H
          LCALL SET_ADDR_LCD
          MOV  A, BATH_HR

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับโรงเรียนที่จัดทำขึ้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV  ADD_BATH, A      ;KEEP RESULT ADD BATH
MOV  LCD_DATA, A
LCALL BCD2LCD
MOV  LCD_ADDR, #008H
LCALL SET_ADDR_LCD
;MOV  MUL_BUF, #02H

MOV  B, ADD_BATH      ;BATH KEEP BATH PER HOUR
MOV  A, MUL_BUF
MUL  AB
MOV  PSW, #18H
MOV  R7, A
MOV  PSW, #00H
LJMP SHO_DATA

```

BATH_ADD:

```

;MOV  LCD_DATA, A      ;CHECK VALUE IN ACC
;LCALL BCD2LCD
MOV  LCD_ADDR, #000H
LCALL SET_ADDR_LCD
MOV  A, HOURS         ;HOURS KEEP TIME NEW
MOV  PSW, #18H
SUBB A, R5             ;R5 KEEP TIME OLD
MOV  PSW, #00H
MOV  MUL_BUF, A
;MOV  A, #04H ;TEST HOURS AT CAR KEEP MAX 04H/10H
CJNE A, #00H, NXT_NUM ;IF A = 00H CHECK HOURS
MOV  PSW, #18H
MOV  R7, BATH_HR
MOV  B, #00H
MOV  PSW, #00H

```

LJMP SHO_DATA

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NXT_NUM:    ;MOV  A, MUL_BUF
            MOV  MUL_BUF, A
            MOV  LCD_DATA, A

            LCALL BCD2LCD
            LCALL DELAY_1s
            MOV  LCD_ADDR, #004H
            LCALL SET_ADDR_LCD
            MOV  A, BATH_HR
            ADD  A, BATH_HR    ;BATH DEF = 10H
            MOV  ADD_BATH, A    ;KEEP RESULT ADD BATH
            MOV  LCD_DATA, A
            LCALL BCD2LCD
            MOV  LCD_ADDR, #008H
            LCALL SET_ADDR_LCD
            ;MOV  MUL_BUF, #02H
            MOV  B, ADD_BATH    ;BATH KEEP BATH PER HOUR
            MOV  A, MUL_BUF
            MUL  AB
            MOV  PSW, #18H
            MOV  R7, A
            MOV  PSW, #00H

```

```

SHO_DATA:  ;MOV  DPTR, #D_BASE
            ;MOV  A, B
            ;MOVC A, @A + DPTR
            ;MOV  B, A
            MOV  LCD_DATA, B
            LCALL BCD2LCD
            MOV  LCD_ADDR, #00AH
            LCALL SET_ADDR_LCD
            MOV  PSW, #18H

```

```
MOV  A, R7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV PSW, #00H
;MOV DPTR, #D_BASE
;MOVC A, @A + DPTR
MOV LCD_DATA, A
LCALL BCD2LCD

```

OUT_MATH:

```

CLR P1.0
SETB P2.4
JB P2.4, $ ;PESS 'M' TIME 1
LCALL DELAY_10ms
JNB P2.4, $
LCALL DELAY_1s
JB P2.4, CONTINU ;PESS 'M' TIME 2
LCALL DELAY_10ms
JNB P2.4, $
LJMP CK_MENU
CONTINU:
LJMP OUT_M

```

CONTINU:

```

;-----OUT INTERNAL MEMORY-----
;-----FUNCTION GET BARCODE-----

```

```

IN_M: MOV NUM_REC, #20H
      MOV R4, #050 ;CONTROL RECORD

```

```

IN_M_REC: MOV LCD_ADDR, #00H
          LCALL SET_ADDR_LCD
          MOV DPTR, #CODE
          LCALL WRLINE_LCD
          MOV LCD_ADDR, #004H
          LCALL SET_ADDR_LCD

```

```

MOV LCD_DATA, #95

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL WRCHAR_LCD
        MOV  R2,#005H      ;CONTROL *
LOOP_D:MOV  LCD_ADDR,R2
        LCALL SET_ADDR_LCD
        MOV  LCD_DATA,#178
        LCALL WRCHAR_LCD
        INC  R2
        CJNE R2,#009H,LOOP_D
        MOV  LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV  LCD_DATA,#84
        LCALL WRCHAR_LCD
        MOV  LCD_DATA,#73
        LCALL WRCHAR_LCD
        MOV  LCD_DATA,#77
        LCALL WRCHAR_LCD
        MOV  LCD_DATA,#69
        LCALL WRCHAR_LCD
        MOV  LCD_DATA,#95
        LCALL WRCHAR_LCD
        MOV  R2,#04      ;CONTROL *
LOOP_D2:
        MOV  LCD_DATA,#178
        LCALL WRCHAR_LCD
        DJNZ R2,LOOP_D2
        MOV  R2,#00
        ;LCALL DELAY_1s
LOOP_REC: MOV  R1,#005H
        MOV  R0,#7CH      ;KEEP START ADDRESS

LOOP_DISPLAY::;CONTROLL DISPLAY GET KEY NUMBER
        MOV  LCD_ADDR,R1
        LCALL SET_ADDR_LCD ;
        LCALL KEY ;KEY_BUF IS GEEP CODE KEY SCII DECIMAL
        MOV  LCD_DATA,KEY_BUF

```

เอกสารนี้เป็นเอกสารของงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL WRCHAR_LCD    ;WRITE 1 CHAR
INC R1
MOV @R0,KEY_BUF ;KEEP CODE KEY SCII CODE ADDR START 7CH
INC R0            ; R0 IS POITER
CJNE R1,#009H,LOOP_DISPLAY ;FULL 4 DIGIT OUT LOOP
LCALL KEY
MOV R3,KEY_BUF
CJNE R3,#69,IN_M    ;69 IS 'E' ENTER KEY
MOV LCD_ADDR,#005H
LCALL SET_ADDR_LCD
MOV LCD_DATA,#178
LCALL WRCHAR_LCD
MOV LCD_ADDR,#006H
LCALL SET_ADDR_LCD
MOV LCD_DATA,#79    ;79 IS 'O'
LCALL WRCHAR_LCD
MOV LCD_ADDR,#007H
LCALL SET_ADDR_LCD
MOV LCD_DATA,#75    ;75 IS 'K'
LCALL WRCHAR_LCD
MOV R1,#008H
LOOP_D4:
MOV LCD_ADDR,R1
LCALL SET_ADDR_LCD
MOV R5,#005
MOV LCD_DATA,#178    ;178 IS SCII DECIMAL *
LCALL WRCHAR_LCD
INC R1
CJNE R1,#009H,LOOP_D4
MOV R1,#000
MOV R5,#000
;LCALL DELAY_1s
LCALL RD_RTC
LCALL SCAN_BAR
LCALL LCD_CLR

```

เอกสารนี้เป็นเอกสารที่เผยแพร่สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DELAY_1s
LCALL IN_RAM      ;KEPT RECORD GOTO INTERNAL RAM
DJNZ  R4, LOOP_REC ;OVER RECORD GOTO OUT
;LCALL RD_INBAR    ;TEST READ RECORD

LJMP  MAIN

```

```

;-----WRITE INRAM-----

```

```

IN_RAM:  MOV  PSW,#18H
;MOV  R1,#01H
;MOV  R2,#00H
MOV  DPH,R1
        MOV  DPL,R2
        MOV  PSW,#00H
        MOV  A,61H ;61H KEEP SCANCODE KEYBROAD DIGIT 1
        ;MOV  DPTR,#0100H ;ADDR ORIGIN 0000H
        MOVX  @DPTR,A
        INC  DPTR
        MOV  A,64H ;64H KEEP SCANCODE KEYBROAD DIGIT 2
        MOVX  @DPTR,A
        INC  DPTR
        MOV  A,67H ;67H KEEP SCANCODE KEYBROAD DIGIT 3
        MOVX  @DPTR,A

```

```

;-----CODE 4 DIGIT-----

```

```

        MOV  R1,#7CH      ;ADDR ORIGIN FOR KEYPESS
        MOV  R6,#004      ;R6 CONTROL NUM CODE
        MOV  R0,#004H

```

```

LOOP_KEY:  INC  DPTR
           MOV  A,@R1
           MOVX  @DPTR,A

```

```

           INC  R1

```

เอกสารนี้เป็นเอกสารต้นฉบับสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DJNZ R6, LOOP_KEY
```

```
;-----KEEP RTC HR AND SEC-----
```

```
LCALL RTC_RD
```

```
INC DPTR
```

```
MOV PSW, #18H
```

```
MOV R3, HOURS
```

```
MOV A, R3
```

```
MOV PSW, #00H
```

```
;MOV A, HOURS
```

```
MOVX @DPTR, A
```

```
INC DPTR
```

```
MOV PSW, #18H
```

```
MOV R4, MINUTES
```

```
MOV A, R4
```

```
MOV PSW, #00H
```

```
;MOV A, MINUTES
```

```
MOVX @DPTR, A
```

```
INC DPTR
```

```
MOV A, #45H ;END OF RECORD ADDR 0009H 45H = 'E'
```

```
MOVX @DPTR, A
```

```
INC DPTR
```

```
MOV PSW, #18H
```

```
MOV R1, DPH
```

```
MOV R2, DPL
```

```
MOV PSW, #00H
```

```
LCALL LCD_CLR
```

```
MOV LCD_ADDR, #002H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA, #'K' ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD
```

```
LCALL DELAY_1s
```

```
LCALL DELAY_1s
```

```
LCALL LCD_CLR
```

```
LCALL TEST_IRAM
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

;-----DISPLAY BARCODE TEST-----

;=====RECORD HAVE 10 BYTE ORIGIN 0100H=====BYTE AT 10 IS'E'=====

RD_INBAR: MOV R1, #003 ;CONTROL NUM BARCODE

 MOV R0, #000H ;CONTROL ADDR LCD

 MOV DPH, #01H ;READ BYTE AT 1 ADDRESS 0100H

 MOV DPL, #00H

;MOV DPH, #01H ;READ BYTE AT 2 ADDRESS 010AH

;MOV DPL, #0AH

;MOV DPH, #01H ;READ BYTE AT 3 ADDRESS 0114H

;MOV DPL, #14H

;MOV DPTR, #0100H

;MOV DPTR, #000AH

LOOP_SCAN_CHK:

 MOVX A, @DPTR ;BYTE AT 1

 CJNE A, BAR_1, CK_EEOR

 INC DPTR

 MOVX A, @DPTR ;BYTE AT 2

 CJNE A, BAR_2, CK_EEOR

 INC DPTR

 MOVX A, @DPTR ;BYTE AT 3

 CJNE A, BAR_3, CK_EEOR

 INC DPTR ;BYTE AT 4

;-----

;-----DISPLAY KEYPESS-----

 ;MOV DPTR, #0002H

 MOV R1, #004 ;CONTROL SCANKEY

 MOV R0, #005H ;CONTROL ADDR LCD

LOOP_K: MOV LCD_ADDR, R0

เอกสารนี้เป็นเอกสารที่ LCALL SET_ADDR_LCD เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVX A, @DPTR    ;BYTE AT 4 TO BYTE 7
MOV LCD_DATA, A  ;DEF SCII TO LCD DATA
LCALL WRCHAR_LCD
INC R0
INC DPTR
DJNZ R1, LOOP_K
LCALL DELAY_1s
LCALL DELAY_1s
;LCALL DELAY_1s
LCALL LCD_CLR
LCALL DELAY_1s
MOV R0, #00H
MOV R6, #000
;-----DISPLAY RTC INTERNALRAM-----
MOVX A, @DPTR    ;BYTE AT 8
;MOV HOURS, A
MOV PSW, #18H
;MOV DPTR, #0107H
MOVX A, @DPTR
MOV R5, A        ;R5 KEEP HOUR
MOV PSW, #00H
INC DPTR
MOVX A, @DPTR    ;BYTE AT 9
;MOV MINUTES, A
MOV PSW, #18H
;MOV DPTR, #0108H
MOVX A, @DPTR

MOV R6, A        ;R6 KEEP SEC
MOV PSW, #00H
LCALL TEST_IRAM ;

INC DPTR        ;BYTE AT 10 = 'E'
INC DPTR        ;BYTE AT 11 NEW RECORD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CK_EEOR:  INC  DPTR

          MOVX  A,@DPTR

          CJNE  A,#45H,CK_EEOR ;CHECK BYTE AT 10

          INC   DPTR           ;NEW RECORD  BYTE AT 11

          LJMP  LOOP_SCAN_CK

```

```

;-----TEST IRAM-----
TEST_IRAM: MOV  LCD_ADDR,#046H

          LCALL SET_ADDR_LCD

          MOV   PSW,#18H

          MOV   A,R5

          MOV   PSW,#00H

          ANL   A,#00110000B ; Get x10 Digit

          JZ    WRITE_TIME_HNN ; Check x10 Digit = 0 ?

          SWAP  A ; If not => Write to LCD

          ADD   A,#030H ; Convert to ASCII

          AJMP  WRITE_TIME_HHH ;

          WRITE_TIME_HNN:MOV  A,#' ' ; x10 = 0 then Write space

          WRITE_TIME_HHH:MOV  LCD_DATA,A ; Write to LCD

          LCALL WRCHAR_LCD ;

          MOV   LCD_ADDR,#047H

          LCALL SET_ADDR_LCD

          MOV   PSW,#18H

          MOV   A,R5

          MOV   PSW,#00H

          ANL   A,#00001111B ;

          ADD   A,#030H ; Convert to ASCII

          MOV   LCD_DATA,A ;

          LCALL WRCHAR_LCD ;

          MOV   LCD_ADDR,#049H

          LCALL SET_ADDR_LCD

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL WRCHAR_LCD ;
LJMP WRITE_MINUTESS ;
WRITE_SPACEE: MOV LCD_DATA,#' ' ;
ACALL WRCHAR_LCD ;
WRITE_MINUTESS:
;MOV DPTR,#0108H
;MOVX A,@DPTR
MOV PSW,#18H
MOV A,R6
MOV PSW,#00H
MOV LCD_DATA,A
;MOV LCD_DATA,MINUTES ; Write Minutes to LCD
LCALL BCD2LCD ; Write from RTC BCD Data
LCALL DELAY_1s
RET
;-----END TEST IRAM-----

;MOV DPTR,#0000H
MOV R7,#003
MOV R0,#000H

DIS_S:
MOV LCD_ADDR,R0
LCALL SET_ADDR_LCD
MOVX A,@DPTR
LCALL SCAN_KEY_CH
MOV R7,BAR_BUF
MOV LCD_DATA,R7 ;DEF SCII TO LCD DATA
LCALL WRCHAR_LCD
INC DPTR
INC R0
DJNZ R7,DIS_S
;-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----TEST KEEP KEY-----
```

```
MOV R0,#7CH;START ADDR FOR AT KEEP SCII CODE 7CH DIGIT 1
```

```
;MOV PSW,#08H
```

```
MOV PSW,#10H
```

```
MOV R1,#045H
```

```
MOV R2,#04
```

```
GO_OUT: ;MOV PSW,#08H
```

```
MOV PSW,#10H
```

```
MOV LCD_ADDR,R1
```

```
LCALL SET_ADDR_LCD
```

```
;MOV LCD_DATA,7FH
```

```
MOV PSW,#00H
```

```
MOV LCD_DATA,@R0
```

```
LCALL WRCHAR_LCD
```

```
LCALL DELAY_1s
```

```
MOV PSW,#00H
```

```
INC R0
```

```
;MOV PSW,#08H
```

```
MOV PSW,#10H
```

```
INC R1
```

```
DJNZ R2,GO_OUT
```

```
MOV PSW,#00H
```

```
;LJMP GO_OUT
```

```
;-----END TEST-----
```

```
OUT_MAIN:
```

```
;LJMP MAIN
```

```
;-----TEST RTC---SET RTC-----
```

```
WR_RTC:
```

```
SETB SDA ;Clear I2C bus
```

```
SETB SCL
```

```
MOV SECONDS,#00h ;initial real time clock
```

```
MOV MINUTES,#02h
```

```
MOV HOURS,#13h
```

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL RTC_WR
```

```
RET
```

```
;-----READ REAL TIME CLOCK-----
```

```
RD_RTC:
```

```
LCALL RTC_RD ; Read RTC
```

```
RTC_INRAM:
```

```
MOV LCD_ADDR, #045H
```

```
LCALL SET_ADDR_LCD
```

```
;RTC_INRAM:
```

```
MOV A,HOURS ; Get Hour
```

```
ANL A,#00110000B ; Get x10 Digit
```

```
JZ WRITE_TIME_HN ; Check x10 Digit = 0 ?
```

```
SWAP A ; If not => Write to LCD
```

```
ADD A,#030H ; Convert to ASCII
```

```
AJMP WRITE_TIME_HH ;
```

```
WRITE_TIME_HN:MOV A,#' ' ; x10 = 0 then Write space
```

```
WRITE_TIME_HH:MOV LCD_DATA,A ; Write to LCD
```

```
LCALL WRCHAR_LCD ;
```

```
MOV LCD_ADDR, #046H
```

```
LCALL SET_ADDR_LCD
```

```
MOV A,HOURS ; Write x1 Digit to LCD
```

```
ANL A,#00001111B ;
```

```
ADD A,#030H ; Convert to ASCII
```

```
MOV LCD_DATA,A ;
```

```
LCALL WRCHAR_LCD ;
```

```
MOV LCD_ADDR, #048H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA,#':' ; Odd => Write ':'
```

```
LCALL WRCHAR_LCD ;
```

```
LJMP WRITE_MINUTES
```

เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRITE_SPACE: MOV    LCD_DATA,#' ' ;
ACALL  WRCHAR_LCD    ;
WRITE_MINUTES:MOV    LCD_DATA,MINUTES ; Write Minutes to LCD
LCALL  BCD2LCD        ; Write from RTC BCD Data
LCALL  DELAY_1s
RET

```

;-----TEST BARCODE-----

TESTBAR:

SCAN_BAR:

```

MOV    R2,#08H        ;CONTROL BYTE
MOV    R1,#60H        ;KEEP ADDR FOR DATA

```

BYTE_L:

```

MOV    R0,#00H        ;CONTROL BIT
INC    R1

```

;-----1 BYTE-----

```

CK_H:   CLR    P1.0
        SETB   P2.4
        JB     P2.4,GET_SCAN
        LJMP   CK_MENU

```

GET_SCAN:

```

        JB     P3.1,CK_H
        INC    R0        ;NUM CLOCK
        CJNE   R0,#0BH,LESS_10
        MOV    @R1,A
        MOV    A,#00H
        AJMP   SHOW_BAR    ;CLOCK AT 11 FREE AND OUT

```

LESS_10: CJNE R0,#0AH,KEEP_IN

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AJMP CK_L ;CLOCK AT 10 FREE
```

```
KEEP_IN: MOV C, P2.0 ;CLOCK AT 1 TO 9
```

```
RRC A
```

```
CK_L: JNB P3.1, CK_L
```

```
AJMP CK_H
```

```
;-----END BYTE-----
```

```
SHOW_BAR: JNB P3.1, SHOW_BAR ;WAIT LOGIC HIGHT
```

```
DJNZ R2, BYTE_L ;CONTROL BYTE
```

```
;-----
```

```
MOV LCD_ADDR, #009H
```

```
LCALL SET_ADDR_LCD
```

```
MOV LCD_DATA, #058 ;DEF SCII TO LCD DATA :
```

```
LCALL WRCHAR_LCD
```

```
;-----SELECT DATA---DIGIT 1-----
```

```
MOV LCD_ADDR, #0AH
```

```
LCALL SET_ADDR_LCD
```

```
MOV KEEP_BAR, #00H
```

```
MOV A, #00H
```

```
MOV R1, #00H
```

```
MOV R1, 61H ;R1 KEPT SCAN KEY
```

```
MOV BAR_1, R1
```

```
MOV KEEP_BAR, R1 ;SEND DATA SCAN CODE TO BUFFER
```

```
MOV A, KEEP_BAR
```

```
LCALL SCAN_KEY_CH
```

```
MOV R7, BAR_BUF
```

```
MOV LCD_DATA, R7 ;DEF SCII TO LCD DATA
```

```
LCALL WRCHAR_LCD ;SHOW SCII AT LCD DISPLAY 1 CHAR
```

```
;-----SELECT DATA---DIGIT 2-----
```

```
MOV LCD_ADDR, #0BH
```

```
LCALL SET_ADDR_LCD
```

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    KEEP_BAR, #00H
MOV    A, #00H
MOV    R1, #00H
MOV    R1, 64H
MOV    BAR_2, R1
MOV    KEEP_BAR, R1
MOV    A, KEEP_BAR
LCALL  SCAN_KEY_CH    ;FUNTION SCAN KEY
MOV    R7, BAR_BUF
MOV    LCD_DATA, R7
LCALL  WRCHAR_LCD
;-----SELECT DATA---DIGIT 3-----
MOV    LCD_ADDR, #0CH
LCALL  SET_ADDR_LCD
MOV    KEEP_BAR, #00H
MOV    A, #00H
MOV    R1, #00H
MOV    R1, 67H
MOV    BAR_3, R1    ;R1 KEPT CODE FOR SCAN KEY
MOV    KEEP_BAR, R1
MOV    A, KEEP_BAR
LCALL  SCAN_KEY_CH
MOV    R7, BAR_BUF
MOV    LCD_DATA, R7
LCALL  WRCHAR_LCD

```

```

LCALL  DELAY_1s

```

```

RET

```

```

;-----BARCODE SCAN-TO NUMBER-----

```

```

SCAN_KEY_CH:

```

```

CJNE  A, #45H, NEXT_1

```

```

MOV   BAR_BUF, #'0'

```

```

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEXT_1:    CJNE    A, #16H, NEXT_2
            MOV     BAR_BUF, #'1'
            RET

NEXT_2:    CJNE    A, #1EH, NEXT_3
            MOV     BAR_BUF, #'2'
            RET

NEXT_3:    CJNE    A, #26H, NEXT_4
            MOV     BAR_BUF, #'3'
            RET

NEXT_4:    CJNE    A, #25H, NEXT_5
            MOV     BAR_BUF, #'4'
            RET

NEXT_5:    CJNE    A, #2EH, NEXT_6
            MOV     BAR_BUF, #'5'
            RET

NEXT_6:    CJNE    A, #36H, NEXT_7
            MOV     BAR_BUF, #'6'
            RET

NEXT_7:    CJNE    A, #3DH, NEXT_8
            MOV     BAR_BUF, #'7'
            RET

NEXT_8:    CJNE    A, #3EH, NEXT_9
            MOV     BAR_BUF, #'8'
            RET

NEXT_9:    CJNE    A, #46H, NEXT_10
            CJNE    A, #00H, NEXT_10

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    BAR_BUF, #'9'
RET
```

NEXT_10:

```
MOV    BAR_BUF, ###
RET
```

;-----FUNCTION KEYPESS-----

```
KEY:    ;MOV    KEY_BUF, #66
        MOV    P1, #11111111B
        MOV    P2, #01111111B
        JB     P1.0, KEY_2
        MOV    KEY_BUF, #49
        JNB    P1.0, $
        RET
```

```
KEY_2:  JB     P1.1, KEY_3
        MOV    KEY_BUF, #50
        JNB    P1.1, $
        RET
```

```
KEY_3:  JB     P1.2, KEY_4
        MOV    KEY_BUF, #51
        JNB    P1.2, $
        RET
```

```
KEY_4:  JB     P1.3, ROW_2
        MOV    KEY_BUF, #52
        JNB    P1.3, $
        RET
```

;-----END ROW 1-----

```
ROW_2:  MOV    P1, #11111111B
        MOV    P2, #10111111B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้เฉพาะงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JB      P1.0,KEYC_2
        MOV     KEY_BUF,#53
        JNB     P1.0,$
        RET

KEYC_2:   JB      P1.1,KEYC_3
        MOV     KEY_BUF,#54
        JNB     P1.1,$
        RET

KEYC_3:   JB      P1.2,KEYC_4
        MOV     KEY_BUF,#55
        JNB     P1.2,$
        RET

KEYC_4:   JB      P1.3,ROW_3
        MOV     KEY_BUF,#56
        JNB     P1.3,$
        RET
;-----END ROW 2-----
ROW_3: MOV     P1,#11111111B
        MOV     P2,#11011111B
        JB      P1.0,KEYCC_2
        MOV     KEY_BUF,#57
        JNB     P1.0,$
        RET

KEYCC_2:  JB      P1.1,KEYCC_3
        MOV     KEY_BUF,#48
        JNB     P1.1,$
        RET

KEYCC_3:  JB      P1.2,KEYCC_4
        MOV     KEY_BUF,#65
        JNB     P1.2,$
        RET

KEYCC_4:  JB      P1.3,ROW_4
        MOV     KEY_BUF,#69
        JNB     P1.3,$
        RET

```

เอกสารนี้เป็นเอกสารที่วางไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----END ROW3-----
ROW_4:      MOV    P1,#11111111B
            MOV    P2,#11101111B
            JB     P1.0,KEYE_2
            MOV    KEY_BUF,#77
            JNB    P1.0,$
            RET
KEYE_2:     JB     P1.1,KEYE_3
            MOV    KEY_BUF,#67
            JNB    P1.1,$
            RET
KEYE_3:     JB     P1.2,KEYE_4
            MOV    KEY_BUF,#66
            JNB    P1.2,$
            RET
KEYE_4:     JB     P1.3,OUT_ROW4
            MOV    KEY_BUF,#68
            JNB    P1.3,$
OUT_ROW4:   LJMP   KEY
            RET
;-----END ROW4-----
;
; BCD Code to show LCD
; I/P:      LCD_DATA
;
BCD2LCD:    PUSH    ACC      ; Push ACC. To Stack
            PUSH    B        ; Push B to Stack
            MOV     A,LCD_DATA ; Get input data value
            MOV     B,A      ; Copy to B
            ANL     A,#11110000B ; Get higher 4 bit
            SWAP    A        ; Swap nibble
            ADD     A,#030H   ; Convert to ASCII
            MOV     LCD_DATA,A ; Write LCD
            ACALL   WRCHAR_LCD

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A,B                ; Restore value
ANL      A,#00001111B      ; Get lower 4 bit
ADD      A,#030H           ; Convert to ASCII
MOV      LCD_DATA,A        ; Write LCD
ACALL    WRCHAR_LCD        ;
POP      B                  ; Pop B from Stack
POP      ACC                ; Pop ACC. from Stack
RET                               ; Return

```

```

;-----
; LCD Initialize
;-----

```

```

INIT_LCD:  LCALL DELAY_100ms    ; Delay
           CLR      LCD_RS      ; Clear LCD_RS Pin

           MOV      P0,#00111000B ; 8bit Mode
           LCALL    LCD_CLK      ; Pulse LCD Clock
           LCALL    DELAY_10ms   ; Delay

           MOV      P0,#00111000B ; 8bit Mode
           LCALL    LCD_CLK      ; Pulse LCD Clock

           LCALL    LCD_OFF      ; Display Off

           LCALL    LCD_CLR      ; Clear Display

           MOV      P0,#00000110B ; Entry Mode
           LCALL    LCD_CLK      ; Pulse LCD Clock

           LCALL    LCD_HOME     ; Return Home Display

```

```

;-----
; LCD Clear Display
;-----

```

```

LCD_CLR:   CLR      LCD_RS      ; Clear LCD_RS Pin

```

เอกสารนี้เป็นเอกสารตัวอย่างไว้สำหรับการศึกษา; Display Clear; อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL LCD_CLK                ; Pulse LCD Clock
        RET

;-----
; LCD Return Home
;-----
LCD_HOME:  CLR          LCD_RS          ; Clear LCD_RS Pin
           MOV          P0,#00000010B    ; Return Home
           LCALL LCD_CLK                ; Pulse LCD Clock
           RET

;-----
; LCD Display Off
;-----
LCD_OFF:   CLR          LCD_RS          ; Clear LCD_RS Pin
           MOV          P0,#00001000B    ; Display Off
           LCALL LCD_CLK                ; Pulse LCD Clock
           RET

;-----
; LCD Clk
;-----
LCD_CLK:   SETB         LCD_EN          ; Pulse Clock to LCD_EN
           LCALL LCD_DELAY
           CLR          LCD_EN
           LCALL LCD_DELAY
           RET

;-----
; LCD Display On
;-----
LCD_ON:    CLR          LCD_RS          ; Clear LCD_RS Pin
           MOV          P0,#00001100B    ; Display On
           ACALL LCD_CLK
           RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; LCD Cursor On
;-----

LCD_BLINK:  CLR          LCD_RS      ; Clear LCD_RS Pin
            MOV          P0,#00001111B  ; Display Cursor and Blink
            ACALL        LCD_CLK      ; Pulse LCD Clock
            RET

;-----
; Set LCD Address
; I/P:      LCD_ADDR
;-----

SET_ADDR_LCD:CLR          LCD_RS      ; Clear LCD_RS Pin
            MOV          A,LCD_ADDR    ; Move LCD_ADDR to ACC.
            SETB         ACC.7        ; Set bit ACC.7
            MOV          P0,A         ; Move to DATABUS
            ACALL        LCD_CLK      ; Pulse LCD Clock
            RET

;-----
; Write Character to show LCD
; I/P:      LCD_DATA
;-----

WRCHAR_LCD: SETB          LCD_RS      ; Set LCD_RS Pin
            MOV          P0,LCD_DATA  ; Move LCD_DATA to DATABUS
            ACALL        LCD_CLK      ; Pulse LCD Clock
            ACALL        LCD_ON      ; Display On
            RET

;-----
; Write Line of 16 Character from ROM
; I/P:      DPTR : Locate ROM Address
;-----

WRLINE_LCD: MOV          R0,#0        ; Clear loop counter

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRLINE_LCD_1:SETB    LCD_RS      ; Set LCD_RS Pin
                    CLR          A          ; Clear ACC.
                    MOVC         A,@A+DPTR ; Move data from @DPTR to ACC.
                    MOV          P0,A      ; Move ACC. to DATABUS
                    ACALL        LCD_CLK   ; Pulse LCD Clock
                    INC          DPTR      ; Increase Pointer
                    INC          R0        ; Increase loop counter
                    CJNE         R0,#8,WRLINE_LCD_1 ; Do until 8 times

                    MOV          LCD_ADDR,#040H ; Set Later 8 Char. Address
                    ACALL        SET_ADDR_LCD

WRLINE_LCD_2:SETB    LCD_RS      ; Set LCD_RS Pin
                    CLR          A          ; Clear ACC.
                    MOVC         A,@A+DPTR ; Move data from @DPTR to ACC.
                    MOV          P0,A      ; Move ACC. to DATABUS
                    ACALL        LCD_CLK   ; Pulse LCD Clock
                    INC          DPTR      ; Increase Pointer
                    INC          R0        ; Increase loop counter
                    CJNE         R0,#16,WRLINE_LCD_2 ; Do until 8+8 times
                    ACALL        LCD_ON    ; Display On
                    RET

;-----FUCTION RTC-----
RTC_RD:    MOV          I2C_ADDR,#RTC_ID ; Set RTC as I2C Write Slave
                    LCALL        I2C_SLAVE ; Connect Slave
                    MOV          I2C_DATA,#000H ; Set Slave Address 00H
                    LCALL        I2C_DATA_WR ; Write Data to Slave
                    MOV          I2C_ADDR,#RTC_ID+1 ;Set RTC as I2C Read Slave
                    LCALL        I2C_SLAVE ; Connect Slave
                    LCALL        I2C_DATA_RD ; Read Data from Slave
                    MOV          SECONDS,I2C_DATA ; Read Data to SECONDS
                    LCALL        I2C_ACK_BIT ; Send Acknowledge
                    LCALL        I2C_DATA_RD ; Read Data from Slave
                    MOV          MINUTES,I2C_DATA ; Read Data to MINUTES
                    LCALL        I2C_ACK_BIT ; Send Acknowledge

```

เอกสารนี้เป็นเอกสารต้นฉบับไว้สำหรับการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL I2C_DATA_RD ; Read Data from Slave
MOV HOURS,I2C_DATA ; Read Data to HOURS
LCALL I2C_ACK_BIT ; Send Acknowledge
LCALL I2C_DATA_RD ; Read Data from Slave
MOV DAY,I2C_DATA ; Read Data to DAY
LCALL I2C_ACK_BIT ; Send Acknowledge
LCALL I2C_DATA_RD ; Read Data from Slave
MOV DATE,I2C_DATA ; Read Data to DATE
LCALL I2C_ACK_BIT ; Send Acknowledge
LCALL I2C_DATA_RD ; Read Data from Slave
MOV MONTH,I2C_DATA ; Read Data to MONTH
LCALL I2C_ACK_BIT ; Send Acknowledge

LCALL I2C_DATA_RD ; Read Data from Slave
MOV YEAR,I2C_DATA ; Read Data to YEAR
LCALL I2C_ACK_BIT ; Send Acknowledge
LCALL I2C_DATA_RD ; Read Data from Slave
MOV CONTROL,I2C_DATA ; Read Data to CONTROL
LCALL I2C_NACK_BIT ; Send Not Acknowledge
LCALL I2C_STOP ; Send Stop Condition
RET ; Return
```

; I2C RTC Write

```
RTC_WR: MOV I2C_ADDR,#RTC_ID ; Set RTC as I2C Write Slave
LCALL I2C_SLAVE ; Connect Slave
MOV I2C_DATA,#000H ; Set Slave Address 00H
LCALL I2C_DATA_WR ; Write Data to Slave
MOV I2C_DATA,SECONDS ; Write SECONDS to RTC
LCALL I2C_DATA_WR ;
MOV I2C_DATA,MINUTES ; Write MINUTES to RTC
LCALL I2C_DATA_WR ;
MOV I2C_DATA,HOURS ; Write HOURS to RTC
LCALL I2C_DATA_WR ;
```

เอกสารนี้เป็นเอกสารตัวอย่างไว้สำหรับใช้ศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL    I2C_DATA_WR          ;
MOV      I2C_DATA,DATE        ; Write DATE to RTC
LCALL    I2C_DATA_WR          ;
MOV      I2C_DATA,MONTH       ; Write MONTH to RTC
LCALL    I2C_DATA_WR          ;
MOV      I2C_DATA,YEAR        ; Write YEAR to RTC
LCALL    I2C_DATA_WR          ;

MOV      I2C_DATA,CONTROL     ; Write CONTROL to RTC
LCALL    I2C_DATA_WR          ;
LCALL    I2C_STOP              ; Send Stop Condition
RET      ; Return
;-----
; I2C Data Write
; I/P:    I2C_DATA
; Reserve: R5
;-----
I2C_DATA_WR:PUSH    ACC        ; Push ACC.
SETB     I2C_ACK          ; Set ACK. Bit
MOV      A,I2C_DATA        ; Get Data
MOV      R5,#008           ; Set loop 8 times
I2C_DATA_WR_1:RLC    A         ; Rotate ACC. to Left with Carry
MOV      SDA,C             ; Move Carry Flag to SDA
LCALL    I2C_CLK           ; Pulse I2C Clock
DJNZ     R5,I2C_DATA_WR_1    ; Do until 8 times
SETB     SDA               ; Set SDA
LCALL    I2C_DELAY         ; Delay
SETB     SCL               ; Set SCL
LCALL    I2C_DELAY         ; Delay
JB       SDA,I2C_DATA_WR_2   ; Check Acknowledge from Slave
CLR      I2C_ACK           ; Clear ACK. Bit
I2C_DATA_WR_2:CLR    SCL      ; Clear SCL
POP      ACC               ; Pop ACC.
RET      ; Return

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; I2C Data Read

; O/P: I2C_DATA

; Reserve: R5

```

I2C_DATA_RD:PUSH    ACC        ; Push ACC.
                    CLR        A        ; Clear ACC.
                    MOV        R5,#008 ; Set loop 8 times

```

```

I2C_DATA_RD_1:LCALL I2C_DELAY ; Delay
                    SETB       SCL      ; Set SCL
                    LCALL      I2C_DELAY ; Delay
                    MOV        C,SDA     ; Get SDA to Carry Flag
                    RLC        A        ; Rotate ACC. to Left with Carry
                    CLR        SCL      ; Clear SCL
                    DJNZ       R5,I2C_DATA_RD_1 ; Do until 8 times
                    MOV        I2C_DATA,A ; Move Data to I2C_DATA
                    POP        ACC      ; Pop ACC.
                    RET          ; Return

```

; I2C Slave Connect

; I/P: I2C_ADDR

; O/P Flag: I2C_ACK

; Reserve: R5

```

I2C_SLAVE:  PUSH    ACC        ; Push ACC.
            SETB     I2C_ACK    ; Set ACK. Bit
            MOV      A,I2C_ADDR ; Get Slave Address
            LCALL    I2C_START  ; Send Start Condition
            MOV      R5,#008    ; Set loop 8 times
I2C_SLAVE_1: RLC      A        ; Rotate ACC. to Left with Carry
            MOV      SDA,C      ; Move Carry Flag to SDA
            LCALL    I2C_CLK    ; Pulse I2C Clock
            DJNZ     R5,I2C_SLAVE_1 ; Do until 8 times
            SETB     SDA        ; Set SDA
            LCALL    I2C_DELAY  ; Delay

```

เอกสารนี้เป็นเอกสารตัวอย่างไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SETB    SCL                                ; Set SCL
LCALL   I2C_DELAY                          ; Delay
JB      SDA,I2C_SLAVE_2 ;Check Acknowledge from Slave
CLR     I2C_ACK                            ; Clear ACK.
I2C_SLAVE_2:CLR    SCL                    ; Clear SCL
POP     ACC                                ; Pop ACC.
RET     ; Return
;-----
; I2C Start Condition
;-----
I2C_START:  JNB    SCL,I2C_START_1      ; Check current SCL set?
CLR        SCL                          ; Clear SCL
I2C_START_1:SETB   SDA                  ; Set SDA
SETB      SCL                          ; Set SCL
LCALL     I2C_DELAY                    ; Delay
CLR       SDA      ; Clear SDA during SCL set
LCALL     I2C_DELAY                    ; Delay
CLR       SCL      ; Clear SCL
RET       ; Return
;-----
; I2C Stop Condition
;-----
I2C_STOP:   JNB    SCL,I2C_STOP_1      ; Check current SCL set?
CLR        SCL                          ; Clear SCL
I2C_STOP_1: CLR    SDA                  ; Clear SDA
LCALL      I2C_DELAY                    ; Delay
SETB       SCL                          ; Set SCL
LCALL      I2C_DELAY                    ; Delay
SETB       SDA      ; Set SDA during SCL set
RET       ; Return
;-----
; I2C Clock
;-----
I2C_CLK:    LCALL   I2C_DELAY          ; Pulse SCL
SETB      SCL
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL    I2C_DELAY        ;
        CLR      SCL              ;
        RET              ; Return
;-----
; I2C Acknowledge
;-----
I2C_ACK_BIT:CLR      SDA          ; Clear SDA
        LCALL    I2C_DELAY        ; Delay
        LCALL    I2C_CLK         ; Pulse I2C Clock
        SETB     SDA
        RET              ; Return
;-----
; I2C Not Acknowledge
;-----
I2C_NACK_BIT:SETB    SDA          ; Set SDA
        LCALL    I2C_DELAY        ; Delay
        LCALL    I2C_CLK         ; Pulse I2C Clock
        SETB     SCL
        RET              ; Return
;-----
; Dummy Delay time I2C_DELAY, LCD_DELAY, 10m, 100m, 1s
;-----
I2C_DELAY: MOV       R6,#09cH      ; Each loop = 50 us
I2C_DELAY_1:NOP
        NOP
        DJNZ     R6,I2C_DELAY_1
        RET

LCD_DELAY: MOV       R7,#004        ; Do 2 times
LCD_DELAY_1:MOV      R6,#0E6H      ; Each loop = 1 ms
LCD_DELAY_2:NOP
        NOP
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DJNZ    R6,LCD_DELAY_2
DJNZ    R7,LCD_DELAY_1
RET

DELAY_10ms: MOV    R7,#020      ; Do 10 times
DELAY_10ms_1:MOV    R6,#0E6H    ; Each loop = 1 ms
DELAY_10ms_2:NOP
NOP
DJNZ    R6,DELAY_10ms_2
DJNZ    R7,DELAY_10ms_1
RET

DELAY_100ms:MOV    R7,#200      ; Do 100 times
DELAY_100ms_1:MOV    R6,#0E6H  ; Each loop = 1 ms
DELAY_100ms_2:NOP
NOP
DJNZ    R6,DELAY_100ms_2
DJNZ    R7,DELAY_100ms_1
RET

DELAY_1s: MOV    R5,#200      ; Do 100 times
DELAY_1s_1:  LCALL    DELAY_10ms
DJNZ    R5,DELAY_1s_1
RET

TIME1: DB 'TIME'
CODE: DB 'CODE'
;00H,01H,02H,03H,04H,05H,06H,07H,08H,09H,0AH,0BH,0CH,0DH,0EH,0FH

END
```

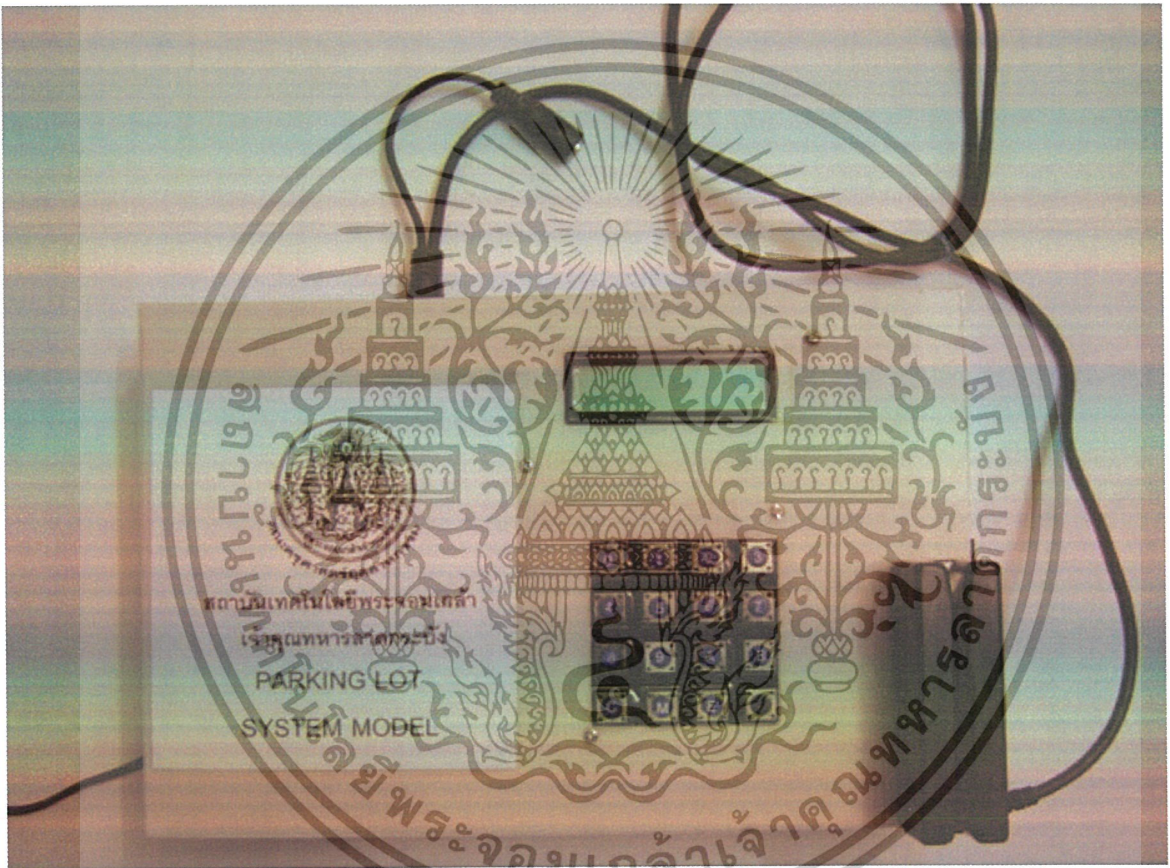
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน แบบจำลองระบบจอดรถ



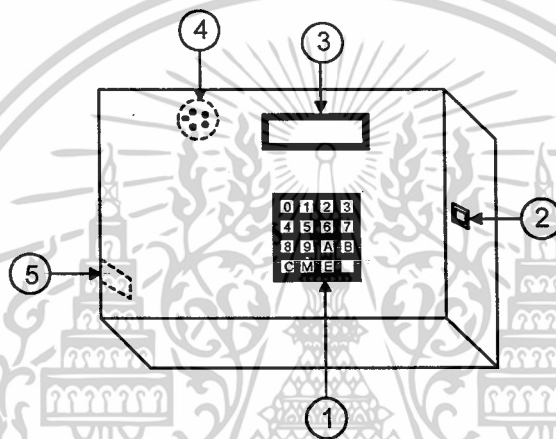
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำแนะนำเบื้องต้น

ก่อนที่จะลงมือใช้งานแบบจำลองระบบจอตลอด ควรทำการศึกษาการใช้งานจากคู่มือให้เข้าใจเพื่อการใช้งานที่ถูกต้อง และเป็นการป้องกันความเสียหายที่เกิดจากความผิดพลาดของผู้ใช้งาน

2. ส่วนประกอบและปุ่มควบคุม



รูปที่ จ.1 ส่วนประกอบและปุ่มควบคุมของตัวเครื่องแบบจำลองระบบจอตลอด

จากรูปที่ จ.1 มีรายละเอียดต่างๆ ดังนี้

1. สวิตช์เมตริกซ์
2. สวิตช์รีเซต
3. จอแสดงผลแบบผลึกเหลว
4. จุดต่อหัววัดของเครื่องอ่านบัตรบาร์โค้ด
5. จุดจ่ายไฟให้กับตัวเครื่อง

3. การติดตั้งและการใช้งาน

- 3.1 ต่อหัวอ่านบัตรบาร์โค้ดเข้ากับตัวเครื่องแบบจำลองระบบจอตลอดทุกครั้ง
- 3.2 ทำการเปิดเครื่อง
- 3.3 เมื่อทำการเปิดเครื่องจอ LCD จะแสดงคำว่า “Mode” ขึ้นมาแสดงว่าเครื่องพร้อมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 เมื่อต้องการรูดบัตรเช็คทะเบียนรถทำการกด B เพื่อเข้าสู่โหมดรับทะเบียนรถหน้าจอ LCD บรรทัดแรกจะแสดงคำว่า “Code” และบรรทัดที่ 2 จะแสดงคำว่า “Time”

3.5 เมื่อหน้าจอแสดงผลตามข้อ 3.4 จึงทำการคีย์หมายเลขทะเบียนรถ 4 หลัก แล้วกด E รหัสที่บรรทัดแรกจะแสดงคำว่า “OK” และ “Time” ในบรรทัดที่ 2 จะแสดงเวลาเข้า

3.6 เมื่อหน้าจอแสดงผลตามข้อ 3.5 จึงทำการรูดบัตรบาร์โค้ดหน้าจอจะแสดงรหัสบาร์โค้ดประมาณ 3 วินาที แล้วจะแสดงตัว “K” ขึ้น และหายไปแสดงว่าข้อมูลได้ถูกเก็บเรียบร้อยแล้ว

3.7 ทำการกดหมายเลขรถคันต่อไป (ถ้าต้องการบันทึกทะเบียนรถคันต่อไป)

3.8 เมื่อต้องการนำรถออกให้ทำการกดตัวเลข “0000” แล้วกด E เพื่อเข้าสู่โหมด

3.9 กด M หน้าจอจะแสดงคำว่า “Mode” และทำการเลือกโหมดการทำงาน

3.10 เมื่อต้องการตั้งการปรับเปลี่ยนค่าเงินกด A เพื่อเข้าสู่โหมดการเปลี่ยนเงินบรรทัดที่ 2 ของจอ LCD จะแสดงค่า “E” ออกมาแล้วทำการเลือกราคาโดยจะมีราคาค่าเงินดังนี้
กด 1=10 บาทต่อชั่วโมง กด 2=20 บาทต่อชั่วโมง กด 3=30 บาทต่อชั่วโมง กด 4=40 บาทต่อชั่วโมง

3.11 เมื่อทำการเลือกราคาจอ LCD จะแสดงคำว่า “Mode” และกด C เพื่อเข้าสู่โหมดการนำรถออกหน้าจอ LCD จะแสดงคำว่า “Code” พร้อมทั้งจะทำการรูดบัตรเพื่อนำรถออก

3.12 ทำการรูดบัตรเพื่อนำรถออกหน้าจอ LCD จะแสดงทะเบียนรถแสดงเวลารถเข้า และรถออกโชว์เวลาที่จอดรถ และจำนวนเงิน

3.13 เมื่อต้องการนำรถคันต่อไปออกให้กด M หน้าจอจะแสดงคำว่า “Mode” แล้วทำตามข้อที่ 3.12

4. การแก้ปัญหาเบื้องต้น

เมื่อท่านประสบปัญหาในการใช้งานแบบจำลองระบบจอดรถสามารถตรวจสอบแนวทางการแก้ไขปัญหาเบื้องต้นได้จากตารางข้างล่างนี้

อาการ	สาเหตุและ/หรือวิธีแก้ไข
จอ LCD ไม่ติด, ตัวหัวอ่านบาร์โค้ด ไม่แสดงไฟการทำงาน	ตรวจสอบแหล่งจ่ายไฟ, ตรวจสอบขั้วเสียบของตัวอ่านบัตรบาร์โค้ด
จอ LCD ไม่ติด	กดปุ่มรีเซ็ตค่าเพื่อเริ่มต้นการทำงานใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การดูแลรักษาและข้อควรระวัง

5.1 การดูแลรักษา

- ตรวจสอบหัวอ่านบาร์โค้ดทุกครั้งหลังการใช้งาน และถอดเก็บทุกครั้งหลังการใช้งาน
- เช็ดทำความสะอาดหัวอ่านบาร์โค้ด และตัวเครื่องด้วยผ้านุ่มห้ามใช้สารใดๆ ที่เป็นตัว

ทำลายเพราะอาจทำให้ตัวเครื่องเป็นรอยเสียหาย

- ถอดปลั๊กทุกครั้งหลังการใช้งาน

5.2 ข้อควรระวัง

- ห้ามต่อหัวอ่านบาร์โค้ดขณะกำลังใช้งาน
- ควรทำตามขั้นตอนการทำงานทุกครั้ง และควรศึกษาคู่มือให้ละเอียดก่อนใช้งาน

6. ข้อมูลจำเพาะ

คุณสมบัติ	รายละเอียด
หลักการอ่านบาร์โค้ด	ใช้หลักการเปลี่ยนรหัสบาร์โค้ดให้เป็นรหัสแอสกีโดยใช้ตัวบาร์โค้ดเบอร์BSR-902M คีย์บอร์ด
โหมดเลือกการใช้งาน	แบ่งเป็น 3 โหมดคือโหมด A, โหมด B, โหมด C
ส่วนแสดงผล	จอแสดงผลแบบผลึกเหลวขนาด 16 x 2
การแสดงค่าการตอบรับของบาร์โค้ด	ใช้ LCD เป็นตัวแสดงผล
ความเที่ยงตรง	± 95 %
ความผิดพลาด	± 5 %
แหล่งจ่ายพลังงาน	ไฟฟ้ากระแสสลับ 220 โวลต์ ความถี่ 50-60 เฮิร์ตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข
รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



8xC251TA/TB/TP/TQ HIGH-PERFORMANCE CHMOS MICROCONTROLLER

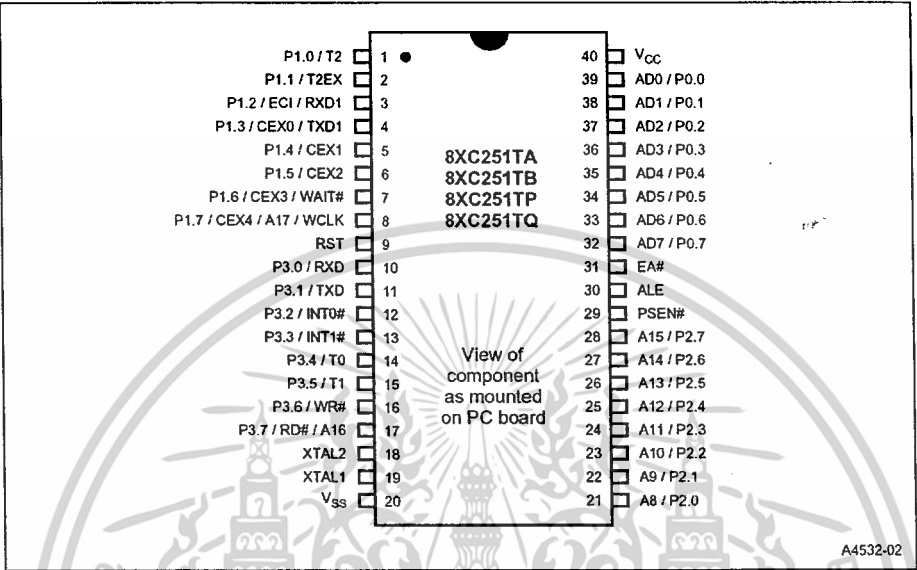


Figure 4. 8xC251TA/TB/TP/TQ 40-pin PDIP Packages

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Table 4. 8xC251TA/TB/TP/TQ Pin Assignment

PLCC	DIP	Name	PLCC	DIP	Name
1		V _{SS1}	23		V _{SS2}
2	1	P1.0/T2	24	21	A8/P2.0
3	2	P1.1/T2EX	25	22	A9/P2.1
4	3	P1.2/ECI/RXD1	26	23	A10/P2.2
5	4	P1.3/CEX0/TXD1	27	24	A11/P2.3
6	5	P1.4/CEX1	28	25	A12/P2.4
7	6	P1.5/CEX2	29	26	A13/P2.5
8	7	P1.6/CEX3/WAIT#	30	27	A14/P2.6
9	8	P1.7/CEX4/A17/WCLK	31	28	A15/P2.7
10	9	RST	32	29	PSEN#
11	10	P3.0/RXD	33	30	ALE
12		V _{CC2}	34		V _{SS2}
13	11	P3.1/TXD	35	31	EA#
14	12	P3.2/INT0#	36	32	AD7/P0.7
15	13	P3.3/INT1#	37	33	AD6/P0.6
16	14	P3.4/T0	38	34	AD5/P0.5
17	15	P3.5/T1	39	35	AD4/P0.4
18	16	P3.6/WR#	40	36	AD3/P0.3
19	17	P3.7/RD#/A16	41	37	AD2/P0.2
20	18	XTAL2	42	38	AD1/P0.1
21	19	XTAL1	43	39	AD0/P0.0
22	20	V _{SS}	44	40	V _{CC}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Table 5. 8xC251TA/TB/TP/TQ PLCC/DIP Pin Assignments Arranged by Functional Category

Address & Data			Input/Output		
Name	PLCC	DIP	Name	PLCC	DIP
AD0/P0.0	43	39	P1.0/T2	2	1
AD1/P0.1	42	38	P1.1/T2EX	3	2
AD2/P0.2	41	37	P1.2/ECI/RXD1	4	3
AD3/P0.3	40	36	P1.3/CEX0/TXD1	5	4
AD4/P0.4	39	35	P1.4/CEX1	6	5
AD5/P0.5	38	34	P1.5/CEX2	7	6
AD6/P0.6	37	33	P1.6/CEX3/WAIT#	8	7
AD7/P0.7	36	32	P1.7/CEX4/A17/WCLK	9	8
A8/P2.0	24	21	P3.0/RXD	11	10
A9/P2.1	25	22	P3.1/TXD	13	11
A10/P2.2	26	23	P3.4/T0	16	14
A11/P2.3	27	24	P3.5/T1	17	15
A12/P2.4	28	25	Power & Ground		
A13/P2.5	29	26	Name	PLCC	DIP
A14/P2.6	30	27	Vcc	44	40
A15/P2.7	31	28	Vcc2	12	
P3.7/RD#/A16	19	17	Vss	22	20
P1.7/CEX4/A17/WCLK	9	8	Vss1	1	
Processor Control			Vss2	23, 34	
Name	PLCC	DIP	EA#	35	31
P3.2/INT0#	14	12	Bus Control & Status		
P3.3/INT1#	15	13	Name	PLCC	DIP
EA#	35	31	P3.6/WR#	18	16
RST	10	9	P3.7/RD#/A16	19	17
XTAL1	21	18	ALE	33	30
XTAL2	20	19	PSEN#	32	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.0 SIGNALS

Table 6. Signal Descriptions (Sheet 1 of 3)

Signal Name	Type	Description	Alternate Function
A17	O	18th Address Bit (A17). Output to memory as 18th external address bit (A17) in extended bus applications, depending on the values of bits RD0 and RD1 in configuration byte UCONFIG0. See also RD# and PSEN#.	P1.7/CEX4/WCLK
A16	O	Address Line 16. See RD#.	RD#
A15:8 ¹	O	Address Lines. Upper address lines for the external bus.	P2.7:0
AD7:0 ¹	I/O	Address/Data Lines. Multiplexed lower address lines and data lines for external memory.	P0.7:0
ALE	O	Address Latch Enable. ALE signals the start of an external bus cycle and indicates that valid address information is available on lines A15:8 and AD7:0. An external latch can use ALE to demultiplex the address from the address/data bus.	
CEX4:0	I/O	Programmable Counter Array (PCA) Input/Output Pins. These are input signals for the PCA capture mode and output signals for the PCA compare mode and PCA PWM mode.	P1.6:4 P1.7/A17/ WAIT# P1.3/TXD1
EA#	I	External Access. Directs program memory accesses to on-chip or off-chip code memory. For EA# = 0, all program memory accesses are off-chip. For EA# = 1, an access is to on-chip ROM if the address is within the range of the on-chip ROM; otherwise the access is off-chip. The value of EA# is latched at reset. For devices without on-chip ROM, EA# must be strapped to ground.	
ECI	I	PCA External Clock Input. External clock input to the 16-bit PCA timer.	P1.2/RXD1
INT1:0#	I	External Interrupts 0 and 1. These inputs set bits IE1:0 in the TCON register. If bits IT1:0 in the TCON register are set, bits IE1:0 are set by a falling edge on INT1#/INT0#. If bits INT1:0 are clear, bits IE1:0 are set by a low level on INT1:0#.	P3.3:2
P0.7:0	I/O	Port 0. This is an 8-bit, open-drain, bidirectional I/O port.	AD7:0
P1.0 P1.1 P1.2 P1.7:3	I/O	Port 1. This is an 8-bit, bidirectional I/O port with internal pullups.	T2 T2EX ECI/RXD1 CEX3:1 CEX4/A17/ WAIT#/ WCLK CEX0/TXD1
P2.7:0	I/O	Port 2. This is an 8-bit, bidirectional I/O port with internal pullups.	A15:8
P3.0 P3.1 P3.3:2 P3.5:4 P3.6 P3.7	I/O	Port 3. This is an 8-bit, bidirectional I/O port with internal pullups.	RXD TXD INT1:0# T1:0 WR# RD#/A16
PSEN#	O	Program Store Enable. Read signal output. This output is asserted for a memory address range that depends on bits RD0 and RD1 in configuration byte UCONFIG0 (see RD#).	—

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



8xC251TA/TB/TP/TQ HIGH-PERFORMANCE CMOS MICROCONTROLLER

Table 6. Signal Descriptions (Sheet 2 of 3)

Signal Name	Type	Description	Alternate Function
RD#	O	Read or 17th Address Bit (A16). Read signal output to external data memory or 17th external address bit (A16), depending on the values of bits RD0 and RD1 in configuration byte UCONFIG0. (See PSEN#).	P3.7/A16
RST	I	Reset. Reset input to the chip. Holding this pin high for 64 oscillator periods while the oscillator is running resets the device. The port pins are driven to their reset conditions when a voltage greater than V_{IH1} is applied, whether or not the oscillator is running. This pin has an internal pulldown resistor, which allows the device to be reset by connecting a capacitor between this pin and V_{CC} . Asserting RST when the chip is in idle mode or powerdown mode returns the chip to normal operation.	—
RXD	I/O	Receive Serial Data. RXD sends and receives data in serial I/O mode 0 and receives data in serial I/O modes 1, 2, and 3.	P3.0
RXD1	I/O	Receive Serial Data 1. RXD1 sends and receives data in serial I/O mode 0 and receives data in serial I/O modes 1, 2, and 3 for the 2nd serial port.	P1.2/ECI
T1:0	I	Timer 1:0 External Clock Inputs. When timer 1:0 operates as a counter, a falling edge on the T1:0 pin increments the count.	P3.5:4
T2	I/O	Timer 2 Clock Input/Output. For the timer 2 capture mode, this signal is the external clock input. For the clock-out mode, it is the timer 2 clock output.	P1.0
T2EX	I	Timer 2 External Input. In timer 2 capture mode, a falling edge initiates a capture of the timer 2 registers. In auto-reload mode, a falling edge causes the timer 2 registers to be reloaded. In the up-down counter mode, this signal determines the count direction: 1 = up, 0 = down.	P1.1
TXD	O	Transmit Serial Data. TXD outputs the shift clock in serial I/O mode 0 and transmits serial data in serial I/O modes 1, 2, and 3.	P3.1
TXD1	O	Transmit Serial Data 1. TXD1 outputs the shift clock in serial I/O mode 0 and transmits serial data in serial I/O modes 1, 2, and 3 for the 2nd serial port.	P1.3/CEX0
V_{CC}	PWR	Supply Voltage. Connect this pin to the +5V supply voltage.	—
V_{CC2}	PWR	Secondary Supply Voltage 2. This supply voltage connection is provided to reduce power supply noise. Connection of this pin to the +5V supply voltage is recommended. However, when using the 8XC251SB as a pin-for-pin replacement for the 8XC51FX, V_{SS2} can be unconnected without loss of compatibility. (Not available on DIP)	—
V_{SS}	GND	Circuit Ground. Connect this pin to ground.	—
V_{SS1}	GND	Secondary Ground. This ground is provided to reduce ground bounce and improve power supply bypassing. Connection of this pin to ground is recommended. However, when using the 8xC251TA/TB/TP/TQ as a pin-for-pin replacement for the 8XC51BH, V_{SS1} can be unconnected without loss of compatibility. (Not available on DIP)	—
V_{SS2}	GND	Secondary Ground 2. This ground is provided to reduce ground bounce and improve power supply bypassing. Connection of this pin to ground is recommended. However, when using the 8xC251TA/TB/TP/TQ as a pin-for-pin replacement for the 8XC51FX, V_{SS2} can be unconnected without loss of compatibility. (Not available on DIP)	—

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Table 6. Signal Descriptions (Sheet 3 of 3)

Signal Name	Type	Description	Alternate Function
WAIT#	I	Real-time Wait State Input. The real-time WAIT# input is enabled by writing a logical '1' to the WCON.0 (RTWE) bit at S:A7H. During bus cycles, the external memory system can signal 'system ready' to the microcontroller in real time by controlling the WAIT# input signal on the port 1.6 input.	P1.6/CEX3
WCLK	O	Wait Clock Output. The real-time WCLK output is driven at port 1.7 (WCLK) by writing a logical '1' to the WCON.1 (RTWCE) bit at S:A7H. When enabled, the WCLK output produces a square wave signal with a period of one-half the oscillator frequency.	P1.7/CEX4/A17
WR#	O	Write. Write signal output to external memory.	P3.6
XTAL1	I	Input to the On-chip, Inverting, Oscillator Amplifier. To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, its output is connected to this pin. XTAL1 is the clock source for internal timing.	—
XTAL2	O	Output of the On-chip, Inverting, Oscillator Amplifier. To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, leave XTAL2 unconnected.	—

NOTE:
The descriptions of A15:8/P2.7:0 and AD7:0/P0.7:0 are for the nonpage-mode chip configuration (compatible with 44-pin PLCC and 40-pin DIP MCS 51 microcontrollers). If the chip is configured for page-mode operation, port 0 carries the lower address bits (A7:0), and port 2 carries the upper address bits (A15:8) and the data (D7:0).

Table 7. Memory Signal Selections (RD1:0)

RD1:0	P1.7/CEX/A17/WCLK	P3.7/RD#/A16	PSEN#	WR#	Features
0 0	A17	A16	Asserted for all addresses	Asserted for writes to all memory locations	256-Kbyte external memory
0 1	P1.7/CEX4/WCLK	A16	Asserted for all addresses	Asserted for writes to all memory locations	128-Kbyte external memory
1 0	P1.7/CEX4/WCLK	P3.7 only	Asserted for all addresses	Asserted for writes to all memory locations	64-Kbyte external memory. One additional port pin.
1 1	P1.7/CEX4/WCLK	RD# asserted for addresses ≤ 7F:FFFFH	Asserted for ≥ 80:0000H	Asserted only for writes to MCS 51 microcontroller data memory locations.	64-Kbyte external memory. Compatible with MCS 51 microcontrollers.



5.0 ADDRESS MAP

Table 8. 8xC251TA/TB/TP/TQ Address Map

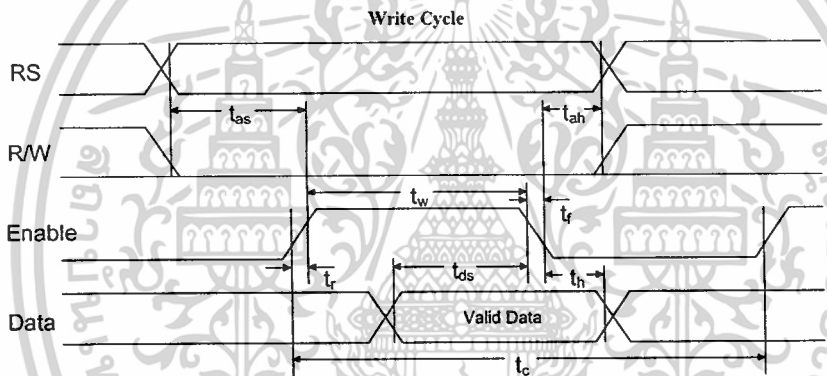
Internal Address)	Description	Notes
FF:FFFFH FF:4000H	External Memory except the top eight bytes (FF:FFF8H–FF:FFFFH) which are reserved for the configuration array.	1, 3, 10
FF:3FFFH FF:0000H	External memory or on-chip nonvolatile memory (8Kbytes FF:0000H - FF:1FFFH, 16Kbytes FF:0000H - FF:3FFFH).	3, 4, 5
FE:FFFFH FE:0000H	External Memory	3
FD:FFFFH 02:0000H	Reserved	6
01:FFFFH 01:0000H	External Memory	3
00:FFFFH 00:E000H	External memory or with configuration bit EMAP# = 0, addresses in this range access on-chip code memory in region FF: (16 Kbyte devices only).	5, 7
00:DFFFH 00:0420H	External Memory	7
00:041FH 00:0080H	On-chip RAM (512 bytes 00:0020H - 00:021FH, 1024 bytes 00:0020H - 00:041FH)	7
00:007FH 00:0020H	On-chip RAM	8
00:001FH 00:0000H	Storage for R0–R7 of Register File	2, 9

NOTES:

1. 18 address lines are bonded out (A15:0, A16:0, or A17:0 selected during chip configuration).
2. The special function registers (SFRs) and the register file have separate internal address spaces.
3. Data in this area is accessible by indirect addressing only.
4. Devices reset into internal or external starting locations depending on the state of EA# and configuration byte information. See EA#.
5. The 16-Kbyte ROM devices allow internal locations FF:2000H–FF:3FFFH to map into region 00:. In this case, if EA# = 1, a data read to 00:E000H–00:FFFFH is redirected to internal ROM (see bit 1 in UCONF0). This is not available for 8-Kbyte ROM devices.
6. This reserved area returns indeterminate values.
7. Data is accessible by direct and indirect addressing.
8. Data is accessible by direct, indirect, and bit addressing.
9. Data is accessible by direct, indirect, and register addressing.
10. Eight addresses at the top of all external memory maps are reserved for current and future device configuration byte information.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

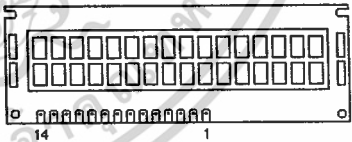
Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
NOP	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Cursor Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero. returns shifted display to original position. DDRAM contents remains unchanged.	3
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction, and specifies automatic shift.	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking(B).	3
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
Function Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width(DL), number of display lines (N,M) and voltage generator control (G).	3
Set CGRAM Addr	0	0	0	1	Character Generator RAM						Sets CGRAM Address	3
Set DDRAM Addr	0	0	1	Display Data RAM Address						Sets DDRAM Address	3	
Busy Flag & Addr	0	1	BF	Address Counter						Reads Busy Flag & Address Counter	0	
Read Data	1	0	Read Data						Reads data from CGRAM or DDRAM		3	
Write Data	1	1	Write Data						Writes data from CGRAM or DDRAM		3	



Parameter	Symbol	Min ⁽¹⁾	Typ ⁽¹⁾	Max ⁽¹⁾	Unit
Enable Cycle Time	t _c	500	-	-	ns
Enable Pulse Width (High)	t _w	230	-	-	ns
Enable Rise/Fall Time	t _r ,t _f	-	-	20	ns
Address Setup Time	t _{as}	40	-	-	ns
Address Hold Time	t _{ah}	10	-	-	ns
Data Setup Time	t _{ds}	80	-	-	ns
Data Hold Time	t _h	10	-	-	ns

Note ¹ The above specifications are a indication only. Timing will vary from manufacturer to manufacturer.

Note ² A 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x 16 character, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x 20 character, 2 lines x 40 character etc. modules compatible with the HD44780 LCD Module.



Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	Vo	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

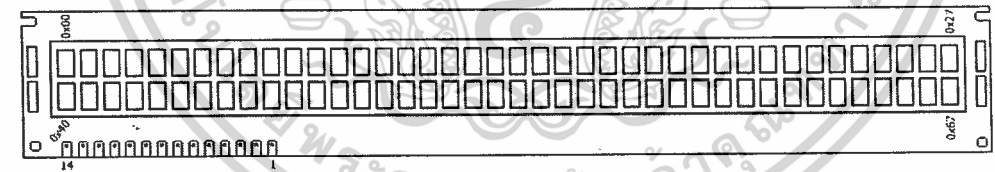
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Extended Concise LCD Data Sheet for HD44780

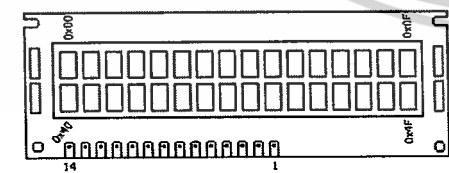
Version: 25.6.1999

Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clock-Cycles
NOP	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clear display & set address counter to zero	165
Cursor Home	0	0	0	0	0	0	0	0	1	x	Set address counter to zero, return shifted display to original position. DD RAM contents remains unchanged.	3
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Set cursor move direction (I/D) and specify automatic display shift (S).	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turn display (D), cursor on/off (C), and cursor blinking (B).	3
Cursor / Display shift	0	0	0	0	0	1	S/C	R/L	x	x	Shift display or move cursor (S/C) and specify direction (R/L).	3
Function Set	0	0	0	0	1	DL	N	F	x	x	Set interface data width (DL), number of display lines (N) and character font (F).	3
Set CGRAM Address	0	0	0	1	CGRAM Address						Set CGRAM address. CGRAM data is sent afterwards.	3
Set DDRAM Address	0	0	1	DDRAM Address						Set DDRAM address. DDRAM data is sent afterwards.	3	
Busy Flag & Address	0	1	BF	Address Counter						Read busy flag (BF) and address counter	0	
Write Data	1	0	Data						Write data into DDRAM or CGRAM	3		
Read Data	1	1	Data						Read data from DDRAM or CGRAM	3		
x : Don't care												
I/D	1	0	Increment Decrement					R/L	1	0	Shift to the right Shift to the left	
	S	1	Automatic display shift					DL	1	0	8 bit interface 4 bit interface	
	D	1	Display ON Display OFF					N	1	0	2 lines 1 line	
	C	1	Cursor ON Cursor OFF					F	1	0	5x10 dots 5x7 dots	
	B	1	Cursor blinking					DDRAM : Display Data RAM CGRAM : Character Generator RAM				
	S/C	1	Display shift Cursor move									

LCD Display with 2 lines x 40 characters :



LCD Display with 2 lines x 16 characters :

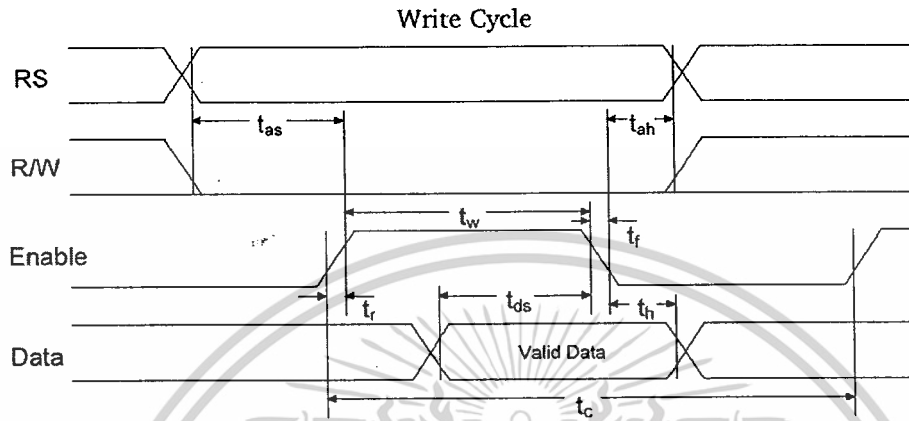


Pin No	Name	Function	Description
1	Vss	Power	GND
2	Vdd	Power	+ 5 V
3	Vee	Contrast Adj.	(-2) 0 - 5 V
4	RS	Command	Register Select
5	R/W	Command	Read / Write
6	E	Command	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bus Timing Characteristics

($T_a = -20$ to $+75^\circ\text{C}$)



Write-Cycle	V_{DD}	2.7 - 4.5 V ⁽²⁾	4.5 - 5.5 V ⁽²⁾	2.7 - 4.5 V ⁽²⁾	4.5 - 5.5 V ⁽²⁾	
Parameter	Symbol	Min ⁽¹⁾	Typ ⁽¹⁾	Max ⁽¹⁾		Unit
Enable Cycle Time	t_c	1000	500	-	-	ns
Enable Pulse Width (High)	t_w	450	230	-	-	ns
Enable Rise/Fall Time	t_r, t_f	-	-	25	20	ns
Address Setup Time	t_{as}	60	40	-	-	ns
Address Hold Time	t_{ah}	20	10	-	-	ns
Data Setup Time	t_{ds}	195	80	-	-	ns
Data Hold Time	t_{dh}	10	10	-	-	ns

(1) The above specifications are indications only (based on Hitachi HD44780). Timing will vary from manufacturer to manufacturer.

(2) Power Supply : HD44780 S : $V_{DD} = 4.5 - 5.5 \text{ V}$
 HD44780 U : $V_{DD} = 2.7 - 5.5 \text{ V}$

This data sheet refers to specifications for the Hitachi HD44780 LCD Driver chip, which is used for most LCD modules.

Common types are :

1 line x 20 characters
 2 lines x 16 characters
 2 lines x 20 characters
 2 lines x 40 characters
 4 lines x 20 characters
 4 lines x 40 characters

© 1998/1999 by

Craig Peacock, Australia
 Peter Luethi, Switzerland

<http://www.beyondlogic.org>
<http://www.electronic-engineering.ch>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายเจตน์กาญจน์ อินทร์ม่วง
วัน เดือน ปีเกิด	14 มกราคม พ.ศ. 2526
ภูมิลำเนา	8/24 ถ.ศรีมาลา ตำบลในเมือง อำเภอเมือง จังหวัดพิจิตร 66000 โทรศัพท์ 0-6549-8237
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนอนุบาลพิจิตร
มัธยมศึกษาตอนต้น	โรงเรียนพิจิตรพิทยาคม
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคพิจิตร
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคพิจิตร
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
ผลงานที่ได้รับรางวัล	รางวัลชมเชยการประกวดสิ่งประดิษฐ์ของคนรุ่นใหม่ระดับชาติ ปี 2544 เยาวชนดีเด่นด้านบำเพ็ญประโยชน์ ประจำปี 2545
ทุนการศึกษา	ทุนสำนักงานคณะกรรมการส่งเสริมและประสานงานเยาวชน แห่งชาติ
คติพจน์	ไม่ว่าจะเดินช้าแค่ไหน ขอเพียงอย่าคิดที่จะหยุดเดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายเด่นศักดิ์ อินตาคำ
วัน เดือน ปีเกิด	8 กันยายน พ.ศ. 2525
ภูมิลำเนา	17 หมู่ 1 ตำบลสบปราบ อำเภอสบปราบ จังหวัดลำปาง 52170 โทรศัพท์ 0-4098-7190
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบ้านกล้วย
มัธยมศึกษาตอนต้น	โรงเรียนเคนเน็ตแม็คเคนซี จังหวัดลำปาง
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคลำปาง
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคลำปาง
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	ทำวันนี้ให้ดีกว่าเมื่อวาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายวีระพล รื่องครอง
วัน เดือน ปีเกิด	4 พฤษภาคม พ.ศ. 2525
ภูมิลำเนา	124 หมู่ 8 ตำบลโพธิ์งาม อำเภอประจันตคาม จังหวัดปราจีนบุรี 25130 โทรศัพท์ 0-1001-9750
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดโพธิ์งาม
มัธยมศึกษาตอนต้น	โรงเรียนประจันตราษฎ์บำรุง จังหวัดปราจีนบุรี
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคปราจีนบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคปราจีนบุรี
ปริญญาตรี	สาขาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.
คติพจน์	ปัจจุบันทำดีอนาคตได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้