

ตารางแสดงเวลาการทำงานโปรแกรมได้

PROGRAMABLE WORKING TIME SCHEDULE BY LCD DISPLAY



เลขที่.....
เลขทะเบียน.....42268
วัน, เดือน, ปี.....16 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงเวลาการทำงานโปรแกรมได้
PROGRAMABLE WORKING TIME SCHEDULE



ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิญญานี้พิมพ์ ปีการศึกษา 2543

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตารางแสดงเวลาการทำงานโปรแกรมได้

ผู้จัดทำ

1. นาย อุทวน บุญลอย รหัส 40011020

2. น.ส. อุษารมย์ พงษ์ารักษ์ รหัส 40011023

3. นาย เอกภพ โอตาการ รหัส 40011035



อาจารย์ที่ปรึกษา

(ศศ. สุชาติ คุณทวีเทพ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงเวลาการทำงาน โปรแกรมได้

นาย อุเทน บุญถอย

น.ศ. อุยารมย์ พงษ์ารักษ์

นาย เอกภพ โอตาการ

ผศ. สุชาติ คุณทวีเทพ (อาจารย์ที่ปรึกษา)

ปีการศึกษา 2543

บทคัดย่อ

โครงการนี้เป็นการนำเสนอการประยุกต์ใช้งานบอร์ดแสดงข้อมูลข่าวสารมาใช้ในการแสดงเวลาการทำงานของอาจารย์ โดยจะแสดงสถานที่ที่สามารถติดต่ออาจารย์ได้ในช่วงเวลาต่าง ๆ ของแต่ละวัน สามารถโปรแกรมเก็บข้อมูลล่วงหน้าได้ทั้งปีการศึกษา โดยใช้ไมโครคอนโทรลเลอร์ MCS-51 ควบคุมการทำงาน หน่วยความจำเป็น RAM แสดงผลผ่านทางจอแสดงผล LCD (Liquid Crystal Display) รับส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 รวมทั้งเปลี่ยนข้อมูลได้ โดยผ่านโปรแกรม ซึ่งสร้างมาจาก โปรแกรม Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMABLE WORKING TIME SCHEDULE

Mr. Utain Boonloy

Miss Usarom Pongsarak

Mr. Akkapob Otakarn

Assist.Prof.Suchart Khuntaweethep

Educational Year 2000

Abstract

This thesis presents the adaptation of Information Board for showing lecturer 's working time schedule , which will show places where students can contact in each period of the day throughout the whole academic year. The system is controlled by Microcontroller MCS-51 , the memory is RAM , and the output is presented by LCD . The data is interfaced through serial port RS-232 and can be changed through the program that is composed by Visual Basic .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำขอขอบพระคุณ อ. สุชาติ คุณทวีภาพ ที่ให้ความกรุณา ให้คำแนะนำปรึกษา ช่วยเหลือมาโดยตลอด รวมทั้งอาจารย์ท่านอื่น ๆ ที่ได้คอยชี้แนะ ให้ความรู้ รวมทั้งเพื่อนๆ ที่ได้คอยให้กำลังใจ

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
สารบัญตาราง	ณ
บทที่ 1 บทนำ	1
1.1 บทนำ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 คุณสมบัติของ LCD	3
2.2 โครงสร้างของ MCS-51	5
2.3 สัญญาณนาฬิกาแบบอนุกรม	10
2.4 หน่วยความจำข้อมูลภายนอก RAM	13
2.5 การเชื่อมต่อ 8255 กับไมโครคอนโทรลเลอร์	14
2.6 การสื่อสารแบบอนุกรม	15
บทที่ 3 การออกแบบ	20
3.1 หน่วยประมวลผล	20
3.2 หน่วยความจำ	21
3.3 จอแสดงผล	21
3.4 สัญญาณนาฬิกา	22
3.5 สวิตช์	23
3.6 ส่วนติดต่อกับคอมพิวเตอร์	23
บทที่ 4 โปรแกรม Visual Basic	25
บทที่ 5 โปรแกรมการทำงานและโครงสร้างของโครงการ	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 6 การทดลองและสรุปผล	45
6.1 ขั้นตอนการทดลอง	45
6.2 สรุปผลการทำงาน	53
6.3 แนวทางการพัฒนา	53
ภาคผนวก ก	A
ภาคผนวก ข	B
ภาคผนวก ค	C
หนังสืออ้างอิง	D



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมของ โครงการงาน	2
รูปที่ 2.1 แสดงการต่อโมดูล LCD ร่วมกับไมโครคอนโทรลเลอร์ 8051	5
รูปที่ 2.2 แสดงโครงสร้างภายในของ MCS-51	6
รูปที่ 2.3 บล็อกไดอะแกรมแสดงหน่วยทำงานพื้นฐานของ MCS-51	7
รูปที่ 2.4 ตำแหน่งและการกำหนดหน้าที่ขาสัญญาณ ไอซี AT89C51	7
รูปที่ 2.5 แสดงการใช้คริสตอลภายนอกต่อเข้ากับวงจรรอสซิทเลเตอร์ภายใน	9
รูปที่ 2.6 แสดงลักษณะรูปร่างทั้งสองแบบของ ไอซีเบอร์ DS1302	10
รูปที่ 2.7 โครงสร้างภายในของ RTC DS1302	11
รูปที่ 2.8 Block Diagram แสดงขาสัญญาณของหน่วยความจำ RAM แบบสแตติก	13
รูปที่ 2.9 วงจรแสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกกับ ไมโครคอนโทรลเลอร์	14
รูปที่ 2.10 แสดงโครงสร้างของไอซี 8255	15
รูปที่ 2.11 รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส	16
รูปที่ 2.12 การจัดขาของคอนเนกเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9 และ DB-25	17
รูปที่ 2.10 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ	18
รูปที่ 3.1 ลักษณะการนำไปใช้งานของ AT89C51	20
รูปที่ 3.2 วงจรเชื่อมต่อหน่วยความจำข้อมูลภายนอกกับไมโครคอนโทรลเลอร์	21
รูปที่ 3.3 วงจรการอินเตอร์เฟส LCD กับไมโครคอนโทรลเลอร์	22
รูปที่ 3.4 ลักษณะการนำไปใช้งานของ DS1302	22
รูปที่ 3.5 วงจรเชื่อมต่อกับคอมพิวเตอร์	23
รูปที่ 3.6 วงจรรวม	24
รูปที่ 4.1 หน้าต่าง LOGIN	25
รูปที่ 4.2 หน้าต่างเตือน USERNAME ยังไม่ได้ลงทะเบียน	25
รูปที่ 4.3 หน้าต่าง ADD USER	26
รูปที่ 4.4 หน้าต่างเตือน USERNAME ซ้ำ	26
รูปที่ 4.5 หน้าต่างเตือน PASSWORD ไม่ถูกต้อง	26
รูปที่ 4.6 หน้าต่างเตือนแสดงการเพิ่มชื่อเสร็จเรียบร้อยแล้ว	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.7 หน้าต่าง DELETE USER	27
รูปที่ 4.8 หน้าต่างแสดงการลบชื่อทิ้งเรียบร้อยแล้ว	27
รูปที่ 4.9 หน้าต่าง CHANGE PASSWORD	28
รูปที่ 4.10 หน้าต่างแสดงการเปลี่ยนรหัสผ่านเรียบร้อยแล้ว	28
รูปที่ 4.11 หน้าต่าง YEAR SETTING	28
รูปที่ 4.12 หน้าต่างเตือนป้อนข้อมูลผิด	29
รูปที่ 4.13 หน้าต่างแสดงกำหนดค่าปีเรียบร้อยแล้ว	29
รูปที่ 4.14 หน้าต่างเตือน ไม่ได้ใส่ค่าปี	29
รูปที่ 4.15 หน้าต่าง SCHEDULE	30
รูปที่ 4.16 หน้าต่างแสดงบันทึกข้อมูลเรียบร้อยแล้ว	30
รูปที่ 4.17 หน้าต่างคำถามบันทึกข้อมูลหรือไม่	31
รูปที่ 4.18 หน้าต่างคำถามต้องการลงข้อมูลหรือไม่	31
รูปที่ 4.19 หน้าต่างแสดงลบข้อมูลเดิมเรียบร้อยแล้ว	31
รูปที่ 4.20 FLOWCHART การทำงานของ โปรแกรม VISUAL BASIC	32
รูปที่ 4.21 FLOWCHART การทำงานของ โปรแกรมย่อย ADD	33
รูปที่ 4.22 FLOWCHART การทำงานของ โปรแกรมย่อย DELETE	34
รูปที่ 4.23 FLOWCHART การทำงานของ โปรแกรมย่อย CHANGE PASSWORD	35
รูปที่ 4.24 FLOWCHART การทำงานของ โปรแกรมย่อย OK	36
รูปที่ 4.25 FLOWCHART การทำงานของ โปรแกรมย่อย YEAR SETTING	37
รูปที่ 4.26 FLOWCHART การทำงานของ โปรแกรมหลัก	38
รูปที่ 5.1 FLOWCHART การทำงานของ โปรแกรมหลัก	39
รูปที่ 5.2 FLOWCHART โปรแกรมย่อยการรับข้อมูลจากคอมพิวเตอร์ ผ่านทาง RS-232	40
รูปที่ 5.3 FLOWCHART โปรแกรมย่อยการแสดงผลตารางการทำงานใน 1 วัน	41
รูปที่ 5.4 FLOWCHART โปรแกรมย่อยการทำงานของสวิทช์	42
รูปที่ 5.5 ภาพรวมของชิ้นงาน	43
รูปที่ 5.6 วงจรภายในของโครงงาน	44
รูปที่ 6.1 หน้าต่างแรกของ โปรแกรม VISUAL BASIC	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 6.2 เพิ่มชื่อผู้ใช้งาน	46
รูปที่ 6.3 แจ้งการเพิ่มชื่อเรียบริอย	46
รูปที่ 6.4 ลงทะเบียนเข้าโปรแกรมเปลี่ยนแปลงข้อมูล	46
รูปที่ 6.5 หน้าต่างข้อมูลวันจันทร์	47
รูปที่ 6.6 หน้าต่างข้อมูลวันอังคาร	47
รูปที่ 6.7 หน้าต่างข้อมูลวันพุธ	48
รูปที่ 6.8 หน้าต่างข้อมูลวันพฤหัสบดี	48
รูปที่ 6.9 หน้าต่างข้อมูลวันศุกร์	49
รูปที่ 6.10 หน้าต่างข้อมูลวันเสาร์	49
รูปที่ 6.11 หน้าต่างข้อมูลวันอาทิตย์	50
รูปที่ 6.12 ชิงงานแสดงข้อมูลวันจันทร์	50
รูปที่ 6.13 ชิงงานแสดงข้อมูลวันอังคาร	51
รูปที่ 6.14 ชิงงานแสดงข้อมูลวันพุธ	51
รูปที่ 6.15 ชิงงานแสดงข้อมูลวันพฤหัสบดี	52
รูปที่ 6.16 ชิงงานแสดงข้อมูลวันศุกร์	52
รูปที่ 6.12 ชิงงานแสดงข้อมูลวันเสาร์	53
รูปที่ 6.12 ชิงงานแสดงข้อมูลวันอาทิตย์	53
รูปที่ 6.12 ชิงงานแสดง ไม่มีข้อมูลที่จะแสดง	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงตำแหน่งขาต่างๆที่ใช้เชื่อมต่อกับ LCD	4
ตารางที่ 2.2 แสดงไมโครคอนโทรลเลอร์MCS-51 เบอร์ต่างๆ	6
ตารางที่ 2.3 แสดงบิตและหน้าที่ต่างๆของพอร์ต 3	8
ตารางที่ 6.1 ตารางการทำงาน	45



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 บทนำ

เนื่องจากในปัจจุบัน เทคโนโลยีทางด้านสารสนเทศเข้ามามีบทบาทเป็นอย่างมากในชีวิตประจำวัน เพราะปัจจุบันเป็นโลกของข่าวสารข้อมูลที่เป็นหนึ่งเดียวกันทั่วโลก โดยเฉพาะอย่างยิ่งด้านธุรกิจและการโฆษณา ดังนั้นในปัจจุบันจึงต้องการสื่อที่สามารถแสดงข่าวสารข้อมูล รวมทั้งภาพเคลื่อนไหวต่าง ๆ เพื่อให้สามารถดึงดูดความสนใจ และสามารถสื่อสารให้ทุกคนได้เข้าใจในข้อมูลข่าวสารนั้น ๆ

กระดานหรือแผ่นป้ายแสดงข่าวสารอิเล็กทรอนิกส์ (Display Board) เป็นอีกสื่อที่นิยมใช้กันมากในปัจจุบัน สามารถพบได้ตามแหล่งชุมชน ย่านธุรกิจ ศูนย์การค้า โรงพยาบาลและตามบริษัทห้างร้านต่าง ๆ ซึ่งในอดีตยังคงใช้ระบบการแสดงผลแบบเซเวนเซกเมนต์ (7-segment) ต่อมามีการพัฒนาเป็นแบบจุดแสดงผล (Dot matrix Display) และจอแสดงผล LCD (Liquid Crystal Display) ซึ่งมีความละเอียดของภาพมากกว่าเดิม

จอแสดงผล LCD ได้ถูกนำมาใช้อย่างแพร่หลาย เช่น เครื่องคิดเลข นาฬิกา โทรศัพท์มือถือ เครื่องใช้ไฟฟ้าต่าง ๆ การแสดงผลมีทั้งแบบตัวอักษรภาษาไทย ภาษาอังกฤษ รวมทั้งภาพกราฟิกต่าง ๆ นอกจากนี้ยังสามารถตัดแปดรูปแบบการแสดงผลแบบอินเวอร์ส โดยใช้สีพื้น (Background) เป็นสีต่าง ๆ แล้วให้ภาพที่ต้องการเป็นดำ หรือดำนั่นเอง

1.2 วัตถุประสงค์ของโครงการ

บอร์ดแสดงตารางเวลาการทำงาน เป็นการประยุกต์การใช้แผ่นป้ายแสดงข้อมูลข่าวสารมาใช้กับชีวิตประจำวัน แสดงตารางเวลาการทำงานของอาจารย์ที่สามารถโปรแกรมได้ โดยการนำระบบฐานข้อมูลมาใช้ สามารถเปลี่ยนข้อมูลผ่านทางคอมพิวเตอร์ สะดวกแก่ผู้ใช้งาน และเพื่อสะดวกกับผู้มาติดต่อพบอาจารย์ จะได้ทราบว่าในช่วงเวลานี้สามารถพบอาจารย์ได้ที่ไหน และสามารถดูข้อมูลของวันอื่น ๆ ได้ด้วย

1.3 ขอบเขตของโครงการ

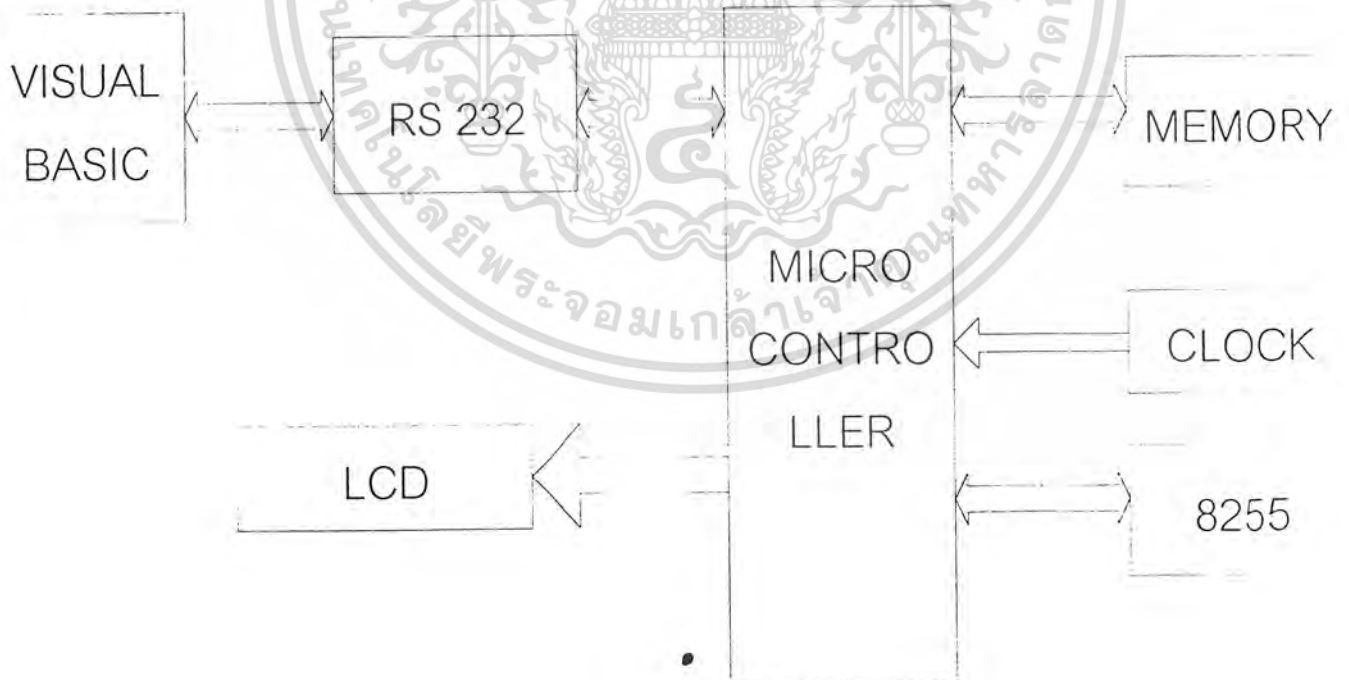
โครงการบอร์ดแสดงตารางเวลาการทำงานแบบโปรแกรมได้ มีส่วนประกอบที่เป็นทั้งฮาร์ดแวร์และซอฟต์แวร์ ดังต่อไปนี้

ส่วนที่เป็นฮาร์ดแวร์ประกอบด้วย

- ส่วนเก็บข้อมูล
- ส่วนประมวลผลข้อมูล
- ส่วนติดต่อกับคอมพิวเตอร์
- ส่วนแสดงผลข้อมูล
- ส่วนสร้างสัญญาณนาฬิกา
- ส่วนของ 8255

ส่วนที่เป็นซอฟต์แวร์ประกอบด้วย

- ฟังก์ชันเชื่อมต่อคอมพิวเตอร์
- ฟังก์ชันการแสดงผล
- ฟังก์ชันการเก็บค่า
- ฟังก์ชันสร้างสัญญาณนาฬิกา



รูปที่ 1.1 บล็อกไดอะแกรมของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎี

2.1 คุณสมบัติของ LCD

LCD (Liquid Crystal Display) เป็นจอแสดงผลที่สามารถแสดงข้อความที่เป็นตัวเลข, ตัวอักษรและสัญลักษณ์อื่นๆ ปัจจุบันเป็นที่นิยมใช้กันอย่างมาก ไม่ว่าจะเป็นเครื่องมือวัดคุมต่าง ๆ เครื่องคอมพิวเตอร์ เเลเซอร์ปริ้นเตอร์ และงานทางอิเล็กทรอนิกส์อื่นๆ สาเหตุที่มีการนำ LCD ไปใช้งานกันมากเนื่องจาก ความสะดวกความสมบูรณ์ของข้อความ

คอนโทรลเลอร์ HD44780 เป็นคอนโทรลเลอร์แบบยี่ดัดคิวหน้าขนาด 80 ขาจากฮิตาชิ (Hitachi) ทำหน้าที่ควบคุมการทำงานต่างๆ ให้จอ LCD ซึ่งได้รับความนิยมเป็นอย่างมาก HD44780 มีสัญญาณติดต่อกับภายนอกเพียง 14 เส้น

แผงของ LCD จะประกอบด้วยเซกเมนต์แสดงผลขนาดเล็กจำนวนมากในเซกเมนต์จะบรรจุชั้นของเหลวเป็นแผ่นบางๆ อยู่ระหว่างชั้นของแก้ว ของเหลวนี้เป็นสารประกอบทำงานโดยอาศัยพลังงานไฟฟ้ามาควบคุมการทำงานหรือการแสดงผลของ LCD เกิดขึ้นเนื่องจากการควบคุมแรงดันที่ตกคร่อมตัวมัน เช่น ถ้าให้แรงดันที่ตกคร่อมเซกเมนต์ ก็จะเกิดเป็นสีค่าหรือทึบแสง แต่ถ้าเอาแรงดันนั้นออก เซกเมนต์นั้นจะสว่างหรือโปร่งแสง ด้วยวิธีการจ่ายแรงดัน และงดจ่ายแรงดันนี้ก็เพียงพอที่จะควบคุมการแสดงผลตัวเลข,ตัวอักษร,สัญลักษณ์ต่างๆ ได้และสาเหตุที่จอ LCD ใช้แรงดันควบคุม ดังนั้นจึงกินกำลังงานต่ำ และมีขนาดเล็กและแบนราบ

โมดูลของ LCD บางรุ่นอาจจะมีหนึ่งแถว หรือมากกว่า การแสดงผลของจอ LCD อยู่ในรูปเมตริก เช่นบางรุ่นแสดงเมตริกซ์ที่มีขนาดกว้าง 5 เซกเมนต์ สูง 8 เซกเมนต์ และสำหรับรุ่น HD44780 สามารถควบคุมการทำงานแสดงผลได้สูงถึง 11 เซกเมนต์ ซึ่งเป็นผลดีในการแสดงตัวอักษรบางตัวเช่น g, p และ q

จอแสดงผล LCD จะใช้ไฟเลี้ยง +5 โวลต์ ป้อนให้ขาที่สอง ส่วนขาสามต่อเพื่อปรับมุมมองการแสดงผลให้เหมาะสม ทั้งนี้ขึ้นอยู่กับผลของแสงในขณะนั้นด้วยรวมไปถึงการติดตั้งและอุณหภูมิ ในตารางที่ 2.1 แสดงขาสัญญาณต่างๆที่ใช้ในการเชื่อมต่อกับไมโครคอนโทรลเลอร์ เนื่องจากการควบคุม LCD ต้องการเวลาเพื่อรอทำงานตามคำสั่ง หรือรับสัญญาณดังนั้นการใช้งานกับคอมพิวเตอร์จะต้องพิจารณาเรื่องเวลาด้วย แต่การเชื่อมต่อกับไมโครคอนโทรลเลอร์ หรือไมโครโปรเซสเซอร์สามารถต่อกันได้ตรงๆ ได้ไม่ต้องมีอุปกรณ์เพิ่ม หรือถ้ามีก็มีเพียงเล็กน้อยเท่านั้น

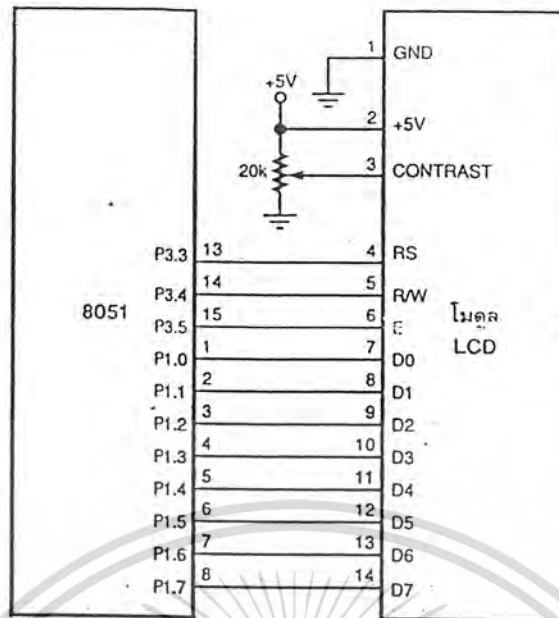
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงตำแหน่งขาต่างๆที่ใช้เชื่อมต่อกับ LCD

ขา	สัญลักษณ์	ฟังก์ชัน
1	Vss	กราวนด์
2	Vdd	+5 โวลต์
3	V0	ปรับความสว่างด้วยแรงดัน (0 - 5 โวลต์)
4	RS	เลือก رجิสเตอร์ (0 = رجิสเตอร์ คำสั่งหรือเฟล็กแสดงสถานะการทำงานและคัวนับแอดเดรส ; 1= رجิสเตอร์ ค่าตัว)
5	RW	เลือกการอ่านหรือเขียน
6	E	Enable การอ่านหรือเขียน LCD
7	D0	ค่าข้อมูล/เอาต์พุตบิตที่ 0
8	D1	ค่าข้อมูล/เอาต์พุตบิตที่ 1
9	D2	ค่าข้อมูล/เอาต์พุตบิตที่ 2
10	D3	ค่าข้อมูล/เอาต์พุตบิตที่ 3
11	D4	ค่าข้อมูล/เอาต์พุตบิตที่ 4
12	D5	ค่าข้อมูล/เอาต์พุตบิตที่ 5
13	D6	ค่าข้อมูล/เอาต์พุตบิตที่ 6
14	D7	ค่าข้อมูล/เอาต์พุตบิตที่ 7

เมื่อเปรียบเทียบ ระหว่างคุณสมบัติระหว่าง LED กับ LCD จะพบว่าในที่ที่มีความสว่างค่อนข้างสูง LED เกือบจะมองไม่เห็น ส่วน LCD นั้นสามารถอ่านในที่ที่มีความสว่างได้ เนื่องจากการทำงานของ LED นั้นจะปล่อยพลังงานแสงออกมา ส่วน LCD นั้นจะใช้การหักเหของแสง โดยให้แสงส่งผ่านตัวมัน ซึ่งในบางสถานะที่มีแสงสว่างน้อยก็ไม่สามารถอ่านค่าจอแสดงผล LCD ได้ วิธีแก้ก็คือการใช้จอ LCD มีแบคไลท์ (Back light) จึงเป็นการใช้ฉากรวม Electroluminescence (EL) ซึ่งมีความสามารถในการเรืองแสงได้นำไปติดตั้งไว้ด้านหลังของจอ LCD โดยที่แสง EL จะทำหน้าที่แพร่กระจายความสว่างจากด้านหลัง ทำให้ LCD มีความสว่างจากด้านหลัง ทำให้ LCD มีความสว่างและทำให้เรามองเห็นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงการต่อโมดูล LCD ร่วมกับไมโครคอนโทรลเลอร์ 8051

2.2 โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายในของมัน บางเบอร์จะมีหน่วยความจำภายในเป็นแบบรอม (ROM) บางเบอร์เป็นแบบ EPROM บางเบอร์มีแรม (RAM) ภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น

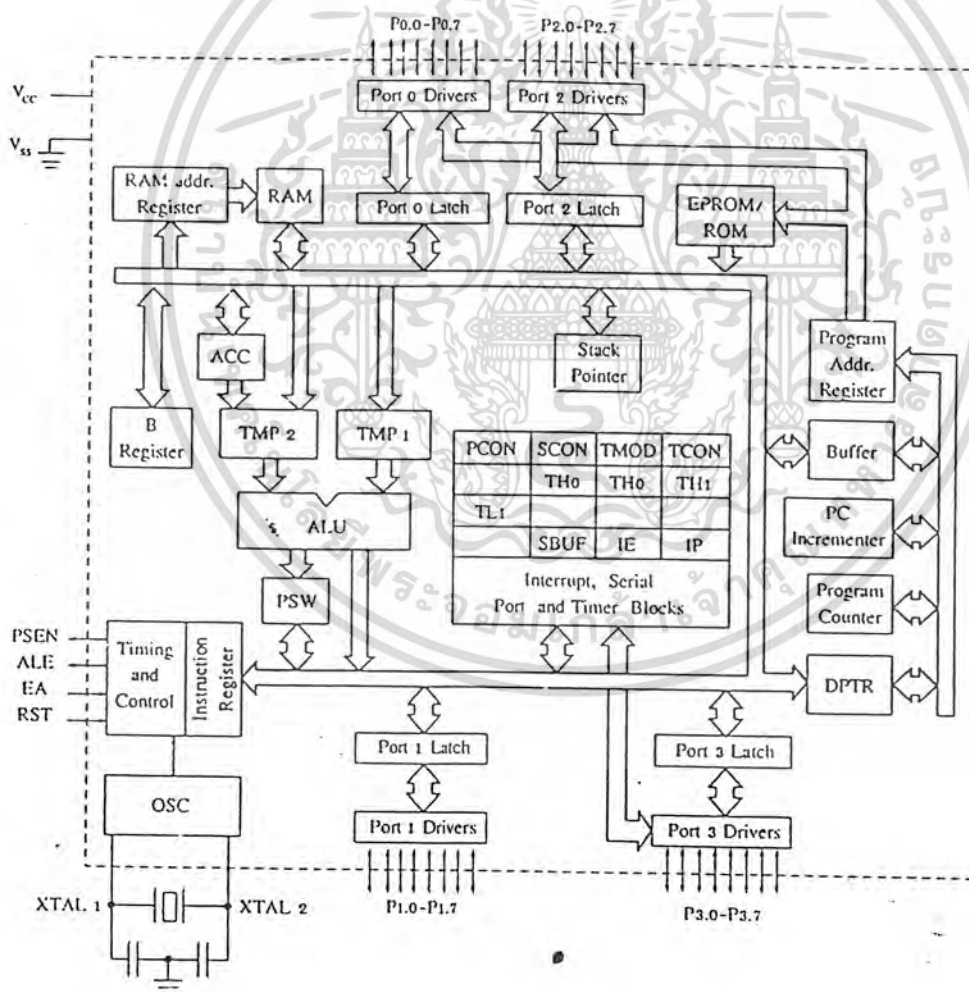
คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

1. หน่วยความจำโปรแกรมภายใน ROM ขนาด 4K byte
2. หน่วยความจำแบบ RAM ภายในขนาด 128 byte
3. มีพอร์ต I/O ขนาด 8 บิต 4 พอร์ต
4. วงจรนับ/จับเวลา ไทเมอร์ (Timer) ขนาด 16 บิต จำนวน 2 วงจร
5. สามารถอินเทอร์รัปต์ได้ 5 แหล่ง
6. วงจรออสซิลเลเตอร์และวงจรมหาภาคภายใน
7. วงจรสื่อสารแบบอนุกรมที่สามารถรับส่งข้อมูลแบบ Full Duplex ความเร็วสูง
8. ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 K byte
9. ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 K byte
10. หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)
11. วงจรควบคุมการอินเทอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้ 2 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

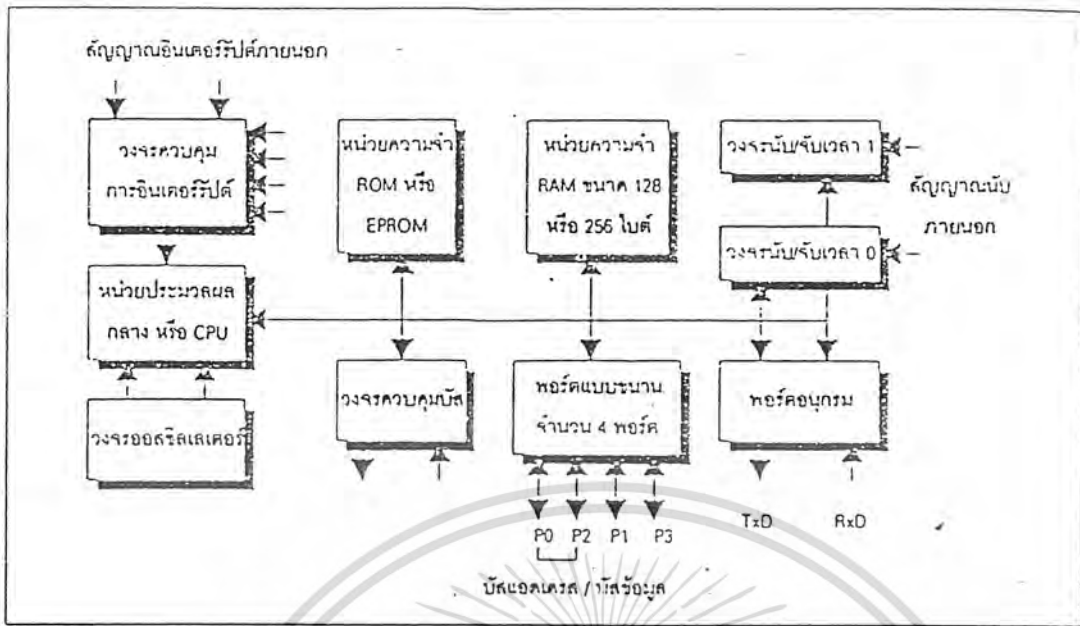
ตารางที่ 2.2 แสดงไมโครคอนโทรลเลอร์ MCS-51 เบอร์ต่างๆ

เบอร์	หน่วยความจำโปรแกรมบนชิพ	หน่วยความจำข้อมูลบนชิพ	TIMERS
8051	4K ROM	128 byte	2
8031	-	128 byte	2
8751	4K EPROM	128 byte	2
8052	8K ROM	256 byte	2
8032	-	256 byte	2
8752	8K EPROM	256 byte	2



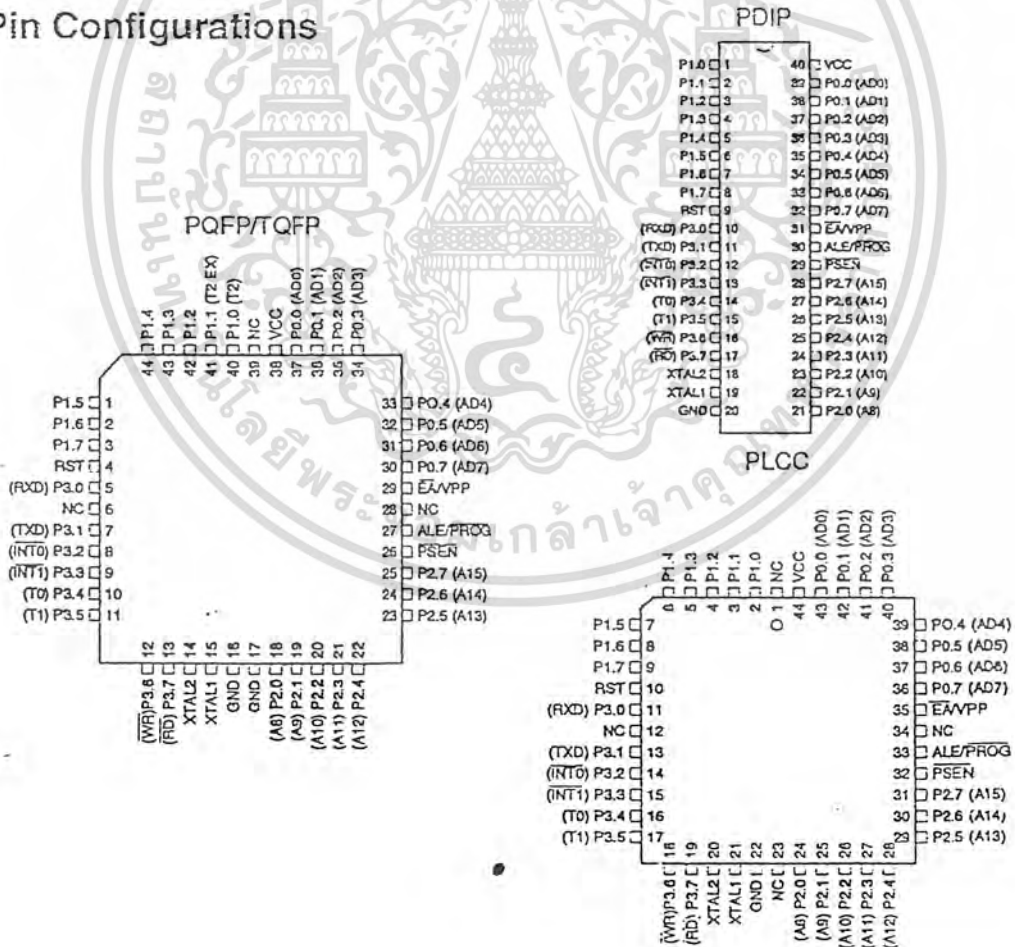
รูปที่ 2.2 แสดงโครงสร้างภายในของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 บล็อกไดอะแกรมแสดงหน่วยทำงานพื้นฐานของ MCS-51

Pin Configurations



รูปที่ 2.4 ตำแหน่งและการกำหนดหน้าที่ขาสัญญาณ ไอซี AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดขาต่างๆของ MCS-51

1. พอร์ต 0 (PORT 0)

พอร์ต 0 ได้แก่ขาที่ 32-39 ของ MCS-51 สามารถใช้เป็นอินพุต,เอาต์พุตได้นอกจากนี้ ในการติดต่อกับหน่วยความจำภายนอกยังใช้เป็นขา Address Bus และ Data Bus อีกด้วย

2. พอร์ต 1 (PORT 1)

พอร์ต 1 ได้แก่ขาที่ 1-8 เป็นพอร์ต 8 บิต สามารถอ้างทีละบิตได้คือ P1.0,p1.1,...etc

3. พอร์ต 2 (PORT 2)

พอร์ต 2 ได้แก่ขาที่ 21-28 จะใช้งานสองหน้าที่ คือใช้เป็นพอร์ต 8 บิตกับใช้เป็นขา แอคเครส 8 บิตในการอ้างหน่วยความจำภายนอก

4. พอร์ต 3 (PORT 3)

พอร์ต 3 ได้แก่ขาที่ 10-17 จะใช้งานสองหน้าที่ คือเป็นพอร์ต อินพุตและเอาต์พุตและ ใช้เป็นขาควบคุมต่างๆ ดังตารางที่ 2.3

ตารางที่ 2.3 แสดงบิตและหน้าที่ต่างๆของพอร์ต 3

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ตอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ตอนุกรม
P3.2	INT ₀	อินเทอร์รัปต์ภายนอกหมายเลข 0
P3.3	INT ₁	อินเทอร์รัปต์ภายนอกหมายเลข 1
P3.4	T ₀	ตัวจับเวลา/ ตัวนับ ตัวที่ 0
P3.5	T ₁	ตัวจับเวลา/ ตัวนับ ตัวที่ 1
P3.6	WR	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	RD	สัญญาณอ่านข้อมูลหน่วยความจำภายนอก

5. PSEN (Program Store Enable)

ขา PSEN เป็นขาที่ส่งสัญญาณออกคือขา 29 ขานี้จะแอกทีฟเมื่อ MCS-51 ต้องการอ่าน code โปรแกรมภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็น EPROM ขา PSEN จะต่อกับขา Output Enable ของ EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ALE (Address Latch Enable)

เนื่องจากพอร์ต 0 สามารถใช้เป็นขาอ้างตำแหน่ง และขาข้อมูล MCS-51 จะมีขา ALE ได้แก่ขา 30 ขานี้จะใช้ Multiplex สัญญาณกับ Address Bus ของพอร์ต 0 ในการใช้งานระบบ MCS-51 นั้นจะต้องมีอุปกรณ์มาต่อกับพอร์ต 0 ที่ทำหน้าที่ Latch สัญญาณ Address Bus เมื่อ MCS-51 ต้องการติดต่อกับหน่วยความจำภายนอก MCS-51 จะส่งสัญญาณ Address Bus ออกมาก่อนทางพอร์ต 0 จากนั้นจะส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่า Address Bus ของพอร์ต 0 ไว้เพื่อพอร์ต 0 เป็น Data Bus ต่อไป

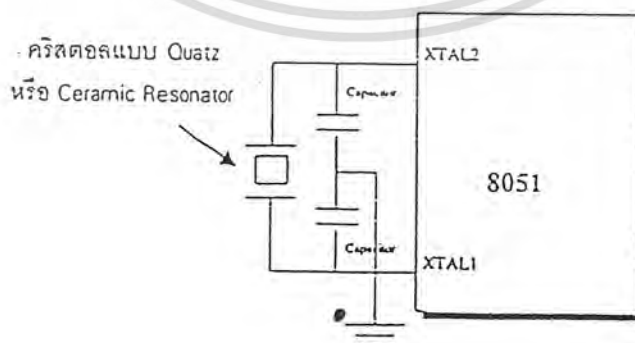
7. EA (External Access)

ขา EA ได้แก่ขาที่ 31 ถ้าขานี้เป็นลอจิก 1 จะใช้กับเบอร์ 8051/8052 เพื่อบอกว่าให้อ่านโปรแกรมจากหน่วยความจำโปรแกรมภายใน แต่ถ้าเป็นลอจิก 0 จะบอกให้ MCS-51 ทำโปรแกรมโดยอ่านจากหน่วยความจำโปรแกรมภายนอก ถ้าหากเป็นเบอร์ 8031 หรือ 8032 ขา EA จะเป็น 0 เสมอ เพราะไม่มีหน่วยความจำภายใน แต่ถ้าใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำโปรแกรมภายในภายในและให้ขา EA เป็น 0 ซึ่งจะ Disable ROM ภายในและจะอ่านโปรแกรมจาก EPROM ภายนอกแทน

8. RST (RESET)

ขา RST ได้แก่ขา 9 จะใช้ในการรีเซ็ต MCS-51 โดยจะใช้ขานี้เป็นลอจิก "1" อย่างน้อย 2 Machine Cycles จึงจะรีเซ็ตระบบได้

MCS-51 มีวงจรออสซิลเลเตอร์อยู่ภายใน สำหรับการสร้างพัลส์ของสัญญาณนาฬิกา ซึ่งจะนำไปเป็นฐานเวลา หรือการกำหนดจังหวะการทำงานของหน่วยการทำงานทั้งหมดให้สอดคล้องกัน (Synchronization) โดยการ ใช้คริสตอลเชื่อมต่อกับขาสัญญาณ XTAL1 และ XTAL2 พร้อมกับตัวเก็บประจุ หรืออาจจะเป็นสัญญาณนาฬิกาจากภายนอกก็ได้



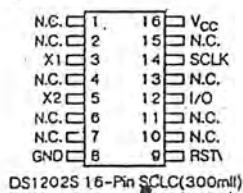
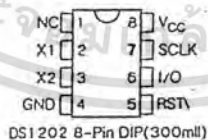
รูปที่ 2.5 แสดงการใช้คริสตอลภายนอกต่อเข้ากับวงจรออสซิลเลเตอร์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สัญญาณนาฬิกาแบบอนุกรม

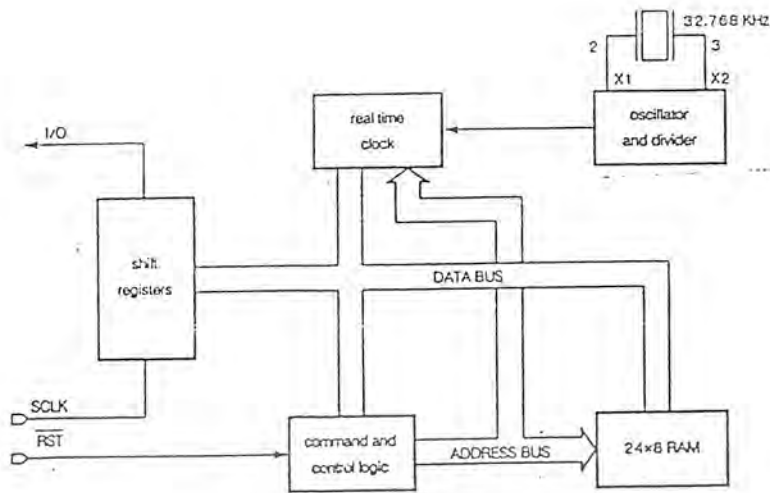
DS1302 “Serial Timekeeper Chip” ชิพRTC ของบริษัท DALLAS SEMICONDUCTOR ทำหน้าที่เป็นนาฬิกาที่สามารถส่งข้อมูลเวลาในขณะใดๆ ที่ให้เวลาเป็นจริง(Real Time Clock) กับระบบไมโครคอนโทรลเลอร์ คุณสมบัติที่สำคัญของ DS 1302

1. ทำหน้าที่นับวินาที นาที ชั่วโมง วันที่ เดือน และ ปี รวมทั้งคำนวณปีอธิกสุรทินโดยอัตโนมัติ
2. มีหน่วยความจำขนาด 24 ไบต์สำหรับเก็บข้อมูลต่างๆไป ส่วนใหญ่ไว้เก็บข้อมูลที่ต้องการสำรองในกรณีที่ไม่มีพลังงานจ่ายให้แก่ระบบ เช่น รหัสที่เปลี่ยนค่าได้ เวลาที่ให้เครื่องจักร ทำให้ไม่จำเป็นต้องสำรองหน่วยความจำของระบบนั่นเอง
3. ใช้การติดต่อแบบอนุกรม จึงใช้สายในการติดต่อเพียงสามเส้นเท่านั้น คือ Reset, SCLK และ I/O
4. ใช้แรงดันเพียง 2.0 ถึง 5.5 โวลท์ และใช้กระแสเพียง 300 นาโนแอมแปร์ที่ระดับแรงดัน 2.0 โวลท์
5. การโอนย้ายข้อมูลสามารถกระทำได้ในแบบครั้งละ 1 ไบต์ (Signal Byte) หรือครั้งละหลายๆ ไบต์ (Multiple Byte or Bust Mode) ไม่ว่าจะเป็นการเขียนข้อมูล
6. ตัวชิปเองมีให้เลือกทั้ง 8 PIN DIP หรือ 16 PIN SOIC เพื่อใช้สำหรับแผ่นวงจรชนิด Surface Mount
7. ระดับสัญญาณ TTL compatible ($V_{cc} = 5 \text{ Volt}$)
8. ช่วงอุณหภูมิในการใช้งานกว้างมาก ระหว่าง -40 องศาเซลเซียส – 88 องศาเซลเซียส



รูปที่ 2.6 แสดงลักษณะรูปร่างทั้งสองแบบของ ไอซีเบอร์ DS1302

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 โครงสร้างภายในของ RTC DS1302

โครงสร้างภายในของ RTC DS1302 ประกอบด้วย

1. Shift Register
2. Control Logic
3. Oscillator
4. Real Time Clock
5. RAM

หน้าที่และการใช้งานของแต่ละขาคือ

1. X1,X2 ต่อกับคริสตอลความถี่ 32.768 KHz เพื่อเป็นฐานข้อมูลให้กับวงจรนาฬิกาภายในตัว DS 1302
2. \overline{RST} เป็นขารีเซ็ตเพื่อทำให้วงจรบางส่วนเริ่มต้นทำงานใหม่
3. I/O DATA in/out สำหรับการอ่านและเขียนข้อมูลกับระบบไมโครคอนโทรลเลอร์
4. SCLK (Serial Clock) กำหนดการเคลื่อนย้ายในแต่ละบิตในการรับและส่งข้อมูล
5. V_{CC} ไฟเลี้ยง 2.0-5.5 โวลต์ โดยกินกระแสไม่น้อยกว่า 300 นาโนแอมป์ที่ 2 โวลต์

หน่วยความจำแรมภายใน DS1302 ขนาด 24 ไบต์ ใช้เก็บข้อมูลที่ต้องการถาวร เพราะแรมส่วนนี้ได้รับการแบคอัพเช่นเดียวกับส่วนนาฬิกาการเขียนอ่านแรมก็ทำได้เช่นเดียวกับข้อมูลนาฬิกา รับส่งข้อมูลได้ทั้งแบบ Single Byte และ Burst Mode และค่าในรีจิสเตอร์ป้องกันการเขียนทับจะมีผลกับการเขียนข้อมูลในแรมส่วนนี้ด้วย

Burst Mode หมายถึงการรับส่งข้อมูลครั้งละหลายไบต์ในการติดต่อแต่ละครั้ง โดยสามารถกำหนดได้ว่าข้อมูลที่ต้องการติดต่อเป็นรีจิสเตอร์สำหรับเก็บเวลาหรือหน่วยความจำ ที่ใช้เก็บข้อมูล Burst Mode นี้มีไว้เพื่อความสะดวกในการรับส่งข้อมูลครั้งละจำนวนมากๆทำให้ไม่ต้องส่ง Command Byte หลายครั้งนั่นเอง

Write Protect Command Byte เวลาที่เดินอยู่ภายในรีจิสเตอร์ที่ทำหน้าที่เก็บเวลาและข้อมูลที่หน่วยความจำทั้ง 24 ตำแหน่ง สามารถป้องกันไม่ให้เขียนข้อมูลใดๆซ้อนลงไปได้ เพื่อป้องกันเหตุการณ์ที่เกิดขึ้น โดยไม่ได้ตั้งใจ

Reset and Clock Control การรับส่งข้อมูลทั้งหมดจะต้องเริ่มโดยใช้ขา RST มีสถานะเป็น 1 ก่อนเสมอ โดย RST มีหน้าที่หลักอยู่ 2 ประการดังนี้

1. RST ใช้ควบคุมการเขียนหรืออ่านข้อมูลใน Shift Register
2. RST ใช้เป็นสัญญาณในการหยุดการทำงานใดๆกับชิป RTC DS1302

Data Input ในตอนเริ่มต้นติดต่อระหว่างชิป RTC DS1302 กับไมโครคอนโทรลเลอร์ไบต์แรกจะเป็น Command Byte เสมอ หากใน Command Byte ระบุว่าเป็นการเขียนข้อมูลไปในชิป ข้อมูลจะถูกรับมาในช่วงเวลาขอบขาขึ้น (Rising Edge) ของ SCLK เท่านั้น

Data Output หลังจากได้รับ command byte แล้ว หากมีการระบุว่าเป็นการอ่านข้อมูลจากชิป RTC DS1302 ข้อมูลจะถูกส่งออกจากชิป สู่ออกนอกในขณะช่วงขอบขาลง (falling edge) ของ SCLK หลังจากมีการรับ command byte เรียบร้อยแล้ว

Clock/Calendar คือ รีจิสเตอร์ 8 บิต โดยข้อมูลเหล่านี้จะอยู่ในรูปรหัส BCD เท่านั้น Clock Halt Flag บิต 7 ของรีจิสเตอร์ที่เก็บค่าวินาทีจะเป็นตัวบอกให้ชิป RTC DS1302 หยุดการทำงานของวงจรในส่วนของออสซิลเลเตอร์เมื่อบิตนี้มีค่าเป็น 1 ซึ่งเป็นผลให้ภายในชิปหยุดการทำงานไปด้วย และจะบังคับให้ชิปอยู่ในสถานะ Low Power Standby Mode โดยใช้กระแสไม่เกิน 100 นาโนแอมป์ และเมื่อบิตนี้เป็น 0 อีกครั้ง วงจรออสซิลเลเตอร์จะเริ่มการทำงานต่อทันที

AM-PM/12-24 mode บิต 7 ของรีจิสเตอร์ที่เก็บค่าชั่วโมงถูกกำหนดให้เป็น 12/24 hour mode select bit โดยเป็นตัวเลือกว่าจะให้รีจิสเตอร์เก็บค่าชั่วโมงแบบ 12 ชั่วโมง หรือ 24 ชั่วโมง

Write Protect บิตที่ 7 ของ Write Protect Register จะเป็น Write Protect Bit โดย 7 บิตแรกถูกบังคับให้เป็น 0 หหมด ทำให้อ่านได้เป็น 0 เสมอ

Clock/Calendar Burst Mode command byte ที่มีค่า BEH จะเป็นการระบุว่ามีเขียนข้อมูลในรีจิสเตอร์ที่เก็บเวลาในแบบ Burst Mode

RAM Burst Mode command byte ที่มีค่า FEH จะเป็นการระบุว่ามีภาระระบุว่ามีภาระเขียนข้อมูลในหน่วยความจำแบบ Burst Mode หาก command byte มีค่า FFH จะระบุว่ามีภาระอ่านข้อมูลในหน่วยความจำแบบ burst mode ทำนองเดียวกันใน clock/calendar burst mode จะมีข้อแตกต่างกันก็เพียงแค่จำนวนรับหรือส่งเท่านั้น เพราะหน่วยความจำใน ชิพเบอร์ DS 1202 มีขนาด 24 ไบต์จะรับส่งกัน โดยเริ่มต้นที่ บิต 0 ของหน่วยความจำตำแหน่ง 0 เสมอ

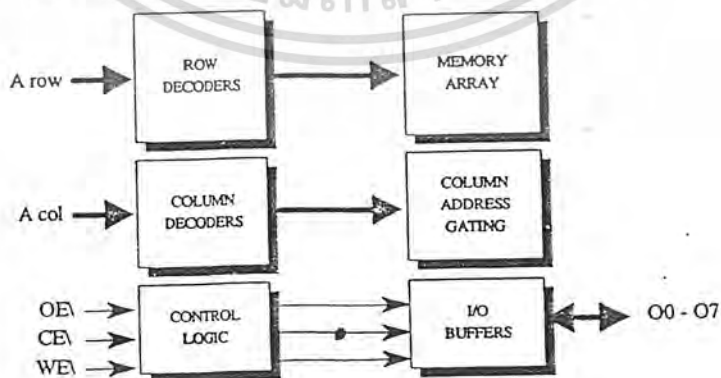
สำหรับการใช้งานจริง ขา RESET, SCLK จะเป็นอินพุตอย่างเดียว ส่วนขา I/O เป็นทั้งอินพุตและเอาต์พุต ขา X1, X2 ต่อกับคริสตอลความถี่ 32.768 KHz ได้โดยตรง วงจรนาฬิกาเที่ยงตรง เนื่องจากไม่ต้องมีส่วนปรับความถี่ และถ้าต่อตัวถังของคริสตัลลงกราวด์ ก็ทำให้มีเสถียรภาพดีขึ้น DS1302 สามารถต่อกับพอร์ตหนึ่งของ CPU ตระกูล MCS-51 ได้โดยตรงถ้าเป็นระบบ Z-80 จะต้องผ่านพอร์ต เช่น พอร์ต C ของ 8255 และเนื่องจาก DS1302 ไม่มีขา อินเทอร์รัปต์ (Interrupt) ดังนั้น CPU จะต้องทำการอ่านข้อมูลจาก DS1302 เอง

2.4 หน่วยความจำข้อมูลภายนอก RAM

เนื่องจากหน่วยความจำข้อมูลสำหรับการเก็บข้อมูลชั่วคราว หรือตัวแปร โปรแกรมมามากกว่าขนาดของหน่วยความจำข้อมูลภายใน ทำให้ต้องมีการเก็บสำรองข้อมูล โดยการใช้ไอซีหน่วยความจำ RAM พร้อมแบริเตอร์สำรองเป็นตัวเก็บข้อมูล

ลักษณะสมบัติของหน่วยความจำ RAM แบบสแตติก คล้ายกับหน่วยความจำ EPROM มาก รวมทั้งประเภทของสัญญาณติดต่อ คือ บัสแอดเดรส บัสข้อมูล และกลุ่มสัญญาณควบคุม มีการเพิ่มสัญญาณ WR เพื่อระบุการนำข้อมูลจากบัสข้อมูล ไปเก็บยังหน่วยความจำตำแหน่งที่ได้ระบุแอดเดรสมา

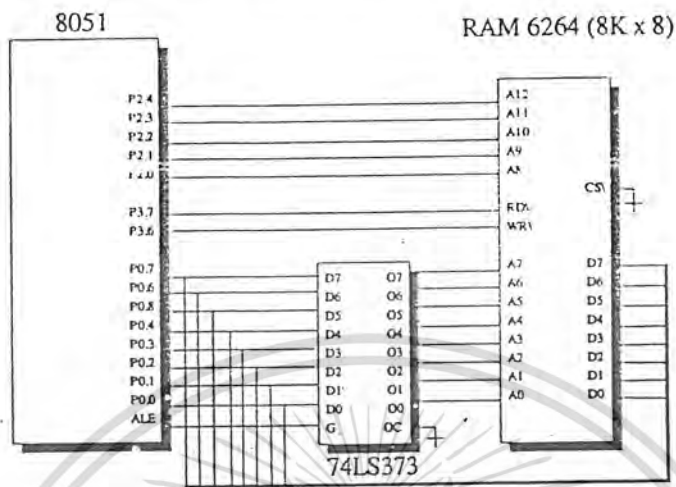
Block Diagram



รูปที่ 2.8 Block Diagram แสดงขาสัญญาณของหน่วยความจำ RAM แบบสแตติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อหน่วยความจำ RAM เข้ากับระบบของไมโครคอนโทรลเลอร์ จะใช้วิธีการ
 เหมือนกับการเชื่อมต่อหน่วยความจำ EPROM ดังรูปที่ 2.9



รูปที่ 2.9 วงจรแสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกกับไมโครคอนโทรลเลอร์

2.5 การเชื่อมต่อ 8255 กับไมโครคอนโทรลเลอร์

เป็นชิพขนาด 40 ขา ได้รับการออกแบบมาเพื่อทำหน้าที่เป็นพอร์ต สำหรับการรับส่งข้อมูล
 แบบขนานระหว่างอุปกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ มี 3 พอร์ต คือ A,B,C เป็นพอร์ต 8
 บิต โดยที่พอร์ต C แบ่งเป็น 4 บิตล่าง และ 4 บิตบน

หน้าที่การทำงานของขาสัญญาณไอซี 8255

1. D0-D7 กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียน หรืออ่าน เชื่อมโยงกับ
 CPU

2. CS₁ สัญญาณเลือกอุปกรณ์ เมื่อขาสัญญาณนี้เป็นระดับลอจิกต่ำ CPU ก็สามารถเขียน
 หรืออ่านข้อมูลจาก 8255 ได้

3. RD₁ สัญญาณบอกสถานะต้องการอ่านข้อมูลจากรีจิสเตอร์ของ 8255

4. WR₁ สัญญาณบอกสถานะต้องการเขียนข้อมูลให้กับรีจิสเตอร์ของ 8255

5. A0-A1 สัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ

6. RESET สัญญาณการรีเซตของวงจรทำงานภายใน 8255 เพื่อเริ่มต้นใหม่

7. PA0-PA7 กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต A ของ 8255

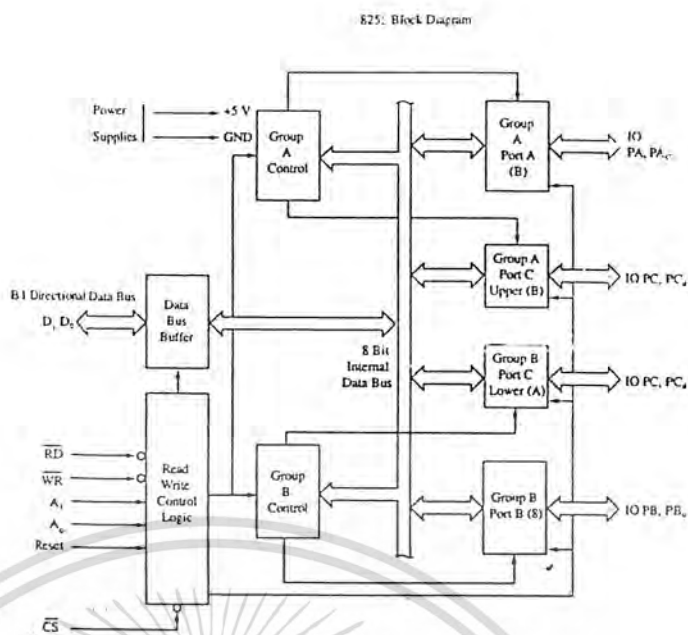
8. PB0-PB7 กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต B ของ 8255

9. PC0-PC7 กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต C ของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pin Names	
D ₇ -D ₀	Data Bus (Bi-Directional)
RESET	Reset Input
CS	Chip Select
RD	Read Input
WR	Write Input
A ₀ , A ₁	Port Address
PA ₇ -PA ₀	Port A (bit)
PB ₇ -PB ₀	Port B (bit)
PC ₇ -PC ₀	Port C (bit)
V _{cc}	+5 Volts
GND	0 Volts



รูปที่ 2.10 แสดงโครงสร้างของไอซี 8255

การทำงานของ 8255 มี 3 โหมด คือ

1. โหมด 0 จะทำให้พอร์ตต่าง ๆ มีหน้าที่เป็นพอร์ตอินพุต หรือเอาต์พุตเพียงลักษณะเดียว มีการทำงานแบบ BASIC I/O ไม่มี Handshake
2. โหมด 1 จะทำให้พอร์ต A และพอร์ต B สามารถใช้งานเป็นการอินพุต หรือเอาต์พุตข้อมูลในลักษณะเดียวกับโหมด 0 แต่พอร์ต C จะถูกนำไปใช้เป็นที่พอร์ตสำหรับการบอกสถานะการติดต่อ (Handshake Signals)
3. โหมด 2 กำหนดไว้ให้ใช้เฉพาะกับการทำงานของพอร์ต A เท่านั้น มีลักษณะเป็นพอร์ตข้อมูลแบบ 2 ทิศทาง

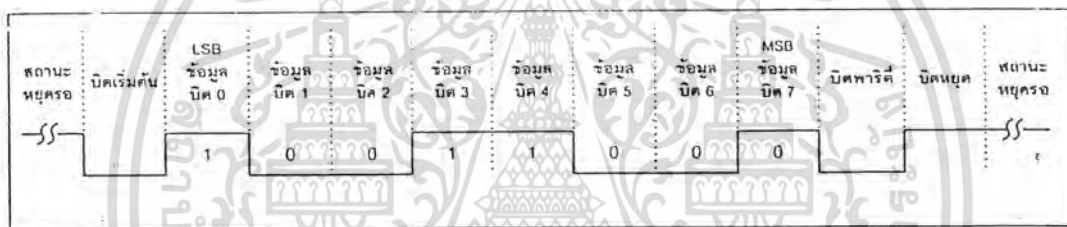
2.6 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรม แบ่งออกได้เป็น 2 แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาพร้อมอยู่กับการรับและส่ง สัญญาณด้วย ตัวอย่างเช่น คีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการสื่อสารอนุกรมแบบซิงโครนัสจะต้องใช้สายในการต่อเชื่อมอย่างน้อยที่สุด 3 สาย คือ สัญญาณนาฬิกา , ข้อมูล และกราวด์

การสื่อสารอนุกรมแบบอะซิงโครนัส คือ การรับและส่งข้อมูลไปในสาย โดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมเหมือนการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5 , 6 , 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1 , 1.5 หรือ 2 บิต

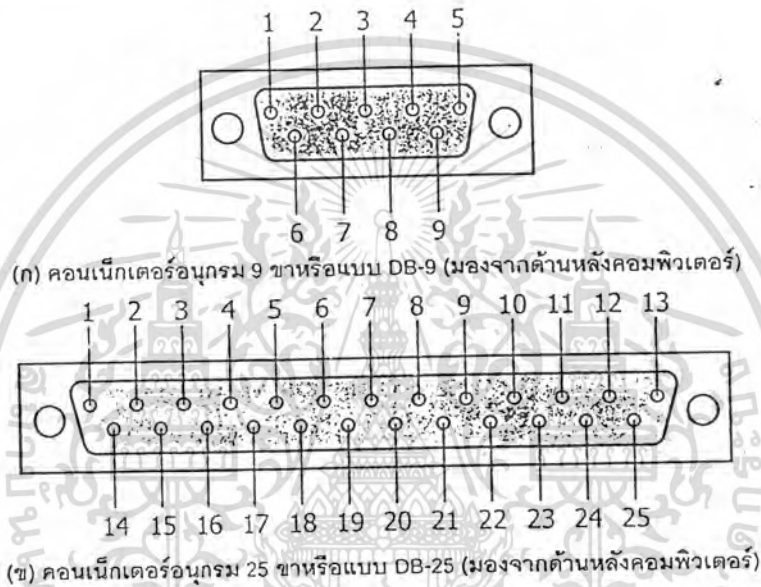


รูปที่ 2.11 รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส

Universal Asynchronous Receiver/Transmitter หรือ UART เป็นอุปกรณ์ที่ออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัส หน้าที่หลักของ UART คือ ทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์นอกจากนี้ยังแจ้งข้อมูลอื่น ๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต) , รูปแบบการส่งข้อมูล , ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี , เฟรมข้อมูล , โอเวอร์รัน)

อัตราความเร็วในการรับและส่งข้อมูลแบบอะซิงโครนัส คือ ค่าบอดเรต บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110 , 150 , 300 , 600 , 1200 , 2400 , 4800 , 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์

มาตรฐานพอร์ตอนุกรมแบบ RS-232 ได้กำหนดรูปแบบของอุปกรณ์การเชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232 มาตรฐาน RS-232 จะใช้คอนเนกเตอร์แบบ DB-9 ตัวผู้ หรือ DB-25 ตัวผู้ ซึ่งคอนเนกเตอร์แบบ DB-25 จะมีขาการต่อใช้งานเพียง 9 เส้น

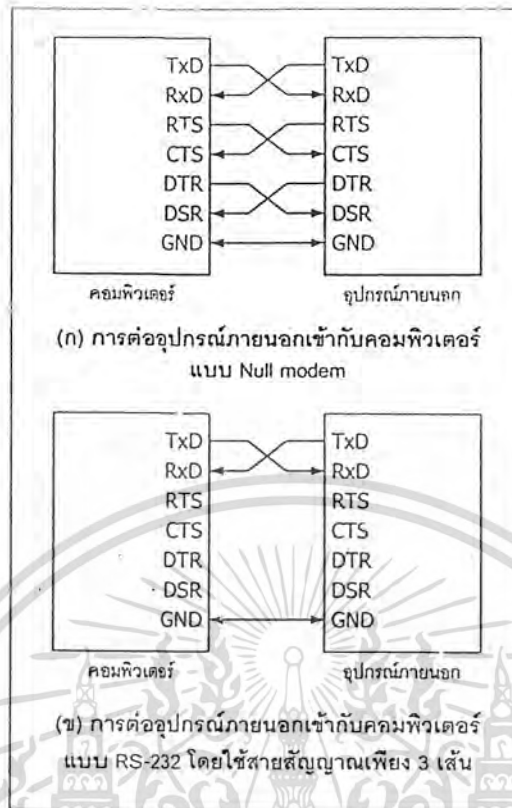


คอนเนกเตอร์ DB-9	คอนเนกเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

รูปที่ 2.12 การจัดขาของคอนเนกเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232

แบบ DB-9 และ DB-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ

หน้าที่การทำงานของแต่ละขาของพอร์ตอนุกรม RS-232

1. Data Carrier Detect : DCD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก
2. Receive Data : RD RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์
3. Transmitted Data : TD TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป
4. Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์ เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อขา DCD ด้วยในกรณีที่ใช้โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห้
5. Signal Ground : GND ขากราวนด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้ จะเป็นขาสำหรับรับข้อมูลจากภายนอก ซึ่งถูกส่งมาจากขา DTR
7. Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
8. Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
9. Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากโทรศัพท์ ปกติในการสื่อสารโดยทั่วไป สายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็ม และโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

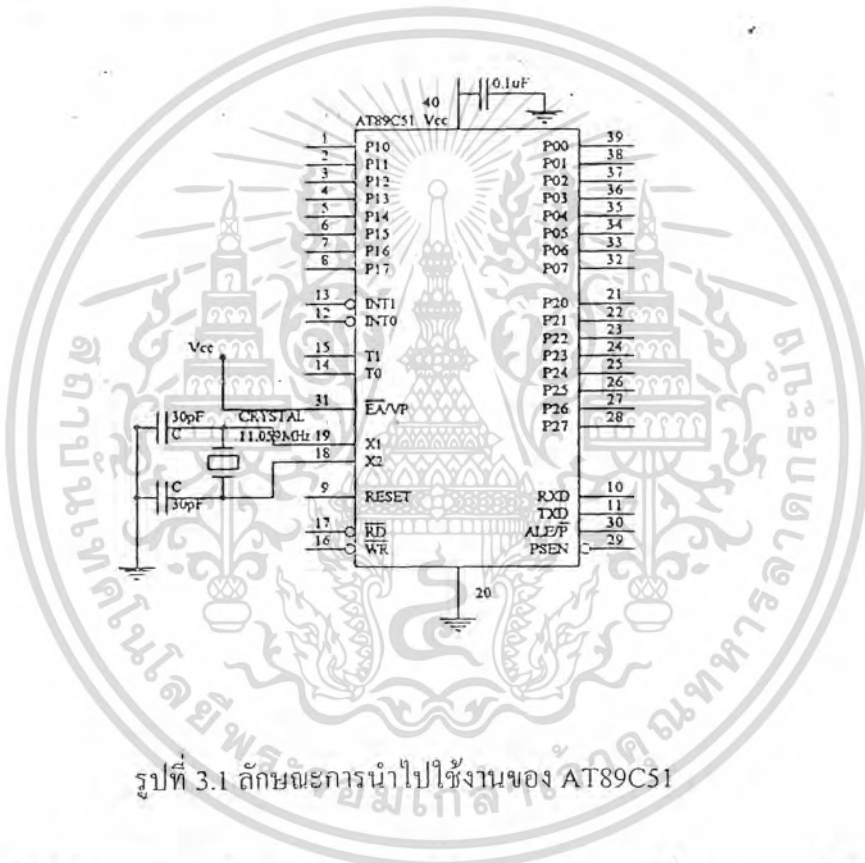
บทที่ 3

การออกแบบ

ในการออกแบบบอร์ดแสดงตารางเวลาการทำงาน มีส่วนประกอบดังนี้

3.1 หน่วยประมวลผล

ในโครงการนี้หน่วยประมวลผลจะใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51 ในตระกูล MCS-51 ซึ่งมีลักษณะเป็นแบบ CMOS คือใช้เทคโนโลยีในการผลิตไอซีซึ่งรวมสูงมาก (LSI) มีคุณลักษณะที่สูงมาก และสิ้นเปลืองกำลังไฟฟ้าน้อยกว่ามาก

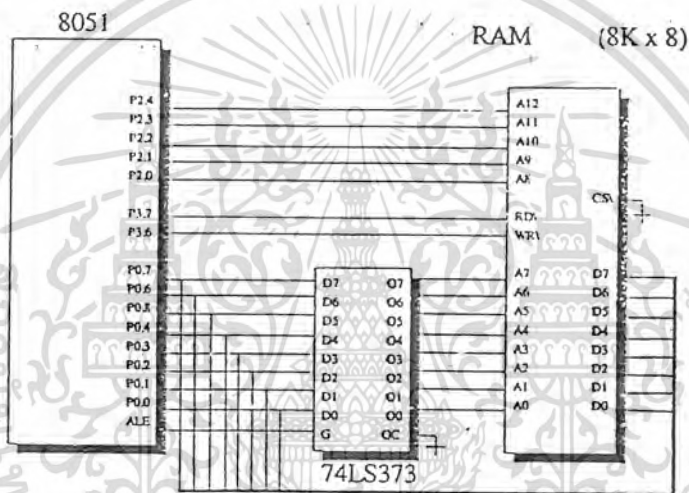


รูปที่ 3.1 ลักษณะการนำไปใช้งานของ AT89C51

ในที่นี้จะใช้ค่าคริสตอลเป็น 11.059 เมกะเฮิร์ต เพราะที่ความถี่นี้ สามารถนำไปใช้ในการเป็นฐานเวลาสำหรับการสร้างควมถี่ในการรับ/ส่งข้อมูลอนุกรมซึ่งเป็นหน่วยการทำงานหนึ่งภายใน AT89C51 เอง

3.2 หน่วยความจำ

ที่เก็บข้อมูลภายในสำหรับไมโครคอนโทรลเลอร์ ในที่นี้ใช้ RAM เบอร์ 682128 มีลักษณะการส่งข้อมูล 2 ทิศทาง คือ การเขียนและการอ่านข้อมูล การกำหนดช่วงเวลาการอ่านและการเขียนจะถูกกำหนดจาก AND Gate . OR Gate . Inverter ต่างๆ รูปที่ 3.4 แสดงถึงตัวอย่างการเชื่อมต่อหน่วยความจำภายนอกกับไมโครคอนโทรลเลอร์ โดยมีไอซี 74LS373 ทำหน้าที่ในการค้าง หรือ แลตช์ (Latch) ค่าแอดเดรสให้กับอินพุตของหน่วยความจำ RAM ขาสัญญาณ OE (หรือ RD) จะต่อเข้ากับขาสัญญาณ RD (P3.7) และขาสัญญาณ WR (P3.6) ของไมโครคอนโทรลเลอร์ เนื่องจากภายในวงจรมีไอซีหน่วยความจำเพียงตัวเดียว จึงทำการต่อสัญญาณ CS กับสัญญาณกราวนด์โดยตรงเพื่อทำการเลือกให้ไอซีทำงานตลอดเวลา



รูปที่ 3.2 วงจรการเชื่อมต่อหน่วยความจำข้อมูลภายนอกกับไมโครคอนโทรลเลอร์

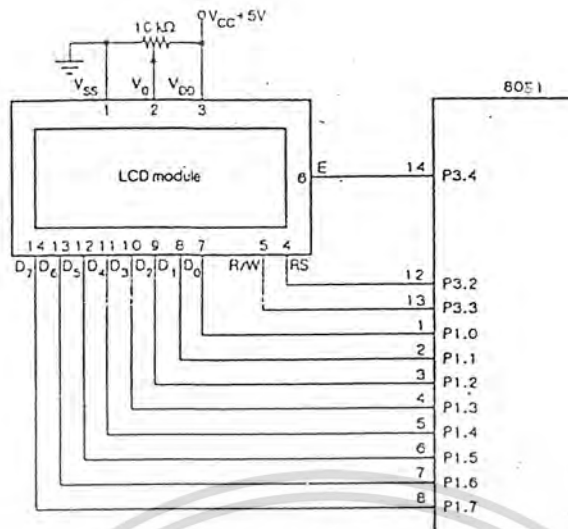
3.3 จอแสดงผล

ส่วนการแสดงผลใช้จอแสดงผล LCD สามารถแบ่ง LCD โมดูล ชนิดคอตเมตริกซ์ ได้เป็นแบบ character LCD module , graphic LCD module , segment display type LCD module

โดยแต่ละแบบจะมีส่วนประกอบใหญ่ ๆ ดังนี้

1. จอ LCD แบบคอตเมตริกซ์ (dot matrix LCD) เป็นตัวแสดงผลที่ทำให้เรามองเห็นในลักษณะการเปิดและปิดตัวเองกับแสง ก็คือ ส่วนของที่เป็นตัวกระจกบรรจุผลึก
2. ไครเวอร์ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD
3. คอนโทรลเลอร์ (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD ให้ทำงาน แสดงผลต่าง ๆ เช่น การลบจอภาพ การเกิดตัวอักษร เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

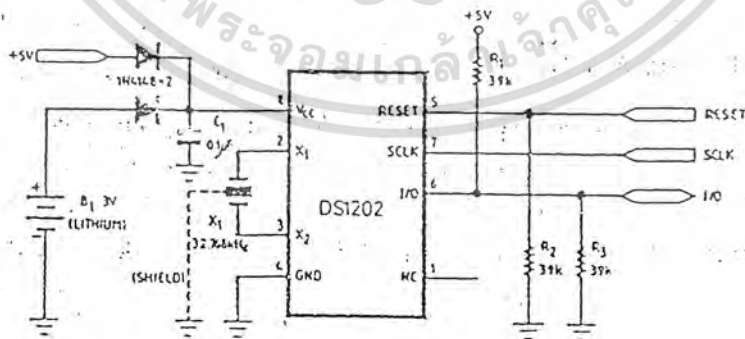


รูปที่ 3.3 วงจรการอินเทอร์เฟส LCD กับไมโครคอนโทรลเลอร์

- ขา P1.0 – P1.7 เป็นคาตาบัส (DB0 – DB7) ในการติดต่อ
- ขา P3.2 เป็นสัญญาณ RS
- ขา P3.3 เป็นสัญญาณ R/W
- ขา P3.4 เป็นสัญญาณ EN (E)

3.4 สัญญาณนาฬิกา

จะใช้ไอซีชิพ DS1302 เป็นสัญญาณนาฬิกา หน่วยความจำรวมภายใน DS1302 ขนาด 24 ไบต์ มีลักษณะการนำไปใช้งาน ดังรูปที่ 3.5



รูปที่ 3.4 ลักษณะการนำไปใช้งานของ DS1302

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

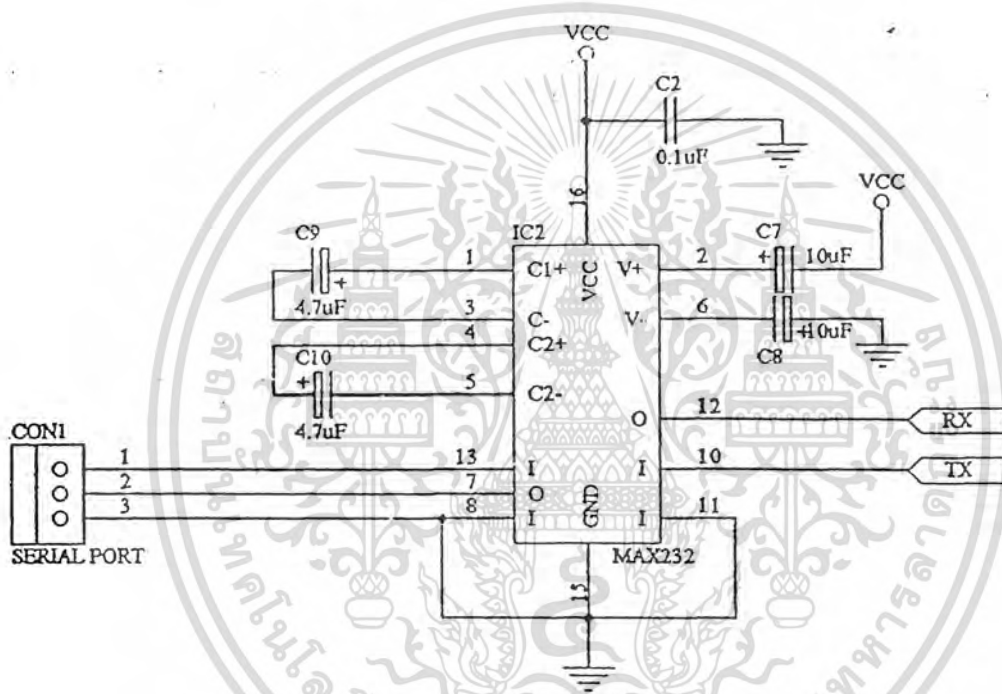
3.5 สวิตช์

สวิตช์กดเป็นแบบกดติด ปล่อยดับ 2 ปุ่ม คือ

1. ปุ่ม PREVIOUS ทำหน้าที่ปรับวัน เวลา ก่อนหน้า
2. ปุ่ม NEXT ทำหน้าที่ปรับวัน เวลา วันถัดไป

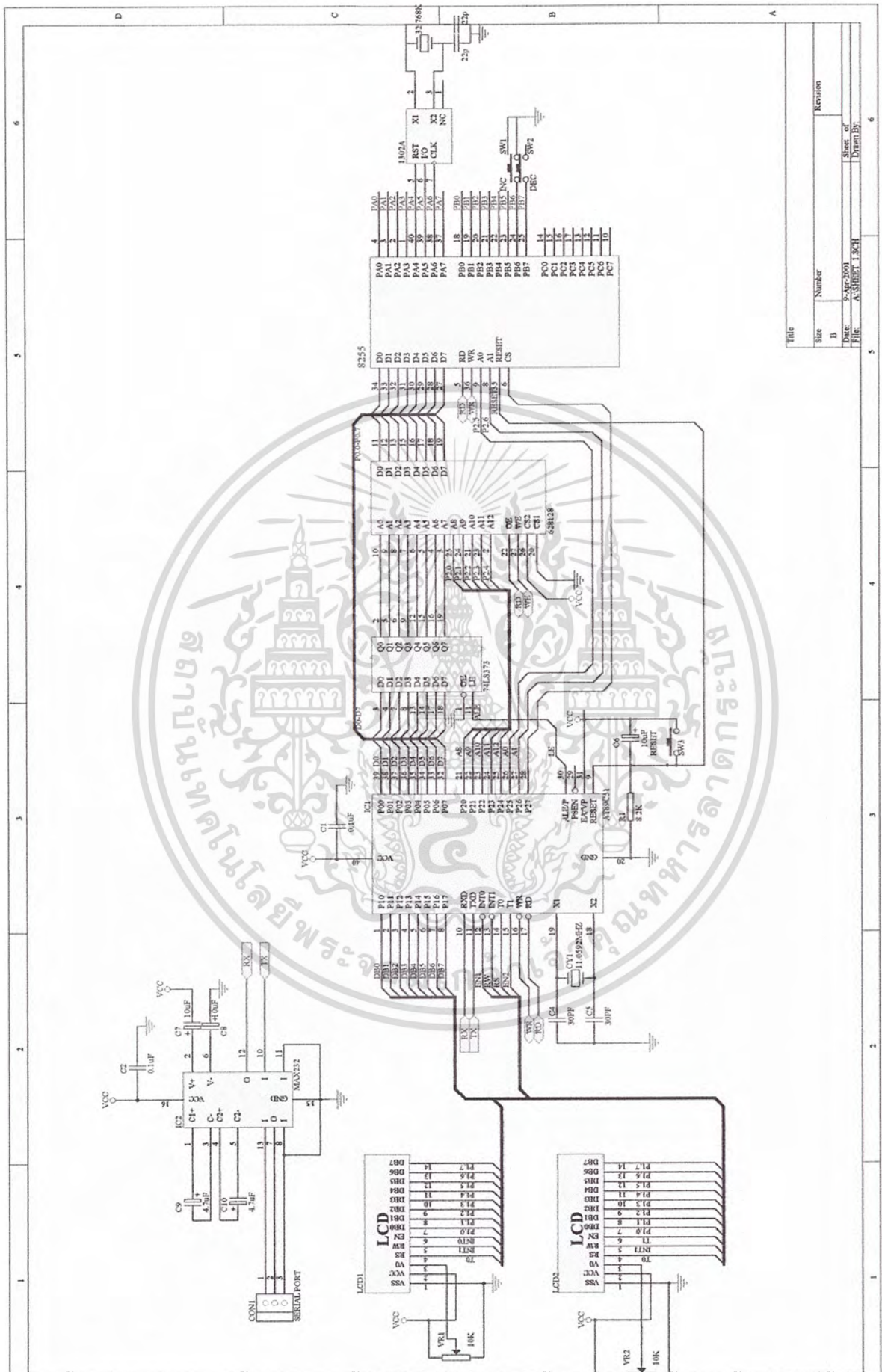
3.6 ส่วนติดต่อกับคอมพิวเตอร์

ในที่นี้จะใช้พอร์ตอนุกรม RS-232 สื่อสารกับคอมพิวเตอร์ โดยมี MAX232 เป็นไอซี ตัวรับ และส่ง มีลักษณะการนำไปใช้ ดังรูปที่ 3.5



รูปที่ 3.5 วงจรเชื่อมต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title	Number	Revision
B		
Date:	9-Apr-2001	Sheet of
File:	A_SSHR1.SCH	Drawn By:
		6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.6 วงจรรวม
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

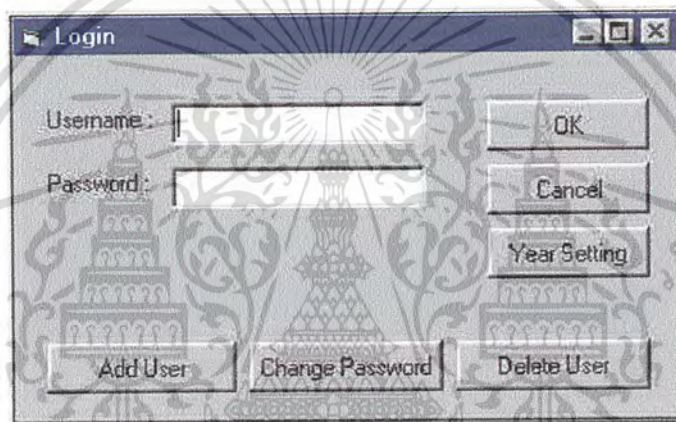
บทที่ 4

โปรแกรม Visual Basic

4.1 โปรแกรม Visual Basic

เป็นส่วนที่ใช้สำหรับติดต่อกับผู้ใช้งานโดยตรง โดยเป็นส่วนที่ใช้สำหรับดูข้อมูลเดิม , เปลี่ยนแปลงหรือเพิ่มข้อมูลใหม่ แล้วทำการส่งข้อมูลผ่านพอร์ตอนุกรม เพื่อทำการแสดงผลต่อไป ซึ่งได้ออกแบบให้มีลักษณะการใช้งาน ดังนี้

เมื่อเริ่มต้นเปิด โปรแกรม จะได้หน้าต่างสำหรับผู้ใส่ชื่อผู้ใช้งาน และรหัสผ่าน ดังรูปที่ 4.1



รูปที่ 4.1 หน้าต่าง LOGIN

ถ้า USERNAME ยังไม่ได้ลงทะเบียน จะไม่สามารถเข้าในส่วนข้อมูลได้ และจะมีหน้าต่างเตือน ดังรูปที่ 4.2



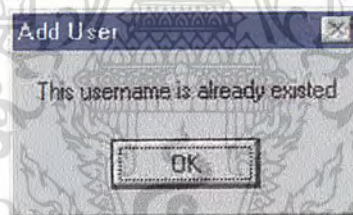
รูปที่ 4.2 หน้าต่างเตือน USERNAME ยังไม่ได้ลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ารับผู้ใช้ใหม่ จะต้องทำการลงทะเบียนก่อน โดยคลิกปุ่ม ADD USER จะเข้าสู่หน้าต่าง สำหรับลงทะเบียนผู้ใช้ใหม่ ดังรูปที่ 4.3

รูปที่ 4.3 หน้าต่าง ADD NEW USER

ถ้า USERNAME ที่ผู้ใช้ใหม่ซ้ำกับ USERNAME ของผู้ใช้เดิม จะมีหน้าต่างเตือนดังรูปที่ 4.4 ผู้ใช้ใหม่ต้องทำการลงทะเบียน USERNAME ใหม่ โดยไม่ให้ซ้ำกับผู้ใช้เดิมที่มีอยู่แล้ว



รูปที่ 4.4 หน้าต่างเตือน USERNAME ซ้ำ

ถ้าผู้ใช้ใหม่ ทำการใส่ PASSWORD ไม่ตรงกัน จะมีหน้าต่างเตือนดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่างเตือน PASSWORD ไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการลงทะเบียนเสร็จเรียบร้อยแล้ว จะมีหน้าต่างแสดงว่า USERNAME ถูกเพิ่มเรียบร้อยแล้ว และเข้าสู่หน้าต่าง LOGIN อีกครั้ง ดังรูปที่ 4.6



รูปที่ 4.6 หน้าต่างแสดงการเพิ่มชื่อเสร็จเรียบร้อยแล้ว

ถ้า USERNAME ต้องการลบชื่อซึ่งทำได้โดยการกดปุ่ม DELETE USER จะเข้าสู่หน้าต่างสำหรับลบทิ้ง ดังรูปที่ 4.7



รูปที่ 4.7 หน้าต่าง DELETE USER

เมื่อโปรแกรมทำการลบชื่อที่เรียบร้อยแล้ว จะมีหน้าต่างแสดง ดังรูปที่ 4.8 และจะกลับเข้าสู่หน้าต่าง LOGIN อีกครั้ง



รูปที่ 4.8 หน้าต่างแสดงลบชื่อที่เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าผู้ใช้งานต้องการเปลี่ยนรหัสผ่าน สามารถทำได้โดย กดปุ่ม CHANGE PASSWORD จะ
ได้หน้าต่างดังรูปที่ 4.9

รูปที่ 4.9 หน้าต่าง CHANGE PASSWORD

เมื่อโปรแกรมทำการเปลี่ยนรหัสผ่านเรียบร้อยแล้ว จะมีหน้าต่างแสดง ดังรูปที่ 4.10 และจะ
เข้าสู่หน้าต่าง LOGIN อีกครั้ง

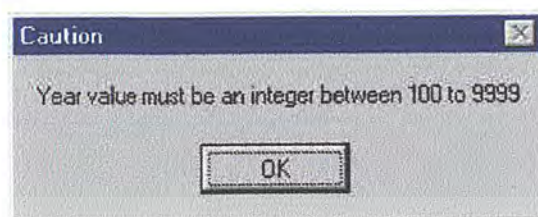
รูปที่ 4.10 หน้าต่างแสดงการเปลี่ยนรหัสผ่านเรียบร้อยแล้ว

ถ้าผู้ใช้งานต้องการกำหนดค่าปีที่ต้องการเก็บข้อมูล ที่ไม่ใช่ปีปัจจุบัน ทำได้โดยการกดปุ่ม
YEAR SETTING จะได้หน้าต่างกำหนดค่าปี ดังรูปที่ 4.11

รูปที่ 4.11 หน้าต่าง YEAR SETTING

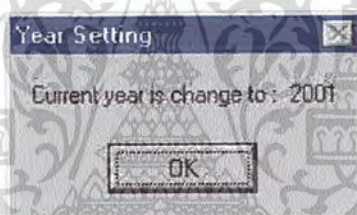
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมได้ทำการออกแบบให้เก็บค่าข้อมูลได้ตั้งแต่ปี ค.ศ. 100- ค.ศ. 9999 ถ้าผู้ใช้งานใส่ค่าที่มากกว่า หรือน้อยกว่า 100 - 9999 จะได้นหน้าต่างเตือน ดังรูปที่ 4.12



รูปที่ 4.12 หน้าต่างเตือนป้อนข้อมูลปีผิด

เมื่อทำการกำหนดค่าปีเรียบร้อยแล้ว จะได้นหน้าต่างดังรูปที่ 4.13 และจะกลับเข้าสู่หน้าต่าง LOGIN อีกครั้ง



รูปที่ 4.13 หน้าต่างแสดงกำหนดค่าปีเรียบร้อยแล้ว

แต่ถ้าผู้ใช้งานกดปุ่ม OK โดยที่ยังไม่ได้ใส่ค่าปี จะได้นหน้าต่างเตือน ดังรูปที่ 4.14



รูปที่ 4.14 หน้าต่างเตือนไม่ได้ใส่ค่าปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากใส่ชื่อผู้ใช้เพื่อยืนยันชื่อ และกดปุ่ม OK ที่หน้าต่าง LOGIN จะได้หน้าต่าง SCHEDULE สำหรับใส่ข้อมูล ถ้ามีข้อมูลเดิมอยู่ จะแสดงข้อมูลเดิม ในช่วงเวลาต่างๆ สามารถเปลี่ยนข้อมูลใหม่ได้ เมื่อหน้าต่างนี้ถูกเปิดขึ้น โปรแกรมจะถูกตั้งค่าแรกที่วัน เดือน ปีปัจจุบัน เมื่อทำการใส่ข้อมูลลงไป ข้อมูลจะถูกเก็บเอาไว้

รูปที่ 4.15 หน้าต่าง SCHEDULE

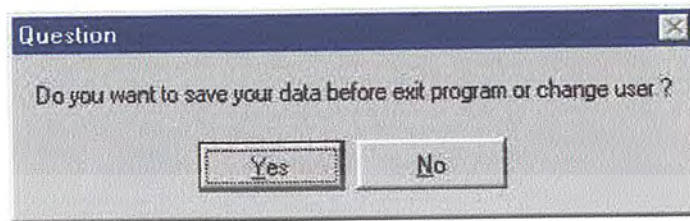
เมื่อใส่ข้อมูลเสร็จเรียบร้อยแล้ว ทำการกดปุ่มบันทึกข้อมูล จะได้หน้าต่าง ดังรูปที่ 4.16



รูปที่ 4.16 หน้าต่างแสดงบันทึกข้อมูลเรียบร้อยแล้ว

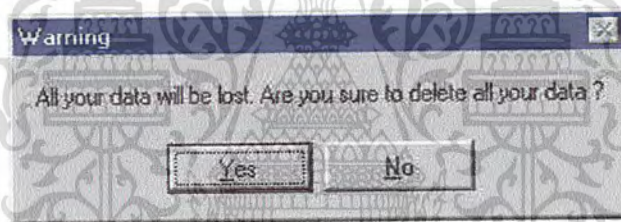
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการออกจากโปรแกรม หรือเปลี่ยนผู้ใช้งาน จะมีหน้าต่างคำถามต้องการบันทึกข้อมูลหรือไม่ ดังรูปที่ 4.17



รูปที่ 4.17 หน้าต่างคำถามบันทึกข้อมูลหรือไม่

เมื่อต้องการลบข้อมูลเดิมทิ้ง ทำได้โดยกดปุ่ม DELETE ALL DATA จะได้นหน้าต่างคำถามต้องการลบข้อมูลหรือไม่ ดังรูปที่ 4.18



รูปที่ 4.18 หน้าต่างคำถามต้องการลบข้อมูลหรือไม่

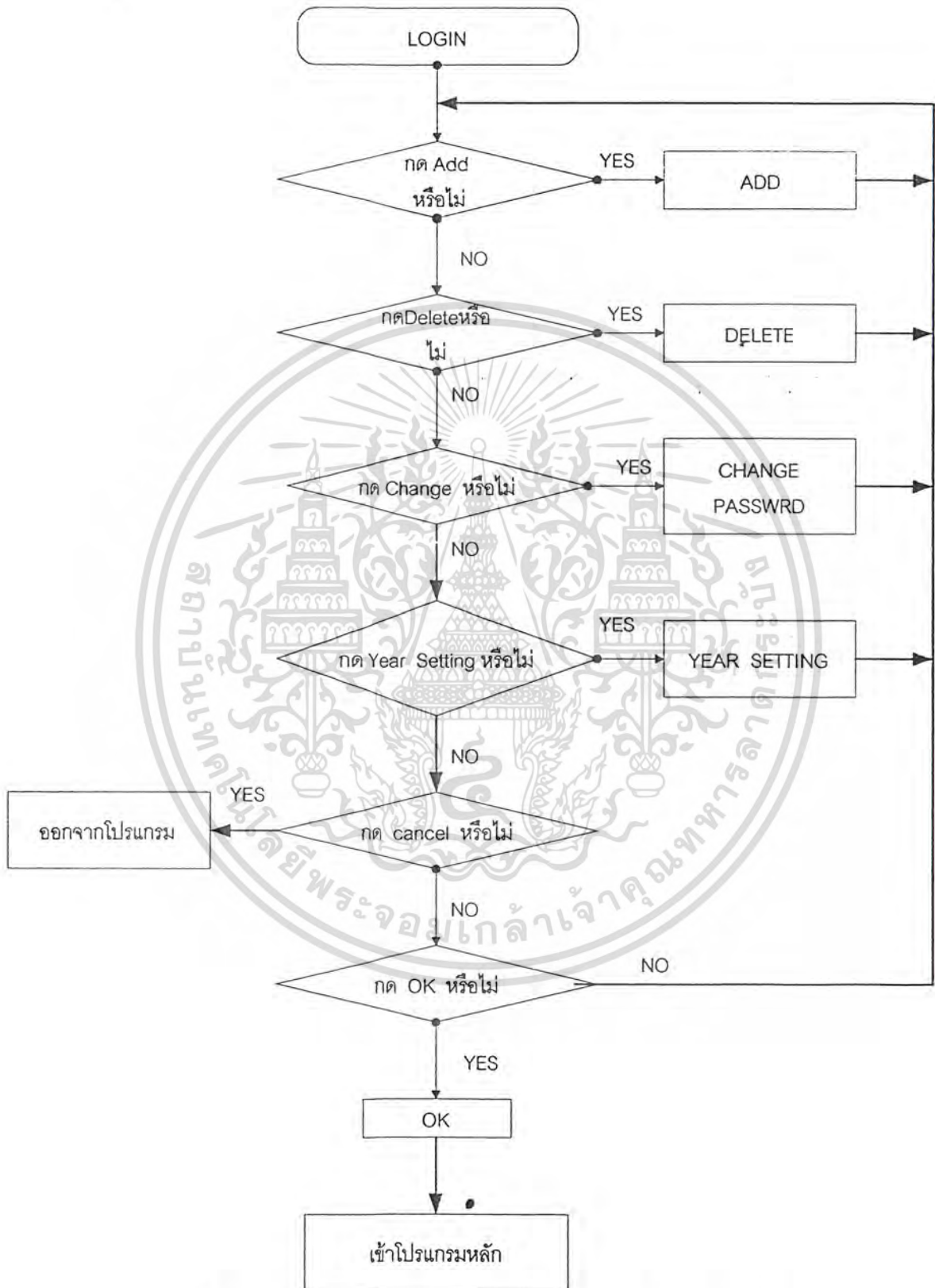
เมื่อทำการลบข้อมูลเดิมเรียบร้อยแล้วจะได้นหน้าต่าง ดังรูปที่ 4.19



รูปที่ 4.19 หน้าต่างแสดงลบข้อมูลเดิมเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

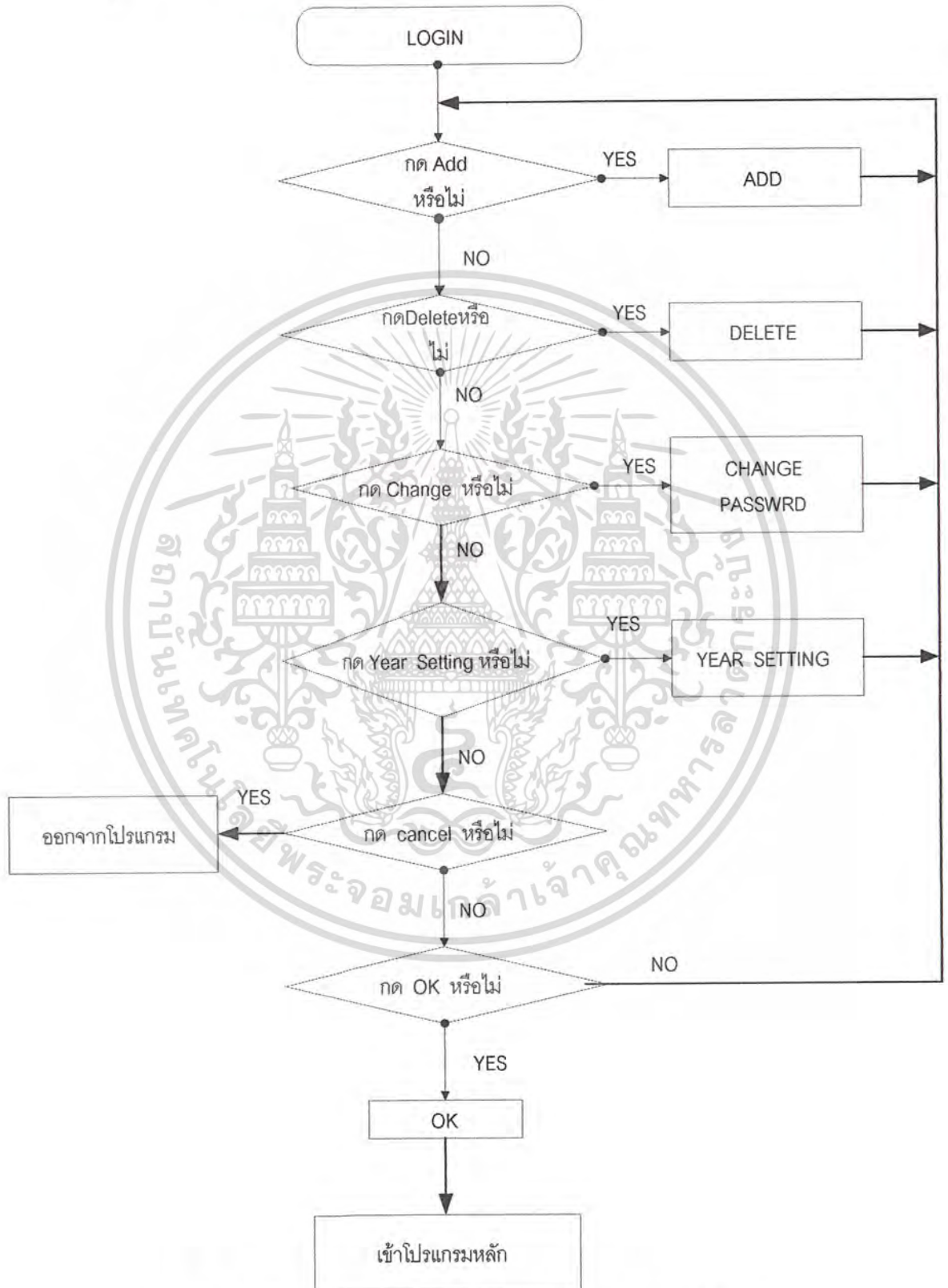
สรุปการทำงานของโปรแกรม Visual Basic ได้ดังนี้



รูปที่ 4.20 FLOWCHART การทำงานของโปรแกรม Visual Basic

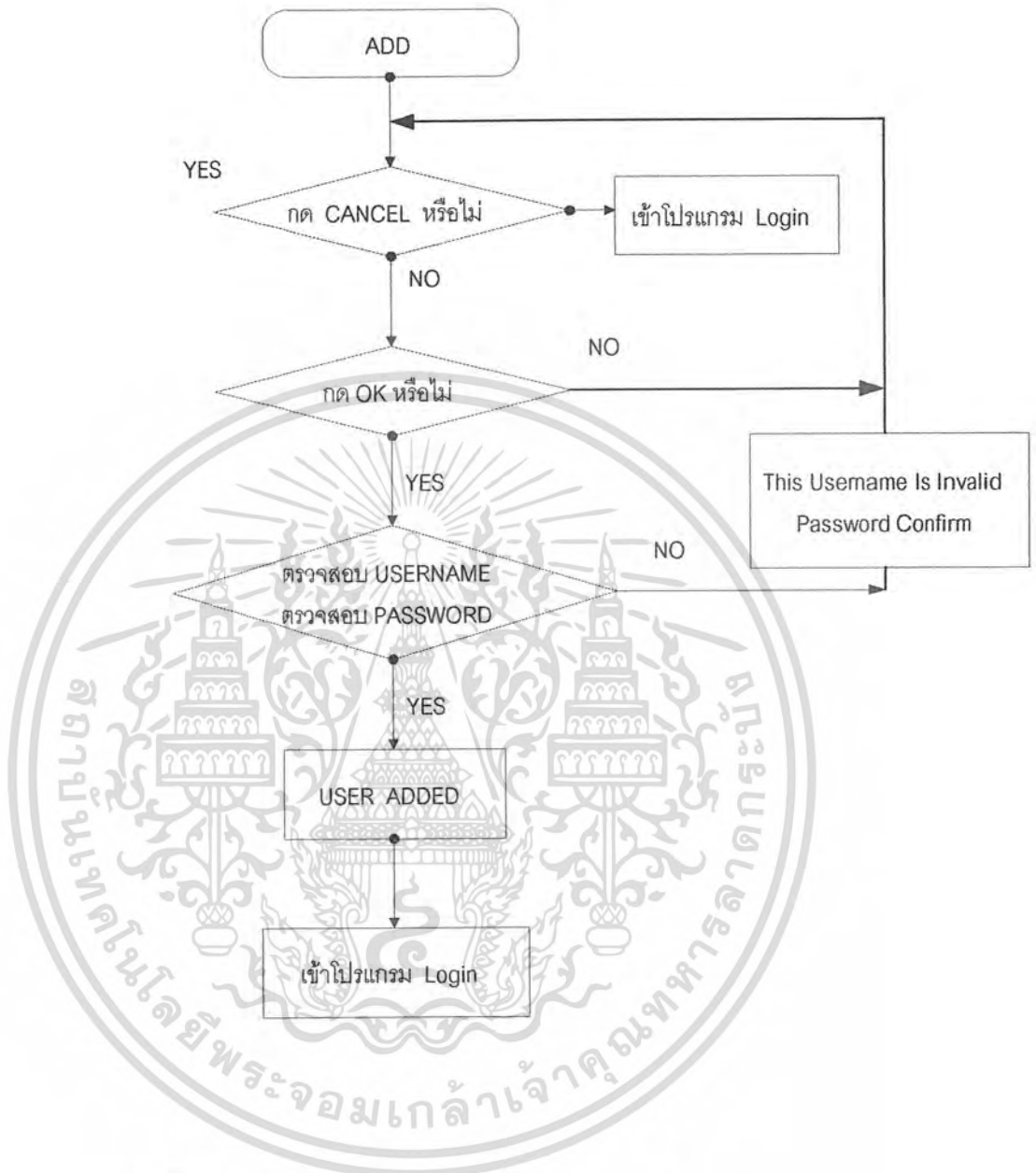
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปการทำงานของโปรแกรม Visual Basic ได้ดังนี้



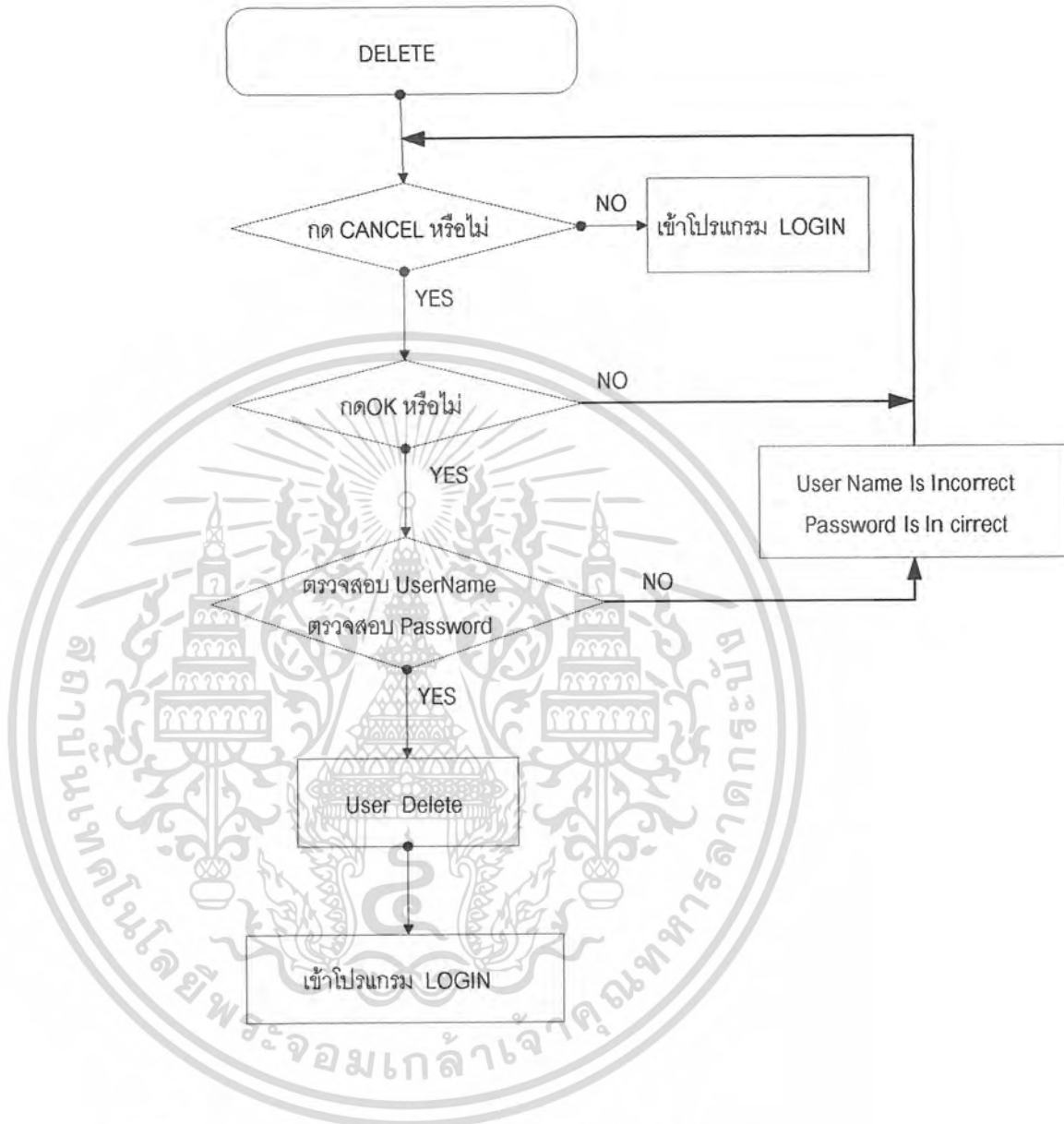
รูปที่ 4.20 FLOWCHART การทำงานของโปรแกรม Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



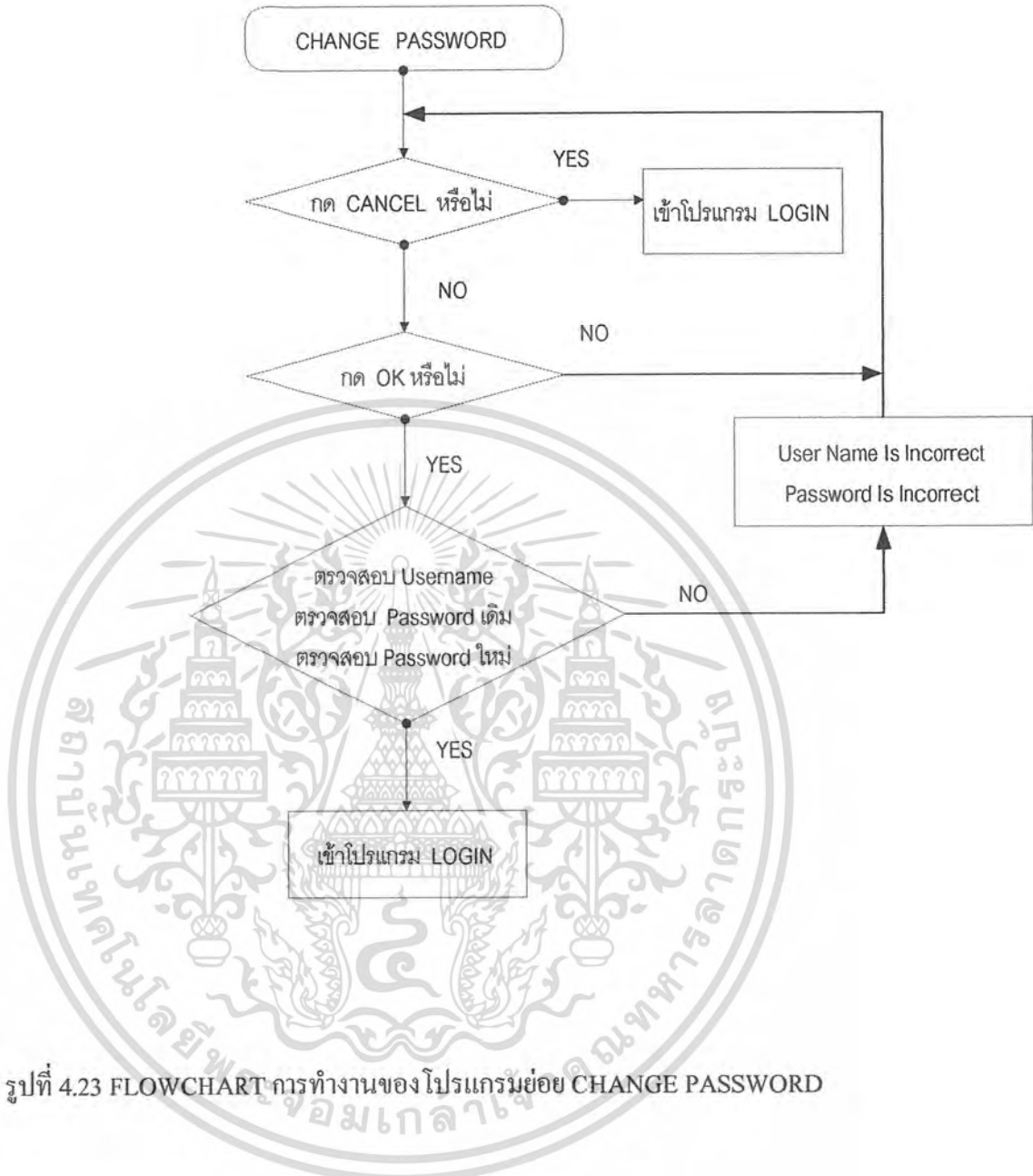
รูปที่ 4.21 FLOWCHART การทำงานของโปรแกรมย่อย ADD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



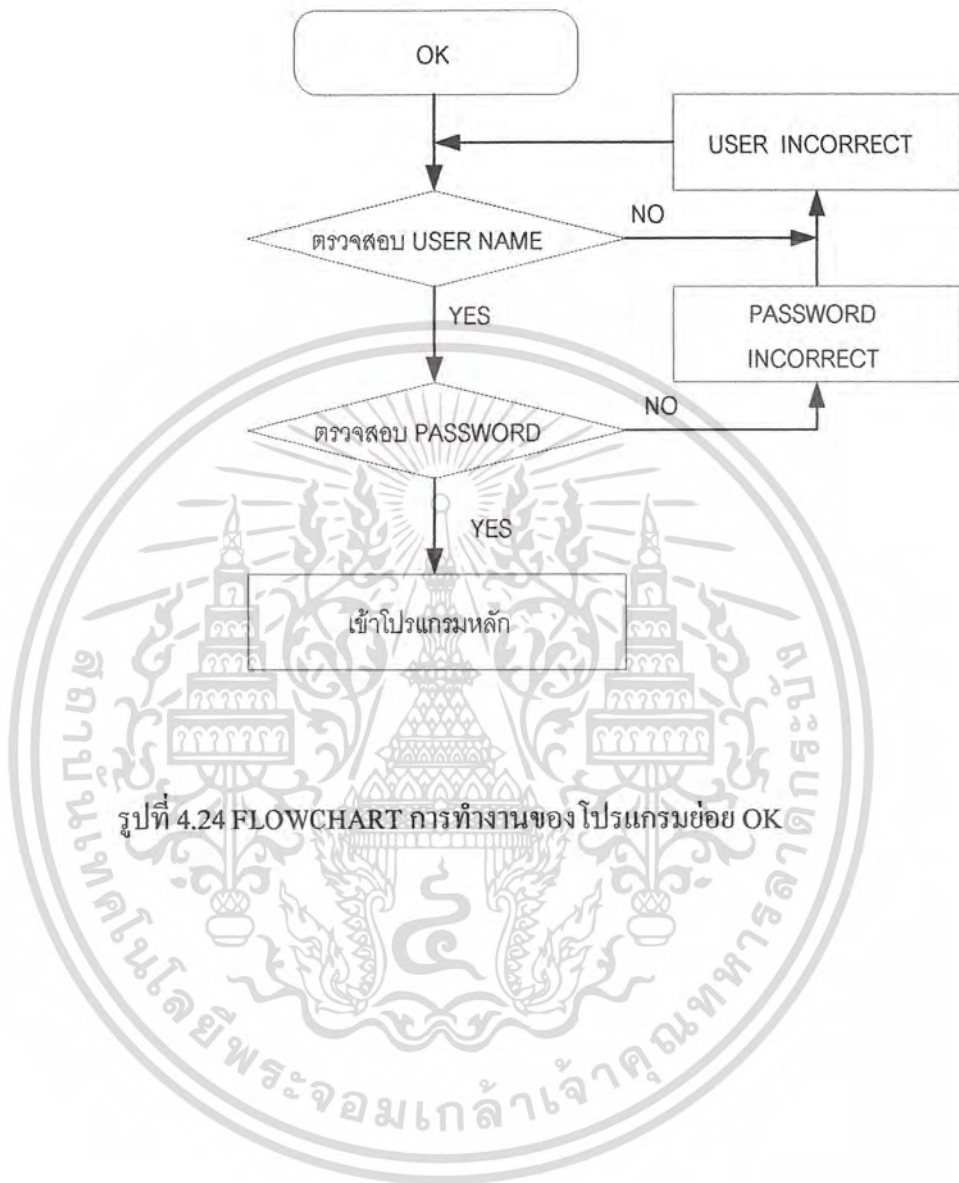
รูปที่ 4.22 FLOWCHART การทำงานของ โปรแกรมย่อย DELETE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

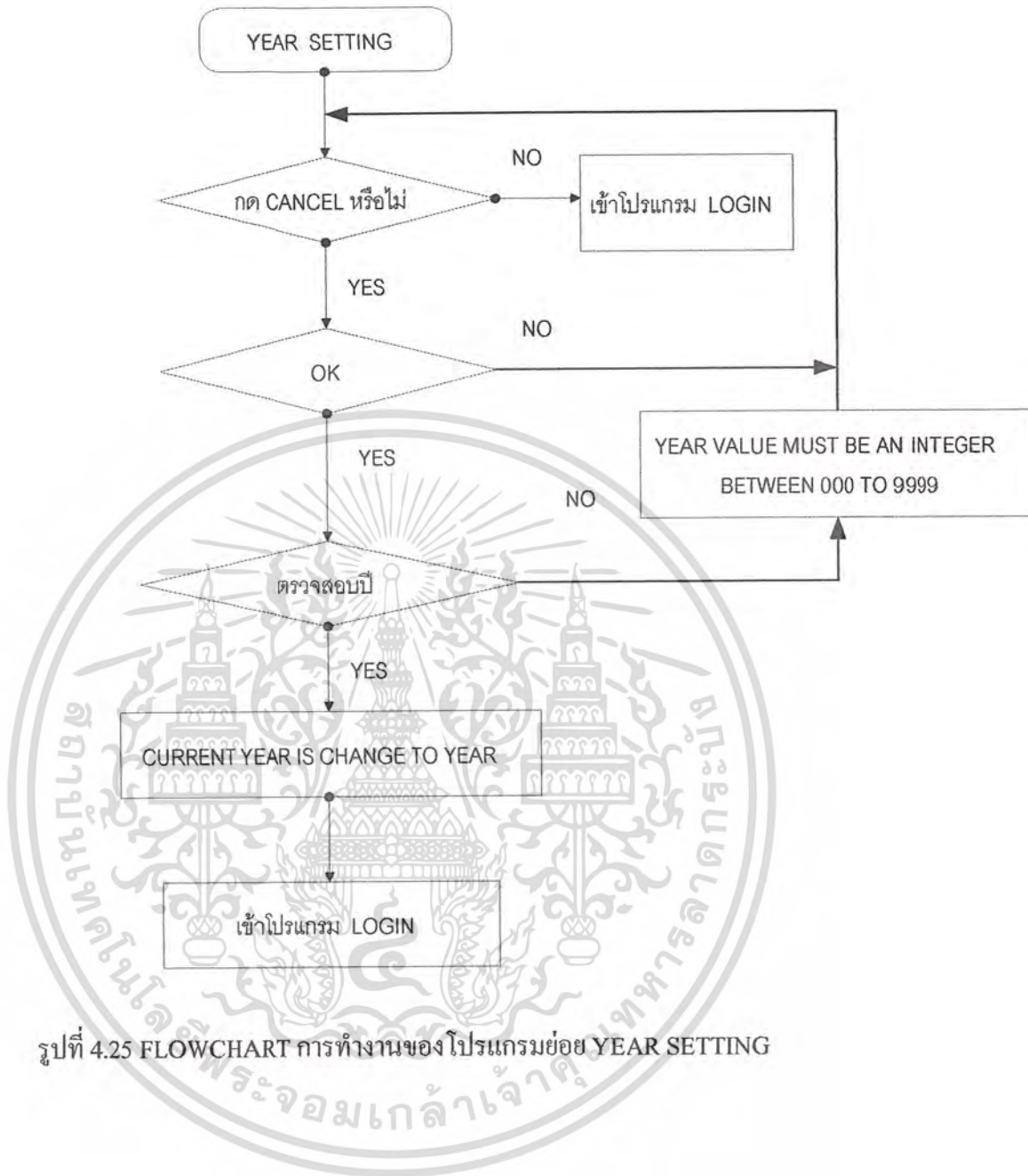


รูปที่ 4.23 FLOWCHART การทำงานของ โปรแกรมย่อย CHANGE PASSWORD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

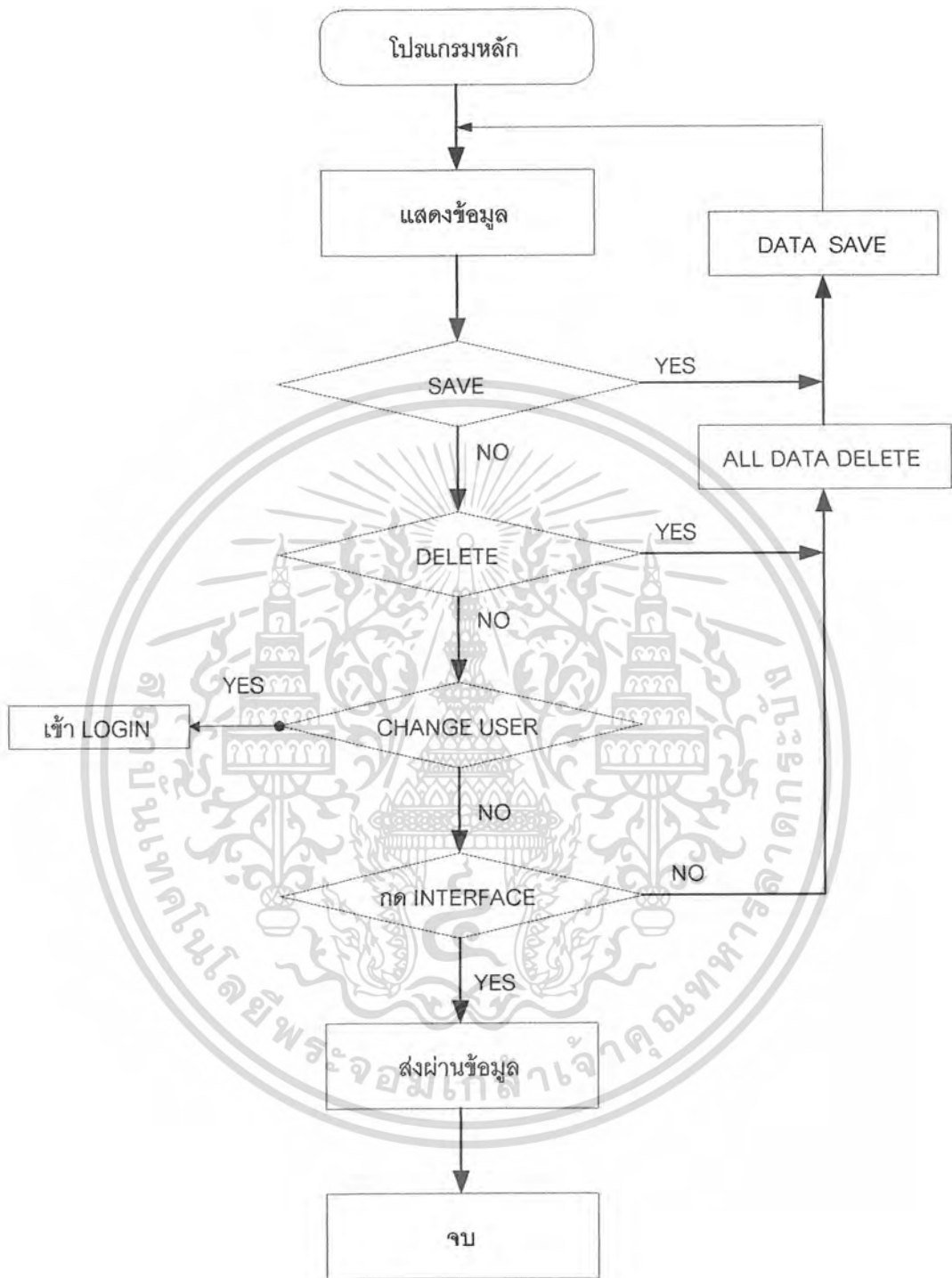


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 FLOWCHART การทำงานของโปรแกรมย่อย YEAR SETTING

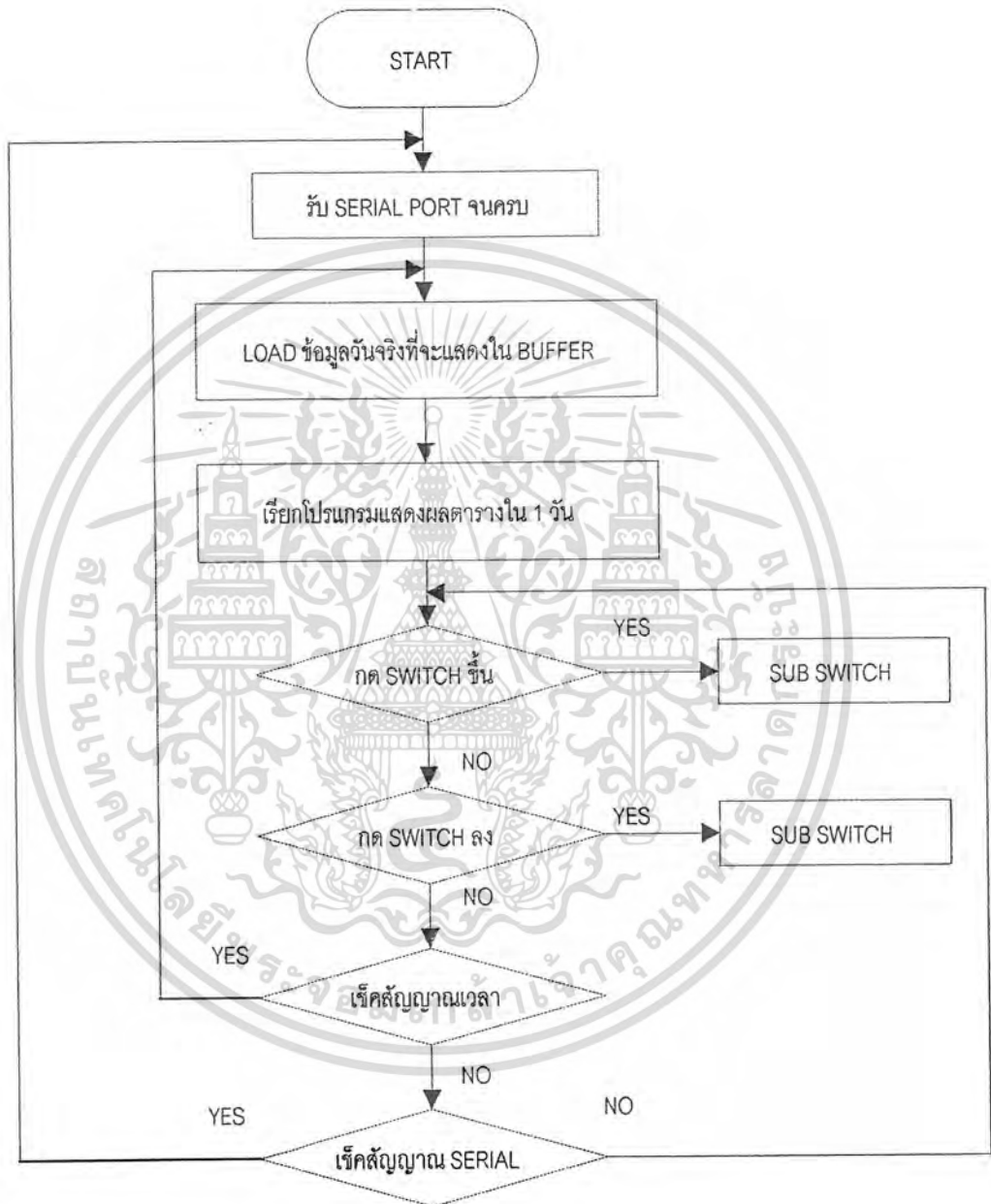
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 FLOWCHART การทำงานของ โปรแกรมหลัก

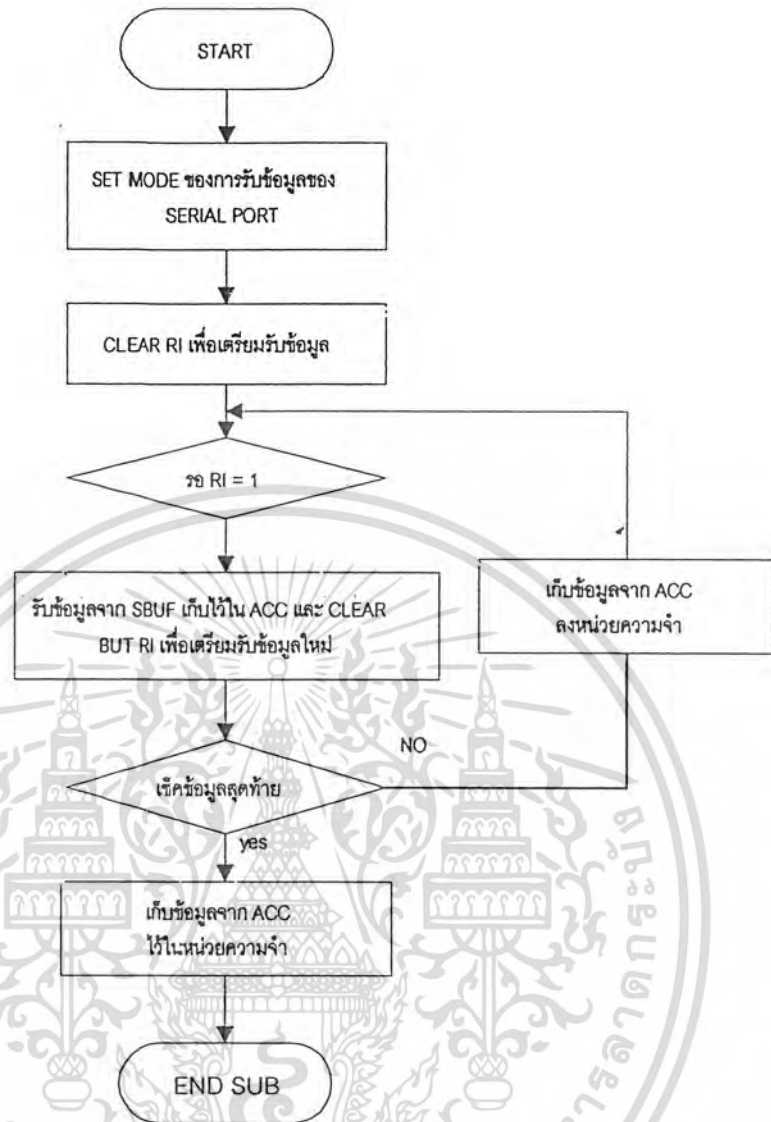
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โปรแกรมการทำงานและโครงสร้างของโครงการ



รูปที่ 5.1 FLOWCHART การทำงานของ โปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



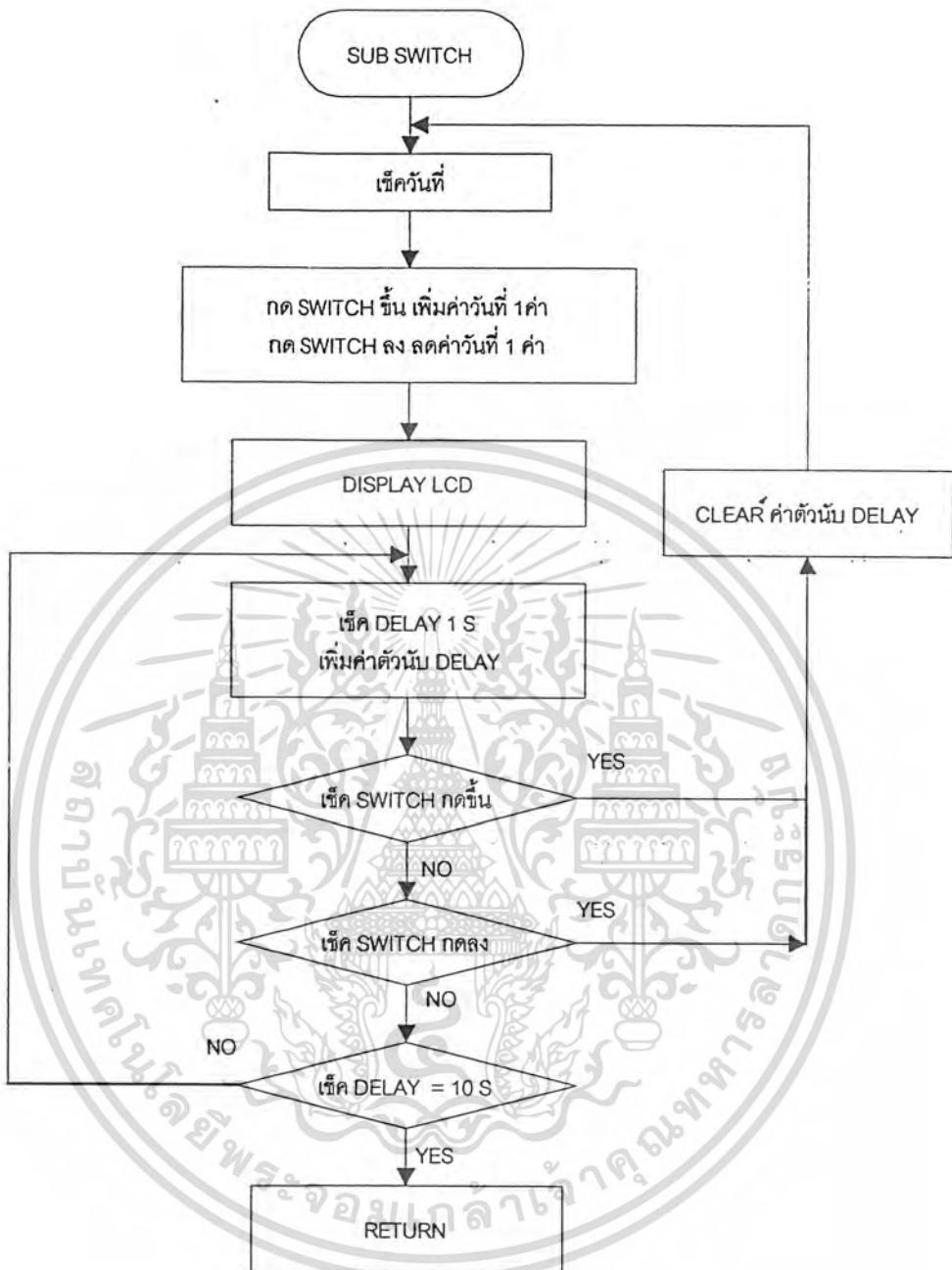
รูปที่ 5.2 FLOWCHART โปรแกรมย่อยการรับข้อมูลจากคอมพิวเตอร์ผ่านทาง RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



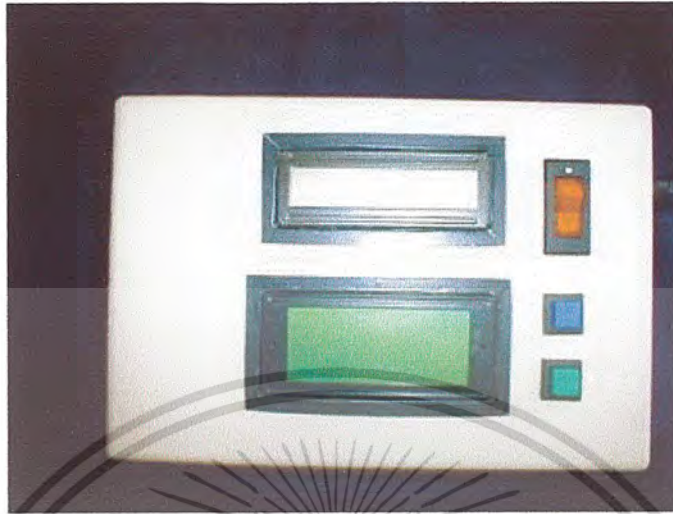
รูปที่ 5.3 FLOWCHART โปรแกรมย่อยการแสดงผลตารางการทำงานใน 1 วัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 FLOWCHART โปรแกรมย่อย การทำงานของสวิตช์

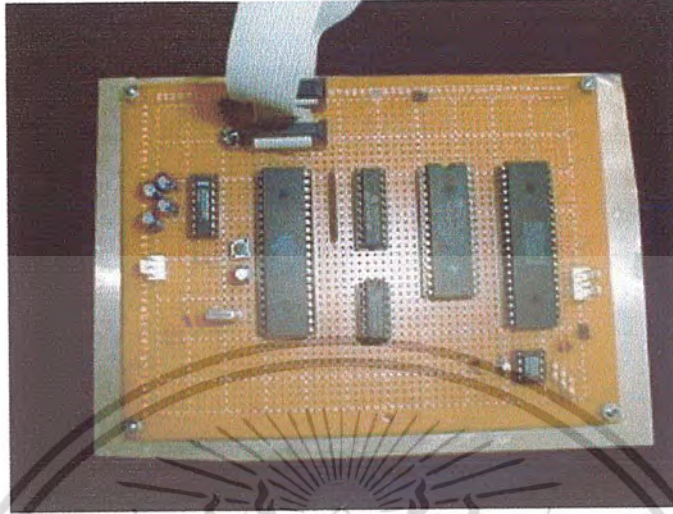
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 ภาพรวมของชิ้นงาน

จากภาพชิ้นงาน จะประกอบด้วย จอแสดงผล LCD 2 ชุด ชุดแรกเป็น จอแสดงผล LCD ขนาด 16 ตัวอักษร 1 บรรทัด ไว้สำหรับแสดงชื่อผู้ใช้งาน ส่วนจอแสดงผล LCD ชุดที่สอง เป็นจอแสดงผล LCD ขนาด 16 ตัวอักษร 4 บรรทัด บรรทัดที่ 1 ไว้สำหรับแสดงวัน เดือน ปี บรรทัดที่ 2 ไว้สำหรับแสดง ข้อมูลในช่วงเวลาเช้า ตั้งแต่ 9.00-12.00 น. บรรทัดที่ 3 ไว้สำหรับแสดง ข้อมูลในช่วงเวลาบ่าย ตั้งแต่ 13.00-16.00 น. บรรทัดที่ 4 ไว้สำหรับแสดง ข้อมูลในช่วงเวลาเย็น ตั้งแต่ 17.00-20.00 น. มีสวิทช์ 2 ปุ่ม ไว้สำหรับกดเลื่อนดูข้อมูลวันก่อนหน้า และวันถัดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 วงจรภายในของโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและสรุปผล

6.1 ขั้นตอนการทดลอง

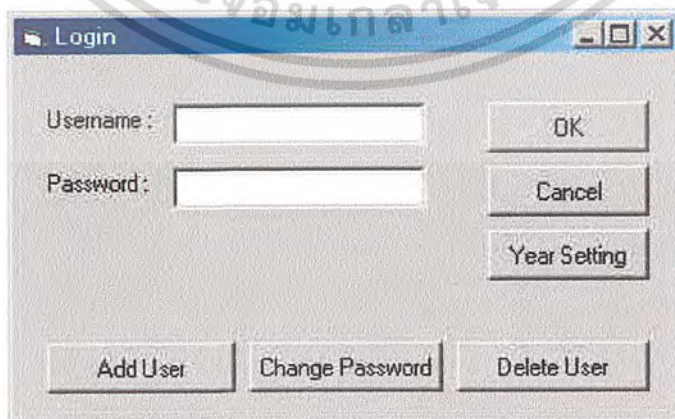
แบ่งเป็น 2 ส่วนคือการเปลี่ยนแปลงข้อมูลการแสดงผลผ่านทางจอคอมพิวเตอร์และส่วนฮาร์ดแวร์การแสดงผลโดย LCD โดยการควบคุมไมโครคอนโทรลเลอร์

	9.00-12.00	13.00-16.00	17.00-20.00
MONDAY	B 405 (B)	E12-710 (B)	HOME
TUESDAY	B 302 (F)	E12-310 (B)	E12-305 (B)
WEDNESDAY	HOME	B 308 (F)	E12-305 (B)
THURSDAY	A 205 (B)	E12-305 (B)	HOME
FRIDAY	E12-305 (F)	A 405 (F)	HOME
SATURDAY	B 405 (F)	A 504 (B)	E12-310 (B)
SUNDAY	A 504 (F)	E12-708 (B)	E12-305 (B)

ตารางที่ 6.1 ตารางการทำงาน

จากตารางการทำงาน นำข้อมูลบันทึกลงชิ้นงาน ดังนี้

1. เข้าสู่โปรแกรมเปลี่ยนแปลงข้อมูล VISUAL BASIC



รูปที่ 6.1 หน้าต่างแรกของโปรแกรม VISUAL BASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้าเป็นผู้ใช้งานใหม่ ต้องทำการเพิ่มชื่อผู้ใช้งานก่อน และโปรแกรมจะแจ้งการเพิ่มชื่อผู้ใช้งานเรียบร้อยแล้ว ดังรูปที่ 6.2 และ 6.3 ตามลำดับ

รูปที่ 6.2 เพิ่มชื่อผู้ใช้งาน

รูปที่ 6.3 แจ้งการเพิ่มชื่อเรียบร้อยแล้ว

3. ทำการ LOGIN อีกครั้ง

รูปที่ 6.4 ลงทะเบียนเข้า โปรแกรมเปลี่ยนแปลงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เข้าสู่ส่วนของการเปลี่ยนแปลงข้อมูล ถ้ามีข้อมูลเดิมอยู่ เมื่อเปิดโปรแกรมจะแสดงข้อมูลเดิมที่มีอยู่ และทำการเปลี่ยนแปลงข้อมูลดังรูปที่ 6.5 – 6.11

Table

Username: TEST YOUR NAME

26 Mar 2001

Monday

9.00 - 12.00 : BUSY at B405

13.00 - 16.00 : BUSY at E12-710

17.00 - 20.00 : HOME

Save Data to File

Delete all data

Change User

Interface

รูปที่ 6.5 หน้าต่างข้อมูลวันจันทร์

Table

Username: TEST YOUR NAME

27 Mar 2001

Tuesday

9.00 - 12.00 : FREE at B302

13.00 - 16.00 : BUSY at E12-310

17.00 - 20.00 : BUSY at E12-305

Save Data to File

Delete all data

Change User

Interface

รูปที่ 6.6 หน้าต่างข้อมูลวันอังคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table

Username: TEST YOUR NAME

28 Mar 2001

Wednesday

9.00 - 12.00 : HOME

13.00 - 16.00 : FREE at B308

17.00 - 20.00 : BUSY at E12-305

Save Data to File

Delete all data

Change User

Interface

รูปที่ 6.7 หน้าต่างข้อมูลวันพุธ

Table

Username: TEST YOUR NAME

29 Mar 2001

Thursday

9.00 - 12.00 : BUSY at A205

13.00 - 16.00 : BUSY at E12-305

17.00 - 20.00 : HOME

Save Data to File

Delete all data

Change User

Interface

รูปที่ 6.8 หน้าต่างข้อมูลวันพฤหัสบดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table

Username: TEST YOUR NAME

30 Mar 2001

Friday

9.00 - 12.00 : FREE at E12-305

13.00 - 16.00 : FREE at A405

17.00 - 20.00 : HOME

Save Data to File

Delete all data

Change User

Interface

รูปที่ 6.9 หน้าต่างข้อมูลวันศุกร์

Table

Username: TEST YOUR NAME

31 Mar 2001

Saturday

9.00 - 12.00 : FREE at B405

13.00 - 16.00 : BUSY at A504

17.00 - 20.00 : BUSY at E12-310

Save Data to File

Delete all data

Change User

Interface

รูปที่ 6.10 หน้าต่างวันเสาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows a software window titled "Table" with the following fields and buttons:

- Username: TEST YOUR NAME
- Room/Day: 1, Apr, 2001
- Day: Sunday
- 9.00 - 12.00 : FREE at A504
- 13.00 - 16.00 : BUSY at E12-708
- 17.00 - 20.00 : BUSY at E12-305
- Buttons: Save Data to File, Delete all data, Change User, Interface

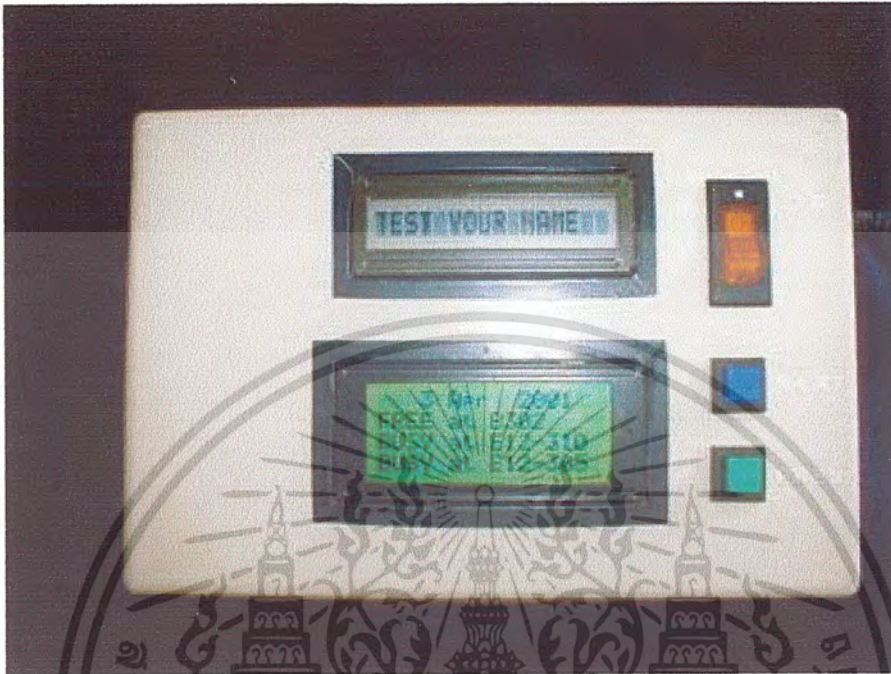
รูปที่ 6.11 หน้าต่างข้อมูลวันอาทิตย์

5. เมื่อทำการเปลี่ยนแปลง หรือเพิ่มข้อมูล เรียบร้อยแล้ว ถ้าต้องการส่งผ่านข้อมูลเพื่อ
ให้แสดงผลทางชิ้นงาน โดยทำการกดปุ่ม Interface ข้อมูลจะถูกส่งผ่านไปยังชิ้นงาน
เมื่อทำการส่งผ่านข้อมูล ชิ้นงานจะแสดงข้อมูลที่ได้ส่งไป ดังรูปที่ 6.12



รูปที่ 6.12 ชิ้นงานแสดงข้อมูลวันจันทร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.13 ชิ้นงานแสดงข้อมูลวันอังคาร

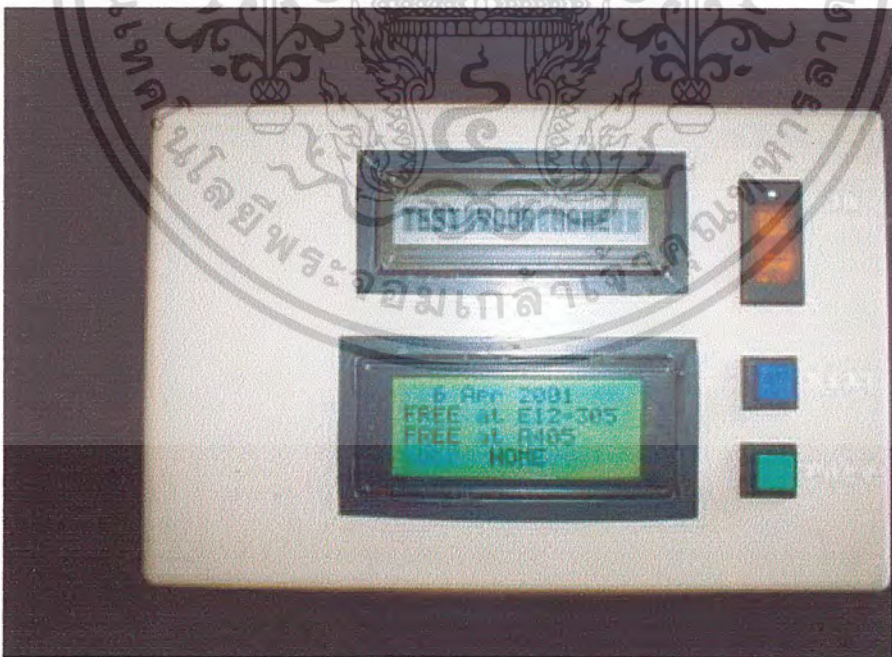


รูปที่ 6.14 ชิ้นงานแสดงข้อมูลวันพุธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.15 ชิ้นงานแสดงข้อมูลวันหยุดหัตถ์



รูปที่ 6.16 ชิ้นงานแสดงข้อมูลวันศุกร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

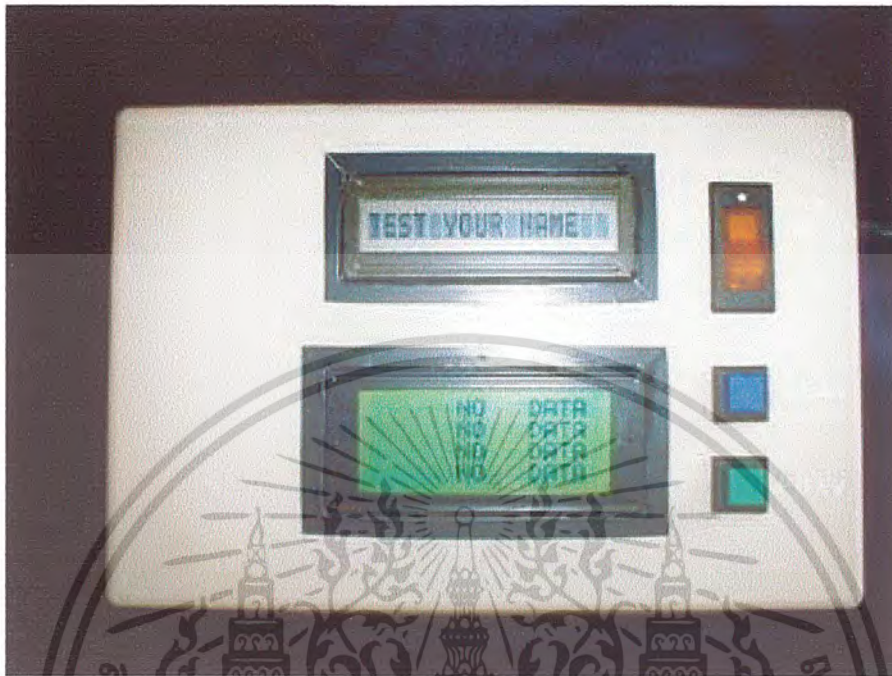


รูปที่ 6.17 ชิ้นงานแสดงข้อมูลวันเสาร์



รูปที่ 6.18 ชิ้นงานแสดงข้อมูลวันอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.19 ชิ้นงานแสดงไม่มีข้อมูลที่จะแสดง

6.2 สรุปผลการทำงาน

การทำงานในส่วนแรกนี้ ชิ้นงานยังไม่ได้เป็นไปตามเป้าหมายที่วางแผนไว้ เนื่องจากจากความผิดพลาดจากการต่อวงจรและการออกแบบที่เปลี่ยนหลายครั้งเพื่อความเหมาะสมกับโครงงานและจุดประสงค์ของโครงงานที่ใช้ในการแสดงตารางการทำงาน การเขียนโปรแกรมควบคุมการทำงานยังไม่สมบูรณ์ ทั้งในส่วนภาษาแอสเซมบลี และ Visual Basic ซึ่งต้องพัฒนาเพิ่มเติมอีก เพื่อให้การทำงานเป็นไปตามจุดประสงค์ที่ต้องการ

ข้อจำกัดของชิ้นงานทำให้รองรับความต้องการของผู้ใช้งานได้ไม่ทั่วถึง เช่น ถ้ามีผู้ใช้งานมากกว่า 1 คน หรือ ผู้ใช้งานต้องการเก็บข้อมูลมากเกินกว่าหน่วยความจำที่มีอยู่ เป็นต้น ทำให้ชิ้นงานจำเป็นต้องมีการพัฒนาให้ดียิ่งขึ้นต่อไป

6.3 แนวทางการพัฒนา

จากการออกแบบในขั้นแรกจะใช้ไดโอดเปล่งแสงในการแสดงตารางโดยใช้แบบ DOT MATRIX LED DISPLAY ซึ่งจะต้องใช้ LED จำนวนมาก และขนาดแผงแสดงผลต้องมีขนาดใหญ่ เพื่อจะได้แสดงข้อมูลได้อย่างครบถ้วน จึงไม่เหมาะสมกับตารางแสดงผลการทำงานที่ต้องการ ซึ่งต้องการส่วนแสดงผลที่ไม่ใหญ่มาก จึงเปลี่ยนมาใช้จอแสดงผลชนิด LCD ที่มีขนาดเล็กกว่ามากเมื่อต้องการจำนวนข้อมูลในปริมาณที่เท่ากัน อีกเหตุผลที่สำคัญ คือ การสิ้นเปลืองพลังงานของการใช้ DOT MATRIX LED DISPLAY จะต้องใช้กำลังงานที่มากกว่า LCD ซึ่งตารางเวลาการทำงานจำเป็นจะต้องให้แสดงตลอดเวลา จึงต้องใช้พลังงานที่น้อย แม้ไฟฟ้าดับก็สามารถจ่ายพลังงานสำรองได้โดยแบตเตอรี่ การใช้ LCD ในการแสดงผลตารางการทำงานจึงเป็นเหตุผลที่เหมาะสมกว่า แต่ข้อเสียของการใช้จอแสดงผล LCD คือ ราคาค่อนข้างแพง และถ้าเสียจะเสียทั้งจอซึ่งแตกต่างจาก LED ที่เสียตัวไหนก็สามารถเปลี่ยนเฉพาะตัวนั้นได้

เนื่องจากข้อจำกัดของขนาด LCD ทำให้ตัวอักษรที่สามารถแสดงผลได้มีเพียง 16 ตัวอักษร ถ้าผู้ใช้งานต้องการแสดงข้อความที่มีตัวอักษรมากกว่า 16 ตัวอักษร ก็จะไม่สามารถทำได้ สามารถแก้ไขได้โดยการเปลี่ยนขนาดของ LCD เป็นแบบ 20 ตัวอักษร หรือ 24 ตัวอักษร เป็นต้น

จากชิ้นงานในตอนที่แล้ว ได้ทำการเปลี่ยนหน่วยความจำจาก EEPROM เป็น RAM ทำให้สามารถเก็บข้อมูลได้มากขึ้น แต่ก็มีผู้ใช้งานหลายคน และมีข้อมูลที่ต้องจัดเก็บมาก ก็อาจจะทำให้หน่วยความจำไม่เพียงพอกับความต้องการได้ แต่สามารถพัฒนาให้เก็บข้อมูลได้มากขึ้น โดยการเปลี่ยนหน่วยความจำจากเดิม ให้มีความจุมากขึ้น หรือทำการขยายหน่วยความจำ

ในระหว่างการพัฒนาวงจรการต่อวงจรโดยทางแผ่นต่อวงจร (Photo Board) ถ้าต่อได้ไม่ดีพอการทำงานของวงจรที่ออกแบบไว้จะไม่เป็นตามที่เราคาดไว้ จึงทำให้หลายครั้งต้องแก้วงจรใหม่ ทั้งที่วงจรเดิมไม่ได้มีอะไรจากการออกแบบ ในขั้นตอนการพัฒนาครั้งนี้เมื่อออกแบบได้แล้วจึงต้องวงจรลงแผ่นวงจรเอนกประสงค์ เพื่อลดความผิดพลาดอันเนื่องมาจากการต่อวงจรและง่ายต่อการพัฒนาวงจรเพิ่มเติมก่อนการลงแผ่นลายปริ้นท์ (Print Circuit Board) ในขั้นตอนสุดท้าย แต่บางครั้งการต่อแผ่นลายปริ้นท์เอนกประสงค์ การเชื่อมสายไฟก็ยังไม่แน่นอน มีการผิดพลาดในส่วนนี้เกิดขึ้นหลังจากการเขียนโปรแกรมควบคุม โดยใช้โปรแกรมเดิมแต่เมื่อขยับสายไฟ การแสดงผลก็ผิดพลาดไปจากเดิมได้

โปรแกรมภาษา Visual Basic 6.0 การเขียนโปรแกรมให้ทำงานตามที่เราคือต้องการจำเป็นจะต้องพัฒนาต่อเพิ่มเติมเพื่อให้การทำงานของโปรแกรมได้สมบูรณ์มากขึ้น สารถเรียกข้อมูลเข้ามาได้อย่างรวดเร็ว มีระบบป้องกันการสูญหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนาในอนาคต นอกเหนือจากแสดงข้อมูลเป็นข้อความแล้ว อาจจะแสดงข้อมูลเป็นเสียง โดยสามารถแปลงข้อความเป็นสัญญาณเสียงให้ข้อมูลแก่ผู้มาติดต่อ มีระบบตอบรับผู้มาติดต่ออัตโนมัติ แทนการเขียนบันทึกข้อความ พัฒนาระบบบันทึกเสียงผู้มาติดต่ออัตโนมัติ หรืออาจจะเชื่อมต่อกับโทรศัพท์ เพื่อให้เป็นระบบตอบรับโทรศัพท์อัตโนมัติ เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----<< defined port pin number at AT89C51 >>-----
LCDEN2    equ  p3.2      ; pin enable LCD2 (16 x 4)
LCDWR     equ  p3.3      ; pin RW LCD1 and LCD2
LCDRS     equ  p3.4      ; pin WR LCD1 and LCD2
LCDEN1    equ  p3.5      ; pin enable LCD1 (16 x 1)
DATA_LCD  equ  p1        ; data bus for LCD module
SW1       equ  p2.5      ; switch UP
SW2       equ  p2.6      ; switch DOWN

```

```

; ---<< defined bytes address >>---

```

```

        org  00H
UserReg: ds   8        ; reg R0 - R7
stack:  ds  24        ; stack depth
Sysflag: ds   1        ; address bit

```

```

; ---<< defined bits address >>---

```

```

; ---<< defined bytes address continue >>--- ;36

```

```

RecordNumber: ds 1 ; RecordNumber

```

```

;-----<< defined XMEM byte address >>-----

```

```

        org 0000h
LCDBUF1: ds 16 ;LCD1 BUFFER for LCD 16 x 1
LCDBUF2x1: ds 16 ;LCD2 BUFFER at line 1
LCDBUF2x2: ds 16 ;LCD2 BUFFER at line 2
LCDBUF2x3: ds 16 ;LCD2 BUFFER at line 3
LCDBUF2x4: ds 16 ;LCD2 BUFFER at line 4
tempEdit: ds 80
TotalRecord: ds 1 ;TotalRecord

```

```

;

```

```

        org 0100h

```

```

RecordMemory: ds 80*100 ;defined external memory for save record

```

```

StartRecord equ RecordMemory

```

```

;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; -----<< power on reset >>-----

```
org 0000h ; power on/reset vector
jmp on_start
org 0003h ; external interrupt 0 vector
reti ; undefined
org 000bh ; timer 0 overflow vector
reti ; undefined
org 0013h ; external interrupt 1 vector
reti ; undefined
org 001bh ; timer 1 overflow vector
reti ; undefined
org 0023h ; serial I/O interrupt vector
reti ; undefined
```

org 0080h

; -----<< Initialize controller registers. >>-----

```
initialize: mov PCON, #0 ; initialize power control register
mov IE, #0 ; deactivate all interrupts
mov TMOD, #00100001b ; Timer 1 Mode 2 Timer 0 Mode 1
mov SCON, #01010000B ; Serial mode 1
mov IE, #00000000b
setb PS ; enable priority level higher
mov a, #0fdh ; Set baud rate 9600
mov TH1, a
mov TL1, a
setb TR1 ; Set timer1 control run
mov p0, #11111111b
mov p1, #11111111b ; initialize port
mov p2, #11111111b
mov p3, #11111111b
call INITLCD
ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ---<< subroutine Fill ProgramMemory to External DataMemory >>---
```

```
; input : dptr = #address Pmem
```

```
; : r0 = #address Xmem
```

```
; : r2 = #number of character
```

```
FillPmemToXmem: mov a, #0
```

```
movc a, @a+dptr
```

```
mov p2, #11100000 ;select page 0
```

```
movx @r0, a
```

```
inc dptr
```

```
inc r0
```

```
djnz r2, FillPmemToXmem
```

```
ret
```

```
; -----<< CLEAR LCD BUFFER >>-----
```

```
CLEARLCDBUF: mov r2, #80 ;DATA 80 BYTES
```

```
mov r0, #LCDBUF1
```

```
mov p2, #11100000 ;select page 0
```

```
CLRLCD: mov a, #' '
```

```
movx @r0, a
```

```
inc r0
```

```
djnz r2, CLRLCD
```

```
ret
```

```
; ---<< initialize LCD Module >>---
```

```
INITLCD: ; clr LCDRS
```

```
; clr LCDWR
```

```
; clr LCDEN1
```

```
; clr LCDEN2
```

```
mov a, #00111000B ;8bit, 2 line, 5x7 dot
```

```
call LCDWI1
```

```
mov a, #00111000B ;8bit, 2 line, 5x7 dot
```

```
call LCDWI2
```

```
mov a, #00001100B ;DISPLAY ON/OFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

call LCDWI1
mov a, #00001100B ;DISPLAY ON/OFF
call LCDWI2
mov a, #00000001B ;clear and home
call LCDWI1
mov a, #00000001B ;clear and home
call LCDWI2
ret

; -----<< LCDLDX SUB >>-----
; LOAD PMEM TO LCDBUF (ALL,LINE)
; IN = DPTR
; REG = A,R0,R2,DPTR
LCDLDPL: mov r2, #80 ;LOAD ALL
mov r0, #LCDBUF1
LCDLDPL1: clr a
movc a, @a+dptr
mov p2, #11100000 ;select page 0
movx @r0, a
inc dptr
inc r0
djnz r2, LCDLDPL1
ret

LCDLD1: mov r2, #16 ;LOAD LINE-1
mov r0, #LCDBUF1
sjmp LCDLDPL1

LCDLD2: mov r2, #16 ;LOAD LINE-2
mov r0, #LCDBUF2x1
sjmp LCDLDPL1

LCDLD3: mov r2, #16 ;LOAD LINE-2
mov r0, #LCDBUF2x2
sjmp LCDLDPL1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCDLD4:   mov  r2, #16      ;LOAD LINE-2
          mov  r0, #LCDBUF2x3
          sjmp LCDLDPL1

LCDLD5:   mov  r2, #16      ;LOAD LINE-2
          mov  r0, #LCDBUF2x4
          sjmp LCDLDPL1

```

```
; -----<< LCDLDI SUB >>-----
```

```
; LOAD LCDBUFFER TO LCD-MODULE (DMC162)
```

```
; IN = LCDBUF
```

```
; REG = A,R0,R2
```

```

LCDLDI:   mov  r0, #LCDBUF1
          mov  a, #80H      ;LCD1 LINE 1 8 char
          call LCDLDX1
          mov  a, #0C0H     ;LCD1 LINE 1 8 char
          call LCDLDX1
          mov  a, #80H      ;LCD2 LINE 1
          call LCDLDX2
          mov  a, #0C0H     ;LCD2 LINE 2
          call LCDLDX2
          mov  a, #90H      ;LCD2 LINE 3
          call LCDLDX2
          mov  a, #0D0H     ;LCD2 LINE 4
          call LCDLDX2
          ret

```

```
LCDLDX1:  call LCDW1      ;LOAD ONE LINE
```

```
          mov  r2, #8      ;8 CHAR.
```

```
LCDLDX1x1: mov  p2, #11100000 ;select page 0
```

```
          movx a, @r0
```

```
          call LCDWD1     ;WRITE DATA
```

```
          inc  r0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

djnz r2, LCDLDX1x1
ret

```

```

LCDLDX2: call LCDWI2 ;LOAD ONE LINE

```

```

mov r2, #16 ;16 CHAR.

```

```

LCDLDX2x1: mov p2, #11100000 ;select page 0

```

```

movx a, @r0

```

```

call LCDWD2 ;WRITE DATA

```

```

inc r0

```

```

djnz r2, LCDLDX2x1

```

```

ret

```

```

; -----<< LCDWI1 >>-----

```

```

; LCD1 WRITE INSTRUCTION (RS=0)

```

```

;input = A

```

```

; REG = A,DPTR

```

```

LCDWI1: clr LCDRS ;RS=0

```

```

clr LCDWR ;RW=0

```

```

clr LCDEN1 ;ENABLE=0

```

```

mov DATALCD, a

```

```

sjmp LCDWI_WD1

```

```

; -----<< LCDWD1 >>-----

```

```

; LCD1 WRITE DATA (RS=1)

```

```

; IN = A

```

```

; REG = A,DPTR

```

```

LCDWD1: setb LCDRS ;RS=1

```

```

clr LCDWR ;RW=0

```

```

clr LCDEN1 ;ENABLE=0

```

```

mov DATALCD, a

```

```

LCDWI_WD1: setb LCDEN1 ;ENABLE=1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov a, #100
call delay_ms
clr LCDEN1 ;ENABLE=0
mov A,#0 ;DELAY
LCDWD1x1: dec A
jnz LCDWD1x1
ret

; -----<< LCDWI2 >>-----
; LCD2 WRITE INSTRUCTION (RS=0)
;input = A
; REG = A,DPTR
LCDWI2: clr LCDRS ;RS=0
clr LCDWR ;RW=0
clr LCDEN2 ;ENABLE=0
mov DATALCD, a
sjmp LCDWI_WD2

; -----<< LCDWD2 >>-----
; LCD2 WRITE DATA (RS=1)
; IN = A
; REG = A,DPTR
LCDWD2: setb LCDRS ;RS=1
clr LCDWR ;RW=0
clr LCDEN2 ;ENABLE=0
mov DATALCD, a
LCDWI_WD2: setb LCDEN2 ;ENABLE=1
mov a, #100
call delay_ms
clr LCDEN2 ;ENABLE=0
mov A,#0 ;DELAY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCDWD2x1: dec  A
           jnz  LCDWD2x1
           ret

```

; -----<< subroutine delay_ms >>-----

; Delay for one mS times the value in the accumulator.

```

delay_ms:  push  acc
           push  b
           mov   b, #0
dd:        djnz  b, $           ; 500 uS @ 12 MHz
           djnz  b, $           ; 500 uS @ 12 MHz
           djnz  acc, dd
           pop   b
           pop   acc
           ret

```

; -----<< subroutine delay_sec >>-----

; Delay for one second times the value in the accumulator.

```

delay_sec: push  acc
           push  b
           mov   b, a
ddd:       mov   a, #250
           lcall delay_ms      ; 250 mS
           lcall delay_ms      ; 500 mS
           lcall delay_ms      ; 750 mS
           lcall delay_ms      ; 1000 mS
           djnz  b, ddd
           pop   b
           pop   acc
           ret

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; -----<< subroutine WriteRecord >>-----
;input : acc = RecordNumber, tempEdit
;output: none
;reg. : acc, b, dptr, r0, r7
;action: tempEdit -- move data --> StartRecord + RecordNumber*80
;

```

```

WriteRecord: mov dptr, #StartRecord

```

```

    mov b, #80

```

```

    mul ab

```

```

    add a, dpl

```

```

    mov dpl, a

```

```

    mov a, b

```

```

    addc a, dph

```

```

    mov dph, a

```

```

;

```

```

    mov r0, #tempEdit

```

```

    mov r7, #80

```

```

WRRecord1: mov p2, #11100000 ;select page 0

```

```

    movx a, @r0

```

```

    movx @dptr, a

```

```

    inc r0

```

```

    inc dptr

```

```

    djnz r7, WRRecord1

```

```

    ret

```

```

; -----<< subroutine ReadRecord >>-----

```

```

;input : acc = RecordNumber, tempEdit

```

```

;output: none

```

```

;reg. : acc, b, dptr, r0, r7

```

```

;action: tempEdit <-- move data -- StartRecord + RecordNumber*80

```

```

;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ReadRecord:  mov  dptr, #StartRecord
```

```
    mov  b, #80
```

```
    mul  ab
```

```
    add  a, dpl
```

```
    mov  dpl, a
```

```
    mov  a, b
```

```
    addc a, dph
```

```
    mov  dph, a
```

```
    ;
```

```
    mov  r0, #tempEdit
```

```
    mov  r7, #80
```

```
RRecord1:  movx  a, @dptr
```

```
    mov  p2, #11100000    ;select page 0
```

```
    movx @r0, a
```

```
    inc  r0
```

```
    inc  dptr
```

```
    djnz r7, RRecord1
```

```
    ret
```

```
; -----<< subroutine Move memory tempEdit to memory LCDBUFFER >>-----
```

```
;input : acc = RecordNumber, tempEdit
```

```
;output: none
```

```
;reg.  : acc, b, dptr, r0, r7
```

```
;action: tempEdit -- move data --> LCDBUFFER
```

```
;
```

```
MovEditToLCDBUF:mov  r0, #tempEdit
```

```
    mov  r1, #LCDBUF1
```

```
    mov  r7, #80
```

```
MoveRecord1: mov  p2, #11100000    ;select page 0
```

```
    movx a, @r0
```

```
    movx @r1, a
```

```
    inc  r0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
inc r1
djnz r7, MoveRecord1
ret
```

```
GendData: mov dptr, #texttest
mov r0, #tempEdit
mov r2, #80
call FillPmemToXmem
mov a, #0
call WriteRecord
```

```
mov dptr, #texttest+(80*1)
mov r0, #tempEdit
mov r2, #80
call FillPmemToXmem
mov a, #1
call WriteRecord
```

```
mov dptr, #texttest+(80*2)
mov r0, #tempEdit
mov r2, #80
call FillPmemToXmem
mov a, #2
call WriteRecord
```

```
mov dptr, #texttest+(80*3)
mov r0, #tempEdit
mov r2, #80
call FillPmemToXmem
mov a, #3
call WriteRecord
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mov  dptr, #texttest+(80*4)
mov  r0, #tempEdit
mov  r2, #80
call  FillPmemToXmem
mov  a, #4
call  WriteRecord
```

```
mov  dptr, #texttest+(80*5)
mov  r0, #tempEdit
mov  r2, #80
call  FillPmemToXmem
mov  a, #5
call  WriteRecord
```

```
mov  dptr, #texttest+(80*6)
mov  r0, #tempEdit
mov  r2, #80
call  FillPmemToXmem
mov  a, #6
call  WriteRecord
```

```
mov  dptr, #texttest+(80*7)
mov  r0, #tempEdit
mov  r2, #80
call  FillPmemToXmem
mov  a, #7
call  WriteRecord
```

```
mov  dptr, #texttest+(80*8)
mov  r0, #tempEdit
mov  r2, #80
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
call FillPmemToXmem
```

```
mov a, #8
```

```
call WriteRecord
```

```
mov dptr, #texttest+(80*9)
```

```
mov r0, #tempEdit
```

```
mov r2, #80
```

```
call FillPmemToXmem
```

```
mov a, #9
```

```
call WriteRecord
```

```
mov dptr, #texttest+(80*10)
```

```
mov r0, #tempEdit
```

```
mov r2, #80
```

```
call FillPmemToXmem
```

```
mov a, #10
```

```
call WriteRecord
```

```
mov dptr, #texttest+(80*11)
```

```
mov r0, #tempEdit
```

```
mov r2, #80
```

```
call FillPmemToXmem
```

```
mov a, #11
```

```
call WriteRecrd
```

```
mov dptr, #texttest+(80*12)
```

```
mov r0, #tempEdit
```

```
mov r2, #80
```

```
call FillPmemToXmem
```

```
mov a, #12
```

```
call WriteRecord
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov  dptr, #texttest+(80*13)
mov  r0, #tempEdit
mov  r2, #80
call FillPmemToXmem
mov  a, #13
call WriteRecord

```

```

mov  dptr, #texttest+(80*14)
mov  r0, #tempEdit
mov  r2, #80
call FillPmemToXmem
mov  a, #14
call WriteRecord
mov  dptr, #texttest+(80*15)
mov  r0, #tempEdit
mov  r2, #80
call FillPmemToXmem
mov  a, #15
call WriteRecord

```

```
ret
```

```
;-----<< main program >>-----
```

```

on_start:  mov  sp, #stack    ;initialize stack pointer
           mov  r0, #8      ;clear INT-RAM and delay
           mov  r2, #80h-8
res0:     mov  @r0, #0
           inc  r0
           djnz r2, res0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov a, #1
call delay_sec
call initialize
;
call CLEARLCDBUF
call LCDLDI
mov RecordNumber, #0
; start add subroutine for test system
call GendData
; end
MainLoop: mov a, RecordNumber
call ReadRecord
;
call MovEditToLCDBUF
call LCDLDI
ChkKey1: jb SW1, ChkKey2 ; Key UP
inc RecordNumber
mov a, RecordNumber
cjne a, #101, ChkNokey
mov RecordNumber, #0
jmp ChkNoKey
ChkKey2: jb SW2, ChkNoKey ; Key DOWN
dec RecordNumber
mov a, RecordNumber
cjne a, #-1, ChkNokey
mov RecordNumber, #100
jmp ChkNoKey
ChkNoKey:
jmp MainLoop
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VISUAL BASIC

***** ADD USER *****

Option Explicit

Dim FilePos As Long

Dim iCounter As Integer

Private Sub AddB_Click()

 LoginForm.UData.Recordset.MoveFirst

 LoginForm.UData.Recordset.FindFirst "UName="" & NameTB.Text & ""

 If LoginForm.UData.Recordset.NoMatch Then

 If PassTB.Text = ConfTB.Text Then

 LoginForm.UData.Recordset.AddNew

 LoginForm.UData.Recordset.UName.Value = NameTB.Text

 LoginForm.UData.Recordset.PassW.Value = PassTB.Text

 LoginForm.UData.Recordset.Update

 FilePos = FreeFile

 Open "c:\UserDir\" + NameTB.Text + ".usr" For Random As #FilePos Len = 1098

 For iCounter = 1 To 1098

 Put #FilePos, iCounter, ""

 Next iCounter

 Close #FilePos

 MsgBox "User : " + NameTB.Text + " Added", vbOKOnly, "Add User"

 Unload Me

 Else

 MsgBox "Invalid password confirmation", vbOKOnly, "Caution"

 End If

 Else

 MsgBox "This username is already existed", vbOKOnly, "Add User"

 End If

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub CancelB_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    LoginForm.Show
```

```
End Sub
```

```
***** CHANGE PASSWORD *****
```

```
Option Explicit
```

```
Private Sub CancelB_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub ChgB_Click()
```

```
    LoginForm.UData.Recordset.MoveFirst
```

```
    LoginForm.UData.Recordset.FindFirst "UName=" & NameTB.Text & ""
```

```
    If LoginForm.UData.Recordset.NoMatch Or  
    LoginForm.UData.Recordset.UName.Value <> NameTB.Text Then
```

```
        MsgBox "Username is incorrect", vbOKOnly, "Caution"
```

```
    Exit Sub
```

```
End If
```

```
If LoginForm.UData.Recordset.PassW.Value = OldPassTB.Text Then
```

```
    If PassTB.Text = ConfTB.Text Then
```

```
        LoginForm.UData.Recordset.Edit
```

```
        LoginForm.UData.Recordset.PassW.Value = PassTB.Text
```

```
        LoginForm.UData.Recordset.Update
```

```
        MsgBox "Password changed successfully", vbOKOnly, "Change Password"
```

```
        Unload Me
```

```
    Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MsgBox "Invalid password confirmation", vbOKOnly, "Caution"
    End If
Else
    MsgBox "Password is incorrect", vbOKOnly, "Caution"
End If
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    LoginForm.Show
End Sub

```

***** DELETE USER *****

```

Option Explicit
Dim FilePos As Long
Dim iCounter As Integer

```

```

Private Sub CancelB_Click()
    Unload Me
End Sub

```

```

Private Sub DelB_Click()
    LoginForm.UData.Recordset.MoveFirst
    LoginForm.UData.Recordset.FindFirst "UName=" & NameTB.Text & ""
    If LoginForm.UData.Recordset.NoMatch Or
LoginForm.UData.Recordset.UName.Value <> NameTB.Text Then
        MsgBox "Username is incorrect", vbOKOnly, "Caution"
    Exit Sub
End If
    If LoginForm.UData.Recordset.PassW.Value = PassTB.Text Then
        LoginForm.UData.Recordset.Delete

```

```

        FilePos = FreeFile

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Open "c:\UserDir" + NameTB.Text + ".usr" For Random As #FilePos Len = 1098
```

```
For iCounter = 1 To 1098
```

```
Put #FilePos, iCounter, ""
```

```
Next iCounter
```

```
Close #FilePos
```

```
MsgBox "User : " + NameTB.Text + " Deleted", vbOKOnly, "Delete User"
```

```
Unload Me
```

```
Else
```

```
MsgBox "Password is incorrect", vbOKOnly, "Caution"
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
LoginForm.Show
```

```
End Sub
```

```
**** LOGIN ****
```

```
Option Explicit
```

```
Dim FilePos As Long
```

```
Dim iCounter As Integer
```

```
Private Sub AddB_Click()
```

```
Unload Me
```

```
AddUserForm.Show
```

```
End Sub
```

```
Private Sub CancelButton_Click()
```

```
End
```

```
End Sub
```

```
Private Sub ChgB_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Unload Me
ChgPWForm.Show
End Sub
```

```
Private Sub DelB_Click()
Unload Me
DelUserForm.Show
End Sub
```

```
Private Sub Form_Load()
FilePos = FreeFile
Open "c:\UserDir\Config.tfp" For Random As #FilePos
Get #FilePos, 1, CYear
Close #FilePos
End Sub
```

```
Private Sub OKButton_Click()
UData.Recordset.MoveFirst
UData.Recordset.FindFirst "UName=" & NameTB.Text & ""
If UData.Recordset.NoMatch Or UData.Recordset.UName.Value <> NameTB.Text
Then
MsgBox "Username is incorrect", vbOKOnly, "Caution"
Exit Sub
End If
```

```
If UData.Recordset.PassW.Value = PassTB.Text Then
```

```
Me.Hide
UsrName = NameTB.Text
Mainform.Show
Mainform.NameTB.Text = UsrName
Call Mainform.DayUD_Change
Unload Me
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    MsgBox "Password is incorrect", vbOKOnly, "Caution"
End If
End Sub

Private Sub SetButton_Click()
    SetYearDialog.Show vbModal, Me
End Sub

***** SET YEAR *****
Option Explicit
Dim FilePos As Long

Private Sub Form_Load()
    CYL.Caption = CYL.Caption + Str(CYear)
End Sub

Private Sub OKButton_Click()
    If Not (IsNumeric(NYTB.Text)) Then
        MsgBox "Invalid Input Format", vbOKOnly, "Caution"
        Exit Sub
    End If

    If Val(NYTB.Text) >= 100 And Val(NYTB.Text) <= 9999 And Val(NYTB.Text) / 1 = Int
    (Val(NYTB.Text)) Then
        CYear = Val(NYTB.Text)
        FilePos = FreeFile
        Open "c:\UserDir\Config.ttp" For Random As #FilePos
        Put #FilePos, 1, CYear
        Close #FilePos
        MsgBox "Current year is change to : " + Str(CYear), vbOKOnly, "Year Setting"
        Unload Me

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Else

MsgBox "Year value must be an integer between 100 to 9999", vbOKOnly, "Caution"

End If

End Sub

***** MAIN FORM *****

Option Explicit

Dim MCon As Integer, AllCon As Integer

Dim MTBSt As String, ATBSt As String, ETBSt As String

Dim TempDate As Date

Dim iCounter As Integer

Dim FilePos As Long

Private Sub ATB_Change()

AArray(AllCon) = ATB.Text

End Sub

Private Sub CUser_Click()

Unload Me

LoginForm.Show

End Sub

Private Sub DayTB_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 Then

If Not (IsNumeric(DayTB.Text)) Then

MsgBox "Invalid date entering", vbOKOnly, "Caution"

DayTB.Text = Str(DayUD.Value)

Exit Sub

End If

If Val(DayTB.Text) <= DayUD.Max And Val(DayTB.Text) >= 1 And Val(DayTB.Text)

/ 1 = Int(Val(DayTB.Text)) Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DayUD.Value = Val(DayTB.Text)
Else
    MsgBox "Invalid date entering", vbOKOnly, "Caution"
    DayTB.Text = Str(DayUD.Value)
End If
End If
End Sub

```

```

Private Sub DayTB_LostFocus()
    Call DayTB_KeyPress(13)
End Sub

```

```

Public Sub DayUD_Change()
    DayTB.Text = DayUD.Value
    TempDate = DateSerial(CYear, MonthUD.Value, DayUD.Value)
    Select Case Weekday(TempDate)
    Case 1
        WDL.Caption = "Sun"
    Case 2
        WDL.Caption = "Mon"
    Case 3
        WDL.Caption = "Tues"
    Case 4
        WDL.Caption = "Wed"
    Case 5
        WDL.Caption = "Thu"
    Case 6
        WDL.Caption = "Fri"
    Case 7
        WDL.Caption = "Sat"
    End Select

```

End Select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AllCon = MCon + DayUD.Value
```

```
MTB.Text = MArray(AllCon)
```

```
ATB.Text = AArray(AllCon)
```

```
ETB.Text = EArray(AllCon)
```

```
End Sub
```

```
Private Sub DelAllB_Click()
```

```
    If MsgBox("All your data will be lost. Are you sure to delete all your data ?", vbYesNo, "Warning") = vbYes Then
```

```
        FilePos = FreeFile
```

```
        Open "c:\UserDir" + UserName + ".usr" For Random As #FilePos Len = 1098
```

```
        For iCounter = 1 To 1098
```

```
            Put #FilePos, iCounter, ""
```

```
        Next iCounter
```

```
        Close #FilePos
```

```
        For iCounter = 1 To 366
```

```
            MArray(iCounter) = ""
```

```
            AArray(iCounter) = ""
```

```
            EArray(iCounter) = ""
```

```
        Next iCounter
```

```
        MTB.Text = ""
```

```
        ATB.Text = ""
```

```
        ETB.Text = ""
```

```
        MsgBox "All data deleted", vbOKOnly, "Delete data"
```

```
    End If
```

```
End Sub
```

```
Private Sub ETB_Change()
```

```
    EArray(AllCon) = ETB.Text
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Form_Load()
    FilePos = FreeFile
    Open "c:\UserDir\" + UserName + ".usr" For Random As #FilePos Len = 1098
    For iCounter = 1 To 366
        Get #FilePos, (iCounter - 1) * 3 + 1, MArray(iCounter)
        Get #FilePos, (iCounter - 1) * 3 + 2, AArray(iCounter)
        Get #FilePos, (iCounter - 1) * 3 + 3, EArray(iCounter)
    Next iCounter
    Close #FilePos
    MonthUD.Value = Month(Now)
    DayUD.Value = Day(Now)
    YearTB.Text = CYear
    'On Error GoTo handleerror:
    With MSComm1
        'Set bitrate = 9600
        .Settings = "9600,n,8,1"
        'Set number of byte to read=1.
        '.InputLen = 1
        'Set port1 to open.
        .PortOpen = True
        'Type of data is text.
        .InputMode = comInputModeText
        'To detect when data has arrived at a port1.
        '.RThreshold = 1
        'To set the number of characters in the transmit buffer that will trigger an OnComm
event.
        '.SThreshold = 1
        'Ask for a start position.
    End With
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    If MsgBox("Do you want to save your data before exit program or change user ?",  
vbYesNo, "Question") = vbYes Then
```

```
        Call SaveB_Click
```

```
    End If
```

```
End Sub
```

```
Private Sub Intface_Click()
```

```
Dim strdata As Variant
```

```
Dim temp1 As String
```

```
Dim temp2 As String
```

```
Dim temp3 As String
```

```
Dim temp4 As String
```

```
Dim temp5 As String
```

```
Dim temp6 As String
```

```
Dim i As Integer
```

```
temp1 = Mid(NameTB.Text & " ", 1, 16)
```

```
temp2 = Mid(DayTB.Text & "/" & MonthTB.Text & "/" & YearTB.Text & ":" & WDL.Caption & " ", 1, 16)
```

```
temp3 = Mid(MTB.Text & " ", 1, 16)
```

```
temp4 = Mid(ATB.Text & " ", 1, 16)
```

```
temp5 = Mid(ETB.Text & " ", 1, 16)
```

```
strdata = "@" & temp1 & temp2 & temp3 & temp4 & temp5 & "#"
```

```
MSComm1.Output = strdata
```

```
For i = 1 To 10000
```

```
    DoEvents
```

```
Next i
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp6 = MSComm1.Input
```

```
Text1.Text = temp6
```

```
End Sub
```

```
Private Sub MonthTB_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 13 Then
```

```
        If MonthTB.Text = "Jan" Or MonthTB.Text = "jan" Or MonthTB.Text = "JAN" Or  
MonthTB.Text = "1" Then
```

```
            MonthTB.Text = "Jan"
```

```
            MonthUD.Value = 1
```

```
        ElseIf MonthTB.Text = "Feb" Or MonthTB.Text = "feb" Or MonthTB.Text = "FEB" Or  
MonthTB.Text = "2" Then
```

```
            MonthTB.Text = "Feb"
```

```
            MonthUD.Value = 2
```

```
        ElseIf MonthTB.Text = "Mar" Or MonthTB.Text = "mar" Or MonthTB.Text = "MAR" Or  
MonthTB.Text = "3" Then
```

```
            MonthTB.Text = "Mar"
```

```
            MonthUD.Value = 3
```

```
        ElseIf MonthTB.Text = "Apr" Or MonthTB.Text = "apr" Or MonthTB.Text = "APR" Or  
MonthTB.Text = "4" Then
```

```
            MonthTB.Text = "Apr"
```

```
            MonthUD.Value = 4
```

```
        ElseIf MonthTB.Text = "May" Or MonthTB.Text = "may" Or MonthTB.Text = "MAY" Or  
MonthTB.Text = "5" Then
```

```
            MonthTB.Text = "May"
```

```
            MonthUD.Value = 5
```

```
        ElseIf MonthTB.Text = "Jun" Or MonthTB.Text = "jun" Or MonthTB.Text = "JUN" Or
```

```
MonthTB.Text = "6" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MonthTB.Text = "Jun"
MonthUD.Value = 6
Elseif MonthTB.Text = "Jul" Or MonthTB.Text = "jul" Or MonthTB.Text = "JUL" Or
MonthTB.Text = "7" Then
MonthTB.Text = "Jul"
MonthUD.Value = 7
Elseif MonthTB.Text = "Aug" Or MonthTB.Text = "aug" Or MonthTB.Text = "AUG" Or
MonthTB.Text = "8" Then
MonthTB.Text = "Aug"
MonthUD.Value = 8
Elseif MonthTB.Text = "Sep" Or MonthTB.Text = "sep" Or MonthTB.Text = "SEP" Or
MonthTB.Text = "9" Then
MonthTB.Text = "Sep"
MonthUD.Value = 9
Elseif MonthTB.Text = "Oct" Or MonthTB.Text = "oct" Or MonthTB.Text = "OCT" Or
MonthTB.Text = "10" Then
MonthTB.Text = "Oct"
MonthUD.Value = 10
Elseif MonthTB.Text = "Nov" Or MonthTB.Text = "nov" Or MonthTB.Text = "NOV" Or
MonthTB.Text = "11" Then
MonthTB.Text = "Nov"
MonthUD.Value = 11
Elseif MonthTB.Text = "Dec" Or MonthTB.Text = "dec" Or MonthTB.Text = "DEC" Or
MonthTB.Text = "12" Then
MonthTB.Text = "Dec"
MonthUD.Value = 12
Else
MsgBox "Invalid month entering", vbOKOnly, "Caution"
Call MonthUD_Change
End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub MonthTB_LostFocus()

 Call MonthTB_KeyPress(13)

End Sub

Private Sub MonthUD_Change()

 Select Case MonthUD.Value

 Case 1

 MonthTB.Text = "Jan"

 MCon = 0

 DayUD.Max = 31

 Case 2

 MonthTB.Text = "Feb"

 MCon = 31

 If CYear Mod 4 = 0 Then

 DayUD.Max = 29

 Else

 DayUD.Max = 28

 End If

 Case 3

 MonthTB.Text = "Mar"

 MCon = 60

 DayUD.Max = 31

 Case 4

 MonthTB.Text = "Apr"

 MCon = 91

 DayUD.Max = 30

 Case 5

 MonthTB.Text = "May"

 MCon = 121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub MSComm1_OnComm()  
    Select Case MSComm1.CommEvent  
        Case 2:  
            Text1.Text = MSComm1.Input  
    End Select  
    'Text1.Text = MSComm1.Input  
End Sub
```

```
Private Sub MTB_Change()  
    MArray(AllCon) = MTB.Text  
End Sub
```

```
Private Sub SaveB_Click()  
    FilePos = FreeFile  
    Open "c:\UserDir\" + UsrName + ".usr" For Random As #FilePos Len = 1098  
    For iCounter = 1 To 366  
        Put #FilePos, (iCounter - 1) * 3 + 1, MArray(iCounter)  
        Put #FilePos, (iCounter - 1) * 3 + 2, AArray(iCounter)  
        Put #FilePos, (iCounter - 1) * 3 + 3, EArray(iCounter)  
    Next iCounter  
    Close #FilePos  
    MsgBox "Data saved", vbOKOnly, "Save Data"  
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

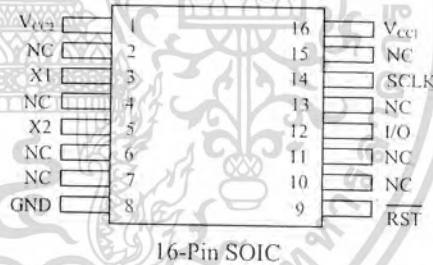
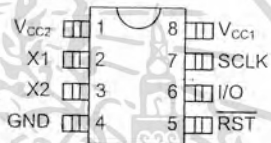
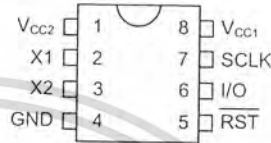


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 31 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.0–5.5V full operation
- Uses less than 300 nA at 2.0V
- Single-byte or multiple-byte (burst mode) data transfer for read or write of clock or RAM data
- 8-pin DIP or optional 8-pin SOICs for surface mount
- Simple 3-wire interface
- TTL-compatible ($V_{CC} = 5V$)
- Optional industrial temperature range $-40^{\circ}C$ to $+85^{\circ}C$
- DS1202 compatible
- Recognized by Underwriters Laboratory

PIN ASSIGNMENT



ORDERING INFORMATION

PART #	DESCRIPTION
DS1302	8-Pin DIP
DS1302N	8-Pin DIP (Industrial)
DS1302S	8-Pin SOIC (200 mil)
DS1302SN	8-Pin SOIC (Industrial)
DS1302Z	8-Pin SOIC (150 mil)
DS1302ZN	8-Pin SOIC (Industrial)
DS1302S-16	16-Pin SOIC (300 mil)
DS1302SN-16	16-Pin SOIC (Industrial)

PIN DESCRIPTION

X1, X2	– 32.768 kHz Crystal Pins
GND	– Ground
RST	– Reset
I/O	– Data Input/Output
SCLK	– Serial Clock
V _{CC1} , V _{CC2}	– Power Supply Pins

DESCRIPTION

The DS1302 Trickle Charge Timekeeping Chip contains a real time clock/calendar and 31 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

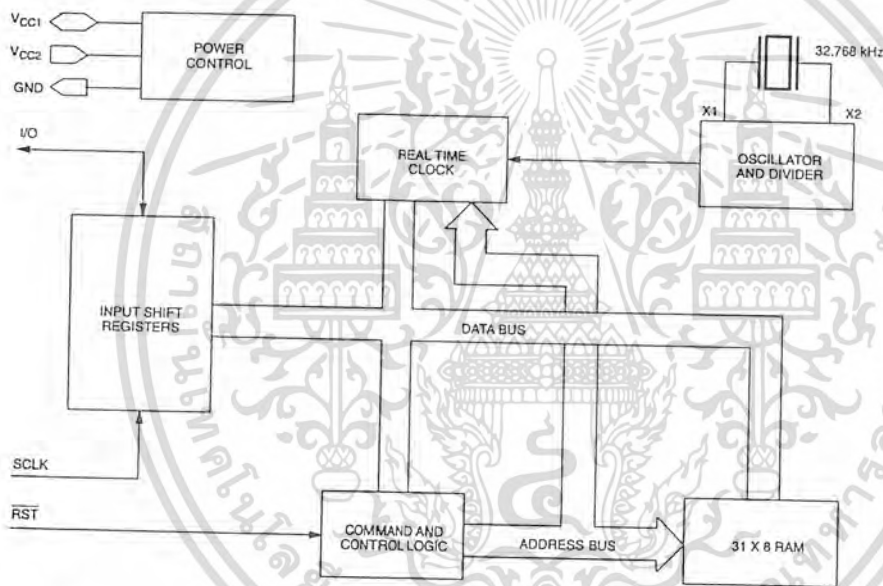
Interfacing the DS1302 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: (1) $\overline{\text{RST}}$ (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/RAM 1 byte at a time or in a burst of up to 31 bytes. The DS1302 is designed to operate on very low power and retain data and clock information on less than 1 microwatt.

The DS1302 is the successor to the DS1202. In addition to the basic timekeeping functions of the DS1202, the DS1302 has the additional features of dual power pins for primary and back-up power supplies, programmable trickle charger for V_{CC1} , and seven additional bytes of scratchpad memory.

OPERATION

The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM.

DS1302 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC1} – V_{CC1} provides low power operation in single supply and battery operated systems as well as low power battery backup. In systems using the trickle charger, the rechargeable energy source is connected to this pin.

V_{CC2} – V_{CC2} is the primary power supply pin in a dual supply configuration. V_{CC1} is connected to a backup source to maintain the time and date in the absence of primary power.

The DS1302 will operate from the larger of V_{CC1} or V_{CC2} . When V_{CC2} is greater than $V_{CC1} + 0.2V$, V_{CC2} will power the DS1302. When V_{CC2} is less than V_{CC1} , V_{CC1} will power the DS1302.

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

I/O (Data Input/Output) – The I/O pin is the bi-directional data pin for the 3-wire interface.

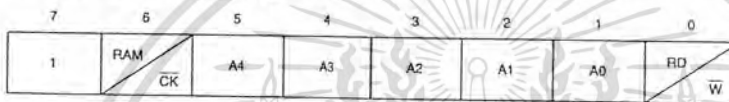
$\overline{\text{RST}}$ (Reset) – The reset signal must be asserted high during a read or a write.

X1, X2 – Connections for a standard 32.768 kHz quartz crystal. The internal oscillator is designed for operation with a crystal having a specified load capacitance of 6 pF. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, “Crystal Considerations with Dallas Real Time Clocks.” The DS1302 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

COMMAND BYTE

The command byte is shown in Figure 2. Each data transfer is initiated by a command byte. The MSB (Bit 7) must be a logic 1. If it is 0, writes to the DS1302 will be disabled. Bit 6 specifies clock/calendar data if logic 0 or RAM data if logic 1. Bits 1 through 5 specify the designated registers to be input or output, and the LSB (bit 0) specifies a write operation (input) if logic 0 or read operation (output) if logic 1. The command byte is always input starting with the LSB (bit 0).

ADDRESS/COMMAND BYTE Figure 2



RESET AND CLOCK CONTROL

All data transfers are initiated by driving the \overline{RST} input high. The \overline{RST} input serves two functions. First, \overline{RST} turns on the control logic which allows access to the shift register for the address/command sequence. Second, the \overline{RST} signal provides a method of terminating either single byte or multiple byte data transfer.

A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on the falling edge of clock. If the \overline{RST} input is low all data transfer terminates and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3. At power-up, \overline{RST} must be a logic 0 until $V_{CC} > 2.0$ volts. Also SCLK must be at a logic 0 when \overline{RST} is driven to a logic 1 state.

DATA INPUT

Following the eight SCLK cycles that input a write command byte, a data byte is input on the rising edge of the next eight SCLK cycles. Additional SCLK cycles are ignored should they inadvertently occur. Data is input starting with bit 0.

DATA OUTPUT

Following the eight SCLK cycles that input a read command byte, a data byte is output on the falling edge of the next eight SCLK cycles. Note that the first data bit to be transmitted occurs on the first falling edge after the last bit of the command byte is written. Additional SCLK cycles retransmit the data bytes should they inadvertently occur so long as \overline{RST} remains high. This operation permits continuous burst mode read capability. Also, the I/O pin is tri-stated upon each rising edge of SCLK. Data is output starting with bit 0.

BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits 1 through 5 = logic 1). As before, bit 6 specifies clock or RAM and bit 0

specifies read or write. There is no data storage capacity at locations 9 through 31 in the Clock/Calendar Registers or location 31 in the RAM registers. Reads or writes in burst mode start with bit 0 of address 0.

When writing to the clock registers in the burst mode, the first eight registers must be written in order for the data to be transferred. However, when writing to RAM in burst mode it is not necessary to write all 31 bytes for the data to transfer. Each byte that is written to will be transferred to RAM regardless of whether all 31 bytes are written or not.

CLOCK/CALENDAR

The clock/calendar is contained in seven write/read registers as shown in Figure 4. Data contained in the clock/ calendar registers is in binary coded decimal format (BCD).

CLOCK HALT FLAG

Bit 7 of the seconds register is defined as the clock halt flag. When this bit is set to logic 1, the clock oscillator is stopped and the DS1302 is placed into a low-power standby mode with a current drain of less than 100 nanoamps. When this bit is written to logic 0, the clock will start. The initial power on state is not defined.

AM-PM/12-24 MODE

Bit 7 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 – 23 hours).

WRITE PROTECT BIT

Bit 7 of the control register is the write-protect bit. The first seven bits (bits 0 – 6) are forced to 0 and will always read a 0 when read. Before any write operation to the clock or RAM, bit 7 must be 0. When high, the write protect bit prevents a write operation to any other register. The initial power on state is not defined. Therefore the WP bit should be cleared before attempting to write to the device.

TRICKLE CHARGE REGISTER

This register controls the trickle charge characteristics of the DS1302. The simplified schematic of Figure 5 shows the basic components of the trickle charger. The trickle charge select (TCS) bits (bits 4 -7) control the selection of the trickle charger. In order to prevent accidental enabling, only a pattern of 1010 will enable the trickle charger. All other patterns will disable the trickle charger. The DS1302 powers up with the trickle charger disabled. The diode select (DS) bits (bits 2 – 3) select whether one diode or two diodes are connected between V_{CC2} and V_{CC1} . If DS is 01, one diode is selected or if DS is 10, two diodes are selected. If DS is 00 or 11, the trickle charger is disabled independently of TCS. The RS bits (bits 0 -1) select the resistor that is connected between V_{CC2} and V_{CC1} . The resistor selected by the resistor select (RS) bits is as follows:

RS Bits	Resistor	Typical Value
00	None	None
01	R1	2 k Ω
10	R2	4 k Ω
11	R3	8 k Ω

If RS is 00, the trickle charger is disabled independently of TCS.

Diode and resistor selection is determined by the user according to the maximum current desired for battery or super cap charging. The maximum charging current can be calculated as illustrated in the following example. Assume that a system power supply of 5 volt is applied to V_{CC2} and a super cap is connected to V_{CC1} . Also assume that the trickle charger has been enabled with one diode and resistor R1 between V_{CC2} and V_{CC1} . The maximum current I_{max} would therefore be calculated as follows:

$$\begin{aligned} I_{max} &= (5.0V - \text{diode drop}) / R1 \\ &\sim (5.0V - 0.7V) / 2 \text{ k}\Omega \\ &\sim 2.2 \text{ mA} \end{aligned}$$

Obviously, as the super cap charges, the voltage drop between V_{CC2} and V_{CC1} will decrease and therefore the charge current will decrease.

CLOCK/CALENDAR BURST MODE

The clock/calendar command byte specifies burst mode operation. In this mode the first eight clock/calendar registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

If the write protect bit is set high when a write clock/calendar burst mode is specified, no data transfer will occur to any of the eight clock/calendar registers (this includes the control register). The trickle charger is not accessible in burst mode.

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

RAM

The static RAM is 31 x 8 bytes addressed consecutively in the RAM address space.

RAM BURST MODE

The RAM command byte specifies burst mode operation. In this mode, the 31 RAM registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

REGISTER SUMMARY

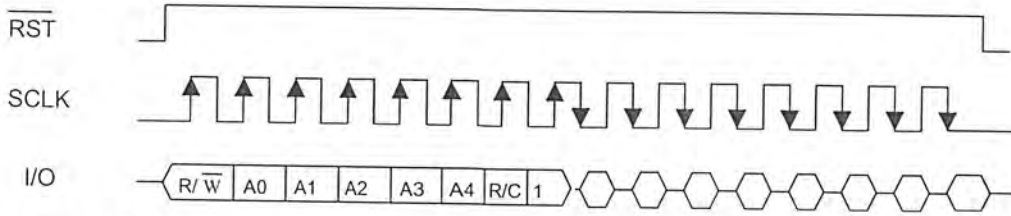
A register data format summary is shown in Figure 4.

CRYSTAL SELECTION

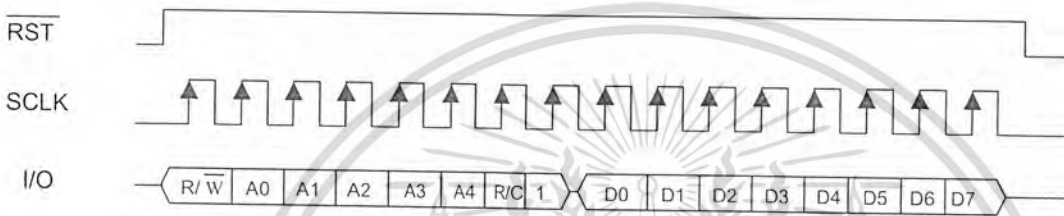
A 32.768 kHz crystal can be directly connected to the DS1302 via X1 and X2. The crystal selected for use should have a specified load capacitance (CL) of 6 pF. For more information on crystal selection and crystal layout consideration, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks."

DATA TRANSFER SUMMARY Figure 3

SINGLE BYTE READ



SINGLE BYTE WRITE



In burst mode, $\overline{\text{RST}}$ is kept high and additional SCLK cycles are sent until the end of the burst.

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

MAX220-MAX249

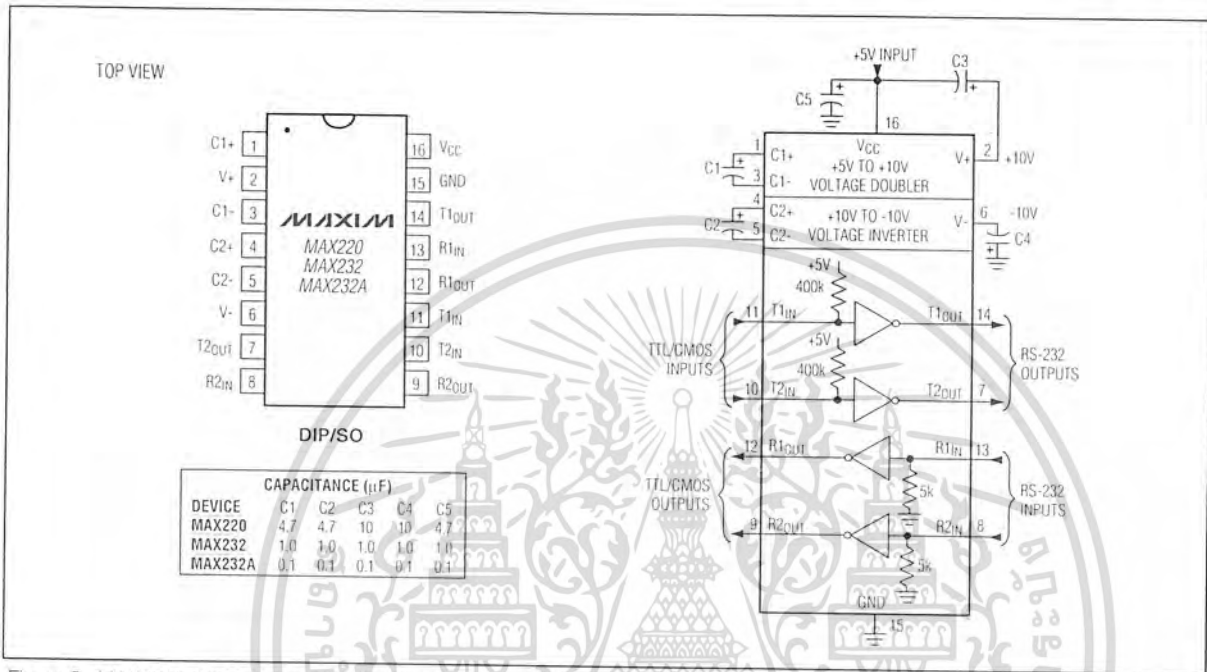


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

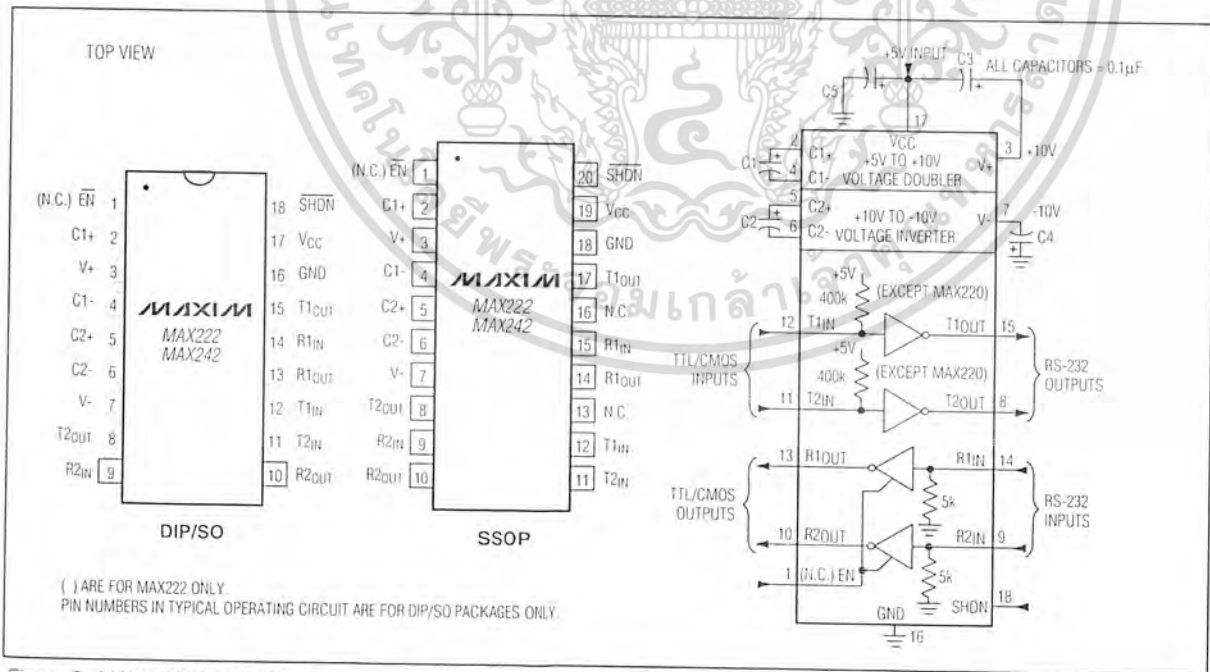


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Ordering Information (continued)

MAX220-MAX249

PART	TEMP. RANGE	PIN-PACKAGE
MAX222CPN	0°C to +70°C	18 Plastic DIP
MAX222CWN	0°C to +70°C	18 Wide SO
MAX222C/D	0°C to +70°C	Dice*
MAX222EPN	-40°C to +85°C	18 Plastic DIP
MAX222EWN	-40°C to +85°C	18 Wide SO
MAX222EJN	-40°C to +85°C	18 CERDIP
MAX222MJN	-55°C to +125°C	18 CERDIP
MAX223CAI	0°C to +70°C	28 SSOP
MAX223CWI	0°C to +70°C	28 Wide SO
MAX223C/D	0°C to +70°C	Dice*
MAX223EAI	-40°C to +85°C	28 SSOP
MAX223EWI	-40°C to +85°C	28 Wide SO
MAX225CWI	0°C to +70°C	28 Wide SO
MAX225EWI	-40°C to +85°C	28 Wide SO
MAX230CPP	0°C to +70°C	20 Plastic DIP
MAX230CWP	0°C to +70°C	20 Wide SO
MAX230C/D	0°C to +70°C	Dice*
MAX230EPP	-40°C to +85°C	20 Plastic DIP
MAX230EWP	-40°C to +85°C	20 Wide SO
MAX230EJP	-40°C to +85°C	20 CERDIP
MAX230MJP	-55°C to +125°C	20 CERDIP
MAX231CPD	0°C to +70°C	14 Plastic DIP
MAX231CWE	0°C to +70°C	16 Wide SO
MAX231CJD	0°C to +70°C	14 CERDIP
MAX231C/D	0°C to +70°C	Dice*
MAX231EPD	-40°C to +85°C	14 Plastic DIP
MAX231EWE	-40°C to +85°C	16 Wide SO
MAX231EJD	-40°C to +85°C	14 CERDIP
MAX231MJD	-55°C to +125°C	14 CERDIP
MAX232CPE	0°C to +70°C	16 Plastic DIP
MAX232CSE	0°C to +70°C	16 Narrow SO
MAX232CWE	0°C to +70°C	16 Wide SO
MAX232C/D	0°C to +70°C	Dice*
MAX232EPE	-40°C to +85°C	16 Plastic DIP
MAX232ESE	-40°C to +85°C	16 Narrow SO
MAX232EWE	-40°C to +85°C	16 Wide SO
MAX232EJE	-40°C to +85°C	16 CERDIP
MAX232MJE	-55°C to +125°C	16 CERDIP
MAX232MLP	-55°C to +125°C	20 LCC
MAX232ACPE	0°C to +70°C	16 Plastic DIP
MAX232ACSE	0°C to +70°C	16 Narrow SO
MAX232ACWE	0°C to +70°C	16 Wide SO

MAX232AC/D	0°C to +70°C	Dice*
MAX232AEPE	-40°C to +85°C	16 Plastic DIP
MAX232AESE	-40°C to +85°C	16 Narrow SO
MAX232AEWE	-40°C to +85°C	16 Wide SO
MAX232AEJE	-40°C to +85°C	16 CERDIP
MAX232AMJE	-55°C to +125°C	16 CERDIP
MAX232AML	-55°C to +125°C	20 LCC
MAX233CPP	0°C to +70°C	20 Plastic DIP
MAX233EPP	-40°C to +85°C	20 Plastic DIP
MAX233ACPP	0°C to +70°C	20 Plastic DIP
MAX233ACWP	0°C to +70°C	20 Wide SO
MAX233AEPP	-40°C to +85°C	20 Plastic DIP
MAX233AEWP	-40°C to +85°C	20 Wide SO
MAX234CPE	0°C to +70°C	16 Plastic DIP
MAX234CWE	0°C to +70°C	16 Wide SO
MAX234C/D	0°C to +70°C	Dice*
MAX234EPE	-40°C to +85°C	16 Plastic DIP
MAX234EWE	-40°C to +85°C	16 Wide SO
MAX234EJE	-40°C to +85°C	16 CERDIP
MAX234MJE	-55°C to +125°C	16 CERDIP
MAX235CPG	0°C to +70°C	24 Wide Plastic DIP
MAX235EPG	-40°C to +85°C	24 Wide Plastic DIP
MAX235EDG	-40°C to +85°C	24 Ceramic SB
MAX235MDG	-55°C to +125°C	24 Ceramic SB
MAX236CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX236CWG	0°C to +70°C	24 Wide SO
MAX236C/D	0°C to +70°C	Dice*
MAX236ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX236EWG	-40°C to +85°C	24 Wide SO
MAX236ERG	-40°C to +85°C	24 Narrow CERDIP
MAX236MRG	-55°C to +125°C	24 Narrow CERDIP
MAX237CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX237CWG	0°C to +70°C	24 Wide SO
MAX237C/D	0°C to +70°C	Dice*
MAX237ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX237EWG	-40°C to +85°C	24 Wide SO
MAX237ERG	-40°C to +85°C	24 Narrow CERDIP
MAX237MRG	-55°C to +125°C	24 Narrow CERDIP
MAX238CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX238CWG	0°C to +70°C	24 Wide SO
MAX238C/D	0°C to +70°C	Dice*
MAX238ENG	-40°C to +85°C	24 Narrow Plastic DIP

* Contact factory for dice specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX238EWG	-40°C to +85°C	24 Wide SO
MAX238ERG	-40°C to +85°C	24 Narrow CERDIP
MAX238MRG	-55°C to +125°C	24 Narrow CERDIP
MAX239CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX239CWG	0°C to +70°C	24 Wide SO
MAX239C/D	0°C to +70°C	Dice*
MAX239ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX239EWG	-40°C to +85°C	24 Wide SO
MAX239ERG	-40°C to +85°C	24 Narrow CERDIP
MAX239MRG	-55°C to +125°C	24 Narrow CERDIP
MAX240CMH	0°C to +70°C	44 Plastic FP
MAX240C/D	0°C to +70°C	Dice*
MAX241CAI	0°C to +70°C	28 SSOP
MAX241CWI	0°C to +70°C	28 Wide SO
MAX241C/D	0°C to +70°C	Dice*
MAX241EAI	-40°C to +85°C	28 SSOP
MAX241EWI	-40°C to +85°C	28 Wide SO
MAX242CAP	0°C to +70°C	20 SSOP
MAX242CPN	0°C to +70°C	18 Plastic DIP
MAX242CWN	0°C to +70°C	18 Wide SO
MAX242C/D	0°C to +70°C	Dice*
MAX242EPN	-40°C to +85°C	18 Plastic DIP
MAX242EWN	-40°C to +85°C	18 Wide SO
MAX242EJN	-40°C to +85°C	18 CERDIP
MAX242MJN	-55°C to +125°C	18 CERDIP

MAX243CPE	0°C to +70°C	16 Plastic DIP
MAX243CSE	0°C to +70°C	16 Narrow SO
MAX243CWE	0°C to +70°C	16 Wide SO
MAX243C/D	0°C to +70°C	Dice*
MAX243EPE	-40°C to +85°C	16 Plastic DIP
MAX243ESE	-40°C to +85°C	16 Narrow SO
MAX243EWE	-40°C to +85°C	16 Wide SO
MAX243EJE	-40°C to +85°C	16 CERDIP
MAX243MJE	-55°C to +125°C	16 CERDIP
MAX244CQH	0°C to +70°C	44 PLCC
MAX244C/D	0°C to +70°C	Dice*
MAX244EQH	-40°C to +85°C	44 PLCC
MAX245CPL	0°C to +70°C	40 Plastic DIP
MAX245C/D	0°C to +70°C	Dice*
MAX245EPL	-40°C to +85°C	40 Plastic DIP
MAX246CPL	0°C to +70°C	40 Plastic DIP
MAX246C/D	0°C to +70°C	Dice*
MAX246EPL	-40°C to +85°C	40 Plastic DIP
MAX247CPL	0°C to +70°C	40 Plastic DIP
MAX247C/D	0°C to +70°C	Dice*
MAX247EPL	-40°C to +85°C	40 Plastic DIP
MAX248CQH	0°C to +70°C	44 PLCC
MAX248C/D	0°C to +70°C	Dice*
MAX248EQH	-40°C to +85°C	44 PLCC
MAX249CQH	0°C to +70°C	44 PLCC
MAX249EQH	-40°C to +85°C	44 PLCC

* Contact factory for dice specifications.

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

36 _____ Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 (408) 737-7600

© 2000 Maxim Integrated Products Printed USA **MAXIM** is a registered trademark of Maxim Integrated Products.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ธีรวัฒน์ ประกอบผล , " การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ " , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) , พ.ศ. 2542
2. สุนทร วิฑูรพจน์ , " การใช้งานไมโครคอนโทรลเลอร์ 8051 " , บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , พ.ศ. 2537
3. วันสุระ ศรีใจดี , " ประยุกต์/อินเทอร์เฟส ไมโครคอนโทรลเลอร์ ภาคปฏิบัติ " , บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , พ.ศ. 2542
4. แผนกหนังสือพิเศษด้านอิเล็กทรอนิกส์ , " Semiconductor Electronic Journal " , บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , พ.ศ. 2535
5. แผนกหนังสือพิเศษด้านอิเล็กทรอนิกส์ , " Semiconductor Electronic Journal " , บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , พ.ศ. 2538
6. แผนกหนังสือพิเศษด้านอิเล็กทรอนิกส์ , " Semiconductor Electronic Journal " , บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , พ.ศ. 2540
7. กฤษฎา ใจเย็น , " เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม " , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , พ.ศ. 2542
8. " Dot Matrix LCD Module " , ETT CO., LTD. , 1997
9. ธาริน สิทธิธรรมชารี , " คู่มือการเขียนโปรแกรม Microsoft Visual Basic Version 6.0 ฉบับเพื่อการใช้งานจริง " , บริษัท ชัคเซลมี่เดีย จำกัด , พ.ศ. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้