

เครื่องรวบรวมข้อมูลอัตโนมัติ

Automatic Data Logger



เลขหนังสือ.....  
เลขทะเบียน..... 42719  
วัน, เดือน, ปี 7 ส.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรวบรวมข้อมูลอัตโนมัติ

Automatic Data Logger



ปริญญาโท

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ปริญญาโทปีการศึกษา 2543

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องรวบรวมข้อมูลอัตโนมัติ

ผู้จัดทำ นายวิฑูรย์ วิทยชำนาญกุล รหัส 40010733

 อาจารย์ที่ปรึกษา  
(อาจารย์ขนิษฐา แซ่ตั้ง)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เครื่องรวบรวมข้อมูลอัตโนมัติ

นาย วิทวัส วิทย์ชำนาญกุล  
อาจารย์ที่ปรึกษา  
อาจารย์ ขนิษฐา แซ่ตั้ง  
ปีการศึกษา 2543

### บทคัดย่อ

เครื่องรวบรวมข้อมูลอัตโนมัติขนาด 4 ช่องสัญญาณนี้ ประยุกต์ใช้ไมโครคอนโทรลเลอร์ PIC16F877 ซึ่งเป็นไมโครคอนโทรลเลอร์แบบ RISC (Reduce Instruction Set Computer) เพื่อควบคุมการทำงานทั้งหมด โดยมีขั้นตอนการทำงานที่สำคัญดังนี้ ตัวตรวจจับ (Sensor) จะทำหน้าที่เปลี่ยนปริมาณทางกายภาพใดๆ ให้เป็นระดับแรงดันไฟฟ้าที่เหมาะสม เข้าสู่วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลขนาด 8 บิต ที่อยู่ในไมโครคอนโทรลเลอร์ ข้อมูลที่ได้จะถูกเก็บลงหน่วยความจำแบบ EEPROM ตามเวลาที่กำหนดไว้ในอุปกรณ์คำนวณค่าเวลาจริง (Real Time Clock) จากนั้นเมื่อต่อเครื่องเข้ากับเครื่องคอมพิวเตอร์ผ่านพอร์ตอนุกรมแบบ RS-232C แล้วเรียกโปรแกรม ข้อมูลทั้งหมดจะถูกถ่ายโอนเข้าสู่เครื่องคอมพิวเตอร์ ผู้ใช้สามารถนำข้อมูลที่นำไปใช้งานด้านสถิติ หรือ ใช้ในการวิเคราะห์ วิจัยต่อไป ทั้งนี้ข้อมูลทั้งหมดจะไม่สูญหายแม้ไม่มีไฟเลี้ยงและไม่ต้องใช้แหล่งจ่ายไฟสำรองเพิ่มเติม จากการทดลองพบว่าเครื่องสามารถทำงานได้อย่างถูกต้องตามที่ได้ออกแบบไว้

## Automatic Data Logger

Withawat Withayachumanankul

Advisor

MS. Kanitta Saetung

2000

### Abstract

This project implements PIC16F877, which is RISC-base architecture microcontroller, to control a 4-channel data logger. Whole task of data logger can describe as; the sensor transforms any physical quantity to electrical signal. Embed analog-to-digital converter circuit in microcontroller converts the received analog signal to 8-bit resolution digital signal. Converted data then stored in EEPROM memory. Launch time and time interval of data-logger held by real-time clock. When connect data logger to computer via RS-232C serial port, and run data-logger's program, all stored data then uploaded onto computer for further statistic or analysis purpose. This equipment doesn't need backup-power to maintain stored data in case of power failure. After a long running, we can conclude that the data logger reaches the intention exactly.

## สารบัญ

บทที่ 1	บทนำ	1
บทที่ 2	ไมโครคอนโทรลเลอร์ PIC16F877	2
2.1	คุณสมบัติสำคัญของไมโครคอนโทรลเลอร์ PIC16F877	2
2.2	หน้าที่ของขาต่างๆ สำหรับ PIC16F877	5
2.3	การจัดการหน่วยความจำ	6
2.3.1	หน่วยความจำโปรแกรม	6
2.3.2	โปรแกรมเคาน์เตอร์ (Program counter: PC) และแอสตค (Stack)	7
2.3.3	หน่วยความจำข้อมูล (รีจิสเตอร์)	8
2.3.4	การอ้างอิงแอดเดรสของรีจิสเตอร์โดยอ้อม	11
2.4	ความสามารถทั่วไปของไมโครคอนโทรลเลอร์	11
2.4.1	การอินเตอร์รัพท์	11
2.4.2	การใช้งานออสซิลเลเตอร์	13
2.4.3	โหมดประหยัดพลังงาน (Sleep mode)	14
2.4.4	การโปรแกรมไมโครคอนโทรลเลอร์ในระบบ	14
2.5	ส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล	15
2.5.1	รีจิสเตอร์ที่เกี่ยวข้องกับการแปลงสัญญาณ	15
2.5.2	ข้อกำหนดในการใช้งาน	17
บทที่ 3	การสื่อสารแบบอนุกรม	18
3.1	การสื่อสารอนุกรมแบบ RS-232	18
3.1.1	รูปแบบการรับส่งข้อมูลแบบอะซิงโครนัส	19
3.1.2	อัตราความเร็วในการรับส่งข้อมูลแบบอะซิงโครนัส	20
3.1.3	คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ	20
3.1.4	Universal Asynchronous Receiver Transmitter	22
3.1.5	ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232	23
3.1.6	แอดเดรสของพอร์ตอนุกรม	24
3.1.7	รีจิสเตอร์ของพอร์ตอนุกรม	24
3.2	ระบบบัส I <sup>2</sup> C	30
3.2.1	การต่อเชื่อมไอซีบนบัส	30
3.2.2	สถานะต่างๆของบัส	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3	รูปแบบการติดต่อสื่อสาร	32
3.2.4	การอ้างอิงแอดเดรส	32
บทที่ 4	การทำงานของวงจรและการออกแบบ	33
4.1	รายละเอียดการทำงานของเครื่อง	33
4.2	โครงสร้างของระบบโดยรวม	34
4.3	การทำงานของระบบ	35
4.4	วงจรและการทำงานของวงจร	37
4.3.1	ไมโครคอนโทรลเลอร์	38
4.3.2	วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล	38
4.3.3	หน่วยความจำแบบ EEPROM	38
4.3.4	อุปกรณ์คำนวณค่าเวลาจริง	38
4.3.5	ส่วนติดต่อกับพอร์ทอนุกรม RS-232	39
4.3.6	ส่วนแสดงผล	39
บทที่ 5	การทดสอบการทำงาน	40
5.1	การทดสอบการทำงานของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล	40
5.2	การทดสอบการทำงานของเครื่อง	43
บทที่ 6	สรุปผลการทำงาน	47
6.1	สรุปผลการทำงาน	47
6.2	แนวทางในการพัฒนาและประยุกต์ใช้งาน	47
ภาคผนวก ก.	เครื่องรวบรวมข้อมูลอัตโนมัติ	48
ภาคผนวก ข.	การใช้งานเครื่องรวบรวมข้อมูลอัตโนมัติ	49
ภาคผนวก ค.	ดาต้าชีทของไอซีที่สำคัญ	53
	กิตติกรรมประกาศ	59
	เอกสารอ้างอิง	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ

รูปที่ 2.1	ไมโครคอนโทรลเลอร์เบอร์ PIC16F877	2
รูปที่ 2.2	สถาปัตยกรรมภายในของ PIC16F877	4
รูปที่ 2.3	แผนผังหน่วยความจำ โปรแกรม และแสดงของ PIC16F877	7
รูปที่ 2.4	การใส่ค่าลงในโปรแกรมเคาน์เตอร์	7
รูปที่ 2.5	แผนผังหน่วยความจำข้อมูล (รีจิสเตอร์)	9
รูปที่ 2.6	การอ้างอิงแอดเดรสของรีจิสเตอร์โดยตรงและโดยอ้อม	11
รูปที่ 2.7	แสดงการต่อคริสตอลออสซิลเลเตอร์	13
รูปที่ 2.8	แสดงการต่อรีจิสเตอร์-คาปาซิเตอร์ออสซิลเลเตอร์	13
รูปที่ 2.9	การต่อไมโครคอนโทรลเลอร์เพื่อทำการโปรแกรม	14
รูปที่ 2.10	วงจรเสมือนแสดงการประจุคาปาซิเตอร์ $C_{HOLD}$	17
รูปที่ 3.1	รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส	19
รูปที่ 3.2	แสดงการจัดขาสัญญาณของพอร์ทอนุกรม และหน้าที่การทำงาน	21
รูปที่ 3.3	การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ทอนุกรมในรูปแบบต่าง ๆ	22
รูปที่ 3.4	การทำงานภายในของขาสัญญาณต่าง ๆ ของพอร์ทอนุกรม	23
รูปที่ 3.5	ตัวอย่างระบบที่ใช้บัส I <sup>2</sup> C	30
รูปที่ 3.6	การต่อไอซีบนบัส I <sup>2</sup> C	30
รูปที่ 3.7	สถานะเริ่มต้นและสถานะหยุดของบัส I <sup>2</sup> C	31
รูปที่ 3.8	สถานะส่งข้อมูลของบัส I <sup>2</sup> C	31
รูปที่ 3.9	สัญญาณสื่อสารของบัส I <sup>2</sup> C	32
รูปที่ 3.10	รูปแบบแอดเดรส 7 บิตของบัส I <sup>2</sup> C	32
รูปที่ 4.1	บล็อกไดอะแกรมแสดงโครงสร้างของระบบ	34
รูปที่ 4.2	ขั้นตอนการทำงานของระบบ	35
รูปที่ 4.3	วงจรของเครื่องรวบรวมข้อมูลอัตโนมัติ	37
รูปที่ 5.1	กราฟแสดงความสัมพันธ์ของค่าแรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลกับค่าแรงดันทดสอบ เมื่อแรงดันอ้างอิงเท่ากับ 2.50 โวลต์ และ 4.00 โวลต์	42
รูปที่ 5.2	ผลการทำงานในโหมด Stand Alone ที่แรงดันอ้างอิง 2.50 โวลต์	43
รูปที่ 5.3	ผลการทำงานในโหมด Computer Base ที่แรงดันอ้างอิง 2.50 โวลต์	44
รูปที่ 5.4	ผลการทำงานในโหมด Stand Alone ที่แรงดันอ้างอิง 4.00 โวลต์	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.5	ผลการทำงานในโหมด Computer Base ที่แรงดันอ้างอิง 4.00 โวลต์	46
รูปที่ ก.1	เครื่องรวบรวมข้อมูลอัตโนมัติ	48
รูปที่ ข.1	หน้าจอหลักขณะยังไม่ติดต่อกับเครื่อง	49
รูปที่ ข.2	หน้าจอหลักขณะติดต่อกับเครื่อง	49
รูปที่ ข.3	หน้าจอสำหรับการตั้งค่าเครื่อง	50
รูปที่ ข.4	หน้าจอแสดงข้อมูล	52

### สารบัญตาราง

ตารางที่ 2.1	แสดงการจัดขาและหน้าที่ของแต่ละขาใน PIC16F877	5
ตารางที่ 3.1	แสดงแอดเดรสฐานและหน่วยความจำที่เก็บแอดเดรสฐานของพอร์ตอนุกรม	24
ตารางที่ 5.1	แสดงค่าแรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 8 บิต เมื่อแรงดันอ้างอิงเท่ากับ 2.50 โวลต์	40
ตารางที่ 5.2	แสดงค่าแรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 8 บิต เมื่อแรงดันอ้างอิงเท่ากับ 4.00 โวลต์	41

## บทที่ 1

### บทนำ

ในปัจจุบันมีไมโครคอนโทรลเลอร์หลายรุ่น หลายตระกูล ถูกผลิตขึ้นเพื่อตอบสนองการใช้งานที่แตกต่างกัน โดยมีความแตกต่างทั้งความเร็วสัญญาณนาฬิกา ความยาวเวิร์ดข้อมูล ขนาดของหน่วยความจำ จำนวนอินพุต เอาท์พุทพอร์ต แต่โดยทั่วไปไมโครคอนโทรลเลอร์สามารถจำแนกได้เป็น 2 กลุ่มใหญ่ตามสถาปัตยกรรมที่ใช้ คือไมโครคอนโทรลเลอร์แบบ CISC (Complex Instruction Set Computer) และไมโครคอนโทรลเลอร์แบบ RISC (Reduce Instruction Set Computer) โดย CISC เช่น MCS-51 มีจุดเด่นที่มีจำนวนชุดคำสั่งมาก แต่ละคำสั่งใช้จำนวนสัญญาณนาฬิกาต่างกัน ส่วน RISC มีชุดคำสั่งน้อยกว่า โดยชุดคำสั่งถูกจัดให้มีโครงสร้างการทำงานคล้ายกัน ทำให้สามารถทำงานแบบไปป์ไลน์ได้ โดยใช้สัญญาณนาฬิกาเพียงหนึ่งลูกต่อหนึ่งคำสั่ง ดังนั้นความเร็วของ RISC จะสูงกว่า CISC ประมาณสองถึงสี่เท่า ที่สัญญาณนาฬิกาเดียวกัน แต่อาจไม่เป็นเช่นนั้นเสมอไป เนื่องจากชุดคำสั่งที่น้อยกว่าทำให้ RISC ต้องใช้คำสั่งมากกว่าหนึ่งคำสั่งเพื่อทำงานอย่างเดียวกับ CISC ที่ใช้เพียงคำสั่งเดียว นอกจากนี้ชุดคำสั่งที่ง่ายกว่าของ RISC ส่งผลให้ขนาดของวงจรรวมชิปเล็กกว่า การเพิ่มวงจรรวมพิเศษเช่น วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลจึงทำได้

ไมโครคอนโทรลเลอร์ PIC16F877 ของบริษัทไมโครชิพเทคโนโลยี (Microchip Technology Inc.) เป็นไมโครคอนโทรลเลอร์แบบ RISC มีโมดูลสำหรับจัดการสิ่งต่างๆหลายส่วน อาทิ วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล ส่วนติดต่อกับบัสแบบ I<sup>2</sup>C ส่วนติดต่อส่งข้อมูลแบบอนุกรมอะซิงโครนัส UART โดยสามารถทำงานที่สัญญาณนาฬิกาสูงสุดถึง 20 เมกกะเฮิรตซ์ ที่กระแส 7 มิลลิแอมป์

เครื่องรวบรวมข้อมูลอัตโนมัติได้ใช้ PIC16F877 ในการควบคุมการทำงานทั้งหมด โดยการออกแบบได้เน้นการใช้ประโยชน์จากโมดูลที่มีอยู่ในตัวไมโครคอนโทรลเลอร์ให้มากที่สุด ทำให้การประยุกต์ใช้งานมีความสะดวก และขนาดของวงจรรวมเล็กลงอีกด้วย

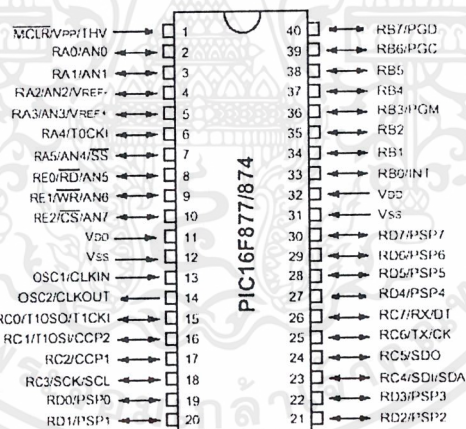
## บทที่ 2

### ไมโครคอนโทรลเลอร์ PIC16F877

ไมโครคอนโทรลเลอร์ตระกูล PIC ของบริษัทไมโครชิพเทคโนโลยี เป็นไมโครคอนโทรลเลอร์ที่ใช้สถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer) ซึ่งแตกต่างจากไมโครคอนโทรลเลอร์ตระกูล 8051 ที่เป็นสถาปัตยกรรมแบบ CISC (Complex Instruction Set Computer)

สำหรับไมโครคอนโทรลเลอร์เบอร์ PIC16F877 ออกสู่ตลาดครั้งแรกในปี ค.ศ. 1998 มีหน่วยความจำโปรแกรมแบบแฟลชขนาด 8 กิโลเวิร์ด นอกจากนี้ ยังมีไมโครคอนโทรลเลอร์ในรุ่นเดียวกันอีก 3 เบอร์ คือ PIC16F873, PIC16F874 และ PIC16F876 ซึ่งมีขนาดหน่วยความจำโปรแกรม และจำนวนพอร์ทอินพุท/เอาต์พุท แตกต่างกันไป

#### 2.1 คุณสมบัติสำคัญของไมโครคอนโทรลเลอร์ PIC16F877



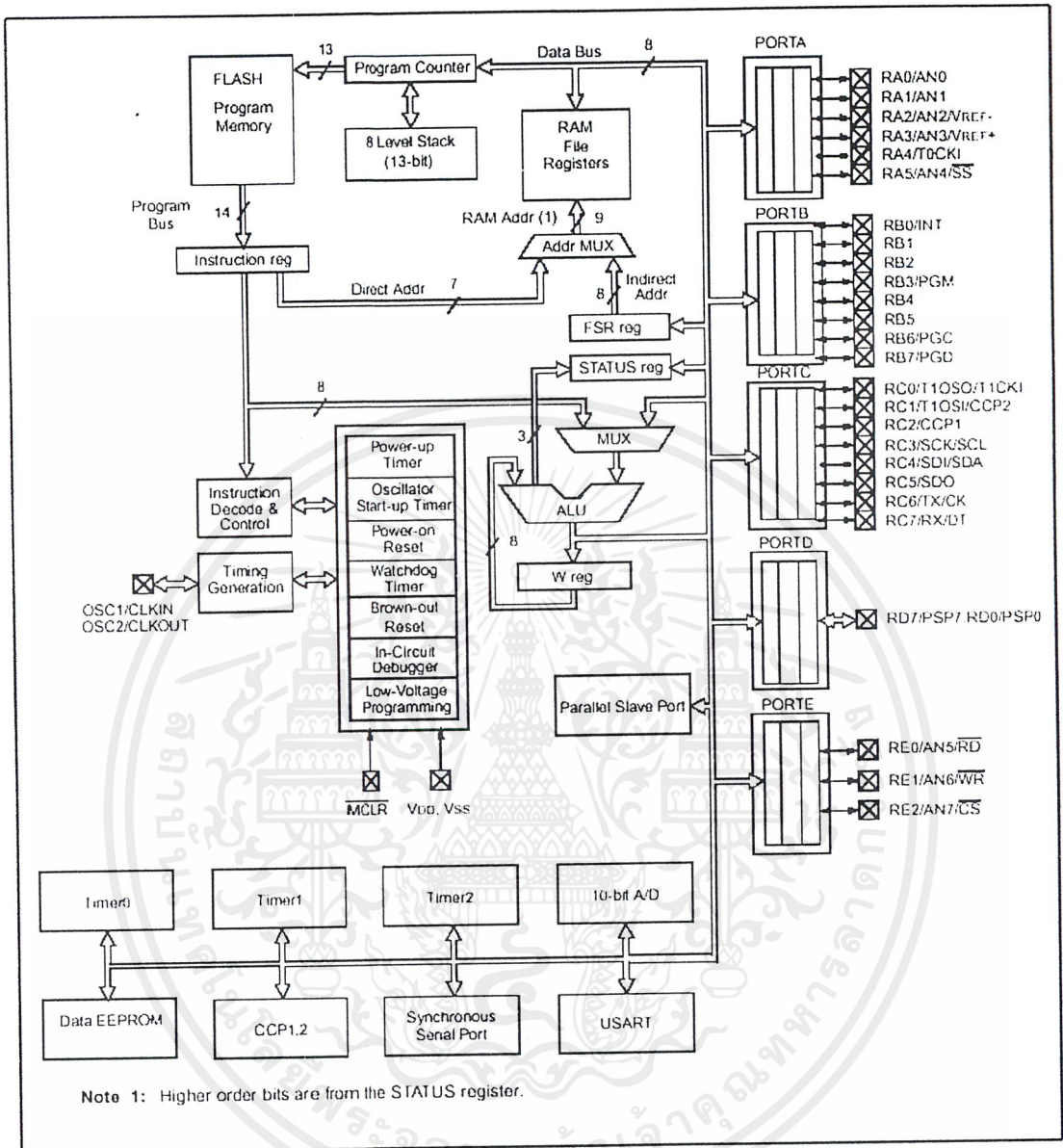
รูปที่ 2.1 ไมโครคอนโทรลเลอร์เบอร์ PIC16F877

- สถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer)
- ใช้การประมวลผลแบบไปป์ไลน์ (Pipelining)
- ทุกคำสั่งประมวลผลเสร็จภายใน 1 รอบสัญญาณนาฬิกา ยกเว้นการเรียกโปรแกรมย่อย ซึ่งใช้ 2 รอบสัญญาณนาฬิกา
- ความเร็วสัญญาณนาฬิกาสูงสุด 20 เมกะเฮิรตซ์
- บัสดำสั่งขนาด 14 บิต แยกอิสระกับบัสดัข้อมูลขนาด 8 บิต
- หน่วยความจำโปรแกรมแบบแฟลช ขนาด 8 กิโลเวิร์ด ( 1 เวิร์ดของคำสั่ง มีขนาด 14 บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำข้อมูลแบบ SRAM (รีจิสเตอร์) ขนาด 368 ไบต์
- หน่วยความจำข้อมูลแบบ EEPROM ขนาด 256 ไบต์
- สามารถอ้างอิงแอดเดรสแบบโดยตรง โดยอ้อม และแบบสัมพันธ์ .
- มีฮาร์ดแวร์แสดง 8 ระดับ
- อินเตอร์รัพท์สูงสุด 14 แหล่ง
- อินพุท/เอาต์พุทพอร์ตรวม 33 บิต (พอร์ต A, B, C, D, E)
- ตัวจับเวลา/ตัวนับขนาด 8 บิต 2 ตัว และขนาด 16 บิต 1 ตัว
- วงจรตรวจจับ, เปรียบเทียบ และ กำเนิดสัญญาณแบบ PWM 2 ตัว
- สนับสนุนการสื่อสารอนุกรมแบบ Synchronous Serial Port (SSP) , I<sup>2</sup>C (Master/Slave) และ Universal Synchronous Asynchronous Receiver Transmitter (USART)
- สนับสนุนการสื่อสารแบบขนาน ขนาด 8 บิต (Parallel Slave Port :PSP)
- วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล ความละเอียด 10 บิต 8 ช่องสัญญาณ
- สามารถโปรแกรมไมโครคอนโทรลเลอร์ที่อยู่ภายในระบบได้โดยตรง ผ่านพอร์ต 2 พอร์ต (In-Circuit Serial Programming : ICSP)
- ระบบรีเซ็ตอัตโนมัติเมื่อเริ่มจ่ายกระแสไฟฟ้าเข้าไมโครคอนโทรลเลอร์ (Power On Reset :POR)
- มีการหน่วงเวลารอไฟเลี้ยงและออสซิลเลเตอร์ให้อยู่ในสถานะเสถียร (Power-Up Timer:PWRT ,Oscillator Start Up Timer: OST)
- ใช้กระแสน้อยกว่า 2 มิลลิแอมป์ที่ 4 เมกะเฮิรตซ์ และน้อยกว่า 1 ไมโครแอมป์เมื่ออยู่ในโหมด Sleep
- สามารถป้องกันการอ่านโปรแกรมจากไอซีได้
- รอบการเขียนหน่วยความจำโปรแกรมแบบแฟลช 1000 ครั้ง รอบการเขียนหน่วยความจำข้อมูลแบบ EEPROM 10,000,000 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 สถาปัตยกรรมภายในของ PIC16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 หน้าที่ของขาต่างๆ สำหรับ PIC16F877

ตารางที่ 2.1 แสดงการจัดขา และหน้าที่ของแต่ละขาใน PIC16F877

ชื่อขา	ตำแหน่งขา	รูปแบบ	อธิบาย (เฉพาะหน้าที่ที่สำคัญ)
OSC1/CLKIN	13	I	สัญญาณออสซิลเลเตอร์ขาเข้า
OSC2/CLKOUT	14	O	สัญญาณออสซิลเลเตอร์ขาออก
MCRL/Vpp/THV	1	I/P	รีเซตเมื่อเป็นลอจิกต่ำ/แรงดันสูงเพื่อโปรแกรม IC
RA0/AN0	2	I/O	พอร์ท A เป็นพอร์ท 2ทิศทาง RA0 เป็นอนาล็อกอินพุตช่อง 0 RA1 เป็นอนาล็อกอินพุตช่อง 1 RA2 เป็นอนาล็อกอินพุตช่อง 2 หรือแรงดันอ้างอิงลบ RA3 เป็นอนาล็อกอินพุตช่อง 3 หรือแรงดันอ้างอิงบวก RA5 เป็นอนาล็อกอินพุตช่อง 4
RA1/AN1	3	I/O	
RA2/AN2/Vref-	4	I/O	
RA3/AN3/Vref+	5	I/O	
RA4/T0CKI	6	I/O	
RA5/SS/AN4	7	I/O	
RB0/INT	33	I/O	พอร์ท B เป็นพอร์ท 2ทิศทาง โปรแกรมให้มีตัวต้านทานพูลอัพได้ RB0 เป็นขารับอินเตอร์รัพท์ภายนอก RB3 เป็นขาโปรแกรมแรงดันต่ำ สร้างอินเตอร์รัพท์เมื่อขาเปลี่ยนสถานะ สร้างอินเตอร์รัพท์เมื่อขาเปลี่ยนสถานะ สร้างอินเตอร์รัพท์เมื่อขาเปลี่ยนสถานะ หรือขารับสัญญาณนาฬิกาสำหรับการ โปรแกรม IC สร้างอินเตอร์รัพท์เมื่อขาเปลี่ยนสถานะ หรือขารับสัญญาณข้อมูลสำหรับการ โปรแกรม IC
RB1	34	I/O	
RB2	35	I/O	
RB3/PGM	36	I/O	
RB4	37	I/O	
RB5	38	I/O	
RB6/PGC	39	I/O	
RB7/PGD	40	I/O	
RC0/T1OSO/T1CLK	15	I/O	พอร์ท C เป็นพอร์ท 2ทิศทาง RC3 เป็นขาสัญญาณนาฬิกาสำหรับการสื่อสารแบบ I <sup>2</sup> C RC3 เป็นขาสัญญาณข้อมูลสำหรับการสื่อสารแบบ I <sup>2</sup> C RC6 เป็น ขาส่งข้อมูลสำหรับ USART แบบอะซิงโครนัส RC7 เป็น ขารับข้อมูลสำหรับ USART แบบอะซิงโครนัส
RC1/T1OSI/CCP2	16	I/O	
RC2/CCP1	17	I/O	
RC3/SCK/SCL	18	I/O	
RC4/SDI/SDA	23	I/O	
RC5/SDO	24	I/O	
RC6/TX/CK	25	I/O	
RC7/RX/DT	26	I/O	

\* I = Input , O=Output, P=Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 2.1 (ต่อ)

ชื่อขา	ตำแหน่งขา	รูปแบบ	อธิบาย (เฉพาะหน้าที่ที่สำคัญ)	
RD0/PSP0	19	I/O	พอร์ท D เป็นพอร์ท 2ทิศทาง หรือพอร์ทขนาน	
RD1/PSP1	20	I/O		
RD2/PSP2	21	I/O		
RD3/PSP3	22	I/O		
RD4/PSP4	27	I/O		
RD5/PSP5	28	I/O		
RD6/PSP6	29	I/O		
RD7/PSP7	30	I/O		
RE0/RD/AN5	8	I/O	พอร์ท E เป็นพอร์ท 2ทิศทาง RE0 เป็นสัญญาณอ่าน สำหรับพอร์ทขนาน หรือเป็นอนาล็อกอินพุทช่อง 5	
RE1/WR/AN6	9	I/O		RE1 เป็นสัญญาณเขียน สำหรับพอร์ทขนาน หรือเป็นอนาล็อกอินพุทช่อง 6
RE2/CS/AN7	10	I/O		RE2 เป็นสัญญาณเลือก สำหรับพอร์ทขนาน หรือเป็นอนาล็อกอินพุทช่อง 7
Vss	12,31	P	กราวด์	
Vdd	11,32	P	แรงดันบวก	

\* I = Input , O=Output, P=Power

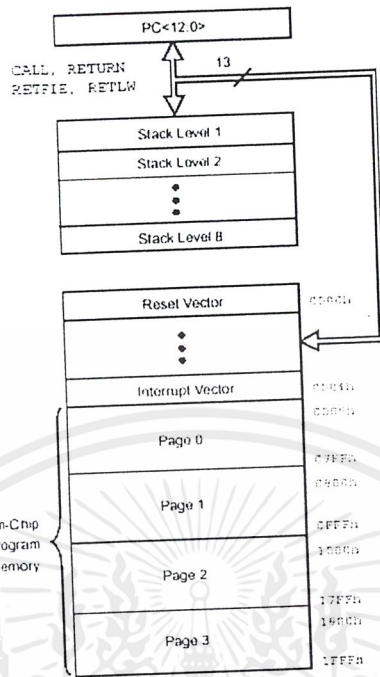
### 2.3 การจัดการหน่วยความจำ

หน่วยความจำของ PIC16F877 แบ่งเป็น 3 ส่วน คือ หน่วยความจำโปรแกรม, หน่วยความจำข้อมูล หรือ รีจิสเตอร์ และ หน่วยความจำข้อมูลแบบ EEPROM ซึ่งข้อมูลไม่สูญหาย แม้ไม่จ่ายไฟเลี้ยง

บัสของหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลจะถูกแยกออกจากกัน ดังนั้นการเข้าถึงหน่วยความจำทั้งสองแบบ สามารถเกิดขึ้นได้พร้อมกัน

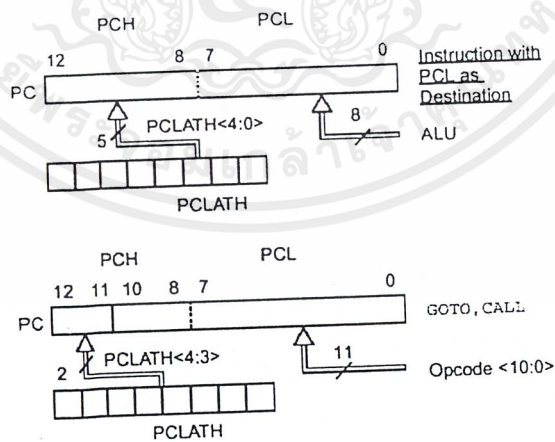
#### 2.3.1 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมขนาด 8 กิโลเวิร์ด ถูกแบ่งเป็น 4 เพจ(page) เพจละ 2 กิโลเวิร์ด โดยมีตำแหน่งของรีเซตเวคเตอร์อยู่ที่ 0000h และตำแหน่งของอินเตอร์รัพท์เวคเตอร์อยู่ที่ 0004h



รูปที่ 2.3 แผนผังหน่วยความจำโปรแกรม และแอสตักของ PIC16F877

2.3.2 โปรแกรมเคาน์เตอร์ (Program counter: PC) และแอสตัก (Stack)



รูปที่ 2.4 การใส่ค่าลงในโปรแกรมเคาน์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเคาน์เตอร์ มีขนาด 13 บิต ได้ไบท์ต่ำได้จากรีจิสเตอร์ PCL ซึ่งสามารถอ่านและเขียนได้ ส่วนไบท์สูง (PC<12:8>) ไม่สามารถอ่านได้ แต่สามารถเขียนได้โดยอ้อม ผ่านทางรีจิสเตอร์ PCLATH เมื่อเกิดการรีเซ็ต รีจิสเตอร์ทั้งสองจะกลายเป็นศูนย์

รูปที่ 2.4 แสดงการใส่ค่าลงในโปรแกรมเคาน์เตอร์ในสองสถานการณ์ รูปแรกค่าไบท์ต่ำได้จาก PCL และไบท์สูงได้จาก PCLATH<4:0> ส่วนรูปที่สองเกิดจากการทำคำสั่ง GOTO หรือ CALL ซึ่ง 11 บิตล่างของโปรแกรมเคาน์เตอร์ได้จาก Opcode <10:0> ส่วน 2 บิตบนได้จาก PCLATH<4:3> ทำให้การกระโดด ทำได้ภายใน 2048 ตำแหน่งในเพจเดียวกันเท่านั้น การอ้างแอดเดรสที่ไกลกว่านี้ ทำได้โดยเปลี่ยนแปลงค่าใน PCLATH<4:3> ซึ่งเป็นค่าที่จะถูกใส่ในโปรแกรมเคาน์เตอร์ 2 บิตบน ทำให้สามารถอ้างแอดเดรสได้ตลอด 8192 ตำแหน่ง การอ้างแอดเดรสที่สูงกว่ากำหนด จะทำให้เกิดการวนซ้ำตำแหน่งขึ้น

แสดงของ PIC16F877 มี 8 ระดับ แสดงไม่ใช่ส่วนของโปรแกรมหรือข้อมูลและแสดงพอยน์เตอร์ก็ไม่สามารถอ่านหรือเขียนได้ โปรแกรมเคาน์เตอร์จะถูกใส่ (Push) ลงในแอสตเมื่อเกิดการคำสั่ง CALL หรือเกิดการเรียกโปรแกรมบริการอินเตอร์รัพท์ และจะคืน(Pop) สู่อุปกรณ์เคาน์เตอร์เมื่อทำคำสั่ง RETURN, RETLW หรือ RETFIE

หากแอสตถูกใส่เกิน 8 ครั้ง ครั้งที่ 9 ค่าในโปรแกรมเคาน์เตอร์จะทับค่าที่ใส่ในแอสตครั้งแรกและวนทับไปเรื่อยๆ ใน PIC16F877 ไม่มีแฟลคที่บ่งบอกการเกิดแอสตโอเวอร์โฟลว์และแอสตอินเตอร์โฟลว์

### 2.3.3 หน่วยความจำข้อมูล (รีจิสเตอร์)

หน่วยความจำข้อมูลถูกแบ่งออกเป็น 4 แบนก์ แต่ละแบนก์มี 128 ไบท์ ประกอบด้วยรีจิสเตอร์ทั่วไป (General Purpose Register) และรีจิสเตอร์เฉพาะ (Special Function Register) การเข้าถึงแต่ละแบนก์ทำโดยการเปลี่ยนบิต RPI และ RPO ในรีจิสเตอร์สถานะ รีจิสเตอร์เฉพาะบางตัวที่สำคัญจะมีซ้ำอยู่ในหลายแบนก์เพื่อความรวดเร็วในการเข้าถึง

การเข้าถึงรีจิสเตอร์สามารถอ้างอิงแบบโดยตรงหรือโดยอ้อมผ่านทางไฟล์ซีเล็ก์รีจิสเตอร์ (File Select Register :FSR)

รีจิสเตอร์เฉพาะสามารถแบ่งเป็นรีจิสเตอร์ที่ถูกใช้โดยหน่วยประมวลผลกลาง และ รีจิสเตอร์สำหรับโมดูลที่ทำหน้าที่พิเศษ เช่นส่วนแปลงสัญญาณอนาล็อกเป็นดิจิตอล สำหรับรีจิสเตอร์เฉพาะที่ใช้โดยหน่วยประมวลผลกลางที่สำคัญคือ รีจิสเตอร์สถานะ

						File Address	
Indirect addr. <sup>(*)</sup>	00h	Indirect addr. <sup>(*)</sup>	80h	Indirect addr. <sup>(*)</sup>	100h	Indirect addr. <sup>(*)</sup>	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved <sup>(2)</sup>	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved <sup>(2)</sup>	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes		General Purpose Register 80 Bytes	
		accesses 70h-7Fh	EFh F0h	accesses 70h-7Fh	16Fh 170h	accesses 70h - 7Fh	1EFh 1F0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

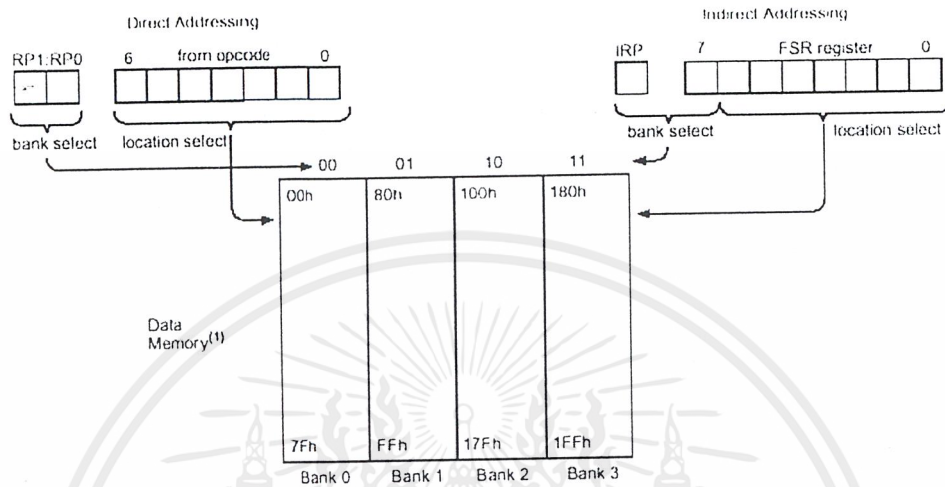
Unimplemented data memory locations, read as '0'.  
 \* Not a physical register.  
 Note 1: These registers are not implemented on 28-pin devices.  
 2: These registers are reserved, maintain these registers clear.

## รูปที่ 2.5 แผนผังหน่วยความจำข้อมูล (รีจิสเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### 2.3.4 การอ้างอิงแอดเดรสของรีจิสเตอร์โดยอ้อม



รูปที่ 2.6 การอ้างอิงแอดเดรสของรีจิสเตอร์โดยตรงและโดยอ้อม

การอ้างอิงแอดเดรสของรีจิสเตอร์โดยอ้อม ทำได้โดยใช้รีจิสเตอร์ INDF คำสั่งใดๆที่ใช้รีจิสเตอร์ตัวนี้ จะเป็นการเข้าถึงรีจิสเตอร์ตัวที่ถูกชี้โดยไฟล์ชี้เลือกรีจิสเตอร์ (FSR) การอ่านค่าจากรีจิสเตอร์ INDF เอง (FSR ที่ชี้ไปที่ตำแหน่ง 00) จะได้ค่า 00h และการเขียนค่าไปที่ตัวมันเอง จะไม่มีอะไรเกิดขึ้น (No Operation)

เนื่องจากรีจิสเตอร์ FSR มีเพียง 8 บิต การอ้างอิงรีจิสเตอร์ได้ทั้งหมดจะต้องใช้อีกหนึ่งบิตจาก บิต IRP (STATUS<7>)

## 2.4 ความสามารถทั่วไปของไมโครคอนโทรลเลอร์

### 2.4.1 การอินเทอร์รัพท์

PIC16F877 มีการอินเทอร์รัพท์ 14 แบบ โดยมีบิต GIE (Global Interrupt Enable) ในรีจิสเตอร์ INTCON เป็นตัวควบคุมการอินเทอร์รัพท์ทั้งหมด นอกจากนี้ในรีจิสเตอร์ INTCON ยังมีบิตควบคุมการอินเทอร์รัพท์ของพอร์ต RB0/INT, พอร์ต B และ TMR0 อีกด้วย ส่วนอินเทอร์รัพท์อื่นๆ มีรีจิสเตอร์ PIE1, PIE2, PIR1 และ PIR2 เป็นตัวควบคุม

เมื่อเกิดการอินเทอร์รัพท์ขึ้น บิต GIE จะถูกเคลียร์ เพื่อป้องกันการเกิดอินเทอร์รัพท์ซ้อน ค่าในโปรแกรมเคาน์เตอร์ขณะนั้นจะถูกเก็บในแสตค และโปรแกรมเคาน์เตอร์จะชี้ไปที่ตำแหน่ง

0004h ซึ่งเป็นตำแหน่งของโปรแกรมบริการอินเทอร์รัพท์ (Interrupt Service Routine) ด้วยการโพลลิ่ง (Polling) อินเทอร์รัพท์แฟล็ก จะสามารถตรวจสอบได้ว่าเกิดอินเทอร์รัพท์จากที่ใด

คำสั่ง RETFIE ใช้เพื่อออกจากโปรแกรมบริการอินเทอร์รัพท์ โดยบิต GIE จะถูกเซตโดยอัตโนมัติเพื่ออนุญาตการอินเทอร์รัพท์อีกครั้ง

อินเทอร์รัพท์แฟล็กจะต้องถูกเคลียร์โดยโปรแกรม ก่อนที่จะออกจากโปรแกรมบริการอินเทอร์รัพท์ หากไม่มีการเคลียร์อินเทอร์รัพท์แฟล็ก เมื่อบิต GIE ถูกเซต จะเกิดการเรียกโปรแกรมบริการอินเทอร์รัพท์ซ้ำ

### รีจิสเตอร์ควบคุมการอินเทอร์รัพท์ (Interrupt Control Register: INTCON)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

บิต7

บิต0

GIE

บิตควบคุมการอินเทอร์รัพท์ทั้งหมด  
1 = อนุญาตการอินเทอร์รัพท์ทั้งหมด  
0 = ไม่อนุญาตการอินเทอร์รัพท์ทั้งหมด

PEIE

บิตควบคุมการอินเทอร์รัพท์ย่อย  
1 = อนุญาตการอินเทอร์รัพท์  
0 = ไม่อนุญาตการอินเทอร์รัพท์

TOIE

บิตควบคุมการอินเทอร์รัพท์เมื่อ TMR0 เกิดโอเวอร์โฟลว์  
1 = อนุญาตการอินเทอร์รัพท์ของ TMR0  
0 = ไม่อนุญาตการอินเทอร์รัพท์ของ TMR0

INTE

บิตควบคุมการอินเทอร์รัพท์ของขา RB0/INT  
1 = อนุญาตการอินเทอร์รัพท์ของ RB0/INT  
0 = ไม่อนุญาตการอินเทอร์รัพท์ของ RB0/INT

RBIE

บิตควบคุมการอินเทอร์รัพท์เมื่อพอร์ท B เปลี่ยนสถานะ  
1 = อนุญาตการอินเทอร์รัพท์ของพอร์ท B  
0 = ไม่อนุญาตการอินเทอร์รัพท์ของพอร์ท B

TOIF

บิตแสดง TMR0 เกิดโอเวอร์โฟลว์  
1 = TMR0 เกิดโอเวอร์โฟลว์ (ต้องรีเซตโดยโปรแกรม)  
0 = TMR0 ไม่เกิดโอเวอร์โฟลว์

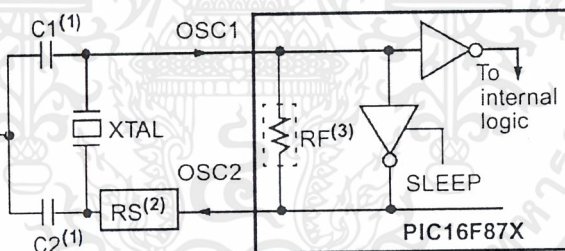
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- INTF**      บิตแสดงการอินเทอร์รัพท์ของขา RB0/INT  
 1 = เกิดการอินเทอร์รัพท์ของขา RB0/INT (ต้องรีเซ็ตโดยโปรแกรม)  
 0 = ไม่เกิดการอินเทอร์รัพท์ของขา RB0/INT
- RBIF**      บิตแสดงการอินเทอร์รัพท์เมื่อพอร์ท B เปลี่ยนสถานะ  
 1 = เกิดการอินเทอร์รัพท์ของพอร์ท B (ต้องรีเซ็ตโดยโปรแกรม)  
 0 = ไม่เกิดการอินเทอร์รัพท์ของพอร์ท B

**2.4.2 การใช้งานออสซิลเลเตอร์**

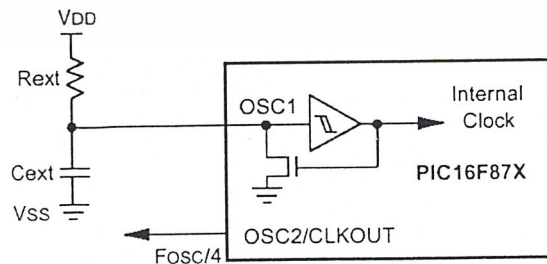
PIC16F877 สามารถเลือกทำงานได้กับออสซิลเลเตอร์ 4 แบบ โดยการโปรแกรมบิต FOSC1 และ FOSC0 ในรีจิสเตอร์ Configuration ออสซิลเลเตอร์ 4 แบบมีดังนี้

- LP คริสตอลพลังงานต่ำ
- XT คริสตอล/เรโซเนเตอร์
- HS คริสตอล/เรโซเนเตอร์ ความเร็วสูง
- RC รีจิสเตอร์-คาปาซิเตอร์



รูปที่ 2.7 แสดงการต่อคริสตอลออสซิลเลเตอร์

สำหรับคริสตอลออสซิลเลเตอร์ ต่อดังรูป 2.7 และควรเลือกใช้คริสตอลแบบตัดขนาน (Parallel cut crystal) เพื่อให้ได้ความถี่ใกล้เคียงกับที่ระบุไว้มากที่สุด



Recommended values:  $3\text{ k}\Omega \leq R_{ext} \leq 100\text{ k}\Omega$   
 $C_{ext} > 20\text{ pF}$

รูปที่ 2.8 แสดงการต่อรีจิสเตอร์-คาปาซิเตอร์ออสซิลเลเตอร์

การใช้รีซิสเตอร์-คาปาซิเตอร์ออสซิลเลเตอร์ เหมาะสำหรับงานที่ไม่ต้องการความถี่ ที่เที่ยงตรงมากนัก โดยค่าความถี่จะขึ้นอยู่กับแรงดัน  $V_{DD}$ , อุณหภูมิ, ค่าความต้านทาน และความจุของรีซิสเตอร์และคาปาซิเตอร์ รูปที่ 2.8 แสดงการต่อรีซิสเตอร์-คาปาซิเตอร์ออสซิลเลเตอร์

### 2.4.3 โหมดประหยัดพลังงาน (Sleep mode)

เมื่อไมโครคอนโทรลเลอร์ทำคำสั่ง SLEEP จะเป็นการเข้าสู่โหมดประหยัดพลังงาน บิตลดพลังงานในรีจิสเตอร์สถานะจะถูกเคลียร์ และออสซิลเลเตอร์จะถูกปิด แต่พอร์ทอินพุท เอาท์พุท จะยังคงสถานะเดิมอยู่ (1,0 หรือ hi-impedance)

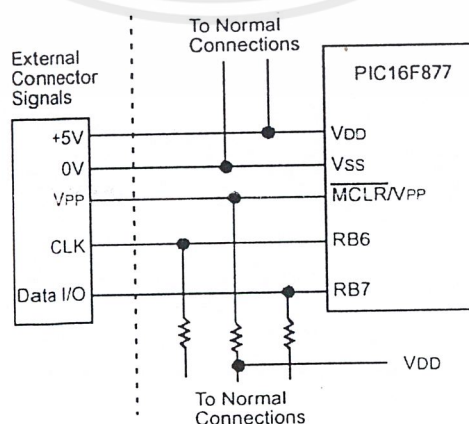
การออกจากโหมดประหยัดพลังงาน จะเกิดขึ้นเมื่อมีเหตุการณ์หนึ่งดังต่อไปนี้

- มีสัญญาณรีเซตที่ขา /MCLR
- การกระตุ้นจากรีลัดดีค็อกไทมเมอร์
- เกิดการอินเทอร์รัพท์

การเกิดอินเทอร์รัพท์จะทำให้ออกจากโหมดประหยัดพลังงาน โดยไม่สนว่าได้อนุญาตการอินเทอร์รัพท์หรือไม่ และหากไม่อนุญาตการอินเทอร์รัพท์ คำสั่งต่อจาก SLEEP จะถูกประมวลผลต่อไปตามปกติ หากอนุญาตการอินเทอร์รัพท์ คำสั่งต่อจาก SLEEP จะถูกประมวลผล แล้วกระโดดไปยังโปรแกรมบริการอินเทอร์รัพท์

### 2.4.4 การโปรแกรมไมโครคอนโทรลเลอร์ในระบบ (In-Circuit Serial Programming)

PIC16F877 สามารถโปรแกรมได้แม้อยู่ในวงจร โดยมีสัญญาณนาฬิกา สัญญาณข้อมูล ไฟเลี้ยง กราวนด์ และแรงดันไฟสูงสำหรับการ โปรแกรม ดังรูป



รูปที่ 2.9 การต่อไมโครคอนโทรลเลอร์เพื่อทำการโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 ส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล

ส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัลในไมโครคอนโทรลเลอร์ มีอนาล็อกอินพุต 8 ช่อง และสามารถเปลี่ยนการใช้งานเป็นดิจิทัลพอร์ทได้ด้วย ขั้นตอนการทำงานเริ่มที่แรงดันที่อนาล็อกอินพุตจะถูกประจุให้ตัวเก็บประจุแซมเปิลแอนด์โฮลด์ (Sample and Hold Capacitor) จากนั้นจะถูกแปลงด้วยวิธี Successive Approximation ที่ความละเอียด 10 บิต แรงดันอ้างอิงทั้งบวกและลบสามารถเลือกได้จากวงจรภายนอก หรือจาก  $V_{DD}$  และ  $V_{SS}$

การแปลงสัญญาณทำได้เมื่ออยู่ในสถานะ Sleep mode โดยต้องเลือกสัญญาณนาฬิกาที่ใช้ในการแปลงจากวงจรรีจิสเตอร์-คาปาซิเตอร์ออสซิลเลเตอร์ภายใน

### 2.5.1 รีจิสเตอร์ที่เกี่ยวข้องกับการแปลงสัญญาณ

มีรีจิสเตอร์ 4 ตัวที่เกี่ยวข้องกับการแปลงสัญญาณ ได้แก่

- ADCON0 ควบคุมการทำงาน
- ADCON1 ควบคุมหน้าที่ของพอร์ท
- ADRESH เก็บผลลัพธ์ไบท์สูง
- ADRESL เก็บผลลัพธ์ไบท์ต่ำ

#### รีจิสเตอร์ ADCON0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
บิต7							บิต0

ADCS1:ADCS0 บิตเลือกความถี่สัญญาณนาฬิกา

$$00 = F_{osc}/2$$

$$01 = F_{osc}/8$$

$$10 = F_{osc}/32$$

$$11 = F_{RC}$$

CHS2:CHS0 บิตเลือกช่องสัญญาณอนาล็อก

$$000 = \text{ช่อง } 0 \text{ (RA0/AN0)}$$

$$001 = \text{ช่อง } 1 \text{ (RA1/AN1)}$$

$$010 = \text{ช่อง } 2 \text{ (RA2/AN2)}$$

$$011 = \text{ช่อง } 3 \text{ (RA3/AN3)}$$

$$100 = \text{ช่อง } 4 \text{ (RA5/AN4)}$$

$$101 = \text{ช่อง } 5 \text{ (RE0/AN5)}$$

$$101 = \text{ช่อง } 6 \text{ (RE1/AN6)}$$

$$111 = \text{ช่อง } 7 \text{ (RE2/AN7)}$$

GO/DONE บิตสถานะของส่วนแปลงสัญญาณ

1 = ส่วนแปลงสัญญาณกำลังทำงาน (การเซตบิตนี้เป็นคำสั่งเริ่มการทำงาน)

0 = ไม่มีการทำงาน (ถูกเคลียร์โดยฮาร์ดแวร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADON เปิด/ปิด ส่วนแปลงสัญญาณ  
 1 = เปิดส่วนแปลงสัญญาณ  
 0 = ปิดส่วนแปลงสัญญาณ ทำให้ส่วนนี้ไม่มีการใช้พลังงาน

## รีจิสเตอร์ ADCON1

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0
บิต7				บิต0			

ADFM รูปแบบของผลลัพธ์ในรีจิสเตอร์ ADRESH และ ADRESL  
 1 = จัดชิดขวา โดย 2 บิตบนเก็บใน ADRESH และ 8 บิตล่างเก็บใน ADRESL  
 0 = จัดชิดซ้าย โดย 8 บิตบนเก็บใน ADRESH และ 2 บิตล่างเก็บใน ADRESL

## PCFG3:PCFG0 กำหนดหน้าที่ของพอร์ท

PCFG3 :PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	$V_{REF+}$	$V_{REF-}$	Chan /Ref
0000	A	A	A	A	A	A	A	A	$V_{DD}$	$V_{SS}$	8/0
0001	A	A	A	A	$V_{REF+}$	A	A	A	RA3	$V_{SS}$	7/1
0010	D	D	D	A	A	A	A	A	$V_{DD}$	$V_{SS}$	5/0
0011	D	D	D	A	$V_{REF+}$	A	A	A	RA3	$V_{SS}$	4/1
0100	D	D	D	D	A	D	A	A	$V_{DD}$	$V_{SS}$	3/0
0101	D	D	D	D	$V_{REF+}$	D	A	A	RA3	$V_{SS}$	2/1
011x	D	D	D	D	D	D	D	D	$V_{DD}$	$V_{SS}$	0/0
1000	A	A	A	A	$V_{REF+}$	$V_{REF-}$	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	$V_{DD}$	$V_{SS}$	6/0
1010	D	D	A	A	$V_{REF+}$	A	A	A	RA3	$V_{SS}$	5/1
1011	D	D	A	A	$V_{REF+}$	$V_{REF-}$	A	A	RA3	RA2	4/2
1100	D	D	D	A	$V_{REF+}$	$V_{REF-}$	A	A	RA3	RA2	3/2
1101	D	D	D	D	$V_{REF+}$	$V_{REF-}$	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	$V_{DD}$	$V_{SS}$	1/0
1111	D	D	D	D	$V_{REF+}$	$V_{REF-}$	D	A	RA3	RA2	1/2

A = ใช้พอร์ทเป็นอนาล็อกอินพุท

D = ใช้พอร์ทเป็นดิจิตอล

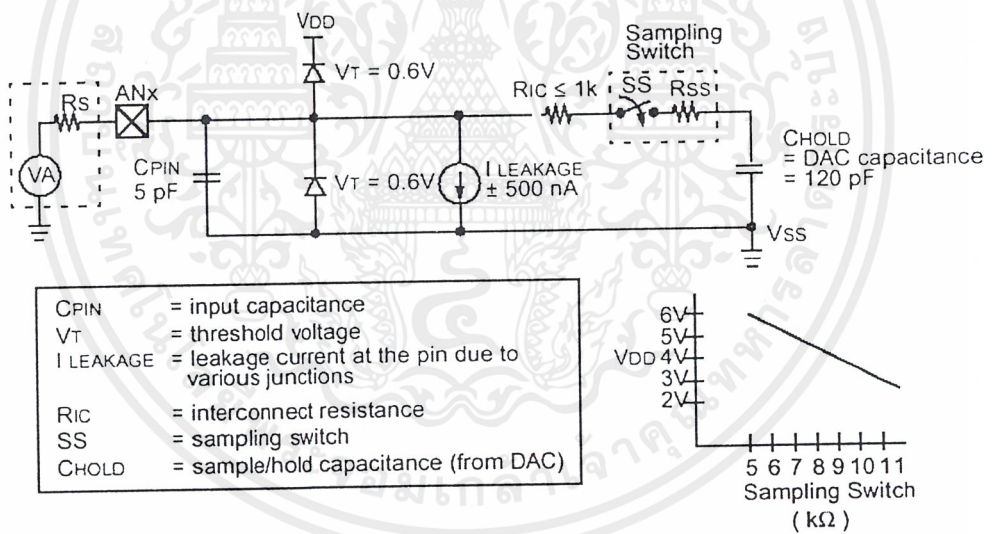
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์เก็บผลลัพท์

รีจิสเตอร์ ADRESH และ ADRESL ใช้เก็บผลลัพท์ไบท์สูงและไบท์ต่ำที่ได้จากการแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 10 บิต แต่ขนาดรวมของรีจิสเตอร์สองตัวนี้เป็น 16 บิต ทำให้ผู้ใช้สามารถเลือกรูปแบบการเก็บข้อมูลในรีจิสเตอร์ให้จัดชิดซ้ายหรือชิดขวาได้โดยการกำหนดบิต ADFM ในรีจิสเตอร์ ADCON1

2.5.2 ข้อกำหนดในการใช้งาน

เพื่อให้การแปลงสัญญาณได้ความแม่นยำมากที่สุด จะต้องออกแบบให้มีเวลาในการประจุคาปาซิเตอร์ ( $C_{HOLD}$ ) เพียงพอ เวลาที่ใช้ขึ้นอยู่กับค่าของรีจิสเตอร์สวิตช์ภายในโมดูล ( $R_{SS}$ ) และค่าอิมพีแดนซ์ของแหล่งกำเนิดสัญญาณ ( $R_S$ ) โดยค่าอิมพีแดนซ์ของแหล่งกำเนิดสัญญาณไม่ควรมากกว่า 10 กิโลโอห์ม



รูปที่ 2.10 วงจรเสมือนแสดงการประจุคาปาซิเตอร์  $C_{HOLD}$

### บทที่ 3

#### การสื่อสารแบบอนุกรม

การสื่อสารมีด้วยกัน 2 รูปแบบคือ การสื่อสารแบบขนานและการสื่อสารแบบอนุกรม การสื่อสารแบบขนาน เป็นการรับหรือส่งข้อมูลคราวละมากกว่า 1 บิตในเวลาเดียวกัน ทำให้การรับและส่งข้อมูลมีความเร็วสูง แต่จำนวนของสายสัญญาณที่ใช้ในการส่งผ่านข้อมูลต้องมีมากตามจำนวนบิตของข้อมูลที่ทำการส่ง นอกจากนี้ยังมีสายที่ใช้สำหรับควบคุมและตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูลก็ได้ ในขณะที่การรับส่งข้อมูลแบบอนุกรมจะเป็นการรับส่งข้อมูลครั้งละ 1 บิต การรับส่งข้อมูลแบบอนุกรมจึงมีข้อดีในเรื่องของจำนวนสายสัญญาณที่น้อยมากและไม่แปรตามจำนวนบิตของข้อมูล แต่ความเร็วในการสื่อสารจะลดลง และโปรแกรมควบคุมจะมีความซับซ้อนมากกว่า

การสื่อสารแบบอนุกรมยังแบ่งได้เป็น 2 แบบคือ การสื่อสารอนุกรมแบบซิงโครนัส (Synchronous) และการสื่อสารอนุกรมแบบอะซิงโครนัส (Asynchronous) การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาพร้อมกับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสคือ คีย์บอร์ดของคอมพิวเตอร์ หรือ巴士แบบ I<sup>2</sup>C ซึ่งสายเส้นหนึ่งเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์

การสื่อสารอนุกรมแบบอะซิงโครนัสรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วย แต่จะใช้การกำหนดค่าอัตราความเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราความเร็วนี้ว่า อัตราบอดหรือบอดเรต (Baud rate) มีหน่วยเป็น บิตต่อวินาที (Bit Per Second: Bps)

#### 3.1 การสื่อสารอนุกรมแบบ RS-232

สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association: EIA) ได้วางมาตรฐานที่มีชื่อว่า EIA RS-232 เป็นมาตรฐานการสื่อสารอนุกรมแบบอะซิงโครนัส 2 ทิศทาง มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต ระดับสัญญาณตั้งแต่ -3 โวลต์ จนถึง -12 โวลต์ แสดงว่ามีข้อมูล (Mark) หรือเทียบเท่าลอจิกหนึ่ง และ +3 โวลต์ ถึง +12 โวลต์ แสดงว่าเป็นช่องว่าง (Space) หรือลอจิกศูนย์

มาตรฐาน RS-232 กำหนดรูปแบบการสื่อสารข้อมูลกันระหว่างอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DEC) อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

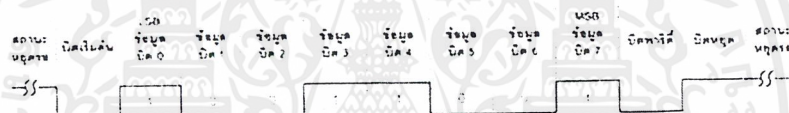
คอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิทข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DEC ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น

ปัจจุบันมาตรฐานนี้ได้พัฒนาถึงเวอร์ชัน C สำหรับการใช้งานในคอมพิวเตอร์ส่วนใหญ่ พอร์ตอนุกรม RS-232 ถูกใช้เพื่อเชื่อมต่อกับโมเด็ม (Modem) หรือเมาส์ (Mouse)

### 3.1.1 รูปแบบการรับส่งข้อมูลแบบอะซิงโครนัส

สัญญาณที่ใช้ในการรับส่งข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนด้วยกันคือ

- บิทเริ่มต้น (Start bit) 1 บิท
- บิทข้อมูลแบบอนุกรม 5,6,7 หรือ 8 บิท
- บิทตรวจสอบพาริตี (Parity bit) 1 บิทหรือไม่มี
- บิทปิดท้ายหรือบิทหยุด (Stop bit) 1,1.5 หรือ 2 บิท



รูปที่ 3.1 รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส

รูปที่ 3.1 แสดงรูปแบบการส่งข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีการส่งข้อมูล ขาข้อมูลจะมีสถานะลอจิกหนึ่ง เรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขาข้อมูล มีลอจิกศูนย์ ด้วยช่วงระยะเวลา 1 บิท เรียกบิทนี้ว่า บิทเริ่มต้น (Start bit) จากนั้นบิทข้อมูลจะถูกส่งออกไปโดยเริ่มจากบิทที่มีนัยสำคัญต่ำสุด (Least Significant Bit: LSB) ก่อน ซึ่งข้อมูลที่ต้องการส่งอาจมีจำนวน 5,6,7 หรือ 8 บิทก็ได้โดยต้องกำหนดให้ตรงกัน จากนั้นตามด้วยบิทพาริตี (Parity bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิทสุดท้ายที่จะส่งคือ บิทปิดท้ายหรือบิทหยุด (Stop bit) โดยจะเป็นการทำให้ขาข้อมูล มีสถานะลอจิกหนึ่งอีกครั้งด้วยระยะเวลา 1 บิท, 1.5 บิทหรือ 2 บิท เพื่อเป็นการแสดงว่าสิ้นสุดไบท์ข้อมูลแล้ว

การตรวจสอบพาริตีสามารถกำหนดให้เป็นพาริตีคี่ (Odd Parity), พาริตีคู่ (Even Parity) หรือไม่มีการตรวจสอบพาริตีก็ได้ พาริตีคี่หรือพาริตีคู่แสดงถึงจำนวนลอจิกหนึ่งทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบท์รวมบิทพาริตีว่ามีจำนวนเป็นเลขคู่หรือเลขคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิท มีค่า 10011001B จะเห็นว่าข้อมูลในไบท์นี้มีจำนวนลอจิกหนึ่ง จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าเป็นพาริตีคู่ ค่าของบิทพาริตีจะต้องมีลอจิกเป็นศูนย์ แต่ถ้ากำหนดเป็นพาริตีคี่ ค่าของบิทพาริตีจะต้องเป็นหนึ่ง เพื่อให้จำนวนบิทที่เป็นหนึ่งรวมบิทพาริตีมีค่าเป็นคี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูล ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตีที่ตรงกันไว้ด้วย หากเกิดความผิดพลาดในการส่งสัญญาณ ทางภาครับจะแสดงข้อผิดพลาดให้ผู้ใช้ทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่สามารถตรวจสอบได้เมื่อบิตของข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น None นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี

### 3.1.2 อัตราความเร็วในการรับส่งข้อมูลแบบอะซิงโครนัส

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอด หรือบอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS - 232 มีหลายค่า ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากบอดเรตคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมติว่า ข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบท์จะเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบท์ต่อวินาที

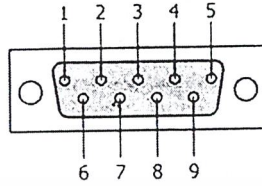
### 3.1.3 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 หรือ DB-9 ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยมีการใช้งานในอดีตไม่ค่อยมีความสำคัญมากนักจึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 3.2

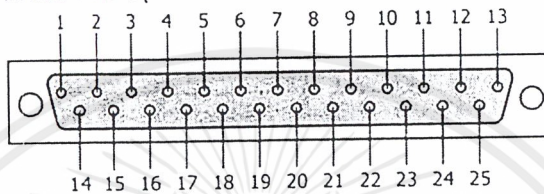
- **Receive Data: RD หรือ RxD** ขานี้ใช้เพื่อรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์ โดยจะนำข้อมูลที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- **Transmitted Data: TD หรือ TxD** ขานี้ใช้เพื่อส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- **Signal Ground: GND** เป็นขาราวด์ของสัญญาณ
- **Data Terminal Ready: DTR** เป็นขาที่ใช้สำหรับส่งสัญญาณออกจากคอมพิวเตอร์ เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อกับอุปกรณ์ปลายทางโดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ และถ้าใช้การเชื่อมต่อแบบ 3 สาย ต้องเชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อขา DTR และ DSR ของพอร์ตอนุกรมเข้าด้วยกัน และจะต้องต่อเชื่อมเข้ากับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการการตรวจจับสัญญาณพาห์



(ก) คอนเน็คเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



(ข) คอนเน็คเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

คอนเน็คเตอร์ DB-9	คอนเน็คเตอร์ 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

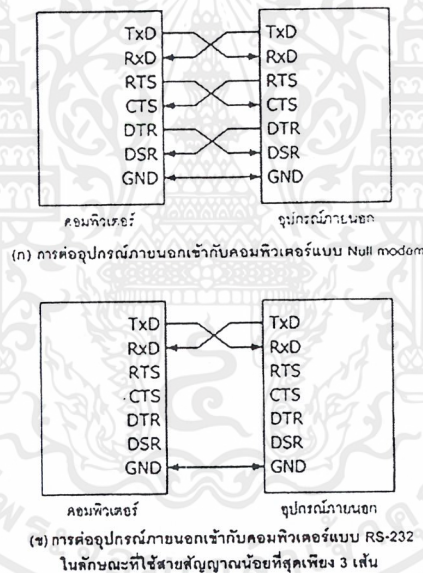
รูปที่ 3.2 แสดงการจัดขาสัญญาณของพอร์ตอนุกรม และหน้าที่การทำงาน

- **Data Set Ready: DSR** ขานี้จะใช้ควบคู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอก
- **Request To Send: RTS** เป็นขาเอาต์พุตสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลมาให้คอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ซึ่งในกรณีที่มีการเชื่อมต่อแบบ 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS เข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- **Clear To Send: CTS** เป็นขาอินพุตทำหน้าที่รอร์ับสัญญาณที่ส่งเข้ามา เมื่อมีการส่งสัญญาณเข้ามาที่ขานี้ ข้อมูลจะถูกส่งออกไปที่ขา TxD ขานี้จะใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลแล้วหรือยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Data Carrier Detect: DCD** หรือ **Carrier Detect: CD** ขานี้จะแอกทีฟเมื่อได้รับสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ใ้ถูกใช้งานมากนัก
- **Ring Indicator: RI** ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มแล้ว ต้องการตรวจสอบสัญญาณเรียกจากสายโทรศัพท์

สำหรับการเชื่อมต่อสายระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในรูปที่ 3.3 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล การเชื่อมต่อในรูปที่ 3.3 (ก) เป็นการเชื่อมต่อแบบ Null Modem ส่วนการเชื่อมต่อในรูปที่ 3.3 (ข) เป็นการเชื่อมต่อโดยใช้สายสัญญาณน้อยที่สุดเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์



รูปที่ 3.3 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในรูปแบบต่าง ๆ

### 3.1.4 Universal Asynchronous Receiver Transmitter

Universal Asynchronous Receiver Transmitter หรือ UART หมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัส หน้าที่หลักของ UART คือแปลงสัญญาณข้อมูลแบบขนานที่ส่งมาจากหน่วยประมวลผลกลางให้เป็นสัญญาณอนุกรมแบบอะซิงโครนัส แล้วทำการส่งออกไป และแปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่หน่วยประมวลผลกลาง ซึ่งนอกจาก UART จะทำหน้าที่แปลงรูปแบบของข้อมูลแล้ว ยังแจ้งราย

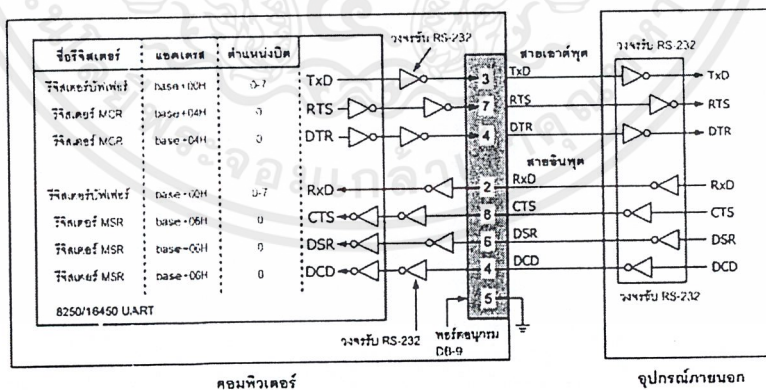
ละเอียดอื่นๆของข้อมูล ให้คอมพิวเตอร์รับทราบด้วย อาทิ อัตราเร็วในการรับส่งข้อมูลหรือบอดเรต, รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูล

ภายใน UART จะมีวงจรสร้างบอดเรตที่โปรแกรมได้ (Programmable Baudrate Gernerator) โดยกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้จะมีขนาด 16 บิต ทำให้สามารถกำหนดตัวหารอยู่ในช่วง 1 ถึง 65,535

ในเครื่องคอมพิวเตอร์ทั่วไปจะมี UART ที่ใช้งานกันอยู่ 2 เบอร์คือ เบอร์ 8250 และ 16550 สำหรับ UART เบอร์ 8250 เป็น UART มาตรฐานที่มีใช้ในคอมพิวเตอร์รุ่น XT โดย UART เบอร์นี้ มีบัฟเฟอร์สำหรับรับและส่งข้อมูลเป็นตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาที สำหรับ UART เบอร์ 16550 ถูกใช้ในคอมพิวเตอร์รุ่น AT จะเพิ่มส่วนของชิพที่รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบท์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ระดับ 256 กิโลบิตต่อวินาทีได้

ไอซี UART เหล่านี้มีระดับแรงดันของลอจิกเป็นแบบทีทีแอล (TTL) แต่เพื่อให้การรับส่งข้อมูลสามารถทำได้ในระยะทางมากขึ้น ระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิกศูนย์จะมีระดับแรงดัน -3 โวลต์ ถึง -12 โวลต์ และลอจิกหนึ่งมีระดับแรงดัน +3 โวลต์จนถึง +12 โวลต์

### 3.1.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232



รูปที่ 3.4 การทำงานภายในของขาสัญญาณต่างๆ ของพอร์ตอนุกรม

สัญญาณเอาต์พุต RTS และ DTR รวมทั้งสัญญาณสถานะอินพุต CTS, DSR และ DCD จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้ง RxD และ TxD จะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอล ดังนั้นสัญญาณที่ส่งออกมาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก UART จะเข้าสู่วงจรขับเพื่อแปลงระดับสัญญาณให้เป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกันเพื่อให้ได้สัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะกลับสถานะสัญญาณ ดังแสดงในรูปที่ 3.4

### 3.1.6 แอดเดรสของพอร์ตอนุกรม

แอดเดรสฐาน (Base) ของพอร์ตอนุกรมในคอมพิวเตอร์มี 4 ตำแหน่ง เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออส (BIOS) จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์

นอกจากนั้นบิตที่ 3 ถึง 1 ของหน่วยความจำตำแหน่ง 0000:0411H ยังใช้เพื่อแสดงจำนวนของพอร์ตอนุกรมที่มีอยู่ในคอมพิวเตอร์อีกด้วย

ตารางที่ 3.1 แสดงแอดเดรสฐานและหน่วยความจำที่เก็บแอดเดรสฐานของพอร์ตอนุกรม

พอร์ต	แอดเดรสฐาน	หน่วยความจำที่เก็บแอดเดรสฐาน
COM1	3F8H	0000:0400H ถึง 0000:0401H
COM2	2F8H	0000:0402H ถึง 0000:0403H
COM3	3E8H	0000:0404H ถึง 0000:0405H
COM4	2E8H	0000:0406H ถึง 0000:0407H

### 3.1.7 รีจิสเตอร์ของพอร์ตอนุกรม

พอร์ตอนุกรมมีรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวณได้จากแอดเดรสฐานของพอร์ตอนุกรม ตัวอย่างเช่น พอร์ตอนุกรม COM1 มีแอดเดรสฐานอยู่ที่ 3F8H แอดเดรสของรีจิสเตอร์ต่างๆจะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

#### รีจิสเตอร์บัฟเฟอร์ 00H

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือพักข้อมูลก่อนที่จะส่งออกไป มีแอดเดรสอยู่ที่ Base + 00H ถ้าเป็น COM1 จะมีแอดเดรสอยู่ที่ 3F8H การติดต่อกับรีจิสเตอร์นี้เพื่อส่งข้อมูลจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) มีสถานะเป็นศูนย์ เมื่อเขียนข้อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลไปยังแอดเดรสนี้ ข้อมูลจะถูกส่งออกไปแบบอนุกรม ในกรณีรับข้อมูล เมื่อ UART รับข้อมูลเข้ามาและแปลงเป็นแบบขนานแล้ว ข้อมูลแบบขนานจะถูกส่งไปยังรีจิสเตอร์บัฟเฟอร์ หลังจากมีการอ่านค่าจากรีจิสเตอร์นี้แล้ว รีจิสเตอร์นี้จะถูกเคลียร์ เพื่อให้พร้อมสำหรับการรับข้อมูลในไบต์ต่อไป

### รีจิสเตอร์ 01H

เป็นรีจิสเตอร์อนุญาตการอินเตอร์รัพท์ ซึ่งจะใช้เพื่อเซตโหมดในการอินเตอร์รัพท์ของพอร์ตอนุกรมซึ่งเป็นการกำหนดให้ UART สร้างสัญญาณอินเตอร์รัพท์ขึ้นมา มีแอดเดรสอยู่ที่ Base+01H ถ้าเป็น COM1 จะมีแอดเดรสอยู่ที่ 3F9H รายละเอียดในแต่ละบิตของรีจิสเตอร์ 01H มีดังนี้

-	-	-	-	SINP	ERBK	TBE	RxRD
							บิต0
บิต7							

SINP	1 = อนุญาตการอินเตอร์รัพท์เมื่อมีการเปลี่ยนสถานะที่ขาอินพุต CTS, DSR, DCD หรือขา RI 0 = ไม่มีการใช้อินเตอร์รัพท์นี้
ERBK	1 = อนุญาตการอินเตอร์รัพท์เมื่อเกิดความผิดพลาดขึ้นจากพริตี, โอเวอร์รัน, เฟรมข้อมูล หรือการหยุดข้อมูล 0 = ไม่มีการใช้อินเตอร์รัพท์นี้
TNE	1 = อนุญาตการอินเตอร์รัพท์เมื่อรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง 0 = ไม่มีการใช้อินเตอร์รัพท์นี้
RxRD	1 = อนุญาตการอินเตอร์รัพท์เมื่อข้อมูล 1 ไบต์ถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์แล้ว 0 = ไม่มีการใช้อินเตอร์รัพท์นี้

### รีจิสเตอร์ 02H

เป็นรีจิสเตอร์แสดงโหมดการอินเตอร์รัพท์ ใช้เพื่อตรวจสอบโหมดของการอินเตอร์รัพท์เมื่อมีการอินเตอร์รัพท์เกิดขึ้น มีแอดเดรสอยู่ที่ Base + 02H โดยมีรายละเอียดของแต่ละบิตดังนี้

-	-	-	-	-	ID1	ID0	PND
							บิต0
บิต7							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ID1:ID0**
- 00 = เกิดการอินเทอร์รัพท์เนื่องจากการเปลี่ยนแปลงของขาอินพุท (มีนัยสำคัญเป็นอันดับ 4 หรือนัยสำคัญต่ำสุด)
  - 01 = เกิดการอินเทอร์รัพท์เนื่องจากรีจิสเตอร์บัพเฟอร์สำหรับส่งข้อมูลว่าง (มีนัยสำคัญอันดับ 3)
  - 10 = เกิดการอินเทอร์รัพท์เนื่องจากข้อมูลถูกเก็บลงในรีจิสเตอร์บัพเฟอร์สำหรับรับข้อมูลแล้ว (มีนัยสำคัญอันดับ 2)
  - 11 = เกิดการอินเทอร์รัพท์เนื่องจากความผิดพลาดในการส่งข้อมูลหรือเกิดการหยุดกะทันหัน (มีนัยสำคัญเป็นอันดับ 1 หรือนัยสำคัญสูงสุด)

เมื่อมีการอินเทอร์รัพท์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะให้เกิดอินเทอร์รัพท์ครั้งต่อไปได้ โดยสามารถทำได้ดังนี้

- กรณีเกิดอินเทอร์รัพท์เนื่องจากการเปลี่ยนแปลงของขาอินพุท (ID1:ID0=0:0) จะต้องอ่านค่าจากรีจิสเตอร์ 06H เพื่อเคลียร์ค่าการอินเทอร์รัพท์
- กรณีเกิดอินเทอร์รัพท์เนื่องจากบัพเฟอร์สำหรับส่งข้อมูลว่าง (ID1:ID0=0:1) จะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัพเฟอร์ส่งข้อมูล (รีจิสเตอร์ 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเทอร์รัพท์ (รีจิสเตอร์ 02H) เพื่อเคลียร์ค่าการอินเทอร์รัพท์
- กรณีเกิดอินเทอร์รัพท์เนื่องจากเก็บข้อมูลลงในรีจิสเตอร์บัพเฟอร์สำหรับรับข้อมูลเรียบร้อยแล้ว สามารถเคลียร์ค่าอินเทอร์รัพท์โดยการอ่านข้อมูลจากรีจิสเตอร์บัพเฟอร์สำหรับเก็บข้อมูล (รีจิสเตอร์ 00H)
- กรณีเกิดอินเทอร์รัพท์เนื่องจากความผิดพลาดในการรับส่งข้อมูลหรือเกิดการหยุดกะทันหัน จะต้องเคลียร์ค่าอินเทอร์รัพท์โดยการอ่านค่ารีจิสเตอร์แสดงสถานะการรับและส่งข้อมูลแบบอนุกรม (รีจิสเตอร์ 05H)

**PND**                      1 = ไม่มีการอินเทอร์รัพท์                      0 = มีการอินเทอร์รัพท์เกิดขึ้น

**รีจิสเตอร์ 03H**

เป็นรีจิสเตอร์กำหนดรูปแบบของข้อมูล มีรายละเอียดของแต่ละบิตดังนี้

DLAB	BRK	PAR2	PAR1	PAR0	STOP	DAB1	DAB0
บิต7						บิต0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DLAB** 1 = เข้าสู่โหมดการหารค่าบอดเรท  
0 = เป็นการเข้าถึงรีจิสเตอร์บัพเฟอร์ (รีจิสเตอร์ 00H) และรีจิสเตอร์สำหรับการอินเตอร์รัพท์ (รีจิสเตอร์ 01H)

เมื่อบิต DLAB มีสถานะเป็น 1 รีจิสเตอร์บัพเฟอร์ และรีจิสเตอร์สำหรับการอินเตอร์รัพท์ จะถูกใช้สำหรับการโหลดค่าการหารความถี่สำหรับกำหนดค่าบอดเรท รีจิสเตอร์ 00H ใช้ในการกำหนดค่าการหารในไบท์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้ในการกำหนดค่าการหารในไบท์สูง โดยค่าบอดเรทเท่ากับ 115200 หารด้วยค่าตัวหาร 16 บิต (ค่า 115200 มาจากการหารคริสตอลความถี่ 1.8432 เมกกะเฮิร์ตซ์ที่อยู่ในวงจร UART)

**BRK** 1 = กำหนดให้สามารถหยุดการรับส่งข้อมูลกระทันหันได้  
0 = ไม่มีการหยุด

**PAR2: PAR1: PAR0** ใช้กำหนดบิตพาริตี

000 = ไม่ใช้บิตพาริตี

001 = กำหนดพาริตีคู่

011 = กำหนดพาริตีคู่

101 = มาร์ก (Mark)

111 = ช่องว่าง (Space)

**STOP**

1 = มีบิตปิดท้าย 2 บิต

0 = มีบิตปิดท้าย 1 บิต

**DAB1: DAB0** ใช้ในการกำหนดจำนวนบิตของข้อมูล

00 = จำนวนบิตข้อมูลเท่ากับ 5 บิต

01 = จำนวนบิตข้อมูลเท่ากับ 6 บิต

10 = จำนวนบิตข้อมูลเท่ากับ 7 บิต

11 = จำนวนบิตข้อมูลเท่ากับ 8 บิต

**รีจิสเตอร์ 04H**

เป็นรีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับการติดต่อกับโมเด็ม เช่น RTS หรือ DTR มีแอดเดรสอยู่ที่ Base + 04H มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

			<b>LOOP</b>	<b>OUT2</b>	<b>OUT1</b>	<b>RTS</b>	<b>DTR</b>
							บิต0
บิต7							

**LOOP** 1 = อนุญาตการส่งค่ากลับ 0 = ไม่อนุญาตการส่งค่ากลับ  
**RST** 1 = อนุญาตการใช้งานขา RTS 0 = ไม่อนุญาตการใช้งาน  
**DTR** 1 = อนุญาตการใช้งานขา DTR 0 = ไม่อนุญาตการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รีจิสเตอร์ 05H

เป็นรีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบบอนุกรมของ UART ซึ่งจะใช้งานร่วมกับรีจิสเตอร์สำหรับแสดงโหมดและสถานะของการอินเทอร์รัพท์ (รีจิสเตอร์ 02H) เพื่อแสดงสาเหตุของการเกิดอินเทอร์รัพท์ มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

	TXE	TBE	BREK	FRME	PARE	OVFE	RxRD
--	-----	-----	------	------	------	------	------

บิต7

บิต0

TXE	(Transmitter Empty) 1 = รีจิสเตอร์บัฟเฟอร์ส่งข้อมูลและชิพที่รีจิสเตอร์ว่าง 0 = มีข้อมูลเก็บอยู่ในรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลและชิพที่รีจิสเตอร์
TBE	(Transmitter Buffer Empty) 1 = รีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่าง 0 = มีข้อมูลเก็บอยู่ในรีจิสเตอร์บัฟเฟอร์ส่งข้อมูล
BREK	(Break) 1 = UART ตรวจพบการหยุดรับส่งข้อมูลกะทันหัน 0 = ไม่มีการหยุดรับส่งข้อมูลกะทันหัน
FRAME	(Frame Error) 1 = UART ตรวจพบความผิดพลาดแบบเฟรม 0 = ไม่มีข้อผิดพลาด
PARE	(Parity Error) 1 = UART ตรวจพบความผิดพลาดแบบพาริตี 0 = ไม่มีข้อผิดพลาด
OVRE	(Overrun Error) 1 = UART ตรวจพบความผิดพลาดแบบโอเวอร์รัน 0 = ไม่มีข้อผิดพลาด
RxRD	(Received Data Ready) 1 = บัฟเฟอร์รับข้อมูลเต็ม 0 = ไม่มีข้อมูลในบัฟเฟอร์รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รีจิสเตอร์ 06H

เป็นรีจิสเตอร์แสดงสถานะของโมเด็ม(Modem Status Register: MSR) ใช้แสดงสถานะของขา DCD, RI, DSR และ CTS มีแอดเดรสอยู่ที่ Base + 06H มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS
บิต7				บิต0			

DCD	1 = สัญญาณที่ขา DCD เป็น 1	0 = สัญญาณที่ขา DCD เป็น 0
RI	1 = สัญญาณที่ขา RI เป็น 1	0 = สัญญาณที่ขา RI เป็น 0
DSR	1 = สัญญาณที่ขา DSR เป็น 1	0 = สัญญาณที่ขา DSR เป็น 0
CTS	1 = สัญญาณที่ขา CTS เป็น 1	0 = สัญญาณที่ขา CTS เป็น 0
DDCD	(Delta Data Carrier Detect)	
	1 = บิต DCD มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว	
	0 = บิต DCD ไม่มีการเปลี่ยนแปลง	
DRI	(Delta Ring Indicator)	
	1 = บิต RI มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว	
	0 = บิต RI ไม่มีการเปลี่ยนแปลง	
DDSR	(Delta Data Set Ready)	
	1 = บิต DSR มีการเปลี่ยนแปลงเมื่อเทียบกับการอ่านค่าครั้งที่แล้ว	
	0 = บิต DSR ไม่มีการเปลี่ยนแปลง	
DCTS	(Delta Clear To Send)	
	1 = บิต CTS มีการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว	
	0 = บิต CTS ไม่มีการเปลี่ยนแปลง	

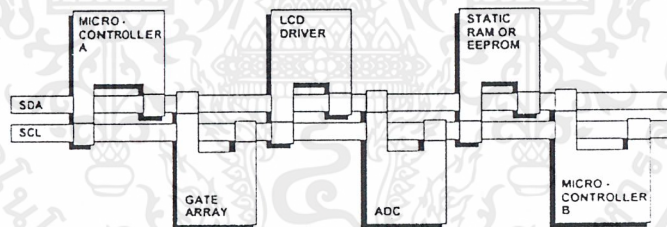
## รีจิสเตอร์ 07H

เป็นรีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราวหรือใช้ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบท์ ซึ่งจะไม่มีผลใดๆ กับการใช้งาน UART

### 3.2 ระบบบัส I<sup>2</sup>C

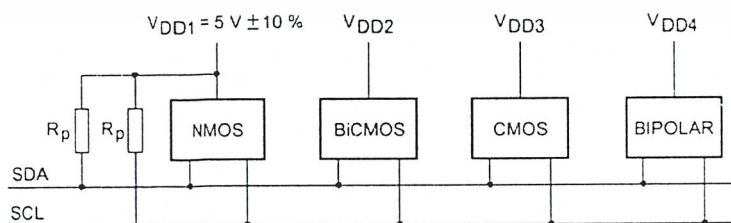
ระบบบัสแบบ I<sup>2</sup>C ย่อมาจาก inter-IC ถูกพัฒนาโดยบริษัทฟิลลิปส์ (Phillips) เป็นการสื่อสารอนุกรมแบบซิงโครนัส 2 ทิศทาง โดยมีลักษณะสำคัญดังนี้

- ใช้สายสัญญาณเพียง 2 สายในการติดต่อสื่อสารระหว่างไอซีมาสเตอร์กับไอซีสเลฟ สายหนึ่งเป็นสัญญาณข้อมูลหรือ SDA อีกสายเป็นสัญญาณนาฬิกาหรือ SCL
- ไอซีที่ต่อพ่วงบนบัสแต่ละตัวมีแอดเดรสเฉพาะที่แตกต่างกัน สามารถอ้างถึงโดยใช้ซอฟต์แวร์
- บนบัสเดียวกันสามารถมีไอซีมาสเตอร์ได้มากกว่าหนึ่งตัว เนื่องจากมีระบบป้องกันการใช้งานบัสพร้อมกัน
- การส่งข้อมูลเป็นแบบอนุกรม 8 บิตสองทิศทาง ที่ความเร็ว 100 กิโลบิตต่อวินาทีใน Standard Mode, 400 กิโลบิตต่อวินาทีใน Fast Mode และ 3.4 เมกกะบิตต่อวินาทีใน High-Speed Mode
- จำนวนไอซีที่ต่อบนบัสสามารถมีได้มากเท่าที่ค่าความจุรวมของบัสไม่เกิน 400 กิโลฟารัด



รูปที่ 3.5 ตัวอย่างระบบที่ใช้บัส I<sup>2</sup>C

#### 3.2.1 การต่อเชื่อมไอซีบนบัส



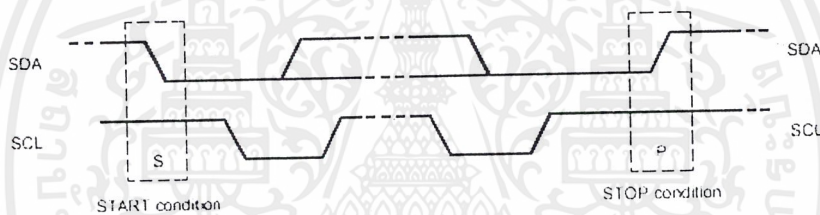
รูปที่ 3.6 การต่อไอซีบนบัส I<sup>2</sup>C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีแต่ละตัวสามารถต่อเชื่อมเข้ากับบัสได้โดยตรง โดยสายสัญญาณ SDA และ SCL จำเป็นต้องมีตัวต้านทานพูลอัพต่ออยู่กับแรงดันบวกตลอดเวลา เพื่อให้แรงดันในสายเป็นลอจิกหนึ่งเมื่อไม่มีการส่งข้อมูล และบัสแบบ I<sup>2</sup>C สามารถใช้กับไอซีที่มีระดับแรงดันต่างกันได้ด้วย

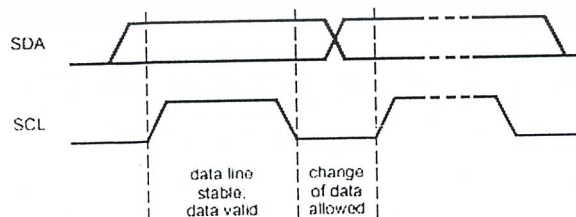
### 3.2.2 สถานะต่างๆของบัส

- สถานะบัสว่าง (Bus Not Busy) สายสัญญาณ SDA และ SCL เป็นลอจิกสูงทั้งคู่ คือไม่มีอุปกรณ์ตัวใดใช้บัสอยู่
- สถานะเริ่มต้น (Start) สายสัญญาณ SDA เปลี่ยนจากลอจิกหนึ่งเป็นศูนย์ในขณะที่สายสัญญาณ SCL เป็นลอจิกหนึ่ง เป็นการเริ่มต้นการติดต่อ
- สถานะหยุด (Stop) สายสัญญาณ SDA เปลี่ยนจากลอจิกศูนย์เป็นหนึ่งในขณะที่สายสัญญาณ SCL เป็นลอจิกหนึ่ง เป็นการหยุดการติดต่อ



รูปที่ 3.7 สถานะเริ่มต้นและสถานะหยุดของบัส I<sup>2</sup>C

- สถานะส่งข้อมูล (Data Validity) สายสัญญาณ SDA จะต้องไม่มีการเปลี่ยนสถานะ เมื่อสายสัญญาณ SCL มีลอจิกหนึ่ง ข้อมูลบิตนั้นจะเท่ากับสถานะของสายสัญญาณ SDA และเมื่อสายสัญญาณ SCL เปลี่ยนกลับมาเป็นลอจิกศูนย์ จึงจะสามารถเปลี่ยนสถานะในสายสัญญาณ SDA ได้



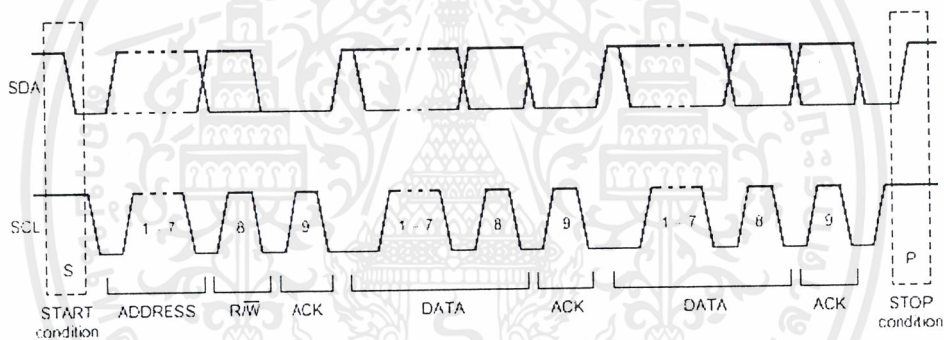
รูปที่ 3.8 สถานะส่งข้อมูลของบัส I<sup>2</sup>C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สถานะรับรู้ (Acknowledge) เมื่อตัวส่งข้อมูลส่งครบ 1 ไบต์หรือ 8 บิตแล้ว ตัวรับข้อมูลจะต้องตอบสนองการส่งข้อมูลนั้น โดยดึงสายสัญญาณ SDA ลงเป็นลอจิกศูนย์

### 3.2.3 รูปแบบการติดต่อสื่อสาร

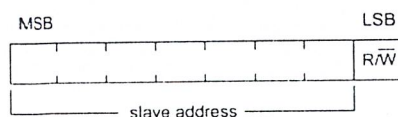
การติดต่อสื่อสารกันระหว่างไอซี ไอซีมาสเตอร์จะเป็นตัวควบคุมสายสัญญาณ SCL หรือเป็นตัวสร้างสัญญาณนาฬิกาตลอดการติดต่อ การติดต่อเริ่มจากไอซีมาสเตอร์สร้างสถานะเริ่ม แล้วส่งแอดเดรสและสัญญาณอ่านหรือเขียนข้อมูลลงบนบัส ไอซีสเลฟตัวที่มีแอดเดรสตรงกับแอดเดรสที่ส่งมาจะสร้างสัญญาณรับรู้กลับไป จากนั้นไอซีมาสเตอร์อาจส่งสัญญาณควบคุม หรือข้อมูลให้ไอซีสเลฟนั้น เมื่อครบแต่ละบิตไอซีสเลฟจะต้องสร้างสัญญาณรับรู้กลับไปทุกครั้ง แต่ถ้าเป็นการอ่านข้อมูลจากไอซีสเลฟ ไอซีมาสเตอร์จะต้องสร้างสัญญาณรับรู้ให้ไอซีสเลฟ เมื่อเสร็จสิ้นการติดต่อไอซีมาสเตอร์จะสร้างสถานะหยุดขึ้น ดังรูปเป็นตัวอย่างการติดต่อสื่อสารกันอย่างสมบูรณ์



รูปที่ 3.9 สัญญาณสื่อสารของบัส I<sup>2</sup>C

### 3.2.4 การอ้างอิงแอดเดรส

การอ้างอิงแอดเดรสแบ่งเป็นแบบ 7 บิตและ 10 บิต โดยการอ้างอิงแอดเดรสแบบ 7 บิต ข้อมูลใน 7 บิตแรกจะเป็นค่าแอดเดรสเริ่มจากบิตที่มีนัยสำคัญสูงสุด และบิตที่เหลือจะเป็นบิตควบคุมว่าจะอ่านหรือเขียนข้อมูล (R/W) ดังรูป



รูปที่ 3.10 รูปแบบแอดเดรส 7 บิตของบัส I<sup>2</sup>C

สำหรับการอ้างอิงแอดเดรสแบบ 10 บิต จะต้องใช้ 2 ไบต์ โดย 5 บิตในไบต์แรกจะต้องเป็น 11110 เท่านั้น ตามด้วยแอดเดรส 2 บิต และบิตควบคุม R/W ส่วนไบต์ที่สองจะเป็นแอดเดรส 8 บิตที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

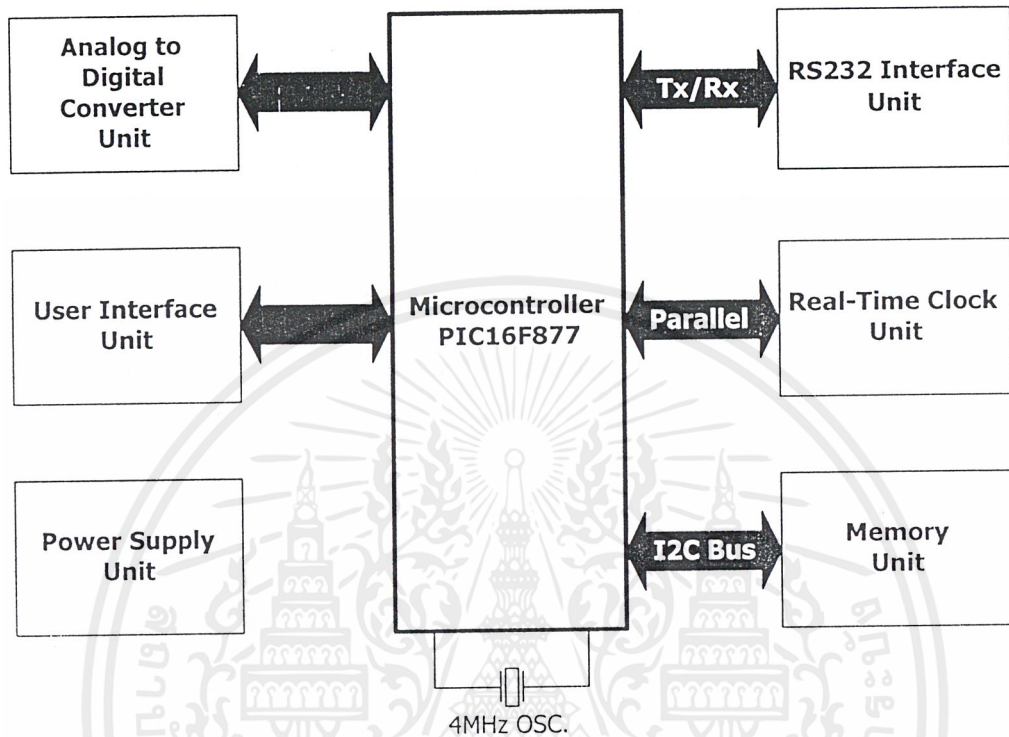
### การทำงานของวงจรและการออกแบบ

#### 4.1 รายละเอียดการทำงานของเครื่อง

เครื่องรวบรวมข้อมูลอัตโนมัติมีรายละเอียดการทำงานดังนี้

- ทำงานได้สองโหมดคือ Stand Alone จะเก็บข้อมูลที่ได้จากการแปลงสัญญาณอนาล็อกเป็นดิจิตอลไว้ใน EEPROM ก่อน จนถึงเวลาที่ต้องการจึงจะส่งข้อมูลไปยังคอมพิวเตอร์ และ Computer Based ข้อมูลที่แปลงได้จะถูกส่งไปยังคอมพิวเตอร์ทันที
- ความละเอียดในการแปลงสัญญาณอนาล็อกเป็นดิจิตอลเท่ากับ 8 บิต
- ช่องสัญญาณอนาล็อกอินพุต 4 ช่องสัญญาณ โดยเลือกใช้ได้เป็น 1, 2 หรือ 4 ช่อง
- มีแรงดันอ้างอิงสำหรับการแปลงสัญญาณ 2 ระดับคือ 2.50 โวลต์และ 4.00 โวลต์
- แสดงการทำงานด้วยจอแสดงผล LCD โดยจะแสดงโหมดการทำงาน, เวลาเริ่มเก็บข้อมูล, จำนวนช่องสัญญาณที่ใช้, ระยะห่างในการเก็บข้อมูล, จำนวนข้อมูลที่เก็บได้ และค่าของข้อมูลที่เก็บได้ในครั้งล่าสุด
- สื่อสารกับคอมพิวเตอร์ผ่านพอร์ต RS-232 ที่ความเร็ว 9600 บิตต่อวินาที และการทำงานส่วนใหญ่ถูกควบคุมจากผู้ใช้ผ่านทางคอมพิวเตอร์
- สามารถตั้งเวลาการทำงานล่วงหน้าได้มากที่สุด 1 วัน และระยะห่างของการเก็บข้อมูลแต่ละครั้งตั้งได้จาก 1 วินาทีถึง 24 ชั่วโมง

## 4.2 โครงสร้างของระบบโดยรวม

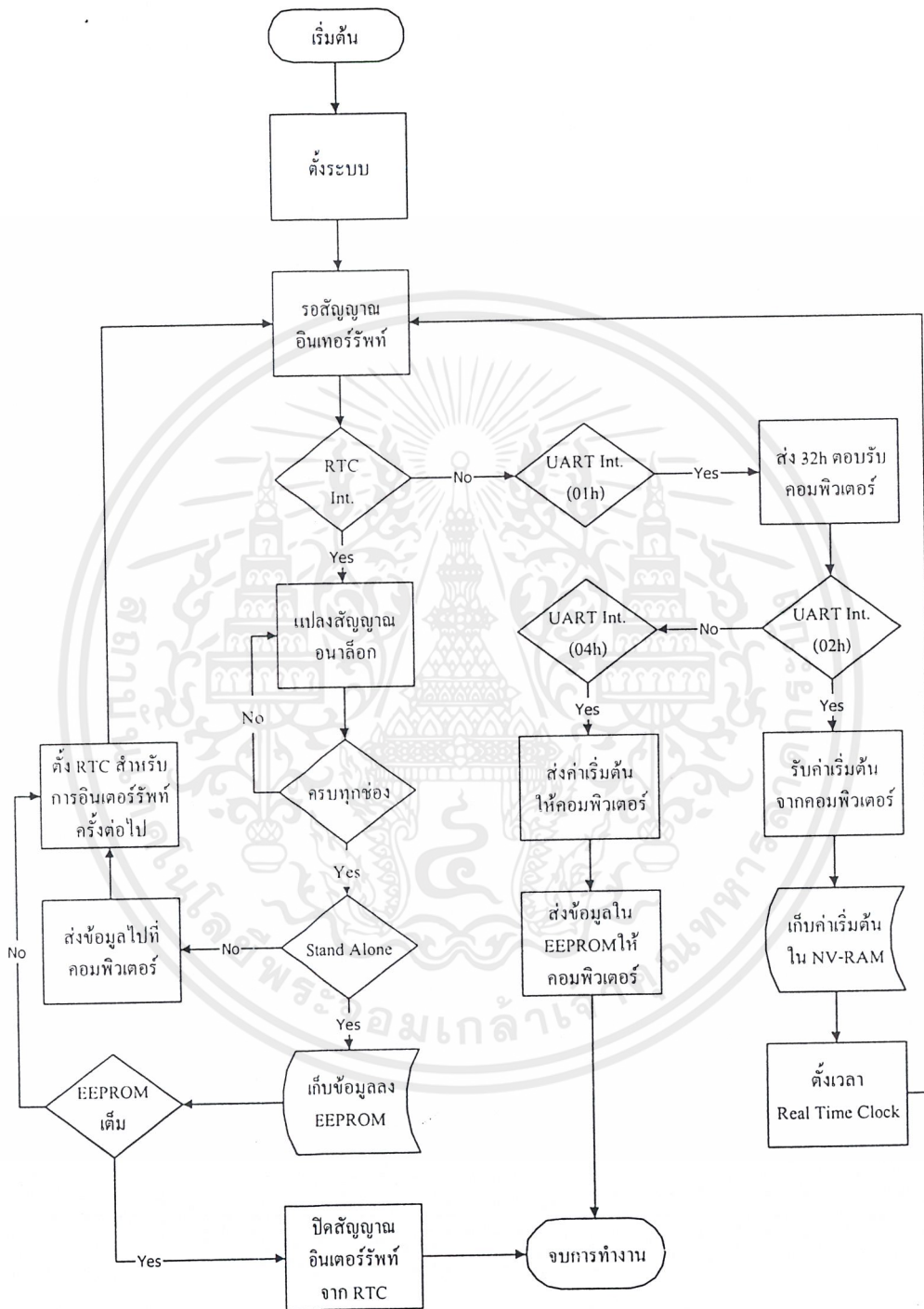


รูปที่ 4.1 บล็อกไดอะแกรมแสดงโครงสร้างของระบบ

โครงสร้างของระบบประกอบด้วย

- ไมโครคอนโทรลเลอร์ PIC16F877 ควบคุมการทำงานทั้งหมด
- ส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัลอยู่ในไมโครคอนโทรลเลอร์ ใช้แรงดันอ้างอิงจากไอซี MAX872 และ MAX874
- หน่วยความจำเก็บข้อมูลดิจิทัลที่แปลงได้ ใช้หน่วยความจำอนุกรม EEPROM เบอร์ 24C64 ขนาด 8 กิโลไบต์ที่ตัวต่อร่วมกันบนบัสแบบ I<sup>2</sup>C
- เวลาถูกควบคุมโดยไอซีคำนวณค่าเวลาจริงเบอร์ DS12877
- ส่วนติดต่อกับพอร์ทอนุกรม RS232-C ของคอมพิวเตอร์ ใช้ไอซี DS275 ในการแปลงระดับแรงดัน
- ส่วนการติดต่อกับผู้ใช้ ใช้ LCD ขนาด 16 ตัวอักษร 2 บรรทัดแสดงผล

## 4.3 การทำงานของระบบ



รูปที่ 4.2 ขั้นตอนการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

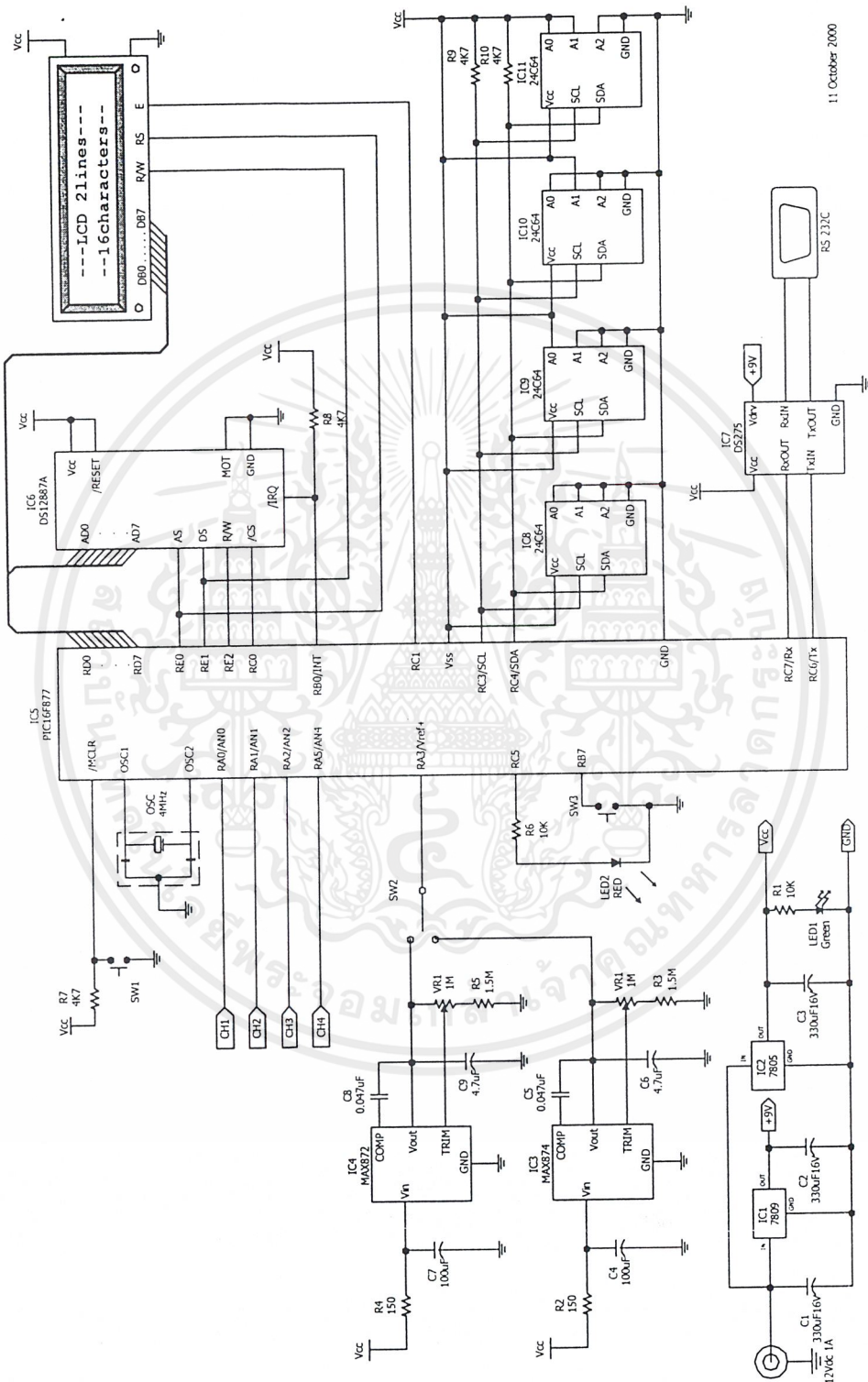
การทำงานของระบบเริ่มขึ้นเมื่อกดปุ่มรีเซ็ตของไมโครคอนโทรลเลอร์ หรือเริ่มจ่ายพลังงานให้กับเครื่อง ไมโครคอนโทรลเลอร์จะตั้งค่าการทำงานของส่วนต่างๆ จากนั้นจะรอรับการอินเทอร์รัพท์ ซึ่งเกิดได้จากสองแห่งคือ UART และอุปกรณ์คำนวณค่าเวลาจริง

UART อินเทอร์รัพท์เกิดขึ้นเมื่อผู้ใช้เรียกโปรแกรมค่าล็อกเกอร์บนวินโดว์ แล้วคลิกปุ่ม Connect หากเครื่องต่ออยู่กับพอร์ตอนุกรม RS232 ของคอมพิวเตอร์ โปรแกรมจะส่งค่า 01h ไปยังเครื่อง เครื่องจะส่งค่า 32h ตอบรับกลับมา จากนั้นผู้ใช้สามารถเลือกว่าจะทำอะไร หากต้องการตั้งค่าเพื่อการทำงานครั้งใหม่ทำโดยกดปุ่ม Setup โปรแกรมจะส่งค่า 02h ไปยังไมโครคอนโทรลเลอร์ และส่งค่าเริ่มต้นได้แก่เวลาเริ่มทำงาน ช่วงห่างของการเก็บข้อมูลแต่ละครั้ง จำนวนช่องใช้งาน (1, 2 หรือ 4 ช่อง) และโหมดการทำงาน ไมโครคอนโทรลเลอร์จะนำค่าที่ได้ไปเก็บใน NVRAM ตั้งเวลาปัจจุบันและเวลาเริ่มเก็บข้อมูลครั้งแรกให้กับอุปกรณ์คำนวณค่าเวลาจริง จากนั้นจะอนุญาตการอินเทอร์รัพท์จากอุปกรณ์คำนวณค่าเวลาจริง และรอการอินเทอร์รัพท์สำหรับการเก็บข้อมูลครั้งแรก

ถ้าต้องการเก็บข้อมูลที่ได้จากการทำงานครั้งล่าสุด ทำโดยกดปุ่ม View Data โปรแกรมจะส่งค่า 04h ไปที่เครื่อง หากเครื่องทำงานในโหมด Stand Alone จะส่งค่าเริ่มต้นและข้อมูลทั้งหมดใน EEPROM กลับมา แต่ถ้าทำงานในโหมด Computer Based เครื่องจะส่งเฉพาะค่าเริ่มต้นกลับมา และข้อมูลที่วัดได้จะส่งมาเมื่อถึงเวลา

การอินเทอร์รัพท์จากอุปกรณ์คำนวณค่าเวลาจริงจะเกิดขึ้นเมื่อถึงเวลาที่ตั้งไว้ เครื่องจะทำการแปลงสัญญาณอนาล็อกเป็นข้อมูลดิจิทัลขนาด 8 บิต หากอยู่ในโหมด Stand Alone ข้อมูลที่ได้จะถูกเก็บลง EEPROM อุปกรณ์คำนวณค่าเวลาจริงจะถูกตั้งเวลาเพื่อการเก็บข้อมูลครั้งต่อไป โดยจะทำงานนี้ไปจนกระทั่ง EEPROM เต็มจึงสิ้นสุดการเก็บข้อมูล แต่หากเครื่องอยู่ในโหมด Computer Based ข้อมูลที่แปลงได้จะถูกส่งไปยังคอมพิวเตอร์ทันที ปริมาณข้อมูลที่เก็บได้จึงขึ้นอยู่กับขนาดหน่วยความจำของคอมพิวเตอร์

4.4 วงจรและการทำงานของวงจร



11 October 2000

รูปที่ 4.3 วงจรของเครื่องรวบรวมข้อมูลอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ PIC16F877 ใช้สัญญาณนาฬิกาจากเซรามิกเรโซเนเตอร์ (Ceramic Resonator) ความถี่ 4 เมกกะเฮิรตซ์ ขา /MCLR หรือขารีเซตต่ออยู่กับแรงดัน  $V_{CC}$  และการกดสวิทช์ SW1 จะทำให้ไมโครคอนโทรลเลอร์รีเซต

#### 4.4.2 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล

วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล ประกอบด้วย 2 ส่วนคือวงจรแรงดันอ้างอิง และวงจรแปลงสัญญาณ

- วงจรแรงดันอ้างอิงใช้ไอซี MAX872 และ MAX874 ให้แรงดันอ้างอิงขนาด 2.5 โวลต์ และ 4 โวลต์เข้าที่ขา  $V_{REF}$  ของไมโครคอนโทรลเลอร์โดยมีสวิทช์ SW2 เปลี่ยนระดับแรงดัน รีซิสเตอร์ R2, R4 และคาปาซิเตอร์ C4, C7 กรองแรงดันกระแสที่มาจาก  $V_{CC}$  คาปาซิเตอร์ C5, C6, C8 และ C9 ลดทรานเซียนขณะเริ่มจ่ายแรงดัน รีซิสเตอร์ VR1, VR2 และ R3, R5 ใช้ปรับแรงดันเอาท์พุท
- วงจรแปลงสัญญาณอยู่ในไมโครคอนโทรลเลอร์ ใช้วิธีแปลงแบบ Successive Approximation ที่ความละเอียด 8 บิต

#### 4.4.3 หน่วยความจำแบบ EEPROM

หน่วยความจำแบบ EEPROM ใช้ไอซีเบอร์ 24C64 มีความจุ 8 กิโลไบต์ x 8 บิต 4 ตัวต่อพ่วงกันบนบัสแบบ I<sup>2</sup>C เข้าสูไมโครคอนโทรลเลอร์ที่ขา SDA และ SCL แอดเดรสของไอซีแต่ละตัวจะแตกต่างกัน เนื่องจากการต่อขาแอดเดรส A0, A1 และ A2 แตกต่างกัน ตัวต้านทาน R9 และ R10 เป็นตัวต้านทานพูลอัพของขา SDA และ SCL

#### 4.4.4 อุปกรณ์คำนวณค่าเวลาจริง

ไอซีเบอร์ DS12887A เป็นไอซีคำนวณค่าเวลาจริงที่สามารถสร้างสัญญาณอินเตอร์รัพท์ได้ที่ขา AD0 ถึง AD7 ซึ่งเป็นขาแอดเดรสและข้อมูลของไอซี ต่อกับขา RD0 ถึง RD7 ของไมโครคอนโทรลเลอร์ ขา MOT ต่อลงกราวด์เป็นการเลือกรูปแบบการติดต่อแบบอินเทล (ถ้าต่อกับ  $V_{CC}$  จะเป็นการติดต่อแบบโมโตโรล่า) ขาควบคุมได้แก่ AS, DS, R/W และ /CS ต่อกับขา RE0, RE1, RE2 และ RC0 ตามลำดับ ส่วนขาอินเตอร์รัพท์ /IRQ ต่อกับขา RB0/INT โดยมีตัวต้านทาน R6 พูลอัพ ขา /RESET ไม่ถูกใช้งานจึงต่อกับ  $V_{CC}$  และสำหรับขา SQW ซึ่งจะให้สัญญาณสี่เหลี่ยมที่สามารถโปรแกรมความถี่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.5 ส่วนติดต่อกับพอร์ตอนุกรม RS-232

เนื่องจากระดับแรงดันสำหรับลอจิกศูนย์และหนึ่งของพอร์ตอนุกรม RS-232 แตกต่างจากระดับแรงดันของทีทีแอล (TTL) หรือซีเอ็มอส (CMOS) จึงจำเป็นที่จะต้องแปลงระดับแรงดันเพื่อให้สามารถติดต่อสื่อสารกันได้ โดยใช้ไอซี DS275 ซึ่งทำหน้าที่นี้โดยเฉพาะ โดยระดับสัญญาณทีทีแอลที่ขา Txin จะถูกแปลงเป็น RS-232 ออกที่ขา Txout และระดับสัญญาณ RS-232 ที่ขา Rxin จะถูกแปลงเป็นทีทีแอลออกที่ขา Rxout

#### 4.4.6 ส่วนแสดงผล

จอ LCD แสดงผลขนาด 16 ตัวอักษร 2 บรรทัดใช้บัสแอสเคอเรสและข้อมูลร่วมกับบัสแอสเคอเรสและข้อมูลของอุปกรณ์คำนวณค่าเวลาจริง ส่วนขา R/W และขา RS ต่อกับขา RE0 และ RE1 ตามลำดับ และขา Enable ต่อกับขา RC1

LED2 ต่อกับพอร์ต RC5 เพื่อแสดงสถานะการทำงาน โดยจะกระพริบเมื่อมีการรีเซทไมโครคอนโทรลเลอร์ หรือเมื่อกำลังเก็บข้อมูล นอกจากนี้การกดสวิตช์ SW3 ค้างไว้ขณะเก็บข้อมูลจะเป็นการแสดงผลข้อมูลครั้งล่าสุดที่เก็บได้ในรูปของเลข 8 บิต

**บทที่ 5**  
**การทดสอบการทำงาน**

**5.1 การทดสอบการทำงานของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล**

การทดสอบการทำงานของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลขนาด 8 บิต (วงจรนี้เป็นส่วนหนึ่งภายในไมโครคอนโทรลเลอร์ PIC16F877) ทดสอบที่แรงดันอ้างอิงสองระดับ คือ +2.50 โวลต์ และ +4.00 โวลต์ โดยต่อแรงดันอ้างอิงบวกจากไอซี MAX872 และ MAX874 เข้าที่ขา RA3/V<sub>REF-</sub> แรงดันอ้างอิงลบเท่ากับกราวด์ และแรงดันทดสอบต่อจากวงจรแบ่งแรงดันเข้าที่ขา RA0/AN0 ปรับเพิ่มแรงดันทดสอบขึ้นครั้งละ 0.200 โวลต์ ได้ผลดังตาราง

ตารางที่ 5.1 แสดงค่าแรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลขนาด 8 บิต เมื่อแรงดันอ้างอิงเท่ากับ 2.5 โวลต์

แรงดันทดสอบ (โวลต์)	ค่าที่อ่านได้จากวงจร ADC (0-255)	แรงดันที่ได้จากวงจร ADC (โวลต์)*	ความผิดพลาด (%)
0.000	0	0.000	0.00
0.200	20	0.196	-2.00
0.400	42	0.412	+3.00
0.600	62	0.608	+1.33
0.800	85	0.833	+4.13
1.000	102	1.000	0.00
1.200	123	1.206	+0.50
1.400	144	1.412	+0.86
1.600	164	1.608	+0.50
1.800	184	1.804	+0.22
2.000	204	2.000	0.00
2.200	224	2.196	-0.18
2.400	244	2.392	-0.33

\*แรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล เท่ากับค่าที่อ่านได้คูณด้วย 2.5/255 โวลต์

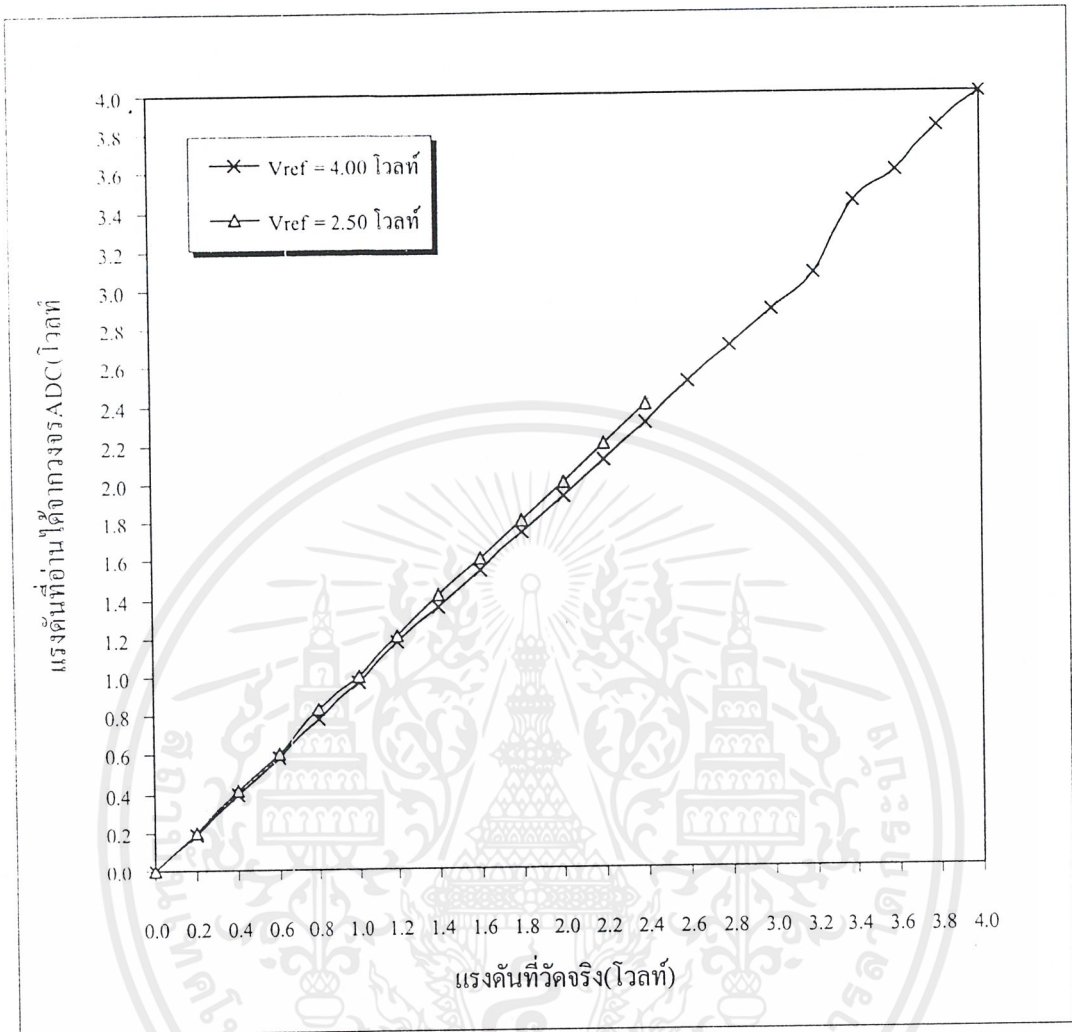
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 แสดงค่าแรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 8 บิต เมื่อแรงดันอ้างอิงเท่ากับ 4 โวลต์

แรงดันทดสอบ (โวลต์)	ค่าที่อ่านได้จากวงจร ADC (0-255)	แรงดันที่ได้จากวงจร ADC (โวลต์)*	ความผิดพลาด (%)
0.000	0	0.000	0.00
0.200	12	0.188	-6.00
0.400	25	0.392	-2.00
0.600	37	0.580	-3.33
0.800	50	0.784	-2.00
1.000	62	0.973	-2.70
1.200	75	1.176	-2.00
1.400	86	1.349	-3.64
1.600	98	1.537	-3.94
1.800	111	1.741	-3.28
2.000	123	1.929	-3.55
2.200	135	2.118	-3.73
2.400	147	2.305	-3.96
2.600	160	2.510	-3.46
2.800	172	2.698	-3.64
3.000	184	2.886	-3.80
3.200	196	3.074	-3.94
3.400	219	3.435	+1.03
3.600	229	3.592	-0.22
3.800	244	3.827	+0.71
4.000	255	4.000	0.00

\*แรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล เท่ากับค่าที่อ่านได้ คูณด้วย 4/255 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 กราฟแสดงความสัมพันธ์ของค่าแรงดันที่ได้จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล กับค่าแรงดันทดสอบ เมื่อแรงดันอ้างอิงเท่ากับ 2.50 โวลต์ และ 4.00 โวลต์

## 5.2 การทดสอบการทำงานของเครื่อง

โดยการทดสอบเครื่องในโหมด Stand Alone และ Computer Based เก็บข้อมูลทุก 5 วินาที โดยใช้แรงดันทดสอบขนาด 0.00, 1.20, 2.00 และ 2.50 โวลต์ที่แรงดันอ้างอิง 2.50 โวลต์ และแรงดันทดสอบขนาด 0.00, 2.00, 3.00 และ 4.00 โวลต์ที่แรงดันอ้างอิง 4.00 โวลต์

Record	Channel1	Channel2	Channel3	Channel4
16:59:05	0.000	1.196	2.000	2.500
16:59:10	0.000	1.186	2.000	2.500
16:59:15	0.000	1.186	2.000	2.500
16:59:20	0.000	1.186	2.000	2.500
16:59:25	0.000	1.196	2.000	2.500
16:59:30	0.000	1.186	2.000	2.500
16:59:35	0.000	1.196	2.000	2.500
16:59:40	0.000	1.186	2.000	2.500
16:59:45	0.000	1.186	2.000	2.500
16:59:50	0.000	1.186	2.000	2.500
16:59:55	0.000	1.186	2.000	2.500
17:00:00	0.000	1.196	2.000	2.500
17:00:05	0.000	1.186	2.000	2.500
17:00:10	0.000	1.186	2.000	2.500
17:00:15	0.000	1.186	2.000	2.500
17:00:20	0.000	1.196	2.000	2.500
17:00:25	0.000	1.196	2.000	2.500

Setting Time : 16:58:39 February 6, 2001  
 Launch Time : 16:58:50 February 6, 2001  
 Last Update : 17:00:25 February 6, 2001

Done | Stand Alone | 4 CH | 5 Seconds | 20 Rec

รูปที่ 5.2 ผลการทำงานในโหมด Stand Alone ที่แรงดันอ้างอิง 2.50 โวลต์

**Upload - DataLogg ME**

File Data

Setting Time : 16:54:43 February 6, 2001  
 Launch Time : 16:55:00 February 6, 2001  
 Last Update : 16:56:35 February 6, 2001

Record	Channel1	Channel2	Channel3	Channel4 ▲
16:55:15	0.000	1.206	2.000	2.500
16:55:20	0.000	1.196	2.000	2.500
16:55:25	0.000	1.196	2.000	2.500
16:55:30	0.000	1.196	2.010	2.500
16:55:35	0.000	1.196	2.000	2.500
16:55:40	0.000	1.196	2.000	2.500
16:55:45	0.000	1.196	2.000	2.500
16:55:50	0.000	1.196	2.000	2.500
16:55:55	0.000	1.196	2.000	2.500
16:56:00	0.000	1.196	2.000	2.500
16:56:05	0.000	1.196	2.000	2.500
16:56:10	0.000	1.206	2.000	2.500
16:56:15	0.000	1.196	2.000	2.500
16:56:20	0.000	1.196	2.000	2.500
16:56:25	0.000	1.186	2.000	2.500
16:56:30	0.000	1.196	2.000	2.500
16:56:35	0.000	1.196	2.000	2.500

Upload Close

Monitoring... Computer Base 4 CH 5 Seconds 20 Rec

รูปที่ 5.3 ผลการทำงานในโหมด Computer Base ที่แรงดันอ้างอิง 2.50 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Upload - DataLog ME				
File Data				
Setting Time :		17:13:40	February 6, 2001	
Launch Time :		17:13:45	February 6, 2001	
Last Update :		17:15:20	February 6, 2001	
Record	Channel1	Channel2	Channel3	Channel4 ▲
17:14:00	0.000	1.459	2.949	4.000
17:14:05	0.000	1.459	2.949	4.000
17:14:10	0.000	1.459	2.949	4.000
17:14:15	0.000	1.459	2.933	4.000
17:14:20	0.000	1.459	2.949	4.000
17:14:25	0.000	1.459	2.949	4.000
17:14:30	0.000	1.459	2.949	4.000
17:14:35	0.000	1.459	2.949	4.000
17:14:40	0.000	1.459	2.949	4.000
17:14:45	0.000	1.459	2.949	4.000
17:14:50	0.000	1.459	2.949	4.000
17:14:55	0.000	1.459	2.949	4.000
17:15:00	0.000	1.459	2.933	4.000
17:15:05	0.000	1.459	2.949	4.000
17:15:10	0.000	1.459	2.949	4.000
17:15:15	0.000	1.459	2.949	4.000
17:15:20	0.000	1.459	2.949	4.000
<input type="button" value="Upload"/> <input type="button" value="Close"/>				
Done	Stand Alone	4 CH	5 Seconds	20 Rec

รูปที่ 5.4 ผลการทำงานในโหมด Stand Alone ที่แรงดันอ้างอิง 4.00 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Setting Time : 17:10:58 February 6, 2001  
 Launch Time : 17:11:20 February 6, 2001  
 Last Update : 17:12:55 February 6, 2001

Record	Channel1	Channel2	Channel3	Channel4 ▲
17:11:35	0.000	1.459	2.949	4.000
17:11:40	0.000	1.475	2.949	4.000
17:11:45	0.000	1.459	2.949	4.000
17:11:50	0.000	1.459	2.949	4.000
17:11:55	0.000	1.459	2.949	4.000
17:12:00	0.000	1.459	2.949	4.000
17:12:05	0.000	1.459	2.933	4.000
17:12:10	0.000	1.459	2.949	4.000
17:12:15	0.000	1.459	2.949	4.000
17:12:20	0.000	1.459	2.949	4.000
17:12:25	0.000	1.459	2.949	4.000
17:12:30	0.000	1.459	2.949	4.000
17:12:35	0.000	1.459	2.949	4.000
17:12:40	0.000	1.459	2.949	4.000
17:12:45	0.000	1.459	2.949	4.000
17:12:50	0.000	1.459	2.949	4.000
17:12:55	0.000	1.459	2.949	4.000

Monitoring... Computer Base 4 CH 5 Seconds 20 Rec

รูปที่ 5.5 ผลการทำงานในโหมด Computer Base ที่แรงดันอ้างอิง 4.00 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### สรุปผลการทำงาน

#### 6.1 สรุปผลการทำงาน

จากการทดสอบการทำงานของส่วนประกอบสำคัญในระบบ สรุปได้ดังนี้

- วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลสามารถแปลงสัญญาณได้ค่อนข้างเป็นเชิงเส้น โดยมีความผิดพลาดในการแปลงสัญญาณน้อยกว่า 6 เปอร์เซ็นต์
- เรียบิลท์บล็อกสามารถสร้างสัญญาณอินเตอร์รัพท์ได้ตามต้องการ
- การรับและส่งสัญญาณอนุกรมอะซิงโครนัสระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ สามารถรับและส่งได้โดยไม่เกิดความผิดพลาดของข้อมูล
- การอ่านและเขียนข้อมูลกับหน่วยความจำแบบ EEPROM สามารถอ่านและเขียนข้อมูลได้ถูกต้อง

เมื่อนำส่วนประกอบทั้งหมดมาสร้างเป็นเครื่องรวบรวมข้อมูลอัตโนมัติตามวงจรรูปที่ 4.3 แล้วทดสอบการทำงานทั้งหมด สรุปได้ว่าเครื่องสามารถทำงานได้ถูกต้องทุกฟังก์ชัน ได้แก่

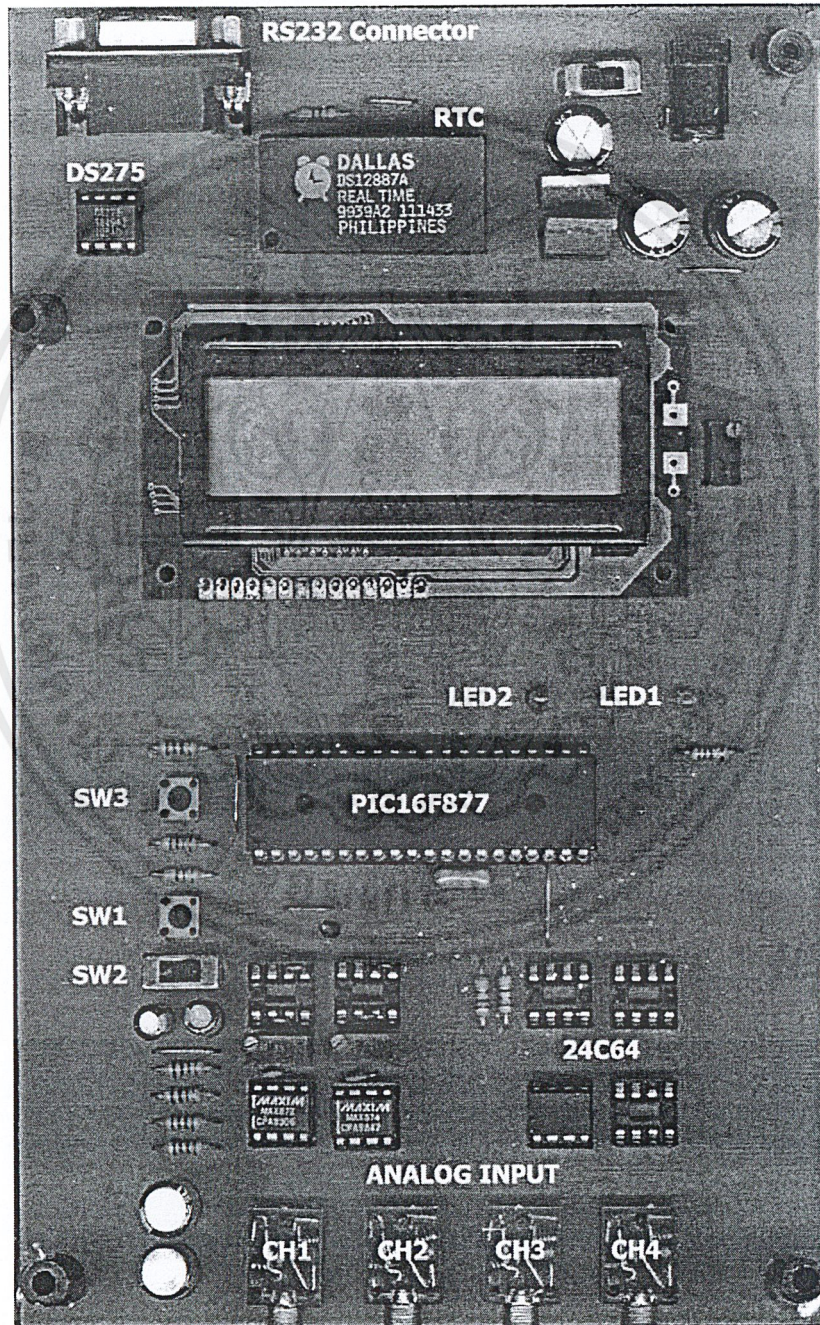
- เวลาเริ่มเก็บข้อมูล ช่วงห่างของการเก็บข้อมูลแต่ละครั้ง สามารถทำได้ตรงเวลา
- จำนวนช่องสัญญาณที่แปลงตรงกับที่ตั้งไว้
- เครื่องทำงานตามโหมดที่กำหนด
- การเก็บข้อมูลลง EEPROM และการอ่านกลับถูกต้อง
- การรับค่าเริ่มต้นจากคอมพิวเตอร์และการส่งข้อมูลกลับถูกต้อง

#### 6.2 แนวทางในการพัฒนาและประยุกต์ใช้งาน

การพัฒนาเครื่องรวบรวมข้อมูลอัตโนมัติสามารถทำได้หลายทาง จึงขอยกตัวอย่างบางประการดังนี้

- การเพิ่มความละเอียดในการแปลงสัญญาณอนาล็อกเป็นดิจิตอล ไมโครคอนโทรลเลอร์ PIC16F877 สามารถแปลงสัญญาณได้ความละเอียดสูงถึง 10 บิต ซึ่งจะทำให้ความถูกต้องของข้อมูลสูงขึ้น
- การเพิ่มจำนวนช่องเก็บข้อมูล ,การเพิ่มความจุของหน่วยความจำ และการลดคาบเวลาการเก็บข้อมูลให้ต่ำกว่า 1 วินาที
- การเพิ่มความสามารถในการจัดการข้อมูลของโปรแกรมคอมพิวเตอร์ เช่น การนำข้อมูลมาวาดกราฟ หรือการจัดเก็บเป็นฐานข้อมูล (Data Base)

ภาคผนวก ก.  
เครื่องรวบรวมข้อมูลอัตโนมัติ



รูปที่ ก.1 เครื่องรวบรวมข้อมูลอัตโนมัติ

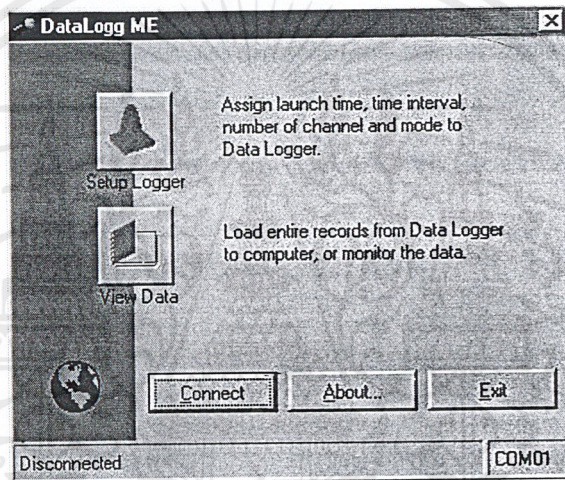
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข.

## การใช้งานเครื่องรวบรวมข้อมูลอัตโนมัติ

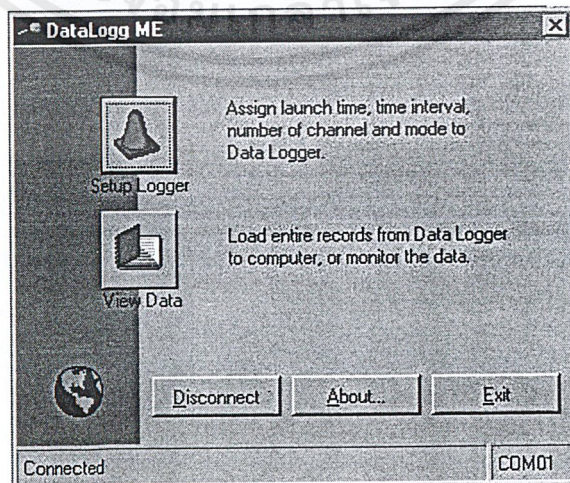
การใช้งานเครื่องรวบรวมข้อมูลอัตโนมัติมีขั้นตอนดังนี้

1. ต่อสายจากพอร์ตอนุกรม RS232 ของคอมพิวเตอร์เข้ากับตัวเครื่อง เปิดเครื่องแล้วเปิดโปรแกรม DataLogg จะขึ้นหน้าจอหลักดังรูป



รูปที่ ข.1 หน้าจอหลักขณะยังไม่ติดต่อกับเครื่อง

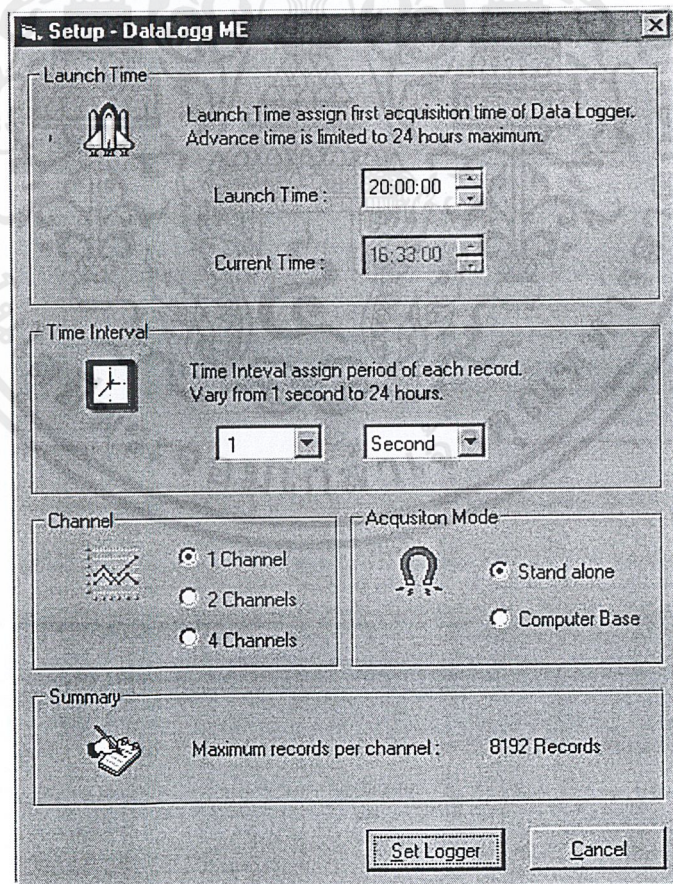
2. เลือกพอร์ตอนุกรมที่ใช้งาน (COM port) จากนั้นกดปุ่ม Connect



รูปที่ ข.2 หน้าจอหลักขณะติดต่อกับเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกปุ่ม Setup Logger เพื่อตั้งค่าการทำงานต่างๆให้กับตัวเครื่องได้แก่
- Launch Time ส่วนตั้งเวลาสำหรับการเริ่มเก็บข้อมูล ซึ่งสามารถตั้งเวลาล่วงหน้าได้ถึง 24 ชั่วโมง
  - Time Interval ช่วงเวลาห่างของการเก็บข้อมูลแต่ละครั้ง จาก 1 วินาที ถึง 1 วัน
  - Channel จำนวนช่องใช้งาน เลือกได้ระหว่าง 1, 2 หรือ 4 ช่อง
  - Acquisition Mode รูปแบบการเก็บข้อมูล เลือก “ Stand alone ” สำหรับการเก็บข้อมูล โดยไม่ต่อกับเครื่องคอมพิวเตอร์ และ “ Computer based ” สำหรับการเก็บข้อมูลแล้วส่งผ่านให้เครื่องคอมพิวเตอร์ทันที
- จากนั้นกดปุ่ม Set Logger เครื่องจะถูกตั้งค่าตามที่กำหนด



รูปที่ ข.3 หน้าจอสำหรับการตั้งค่าเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือกแรงดันอ้างอิงสำหรับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล 2.50 หรือ 4.00 โวลต์ ที่สวิตช์ SW2
5. หากเลือกโหมด Stand Alone จะสามารถถอดสายอนุกรมออกได้และนำเครื่องไปวางยังตำแหน่งที่ต้องการ แต่ถ้าเป็นโหมด Computer Based จะต้องต่อเครื่องไว้กับพอร์ตอนุกรมตลอด และเปิดหน้าจอ View Data ด้วย
6. เครื่องจะทำการเก็บข้อมูลเมื่อถึงเวลาที่กำหนดโดยอัตโนมัติ สังเกตได้จาก LED2 ซึ่งจะติดในขณะที่เก็บข้อมูล ผู้ใช้สามารถดูโหมดการทำงาน สถานะของเครื่อง จำนวนช่องใช้งาน เวลาเริ่มทำงาน ระยะเวลาของการเก็บข้อมูล และจำนวนข้อมูลที่เก็บได้จากหน้าจอ LCD นอกจากนี้เครื่องยังแสดงข้อมูลครั้งล่าสุดที่เก็บได้โดยกดสวิตช์ SW3 ค้างไว้
7. เมื่อเก็บข้อมูลได้ตามต้องการแล้ว ต่อเครื่องเข้ากับพอร์ตอนุกรมอีกครั้งและเลือก View Data จากหน้าจอหลัก จากนั้นกดปุ่ม Upload ข้อมูลจะถูกส่งจากเครื่องไปแสดงผลที่คอมพิวเตอร์ ดังรูป ข.4 (สำหรับโหมด Computer Based ในระหว่างที่เครื่องทำงานจะขึ้นหน้าจอนี้อยู่แล้ว) หน้าจอจะแสดงรายละเอียดต่างๆ ดังนี้
  - Setting Time แสดงเวลาที่ตั้งการทำงานเครื่อง
  - Launch Time แสดงเวลาเริ่มเก็บข้อมูลของเครื่อง
  - Last Update แสดงเวลาที่เก็บข้อมูลครั้งล่าสุด
  - ตารางแสดงข้อมูลที่เก็บได้ในแต่ละช่องสัญญาณ โดยมีวันที่และเวลากำกับ
  - แถบสถานะ (Status Bar) แสดงสถานะของโปรแกรม และรายละเอียดอื่นๆ
8. ผู้ใช้สามารถเก็บข้อมูลทั้งหมดลงในไฟล์ได้โดยกดปุ่ม File > Save as ที่แถบรายการ (Menu Bar)
9. เพื่อเริ่มการเก็บข้อมูลครั้งใหม่ กดปุ่ม Close เพื่อออกจากหน้าจอ View Data แล้วกดปุ่ม Disconnect ที่หน้าจอหลัก จากนั้นกดปุ่มรีเซ็ตที่ตัวเครื่อง (SW1)

Setting Time : 23:57:18      January 28, 2001  
 Launch Time : 23:57:30      January 28, 2001  
 Last Update : 00:00:25      January 29, 2001

Record	Channel1	Channel2	Channel3	Channel4
23:59:05	1.343	1.294	1.245	1.176
23:59:10	0.931	0.833	0.735	0.637
23:59:15	0.480	0.422	0.392	0.412
23:59:20	0.441	0.500	0.549	0.676
23:59:25	0.745	0.833	0.941	1.098
23:59:30	1.186	1.304	1.402	1.539
23:59:35	1.490	1.500	1.520	1.510
23:59:40	1.333	1.284	1.225	1.147
23:59:45	0.971	0.892	0.833	0.735
23:59:50	0.559	0.500	0.441	0.373
23:59:55	0.255	0.216	0.176	0.147
29Jan2001	0.088	0.078	0.069	0.108
00:00:05	0.069	0.078	0.078	0.147
00:00:10	0.108	0.098	0.108	0.176
00:00:15	0.127	0.118	0.108	0.167
00:00:20	0.127	0.118	0.108	0.167
00:00:25	0.147	0.147	0.167	0.245

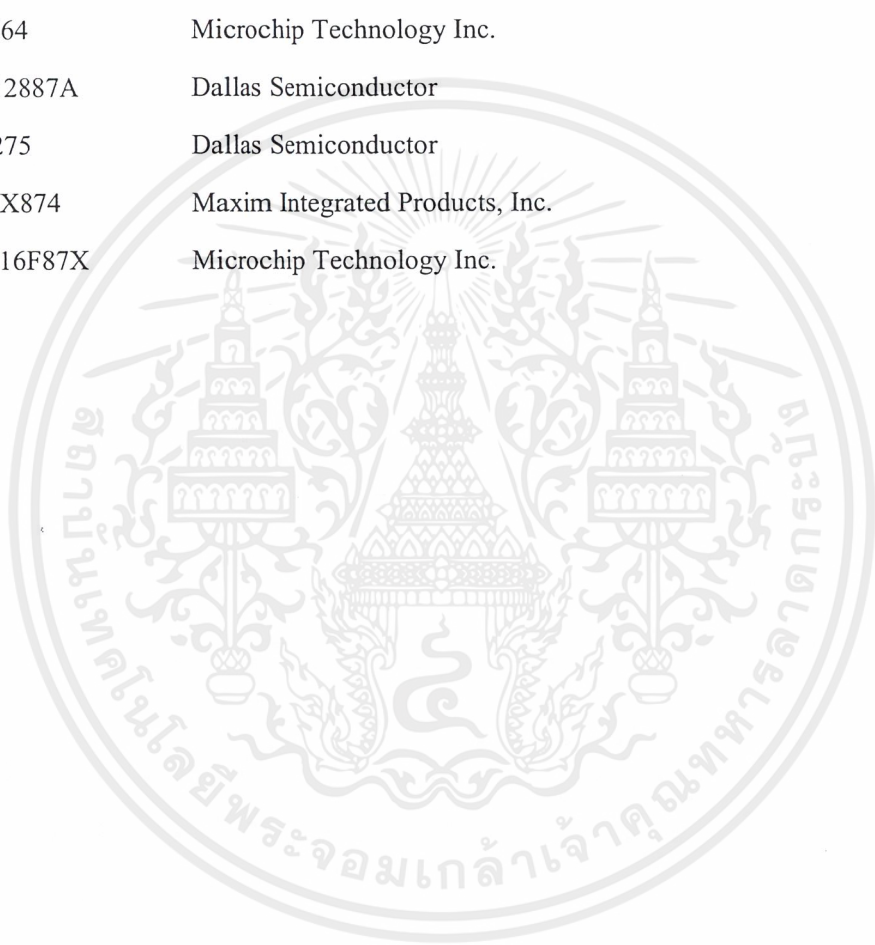
Monitoring...    Computer Base    4 CH    5 Seconds    36 Rec

รูปที่ ข.4 หน้าจอแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.  
ดาต้าชีทของไอซีที่สำคัญ

- 24C64 Microchip Technology Inc.
- DS12887A Dallas Semiconductor
- DS275 Dallas Semiconductor
- MAX874 Maxim Integrated Products, Inc.
- PIC16F87X Microchip Technology Inc.





# 24AA64/24LC64

## 64K I<sup>2</sup>C™ CMOS Serial EEPROM

### DEVICE SELECTION TABLE

Part Number	Vcc Range	Max Clock Frequency	Temp Ranges
24AA64	1.8-5.5V	400 kHz†	C
24LC64	2.5-5.5V	400 kHz	I, E

†100 kHz for Vcc < 2.5V.

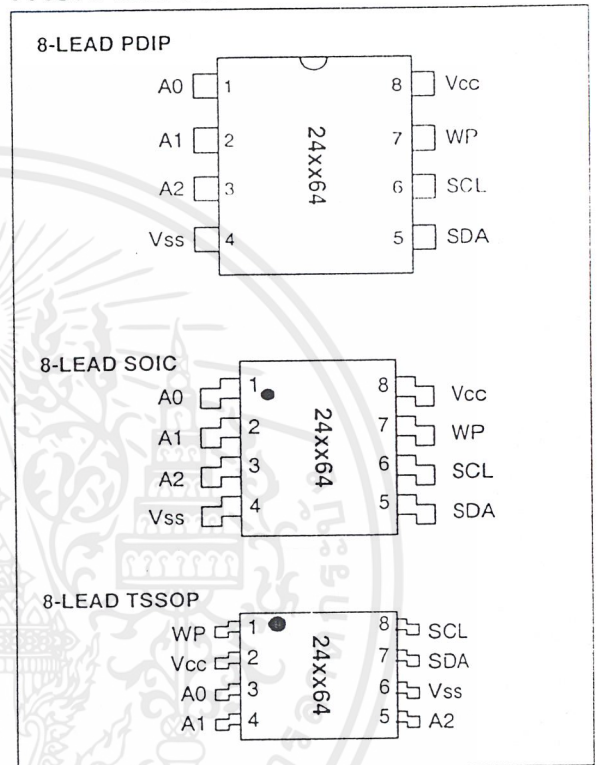
### FEATURES

- Low power CMOS technology
  - Maximum write current 3 mA at 5.5V
  - Maximum read current 400  $\mu$ A at 5.5V
  - Standby current 100 nA typical at 5.5V
- 2-wire serial interface bus, I<sup>2</sup>C compatible
- Cascadable for up to eight devices
- Self-timed ERASE/WRITE cycle
- 32-byte page or byte write modes available
- 5 ms max write cycle time
- Hardware write protect for entire array
- Output slope control to eliminate ground bounce
- Schmitt trigger inputs for noise suppression
- 1,000,000 erase/write cycles guaranteed
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP, SOIC (150 and 208 mil) and TSSOP packages
- Temperature ranges:
  - Commercial (C): 0°C to 70°C
  - Industrial (I): -40°C to +85°C
  - Automotive (E) -40°C to +125°C

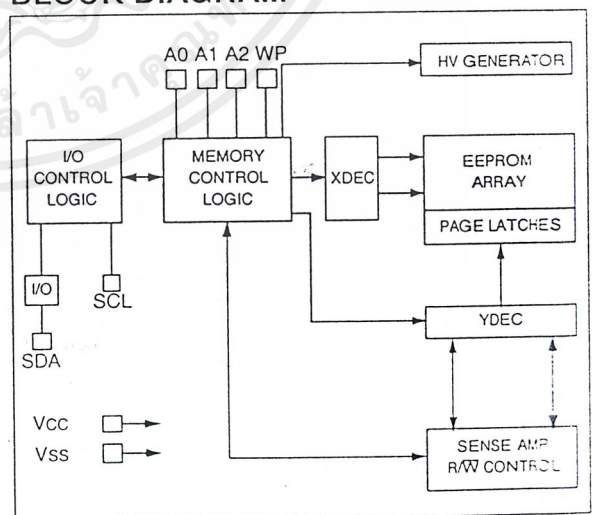
### DESCRIPTION

The Microchip Technology Inc. 24AA64/24LC64 (24xx64\*) is a 8K x 8 (64K bit) Serial Electrically Erasable PROM capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low power applications such as personal communications or data acquisition. This device also has a page-write capability of up to 32 bytes of data. This device is capable of both random and sequential reads up to the 64K boundary. Functional address lines allow up to eight devices on the same bus, for up to 512 Kbits address space. This device is available in the standard 8-pin plastic DIP, 8-pin SOIC (150 and 208 mil), and 8-pin TSSOP.

### PACKAGE TYPE



### BLOCK DIAGRAM



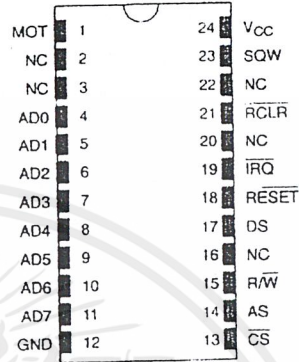
I<sup>2</sup>C is a trademark of Philips Corporation.

\*24xx64 is used in this document as a generic part number for the 24AA64/24LC64 devices.

## FEATURES

- Drop-in replacement for IBM AT computer clock/calendar
- Pin compatible with the MC146818B and DS1287A
- Totally nonvolatile with over 10 years of operation in the absence of power
- Self-contained subsystem includes lithium, quartz, and support circuitry
- Counts seconds, minutes, hours, days, day of the week, date, month, and year with leap year compensation valid up to 2100
- Binary or BCD representation of time, calendar, and alarm
- 12- or 24-hour clock with AM and PM in 12-hour mode
- Daylight Savings Time option
- Selectable between Motorola and Intel bus timing
- Multiplex bus for pin efficiency
- Interfaced with software as 128 RAM locations
  - 14 bytes of clock and control registers
  - 114 bytes of general purpose RAM
- Programmable square wave output signal
- Bus-compatible interrupt signals ( $\overline{\text{IRQ}}$ )
- Three interrupts are separately software-maskable and testable
  - Time-of-day alarm once/second to once/day
  - Periodic rates from 122  $\mu\text{s}$  to 500 ms
  - End of clock update cycle

## PIN ASSIGNMENT



24-PIN ENCAPSULATED PACKAGE

## PIN DESCRIPTION

- AD0-AD7 - Multiplexed Address/Data Bus
- NC - No Connection
- MOT - Bus Type Selection
- $\overline{\text{CS}}$  - Chip Select
- AS - Address Strobe
- $\overline{\text{R/W}}$  - Read/Write Input
- DS - Data Strobe
- $\overline{\text{RESET}}$  - Reset Input
- $\overline{\text{IRQ}}$  - Interrupt Request Output
- SQW - Square Wave Output
- V<sub>CC</sub> - +5 Volt Supply
- $\overline{\text{RCLR}}$  - RAM Clear
- GND - Ground

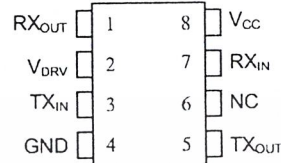
## DESCRIPTION

The DS12887A Real Time Clock plus RAM is designed to be a direct replacement for the DS1287A. The DS12887A is identical in form, fit, and function to the DS1287A, and has an additional 64 bytes of general purpose RAM. Access to this additional RAM space is determined by the logic level presented on AD6 during the address portion of an access cycle. The  $\overline{\text{RCLR}}$  pin is used to clear (set to logic 1) all 114 bytes of general purpose RAM but does not affect the RAM associated with the real time clock. In order

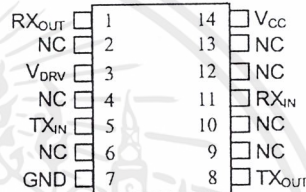
### FEATURES

- Low-power serial transmitter/receiver for battery-backed systems
- Transmitter steals power from receive signal line to save power
- Ultra-low static current, even when connected to RS-232-E port
- Variable transmitter level from +5 to +12 volts
- Compatible with RS-232-E signals
- Available in 8-pin, 150 mil wide SOIC package (DS275S)
- Low-power CMOS

### PIN ASSIGNMENT



DS275 8-Pin DIP (300-mil)  
DS275 8-Pin SOIC (150-mil)



DS275E 14-Pin TSSOP

### ORDERING INFORMATION

DS275	8-pin DIP
DS275S	8-pin SOIC
DS275E	14-pin TSSOP

### PIN DESCRIPTION

RX <sub>OUT</sub>	- RS-232 Receiver Output
V <sub>DRV</sub>	- Transmit driver +V
TX <sub>IN</sub>	- RS-232 Driver Input
GND	- System Ground (0V)
TX <sub>OUT</sub>	- RS-232 Driver Output
NC	- No Connection
RX <sub>IN</sub>	- RS-232 Receive Input
V <sub>CC</sub>	- System Logic Supply (+5V)

### DESCRIPTION

The DS275 Line-Powered RS-232 Transceiver Chip is a CMOS device that provides a low-cost, very low-power interface to RS-232 serial ports. The receiver input translates RS-232 signal levels to common CMOS/TTL levels. The transmitter employs a unique circuit which steals current from the receive RS-232 signal when that signal is in a negative state (marking). Since most serial communication ports remain in a negative state statically, using the receive signal for negative power greatly reduces the DS275's static power consumption. This feature is especially important for battery-powered systems such as laptop computers, remote sensors, and portable medical instruments. During an actual communication session, the DS275's transmitter will use system power (5-12 volts) for positive transitions while still employing the receive signal for negative transitions.

# MAXIM

## 10 $\mu$ A, Low-Dropout, Precision Voltage References

MAX872/MAX874

### General Description

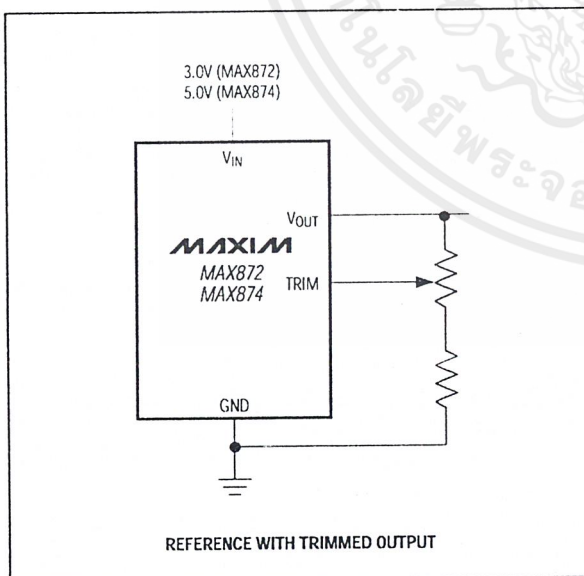
The MAX872/MAX874 precision 2.5V and 4.096V micro-power voltage references consume a maximum of only 10 $\mu$ A and operate from supply voltages up to 20V. The combination of ultra-low quiescent current and low 200mV dropout makes them ideal for battery-powered equipment. They source and sink up to 500 $\mu$ A with only 200mV input voltage headroom, which makes the 2.5V MAX872 ideal for use with a 3V supply and the 4.096V MAX874 ideal for use with a 5V supply.

Initial accuracy of 0.2% at +25°C ( $\pm$ 5mV for the MAX872,  $\pm$ 8mV for the MAX874) and low 40ppm/°C max drift make these references suitable for a wide range of precision applications.

### Applications

Hand-Held Instruments  
Battery-Operated Equipment  
Power Supplies

### Typical Operating Circuit



### Features

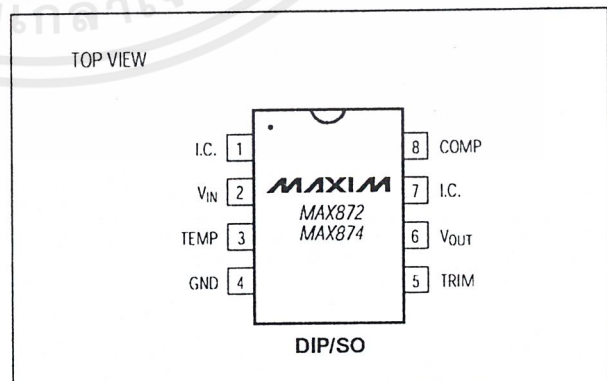
- ◆ Output Voltage
  - 2.500V  $\pm$ 0.2% (MAX872)
  - 4.096V  $\pm$ 0.2% (MAX874)
- ◆ Wide Operating Voltage Range
  - 2.7V to 20V (MAX872)
  - 4.3V to 20V (MAX874)
- ◆ 10 $\mu$ A Max Supply Current
- ◆ 40ppm/°C Max Drift Over Extended Temp. Range
- ◆ Line Regulation Over Temp.
  - 20 $\mu$ V/V (MAX872)
  - 75 $\mu$ V/V (MAX874)
- ◆ Load Regulation Over Temp.
  - 0.6mV/mA Max (MAX872)
  - 1.0mV/mA Max (MAX874)
- ◆  $\pm$ 500 $\mu$ A Sink/Source Current

### Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX872CPA	0°C to +70°C	8 Plastic DIP
MAX872GSA	0°C to +70°C	8 SO
MAX872C/D	0°C to +70°C	Dice*
MAX872EPA	-40°C to +85°C	8 Plastic DIP
MAX872ESA	-40°C to +85°C	8 SO
MAX874CPA	0°C to +70°C	8 Plastic DIP
MAX874CSA	0°C to +70°C	8 SO
MAX874C/D	0°C to +70°C	Dice*
MAX874EPA	-40°C to +85°C	8 Plastic DIP
MAX874ESA	-40°C to +85°C	8 SO

\* Dice are specified at +25°C only.

### Pin Configuration



MAXIM

Maxim Integrated Products 1

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MICROCHIP

# PIC16F87X

## 28/40-pin 8-Bit CMOS FLASH Microcontrollers

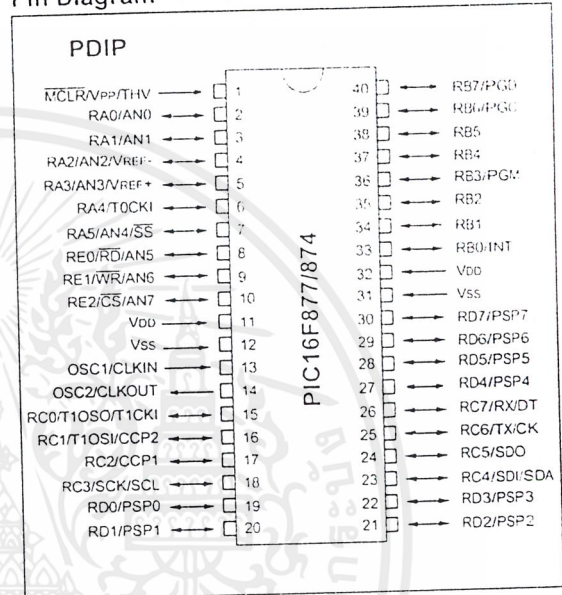
### Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

### Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory.  
Up to 368 x 8 bytes of Data Memory (RAM)  
Up to 256 x 8 bytes of EEPROM data memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram



### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณ อาจารย์ขนิษฐา แซ่ตั้ง อาจารย์ที่ปรึกษาที่ได้ให้แนวทางในการทำงาน คำแนะนำต่างๆ คุณพ่อและคุณแม่ที่สนับสนุนการทำงานหลายๆส่วน คุณกฤษฎา ใจเย็น เจ้าหน้าที่ บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ ที่ช่วยแนะนำการใช้งานไมโครคอนโทรลเลอร์ PIC เพื่อนๆที่ได้ช่วย ชื้อหาชิ้นส่วนอุปกรณ์ และอีกหลายท่านที่สนับสนุนผู้จัดทำ จนกระทั่งงานนี้สำเร็จลุล่วงไปได้

หากมีข้อผิดพลาดใดๆ ผู้จัดทำขออภัยมา ณ. ที่นี้ด้วย

วิทวัส วิทย์ชำนานกุล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

1. กิตติ ภัคดีวัฒนกุล. “Visual Basic 6 ฉบับโปรแกรมเมอร์”, พิมพ์ครั้งที่7. บริษัทเคทีพีคอมพิวเตอร์คอนซัลท์ จำกัด. 2543.
2. ชัยวัฒน์ ลิ้มพรจิตรวิไล. “เรียนรู้การเชื่อมต่อไอซีกับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม”, บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด. 2542.
3. ชีรวัฒน์ ประกอบผล. “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์”. สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), พิมพ์ครั้งที่3, 2542.
4. นพพน เลิศชูวงศา. “อุปกรณ์รวบรวมข้อมูลเคลื่อนที่”. วิทยานิพนธ์, ภาควิชาโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2541.
5. “24AA64/24LC64 64K I2C CMOS Serial EEPROM”, DS21189D. Microchip Technology Incorporated, 2000.
6. “DS12C887 Real Time Clock”, Dallas Semiconductor.
7. “DS275 Line-Powered RS232 Transceiver Chip”, Dallas Semiconductor.
8. “MAX872/MAX874 10 $\mu$ A, Low-Dropout, Precision Voltage References”. Maxim Integrated Products, 1997.
9. Pei An, Dr. “Stand-alone data logger”, page194-200. Electronics World, March 1998.
10. “PIC16F8X”, DS30430C. Microchip Technology Incorporated, 1998.
11. “PIC16F87X”, DS30292B. Microchip Technology Incorporated, 1999.
12. “The I2C-Bus Specification”. Phillips Semiconductors, 2000.