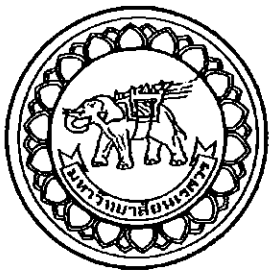
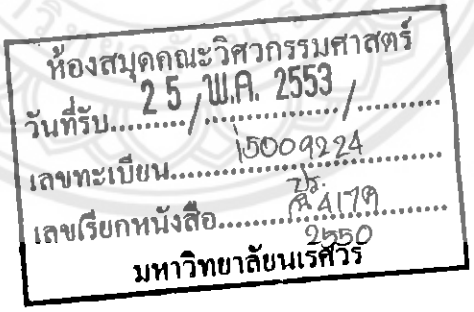


เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

Volt meter and recorder instrument

นายเฉลิมชัย ทิพหมึก รหัส 46380180

นายแสนเพชร์ มัคสัมพันธ์ รหัส 46380323



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า		
ผู้ดำเนินโครงงาน	นายเฉลิมชัย	ทิพหมึก	รหัส 46380180
	นายแสนเพชร์	มัคสัมพันธ์	รหัส 46380323
อาจารย์ที่ปรึกษา	ดร. สมพร เรืองสินชัยวานิช		
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2550		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.สมพร เรืองสินชัยวานิช)

.....กรรมการ

(ดร.ชัยรัตน์ พินทอง)

.....กรรมการ

(อ.ปิยคนัย ภาชนะพรรณ์)

หัวข้อโครงงาน	เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า
ผู้ดำเนินโครงงาน	นายเฉลิมชัย ทิพหมึก รหัส 46380180
	นายแสนเพชร์ มัคสัมพันธ์ รหัส 46380323
อาจารย์ที่ปรึกษา	ดร. สมพร เรืองสินชัยวานิช
สาขา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

บทคัดย่อ

เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า ใช้ไมโครคอนโทรลเลอร์ PIC 16F877 เป็นตัวรับสัญญาณทางไฟฟ้า คือแรงดันไฟฟ้า และทำการแปลงสัญญาณอะนาลอกเป็นสัญญาณดิจิตอล โดยใช้หน่วยประมวลผลของไมโครคอนโทรลเลอร์ในส่วนของ An analog-to-digital converter แล้วแสดงผล และบันทึกผลลงบนหน่วยความจำ 24LC512 โดยมีไอซี DS1307 เป็นตัวควบคุมในการกำหนดช่วงเวลาของการบันทึกค่าลงหน่วยความจำ และสามารถทำการดึงข้อมูลจากหน่วยความจำลงคอมพิวเตอร์ได้ โดยมีการเชื่อมต่อ ผ่านทางพอร์ตอนุกรม RS232 แล้วใช้โปรแกรม DELPHI เป็นตัวควบคุม

Project title	Volt meter and recorder instrument	
Name	Mr.Chalermchai Tipmuek	ID. 46380180
	Mr.Sanphet Maksamphan	ID. 46380323
Project advisor	Dr. Somporn Ruangsinchaiwanich	
Major	Electrical Engineering	
Department	Electrical and Computer Engineering	
Academic year	2007	

ABSTRACT

Volt meter and recorder instrument is a microcontroller No PIC 16F877 in use by receive an electricity signal are voltage and analog changing to digital signal by used an analog – to - digital converter to monitor and record to memory 24LC512 by IC No DS1307 control timing when have a record to memory and then can transfer all data to computer by connect on RS232 port by follow on DELPHI program.

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ในครั้งนี้ สำเร็จลุล่วงไปด้วยดีโดยได้รับความช่วยเหลือและให้คำแนะนำ จาก ดร. สมพร เรืองสิชัยวานิช ที่เป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ในครั้งนี้ และได้กรุณาให้แนวความคิด ช่วยชี้แนะแนวทางในการทำโครงงาน ตลอดจนกรุณาเอื้อเฟื้อเอกสารต่างๆ ที่เกี่ยวข้องกับโครงงานนี้ อีกทั้งยังช่วยแนะนำแหล่งข้อมูลในการค้นคว้าข้อมูลเพิ่มเติม ทำให้เป็นประโยชน์ต่อการทำโครงงานของผู้จัดทำเป็นอย่างมาก

ท้ายสุดขอขอบพระคุณ คุณพ่อ คุณแม่ ญาติพี่น้อง ที่คอยดูแล คอยเป็นกำลังใจและเป็นผู้สนับสนุนในด้านต่างๆ มาโดยตลอดในการทำปริญญานิพนธ์นี้ และขอขอบคุณบุคคลต่างๆที่ไม่ได้กล่าวถึงรวมถึงแหล่งข้อมูลที่เอื้อต่อการทำปริญญานิพนธ์ในครั้งนี้ด้วย

คณะผู้จัดทำ

นายแสนเพชร์ มัคสัมพันธ์

นายเฉลิมชัย ทิพหมึก

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 หลักการและเหตุผล 1

1.2 วัตถุประสงค์ 1

1.3 ขอบข่ายโครงงาน 2

1.4 กิจกรรมการดำเนินงาน 2

1.5 ผลที่คาดว่าจะได้รับ 2

1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎี

2.1 การเชื่อมต่อกับพอร์ตอนุกรม 4

2.2 สถาปัตยกรรมและโครงสร้างทางด้านฮาร์ดแวร์ของ PIC16F877 4

2.3 การจัดสรรพื้นที่หน่วยความจำ 6

2.4 รีจีสเตอร์ที่สำคัญต่างๆ ของ PIC16F877 9

2.5 ลักษณะทางฮาร์ดแวร์ 11

2.6 หลักการสื่อสารข้อมูล (Data Communication) 17

2.7 ไอซีฐานเวลาจริง DS1307 19

2.8 วงจร A/D Converter 22

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและสร้าง เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

3.1 ส่วนของคอมพิวเตอร์ 23

3.2 ส่วนของไมโครคอนโทรลเลอร์ 25

3.3 การต่อวงจรทั้งหมดของแหล่งจ่ายแรงดันไฟฟ้า (Hardware) 32

บทที่ 4 การใช้งานเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

4.1 การใช้งานตัวเครื่องวัด 33

4.2 การเริ่มใช้งานโปรแกรม 36

4.3 การวัดด้วยเครื่องวัดและบันทึกแรงดัน 38

บทที่ 5 บทสรุป

5.1 สรุปผล 46

5.2 ความสามารถของเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า 46

5.3 ประเมินผล 46

5.4 ปัญหา ข้อเสนอแนะ และแนวทางแก้ไข 47

5.5 แนวทางในการพัฒนาต่อไป 47

เอกสารอ้างอิง 48

ภาคผนวก 49

ภาคผนวก ก 50

ภาคผนวก ข 67

ประวัติผู้เขียนโครงงาน 71

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงค่าที่กำหนดในรีจีสเตอร์ 9

2.2 ข้อกำหนดมาตรฐาน RS-232 12

2.3 DB-9 ตัวผู้ทางด้านคอมพิวเตอร์ 13

2.4 DB-25 ตัวผู้ทางด้านคอมพิวเตอร์ 14

2.5 หน้าที่ของสัญญาณต่างๆ 14

2.6 การเซตค่าพอร์ตอนุกรม 16

2.7 แสดงรีจีสเตอร์ที่ใช้เก็บค่าต่างๆ 21

4.1 ตารางแสดงผลการวัดค่าและบันทึกค่าด้วยอัตราการบันทึก 1 ครั้งต่อนาที 39

4.2 ตารางแสดงผลการวัดค่าและบันทึกค่าด้วยอัตราการบันทึก 1 ครั้งต่อวินาที 42

สารบัญรูป

รูปที่ หน้า

2.1 ตำแหน่งขาสัญญาณต่างๆของ PIC 16F877 5

2.2 สถาปัตยกรรมภายในของ PIC 16F877 6

2.3 การจัดสรรพื้นที่หน่วยความจำของ PIC 16F877 7

2.4 การจัดวางพื้นที่ของหน่วยความจำข้อมูล 8

2.5 แสดงรายละเอียดของบิตที่ใช้ในการเลือกแบงก์ข้อมูล 9

2.6 โครงสร้างของรีจีสเตอร์ PC 11

2.7 แสดงลักษณะของหัวต่อแบบ DB-9 12

2.8 ลักษณะของหัวต่อแบบ DB-25 13

2.9 แสดงการเชื่อมต่อเพื่อสื่อสารทางเดียว 15

2.10 แสดงการเชื่อมต่อเพื่อการสื่อสารสองทาง 15

2.11 RS-232 Wave Form 16

2.12 แสดงส่วนประกอบของการสื่อสารข้อมูล 17

2.13 การส่งข้อมูลแบบขนาน (Parallel) 18

2.14 การส่งข้อมูลแบบอนุกรม (Serial) 19

2.15 แผนผังหน่วยความจำ 21

2.16 การทำงานของ A/D Converter 22

3.1 แสดงบล็อกไดอะแกรม 23

3.2 แสดงการทำงานของโปรแกรม DELPSI 24

3.3 แสดงบอร์ดไมโครคอนโทรลเลอร์ 25

3.4 ไมโครคอนโทรลเลอร์ PIC 16F877 26

3.5 ไอซี RS232 26

3.6 ไอซีเวลา 26

3.7 หน่วยความจำ 27

3.8 ไดโอด 27

3.9 ตัวเก็บประจุ 27

3.10 ตัวต้านทาน 28

3.11 แบตเตอรี่ 28

3.12 หม้อแปลง 28

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.13 วงจรแสดงการเชื่อมต่อของ PIC16F877	29
3.14 วงจรแสดงการเชื่อมต่อของ LED 7 Segment	30
3.15 วงจรแสดงการเชื่อมต่อของ RS232	30
3.16 วงจรส่วนของหน่วยความจำ 24LC512 และไอซีบอกเวลา DS1307	31
3.17 วงจรไฟเลี้ยงไอซีในไมโครคอนโทรลเลอร์	31
4.1 หน้าจอเครื่องจำลองของเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า	32
4.2 เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า	32
4.3 หน้าจอเครื่องวัดและบันทึกค่าแรงดันไฟฟ้าขณะไม่ใช้งาน	33
4.4 สายเชื่อมต่อกับคอมพิวเตอร์	33
4.5 แสดงการวัดเปรียบเทียบระหว่างเครื่องวัดกับโวลต์มิเตอร์ที่แรงดัน 220 Vac	38
4.6 แสดงหน้าจอของเครื่องวัดขณะใช้งานในการวัดและบันทึกค่า	38
4.7 แสดงหน้าจอของเครื่องวัดขณะใช้งานต่อกับคอมพิวเตอร์	39
4.8 กราฟแสดงค่าแรงดันจากการวัดและบันทึกค่า1 ครั้งต่อนาที (90 ครั้ง)	42
4.9 กราฟแสดงค่าแรงดันจากการวัดและบันทึกค่า 1 ครั้งต่อวินาที (612 ครั้ง)	45

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

Data Logger เป็นเครื่องมือที่ช่วยในการเก็บบันทึกข้อมูล ภายในช่วงของเวลา เริ่มแรกมนุษย์จะบันทึกข้อมูลโดยใช้เครื่องมือวัด วัดค่าทีละค่าตามระยะเวลาที่กำหนด และบันทึกค่าที่ได้ลงในกระดาษ และทำการพล็อตลงกระดาษกราฟซึ่งทำให้เสียเวลามาก ต่อมาได้มีการพัฒนาการบันทึกค่าโดยการใช้แขนกล รับค่าผ่านเซ็นเซอร์ และจดลงบนกระดาษ ซึ่งจะทำงานโดยอัตโนมัติ แต่มีข้อเสียคือ สิ้นเปลืองกระดาษที่ใช้บันทึกเนื่องจากเป็นข้อมูลที่ขึ้นกับเวลาทำให้กระดาษที่ใช้มีความยาวตามไปด้วย ปัจจุบันมนุษย์ได้หันมาใช้คอมพิวเตอร์ในชีวิตประจำวันมากยิ่งขึ้น ซึ่งคอมพิวเตอร์ในทุกวันนี้มีความสามารถหลายอย่าง ทั้งประมวลผล และเก็บข้อมูล ผู้ทำโครงงานจึงนำ Data Logger มาประยุกต์ใช้กับคอมพิวเตอร์ เพื่อเป็นการเพิ่มประสิทธิภาพของเครื่องคอมพิวเตอร์ให้มากยิ่งขึ้น โดยผู้ทำโครงงานจะทำการส่งข้อมูลที่เป็นสัญญาณทางไฟฟ้าส่งไปยังอุปกรณ์แปลงสัญญาณอนาลอกเป็นดิจิตอลบนไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะทำการส่งข้อมูลไปยังหน่วยความจำเสริมของไมโครคอนโทรลเลอร์ และสามารถดึงข้อมูลจากหน่วยความจำเข้าสู่คอมพิวเตอร์ผ่านทางพอร์ตอนุกรม เพื่อทำการเก็บบันทึกข้อมูลลงคอมพิวเตอร์

1.2 วัตถุประสงค์

1.2.1 ออกแบบอุปกรณ์เก็บข้อมูลโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวรับ-ส่ง และจัดเก็บข้อมูล

1.2.2 ศึกษาการเก็บข้อมูลในระยะเวลาที่กำหนด

1.2.3 ศึกษาการรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ และแสดงผลทางจอคอมพิวเตอร์

1.2.4 ศึกษาการทำงานของ DELPHI เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์

1.3 ขอบข่ายโครงงาน

โครงงานนี้เป็นโครงงานที่เกี่ยวกับการแสดงผลแล้วเก็บบันทึกข้อมูล ลงหน่วยความจำ และส่งต่อข้อมูลไปยังคอมพิวเตอร์ได้ ซึ่งมีหัวข้อหลักที่จะทำการศึกษาเพื่อสร้างชิ้นงาน ดังนี้

1.3.1 สามารถบันทึกค่าแรงดันไฟฟ้าลงหน่วยความจำของไมโครคอนโทรลเลอร์

1.3.2 ศึกษาการทำงานร่วมกันระหว่าง คอมพิวเตอร์กับไมโครคอนโทรลเลอร์

1.3.3 สามารถเก็บบันทึกข้อมูลที่บันทึกลงสู่คอมพิวเตอร์ได้

1.3.4 เขียนโปรแกรมที่ใช้ในการรับส่งข้อมูลที่เป็นค่าแรงดันไฟฟ้า

1.4 กิจกรรมการดำเนินงาน

กิจกรรม	ปี 2550					
	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.
1. รวบรวมข้อมูลและเอกสาร						
2. ออกแบบเครื่องเก็บข้อมูล						
3. ออกแบบโปรแกรมคอมพิวเตอร์ที่ใช้ดึงข้อมูล						
4. สร้างอุปกรณ์และเขียนโปรแกรม						
5. ทดสอบ และแก้ไขชิ้นงาน						
6. สรุปผลการทดลอง และวิจารณ์						

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 สามารถบันทึกข้อมูลลงหน่วยความจำได้

1.5.2 สามารถส่งข้อมูลที่บันทึกเข้าไปเก็บในคอมพิวเตอร์ได้

1.5.3 สามารถใช้ DELPSI ควบคุมการทำงานของไมโครคอนโทรลเลอร์

1.5.4 นำความรู้ทางด้านไมโครคอนโทรลเลอร์ที่ได้รับไปประยุกต์ใช้กับงานได้

1.5.5 สามารถนำอุปกรณ์ไปประยุกต์ใช้ในระบบแรงดันไฟฟ้าสูงได้

1.6 งบประมาณ

1.6.1 ค่าอุปกรณ์อิเล็กทรอนิกส์	800 บาท
1.6.2 กระดาษ และถ่ายเอกสาร	200 บาท
1.6.4 ค่าอุปกรณ์คอมพิวเตอร์	150 บาท
1.6.5 ค่าจัดทำรูปเล่ม	600 บาท
1.6.6 หนังสือประกอบการทำโครงงาน	250 บาท
รวมเป็นเงิน	2,000 บาท (สองพันบาทถ้วน)

บทที่ 2
หลักการและทฤษฎี

2.1 การเชื่อมต่อกับพอร์ตอนุกรม

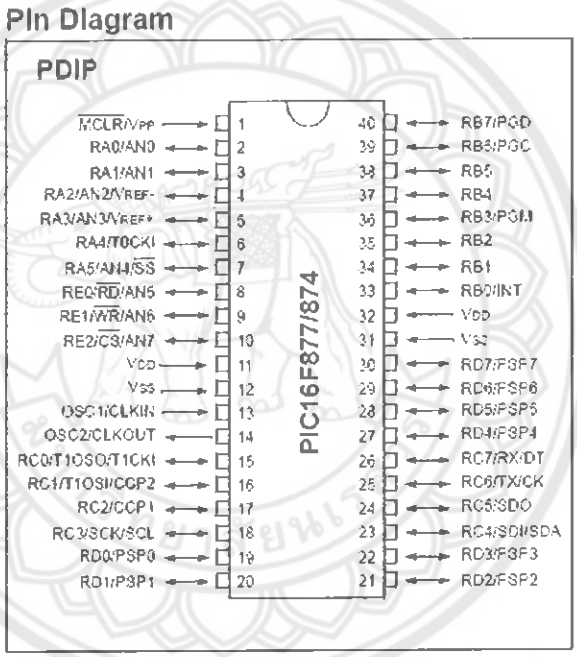
เป็นวิธีการอินเตอร์เฟสของคอมพิวเตอร์วิธีหนึ่งเพื่อทำการเชื่อมต่อกับอุปกรณ์อื่นๆ ซึ่งวิธีนี้จะเป็นการติดต่อแบบอนุกรมผ่านทางพอร์ตอนุกรม ซึ่งการเชื่อมต่อแบบนี้จะมีความเร็วในการเชื่อมต่อช้ากว่าพอร์ตขนาน แต่อุปกรณ์ที่ใช้พอร์ตขนานจะมีราคาแพงกว่า และจะต้องใช้สายสัญญาณมากกว่า การส่งข้อมูลทางพอร์ตอนุกรมนั้นในเครื่องคอมพิวเตอร์จะใช้มาตรฐาน RS-232 ซึ่งเป็นมาตรฐานการส่งข้อมูลระยะทางไม่เกิน 15 เมตร ส่วนอุปกรณ์ที่ใช้ในการควบคุมการทำงานนั้น ยังคงมีการพัฒนาขึ้นมาใหม่อย่างสม่ำเสมอ ดังนั้นจึงต้องเข้าใจมาตรฐานที่ใช้อยู่ตลอดเวลา เพื่อนำไปประยุกต์และใช้งานตามความต้องการได้

2.2 สถาปัตยกรรมและโครงสร้างทางด้านฮาร์ดแวร์ ของ PIC16F877

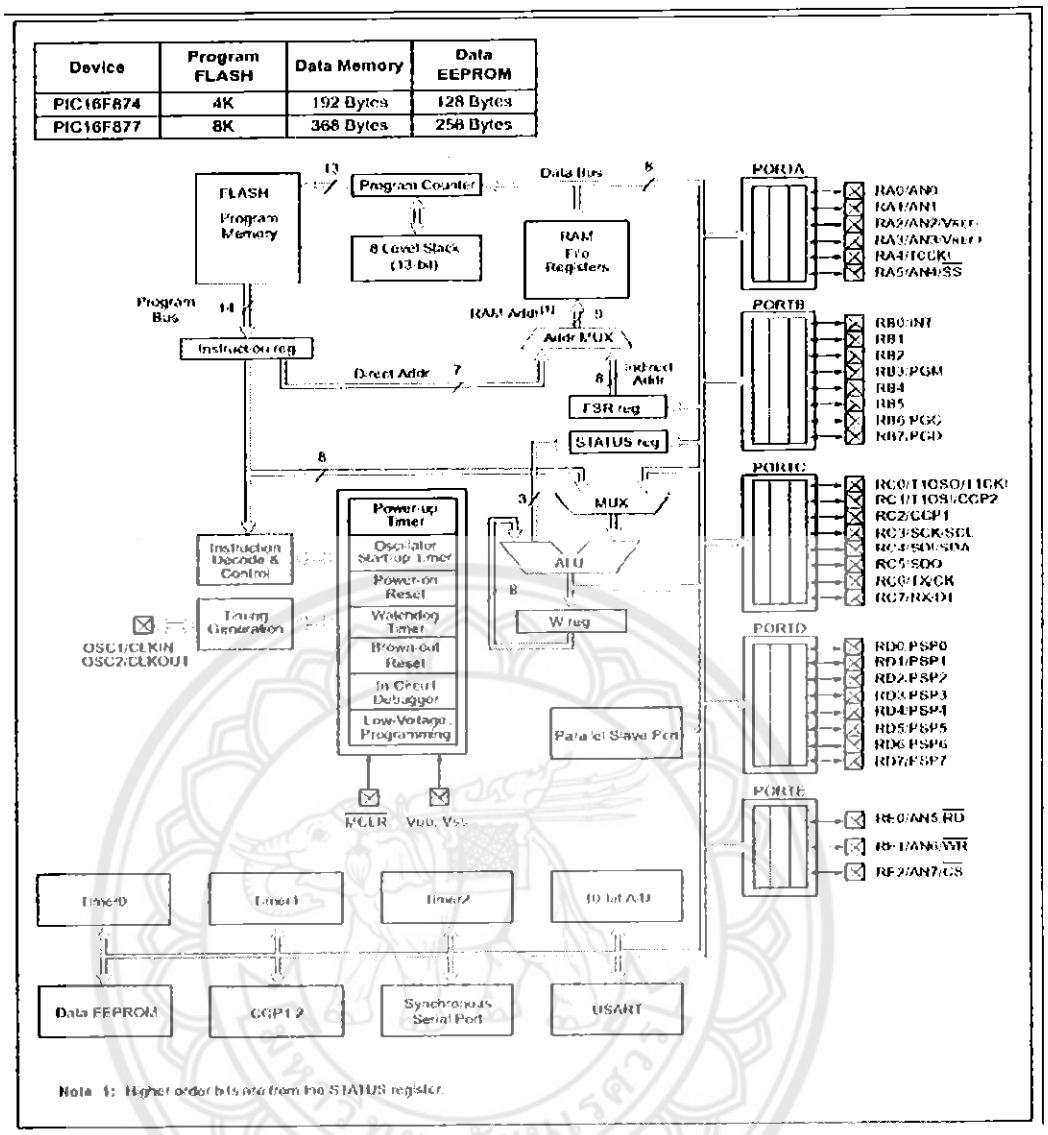
PIC16F877 เป็น CPU ของบริษัท Microchip Technology ซึ่งเป็นผู้ผลิต CPU ตระกูล PIC โดยในเบอร์ 16F877 เป็นเบอร์ที่มีความสามารถสูง ประกอบไปด้วยฟังก์ชันการทำงาน สรุปคุณสมบัติดังนี้

- มี 35 Instruction คำสั่ง
- ในการปฏิบัติงานคำสั่งต่างๆจะใช้ 1 Cycle และใช้ 2 Cycle ในคำสั่งที่เป็นการกระโดด
- ความถี่สูงสุดที่ทำงานได้คือ 20 MHz (16F877-20/P)
- การทำงานจะเป็นลักษณะ Pipeline ทำให้มีการทำงานที่เร็วขึ้น
- หน่วยความจำโปรแกรม FLASH Program Memory มีขนาด 8k (14-Bit Words)
- หน่วยความจำข้อมูล (RAM) 368 Bytes
- หน่วยความจำข้อมูล (EEPROM) 256 Bytes
- สามารถตอบสนองการอินเตอร์รัพท์ได้ถึง 14 แหล่ง
- มี STACK 8 ระดับ
- มีเพาเวอร์ออนรีเซต (POR), เพาเวอร์อัพไทเมอร์ (PWRT) และ Oscillator Start-Up Timer
- Watchdog Timer
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- มีโหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของ สัญญาณนาฬิกาได้หลายโหมด
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้

- มีฟังก์ชันการโปรแกรมแบบ ICSP (In-Circuit Serial Programming)
- ทำงานที่ไฟเลี้ยง 2.0V ถึง 5.5V
- กระแสทั้งซิงก์ และซอร์สของพอร์ตคือ 25mA
- มี Timer/Counter จำนวน 3 ตัว คือ Timer0, Timer1 และ Timer2
- มีโมดูล Capture/Compare/PWM จำนวน 2 ชุด
- มี Analog to Digital Converter ความละเอียด 10 บิต 8 แชนเนล ภายในตัว
- มีโมดูลการสื่อสาร USART
- มีโมดูลตรวจจับระดับไฟเลี้ยง Brown – out reset (BOR)
- มีพอร์ต I/O 5 พอร์ตประกอบด้วย A, B, C, D และ E แต่ละพอร์ตจะมีจำนวนบิตไม่เท่ากันซึ่งรวมแล้วจะมี I/O จำนวน 33 บิต



รูปที่ 2.1 ตำแหน่งขาสัญญาณต่าง ๆ ของ PIC16F877



รูปที่ 2.2 สถาปัตยกรรมภายในของ PIC16F877

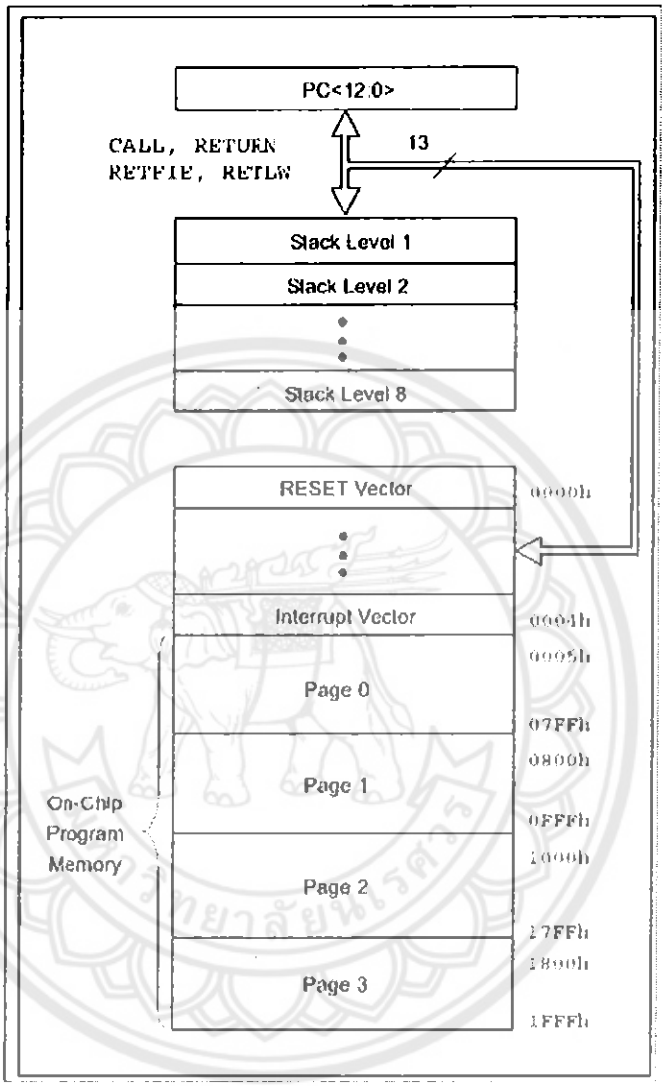
2.3 การจัดสรรพื้นที่หน่วยความจำและรีจิสเตอร์ต่างๆ

การจัดหน่วยความจำของ PIC16F877 แบ่งออกเป็น 3 ส่วนคือ หน่วยความจำโปรแกรม, หน่วยความจำข้อมูล (RAM) และหน่วยความจำข้อมูลที่เป็น EEPROM ซึ่งรายละเอียดในแต่ละส่วนดังนี้

2.3.1 หน่วยความจำโปรแกรม

PIC16F877 มี Program Counter ขนาด 13 บิตซึ่งสามารถอ้างถึงตำแหน่งข้อมูลได้ถึง 8 กิโลเวิร์ด โดยจะมีตำแหน่ง Reset Vector ที่ 0000h และ Interrupt Vector ที่ 0004h ดังนั้นในการเขียนโปรแกรมจึงควรสงวนพื้นที่ส่วนนี้ไว้สำหรับการใช้งานอินเตอร์รัพท์ จากรูปที่ 2.3 จะเห็นว่ามีพื้นที่

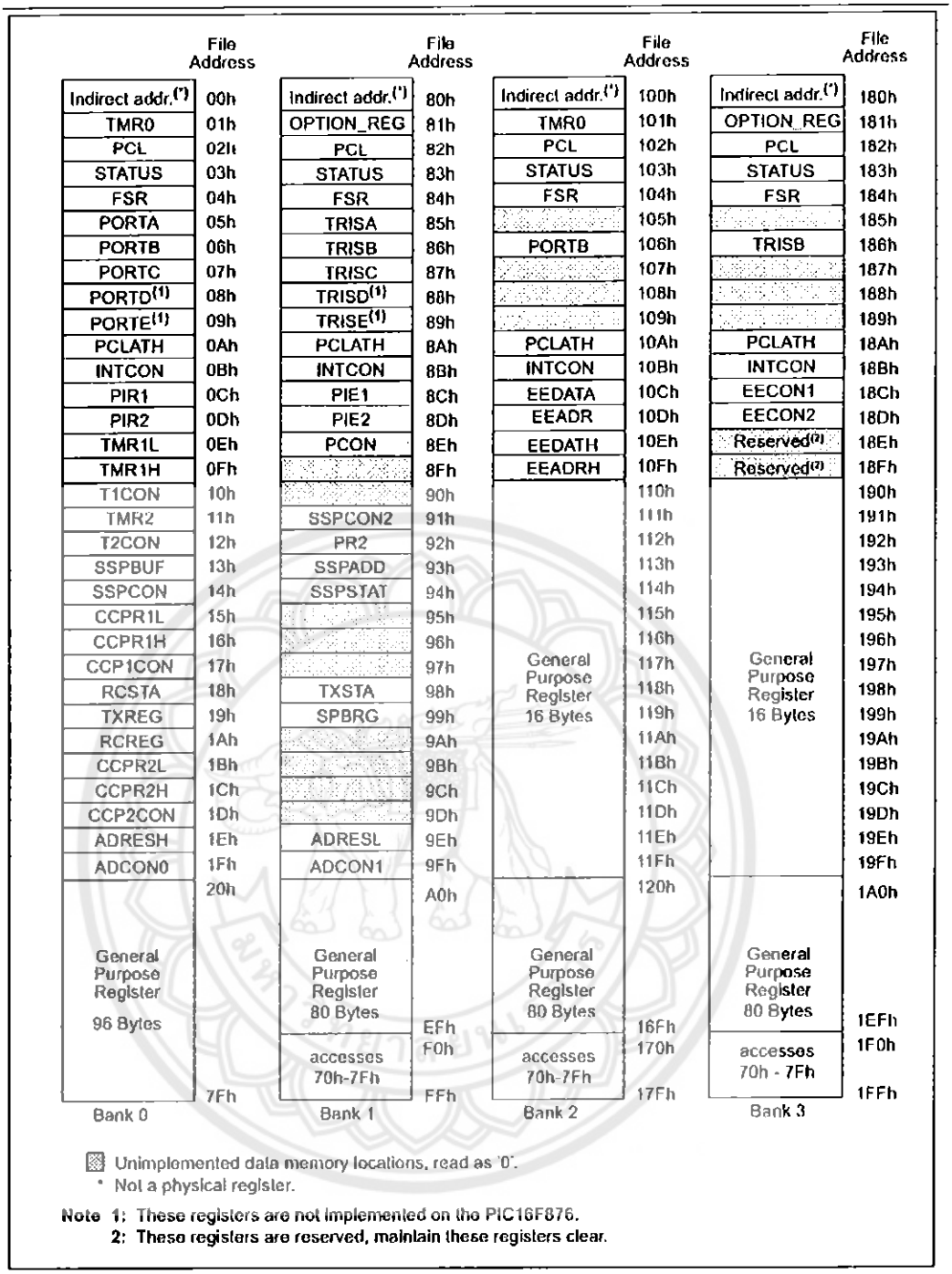
ของ Stack 8 ระดับ และ หน่วยความจำโปรแกรมแบ่งออกเป็น 4 Page (8 kwords) ซึ่งพื้นที่ในส่วนนี้มีไว้สำหรับเก็บข้อมูลคำสั่งทั้งหมดโดยโครงสร้างจะเป็นแบบแฟลช (flash memory) ทำให้ลบและเขียนใหม่ได้หลายครั้ง



รูปที่ 2.3 การจัดสรรพื้นที่หน่วยความจำของ PIC 16F877

2.3.2 หน่วยความจำข้อมูล

ใน PIC16F877 หน่วยความจำข้อมูลจะแบ่งออกเป็นพื้นที่ของ RAM หน่วยความจำใช้งานทั่วไป (General Purpose Register) ขนาด 368 Bytes และ พื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Registers) ในการจัดวางพื้นที่จะแบ่งออกเป็น 4 แบงค์ ตั้งแต่แอดเดรส 00h ถึง 1FFh ดังรูปที่ 2.4



รูปที่ 2.4 การจัดวางพื้นที่ของหน่วยความจำข้อมูล

ซึ่งจากรูปที่ 2.4 ในการเข้าถึงข้อมูลในแต่ละส่วน หรือแต่ละแบงก์ สามารถทำได้โดยการกำหนดค่าในรีจิสเตอร์ STATUS ในบิตที่ 5 และ 6 (RP0, RP1) ซึ่งมีความหมายดังนี้

RP1	RP0	Bank Select
0	0	Bank0 : 00h – 7Fh
0	1	Bank1 : 80h – FFh
1	0	Bank2 : 100h – 17Fh
1	1	Bank3 : 180h – 1FFh

ตารางที่ 2.1 แสดงการกำหนดค่าในรีจิสเตอร์

2.3.3 หน่วยความจำข้อมูล EEPROM

PIC16F877 มีหน่วยความจำแบบ EEPROM จำนวน 256 ไบต์ โดยสามารถอ่านและเขียนในขณะทำงานปกติได้แต่ต้องไม่มีการ Enable Code protect bit โดยการเข้าถึงนั้นจะต้องทำผ่านรีจิสเตอร์พิเศษ (Special Function Register) ซึ่งต้องใช้ถึง 4 ตัวดังนี้

EECON1: ควบคุมการเข้าถึงหน่วยความจำ

EECON2: จัดลำดับการเขียนข้อมูล

EEDATA: เป็นบัฟเฟอร์ใช้เก็บข้อมูล 8 บิต สำหรับการอ่านและเขียน

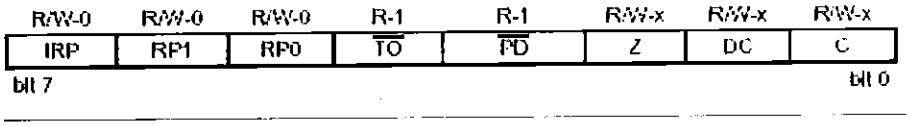
EEADR: รีจิสเตอร์ที่เก็บแอดเดรส 00h – FFh (256 ไบต์)

2.4 รีจิสเตอร์ที่สำคัญต่างๆ ของ PIC16F877

รีจิสเตอร์จัดได้ว่าเป็นส่วนประกอบที่มีความสำคัญต่อการพัฒนาโปรแกรมเป็นอย่างมาก ซึ่งใน PIC16F877 มีรีจิสเตอร์ต่างๆ จำนวนมาก รีจิสเตอร์ที่ใช้งานหลักๆ มีดังนี้

2.4.1 รีจิสเตอร์ STATUS

เป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลสถานะ การทำงานของไมโครคอนโทรลเลอร์ ไม่ว่าจะเป็นแฟลก สถานะต่างๆ ที่ใช้บอกผลลัพธ์การทำงานของส่วนต่างๆ เช่น การคำนวณทางคณิตศาสตร์ การเลือกแบงก์ข้อมูล ซึ่งมีรายละเอียดในแต่ละบิตดังนี้



รูปที่ 2.5 แสดงรายละเอียดของบิตที่ใช้ในการเลือกแบงก์ข้อมูล

IRP0 (Indirect Register Bank Select bit – บิต 7) ใช้เลือกแบงก์ในหน่วยความจำชนิด RAM ในกรณีที่ใช้การอ้างถึงข้อมูลแบบทางอ้อม (indirect addressing mode)

"0" : เลือกแบงก์ 0 และ 1

"1" : เลือกแบงก์ 2 และ 3

RP1, RP0 (Register Bank Select) ใช้เลือกแบงก์ของหน่วยความจำข้อมูลแรม และรีจิสเตอร์ไฟล์เมื่อเป็นการเข้าถึงข้อมูลแบบโดยตรง (direct addressing mode)

"00" เลือกแบงก์ 0

"01" เลือกแบงก์ 1

"10" เลือกแบงก์ 2

"11" เลือกแบงก์ 3

TO (Time-out bit) เป็นบิตที่แสดงการเกิดไทม์เอาต์ของวอชด็อกไทเมอร์แอกทีฟลอจิก "0"

PD (Power-down) บิตแสดงการทำงานในโหมดสลีป (Sleep mode) คือ เมื่อเข้าสู่ Sleep mode บิตนี้จะกลายเป็น "0"

Z (Zero bit) เป็นบิตแสดงสถานะการทำงานทางคณิตศาสตร์

"0" เมื่อผลลัพธ์ทางคณิตศาสตร์ลอจิกไม่เป็นศูนย์

"1" เมื่อผลลัพธ์ทางคณิตศาสตร์ลอจิกเป็นศูนย์

DC (Digit carry/borrow) เป็นบิตทดหรือยืมระหว่างหลัก

"0" เมื่อไม่เกิดการทดจากบิต 3 ไปบิต 4 หรือ หากเกิดการยืมจากบิต 4 มาบิต 3

"1" เมื่อเกิดการทดจากบิต 3 ไปบิต 4 หรือ หากไม่เกิดการยืมจากบิต 4 มาบิต 3

C (Carry/borrow) บิตทดหรือยืม ใช้แสดงสถานะการทด หรือการยืมทางคณิตศาสตร์

"0" เมื่อไม่มีการทดบิต 7 (MSB) หรือ เกิดการยืมจากบิต 7 (MSB)

"1" เมื่อมีการทดจากบิต 7 (MSB) หรือ ไม่เกิดการยืมค่าของบิต 7 (MSB)

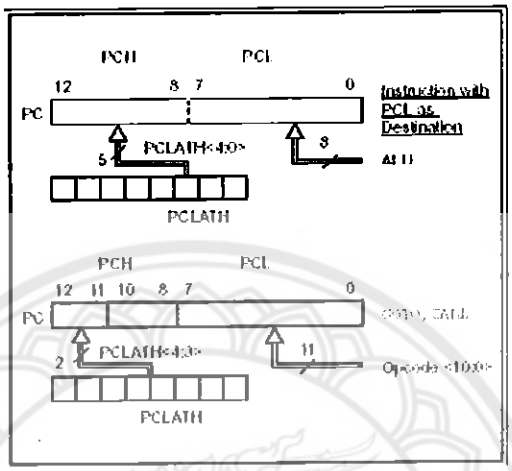
2.4.2 รีจิสเตอร์ W (Working register)

เป็นรีจิสเตอร์ที่มีความสำคัญมากที่สุดตัวหนึ่งของไมโครคอนโทรลเลอร์ PIC ซึ่งรีจิสเตอร์ W เป็น รีจิสเตอร์ขนาด 8 บิต จะทำหน้าที่เหมือนเป็นรีจิสเตอร์ แอกคิวมูลเลเตอร์ ในการทำคำสั่งทางคณิตศาสตร์หรือการโอนย้ายข้อมูล จะต้องผ่านรีจิสเตอร์ W ทั้งสิ้น

2.4.3 โปรแกรมเคาน์เตอร์ (PC)

เป็นรีจิสเตอร์ที่สำคัญอีกตัวหนึ่ง ซึ่งทำหน้าที่เก็บตำแหน่งแอดเดรสที่จะให้ CPU ไปทำงาน โดยรีจิสเตอร์ PC ของ PIC16F877 จะมีขนาด 13 บิต โดย 8 บิตล่าง PC< 7:0 > จะอยู่ที่รีจิสเตอร์ PCL สามารถอ่านและเขียนได้เหมือนรีจิสเตอร์ทั่วไป ส่วน PC<12 :8 > จะไม่สามารถเข้าถึงได้ตามปกติ การเข้าถึงจะกระทำ ผ่านรีจิสเตอร์ PCLATH < 4 :0 > และเมื่อเกิดการรีเซต PCLATH จะ

เคลียร์ สถานะเป็น "0" ในการใช้คำสั่งในการกระโดดเช่น CALL หรือ GOTO จะเป็นการนำค่าแอดเดรสมาใส่ในรีจิสเตอร์ PC <10:0 > เพียง 11 บิตเท่านั้นส่วน PC<12:11> จะไม่เปลี่ยนแปลง ทำให้การใช้คำสั่ง CALL หรือ GOTO ไปได้ไกลเพียง 2 Kwords (2048 ตำแหน่ง) แต่หน่วยความจำทั้งหมดมี 8 Kwords แบ่งเป็น 4 Page ฉะนั้นในการกระโดดข้าม Page จะต้องมีการกำหนดค่าให้กับ PC<12:11> ด้วยโดยผ่านทางรีจิสเตอร์ PCLATH



รูปที่ 2.6 โครงสร้างของรีจิสเตอร์ PC

2.5 ลักษณะทางฮาร์ดแวร์

เพื่อที่จะให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันกว้างขวางที่สุดคือ RS-232 ถูกประกาศในปี 1969 โดย Electronic Industries Association (EIA) ซึ่งเป็นการกำหนดมาตรฐานต่างๆเช่น ลักษณะการเชื่อมต่อ สัญญาณทางไฟฟ้าที่ใช้เพื่อป้องกันไม่ให้อุปกรณ์ส่งข้อมูลภายในเส้นเดียวกัน อุปกรณ์สื่อสารจึงจำแนกออกเป็น 2 ประเภท คือ

DTE (Data Terminal Equipment) คืออุปกรณ์สำหรับส่งข้อมูล เช่น คอมพิวเตอร์

DCE (Data Communication Equipment) คืออุปกรณ์สำหรับการติดต่อ เช่น โมเด็ม

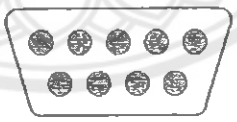
ตามมาตรฐาน RS-232 อุปกรณ์ DTE ควรใช้หัวต่อตัวผู้ และอุปกรณ์ DCE ควรใช้หัวต่อตัวเมียซึ่งหัวต่อที่นิยมใช้กันจะเป็นชนิด D แบบ 9 ขา และ 25 ขา (บางครั้งเรียก DB-25 และ DB-9)

Specifications	RS-232
Mode of Operation	Single-Ended
Total Number of Drivers and Receivers on One Line	1Driver and 1 Receiver
Maximum Cable Length	50FT
Maximum Data Rate	20KBPS
Maximum Driver Output Voltage	+/-25V
Driver Output Signal Level (Loaded Min)	+/-5V to +/-15V
Driver Output Signal Level (Loaded Max)	+/-25V
Driver load Impedance (Ω)	3k to 7k
Max. Driver Current in High Z Stage (Power On)	N/A
Max. Driver Current in High Z Stage (Power Off)	+/-6mA @ +/-2V
Slew Rate (Max.)	30V/ μs
Receiver Input Voltage Range	+/-15V
Receiver Input Sensitivity	+/-3V
Receiver Input Resistance (Ohms)	3k to 7k

ตารางที่ 2.2 ข้อกำหนดของมาตรฐาน RS-232

2.5.1 ลักษณะของหัวต่อตามมาตรฐาน RS-232C (DB-25 & DB-9)

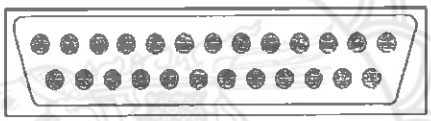
พอร์ตอนุกรมมีขั้วต่อ 2 แบบ คือ แบบ D 25 ขา และแบบ D 9 ขา ซึ่งทั้ง 2 แบบจะเป็นชนิดตัวผู้ทางด้านของคอมพิวเตอร์ ดังนั้นอุปกรณ์ที่จะนำมาต่อกับคอมพิวเตอร์จึงต้องใช้ขั้วต่อชนิดตัวเมีย



รูปที่ 2.7 แสดงลักษณะของหัวต่อแบบ DB-9

Pin	Signal	Direction	Description
1	CD	←	Carrier Detect
2	RxD	←	Receive Data
3	TxD	→	Transmit Data
4	DTR	→	Data Terminal Ready
5	GND	-	System Ground
6	DSR	←	Data Set Ready
7	RTS	→	Require to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

ตารางที่ 2.3 DB-9 ตัวผู้ทางด้านคอมพิวเตอร์



รูปที่ 2.8 ลักษณะของหัวต่อแบบ DB-25

Pin	Signal	Direction	Description
1	SHIELD	-	Shield Ground
2	TxD	→	Transmit Data
3	RxD	←	Receive Data
4	RTS	→	Require to Send
5	CTS	←	Clear to Send
6	DSR	←	Data Set Ready
7	GND	-	System Ground
8	CD	←	Carrier Detect
9	N/C	→	+12 Volt
10	N/C	→	-12 Volt
11	N/C	-	Reserved
12	N/C	←	Secondary Line Detect

Pin	Signal	Direction	Description
13	N/C	←	Secondary Clear to Send
14	N/C	→	Secondary Transmit Data
15	N/C	→	DCE Transmit Signal Timing
16	N/C	←	Secondary Receive Data
17	N/C	←	Receiver Signal Timing
18	N/C	-	Not Used
19	N/C	→	Secondary Request to Send
20	DTR	→	Data Terminal Ready
21	N/C	←	Signal Quality Detector
22	RI	←	Ring Indicator
23	N/C	→	Data Rate Selector
24	N/C	→	DTE Transmitter Signal Timing
25	N/C	-	Reserved

ตารางที่ 2.4 DB-25 ตัวผู้ทางด้านคอมพิวเตอร์

สัญญาณ	ชื่อ	ตัวส่ง	การทำงาน
TxD	Transmit Data	DTE	ส่งข้อมูลทีละบิตจาก DTE ไปยัง DCE
RxD	Receive Data	DCE	รับข้อมูลทีละบิตจาก DCE ไปยัง DTE
CTS	Clear to Send	DCE	ตรวจจับสัญญาณจาก DCE ว่าพร้อมจะรับข้อมูลจาก DTE
CD	Carrier Detect	DCE	เมื่อไรที่ตรวจจับสัญญาณเจอที่ปลายทางของสัญญาณจะทำให้สายสัญญาณ Active
DSR	Data Set Ready	DCE	บอก DTE ว่า DCE พร้อมที่จะทำงานแล้ว
DTR	Data Terminal Ready	DTE	สัญญาณจาก DTE บอกให้ DCE เตรียมพร้อม
RTS	Require to Send	DTE	สัญญาณจาก DTE บอกให้ DCE เตรียมพร้อมที่จะรับข้อมูลแล้ว
RI	Ring Indicator	DCE	ตรวจจับสัญญาณของสายโทรศัพท์

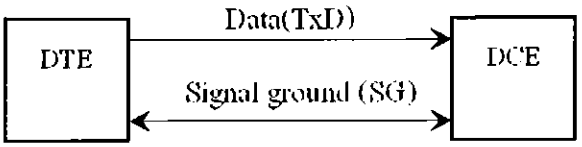
ตารางที่ 2.5 หน้าที่ของสัญญาณต่างๆ

2.5.2 การสื่อสารทางเดียว

สัญญาณหลักที่ใช้สำหรับการสื่อสารมีอยู่ 2 สัญญาณ ได้แก่

1. สายสัญญาณสำหรับข้อมูลจาก DTE → DCE

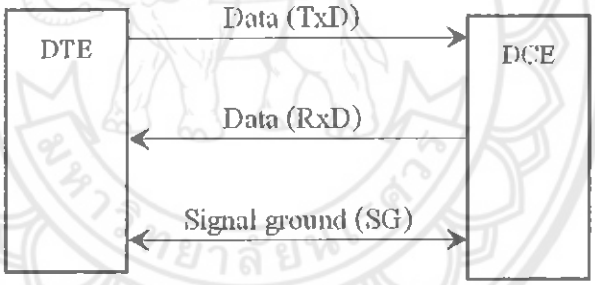
2. สายสัญญาณสำหรับกราวด์ (SG) ซึ่งเป็นจุดอ้างอิงร่วมสำหรับขั้วและแรงดันไฟฟ้าของสายอื่น



รูปที่ 2.9 แสดงการเชื่อมต่อเพื่อสื่อสารทางเดียว

2.5.3 การสื่อสารสองทาง

ในกรณีที่ข้อมูลถูกส่งผ่านในสองทิศทาง โดยเฉพาะเมื่อคอมพิวเตอร์สองตัวสื่อสารกัน จำนวนสายน้อยที่สุดในการสื่อสารสองทางคือ 3 เส้น ได้แก่ สายข้อมูลแต่ละทิศทาง และสายสัญญาณกราวด์ดังรูปที่ 2.12



รูปที่ 2.10 แสดงการเชื่อมต่อเพื่อการสื่อสารสองทาง

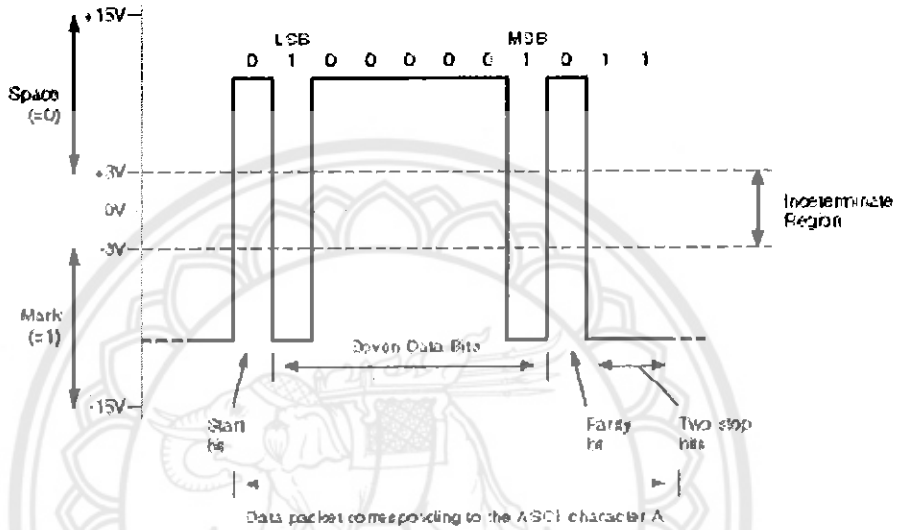
2.5.4 สัญญาณทางไฟฟ้า

ตามมาตรฐาน RS-232 ได้กำหนดลักษณะของสัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่อแบบอนุกรมมี 2 ลักษณะ คือ สเปสหมายถึง ลอจิก '0' และมาร์ค หมายถึง ลอจิก '1' โดย

สเปส จะเป็นแรงดันไฟฟ้าบวก	เอาต์พุตอยู่ในช่วง +5 ถึง +15 โวลต์
	อินพุตอยู่ในช่วง +3 ถึง +15 โวลต์
มาร์ค จะเป็นแรงดันไฟฟ้าลบ	เอาต์พุตอยู่ในช่วง -5 ถึง -15 โวลต์
	อินพุตอยู่ในช่วง -3 ถึง -15 โวลต์

ค่าระว่าง -3 โวลต์ถึง +3 โวลต์ จะเป็นช่วงบอกค่าไม่ได้

ความแตกต่างของเอาต์พุตและอินพุตมีไว้เผื่อกรณีที่แรงดันไฟฟ้าสูญหายเนื่องจากความยาวของสายสัญญาณ และจะพบว่าเมื่อให้สายสัญญาณยาวเกินไป ระดับแรงดันไฟฟ้าจะตกลงเกินขอบเขตที่ยอมรับได้ นอกจากนี้ความจุไฟฟ้าที่เกิดขึ้นจะมีผลกับคุณภาพของสัญญาณทำให้สถานะจากแรงดันไฟฟ้าบวกและลบไม่ชัดเจน ทำให้การติดต่อไม่ได้ระยะไกลนัก แต่ถ้าจำเป็นต้องใช้ต้องมีอุปกรณ์อื่นเพิ่มเติม



รูปที่ 2.11 RS-232 Wave Form

2.5.5 รีจิสเตอร์ของพอร์ตอนุกรม

ค่ารีจิสเตอร์ของพอร์ตอนุกรมสำหรับการเขียนโปรแกรมควบคุม มีดังนี้

Name	I/O Address	IRQ	interrupt Vector	Programmable Interrupt Controller	Mask IRQ Value
COM1	3F8	4	0x0C	0xEF	0x10
COM2	2F8	3	0x0B	0xF7	0x08
COM3	3E8	4	0x0C	0xEF	0x10
COM4	2E8	3	0x0B	0xF7	0x08

ตารางที่ 2.6 การเซตค่าพอร์ตอนุกรม

2.6 หลักการสื่อสารข้อมูล (Data Communication)

การสื่อสารข้อมูล คือ ขบวนการของการใช้ร่วมหรือแลกเปลี่ยนข้อมูลข่าวสารที่ถูกแปลงรหัสระหว่างอุปกรณ์สองตัวหรือมากกว่านั้น การสื่อสารข้อมูลตัวส่งและตัวรับเป็นอุปกรณ์หรือเครื่องมือ และข้อมูลข่าวสารที่ถูกแปลงรหัส หมายถึง ข้อมูลข่าวสารที่ถูกส่งผ่านไปในลักษณะของการส่งผ่านข้อมูลแบบอนุกรมของสัญญาณไฟฟ้า ผ่านตัวกลางที่ใช้ในการส่งข้อมูล เช่น เมื่อเจ้าหน้าที่คอมพิวเตอร์กดปุ่มอักษรบนแป้นพิมพ์ สัญญาณทางไฟฟ้าถูกส่งผ่านพอร์ตอนุกรมออกไปยังคอมพิวเตอร์และคอมพิวเตอร์จะแปลงข้อมูลข่าวสารกลับมา

รูปที่ 2.12 แสดงส่วนประกอบของการสื่อสารข้อมูล

2.6.1 ส่วนประกอบของการสื่อสารข้อมูลประกอบด้วย

1. DTE (Data Terminal Equipment) เป็นอุปกรณ์ต้นทางหรือปลายทางข้อมูล ทำหน้าที่รับหรือส่งข้อมูล โดยอีกนัยหนึ่งคือผู้ใช้ข้อมูล เช่น คอมพิวเตอร์ แป้นพิมพ์ จอภาพ เครื่องพิมพ์ เครื่องทอร์มินอล เป็นต้น

2. DCE (Data Communication Equipment) เป็นอุปกรณ์การสื่อสารข้อมูล ทำหน้าที่ในการเชื่อมต่อ และควบคุมการส่งผ่านข้อมูลระหว่างตัว DTE ทั้งสองตัว ผ่านตัวกลางในการสื่อสารหรือช่องผ่านสัญญาณ เช่น โมเด็ม มัลติเพล็กเซอร์ เป็นต้น

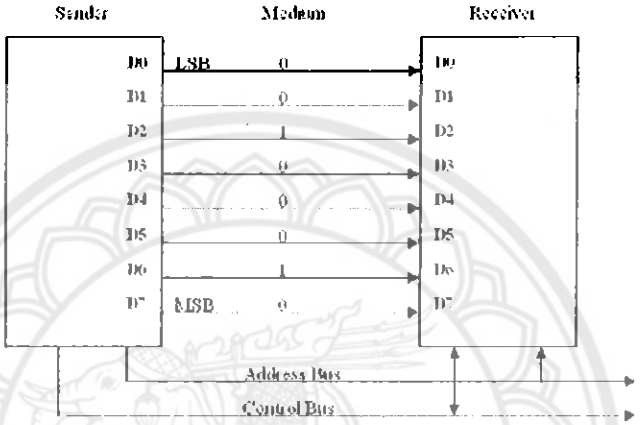
3. Medium เป็นตัวกลางที่ใช้ในการสื่อสารข้อมูลหรืออาจจะเรียกว่า ช่องสัญญาณสื่อสาร เช่น คาต้าลิงค์เน็ตเวิร์ค สายโคแอ็กเชียล ไฟเบอร์ออฟติก และช่องสัญญาณดาวเทียม เป็นต้น

2.6.2 เทคนิคการส่งผ่านข้อมูล

ในการส่งผ่านข้อมูล สามารถแบ่งลักษณะของการส่งผ่านข้อมูลได้เป็น 2 แบบ คือการส่งผ่านข้อมูลแบบขนาน (Parallel) และ การส่งผ่านข้อมูลแบบอนุกรม (Serial) การส่งผ่านข้อมูลโดยใช้สายคู่เดียว เป็นการส่งผ่านข้อมูลแบบอนุกรม และการส่งผ่านข้อมูล โดยใช้สายหลายคู่สาย เป็นการส่งผ่านข้อมูลแบบขนาน ในการส่งผ่านข้อมูลแบบอนุกรม ข้อมูลไบนารีจะถูกส่งออกไปครั้งละ 1 บิตที่เวลาหนึ่งส่วนในการส่งข้อมูลแบบขนาน ข้อมูลแต่ละบิตจะมีสายส่งเฉพาะ และทุกบิตของข้อมูลที่แต่ละสายส่ง จะถูกส่งออกไปในเวลาเดียวกัน ดังแสดงในรูปที่ 2.15 แสดงตัวอย่างของการส่งผ่านข้อมูลแบบขนาน ข้อมูลแต่ละบิตจะมีสายส่งเฉพาะ และทุกบิตของข้อมูลที่แต่ละสายส่ง จะถูกส่งออกในเวลาเดียวกัน

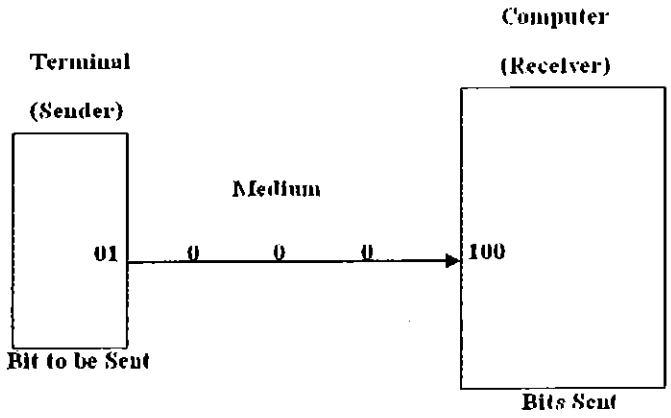
การส่งผ่านข้อมูลแบบขนานย่อมเร็วกว่าการส่งข้อมูลแบบอนุกรม เพราะว่าทุกบิตถูกส่งออกไปพร้อมกัน ดังนั้นในการส่งผ่านข้อมูลแบบขนาน จะถูกนำมาใช้สำหรับการทำงานภายในคอมพิวเตอร์ เช่นในการส่งข้อมูลระหว่าง ซีพียู กับหน่วยความจำ หรือระหว่างซีพียูและอุปกรณ์อินพุตเอาท์พุต เป็นต้น

อย่างไรก็ตามการใช้สายหลายๆ เส้น ทำให้สับสน ราคาสูงและรับสัญญาณรบกวนได้ง่าย ถ้านำการส่งผ่านข้อมูลแบบขนาน มาใช้กับการส่งผ่านข้อมูลระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ หรือคอมพิวเตอร์กับอุปกรณ์รอบนอก เช่น คอมพิวเตอร์กับเครื่องพิมพ์



รูปที่ 2.13 การส่งข้อมูลแบบขนาน (Parallel)

แม้ว่าในบางระบบจะใช้การส่งผ่านข้อมูลแบบขนาน สำหรับการส่งผ่านข้อมูลไปยังอุปกรณ์ภายนอก แต่ส่วนใหญ่จะใช้การส่งผ่านข้อมูลแบบอนุกรม ซึ่งมีอุปกรณ์อินพุตเอาต์พุต ทำการแปลงข้อมูลแบบขนานจากซีพียู เพื่อเป็นข้อมูลแบบอนุกรม ก่อนที่จะส่งข้อมูลออกไป และอุปกรณ์อินพุตเอาท์พุตเหล่านี้สามารถรับข้อมูลแบบอนุกรม และแปลงกลับมาเป็นแบบขนาน เพื่อให้สามารถอ่านได้โดย ซีพียู หรือ หน่วยความจำได้ง่าย เพราะว่าในการสื่อสารข้อมูลจะเกิดขึ้นระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ หรือ คอมพิวเตอร์กับอุปกรณ์รอบนอก



รูปที่ 2.14 การส่งข้อมูลแบบอนุกรม(Serial)

2.6.3 การส่งผ่านข้อมูลแบบอนุกรม (Serial Communication)

การส่งผ่านสัญญาณแบบอนุกรมแบ่งออกเป็น 2 วิธี คือ การส่งสัญญาณแบบ อะซิงโครนัส (Asynchronous Transmission) และการส่งสัญญาณแบบ ซิงโครนัส (Synchronous Transmission) เพื่อทำให้ตัวส่งและตัวรับสามารถทำงานได้สอดคล้องกันทั้งคู่ จึงจะต้องใช้วิธีการส่งสัญญาณแบบเดียวกัน คือตัวรับต้องสามารถตรวจจับการเริ่มต้นและการสิ้นสุดของอักขระ (Character) 1 ตัวได้ สำหรับการส่งข้อมูลแบบอะซิงโครนัส ส่วนในแบบซิงโครนัสจะดูที่บล็อกของการเริ่มต้น และการสิ้นสุดของอักขระ

2.6.4 อัตราการส่งข้อมูล

ความเร็วในการส่งข้อมูลแบบอนุกรมจะบอกเป็นจำนวนบิตต่อวินาที (bits per second: bps) ที่เรียกว่า บอดเรต(baud rate) พอร์ตอนุกรมของ PIC 16F877 สามารถติดต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคลได้แต่ต้องกำหนดอัตราเร็วให้เท่ากัน การสื่อสารทางพอร์ตอนุกรม RS-232 ของคอมพิวเตอร์ ได้กำหนดอัตราเร็วไว้หลายค่าตั้งแต่ 100 ถึง 19200 bps สำหรับคอมพิวเตอร์ที่ใช้ไมโครโปรเซสเซอร์เพนเทียมสามารถส่งข้อมูลได้เร็วสูงถึง 56 กิโลบิตต่อวินาที

2.7 ไอซีฐานเวลาจริง DS1307

การสร้างฐานเวลาให้กับระบบไมโครคอนโทรลเลอร์อาจทำได้โดยการใช้วงจรกำเนิดความถี่ป้อนให้กับตัวนับ/ตัวจับเวลาในไมโครคอนโทรลเลอร์ แต่การใช้งานแบบนี้ผู้ใช้จะต้องเขียนโปรแกรมควบคุมการทำงาน มีการคำนวณคาบเวลาให้ถูกต้อง และต้องตั้งค่าใหม่ทุกครั้งที่มีการเริ่มทำงาน ระบบการสร้างฐานเวลาแบบนี้สามารถใช้งานง่ายๆได้ แต่ไม่สามารถบอกเวลาจริงให้กับระบบได้

ระบบที่ใช้ไมโครคอนโทรลเลอร์ในการควบคุมบางระบบจะต้องมีเวลาเข้ามาเกี่ยวข้องด้วย โดยจะมีการบอกเวลาเป็นชั่วโมง นาที และวินาที หรือมีการบอกวัน เดือน และปี ให้กับระบบด้วย

ดังนั้นการสร้างฐานเวลาต้องถูกต้องและแม่นยำ แม้ว่าระบบหยุดทำงานและเริ่มทำงานใหม่ค่าเวลาต่างๆจะต้องถูกต้องด้วย ในปัจจุบันการทำงานประเภทนี้จะใช้ไอซีที่ทำหน้าที่สร้างฐานเวลาจริงให้กับระบบ (RTC หรือ Real Time Clock) ไอซีประเภทนี้จะมีวงจรจัดการด้านเวลาจริงอยู่ภายในการใช้งานเพียงต่อคริสตอลให้กับไอซี การทำงานด้านเวลาจะเป็นไปอย่างอัตโนมัติ เสมือนกับว่าเป็นนาฬิกาและปฏิทินให้กับระบบ ถ้าหากเมื่อใดต้องการทราบเวลาสามารถอ่านค่าเวลาจากหน่วยความจำภายในของไอซีได้โดยตรง

โดยทั่วไปแล้วไอซีที่ทำหน้าที่เป็น RTC จะมีอยู่หลายเบอร์ บางเบอร์เมื่อเวลาเดินมาถึงค่าที่กำหนดสามารถใช้อินเตอร์รัปต์ไมโครคอนโทรลเลอร์ได้ ในที่นี้จะแนะนำเบอร์ DS1307 ของบริษัท Dallas Semiconductor ซึ่งเป็นไอซีที่มีความแม่นยำสูง โดยเป็นไอซีแบบ 8 ขา การเชื่อมต่อกับไมโครคอนโทรลเลอร์จะใช้การอินเตอร์เฟสแบบอนุกรม 2 สาย หรือแบบ I^2C ไอซีเบอร์นี้กินพลังงานต่ำมาก พร้อมทั้งมีปฏิทินเวลาแบบ BCD สามารถใช้ข้อมูลเกี่ยวกับเวลา เช่น วินาที นาที ชั่วโมง (ทั้งแบบ 24 ชั่วโมง/12ชั่วโมง พร้อมทั้งระบบค่า AM/PM) และบอกวัน เดือน ปี ได้ โดยจะมีการปรับวันที่โดยอัตโนมัติ ในแต่ละเดือนจะแสดงวันได้สูงสุดได้ไม่เกิน 31 วัน และจะปรับวันต่างๆ อย่างถูกต้องเมื่อครบปี การใช้งานกับไมโครคอนโทรลเลอร์จะใช้การส่งข้อมูลและแอดเดรสของค่าต่างๆ แบบอนุกรม โดยมีขาหนึ่งเป็นขาสัญญาณ อีกขาหนึ่งเป็นตัวกำหนดสัญญาณนาฬิกา

ไอซี DS1307 เป็นไอซี 8 ขา ขาต่างๆ ของ DS1307 เป็นดังต่อไปนี้

ขา SDA (Serial Data Input/Output) เป็นขารับส่งข้อมูลแบบอนุกรม ในการอินเตอร์เฟสจะต้องมีตัวต้านทานพูลอัพภายนอกด้วย

ขา SCL (Serial Clock Input) เป็นขาอินพุตสัญญาณนาฬิกาอนุกรมเพื่อให้เกิดการซิงโครไนซ์ในการรับส่งข้อมูลแบบอนุกรม

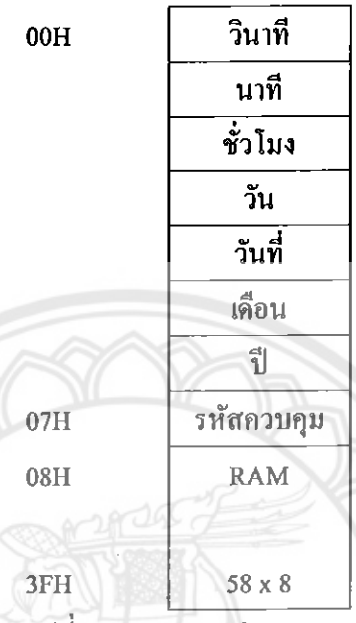
ขา SQW/OUT เป็นขาส่งสัญญาณคลื่นสี่เหลี่ยมออกมาทางเอาต์พุต เมื่อเริ่มทำงานบิตนี้จะถูกเซตเป็นลอจิก "1" สัญญาณเอาต์พุตที่ได้จะมีอยู่ 4 ค่า คือ 1 Hz, 4kHz, 8kHz และ 32 kHz โดยสามารถเลือกได้ในการใช้งานจะต้องมีตัวต้านทานต่อพูลอัพภายนอกด้วย ในการใช้งานบางประเภทไม่จำเป็นต้องใช้ขานี้

ขา X1, X2 เป็นขาที่ใช้ต่อกับคริสตอลภายนอก โดยใช้ความถี่ 32.768 kHz วงจรสร้างสัญญาณนาฬิกาภายในออกแบบให้ทำงานรวมกับคริสตอลที่มีตัวเก็บประจุ 12.5 พิโกฟารัดต่อรวมอยู่ด้วย

ขา V_{cc} และ GND เป็นขาที่ใช้ต่อกับไฟเลี้ยง โดยทั่วไปแล้วจะต่อกับแรงดันไฟ 5 โวลต์ ถ้าหากแรงดันไฟต่อลดลงต่ำกว่า 1.25 x V_{bat} การรักษาเวลาสำรองภายในจะทำงานต่อ โดยจะรับพลังงานจากแบตเตอรี่สำรอง

V_{bat} เป็นขาต่อกับแรงดันไฟเลี้ยงสำรอง โดยทั่วไปแล้วจะใช้แบตเตอรี่แรงดัน 3 โวลต์

หน่วยความจำ RAM ภายใน และส่วนที่ใช้เป็น RTC มีการจัดแอดเดรสดังรูปที่ 2.15 โดยตำแหน่ง 00H ถึง 07H จะใช้เป็น RTC ส่วนตำแหน่ง 08H ถึง 3FH จะเป็นหน่วยความจำ RAM จากรูปที่ 2.15 จะเห็นว่าที่ตำแหน่ง 00H จะเก็บเวลาเป็น วินาที ตำแหน่ง 01 จะเก็บเวลาเป็นวินาที โดยข้อมูลที่เก็บจะอยู่ในรูปแบบรหัส BCD



รูปที่ 2.15 แผนผังหน่วยความจำ

สำหรับการอ่านและเขียนค่าเวลากับไอซีตัวนี้ทำได้โดยการอ่านและเขียน**ข้อมูลในลักษณะตัวเลข** BCD กับรีจิสเตอร์ที่เก็บค่าต่างๆ ตามที่กำหนดเอาไว้

	บิต 7							บิต 0	
00H	CH	วินาที หลักสิบ			วินาที				00-59
01H	X	นาที หลักสิบ			นาที				00-59
02H	X	12.24	A/P	10 HP	ชั่วโมง				01-12 . 00-23
03H	X	X	X	X	X	วัน			0-7
04H	X	X	วันที่ หลักสิบ		วันที่				01-31
05H	X	X	เดือน หลักสิบ		เดือน				01-12
06H	ปี หลักสิบ				ปี				00-99
07H	OUT	X	X	SQWE	X	X	RS1	RS2	

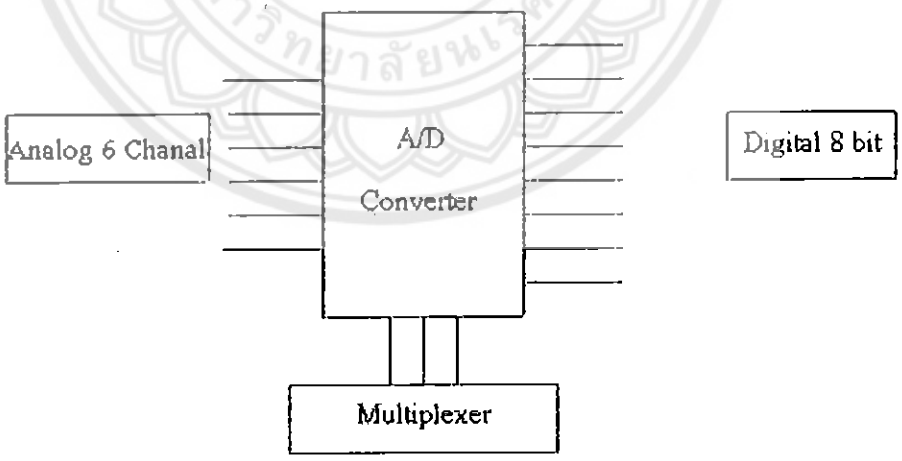
ตารางที่ 2.7 แสดงรีจิสเตอร์ที่ใช้เก็บค่าต่างๆ

ในรีจีสเตอร์ตำแหน่ง 00H ซึ่งเป็นรีจีสเตอร์วินาที ถ้าบิตที่ 7 ถูกเซต ให้เป็นลอจิก "1" วงจรออสซิลเลเตอร์จะหยุดทำงาน ถ้าเป็น ลอจิก "0" วงจรออสซิลเลเตอร์จะทำงานต่อไป สำหรับรีจีสเตอร์ตำแหน่ง 01H เป็นรีจีสเตอร์ตำแหน่งนาที ซึ่งเวลาในหลักสิบของวินาทีจะมีค่าตั้งแต่ 0 ถึง 5 ดังนั้นบิตที่ 7 จะไม่ใช้ สำหรับรีจีสเตอร์ตำแหน่ง 02H ซึ่งเป็นรีจีสเตอร์ตำแหน่งชั่วโมง สามารถเซตได้ว่าจะให้เก็บข้อมูลแบบ 12 ชั่วโมงหรือ 24 ชั่วโมง และยังบอกว่า AM/PM ได้อีกด้วย ส่วนรีจีสเตอร์ควบคุมเพื่อกำหนดว่าจะให้มีเอาต์พุตออกมาทางขา SQW/OUT หรือไม่ และเอาต์พุตที่ออกมาจะมีความถี่เท่าใด รายละเอียดต่างๆ สามารถศึกษาได้จากคู่มือของไอซีได้โดยตรง ซึ่งไม่กล่าวถึงในที่นี้

การจัดการหน่วยความจำภายในของไอซีฐานเวลาในลักษณะนี้ ทำให้ผู้ใช้สามารถเลือกอ่านหรือเขียนข้อมูลใดๆ ในตำแหน่งที่กำหนดได้ เช่นถ้าหากอ่านค่าในตำแหน่ง 01H ออกมาได้เป็น 0011 0010 หรือ 32 ในระบบ BCD จะหมายความว่า เป็นเวลา 32 นาที ถ้าอ่านค่าในตำแหน่งที่ 02H ออกมาได้เป็น 0010 0010 หรือ 22 ในระบบเลข BCD หมายความว่าเป็นเวลา 22 ชั่วโมง เป็นต้น

2.8 วงจร A/D converter

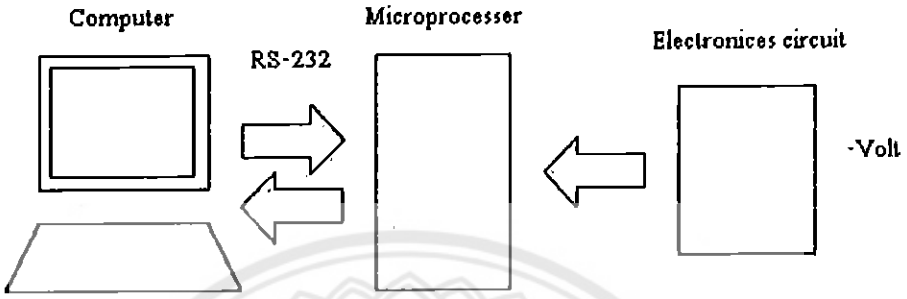
เป็นวงจรที่เปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล โดยจะต้องใช้ A/D converter 8 บิต คือจะมีความละเอียดในการแปลงสัญญาณได้ 28 หรือ 256 ระดับ และยังต้องใช้ A/D converter ที่มีมัลติเพล็กเซอร์ที่มากกว่าหรือเท่ากับ 6 ช่องสัญญาณ เพื่อที่จะได้ใช้แปลงสัญญาณทั้ง 6 ช่องสัญญาณ



รูปที่ 2.16 การทำงานของ A/D Converter

บทที่ 3
การออกแบบและสร้าง เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

บล็อกไดอะแกรมการทำงาน



รูปที่ 3.1 แสดงบล็อกไดอะแกรม

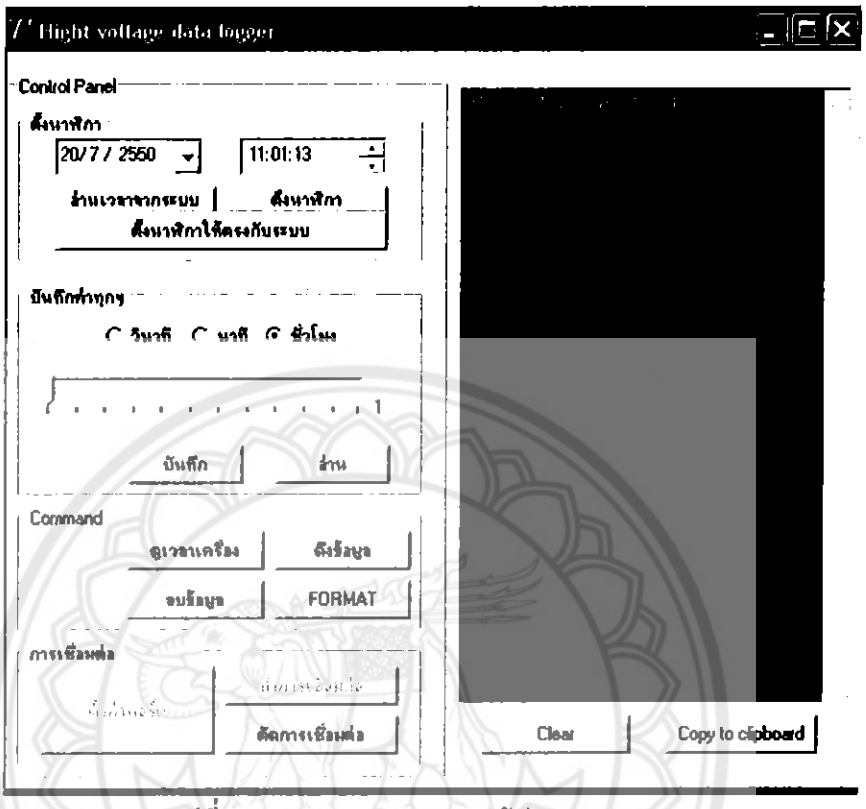
3.1 ส่วนของคอมพิวเตอร์

เป็นอุปกรณ์ควบคุมการทำงานของไมโครคอนโทรลเลอร์ โดยใช้โปรแกรม DELPSI ส่งค่าตัวแปรที่ใช้ในการควบคุมการทำงานให้แก่ไมโครคอนโทรลเลอร์ และรับข้อมูลจากไมโครคอนโทรลเลอร์มาวิเคราะห์แสดงผล ในรูปแบบของข้อมูลตัวเลข และทำการเก็บข้อมูลที่รับเข้ามาไว้ในโปรแกรม notepad

3.1.1 อัตราความเร็วในการรับ-ส่งข้อมูลอนุกรม

การที่คอมพิวเตอร์กับไมโครคอนโทรลเลอร์จะติดต่อสื่อสารกันได้นั้น จะต้องทำงานด้วยอัตราเร็วเท่ากันซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสที่ใช้ในโครงงานนี้ คือ ค่าบอดเรต 9600 บิตต่อวินาที ซึ่งเป็นค่าอัตราเร็วในการสื่อสารแบบอนุกรม RS-232

3.1.2 ส่วนของโปรแกรมควบคุมการทำงาน(Delphi)



รูปที่ 3.2 แสดงการทำงานของโปรแกรม Delphi

ส่วนของ DELPSI ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์

1. ส่วนของ ตั้งนาฬิกา

   1.1 ปุ่ม "อ่านเวลาจากระบบ" ทำหน้าที่อ่านค่าเวลาจากคอมพิวเตอร์ที่ต่อกับอุปกรณ์

   1.2 ปุ่ม "ตั้งนาฬิกา" ทำหน้าที่ ตั้งเวลาของ DS1307 ไอซีเวลาของบอร์ด

   1.3 ปุ่ม "ตั้งนาฬิกาให้ตรงกับระบบ" ทำหน้าที่ตั้งเวลาของบอร์ดให้ตรงกับเวลาของคอมพิวเตอร์

2. ส่วนของ บันทึกค่าทุกๆ

   2.1 ปุ่ม "บันทึก" ทำหน้าที่ ตั้งค่าช่วงเวลา ความถี่ในการบันทึกข้อมูลที่วัดได้ลงบอร์ด

   2.2 ปุ่ม "อ่าน" ทำหน้าที่ อ่านค่าช่วงเวลาความถี่ในการบันทึกข้อมูลที่ถูกตั้งค่าไว้

ปร.
A4170
2550
ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร 15009224

3. ส่วนของ Command

3.1 ปุ่ม "ดูเวลาเครื่อง" ทำหน้าที่ ดูเวลาเวลาของ DS1307 ไอซีเวลาของบอร์ด

3.2 ปุ่ม "ดึงข้อมูล" ทำหน้าที่ ดึงข้อมูลจากหน่วยความจำของบอร์ด เข้าสู่คอมพิวเตอร์

3.3 ปุ่ม "ลบข้อมูล" ทำหน้าที่ ลบข้อมูลการบันทึกของหน่วยความจำ

3.4 ปุ่ม "FORMAT" ทำหน้าที่ ล้างข้อมูลทั้งหมดในหน่วยความจำ

4. ส่วนของ หน้าจอแสดงผล

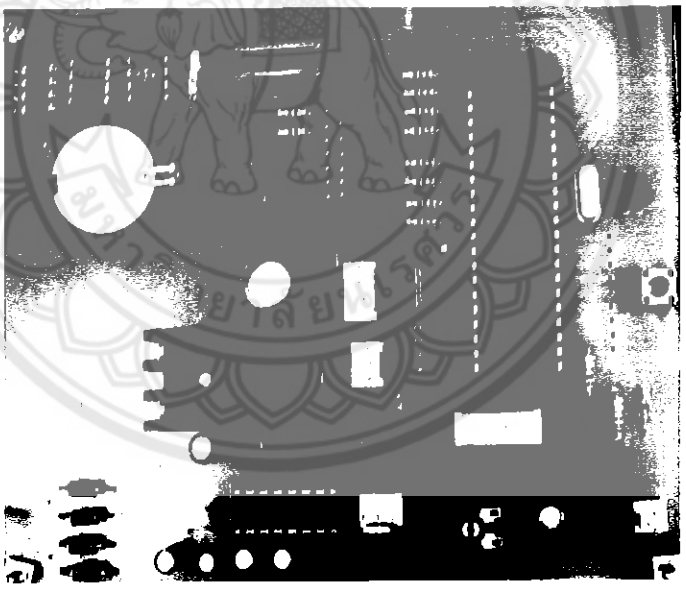
4.1 หน้าจอแสดงผล ทำหน้าที่ แสดงผลข้อมูลที่ดึงมาจากหน่วยความจำ

4.2 ปุ่ม "Clear" ทำหน้าที่ ล้างหน้าจอแสดงผล

4.3 ปุ่ม "Copy to clipboard" ทำหน้าที่ คัดลอกข้อมูล จากหน้าจอแสดงผล

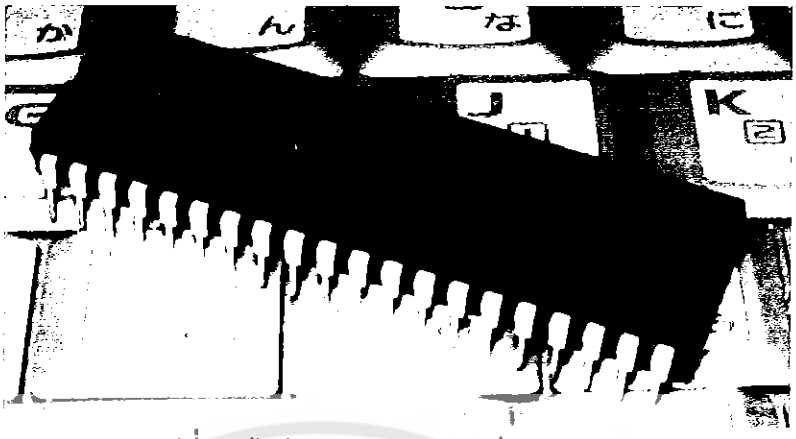
3.2 ส่วนของไมโครคอนโทรลเลอร์

เป็นฮาร์ดแวร์ที่ใช้โปรแกรมควบคุมการ รับ-ส่ง ข้อมูล ผ่าน พอร์ต RS-232 ควบคุมการทำงานโดย PIC 16F877 บันทึกข้อมูล ด้วย 24LC512 และควบคุมเวลาด้วย DS1307



รูปที่ 3.3 บอร์ดไมโครคอนโทรลเลอร์

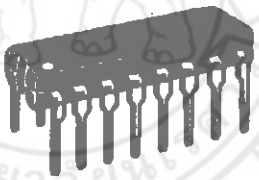
3.2.1 ไมโครคอนโทรลเลอร์ PIC16F877



รูปที่ 3.4 ไมโครคอนโทรลเลอร์ PIC 16F877

3.2.2 ไอซี RS232

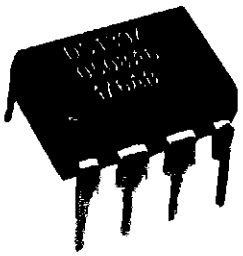
ส่วนเชื่อมต่อกับพอร์ตของคอมพิวเตอร์



รูปที่ 3.5 ไอซี RS232

3.2.3 ไอซีเวลา DS1307

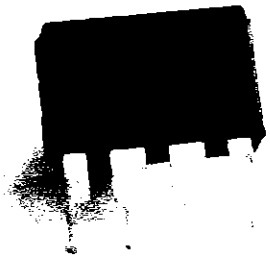
เป็นไอซีบอกเวลา



รูปที่ 3.6 ไอซีเวลา

3.2.4 ไอซีหน่วยความจำ

หน่วยความจำ 24LC512 เป็นหน่วยความจำหลักในการเก็บค่าของข้อมูลที่บันทึกไว้



รูปที่ 3.7 หน่วยความจำ

3.2.5 ไดโอด

ต่อเป็นวงจรบริดจ์ ทำหน้าที่เป็นวงจรเรดติไฟเออร์ โดยเปลี่ยนไฟฟ้ากระแสสลับเป็นกระแสตรง และกลับขั้วไฟฟ้าให้ถูกต้อง

รูปที่ 3.8 ไดโอด

3.2.6 ตัวเก็บประจุ

ตัวเก็บประจุ จะทำหน้าที่กรองแรงดันหรือฟิลเตอร์ (Filter) เพราะในการเปลี่ยนแรงดันฟ้ากระแสสลับให้เป็นแรงดันไฟฟ้ากระแสตรง จะยังมีการกระเพื่อมของแรงดันไฟตรง ที่เรียกว่า ริปเปิล (Ripple) ตัวเก็บประจุจะลดค่าแรงดันริปเปิล โดยการเก็บค่าประจุไว้ เมื่อช่วงแรงดันสูง และจ่ายให้กับโหลดเมื่อมีการกระเพิ่มแรงดันต่ำ ดังนั้นโหลดจะได้แรงดันที่ราบเรียบขึ้น

รูปที่ 3.9 ตัวเก็บประจุ

3.2.7 ตัวต้านทาน

รูปที่ 3.10 ตัวต้านทาน

3.2.8 แบตเตอรี่

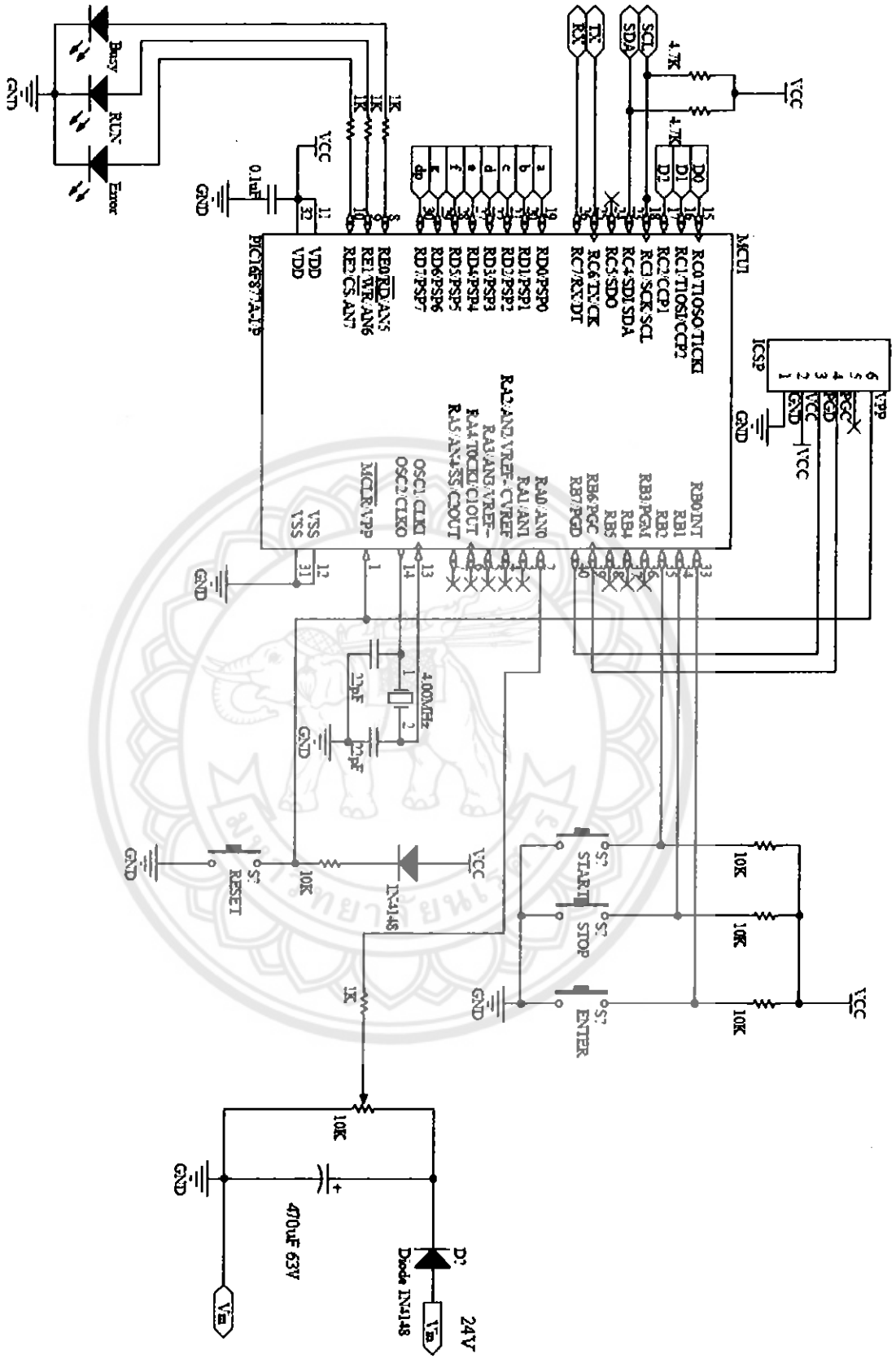
แบตเตอรี่ขนาด 12 VDC จะเป็นแหล่งจ่ายไฟฟ้าสำรองให้กับตัวอุปกรณ์ในบอร์ดของเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า เมื่อแหล่งจ่ายหลักจากหม้อแปลงไม่สามารถจ่ายพลังงานได้

รูปที่ 3.11 แบตเตอรี่

3.2.9 หม้อแปลง

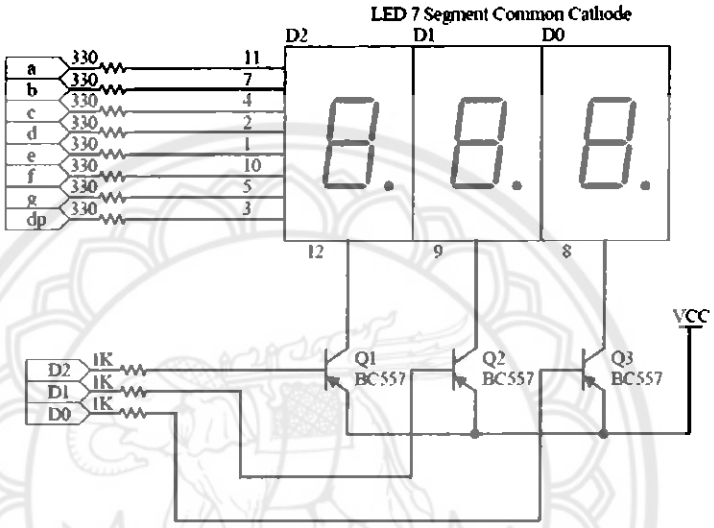
หม้อแปลงเป็นตัวแปลงแรงดัน ขนาด 220 โวลต์เป็น 12 โวลต์เพื่อจ่ายเป็นไฟเลี้ยงให้กับบอร์ดของเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

รูปที่ 3.12 หม้อแปลง

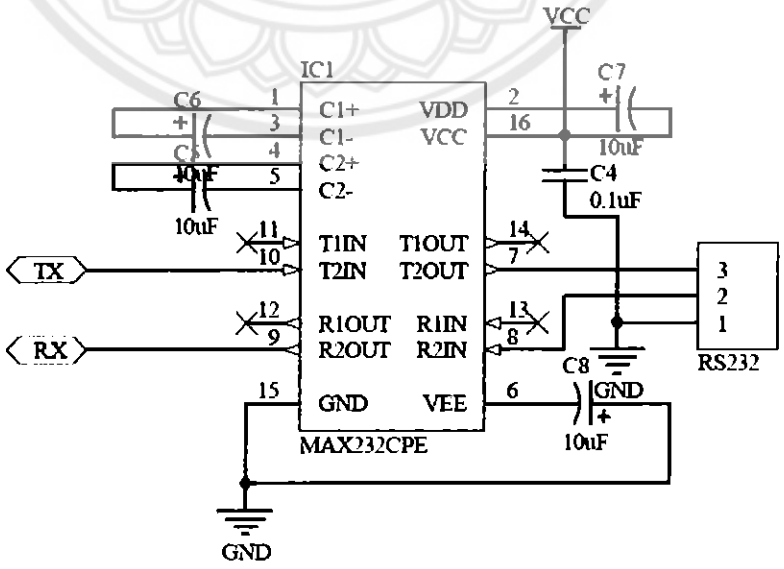


รูปที่ 3.13 วงจรแสดงการเชื่อมต่อของ PIC16F877

แรงดันไฟฟ้าที่เป็นสัญญาณอะนาลอกจะต้องเป็นแรงดันไฟฟ้ากระแสตรงที่มีค่าไม่เกิน +5 VDC ดังนั้นเมื่อแรงดันไฟฟ้ากระแสสลับที่เข้ามาผ่านไดโอดจะผ่านได้เฉพาะแรงดันไฟฟ้าด้านบวก และตัวเก็บประจุจะช่วยให้แรงดันไฟฟ้าเดินเรียบขึ้น และผ่านโวลต์เตจดิไวเดอร์ทำให้แรงดันมีค่าไม่เกิน +5 VDC โดยแรงดันไฟฟ้าจะเข้าที่ขาสองของ PIC16F877 และPIC ยังต่อเข้ากับหน่วยความจำ 24LC512 และไอซีบอกเวลา DS1307 เพื่อรับสัญญาณนาฬิกาแล้วทำงานตามเวลาที่โปรแกรมได้ตั้งค่าไว้

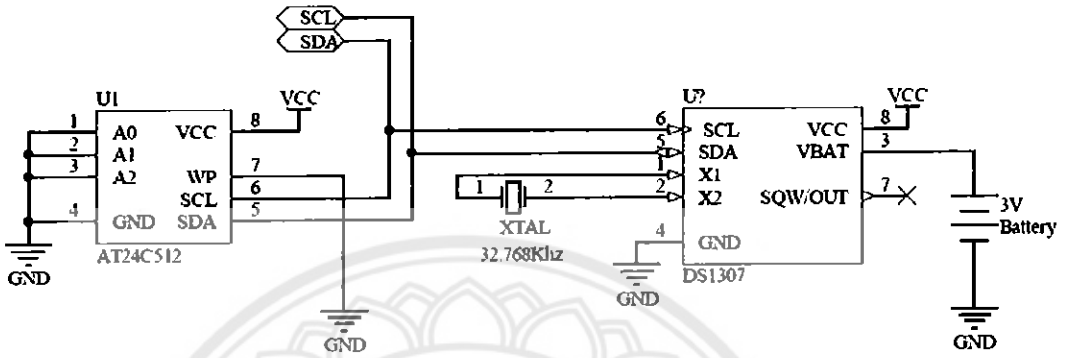


รูปที่ 3.14 วงจรแสดงการเชื่อมต่อของ LED 7 Segment



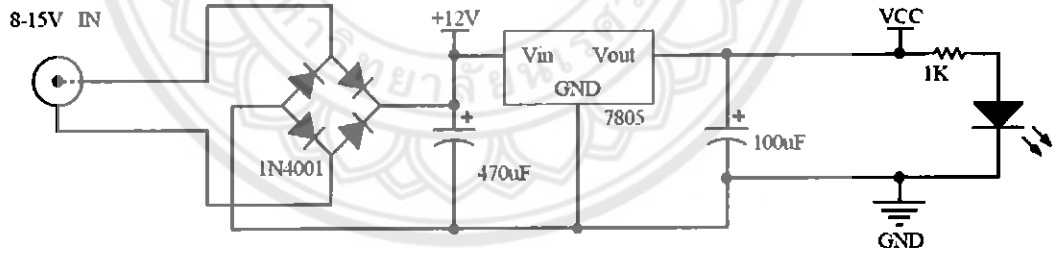
รูปที่ 3.15 วงจรแสดงการเชื่อมต่อของ RS232

ไอซี MAX232CPE ทำหน้าที่แปลงสัญญาณจากไมโครคอนโทรลเลอร์ให้สามารถส่งต่อไปยังคอมพิวเตอร์ได้และแปลงสัญญาณจากคอมพิวเตอร์ให้สามารถส่งต่อไปยังไมโครคอนโทรลเลอร์ได้โดยสัญญาณจากไมโครคอนโทรลเลอร์จะมีค่าแรงดัน 5 โวลต์และสัญญาณจากคอมพิวเตอร์จะมีค่าแรงดัน 12 โวลต์



รูปที่ 3.16 วงจรส่วนของหน่วยความจำ 24LC512 และไอซีบอกเวลา DS1307

หน่วยความจำ 24LC512 จะทำหน้าที่เก็บบันทึกข้อมูลตามโปรแกรมที่เขียนไว้ในไมโครคอนโทรลเลอร์ ไอซีบอกเวลา DS1307 จะทำหน้าที่เป็นเหมือนนาฬิกาคอยบอกเวลาให้ไมโครคอนโทรลเลอร์รับรู้ ส่วยแบตเตอรี่ 3 โวลต์ ทำหน้าที่เป็นเหมือนกับถ่านนาฬิกาของ DS1307



รูปที่ 3.17 วงจรไฟเลี้ยงไอซีในไมโครคอนโทรลเลอร์

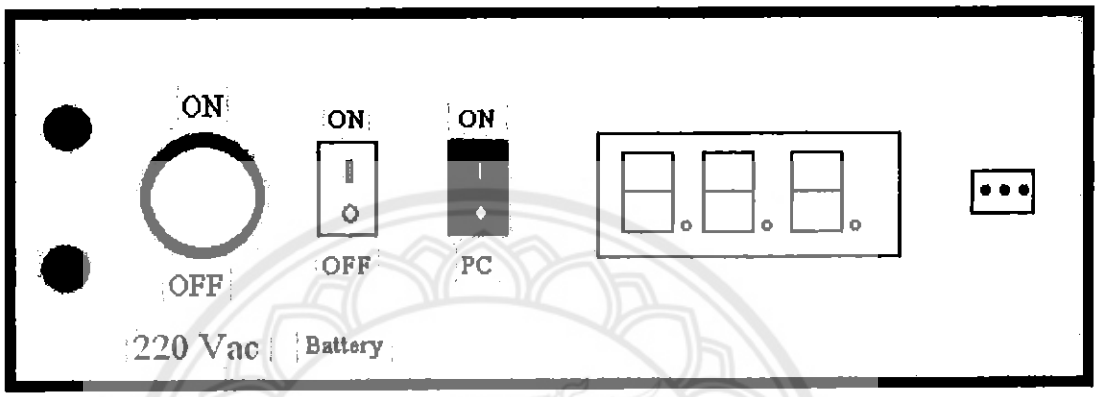
วงจรบริดจ์จะทำหน้าที่เปลี่ยนแรงดันไฟฟ้ากระแสสลับเป็นแรงดันฟ้ากระแสตรงโดยเป็นวงจรบริดจ์แบบเต็มคลื่น เมื่อผ่านวงจรบริดจ์มาแล้ว ไอซีเบอร์ 7805 จะทำหน้าที่ลดแรงดันไฟฟ้าให้เหลือเพียง 5 โวลต์ แล้วจ่ายเป็นไฟเลี้ยงกับตัวไอซีอื่นๆ

3.3 การต่อวงจรทั้งหมดของแหล่งจ่ายแรงดันไฟฟ้า (Hardware)

การต่อวงจรทั้งหมดของแหล่งนั้นจะต้องคำนึงถึงส่วนประกอบต่างๆ ว่าอยู่ในรูปที่สามารถเข้ากันได้ เช่น ข้อมูลก่อนที่จะเข้า A/D converter นั้นจะต้องเป็นข้อมูลที่อยู่ในรูปของสัญญาณแรงดันและมีช่วงของสัญญาณไม่เกินขนาดของแรงดันอ้างอิง คือ 5 Vdc ที่ป้อนให้กับ A/D converter หรือสัญญาณที่ออกจากไมโครคอนโทรลเลอร์ที่จะต่อเข้ากับคอมพิวเตอร์จะต้องเป็นข้อมูลอนุกรม มีอัตราความเร็วในการส่งข้อมูล (Baud Rate) ตรงกับอัตราเร็วที่โปรแกรมในการคำนวณต้องคำนึงถึงมาตรฐานแรงดันที่ Port RS-232 กำหนดไว้เพื่อที่จะได้สามารถส่งข้อมูลได้อย่างไม่ผิดพลาด

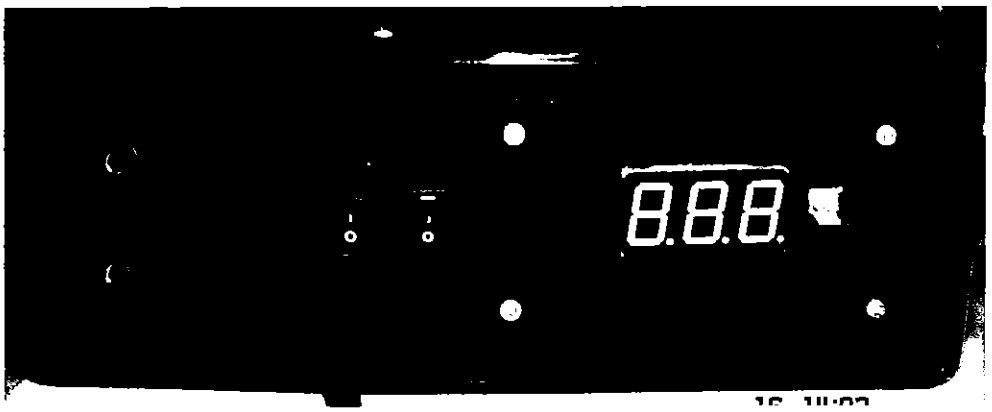
บทที่ 4
การใช้งานเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

4.1 การใช้งานตัวเครื่องวัด



รูปที่ 4.1 หน้าจอเครื่องจำลองของเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

รูปที่ 4.2 เครื่องวัดและบันทึกค่าแรงดันไฟฟ้า



รูปที่ 4.3 หน้าจอเครื่องวัดและบันทึกค่าแรงดันไฟฟ้าขณะไม่ใช้งาน

รูปที่ 4.4 สายเชื่อมต่อกับคอมพิวเตอร์

4.1.1 ขณะปิดเครื่องสวิตซ์ทั้งสามตัวต้องเป็นดังนี้

4.1.1.1 สวิตซ์ 220 Vac ต้องอยู่ตำแหน่ง OFF

4.1.1.2 สวิตซ์ Battery ต้องอยู่ตำแหน่ง OFF

4.1.1.3 สวิตซ์ PC ต้องอยู่ตำแหน่ง PC

4.1.2 เมื่อต้องการเปิดเครื่อง

4.1.2.1 เปิดโดยไม่เสียบปลั๊ก 220 Vac ให้กดสวิตซ์ Battery ให้อยู่ตำแหน่ง ON ตัวเครื่องจะใช้ไฟเลี้ยงจากแบตเตอร์รี่ขนาด 12 Vdc

4.1.2.2 เปิดโดยเสียบปลั๊ก 220 Vac ให้กดสวิตซ์ 220 Vac ให้อยู่ในตำแหน่ง ON และกดสวิตซ์ Battery ให้อยู่ตำแหน่ง ON แรงดัน 220 Vac จะแปลงเหลือ 12 Vac และทำการจ่ายไฟเลี้ยงเครื่องและทำการชาร์ตแบตเตอร์รี่ 12 Vdc

4.1.3 เมื่อต้องการต่อเครื่องวัดเข้ากับคอมพิวเตอร์

4.1.3.1 นำสายต่อพอร์ตอนุกรมมาต่อกับเครื่องและคอมพิวเตอร์

4.1.3.2 เปิดเครื่องและปรับสวิตซ์ PC ให้อยู่ในตำแหน่ง PC

4.1.3.3 ทำการเชื่อมต่อเครื่องกับคอมพิวเตอร์ด้วยโปรแกรม DELPHI

4.1.4 เมื่อต้องการวัดและบันทึกค่าแรงดันไฟฟ้า

4.1.4.1 ต่อสายที่ใช้ในการวัดกับจุดที่ต้องการวัดค่า

4.1.4.2 ปรับสวิตซ์ PC ให้อยู่ตำแหน่ง ON

4.1.4.3 เครื่องจะเริ่มทำการเก็บค่าแรงดันที่วัดได้ลงหน่วยความจำ ตามความถี่ของการบันทึกที่ตั้งค่าไว้

4.1.5 เมื่อต้องการตั้งค่าความถี่ในการบันทึกค่า

4.1.5.1 ทำการเชื่อมต่อเครื่องกับคอมพิวเตอร์

4.1.5.2 ทำการตั้งค่าผ่านทางโปรแกรม DataLogger

4.1.6 เมื่อต้องการอ่านค่าที่บันทึกไว้ในหน่วยความจำ

4.1.6.1 ใช้โปรแกรม Datalogger ในการดึงข้อมูล

4.1.7 เมื่อต้องการนำข้อมูลที่ได้ไปประมวลผล

4.1.7.1 ทำการดึงข้อมูลให้แสดงผลในโปรแกรม Datalogger

4.1.7.2 กด Copy to clipboard

4.1.7.3 เปิดโปรแกรม Notepad

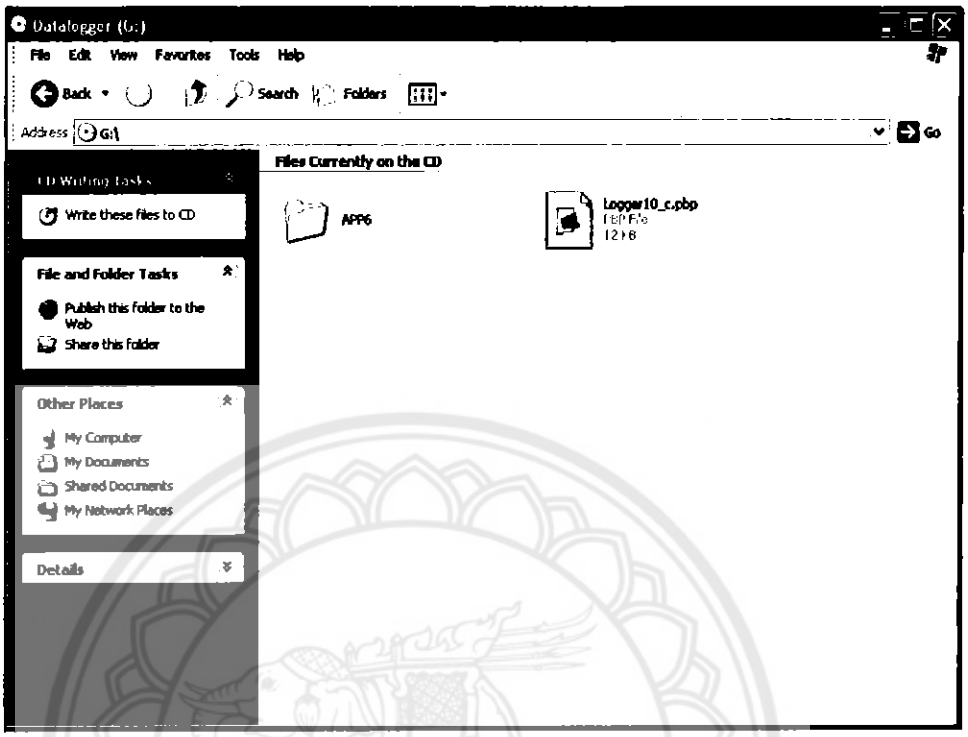
4.1.7.4 กด Ctrl+V

4.1.7.5 Save file โดยใช้นามสกุล .CSV

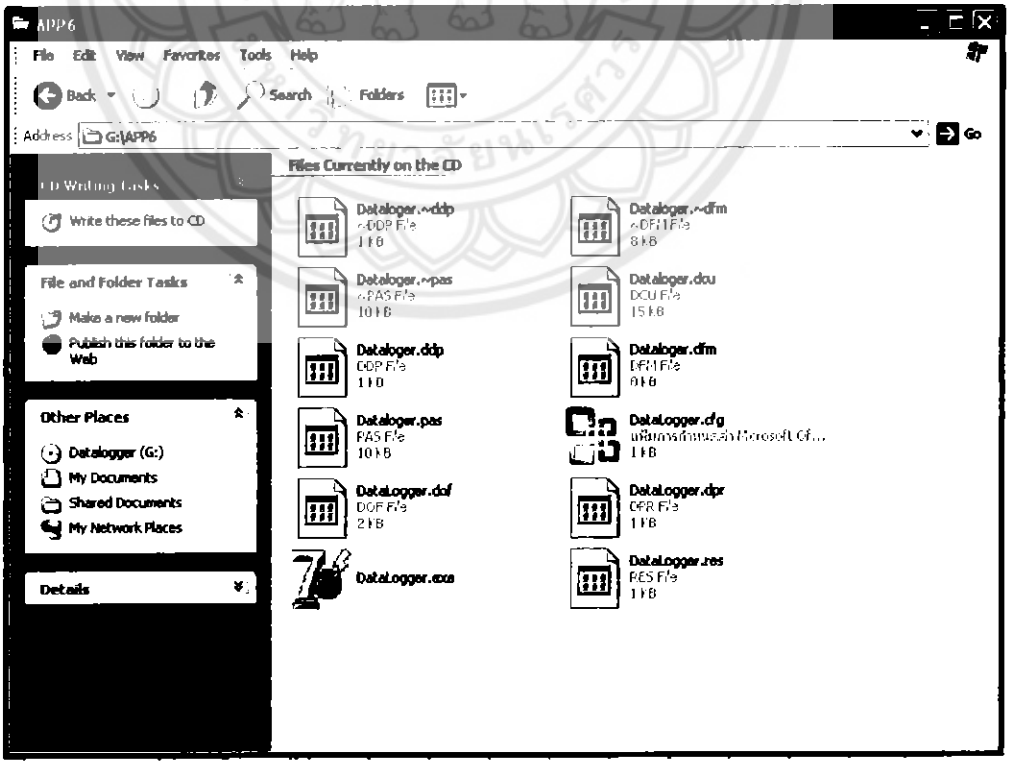
4.1.7.6 เปิดเอกสารที่บันทึกไว้ โดยโปรแกรม Microsoft Excel

4.2 การเริ่มใช้งานโปรแกรม Data Logger

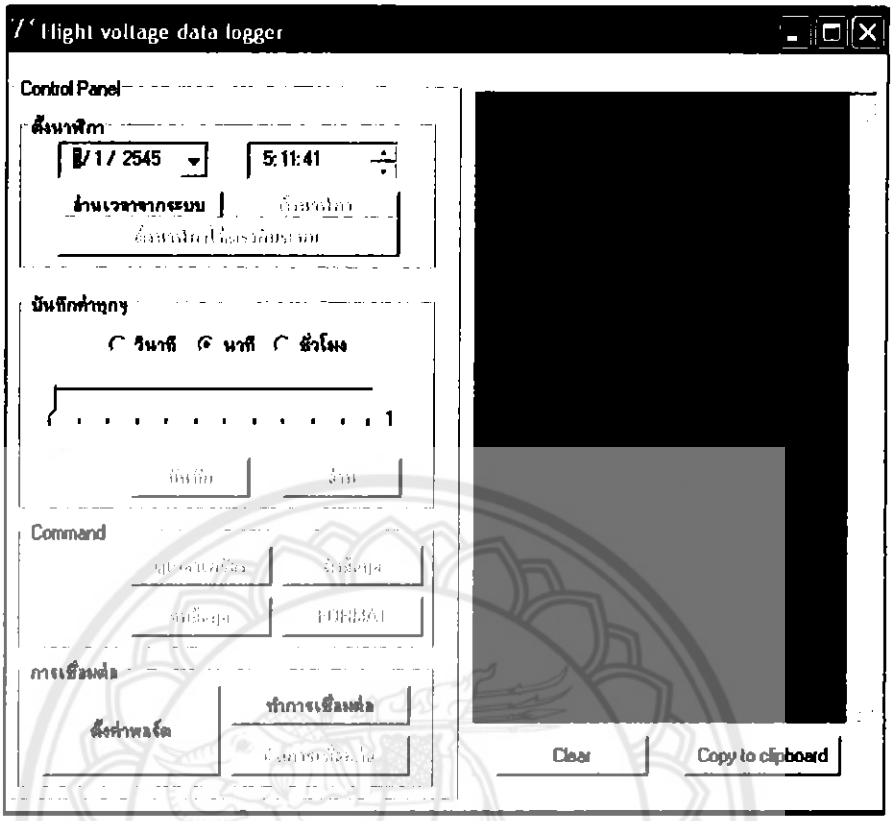
1.ใส่แผ่น ซีดี Data Logger



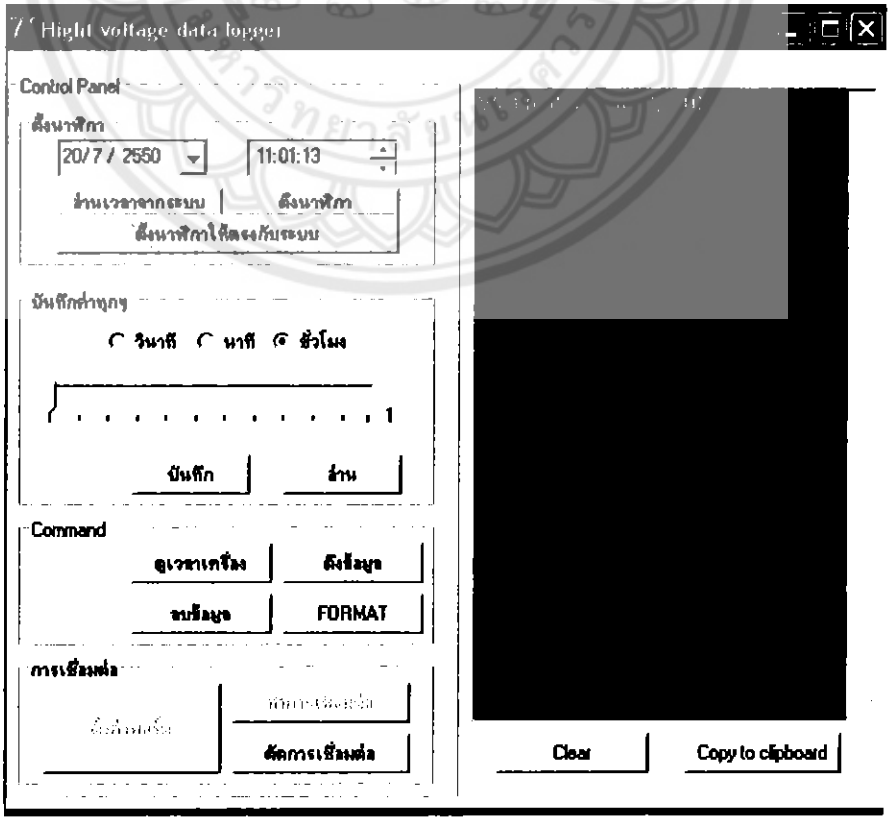
2.เปิด Folder APP6



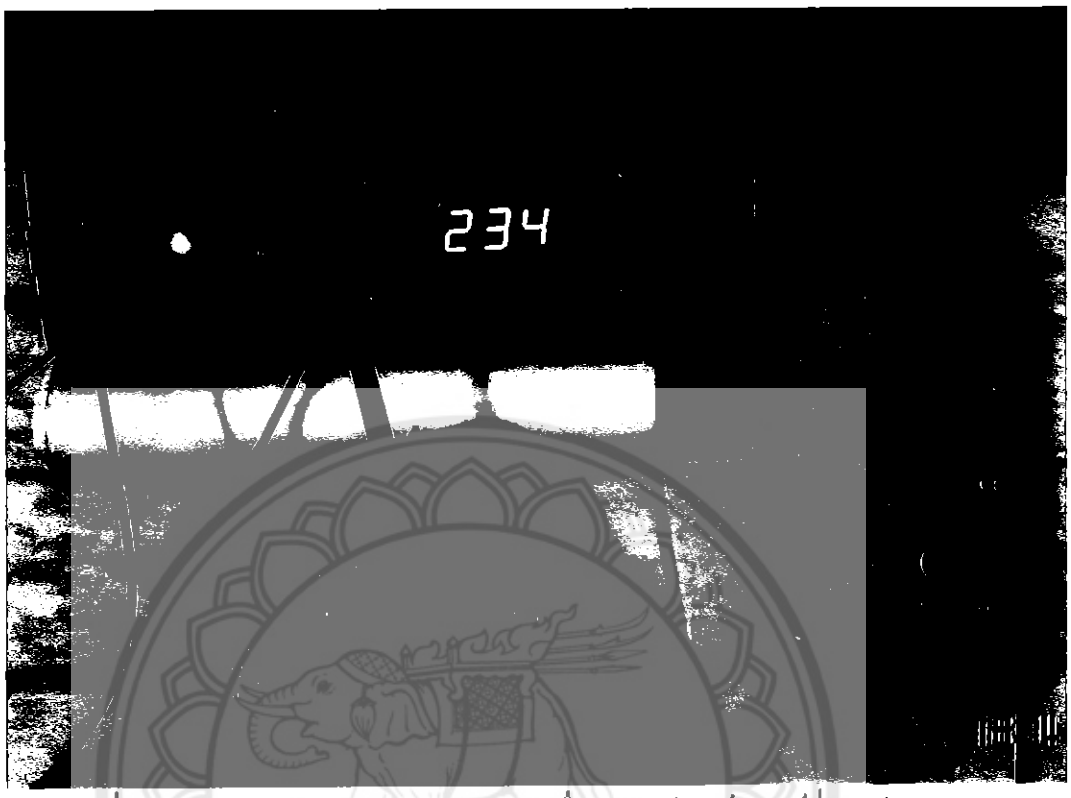
3.Double Click DataLogger.exe รันโปรแกรม



4. ทำการเชื่อมต่อเครื่องวัดและบันทึกค่ากับคอมพิวเตอร์



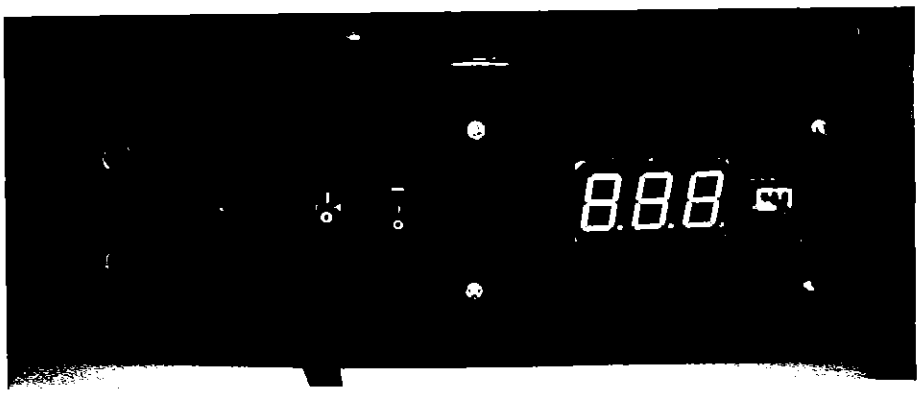
4.3 การวัดด้วยเครื่องวัดและบันทึกแรงดันไฟฟ้า



รูปที่ 4.5 แสดงการวัดเปรียบเทียบระหว่างเครื่องวัดกับโวลต์มิเตอร์ที่แรงดัน 220 Vac

จากรูปโวลต์มิเตอร์วัดค่าแรงดันไฟฟ้าได้ 234 โวลต์ และเครื่องวัดและบันทึกค่าแรงดันไฟฟ้าวัดได้ 234 โวลต์ มีค่าความผิดพลาดเท่ากับ [(234-234)/234] x 100 = 0 %

รูปที่ 4.6 แสดงหน้าจอของเครื่องวัดขณะใช้งานในการวัดและบันทึกค่า



รูปที่ 4.7 แสดงหน้าจอของเครื่องวัดขณะใช้งานต่อกับคอมพิวเตอร์

ตารางที่ 4.1 ตารางแสดงผลการวัดค่าและบันทึกค่าด้วยอัตราการบันทึก 1 **ครั้งต่อนาที**

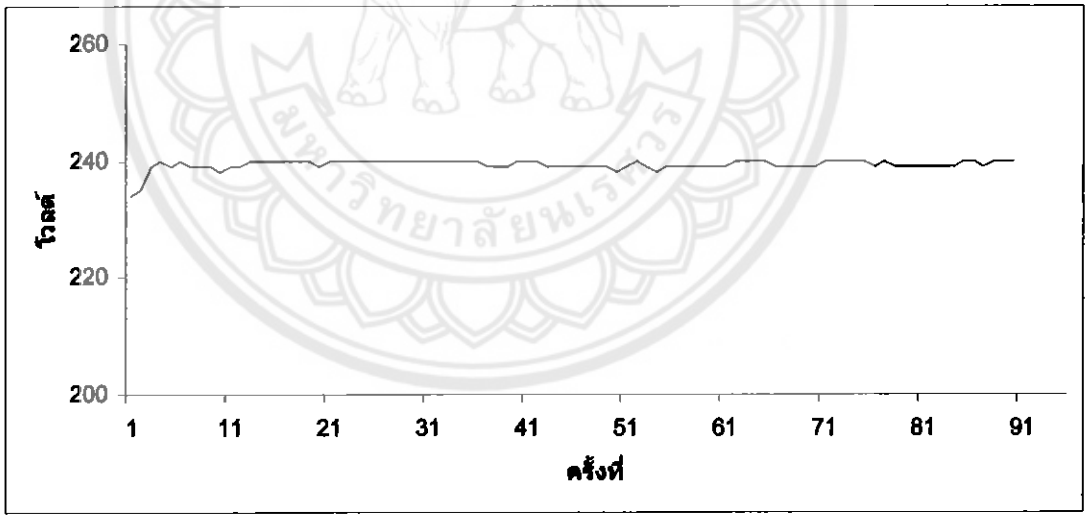
ครั้งที่	วันที่	เวลา	ค่าแรงดัน
1	16/09/07	2:46:10	234
2	16/09/07	2:47:00	235
3	16/09/07	2:48:00	239
4	16/09/07	2:49:00	240
5	16/09/07	2:50:00	239
6	16/09/07	2:51:00	240
7	16/09/07	2:52:00	239
8	16/09/07	2:53:00	239
9	16/09/07	2:54:00	239
10	16/09/07	2:55:00	238
11	16/09/07	2:56:00	239
12	16/09/07	2:57:00	239
13	16/09/07	2:58:00	240
14	16/09/07	2:59:00	240
15	16/09/07	3:00:00	240
16	16/09/07	3:01:00	240
17	16/09/07	3:02:00	240
18	16/09/07	3:03:00	240
19	16/09/07	3:04:00	240
20	16/09/07	3:05:00	239

ครั้งที่	วันที่	เวลา	ค่าแรงดัน
21	16/09/07	3:06:00	240
22	16/09/07	3:07:00	240
23	16/09/07	3:08:00	240
24	16/09/07	3:09:00	240
25	16/09/07	3:10:00	240
26	16/09/07	3:11:00	240
27	16/09/07	3:12:00	240
28	16/09/07	3:13:00	240
29	16/09/07	3:14:00	240
30	16/09/07	3:15:00	240
31	16/09/07	3:16:00	240
32	16/09/07	3:17:00	240
33	16/09/07	3:18:00	240
34	16/09/07	3:19:00	240
35	16/09/07	3:20:00	240
36	16/09/07	3:21:00	240
37	16/09/07	3:22:00	239
38	16/09/07	3:23:00	239
39	16/09/07	3:24:00	239
40	16/09/07	3:25:00	240
41	16/09/07	3:26:00	240
42	16/09/07	3:27:00	240
43	16/09/07	3:28:00	239
44	16/09/07	3:29:00	239
45	16/09/07	3:30:00	239
46	16/09/07	3:31:00	239
47	16/09/07	3:32:00	239
48	16/09/07	3:33:00	239
49	16/09/07	3:34:00	239
50	16/09/07	3:35:00	238

ครั้งที่	วันที่	เวลา	ค่าแรงดัน
51	16/09/07	3:36:00	239
52	16/09/07	3:37:00	240
53	16/09/07	3:38:00	239
54	16/09/07	3:39:00	238
55	16/09/07	3:40:00	239
56	16/09/07	3:41:00	239
57	16/09/07	3:42:00	239
58	16/09/07	3:43:00	239
59	16/09/07	3:44:00	239
60	16/09/07	3:45:00	239
61	16/09/07	3:46:00	239
62	16/09/07	3:47:00	240
63	16/09/07	3:48:00	240
64	16/09/07	3:49:00	240
65	16/09/07	3:50:00	240
66	16/09/07	3:51:00	239
67	16/09/07	3:52:00	239
68	16/09/07	3:53:00	239
69	16/09/07	3:54:00	239
70	16/09/07	3:55:00	239
71	16/09/07	3:56:00	240
72	16/09/07	3:57:00	240
73	16/09/07	3:58:00	240
74	16/09/07	3:59:00	240
75	16/09/07	4:00:00	240
76	16/09/07	4:01:00	239
77	16/09/07	4:02:00	240
78	16/09/07	4:03:00	239
79	16/09/07	4:04:00	239
80	16/09/07	4:05:00	239

ครั้งที่	วันที่	เวลา	ค่าแรงดัน
81	16/09/07	4:06:00	239
82	16/09/07	4:07:00	239
83	16/09/07	4:08:00	239
84	16/09/07	4:09:00	239
85	16/09/07	4:10:00	240
86	16/09/07	4:11:00	240
87	16/09/07	4:12:00	239
88	16/09/07	4:13:00	240
89	16/09/07	4:14:00	240
90	16/09/07	4:15:00	240

จะได้ค่าเฉลี่ยจากการวัด เท่ากับ 239.3556 โวลต์ จากการบันทึก 90 ครั้ง ส่วนค่าที่โวลต์มิเตอร์วัดได้คือ 239 โวลต์ มีค่าความผิดพลาดเท่ากับ [(239.3556-239)/239.3556] x 100 = 0.15 %



รูปที่ 4.8 กราฟแสดงค่าแรงดันจากการวัดและบันทึกค่า1 ครั้งต่อนาที (90 ครั้ง)

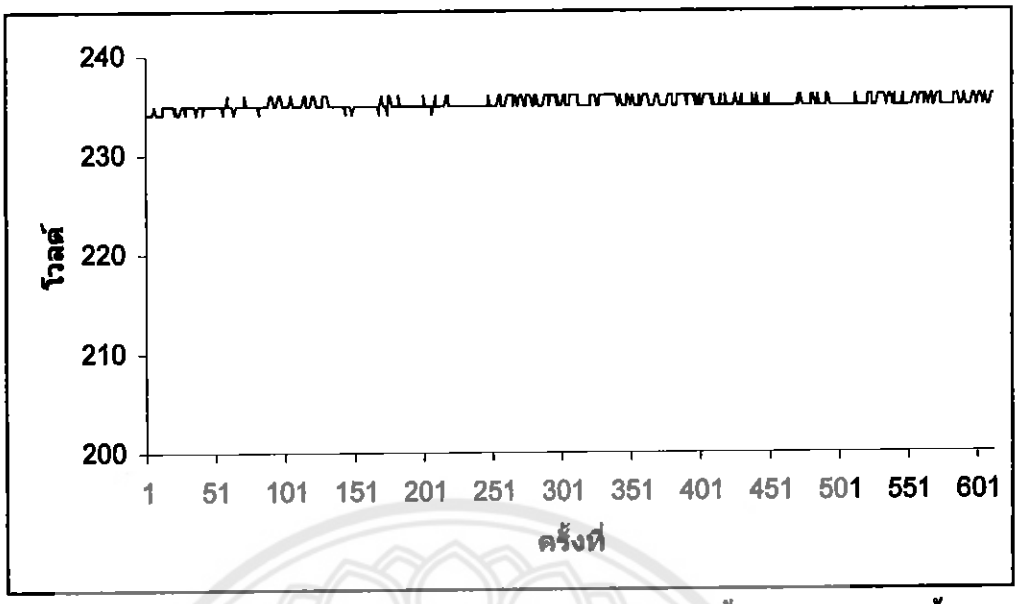
ตารางที่ 4.2 ตารางแสดงผลการวัดค่าและบันทึกค่าด้วยอัตราการบันทึก 1 ครั้งต่อวินาที

ครั้งที่	วันที่	เวลา	ค่าแรงดัน
1	16/09/07	14:53:31	234
2	16/09/07	14:53:32	234
3	16/09/07	14:53:33	234

ครั้งที่	วันที่	เวลา	ค่าแรงดัน
4	16/09/07	14:53:34	234
5	16/09/07	14:53:35	234
6	16/09/07	14:53:36	235
7	16/09/07	14:53:37	235
8	16/09/07	14:53:38	234
9	16/09/07	14:53:39	234
10	16/09/07	14:53:40	234
11	16/09/07	14:53:41	234
12	16/09/07	14:53:42	234
13	16/09/07	14:53:43	235
14	16/09/07	14:53:44	235
15	16/09/07	14:53:45	235
16	16/09/07	14:53:46	235
17	16/09/07	14:53:47	235
18	16/09/07	14:53:48	235
19	16/09/07	14:53:49	235
20	16/09/07	14:53:50	235
21	16/09/07	14:53:51	234
22	16/09/07	14:53:52	235
23	16/09/07	14:53:53	234
24	16/09/07	14:53:54	234
25	16/09/07	14:53:55	234
26	16/09/07	14:53:56	235
27	16/09/07	14:53:57	235
28	16/09/07	14:53:58	235
29	16/09/07	14:53:59	234
30	16/09/07	14:54:00	235
31	16/09/07	14:54:01	235
32	16/09/07	14:54:02	235
33	16/09/07	14:54:03	235

ครั้งที่	วันที่	เวลา	ค่าแรงดัน
34	16/09/07	14:54:04	235
35	16/09/07	14:54:05	235
36	16/09/07	14:54:06	235
37	16/09/07	14:54:07	234
38	16/09/07	14:54:08	235
39	16/09/07	14:54:09	235
40	16/09/07	14:54:10	235
41	16/09/07	14:54:11	235
42	16/09/07	14:54:12	234
43	16/09/07	14:54:13	235
44	16/09/07	14:54:14	235
45	16/09/07	14:54:15	235
46	16/09/07	14:54:16	235
47	16/09/07	14:54:17	235
48	16/09/07	14:54:18	235
49	16/09/07	14:54:19	235
50	16/09/07	14:54:20	235
51	16/09/07	14:54:21	235
52	16/09/07	14:54:22	235
53	16/09/07	14:54:23	235
54	16/09/07	14:54:24	235
55	16/09/07	14:54:25	235
56	16/09/07	14:54:26	234
57	16/09/07	14:54:27	235
58	16/09/07	14:54:28	235
59	16/09/07	14:54:29	236
60	16/09/07	14:54:30	235

จะได้ค่าเฉลี่ยจากการวัด เท่ากับ 234.7167 โวลต์ จากการบันทึก 60 ครั้ง ส่วนค่าที่โวลต์มิเตอร์วัดได้คือ 235 โวลต์ มีค่าความผิดพลาดเท่ากับ [(235-234.7167)/234.7167] x 100 = 0.12 %



รูปที่ 4.9 กราฟแสดงค่าแรงดันจากการวัดและบันทึกค่า 1 ครั้งต่อวินาที **(612 ครั้ง)**

จะได้ค่าเฉลี่ยจากการวัด เท่ากับ 235.25 โวลต์ จากการบันทึก 60 ครั้ง ส่วน**ค่าที่โวลต์มิเตอร์วัดได้คือ** 235 โวลต์ มีค่าความผิดพลาดเท่ากับ [(235.25-235)/235.25] x 100 = 0.11 %

บทที่ 5
บทสรุป

ในบทนี้จะเป็นการสรุปผลที่ได้จากการทดลองในโครงงานนี้ พร้อมเสนอแนะแนวทางในการนำโครงงานนี้ไปพัฒนาให้มีประสิทธิภาพมากยิ่งขึ้นต่อไปในอนาคต

5.1 สรุปผล

ในโครงงานนี้ ได้ออกแบบเครื่องวัดและบันทึกค่าแรงดันไฟฟ้าโดยใช้ PIC16F877 ซึ่งมีวงจรแปลงอะนาลอกเป็นดิจิตอลภายใน ในการรับรับค่าและแสดงผลของแรงดันไฟฟ้าและใช้ 24LC512 เป็นหน่วยความจำในการเก็บบันทึกค่าที่วัดได้ รวมออกแบบโปรแกรม DataLogger เพื่อใช้ในการดึงผลบันทึกข้อมูลแรงดันไฟฟ้าจากหน่วยความจำลงสู่คอมพิวเตอร์ สามารถแสดงผลการบันทึกค่าของข้อมูลที่รับเข้ามาในรูปแบบของไฟล์ Text

5.2 ความสามารถของเครื่องวัดและบันทึกค่าแรงดันไฟฟ้า

5.2.1 สามารถวัดแรงดันไฟฟ้าได้ในช่วง 0-220 VAC

5.2.2 สามารถบันทึกค่าแรงดันที่วัดได้ลงหน่วยความจำ ได้ 8,000 ค่า

5.2.3 สามารถตั้งค่าความถี่ในการบันทึกค่าแรงดันไฟฟ้าลงบอร์ดได้ตั้งแต่ 1 ครั้งต่อวินาที ถึง 1 ครั้งต่อ 99 ชั่วโมง

5.2.4 สามารถเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม เพื่อนำข้อมูลที่บันทึกไว้ลงคอมพิวเตอร์ได้

5.3 ประเมินผล

จากการดำเนินงานโครงงาน เมื่อเทียบกับวัตถุประสงค์ ได้ผลดังนี้

5.3.1 สามารถออกแบบอุปกรณ์เก็บข้อมูลโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวรับ-ส่งข้อมูล ให้กับคอมพิวเตอร์ และสามารถแสดงผลการทำงานต่างๆ ได้

5.3.2 สามารถรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ได้

5.3.3 สามารถใช้ DELPHI ควบคุมการทำงานของไมโครคอนโทรลเลอร์ได้

5.3.4 สามารถเก็บข้อมูลในระยะเวลาที่กำหนด และแสดงผลทางจอคอมพิวเตอร์ได้

5.4 ปัญหา ข้อเสนอแนะ และแนวทางแก้ไข

5.4.1 ปัญหาเกิดจากการต่อวงจรผิดพลาด ทำให้อุปกรณ์ได้รับเสียหาย จึงต้องทำการจัดซื้ออุปกรณ์มาใหม่ ทำให้งานที่ออกมาช้ากว่ากำหนด

5.4.2 ปัญหาเกิดจากการจ่ายแรงดันไฟฟ้าให้กับอุปกรณ์อิเล็กทรอนิกส์ ถ้าจ่ายแรงดันไม่เหมาะสมสัญญาณต่าง ๆจะไม่ได้ตามที่ต้องการ จะต้องจ่ายแรงดันที่เหมาะสมให้กับอุปกรณ์อิเล็กทรอนิกส์ถึงจะทำงานได้

5.4.3 ปัญหาเกิดจากการเลือกอุปกรณ์อิเล็กทรอนิกส์ไม่เหมาะสมกับขนาดวงจร และค่าของอุปกรณ์

5.5 แนวทางในการพัฒนาต่อไป

ผลที่ได้จากการทดลองในโครงงานนี้ สามารถนำไปประยุกต์ใช้กับงานที่ต้องการสังเกตความเปลี่ยนแปลงที่เกิดขึ้นในระบบโดยอาศัยการวัดตามช่วงเวลาที่กำหนด และเหมาะสมที่จะนำไปพัฒนาต่อให้สามารถใช้กับการวัดในรูปแบบต่างๆ เช่น การประยุกต์ใช้ในการวัดที่แรงดันไฟฟ้าสูง

เอกสารอ้างอิง

[1] กฤษฎา ใจเย็น, ณัฏฐพล วงศ์สุนทรชัย, ชัยวัฒน์ ลิ้มพรจิตรวิลัย. **PICBASIC PRO คอมไพเลอร์.** กรุงเทพมหานคร: อินโนเวตีฟ เอ็กเพอริเมนต์, พ.ศ. 2521

[2] ณัฐพล วงศ์สุนทรชัย, ชัยวัฒน์ ลิ้มพรจิตรลัย. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F877.** กรุงเทพมหานคร: อินโนเวตีฟ เอ็กเพอริเมนต์, พ.ศ. 2521

[3] ผศ. วิศรุต ศรีรัตนะ, ผศ.อัมผวัน ใจกล้า, อ.พิทยา ปานนิล. **ปฏิบัติการวิศวกรรมการวัดคุม 1.** พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: ลาดกระบัง, พ.ศ. 2548

[4] วัชรินทร์ เคารพ. **เรียนรู้และเข้าใจไมโครคอนโทรลเลอร์ PIC ด้วยภาษาเบสิก PicBasic Pro.** กรุงเทพมาหานคร: อีทีที, พ.ศ. 2547

[5] อดุลย์ กัลยาแก้ว. **อุปกรณ์อิเล็กทรอนิกส์และวงจร.** นนทบุรี: เจริญรุ่งเรืองการพิมพ์, พ.ศ. 2546

ภาคผนวก

ภาคผนวก ก

โปรแกรม PIC BASIC ใช้ควบคุมไมโครคอนโทรลเลอร์

1. ส่วนของการตั้งค่าการใช้งานไมโครคอนโทรลเลอร์

```
@ DEVICE HS_OSC
DEFINE OSC 4
DEFINE ADC_BITS 10
DEFINE ADC_CLOCK 3
define ADC_SAMPLEUS 500
TRISC=%11111000
TRISD=0
TRISA=%11111111
ADCON1 = 0
ADCON1.7 = 1
countx var word
DISP_buffer var word
digit var byte
disp var byte
Vin var word
PC_Vin var word
Log_time con  5   'Log every ? sec
Trig_min con 0
Trig_sec con 1
Trig_hour con 2
Trig_mode var byte
Trig_var  var byte
 DB0     VAR BYTE[8] ' Data byte array
 Time_Sec var byte
 Time_Min var byte
 Time_Hour var byte
 Time_Date var byte
 Time_Month var byte
```

```
Time_Year var byte
'   EEP
I2C_control var byte
I2C_address_16 var word
I2C_data var byte
SDat var byte[8]
EEP_Trig_Mode con 0
EEP_Trig_var con 1
'Reccord var word
LastRecordNumber var word
CurrentRecordNumber var word
Time_count var word
Time_old var byte
test_addr var word
test_dat var byte
VoltMSB var byte
VoltLSB var byte
tmp1 var byte
tmp2 var byte
tmp3 var byte
CMD var byte
SDA    VAR PORTC.4 ' DS1307 SDA pin #5
SCL    VAR PORTC.3 ' DS1307 SCL pin #6
TXD var PORTC.6
RXD var PORTC.7
LED var PORTB.5
SW_StartLog var PORTB.2
SW_StopLog var PORTB.1
SW_PC_connect var PORTB.0
input SW_StartLog
input SW_StopLog
input SW_PC_connect
```

```
PAUSE 1000 ' Allow power to stabilize on power-up
 ' GOSUB Write_1307 ' Write time & date on entry
test_dat=0
Trig_Mode=Trig_min
Trig_Var=1
eeprom 0,[EEP_Trig_Mode,EEP_Trig_Var]
Init:
read  EEP_Trig_Mode,Trig_Mode
read  EEP_Trig_Var,Trig_Var
gosub Find_Last_Rec
'disp=%10001100              'P
'disp=%11000110              'C
SW_CHK:
if SW_PC_connect = 0 then MAIN
if SW_StartLog = 0 then
read  EEP_Trig_Mode,Trig_Mode
read  EEP_Trig_Var,Trig_Var
goto Mode_RUN
endif
disp=%10111111
PORTD = disp
PORTC = %11111011
pause 2
disp=%10111111
PORTD = disp
PORTC = %11111101
pause 2
disp=%10111111
PORTD = disp
PORTC = %11111110
pause 2
```

```
goto SW_CHK
```

2. ส่วนของการเชื่อมต่อกับคอมพิวเตอร์

```
MAIN:
tmp3=0
MAIN_Loop:
if tmp3 = 0 then
disp=%10001100              'P
PORTD = disp
PORTC = %11111011
endif
if tmp3 = 1 then
disp=%11000110              'C
PORTD = disp
PORTC = %11111101
endif
tmp3=tmp3+1
if tmp3  >1 then  tmp3=0
if SW_StartLog=0 then Init
SERIN2
RXD,84,100,NoData,[WAIT("CMD:"),CMD,SDat[0],SDat[1],SDat[2],SDat[3],SDat[4],SDat[5],
SDat[6],SDat[7]]
if CMD = "1" then
pause 10
gosub ReadLog
endif
if CmD = "2" then
pause 10
serout2 TXD,84,["This time is "]
gosub Read_Time
endif
if CmD = "3" then
```

```
pause 10
gosub Write_Date
serout2 TXD,84,["Update time to "]
gosub Read_Time
endif
if CmD = "4" then
pause 10
gosub Write_Time
endif
if CmD = "5" then
pause 10
gosub Write_Trig
serout2 TXD,84,["Update interval to"]
gosub Read_Trig
endif
if CmD = "6" then
pause 10
serout2 TXD,84,["Interval is"]
gosub Read_Trig
endif
if CmD = "7" then
pause 10
serout2 TXD,84,["Start format... ",13,10]
gosub FormatEEP
serout2 TXD,84,[13,10,"Format complete !!!",13,10]
endif
if CmD = "8" then
pause 10
serout2 TXD,84,["Delete... ",13,10]
gosub QFormatEEP
serout2 TXD,84,[13,10,"Delete complete...",13,10]
endif
```

```
if CmD = "9" then
pause 10
serout2 TXD,84,["Ok.",13,10]
endif
NoData:
goto MAIN_Loop
```

3. ส่วนของการวัดค่า

```
Mode_RUN:
if SW_PC_connect=0 then SW_CHK
countx = countx+1
if countx >= 50 then
  ADCIN 0,Vin
  PC_Vin=PC_Vin
  Vin=Vin*49/100
  if Vin > 1 then Vin=Vin+5
  DISP_buffer=Vin
  countx = 0
  gosub Read_1307
    if Trig_mode = Trig_sec then
        if Time_old <> Time_Sec then
    Time_old = Time_Sec
    Trig_var=Trig_var-1
      if Trig_var = 0 then
      high LED
      gosub WriteLog
      serout2 TXD,84,["Rec No.",dec4 LastRecordNumber-1,": ",HEX2 Time_Date,"/",HEX2
Time_Month,"/",HEX2 Time_Year,",",HEX2 Time_Hour,":",HEX2 Time_Min,":",HEX2
Time_Sec]
      serout2 TXD,84,[",",DEC VoltMSB/10,".",dec VoltMSB//10,13,10]
      read EEP_Trig_var,Trig_var
```

```
        low LED
        endif
      endif
    endif
    if Trig_mode = Trig_min then
      if Time_old <> Time_Min then
      Time_old = Time_Min
      Trig_var=Trig_var-1
        if Trig_var = 0 then
        high LED
        gosub WriteLog
        serout2 PORTC.6,84,["Rec No.",dec LastRecordNumber-1," Addr:",HEX4
I2C_address_16-10,13,10]
        read EEP_Trig_var,Trig_var
        low LED
        endif
      endif
    endif
    if Trig_mode = Trig_hour then
      if Time_old <> Time_Hour then
      Time_old = Time_Hour
      Trig_var=Trig_var-1
        if Trig_var = 0 then
        high LED
        gosub WriteLog
        serout2 TXD,84,["Rec No.",dec LastRecordNumber-1," Addr:",HEX4 I2C_address_16-
10,13,10]
        read EEP_Trig_var,Trig_var
        low LED
        endif
      endif
```

```
   endif
endif
gosub DISPLAY
goto Mode_RUN
```

4. ส่วนของการแสดงผล

```
DISPLAY:
digit = DISP_buffer DIG 0
LookUp digit,[$C0,$F9,$A4,$B0,$99,$92,$82,$F8,$80,$90],disp
PORTD = disp
PORTC = %11111110
pause 2
digit = DISP_buffer DIG 1
LookUp digit,[$C0,$F9,$A4,$B0,$99,$92,$82,$F8,$80,$90],disp
disp=disp & %01111111
PORTD = disp
PORTC = %11111101
pause 2
digit = DISP_buffer DIG 2
LookUp digit,[$C0,$F9,$A4,$B0,$99,$92,$82,$F8,$80,$90],disp
PORTD = disp
PORTC = %11111011
pause 2
return
Read_1307:
 I2CREAD SDA,SCL,$D0,$00,[STR DB0\8]  ' Read 8 bytes from DS1307
 Time_Sec = DB0[0]
 Time_Min = DB0[1]
 Time_Hour = DB0[2]
 Time_Date = DB0[4]
 Time_Month = DB0[5]
 Time_Year = DB0[6]
```

```
return
EEP_write:
I2C_control=%10100000
I2CWRITE SDA,SCL,I2C_control,I2C_address_16,[I2C_data]
pause 5
return
EEP_read:
I2C_control=%10100000
I2Cread SDA,SCL,I2C_control,I2C_address_16,[I2C_data]
return
Address_CAL:
I2C_address_16=LastRecordNumber*10
return
Find_Last_Rec:
Search:
LastRecordNumber=1
Search_Loop:
gosub Address_CAL
gosub EEP_read
if I2C_address_16 > 65500 then Not_Found_Rec
if I2C_data = $FF then Found_Rec
LastRecordNumber=LastRecordNumber+1
goto Search_Loop
Not_Found_Rec:
LastRecordNumber=0
Found_Rec:
return
```

5. ส่วนของการบันทึกข้อมูล

```
WriteLog:
VoltMSB=DISP_buffer
gosub Read_1307
```

```
gosub Address_CAL
I2C_data=%00  'Mark as use block
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=Time_Date  'Date
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=Time_Month  'Month
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=Time_Year  'Year
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=Time_Hour  'Hour
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=Time_Min  'Minute
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=Time_Sec  'Second
gosub EEP_write

I2C_address_16=I2C_address_16+1
I2C_data=VoltMSB
gosub EEP_write
```

```
LastRecordNumber=LastRecordNumber+1
return
ReadLog:
LastRecordNumber=1
ReadLogLoop:
high LED
gosub Address_CAL
gosub EEP_read
if I2C_data <> %00 or I2C_address_16 > 65500 then End_ReadLog
I2C_address_16=I2C_address_16+1
gosub EEP_read
Time_Date=I2C_data  'Date

I2C_address_16=I2C_address_16+1
gosub EEP_read
Time_Month  = I2C_data 'Month

I2C_address_16=I2C_address_16+1
gosub EEP_read
Time_Year=I2C_data  'Year

I2C_address_16=I2C_address_16+1
gosub EEP_read
Time_Hour=I2C_data  'Hour

I2C_address_16=I2C_address_16+1
gosub EEP_read
Time_Min=I2C_data  'Minute

I2C_address_16=I2C_address_16+1
gosub EEP_read
Time_Sec=I2C_data  'Second
```

```
I2C_address_16=I2C_address_16+1
gosub EEP_read
VoltMSB=I2C_data
serout2 TXD,84,[dec4 LastRecordNumber,",",HEX2 Time_Date,"/",HEX2
Time_Month,"/",HEX2 Time_Year,",",HEX2 Time_Hour,":",HEX2 Time_Min,":",HEX2
Time_Sec]
serout2 TXD,84,[",",DEC VoltMSB/10,dec VoltMSB//10,13,10]
LastRecordNumber=LastRecordNumber+1
goto ReadLogLoop
End_ReadLog:
low LED
return
Read_Time:
gosub Read_1307
serout2 TXD,84,[HEX2 Time_Date,"/",HEX2 Time_Month,"/",HEX2 Time_Year,"  ",HEX2
Time_Hour,":",HEX2 Time_Min,":",HEX2 Time_Sec,13,10]
return
Write_Date:
 tmp1=SDat[0]
 tmp2=SDat[1]
 gosub combineByte
 Time_Date = tmp1
  tmp1=SDat[2]
 tmp2=SDat[3]
 gosub combineByte
 Time_Month = tmp1
  tmp1=SDat[4]
 tmp2=SDat[5]
 gosub combineByte
 Time_Year = tmp1
  I2CWRITE SDA,SCL,$D0,$04,[Time_Date,Time_Month,Time_Year] ' Write to DS1307
   return
```

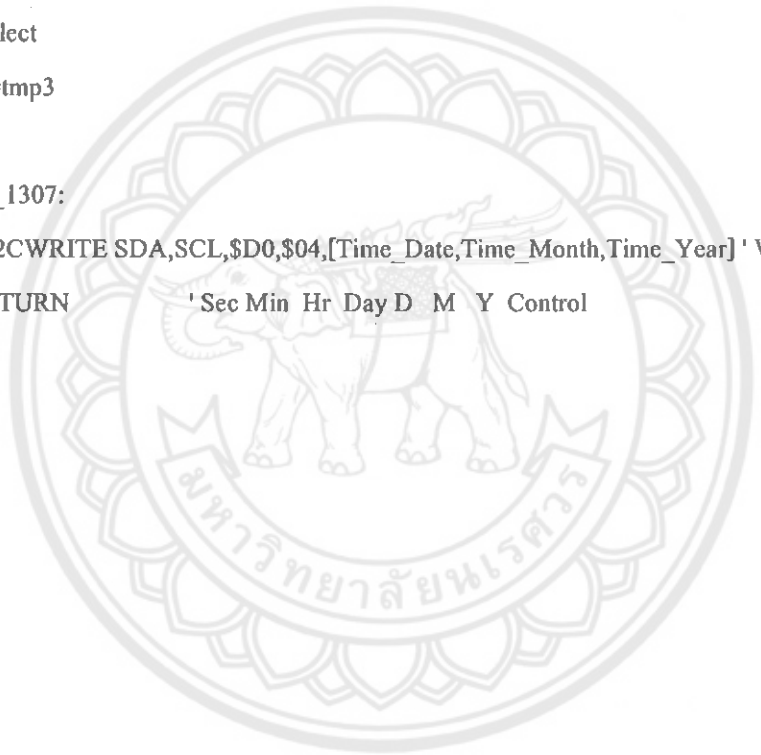
```
Write_Time:
 tmp1=SDat[0]
 tmp2=SDat[1]
 gosub combineByte
 Time_Hour = tmp1
 tmp1=SDat[2]
 tmp2=SDat[3]
 gosub combineByte
 Time_Min = tmp1
 tmp1=SDat[4]
 tmp2=SDat[5]
 gosub combineByte
 Time_Sec = tmp1
 I2CWRITE SDA,SCL,$D0,$00,[Time_Sec,Time_Min,Time_Hour] ' Write to DS1307
return
Write_Trig:
 tmp1=SDat[0]
 gosub Char2DEC
 Trig_mode = tmp1
 tmp1=SDat[1]
 gosub Char2DEC
 Trig_var = tmp1*10
 tmp1=SDat[2]
 gosub Char2DEC
 Trig_var = tmp1+Trig_var
 write  EEP_Trig_Mode,Trig_Mode
 write  EEP_Trig_var,Trig_var
return
Read_Trig:
 read  EEP_Trig_Mode,Trig_Mode
 read  EEP_Trig_var,Trig_var
if Trig_mode = Trig_sec then serout2 TXD,84,[" ",DEC Trig_var," Second",13,10]
```

```
if Trig_mode = Trig_min then serout2 TXD,84,[" ",DEC Trig_var," Minute",13,10]
if Trig_mode = Trig_hour then serout2 TXD,84,[" ",DEC Trig_var," Hour",13,10]
return
FormatEEP:
high LED
I2C_address_16=10
I2C_data=$FF
gosub EEP_write
FormatEEP_loop:
I2C_address_16=I2C_address_16+10
if I2C_address_16 > 65500 then Format_EEP_end
I2C_data=$FF
gosub EEP_write
if I2C_address_16 // 100=0 then serout2 TXD,84,[">"]
goto FormatEEP_loop
Format_EEP_end:
LOW LED
return
QFormatEEP:
 high LED
I2C_address_16=10
I2C_data=$FF
gosub EEP_write
serout2 TXD,84,["REC:",DEC4 I2C_address_16/10,13,10]
QFormatEEP_loop:
I2C_address_16=I2C_address_16+10
gosub EEP_read
if I2C_address_16 > 65500 or I2C_data = $FF then Format_EEP_end
I2C_data=$FF  'Year
gosub EEP_write
serout2 TXD,84,["REC:",DEC4 I2C_address_16/10,13,10]
goto QFormatEEP_loop
```

```
QFormat_EEP_end:
LOW LED
return
combineByte:
select case tmp1
  case "0"
    tmp3=$0
  case "1"
    tmp3=$10
  case "2"
    tmp3=$20
  case "3"
    tmp3=$30
  case "4"
    tmp3=$40
  case "5"
    tmp3=$50
  case "6"
    tmp3=$60
  case "7"
    tmp3=$70
  case "8"
    tmp3=$80
  case "9"
    tmp3=$90
end select
tmp1=tmp3
select case tmp2
  case "0"
    tmp3=$0
  case "1"
    tmp3=$1
```

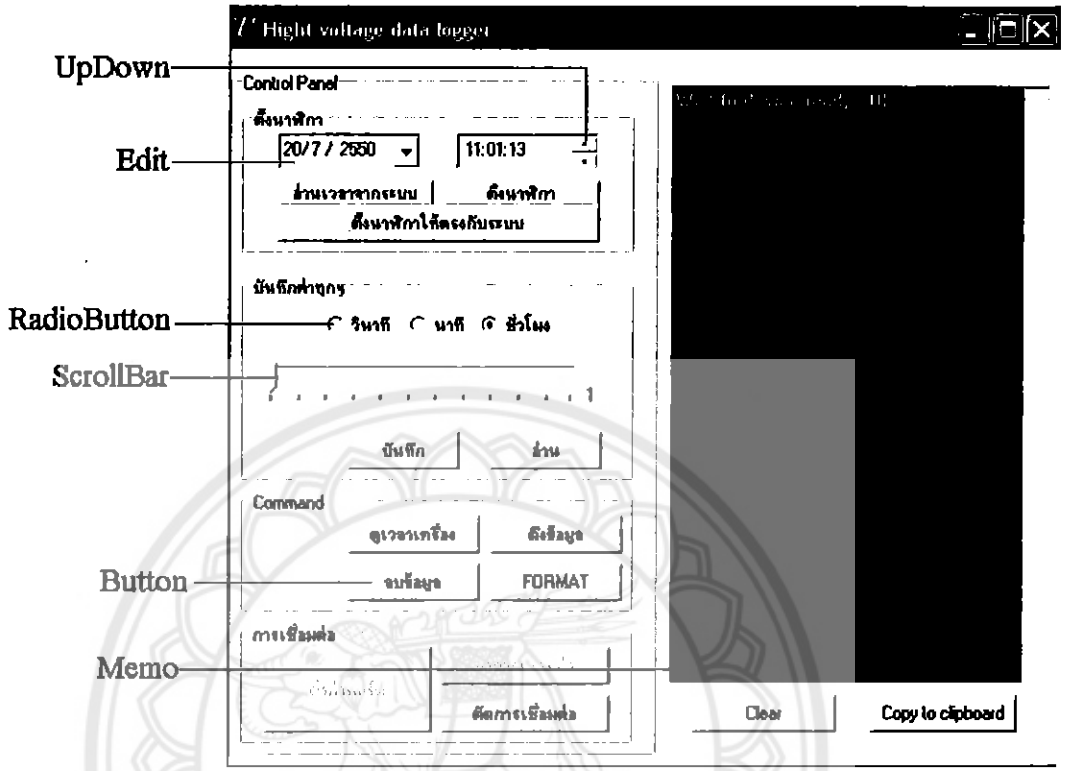
```
    case "2"
      tmp3=$2
    case "3"
      tmp3=$3
    case "4"
      tmp3=$4
    case "5"
      tmp3=$5
    case "6"
      tmp3=$6
    case "7"
      tmp3=$7
    case "8"
      tmp3=$8
    case "9"
      tmp3=$9
end select
tmp2=tmp3
tmp1=tmp1|tmp2
return
Char2DEC:
select case tmp1
  case "0"
    tmp3=0
  case "1"
    tmp3=1
  case "2"
    tmp3=2
  case "3"
    tmp3=3
  case "4"
    tmp3=4
```

```
    case "5"
       tmp3=5
    case "6"
       tmp3=6
    case "7"
       tmp3=7
    case "8"
       tmp3=8
    case "9"
       tmp3=9
end select
tmp1=tmp3
return
Write_1307:
     I2CWRITE SDA,SCL,$D0,$04,[Time_Date,Time_Month,Time_Year] ' Write to DS1307
   RETURN              ' Sec Min  Hr  Day D   M   Y  Control
```



ภาคผนวก ข

โปรแกรม DELPHI



Edit abl

เป็นคอมโพเนนต์ที่ใช้สร้างส่วนติดต่อกับผู้ใช้ ซึ่งสามารถใช้แสดงข้อความและรับข้อมูลที่ผู้ใช้ป้อนเข้ามา

Properties ที่สำคัญมีดังนี้

Text	ใช้สำหรับเก็บข้อความที่อยู่ใน Edit
Auto Select	ใช้กำหนดให้ข้อความใน Edit ถูกเลือกโดยอัตโนมัติเมื่อมีการกด Tab ไปยัง Edit
Hide Select	ถ้าเลือกเป็น False จะทำให้ข้อความที่ถูกเลือกใน Edit ยังคงแสดง Highlight อยู่ แม้ว่าจะย้ายการทำงานไปยังคอมโพเนนต์อื่นแล้วก็ตาม
SelLengh	เป็นค่าจำนวนอักษรที่ถูกเลือกและสามารถใช้กำหนดจำนวนอักษรที่ถูกเลือกได้
SelStart	เป็นตำแหน่งเริ่มต้นของตัวอักษรที่ถูกเลือก เริ่มนับตำแหน่งแรกเป็น 0
SelText	เป็นค่าตัวอักษรใน Edit เฉพาะส่วนที่ถูกเลือก

Memo

เป็นคอมโพเนนต์ที่สามารถรับและแสดงข้อความได้หลายๆบรรทัด

Properties ที่สำคัญมีดังนี้

Text เป็นข้อความทั้งหมดที่อยู่ใน Memo โดยแต่ละบรรทัดจะคั่นด้วย #B#10

Modified ไม่แสดงใน Object Inspector แต่เรียกใช้ได้ในขณะรันแอพพลิเคชัน โดยนำค่าจาก Properties นี้ มาใช้ในการตรวจสอบว่ามีการแก้ไขข้อความใน Memo หรือไม่

ScrollBars ใช้กำหนดการแสดง Scrollbars ใน Memo มี 4 ค่าให้เลือกคือ

- ssNone ไม่แสดง Scrollbars
- ssHorizontal แสดง Scrollbars ในแนวนอนเท่านั้น
- ssVertical แสดง Scrollbars ในแนวตั้งเท่านั้น
- ssBoth แสดง Scrollbars ทั้งแนวตั้งและแนวนอน

WantReturns กำหนดให้การกด Enter ใน Memo เป็นการขึ้นบรรทัดใหม่

WantTabs กำหนดให้กด Tab ใน Memo เป็นการเว้นแท็บถ้าเป็น True แต่ถ้าเป็น False จะเป็นการย้ายการทำงานไปยังคอมโพเนนต์อื่น

Lines เก็บค่าข้อมูลที่อยู่ใน Memo ถ้าคลิกที่ปุ่มอีลปในช่องทางขวาของ Properties จะปรากฏไดอะล็อกบ็อกซ์ String list editor สำหรับป้อนและแก้ไขข้อความ

Button

เป็นคอมโพเนนต์ที่มีลักษณะเป็นปุ่มกดสำหรับให้ผู้ใช้คลิกเพื่อสั่งในการทำงาน

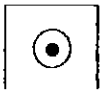
Properties ที่สำคัญมีดังนี้

Caption เป็นข้อความบนปุ่ม

Cancel เป็น Properties ที่กำหนดได้เป็น True หรือ False ถ้าเป็น True จะทำให้เกิดอีเวนต์ OnClick เมื่อมีการกด Esc

Default เป็น Properties ที่กำหนดได้เป็น True หรือ False ถ้าเป็น True จะทำให้เกิดอีเวนต์ OnClick เมื่อมีการกด Enter

ModalResult เป็น Properties ที่ใช้เก็บค่า

RadioButton 

เป็นคอมโพเนนต์ที่ใช้สำหรับกำหนดตัวเลือกคล้ายกับการเช็คบ๊อกซ์ แต่การใช้งาน RadioButton จะใช้ตัวเดียวเดี่ยวๆไม่ได้ ในการใช้งานต้องมี RadioButton อย่างน้อย 2 ตัวขึ้นไป เพราะเมื่อเลือกตัวใดตัวหนึ่งจะทำให้อีกตัวที่เคยเลือกไว้เปลี่ยนเป็นไม่เลือก

Properties ที่สำคัญมีดังนี้

Caption สำหรับกำหนดข้อความของ RadioButton

Checked เป็น Properties ซึ่งจะบอกว่า RadioButton นั้นถูกเลือกหรือไม่

ScrollBar

เป็นคอมโพเนนต์ที่ใช้การความคุมการเลื่อนตำแหน่งภายในขอบเขตที่กำหนด

Properties ที่สำคัญมีดังนี้

Kind ใช้สำหรับกำหนดลักษณะของ Scrollbar ว่าจะให้แสดงแนวตั้งหรือแนวนอน

ssHorizontal แสดง Scrollbar ในแนวนอน

ssVertical แสดง Scrollbar ในแนวตั้ง

Position สำหรับกำหนดตำแหน่งปัจจุบันของปุ่มใน Scrollbar ซึ่งจะเป็นค่าที่อยู่ระหว่าง Min กับ Max

LargeChange ใช้กำหนดค่าที่เปลี่ยนให้กับ Position เมื่อมีการคลิกที่บาร์

SmallChange ใช้กำหนดค่าที่เปลี่ยนให้กับ Position เมื่อมีการคลิกปุ่มที่ปลายบาร์

Min ใช้กำหนดค่าต่ำสุดของ Position

Max ใช้กำหนดค่าสูงสุดของ Position

UpDown

เป็นคอมโพเนนต์ที่แสดงปุ่มลูกศรขึ้นและลง เพื่อใช้ในการเพิ่มและลดค่าโดยจะใช้ร่วมกับคอมโพเนนต์ตัวอื่นๆ

Properties ที่สำคัญมีดังนี้

Associate ใช้ในการกำหนดคอมโพเนนต์ที่จะนำมาใช้ร่วมกับ UpDown

AlignButton สำหรับกำหนดตำแหน่งของ UpDown เมื่อใช้ร่วมกับคอมโพเนนต์ตัวอื่นมีค่าในการกำหนด 2 ค่าดังนี้

- udLeft กำหนดให้ UpDown อยู่ทางซ้าย
- udRight กำหนดให้ UpDown อยู่ทางขวา

Orientation สำหรับกำหนดทิศทางของลูกศร มีค่ากำหนด 2 ค่าดังนี้

- udHorizontal ลูกศรจะชี้ไปซ้ายและขวา
- udVertical ลูกศรจะชี้ขึ้นและลง

Min สำหรับกำหนดค่าต่ำสุดที่เลื่อนไปได้

Max สำหรับกำหนดค่าสูงสุดที่เลื่อนไปได้

Position เป็น Properties ที่เก็บค่าปัจจุบันของการใช้ UpDown

ประวัติผู้เขียนโครงงาน

ชื่อ นายแสนเพชร์ มัคสัมพันธ์

ภูมิลำเนา 268 หมู่ 11 ต.บ้านโข้ง อ.อู่ทอง จ.สุพรรณบุรี 72160

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนสระกระโจมโสภณพิทยา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

E-mail : jaja_phet@hotmail.com

ชื่อ นายเฉลิมชัย ทิพหมึก

ภูมิลำเนา 45 หมู่ 20 ต.เวียง อ.เทิง จ.เชียงราย 57160

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนสามัคคีวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

E-mail : nayno-ok@hotmail.com