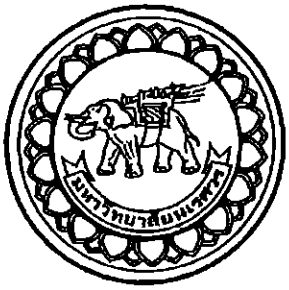


# Computer Based Data Logger


นายคมสัน อารีเอื้อ รหัส 45380260

นายวิทยา แสนธนโชคชัย รหัส 46380164



i 5081415 e.2

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 27 พ.ย. 2550
เลขทะเบียน.....
เลขเรียกหนังสือ.....
มหาวิทยาลัยนเรศวร

ปร.
ศ152ค
2549.


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549

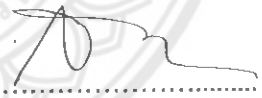
# ใบรับรองโครงงานวิจัย

| | |
|---|---|
| **หัวข้อโครงงาน** | Computer Based Data Logger |
| **ผู้ดำเนินโครงงาน** | **นายคมสัน อารีเอื้อ รหัส 45380260** |
| | นายวิทยา แสนธนโชคชัย รหัส 46380164 |
| **อาจารย์ที่ปรึกษา** | ดร.สมยศ เกียรติวนิชวิไล |
| **สาขาวิชา** | วิศวกรรมไฟฟ้า |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2549 |

.....

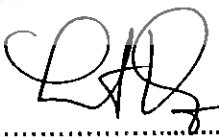
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิจัย



.....ประธานกรรมการ

(ดร.สมยศ เกียรติวนิชวิไล)



.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุรเชษฐ์ กานต์ประชา)

ลาศึกษาต่อ

.....กรรมการ

(อ.พนัส นัถฤทธิ์)

| | | | |
|---|---|---|---|
| **หัวข้อโครงงาน** | Computer Based Data Logger | | |
| **ผู้ดำเนินโครงงาน** | นายคมสัน | อารีเอื้อ | รหัส 45380260 |
| | นายวิทยา | แสนธนไชคชัย | รหัส 46380164 |
| **อาจารย์ที่ปรึกษา** | ดร.สมยศ เกียรติวนิชวิไล | | |
| **สาขา** | วิศวกรรมไฟฟ้า | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | |
| **ปีการศึกษา** | 2549 | | |


## บทคัดย่อ

Computer Based Data Logger เป็นการนำพื้นฐานในการวัดมาประยุกต์ใช้ การล็อกข้อมูล คือ การวัดและบันทึก ของสัญญาณทางไฟฟ้าภายในคาบของเวลา เครื่องมือนี้สามารถใช้ในการเก็บสัญญาณทางไฟฟ้าในรูปแบบของ แรงดันไฟฟ้าและกระแสไฟฟ้า CP-PIC 16F877 จะเป็นตัวรับสัญญาณทางไฟฟ้า หน่วยประมวลผลหลัก A/D converter การแปลงสัญญาณ และ Real time clock Computer Based Data Logger สามารถแสดงผล และบันทึก บนคอมพิวเตอร์โดยมีการเชื่อมต่อผ่านทางพอร์ตอนุกรม RS-232 โครงงานนี้ Microsoft Visual Basic 6.0 เป็นตัวควบคุม

ในโครงงานนี้ไมโครคอนโทรลเลอร์ถูกใช้เป็นตัวควบคุม นอกจากนี้ ผู้ใช้สามารถติดต่อกับไมโครคอนโทรลเลอร์ ผ่านทางโปรแกรมคอมพิวเตอร์ ซึ่งสามารถเลือกการทำงานและปรับค่าหน่วงเวลา ผลของโครงงานนี้สามารถแสดงข้อมูลบน Visual Basic 6.0 และบันทึกค่าที่แสดงบนไฟล์ Notepad

| | | | |
|---|---|---|---|
| **Project title** | Computer Based Data Logger | | |
| **Name** | Mr.Khomsan | Ari-uea | ID. 45380260 |
| | Mr.Wittaya | Santanachokchai | ID. 46380164 |
| **Project advisor** | Somyot Kiattivanichvilai, Ph.D. | | |
| **Major** | Electrical Engineering. | | |
| **Department** | Electrical and Computer Engineering. | | |
| **Academic year** | 2006 | | |


## ABSTRACT

Computer Based Data Logger is common measurement application. Data logging is the measurement and recording of electrical signal over period of time. Electrical signal can be saved in this equipment are voltage and current. Signals obtained from CP-PIC 16F877, main processor, A/D converter and Real time clock. Computer Based Data Logger can be displayed and record on computer via RS-232 port. This project controlled by Microsoft Visual Basic 6.0.

In this project, microcontroller is used as a controller. Not only that, the microcontroller is able to interface with user via computer program that the operating modes can be selected by user and the control sampling time are also be adjusted. The result of this project can be displayed on Visual basic 6.0 and also recorded on notepad file.

# กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ในครั้งนี้ สำเร็จลุล่วงไปด้วยดีโดยได้รับความช่วยเหลือและให้คำแนะนำ จาก ดร.สมยศ เกียรติวนิชวิไล ที่เป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ในครั้งนี้ และขอขอบคุณอาจารย์พนัส นัถฤทธิ์ ที่ได้กรุณาให้แนวความคิด ช่วยชี้แนะแนวทางในการทำโครงงานตลอดจนกรุณาเอื้อเฟื้อเอกสารต่างๆที่เกี่ยวข้องกับโครงงานนี้ อีกทั้งยังช่วยแนะนำแหล่งข้อมูลในการค้นคว้าข้อมูลเพิ่มเติม ทำให้เป็นประโยชน์ต่อการทำโครงงานของผู้จัดทำเป็นอย่างมาก

ท้ายสุดขอขอบพระคุณ คุณพ่อ คุณแม่ ญาติพี่น้อง ที่คอยดูแล คอยเป็นกำลังใจและเป็นผู้สนับสนุนในด้านต่างๆ มาโดยตลอดในการทำปริญญานิพนธ์นี้ และขอขอบคุณบุคคลต่างๆที่ไม่ได้กล่าวถึงรวมถึงแหล่งข้อมูลที่เอื้อต่อการทำปริญญานิพนธ์ในครั้งนี้ด้วย

คณะผู้จัดทำ
นายคมสัน อารีเอื้อ
นายวิทยา แสนธนโชคชัย

# 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 ส่วนประกอบและข้อจำกัดต่างๆ ของบอร์ด CP-PIC16F877..... 10

2.6 ลักษณะทางฮาร์ดแวร์..... 11

2.7 หลักการสื่อสารข้อมูล (Data Communication)..... 18

2.8 ไอซีฐานเวลาจริง DS1307..... 20

2.9 วงจร A/D Converter..... 23

บทที่ 3 การออกแบบและสร้าง Computer Based Data Logger

3.1 ส่วนของคอมพิวเตอร์..... 24

3.2 ส่วนของไมโครคอนโทรลเลอร์ (CP-PIC 16F877)..... 25

# สารบัญ (ต่อ)

หน้า

3.3 การต่อวงจรทั้งหมดของแหล่งจ่ายแรงดันไฟฟ้า (Hardware).....26

บทที่ 4 ผลการทดลอง.....29

บทที่ 5 บทสรุป

5.1 สรุปผล.....34

5.2 สรุปการเปรียบเทียบค่าหน่วงเวลาและกราฟที่ได้จากการวัดด้วย Oscilloscope.....34

5.3 สรุปจำนวนข้อมูลที่รับเข้ามาต่อนาทีในไฟล์ Text.....38

5.4 ประเมินผล.....39

5.5 ปัญหา ข้อเสนอแนะ และแนวทางแก้ไข.....40

5.6 แนวทางในการพัฒนาต่อไป.....40

เอกสารอ้างอิง.....41

ภาคผนวก.....42

ภาคผนวก ก.....43

ภาคผนวก ข.....49

ภาคผนวก ค.....63

ประวัติผู้เขียนโครงงาน.....78

# สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงค่าที่กำหนดในรีจีสเตอร์..... 9

2.2 ข้อกำหนดมาตรฐาน RS-232..... 13

2.3 DB-9 ตัวผู้ทางด้านคอมพิวเตอร์..... 14

2.4 DB-25 ตัวผู้ทางด้านคอมพิวเตอร์.....15

2.5 หน้าที่ของสัญญาณต่างๆ.....15

2.6 การเซตค่าพอร์ตอนุกรม.....17

2.7 แสดงรีจีสเตอร์ที่ใช้เก็บค่าต่างๆ.....22

4.1 ตารางเปรียบเทียบค่าที่วัดได้จาก Multimeter เทียบกับการแสดงผลใน Visual Basic.....29

# 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 แสดงส่วนประกอบของบอร์ด PIC 16F877..... 11
2.8 แสดงการต่อใช้งาน RS-232..... 12
2.9 แสดงลักษณะของหัวต่อแบบ DB-9..... 13
2.10ลักษณะของหัวต่อแบบ DB-25..... 14
2.11แสดงการเชื่อมต่อเพื่อสื่อสารทางเดียว..... 16
2.12แสดงการเชื่อมต่อเพื่อการสื่อสารสองทาง..... 16
2.13RS-232 Wave Form..... 17
2.14แสดงส่วนประกอบของการสื่อสารข้อมูล..... 18
2.15การส่งข้อมูลแบบขนาน (Parallel) ..... 19
2.16การส่งข้อมูลแบบอนุกรม (Serial) ..... 20
2.17แผนผังหน่วยความจำ..... 22
2.18การทำงานของ A/D Converter..... 23
3.1 แสดงบล็อกไดอะแกรมของ Computer Based Data Logger..... 24
3.2 แสดงการทำงานของโปรแกรม Visual Basic 6.0..... 25
3.3 แสดงบอร์ดไมโครคอนโทรลเลอร์ CP-PIC 16F877 V.1..... 25
3.4 Flow-Chart แสดงการทำงานของ Computer Based Data Logger..... 26
3.5 แสดงวงจรรวมที่ใช้ในการจ่ายแรงดันเข้ากับบอร์ด..... 27
3.6 แสดงวงจรจ่ายไฟเลี้ยง +/- 15 Vdc ให้กับไอซี LM358..... 27
3.7 แสดงวงจรจ่ายแรงดัน 0 - 5 Vdc โดยใช้ไอซีเรกกูเลเตอร์เบอร์ 7805..... 28
3.8 แสดงวงจรเปลี่ยนแรงดันเป็นกระแส-เปลี่ยนกระแสเป็นแรงดัน..... 28
4.1 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1ms..... 30
4.2 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 10ms..... 30

# สารบัญรูป(ต่อ)


รูปที่ หน้า

4.3 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 100ms.....31

4.4 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1000ms.....32

4.5 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 2000ms.....32

4.6 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 5000ms.....33

5.1 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1ms.....34

5.2 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 10ms.....35

5.3 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 100ms.....36

5.4 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1000ms.....36

5.5 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 2000ms.....37

5.6 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 5000ms.....38

# 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 จากนั้นไมโครคอนโทรลเลอร์จะทำการส่งข้อมูลไปยังคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม เพื่อทำการเก็บบันทึกข้อมูลโดยใช้โปรแกรม Visual Basic 6.0 ในการวิเคราะห์และแสดงผลผ่านทางหน้าจอคอมพิวเตอร์

## 1.2 วัตถุประสงค์

1.2.1 ออกแบบอุปกรณ์เก็บข้อมูลโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวรับ-ส่งข้อมูลให้กับคอมพิวเตอร์และสามารถแสดงผลการทำงานต่างๆ

1.2.2 การรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

1.2.3 ศึกษาการทำงานของ Visual Basic 6.0 เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์

1.2.4 ศึกษาการเก็บข้อมูลในระยะเวลาที่กำหนด และแสดงผลทางจอคอมพิวเตอร์

## 1.3 ขอบข่ายโครงงาน

โครงงานนี้เป็นโครงงานที่เกี่ยวกับการแสดงผลแล้วเก็บบันทึกข้อมูล ผ่านคอมพิวเตอร์ ซึ่งมีหัวข้อหลักที่จะทำการศึกษาเพื่อสร้างชิ้นงาน ดังนี้

1.3.1 ศึกษาการทำงานร่วมกันระหว่าง คอมพิวเตอร์กับไมโครคอนโทรลเลอร์

1.3.2 สามารถเก็บบันทึกข้อมูลที่เป็นค่าสัญญาณทางไฟฟ้า เช่น กระแสและแรงดันได้

1.3.3 เขียนโปรแกรมที่ใช้ในการรับส่งข้อมูลที่เป็นค่าสัญญาณทางไฟฟ้า

## 1.4 กิจกรรมการดำเนินงาน

| กิจกรรม | ปี 2548 | ปี 2549 | | | | ปี 2550 | |
|---|---|---|---|---|---|---|---|
| | พ.ย. - ธ.ค. | ม.ค. - ก.พ. | มี.ค. - มิ.ย. | ก.ค. - ก.ย. | ต.ค. - ธ.ค. | ม.ค. - ก.พ. | มี.ค. |
| 1. รวบรวมข้อมูลและเอกสาร | ← | — | → | | | | |
| 2. ออกแบบเครื่องเก็บข้อมูล | | | ← | → | | | |
| 3. ออกแบบโปรแกรมคอมพิวเตอร์ที่ใช้บันทึกข้อมูล | | | | ← | → | | |
| 4. สร้างอุปกรณ์และเขียนโปรแกรม | | | | | ← | → | |
| 5. ทดสอบ และแก้ไขชิ้นงาน | | | | | | ← | → |
| 6. สรุปผลการทดลองและวิจารณ์ | | | | | | | ← → |

## 1.5 ผลที่คาดว่าจะได้รับ

1.5.1 อุปกรณ์รับ-ส่งข้อมูลที่สามารถบันทึกข้อมูลที่รับมาได้

1.5.2 สามารถใช้ Visual Basic 6.0 ควบคุมการทำงานของไมโครคอนโทรลเลอร์

1.5.3 นำความรู้ที่ได้รับไปประยุกต์ใช้กับงานทางด้านไมโครคอนโทรลเลอร์

## 1.6 งบประมาณ

| | |
|---|---|
| 1.6.1 ค่าอุปกรณ์อิเล็กทรอนิกส์ | 1,000 บาท |
| 1.6.2 กระดาษ และถ่ายเอกสาร | 200 บาท |
| 1.6.3 ค่าอุปกรณ์คอมพิวเตอร์ | 400 บาท |
| 1.6.4 ค่าจัดทำรูปเล่ม | 200 บาท |
| 1.6.5 หนังสือประกอบการทำโครงงาน | 200 บาท |
| รวมเป็นเงิน | 2,000 บาท |
| | (สองพันบาทถ้วน) |

# บทที่ 2
# หลักการและทฤษฎี

## 2.1 การเชื่อมต่อกับพอร์ตอนุกรม

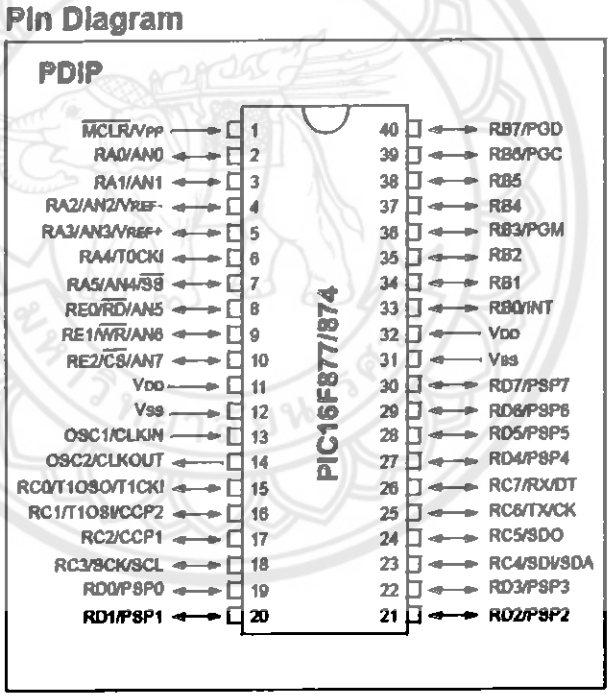
เป็นวิธีการอินเตอร์เฟสของคอมพิวเตอร์วิธีหนึ่งเพื่อทำการเชื่อมต่อกับอุปกรณ์อื่นๆ ซึ่งวิธีนี้จะเป็นการติดต่อแบบอนุกรมผ่านทางพอร์ตอนุกรม ซึ่งการเชื่อมต่อแบบนี้จะมีความเร็วในการเชื่อมต่อช้ากว่าพอร์ตขนาน แต่อุปกรณ์ที่ใช้พอร์ตขนานจะมีราคาแพงกว่า และจะต้องใช้สายสัญญาณมากกว่า การส่งข้อมูลทางพอร์ตอนุกรมนั้นในเครื่องคอมพิวเตอร์จะใช้มาตรฐาน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

## 2.2 สถาปัตยกรรมและโครงสร้างทางด้านฮาร์ดแวร์ ของ PIC16F877

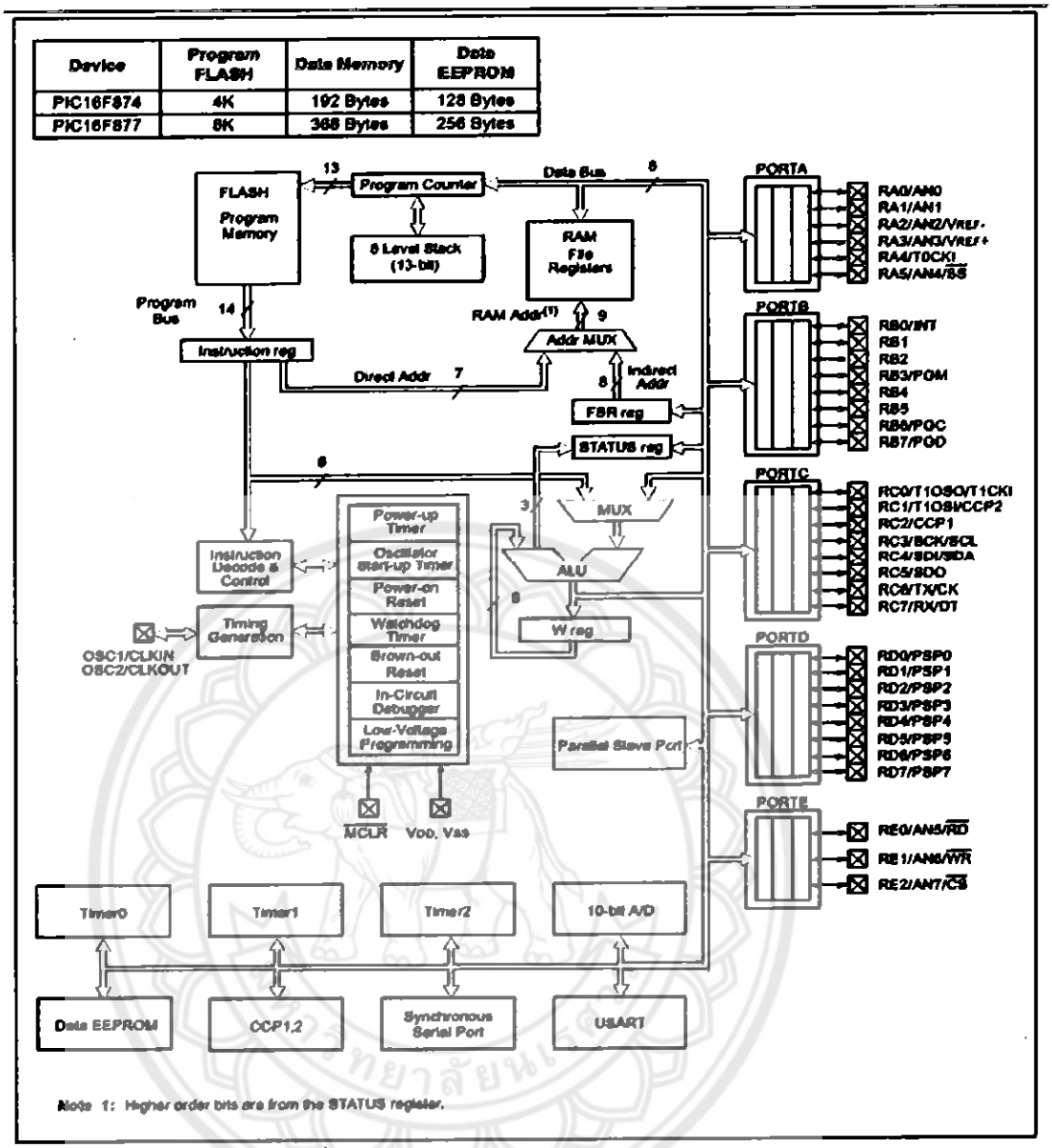
PIC16F877 เป็น CPU ของบริษัท Microchip Technology ซึ่งเป็นผู้ผลิต CPU ตระกูล PIC โดยในเบอร์ 16F877 เป็นเบอร์ที่มีความสามารถสูง ประกอบไปด้วยฟังก์ชันการทำงาน สรุปคุณสมบัติดังนี้

- 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

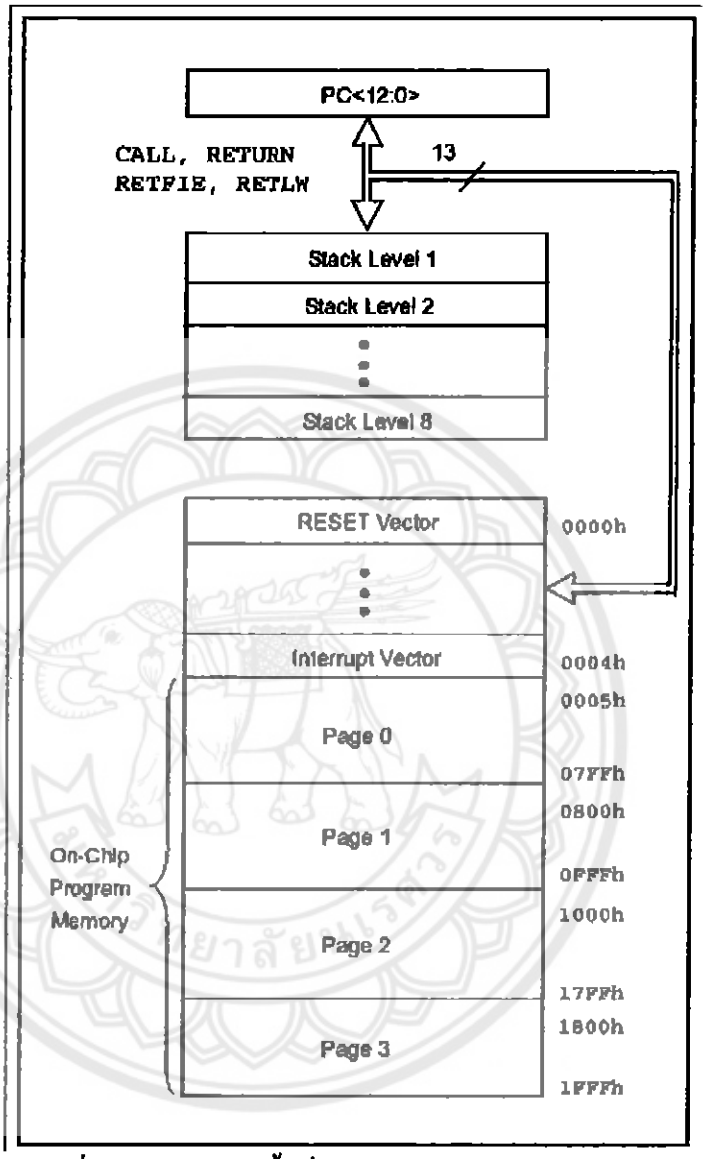
## 2.3 การจัดสรรพื้นที่หน่วยความจำและรีจิสเตอร์ต่างๆ

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 จะเห็นว่ามีพื้นที่ของ Stack 8 ระดับ และ หน่วยความจำโปรแกรมแบ่งออกเป็น 4 Page (8 kwords) ซึ่งพื้นที่ใน

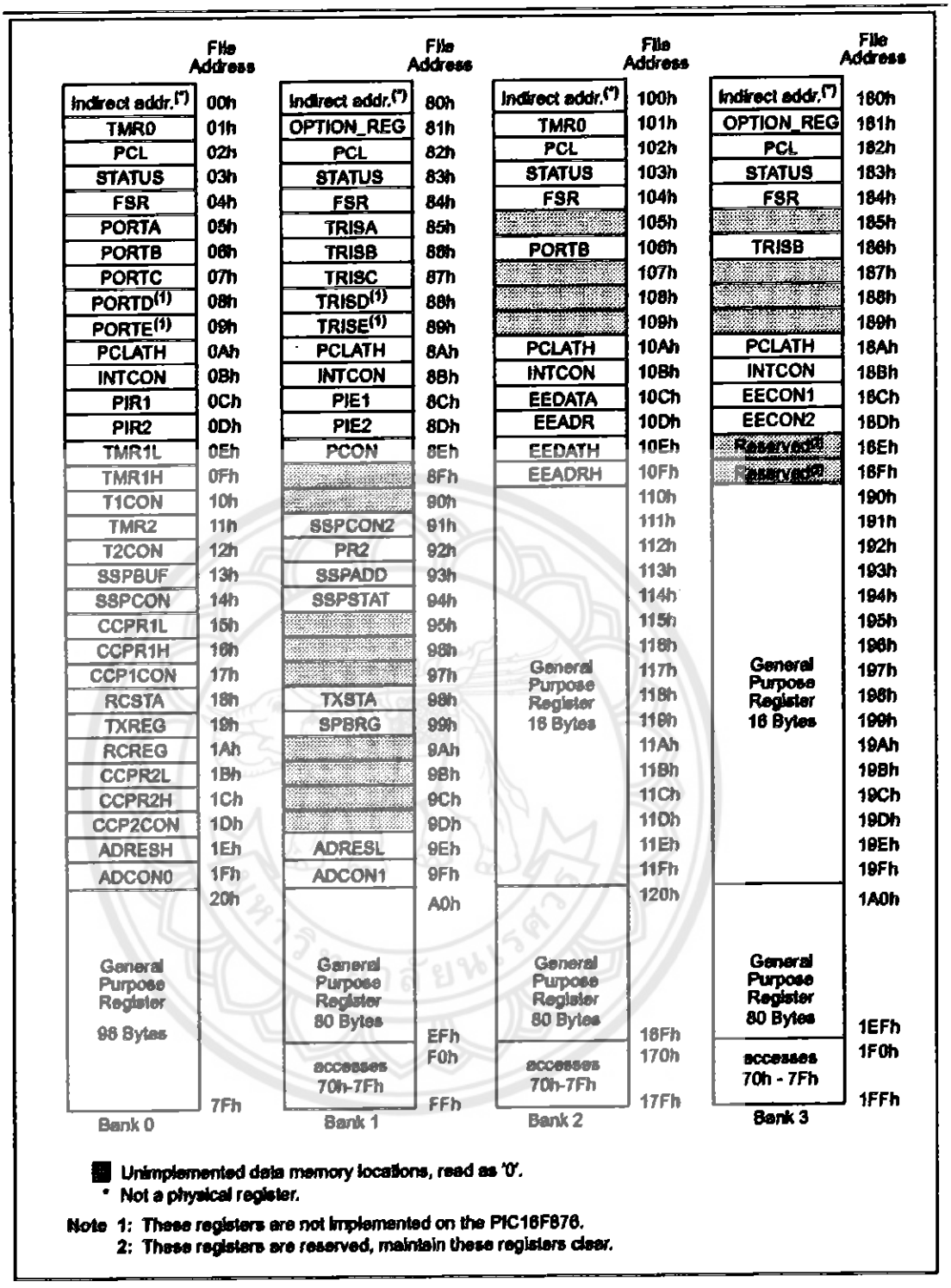
ส่วนนี้มีไว้สำหรับเก็บข้อมูลคำสั่งทั้งหมดโดยโครงสร้างจะเป็นแบบแฟลช (flash memory) ทำให้ลบและเขียนใหม่ได้หลายครั้ง




**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

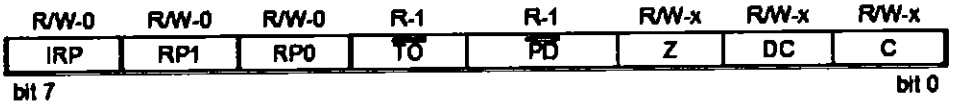
EEADR: รีจิสเตอร์ที่เก็บแอดเดรส 00h – FFh (256 ไบต์)

## 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) เป็นบิตที่แสดงการเกิดไทม์เอาต์ของวอตช์ด็อกไทเมอร์แอกทีฟลอจิก "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)

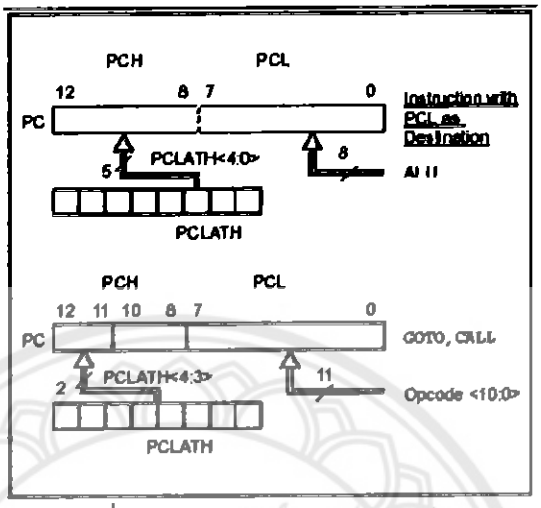
2.4.2 รีจิสเตอร์ W (Working register)

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

**2.4.3 โปรแกรมเคาน์เตอร์ (PC)**

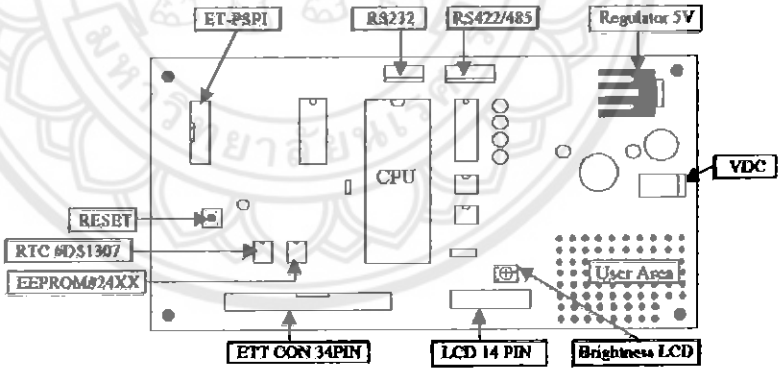
เป็นรีจิสเตอร์ที่สำคัญอีกตัวหนึ่ง ซึ่งทำหน้าที่เก็บตำแหน่งแอดเดรสที่จะให้ CPU ไปทำงานโดยรีจิสเตอร์ PC ของ PIC16F877 จะมีขนาด 13 บิต โดย 8 บิตล่าง PC< 7:0 > จะอยู่ที่รีจิสเตอร์ PCL สามารถอ่านและเขียนได้เหมือนรีจิสเตอร์ทั่วไป ส่วน PC<12 :8 > จะไม่สามารถเข้าถึงได้ตามปกติ การเข้าถึงจะกระทำ ผ่านรีจิสเตอร์ PCLATH < 4 :0 > และเมื่อเกิดการรีเซต PCLATH จะเคลียร์ สถานะเป็น "0" ในการใช้คำสั่งในการกระโดดเช่น CALL หรือ GOTO จะเป็นการนำค่าแอดเดรสมาใส่ในรีจิสเตอร์ PC <10:0 > เพียง 11 บิตเท่านั้นส่วน PC<12:11> จะไม่

เปลี่ยนแปลง 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 ส่วนประกอบและข้อจำกัดต่างๆ ของบอร์ด CP-PIC877 V1.0




**รูปที่ 2.7** แสดงส่วนประกอบของบอร์ด PIC16F877

- ใช้ CPU PIC 16F877 RUN 4 MHZ
- ON CHIP FLASH PROGRAM MEMORY 8K x 14 WORDS
- ON CHIP 368 BYTES RAM / 256 BYTES EEPROM
- 31 BIT I/O PORT ใช้งานของ CPU บนบอร์ด CP-PIC877 V1.0 (34 PIN I/O ET-BUS)
- PORT แบบ SPI, I2C, RS232, RS422/485 (OPTIONS)
- กระแสสำหรับทำการซิงค์/ซอร์สสูงสุด (HIGH SINK / SOURCE CURRENT) 25 MA

• TWO CAPTURE, COMPARE, PWM MODULES

• ระบบ RTC ใช้ชิพ DS1307 (OPTIONS)

• Serial EEPROM 24XX (OPTIONS)

• POWER ON RESET/WATCHDOG TIMER

• A-TO-D ขนาด 10 BIT 8 CH (On Chip)

• LCD PORT 14 PIN ET-BUS สำหรับ LCD แบบตัวอักษร

• 7805 POWER SUPPLY ON BOARD

• PCB SIZE CP-PIC877V1.0 12 x 8.5 cm

**2.5.1 การใช้งาน CP-PIC877 V1.0**

ในบอร์ด CP-PIC877V1.0 ประกอบไปด้วยทรัพยากรต่างๆ ที่ได้ออกแบบไว้ให้สะดวกต่อการนำไปใช้งานต่างๆ ประกอบไปด้วยส่วนต่างๆ ดังนี้

**2.5.2 พอร์ต RS-232**

ในส่วนของ RS-232 มีไว้เพื่อเพิ่มความสามารถในการสื่อสารกันระหว่าง CPU กับคอมพิวเตอร์ หรือกับบอร์ดอื่นๆ ซึ่งเป็นการสื่อสารแบบอนุกรมโดยในบอร์ดนี้ใช้ MAX232 หรือ ICL232 เป็น Line Driver ซึ่งสามารถสื่อสารโดยใช้สายสัญญาณเพียง 3 เส้นได้ไกลถึง 50 ฟุต หรือประมาณ 15 เมตร การใช้งานนั้นทำได้โดยการต่อสายที่ออกจากคอนเนคเตอร์ RS-232 บนบอร์ดไปต่อกับคอมพิวเตอร์ หรือบอร์ดตัวอื่นแล้วเขียนโปรแกรมควบคุมให้ CPU ทำงาน แต่เนื่องจากสัญญาณอนุกรมที่ออกจาก CPU ต้องไปต่อกับ Line Driver ของ RS-422/RS-485 ด้วย ดังนั้นการใช้งาน RS-232 ให้ถอดไอซีที่เป็น Line Driver ของ RS-422/RS-485 ออกทั้ง 2 ตัวด้วย คือเบอร์ 75176

**รูปที่ 2.8** แสดงการต่อใช้งาน RS-232

## 2.6 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

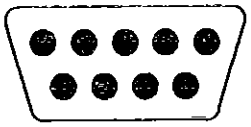
ตามมาตรฐาน RS-232 อุปกรณ์ DTE ควรใช้หัวต่อตัวผู้ และอุปกรณ์ DCE ควรใช้หัวต่อตัวเมียซึ่งหัวต่อที่นิยมใช้กันจะเป็นชนิด D แบบ 9 ขา และ 25 ขา (บางครั้งเรียก DB-25 และ DB-9)

| 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/ $\mu s$ |
| Receiver Input Voltage Range | +/-15V |
| Receiver Input Sensitivity | +/-3V |
| Receiver Input Resistance (Ohms) | 3k to 7k |

ตารางที่ 2.2 ข้อกำหนดของมาตรฐาน RS-232

### 2.6.1 ลักษณะของหัวต่อตามมาตรฐาน RS-232C (DB-25 & DB-9)

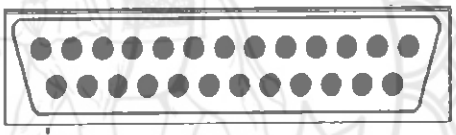
พอร์ตอนุกรมมีขั้วต่อ 2 แบบ คือ แบบ D 25 ขา และแบบ D 9 ขา ซึ่งทั้ง 2 แบบจะเป็นชนิดตัวผู้ทางด้านของคอมพิวเตอร์ ดังนั้นอุปกรณ์ที่จะนำมาต่อกับคอมพิวเตอร์จึงต้องใช้ขั้วต่อชนิดตัวเมีย



รูปที่ 2.9 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 2.10 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 |

| | | | |
|---|---|---|---|
| 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

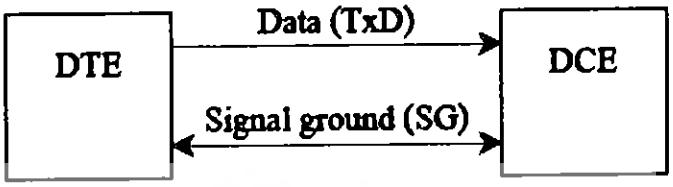
| สัญญาณ | ชื่อ | ตัวส่ง | การทำงาน |
|---|---|---|---|
| TxD | Transmit Data | DTE | ส่งข้อมูลทีละบิตจาก DTE ไปยัง DCE |
| RxD | Receive Data | DCE | รับข้อมูลทีละบิตจาก DCE ไปยัง DTE |
| CTS | Clear to Send | DCE | ตรวจจับสัญญาณจาก DCE ว่าพร้อมจะรับข้อมูลจาก DTE |
| CD | Carrier Detect | DCE | เมื่อไรที่ตรวจจับสัญญาณเจอที่ปลายทางของสัญญาณจะทำให้สายสัญญาณ 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**

**2.6.2 การสื่อสารทางเดียว**

สัญญาณหลักที่ใช้สำหรับการสื่อสารมีอยู่ 2 สัญญาณ ได้แก่

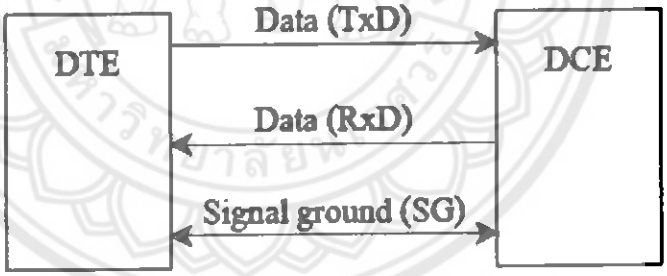
1. สายสัญญาณสำหรับข้อมูลจาก DTE → DCE
2. สายสัญญาณสำหรับกราวด์ (SG) ซึ่งเป็นจุดอ้างอิงร่วมสำหรับชั้วและแรงดันไฟฟ้าของสายอื่น




รูปที่ 2.11 แสดงการเชื่อมต่อเพื่อสื่อสารทางเดียว

**2.6.3 การสื่อสารสองทาง**

ในกรณีที่ข้อมูลถูกส่งผ่านในสองทิศทาง โดยเฉพาะเมื่อคอมพิวเตอร์สองตัวสื่อสารกัน จำนวนสายน้อยที่สุดในการสื่อสารสองทางคือ 3 เส้น ได้แก่ สายข้อมูลแต่ละทิศทาง และสายสัญญาณกราวด์ดังรูปที่ 2.12




รูปที่ 2.12 แสดงการเชื่อมต่อเพื่อการสื่อสารสองทาง

**2.6.4 สัญญาณทางไฟฟ้า**

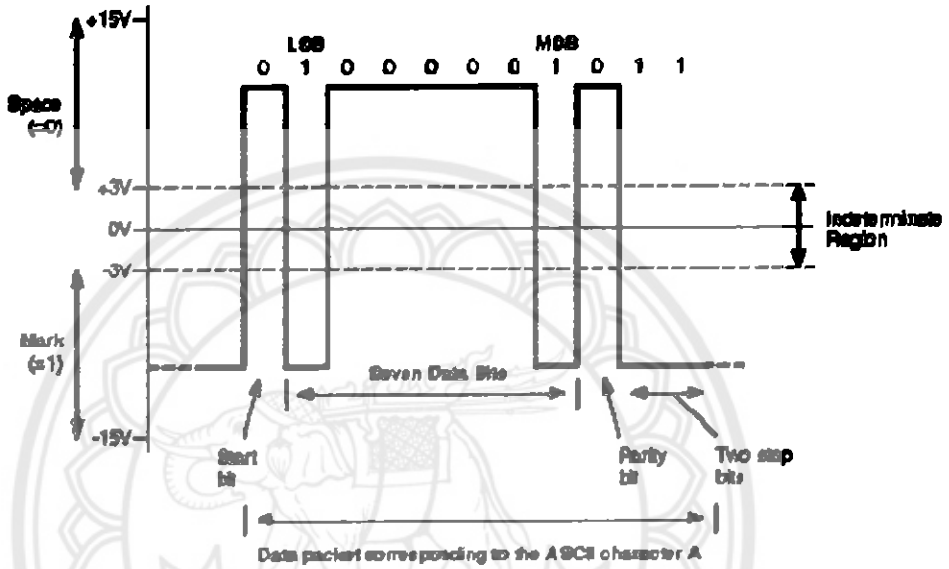
ตามมาตรฐาน RS-232 ได้กำหนดลักษณะของสัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่อแบบอนุกรมมี 2 ลักษณะ คือ สเปสหมายถึง ลอจิก '0' และมาร์ค หมายถึง ลอจิก '1' โดย

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 2.13 RS-232 Wave Form

### 2.6.5 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

## 2.7 หลักการสื่อสารข้อมูล (Data Communication)

การสื่อสารข้อมูล คือ ขบวนการของการใช้ร่วมหรือแลกเปลี่ยนข้อมูลข่าวสารที่ถูกแปลงรหัสระหว่างอุปกรณ์สองตัวหรือมากกว่านั้น การสื่อสารข้อมูลตัวส่งและตัวรับเป็นอุปกรณ์หรือเครื่องมือ และข้อมูลข่าวสารที่ถูกแปลงรหัส หมายถึง ข้อมูลข่าวสารที่ถูกส่งผ่านไปในลักษณะของการส่งผ่านข้อมูลแบบอนุกรมของสัญญาณไฟฟ้า ผ่านตัวกลางที่ใช้ในการส่งข้อมูล เช่น เมื่อเจ้าหน้าที่คอมพิวเตอร์กดปุ่มอักษรบนแป้นพิมพ์ สัญญาณทางไฟฟ้าถูกส่งผ่านพอร์ตอนุกรมออกไปยังคอมพิวเตอร์และคอมพิวเตอร์จะแปลงข้อมูลข่าวสารกลับมา

รูปที่ 2.14 แสดงส่วนประกอบของการสื่อสารข้อมูล

### 2.7.1 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

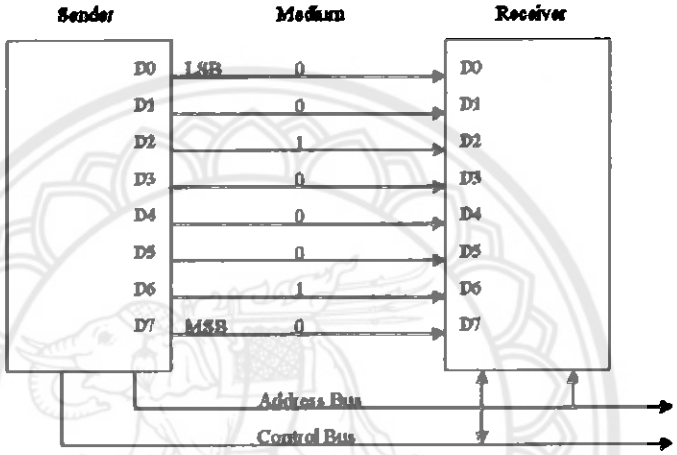
3. Medium เป็นตัวกลางที่ใช้ในการสื่อสารข้อมูลหรืออาจจะเรียกว่า ช่องสัญญาณสื่อสาร เช่น คาต้าลิงค์เน็ตเวิร์ค สายโคแอ็กเชียล ไฟเบอร์ออฟติก และช่องสัญญาณดาวเทียม เป็นต้น

### 2.7.2 เทคนิคการส่งผ่านข้อมูล

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

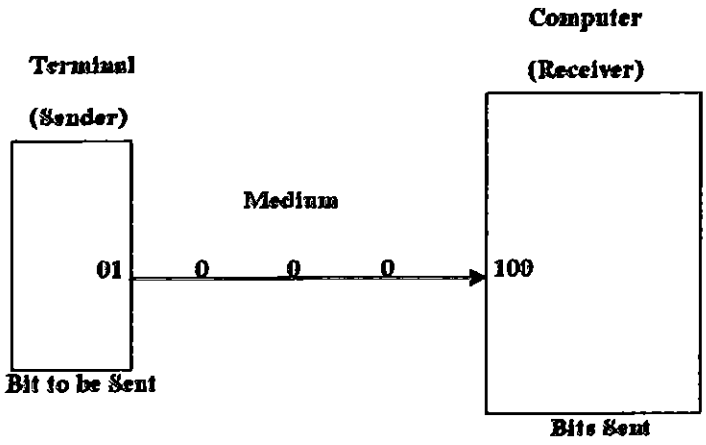
การส่งผ่านข้อมูลแบบขนานย่อมเร็วกว่าการส่งข้อมูลแบบอนุกรม เพราะว่าทุกบิตถูกส่งออกไปพร้อมกัน ดังนั้นในการส่งผ่านข้อมูลแบบขนาน จะถูกนำมาใช้สำหรับการทำงานภายในคอมพิวเตอร์ เช่นในการส่งข้อมูลระหว่าง ซีพียู กับหน่วยความจำ หรือระหว่างซีพียูและอุปกรณ์อินพุตเอาท์พุต เป็นต้น

อย่างไรก็ตามการใช้สายหลายๆ เส้น ทำให้สับสน ราคาสูงและรับสัญญาณรบกวนได้ง่าย ถ้านำการส่งผ่านข้อมูลแบบขนาน มาใช้กับการส่งผ่านข้อมูลระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ หรือคอมพิวเตอร์กับอุปกรณ์รอบนอก เช่น คอมพิวเตอร์กับเครื่องพิมพ์




รูปที่ 2.15 การส่งข้อมูลแบบขนาน (Parallel)

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




**รูปที่ 2.16** การส่งข้อมูลแบบอนุกรม(Serial)

2.7.3 การส่งผ่านข้อมูลแบบอนุกรม (Serial Communication)

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

2.7.4 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

## 2.8 ไอซีฐานเวลาจริง DS1307

การสร้างฐานเวลาให้กับระบบไมโครคอนโทรลเลอร์อาจทำได้โดยการใช้วงจรกำเนิดความถี่ป้อนให้กับตัวนับ/ตัวจับเวลาในไมโครคอนโทรลเลอร์ แต่การใช้งานแบบนี้ผู้ใช้จะต้องเขียนโปรแกรมควบคุมการทำงาน มีการคำนวณคาบเวลาให้ถูกต้อง และต้องตั้งค่าใหม่ทุกครั้งที่มีการเริ่มทำงาน ระบบการสร้างฐานเวลาแบบนี้สามารถใช้งานง่ายๆได้ แต่ไม่สามารถบอกเวลาจริงให้กับระบบได้

ระบบที่ใช้ไมโครคอนโทรลเลอร์ในการควบคุมบางระบบจะต้องมีเวลาเข้ามาเกี่ยวข้องด้วยโดยจะมีการบอกเวลาเป็นชั่วโมง นาที และวินาที หรือมีการบอกวัน เดือน และปี ให้กับระบบ

ด้วย ดังนั้นการสร้างฐานเวลาต้องถูกต้องและแม่นยำ แม้ว่าระบบหยุดทำงานและเริ่มทำงานใหม่ค่าเวลาต่างๆจะต้องถูกต้องด้วย ในปัจจุบันการทำงานประเภทนี้จะใช้ไอซีที่ทำหน้าที่สร้างฐานเวลาจริงให้กับระบบ (RTC หรือ Real Time Clock) ไอซีประเภทนี้จะมีวงจรจัดการด้านเวลาจริงอยู่ภายใน การใช้งานเพียงต่อคริสตอลให้กับไอซี 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 2.17 โดยตำแหน่ง 00H ถึง 07H จะใช้เป็น RTC ส่วนตำแหน่ง 08H ถึง 3FH จะเป็นหน่วยความจำ RAM จากรูปที่ 2.17 จะเห็นว่าที่ตำแหน่ง 00H จะเก็บเวลาเป็น วินาที ตำแหน่ง 01 จะเก็บเวลาเป็นวินาที โดยข้อมูลที่เก็บจะอยู่ในรูปแบบรหัส BCD

รูปที่ 2.17 แผนผังหน่วยความจำ

สำหรับการอ่านและเขียนค่าเวลากับไอซีตัวนี้ทำได้โดยการอ่านและเขียนข้อมูลในลักษณะตัวเลข 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 | |

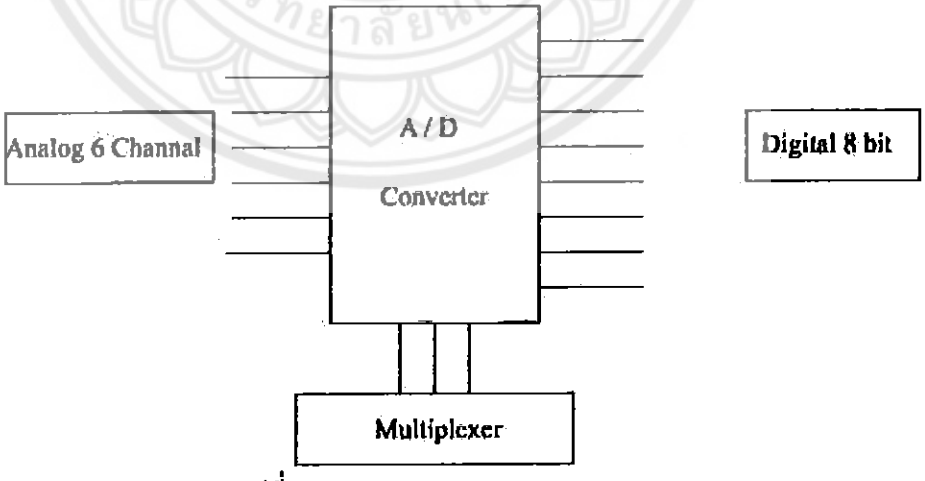
ตารางที่ 2.7 แสดงรีจีสเตอร์ที่ใช้เก็บค่าต่างๆ

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

## 2.9 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

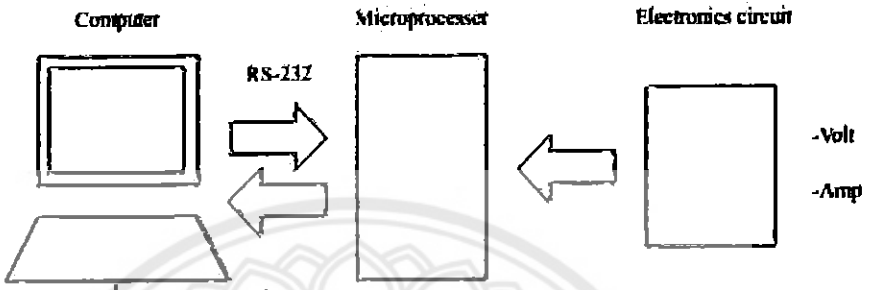



**รูปที่ 2.18** การทำงานของ A/D Converter

# บทที่ 3

# การออกแบบและสร้าง Computer Based Data Logger

## บล็อกไดอะแกรมการทำงาน




รูปที่ 3.1 แสดงบล็อกไดอะแกรมของ Computer Based Data Logger

### 3.1 ส่วนของคอมพิวเตอร์

เป็นอุปกรณ์ควบคุมการทำงานของไมโครคอนโทรลเลอร์ โดยใช้โปรแกรม Visual Basic 6.0 ส่งค่าตัวแปรที่ใช้ในการควบคุมการทำงานให้แก่ไมโครคอนโทรลเลอร์ และรับข้อมูลจากไมโครคอนโทรลเลอร์มาวิเคราะห์แสดงผล ในรูปแบบของข้อมูลตัวเลข และทำการเก็บข้อมูลที่รับเข้ามาไว้ในโปรแกรม notepad

#### 3.1.1 อัตราความเร็วในการรับ-ส่งข้อมูลอนุกรม

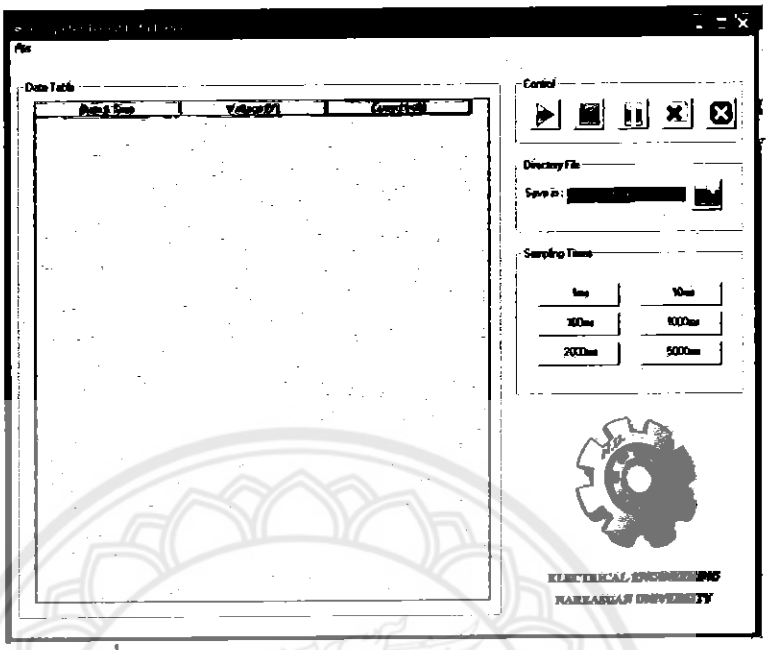
การที่คอมพิวเตอร์กับไมโครคอนโทรลเลอร์จะติดต่อสื่อสารกันได้นั้น จะต้องทำงานด้วยอัตราเร็วเท่ากันซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสที่ใช้ในโครงงานนี้ คือ ค่าบอดเรต 9600 บิตต่อวินาที ซึ่งเป็นค่าอัตราเร็วในการสื่อสารแบบอนุกรม RS-232

ห้องสมุดคณะวิศวกรรมศาสตร์



I 5081415 e.2

### 3.1.2 ส่วนของโปรแกรมควบคุมการทำงาน(Visual Basic 6.0)



มร.
ท152ค
2549.

รูปที่ 3.2 แสดงการทำงานของโปรแกรม Visual Basic 6.0

ส่วนของ Visual Basic 6.0 ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์

เฟรมที่ 1 มีชื่อว่า Data Table ทำหน้าที่แสดงผลข้อมูล

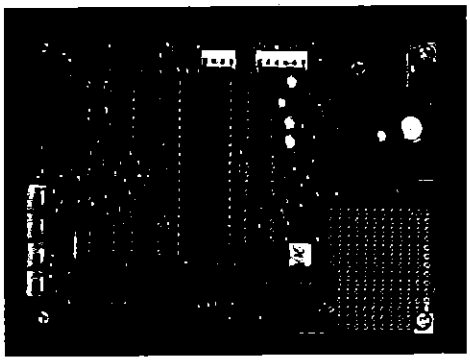
เฟรมที่ 2 มีชื่อว่า Control ทำหน้าที่ควบคุมการทำงานหลักของโปรแกรม

เฟรมที่ 3 มีชื่อว่า Directory File ใช้เพื่อสร้างโฟลเดอร์ที่ใช้ในการจัดเก็บข้อมูล

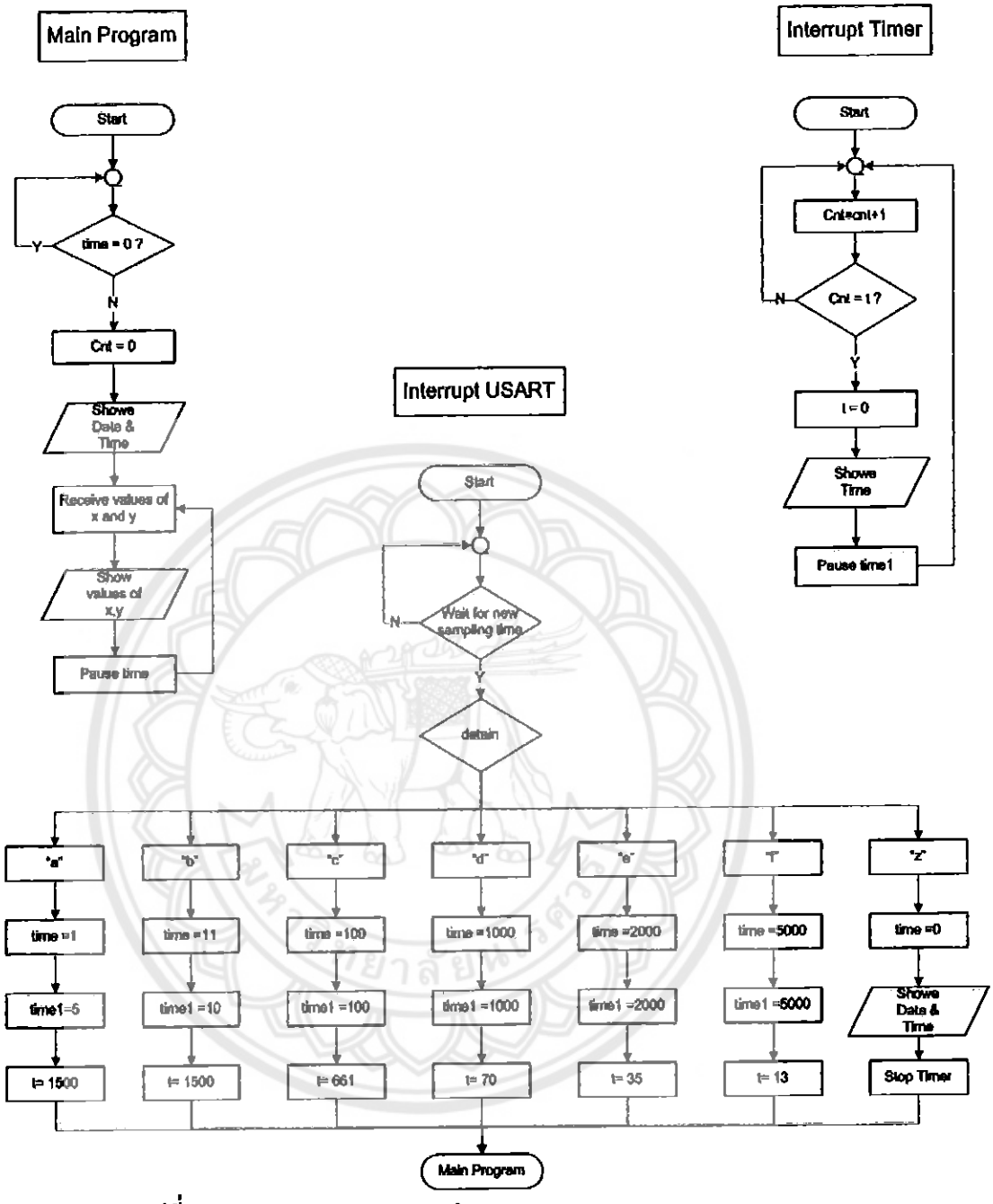
เฟรมที่ 4 มีชื่อว่า Sampling Times ใช้เพื่อกำหนดความเร็วในการแสดงผล และการบันทึกข้อมูล

### 3.2 ส่วนของไมโครคอนโทรลเลอร์(CP-PIC 16F877)

เป็นฮาร์ดแวร์ที่ใช้โปรแกรมควบคุมการ รับ-ส่ง ข้อมูล ผ่าน พอร์ต RS-232



รูปที่ 3.3 แสดงบอร์ดไมโครคอนโทรลเลอร์ CP-PIC 16F877 V.1

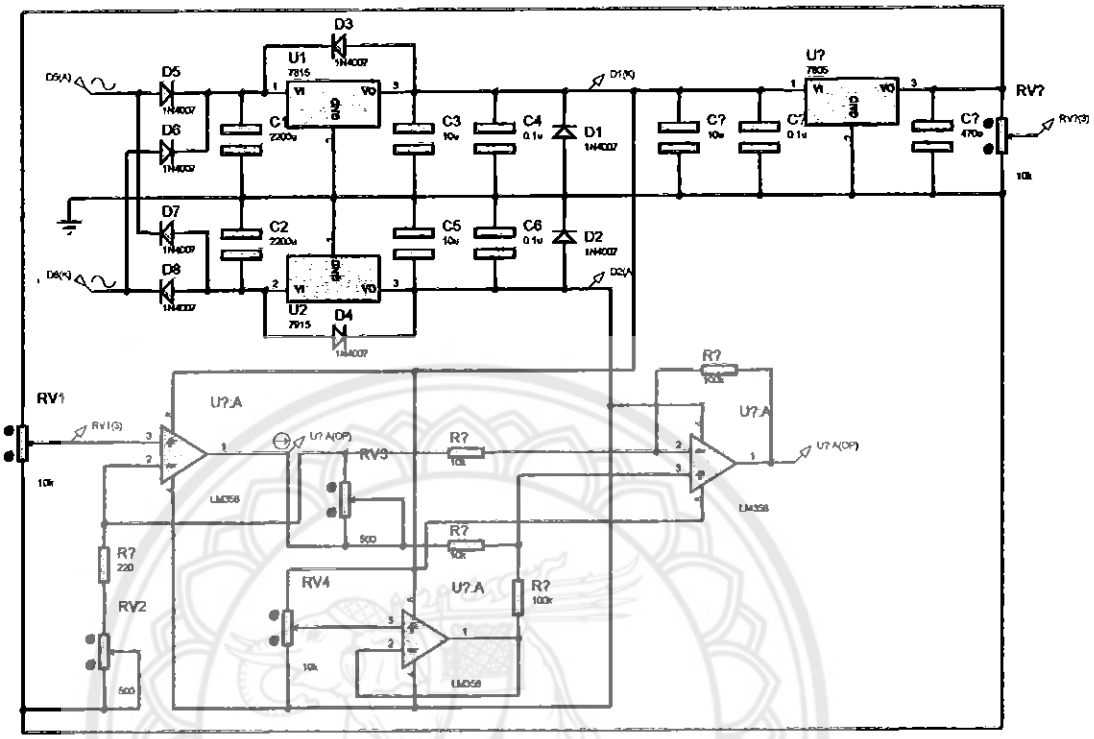



**รูปที่ 3.4** Flow-Chart แสดงการทำงานของ Computer Based Data Logger

## 3.3 การต่อวงจรทั้งหมดของแหล่งจ่ายแรงดันไฟฟ้า (Hardware)

การต่อวงจรทั้งหมดของแหล่งนั้นจะต้องคำนึงถึงส่วนประกอบต่างๆ ว่าอยู่ในรูปที่สามารถเข้ากันได้ เช่น ข้อมูลก่อนที่จะเข้า A/D converter นั้นจะต้องเป็นข้อมูลที่อยู่ในรูปของสัญญาณแรงดันและมีช่วงของสัญญาณไม่เกินขนาดของแรงดันอ้างอิง คือ 5 Vdc ที่ป้อนให้กับ A/D converter หรือสัญญาณที่ออกจากไมโครคอนโทรลเลอร์ที่จะต่อเข้ากับคอมพิวเตอร์จะต้องเป็นข้อมูลอนุกรม มีอัตราความเร็วในการส่งข้อมูล (Baud Rate) ตรงกับอัตราเร็วที่โปรแกรมในการ

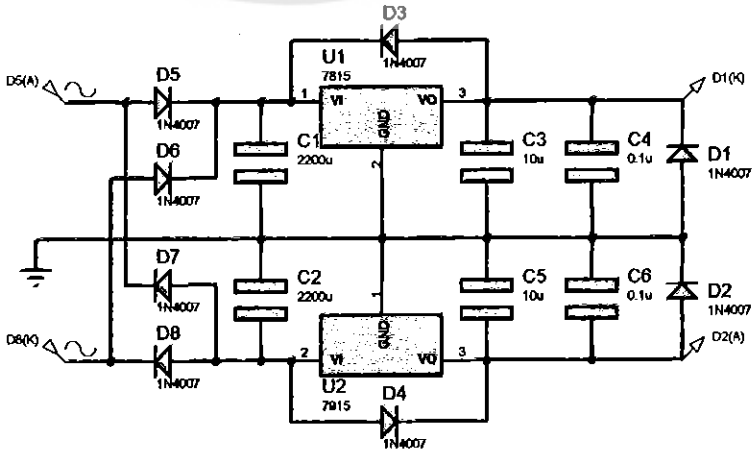
คำนวณต้องคำนึงถึงมาตรฐานแรงดันที่ Port RS-232 กำหนดไว้เพื่อที่จะได้สามารถส่งข้อมูลได้อย่างไม่ผิดพลาด




รูปที่ 3.5 แสดงวงจรรวมที่ใช้ในการจ่ายแรงดันเข้ากับบอร์ด

### 3.3.1 การต่อไอซี 7815 และ 7915 กับวงจร

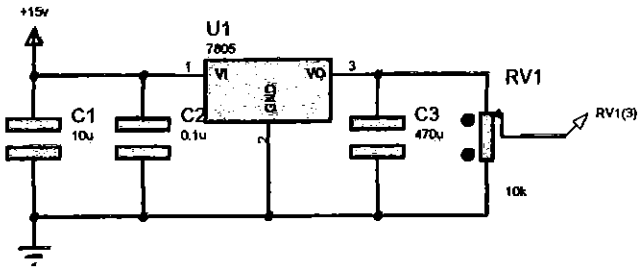
ในการสร้างวงจรที่ใช้ในการแปลงกระแสเป็นแรงดัน - แรงดันเป็นกระแสนั้นจำเป็นต้องมีไฟเลี้ยงไอซี LM358 ที่มีค่า +/-15 Vdc ดังนั้นจึงต้องอาศัยวงจรที่ใช้จ่ายไฟให้กับไอซีดังรูปที่ 3.6




รูปที่ 3.6 แสดงวงจรจ่ายไฟเลี้ยง +/- 15 Vdc ให้กับไอซี LM358

### 3.3.2 การต่อไอซี 7805 กับวงจร

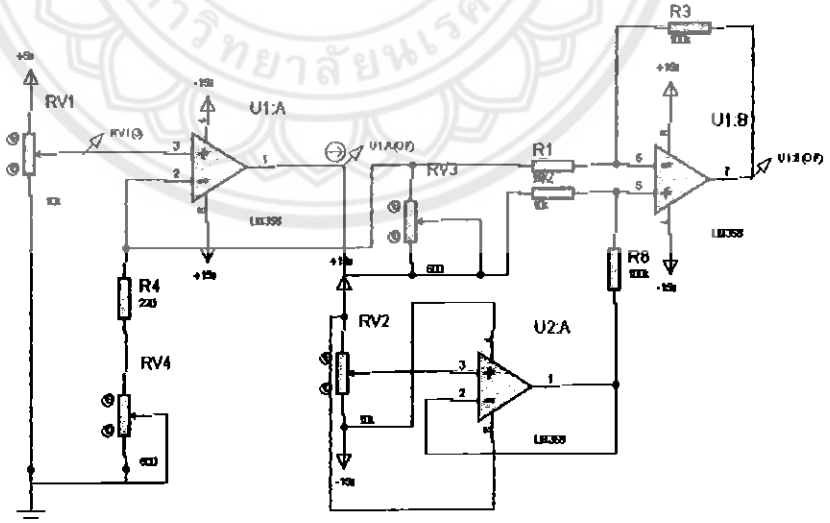
7805 เป็นไอซีเรกกูเรเตอร์ที่ใช้ควบคุมแรงดันที่ต้องการจ่ายให้กับไมโครคอนโทรลเลอร์ ซึ่งแรงดันที่ถูกจ่ายออกมาจาก 7805 จะมีค่าไม่เกิน 5 Vdc




รูปที่ 3.7 แสดงวงจรจ่ายแรงดัน 0-5 Vdc โดยใช้ไอซีเรกกูเลเตอร์เบอร์ 7805

### 3.3.3 การต่อไอซี LM358 กับวงจร

สัญญาณมาตรฐานที่ใช้ในการควบคุมกระบวนการต่างๆ ในอุตสาหกรรมอาจแบ่งเป็น 3 ประเภท คือสัญญาณลม (Pneumatics) อิเล็กทรอนิกส์ (Electronics) และสัญญาณดิจิตอล (Digital) โดยขนาดของสัญญาณมาตรฐานที่ใช้ในระบบอิเล็กทรอนิกส์ในปัจจุบัน มี 2 แบบคือ แรงดันไฟฟ้ามาตรฐาน 1-5 Vdc และที่พบน้อยมากคือ 0-10 Vdc และสัญญาณกระแสไฟฟ้ามาตรฐาน 4-20 mAdc และที่พบน้อยมากคือ 10-50 mAdc ดังนั้นการนำอุปกรณ์ดังกล่าวไปทำงานร่วมกันจำเป็นต้องมีการแปลงสัญญาณเสียก่อน




**รูปที่ 3.8 แสดงวงจรเปลี่ยนแรงดันเป็นกระแส – เปลี่ยนกระแสเป็นแรงดัน**

# บทที่ 4

# ผลการทดลองและวิธีการทดลอง

## 4.1 การทดสอบระดับแรงดันจากวงจรเปรียบเทียบกับผลที่แสดงใน Visual Basic

จากวงจรที่ได้ออกแบบมาแล้ว เมื่อนำไปทดสอบวัดแรงดัน เทียบกับการแสดงผลที่ได้จากโปรแกรม Visual Basic จะได้ผลดังตารางที่ 4.1

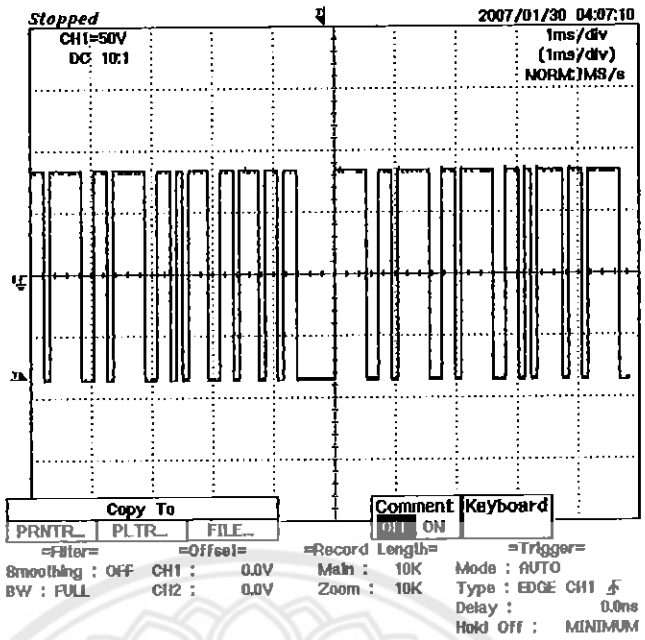
| แรงดัน (V) | | กระแส (mA) | |
|---|---|---|---|
| Multimeter | VB | Mutimeter | VB |
| 1 | 0.98 | 4 | 3.92 |
| 2 | 1.98 | 8 | 7.92 |
| 3 | 2.90 | 12 | 11.84 |
| 4 | 3.90 | 16 | 15.84 |
| 5 | 4.90 | 20 | 19.84 |

**ตารางที่ 4.1** ตารางเปรียบเทียบค่าที่วัดได้จาก Multimeter เทียบกับการแสดงผลใน Visual Basic

## 4.2 การทดสอบนับจำนวนข้อมูลเทียบกับเวลาและคลื่นสัญญาณแรงดัน

**- ผลการทดลองค่าหน่วงเวลา 1ms**

| นาที | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล/นาที | 6859 | 6857 | 6856 | 6861 | 6714 | 6817 | 6858 | 6860 | 6758 | 6861 |

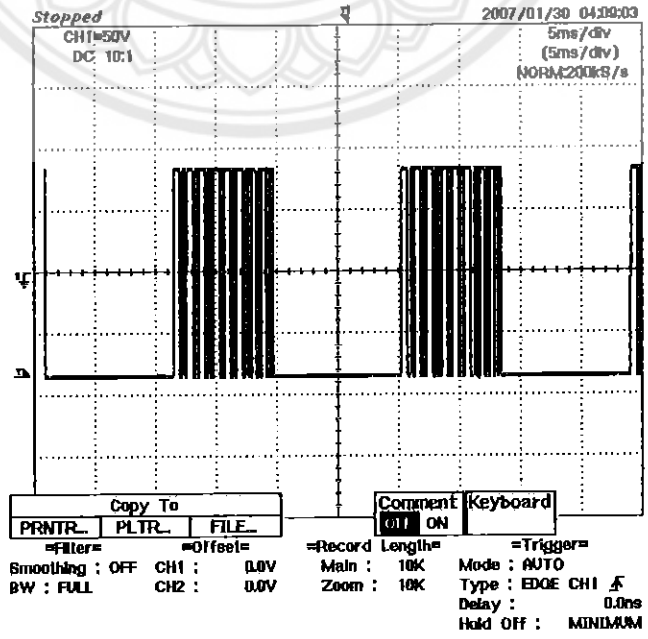



**รูปที่ 4.1** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1ms

จากรูปค่า time/div มีค่าเท่ากับ 1ms/div จำนวนช่องสัญญาณจากรูปไม่เท่ากับ 1 ช่องสัญญาณ จึงมีค่าเท่าไม่กับ 1ms

**- ผลการทดลองค่าหน่วงเวลา 10ms**

| นาที | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล/นาที | 3197 | 3197 | 3194 | 3197 | 3192 | 3197 | 3196 | 3197 | 3197 | 3196 |

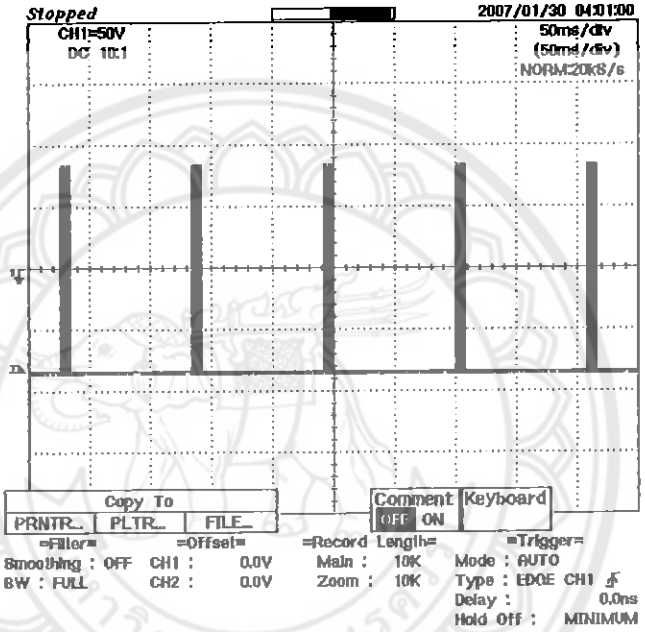



**รูปที่ 4.2** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 10ms

จากรูปค่า time/div มีค่าเท่ากับ 5ms/div จำนวนช่องสัญญาณจากรูปมากกว่า 2 ช่องสัญญาณ จึงมีค่าเท่าไม่กับ 10ms

**- ผลการทดลองค่าหน่วงเวลา 100ms**

| นาที | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล/นาที | 557 | 558 | 550 | 558 | 556 | 558 | 556 | 558 | 558 | 556 |

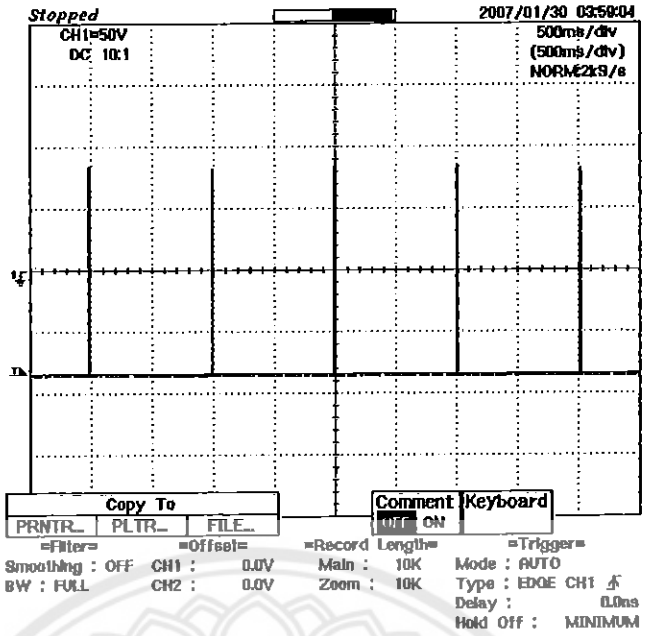



รูปที่ 4.3 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 100ms

จากรูปค่า time/div มีค่าเท่ากับ 50ms/div จำนวนช่องสัญญาณจากรูปมีค่า 2 ช่องสัญญาณ จึงมีค่าเท่ากับ 100ms

**- ผลการทดลองค่าหน่วงเวลา 1000ms**

| นาที | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล/นาที | 59 | 60 | 59 | 60 | 60 | 59 | 60 | 60 | 60 | 59 |

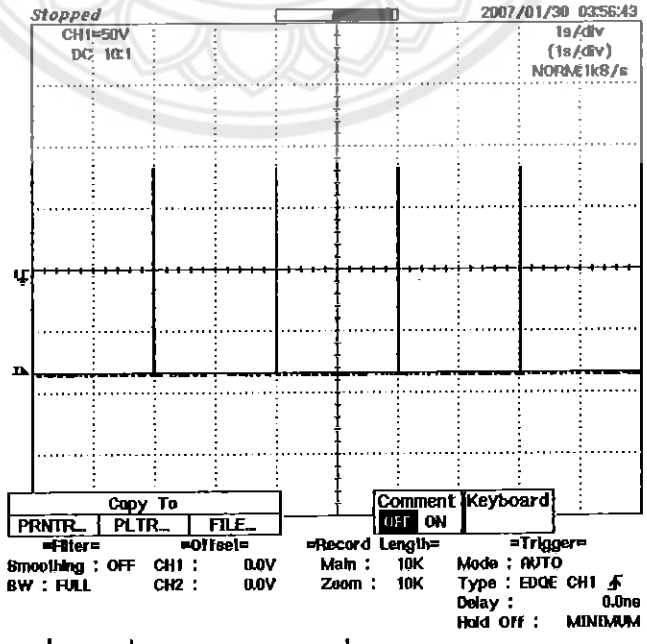



**รูปที่ 4.4** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1000ms

จากรูปค่า time/div มีค่าเท่ากับ 500ms/div จำนวนช่องสัญญาณจากรูปมีค่า 2 ช่องสัญญาณ จึงมีค่าเท่ากับ 1000ms

**- ผลการทดลองค่าหน่วงเวลา 2000ms**

| นาที | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล/นาที | 30 | 30 | 30 | 30 | 30 | 31 | 30 | 31 | 30 | 30 |

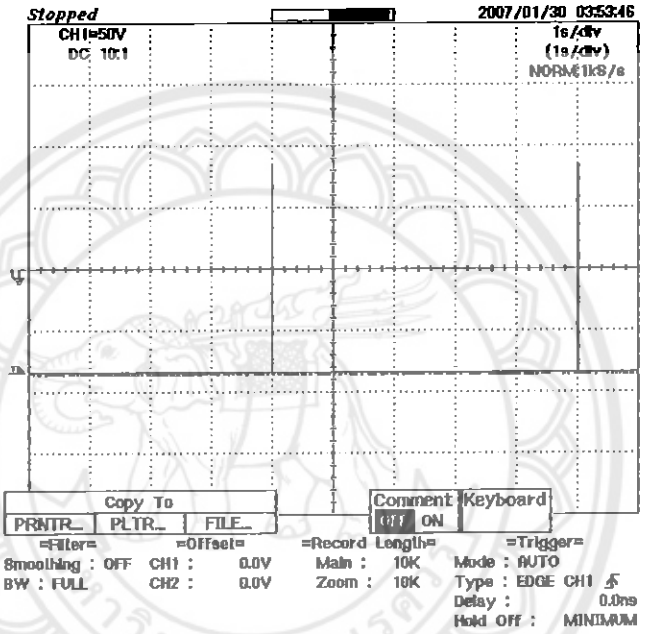



**รูปที่ 4.5** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 2000ms

จากรูปค่า time/div มีค่าเท่ากับ 1s/div จำนวนช่องสัญญาณจากรูปมีค่า 2 ช่องสัญญาณ จึงมีค่าเท่ากับ 2000ms

**- ผลการทดลองค่าหน่วงเวลา 5000ms**

| นาที | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล/นาที | 12 | 12 | 12 | 12 | 12 | 11 | 12 | 12 | 12 | 12 |




รูปที่ 4.6 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 5000ms

จากรูปค่า time/div มีค่าเท่ากับ 1ms/div จำนวนช่องสัญญาณจากรูปมีค่า 5 ช่องสัญญาณ จึงมีค่าเท่ากับ 5000ms

จากรูปที่ 4.2-4.6 เป็นการวัดคลื่นสัญญาณของแรงดันที่ได้จากการส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 จะเห็นได้ว่า ช่วงเวลาของการ on-off จะเป็นไปตามค่า Sampling Time ที่กำหนด

# บทที่ 5
# บทสรุป

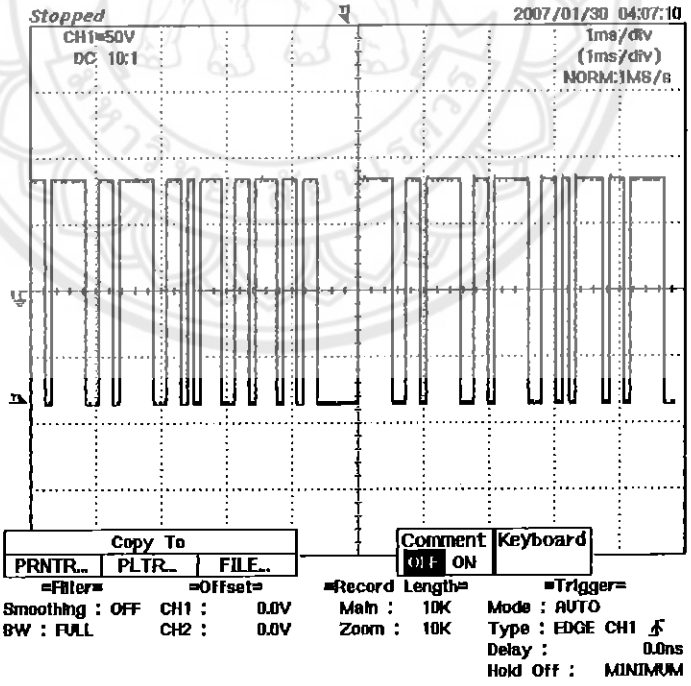
ในบทนี้จะเป็นการสรุปผลที่ได้จากการทดลองในโครงงานนี้ พร้อมเสนอแนะแนวทางในการนำโครงงานนี้ไปพัฒนาให้มีประสิทธิภาพมากยิ่งขึ้นต่อไปในอนาคต

## 5.1 สรุปผล

ในโครงงานนี้ ได้ออกแบบโปรแกรม Computer Based Data Logger เพื่อใช้ในการบันทึกข้อมูลแรงดันและกระแสไฟฟ้าไฟตรง โดยการรับสัญญาณผ่านไมโครคอนโทรลเลอร์ซึ่งมีวงจรแปลงอะนาลอกเป็นดิจิตอลภายใน จากผลการทดลองสามารถแสดงและบันทึกค่าของข้อมูลที่รับเข้ามาในรูปแบบของไฟล์ Text ภายในระยะเวลาที่กำหนด

## 5.2 สรุปการเปรียบเทียบค่าหน่วงเวลาและกราฟที่ได้จากการวัดด้วย Oscilloscope

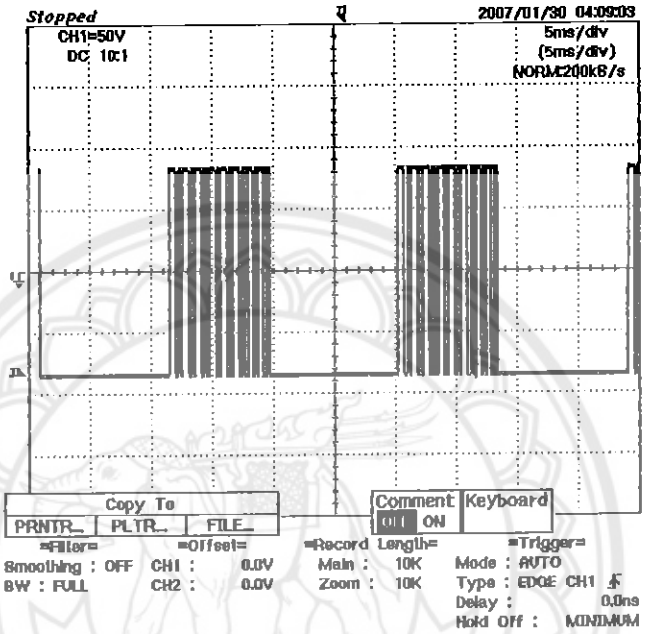
**- ผลการทดลองค่าหน่วงเวลา 1ms**




**รูปที่ 5.1** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1ms

จากรูปค่า time/div มีค่าเท่ากับ 1ms/div จำนวนช่องสัญญาณจากรูปไม่เท่ากับ 1 ช่องสัญญาณ จึงมีค่าเท่าไม่กับ 1ms ดังนั้นสรุปได้ว่าค่าที่ได้จากการวัดมีความคลาดเคลื่อนจากค่าหน่วงเวลาที่กำหนด เนื่องจากจำนวนของข้อมูลมีจำนวนมากกว่าค่าหน่วงเวลา

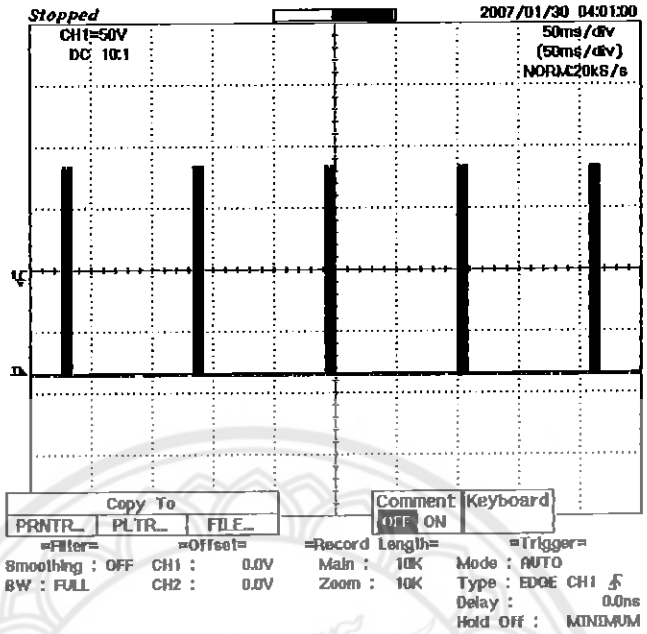
**- ผลการทดลองค่าหน่วงเวลา 10ms**




รูปที่ 5.2 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 10ms

จากรูปค่า time/div มีค่าเท่ากับ 5ms/div จำนวนช่องสัญญาณจากรูปมากกว่า 2 ช่องสัญญาณ จึงมีค่าเท่าไม่กับ 10ms ดังนั้นสรุปได้ว่าค่าที่ได้จากการวัดมีความคลาดเคลื่อนจากค่าหน่วงเวลาที่กำหนด เนื่องจากจำนวนของข้อมูลมีจำนวนมากกว่าค่าหน่วงเวลา

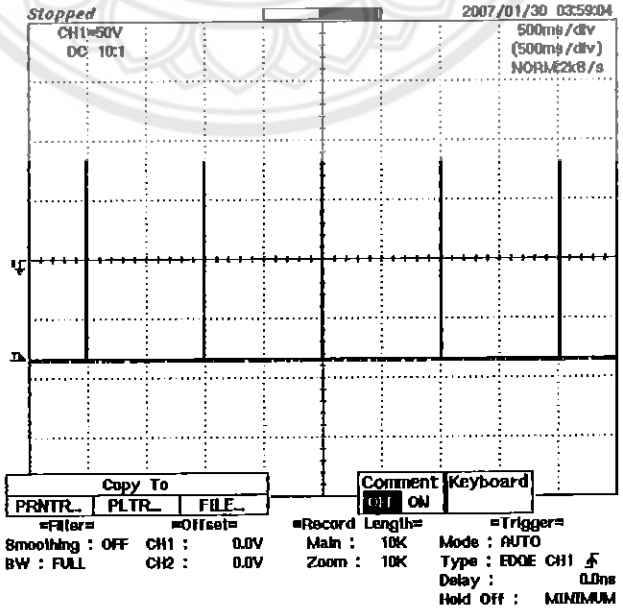
**- ผลการทดลองค่าหน่วงเวลา 100ms**




**รูปที่ 5.3** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 100ms

จากรูปค่า time/div มีค่าเท่ากับ 50ms/div จำนวนช่องสัญญาณจากรูป**มีค่า 2 ช่องสัญญาณ จึงมีค่า**เท่ากับ 100ms ดังนั้นสรุปได้ว่าค่าที่ได้จากการวัดตรงกับค่าหน่วงเวลาที่**กำหนด**

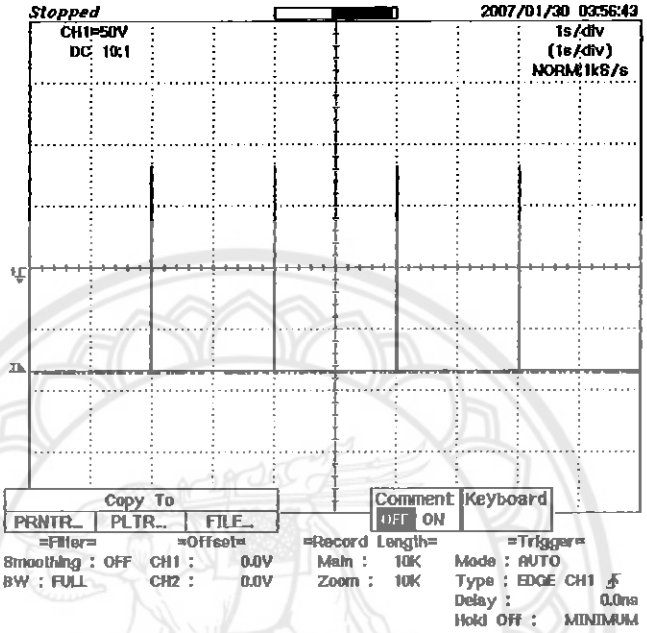
**- ผลการทดลองค่าหน่วงเวลา 1000ms**




**รูปที่ 5.4** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 1000ms

จากรูปค่า time/div มีค่าเท่ากับ 500ms/div จำนวนช่องสัญญาณจากรูปมีค่า 2 ช่องสัญญาณ จึงมีค่าเท่ากับ 1000ms ดังนั้นสรุปได้ว่าค่าที่ได้จากการวัดตรงกับค่าหน่วงเวลาที่กำหนด

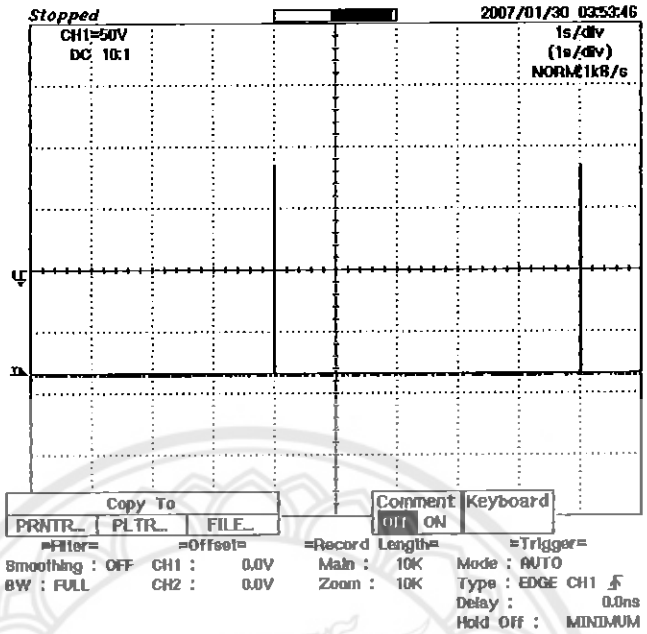
**- ผลการทดลองค่าหน่วงเวลา 2000ms**




รูปที่ 5.5 คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 2000ms

จากรูปค่า time/div มีค่าเท่ากับ 1s/div จำนวนช่องสัญญาณจากรูปมีค่า 2 ช่องสัญญาณ จึงมีค่าเท่ากับ 2000ms ดังนั้นสรุปได้ว่าค่าที่ได้จากการวัดตรงกับค่าหน่วงเวลาที่กำหนด

**- ผลการทดลองค่าหน่วงเวลา 5000ms**




**รูปที่ 5.6** คลื่นสัญญาณแรงดันที่ค่า Sampling Time = 5000ms

จากรูปค่า time/div มีค่าเท่ากับ 1ms/div จำนวนช่องสัญญาณจากรูปมีค่า 5 ช่องสัญญาณ จึงมีค่าเท่ากับ 5000ms ดังนั้นสรุปได้ว่าค่าที่ได้จากการวัดตรงกับค่าหน่วงเวลาที่กำหนด

จะเห็นว่าระยะห่างของรูปสัญญาณในแต่ละกราฟจะมีค่าตรงกับค่าหน่วงเวลา มีเพียงค่าหน่วงเวลา 1 ms และ 10 ms เท่านั้นที่มีความคลาดเคลื่อนเนื่องจากมีการประมวลผลอย่างรวดเร็ว

## 5.3 สรุปจำนวนข้อมูลที่รับเข้ามาต่อนาทีในไฟล์ Text

**- ผลการทดลองค่าหน่วงเวลา 1ms**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล | 6859 | 6857 | 6856 | 6861 | 6714 | 6817 | 6858 | 6860 | 6758 | 6861 |

**จากข้อมูลที่ได้ ค่าหน่วงเวลา 1ms จะมีจำนวนข้อมูลที่บันทึกเฉลี่ยประมาณ 6830 ชุดข้อมูลภายใน 1 นาที**

**- ผลการทดลองค่าหน่วงเวลา 10ms**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล | 3197 | 3197 | 3194 | 3197 | 3192 | 3197 | 3196 | 3197 | 3197 | 3196 |

จากข้อมูลที่ได้ ค่าหน่วงเวลา 10ms จะมีจำนวนข้อมูลที่บันทึกเฉลี่ยประมาณ 3196 ชุด ข้อมูลภายใน 1 นาที

**- ผลการทดลองค่าหน่วงเวลา 100ms**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล | 557 | 558 | 550 | 558 | 556 | 558 | 556 | 558 | 558 | 556 |

จากข้อมูลที่ได้ ค่าหน่วงเวลา 100ms จะมีจำนวนข้อมูลที่บันทึกเฉลี่ยประมาณ 557 ชุด ข้อมูลภายใน 1 นาที

**- ผลการทดลองค่าหน่วงเวลา 1000ms**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล | 59 | 60 | 59 | 60 | 60 | 59 | 60 | 60 | 60 | 59 |

จากข้อมูลที่ได้ ค่าหน่วงเวลา 1000ms จะมีจำนวนข้อมูลที่บันทึกเฉลี่ยประมาณ 60 ชุด ข้อมูลภายใน 1 นาที

**- ผลการทดลองค่าหน่วงเวลา 2000ms**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล | 30 | 30 | 30 | 30 | 30 | 31 | 30 | 31 | 30 | 30 |

จากข้อมูลที่ได้ ค่าหน่วงเวลา 2000ms จะมีจำนวนข้อมูลที่บันทึกเฉลี่ยประมาณ 30 ชุด ข้อมูลภายใน 1 นาที

**- ผลการทดลองค่าหน่วงเวลา 5000ms**

| ครั้งที่ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| จำนวนข้อมูล | 12 | 12 | 12 | 12 | 12 | 11 | 12 | 12 | 12 | 12 |

จากข้อมูลที่ได้ ค่าหน่วงเวลา 5000ms จะมีจำนวนข้อมูลที่บันทึกเฉลี่ยประมาณ 12 ชุด ข้อมูลภายใน 1 นาที

## 5.4 ประเมินผล

จากการดำเนินงานโครงงาน เมื่อเทียบกับวัตถุประสงค์ ได้ผลดังนี้

5.4.1 สามารถออกแบบอุปกรณ์เก็บข้อมูลโดยใช้ไมโครคอนโทรลเลอร์เป็นตัวรับ-ส่งข้อมูล ให้กับคอมพิวเตอร์ และสามารถแสดงผลการทำงานต่างๆ ได้

5.4.2 สามารถรับส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ได้

5.4.3 สามารถใช้ Visual Basic 6.0 ควบคุมการทำงานของไมโครคอนโทรลเลอร์ได้

5.4.4 สามารถเก็บข้อมูลในระยะเวลาที่กำหนด และแสดงผลทางจอคอมพิวเตอร์ได้

## 5.5 ปัญหา ข้อเสนอแนะ และแนวทางแก้ไข

5.5.1 ปัญหาเกิดจากการต่อวงจรผิดพลาด ทำให้อุปกรณ์ได้รับเสียหาย จึงต้องทำการจัดซื้ออุปกรณ์มาใหม่ ทำให้งานที่ออกมาช้ากว่ากำหนด

5.5.2 ปัญหาเกิดจากการจ่ายแรงดันไฟฟ้าให้กับอุปกรณ์อิเล็กทรอนิกส์ ถ้าจ่ายแรงดันไม่เหมาะสมสัญญาณต่าง ๆจะไม่ได้ตามที่ต้องการ จะต้องจ่ายแรงดันที่เหมาะสมให้กับอุปกรณ์อิเล็กทรอนิกส์ถึงจะทำงานได้

5.5.3 ปัญหาเกิดจากการเลือกอุปกรณ์อิเล็กทรอนิกส์ไม่เหมาะสมกับขนาดวงจร และค่าของอุปกรณ์

## 5.6 แนวทางในการพัฒนาต่อไป

ผลที่ได้จากการทดลองในโครงงานนี้ สามารถนำไปประยุกต์ใช้กับงานที่ต้องการความแม่นยำในการบันทึกค่าของข้อมูลภายในระยะเวลาที่กำหนด และเหมาะสมที่จะนำไปพัฒนาต่อให้สามารถใช้กับการวัดในรูปแบบต่างๆ เช่น การบันทึกค่าอุณหภูมิ ความแม่นยำของเซนเซอร์

# เอกสารอ้างอิง

[1] กิตติ ภักดีวัฒนะกุล, จำลอง ครูอุตสาหะ. **Visual Basic Version 6.** พิมพ์ครั้งที่ 11. กรุงเทพมหานคร: ไทยเจริญการพิมพ์, พ.ศ. 2546

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

[3] ฉัททวุฒิ พืชผล, พิชิต สันติกุลานนท์, พร้อมเลิศ หล่อวิจิตร. **คู่มือเรียน Visual Basic 6.** พิมพ์ครั้งที่ 11. กรุงเทพมหานคร: โปรวิชั่น, พ.ศ. 2547

[4] ณัฐพล วงศ์สุนทรชัย, ชัยวัฒน์ ลิ้มพรจิตรลัย. เรียนรู้และปฏิบัติการ**ไมโครคอนโทรลเลอร์ PIC16F877.** กรุงเทพมหานคร: อินโนเวตีฟ เอ็กเพอริเมนต์, พ.ศ. 2521

[5] ธารินทร์ สิทธิธรรมชารี. **Microsoft Visual Basic 6 DataBase & Web Programming.** กรุงเทพมหานคร: ซัคเซส มิเดีย, พ.ศ. 2521

[6] ธารินทร์ สิทธิธรรมชารี. **Visual Basic Version 6.0.** พิมพ์ครั้งที่ 12. กรุงเทพมหานคร: ซัคเซส มิเดีย, พ.ศ. 2548

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

[8] พิษณุ วิทยศิลป์, มังกร หริรักษ์. **สู่โลกอิเล็กทรอนิกส์.** กรุงเทพมหานคร: เอช.เอ็น.กรุ๊ป, พ.ศ. 2539

[9] รศ.ธีรวัฒน์ ประกอบผล. **ภาษาแอสเซมบลี สำหรับ MCS-51.** พิมพ์ครั้งที่ 2. กรุงเทพมหานคร: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), พ.ศ. 2547

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

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

[12] อภิชาติ ภู่พลับ. **เขียนโปรแกรม Hardware Interface ด้วย VB6.** นนทบุรี: ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์, พ.ศ. 2548

# ภาคผนวก

# ภาคผนวก ก
# โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

```
'*************************************************************************
'* Name    : UNTITLED.BAS                                                *
'* Author  : [select VIEW...EDITOR OPTIONS]                              *
'* Notice  : Copyright (c) 2007 [select VIEW...EDITOR OPTIONS]           *
'*         : All Rights Reserved                                         *
'* Date    : 25/4/2007                                                   *
'* Version : 1.0                                                         *
'* Notes   :                                                             *
'*************************************************************************
'******************************** Defines USART parameters ***************************
DEFINE HSER_RCSTA 90h
DEFINE HSER_TXSTA 24h
DEFINE HSER_BAUD 9600
'******************************** Define ADCIN parameters ****************************
DEFINE ADC_BITS 8
DEFINE ADC_CLOCK 3
DEFINE ADC_SAMPLEUS 50

SO      VAR PORTC.6
SI      VAR PORTC.7
DPIN    VAR PORTC.4
CPIN    VAR PORTC.3
' ******************************** Declare variables ********************************
cnt     VAR WORD
B1      VAR BYTE
B2      VAR BYTE
B3      VAR BYTE
B4      VAR BYTE
B5      VAR BYTE
B6      VAR BYTE
B7      VAR BYTE
X       VAR WORD
```

```
Y       VAR WORD

datain  VAR BYTE
discard VAR BYTE
time    VAR WORD
time1   VAR WORD
t       VAR WORD
ADCON1 = %00000000
TRISA = %11111111
TRISB = 0
TRISC = %10000000
PORTB = 0
time = 0
'****************************** interrupt settings ***********************************
TXSTA = $24
RCSTA = $90
SPBRG = 25
PIE1.5 = 1
INTCON = %11100000
RCIF VAR PIR1.5

start:
  RCSTA = $90
ON INTERRUPT GOTO myint
  WHILE time = 0
  WEND
  CNT = 00
TIMER2_SETUP :
  T2CON = %01001011
  PR2 = 249
  PIR1.1 = 0
  PIE1.1 = 1
```

```
  INTCON = %11000000
  T2CON.2 = 1
'***************************** main program starts here ******************************
I2CREAD DPIN,CPIN,$D0,$00,[B1,B2,B3,B4,B5,B6,B7]
HSEROUT [hex2 b5,"/",hex2 b6,"/",hex2 b7,hex2 b3,".",hex2 b2,".",hex2 b1]
PAUSE 10
lp1:  ADCIN 0,X : ADCIN 1,y
     HSEROUT [dec3 x, "!", dec3 Y, "$"]
     PAUSE time
    GOTO lp1
'********************************* Interrupt Timer ***********************************
myint:
DISABLE
timer2_int:
  IF pir1.1 = 1 THEN
  cnt = cnt + 1
    IF cnt >= t THEN
      cnt = 0
      I2CREAD DPIN,CPIN,$D0,$00,[B1,B2,B3]
      HSEROUT [hex2 b3,".",hex2 b2,".",hex2 b1]
      PAUSE time1
    ENDIF
  pir1.1=0
  ENDIF
'********************************* Interrupt USART ***********************************
usartinterrupt:
  IF pir1.5 = 1 THEN
    RCSTA = $90
    datain = RCREG
      WHILE RCIF
      discard = RCREG
      WEND
```

```
SELECT CASE datain
  CASE "a"
  time = 1
  time1 = 5
  t = 1500
  CASE "b"
  time = 11
  time1 = 10
  t = 1500
  CASE "c"
  time = 100
  time1 = 100
  t = 661
  CASE "d"
  time = 1000
  time1 = 1000
  t = 70
  CASE "e"
  time = 2000
  time1 = 2000
  t = 35
  CASE "f"
  time = 5000
  time1 = 5000
  t = 13
  CASE "z"
  time = 0
  I2CREAD DPIN,CPIN,$D0,$00,[B1,B2,B3,B4,B5,B6,B7]
  HSEROUT [hex2 b5,"/",hex2 b6,"/",hex2 b7,hex2 b3,".",hex2 b2,".",hex2 b1]
  t2con.2 = 0
  GOTO start
END SELECT
```

```
    RCSTA = $90
  ENDIF
  RESUME
ENABLE
'************************************ End Program ***********************************
```

# ภาคผนวก ข

# โปรแกรม Computer Based Data Logger

```
Dim rdat, date_mcu, v1, v2, data As String

Private Sub Command1_Click()
MSComm1.Output = "a"
End Sub

Private Sub Command2_Click()
MSComm1.Output = "b"
End Sub

Private Sub Command3_Click()
MSComm1.Output = "c"
End Sub

Private Sub Command4_Click()
MSComm1.Output = "d"
End Sub

Private Sub Command5_Click()
MSComm1.Output = "e"
End Sub

Private Sub Command6_Click()
MSComm1.Output = "f"
End Sub

Private Sub Command7_Click()
End
End Sub

Private Sub Pause_Click()
If MSComm1.PortOpen = True Then
```

```
    MSComm1.PortOpen = False
Else
    MSComm1.PortOpen = True
End If
End Sub

Private Sub Start_Click()
If MSComm1.PortOpen = False Then
    MSComm1.PortOpen = True
    End If
End Sub

Private Sub Clear_Click()
 If MSComm1.PortOpen = True Then
    MSComm1.PortOpen = False
    End If
    initGrid
End Sub

Private Sub Stop_Click()
    MSComm1.Output = "z"
End Sub

Private Sub Exit_Click(Index As Integer)
End
End Sub

Private Sub Create_Click()
    strname = TxtPath.Text
    If Dir(strname, vbDirectory) = "" Then
        MkDir (strname)
    End If
```

```
End Sub

Private Sub Save()
   strname = TxtPath.Text & "\" & DateStr & ".text"
   Open strname For Append As #1
      Print #1, data
   Close #1
End Sub

Private Sub Form_Load()
   rdat = ""
   initGrid
End Sub

Public Sub initGrid()
Dim w As Double
   w = (MSFlexGrid1.Width) / 78
   MSFlexGrid1.Rows = 1
   Call SetHeadMfg(MSFlexGrid1, 0, "Date & Time", flexAlignCenterCenter, w * 25)
   Call SetHeadMfg(MSFlexGrid1, 1, "Voltage (V)", flexAlignCenterCenter, w * 25)
   Call SetHeadMfg(MSFlexGrid1, 2, "Current (mA)", flexAlignCenterCenter, w * 25)
End Sub

Private Sub MSComm1_OnComm()
Listen
End Sub

Sub Listen()
      On Error Resume Next
      rdat = MSComm1.Input
   Filter
   Save
```

```
    InsertData
    rdat = ""
    data = ""
End Sub

Private Sub Filter()
    On Error Resume Next
    date_mcu = Mid(rdat, InStr(rdat, "/") - 2, 8)
    On Error Resume Next
    date_mcu = Mid(rdat, InStr(rdat, ".") - 2, 8)

    v1 = ""
    v1 = Mid(rdat, InStr(rdat, "!") - 3, 3)
    v1 = Round((CDbl(v1) * 5) / 255, 2)
    v1 = Format(v1, "#0.00")

    v2 = ""
    v2 = Mid(rdat, InStr(rdat, "$") - 3, 3)
    v2 = Round((CDbl(v2) * 20) / 255, 2)
    v2 = Format(v2, "#0.00")

    data = date_mcu & " " & time_mcu & v1 & " " & v2
End Sub

Sub InsertData()
    Dim iRow As Long
    On Error Resume Next
    MSFlexGrid1.Rows = MSFlexGrid1.Rows + 1
    iRow = MSFlexGrid1.Rows - 1

    SetDataMfg MSFlexGrid1, iRow, 0, date_mcu, flexAlignCenterCenter
    date_mcu = ""
```

```
    SetDataMfg MSFlexGrid1, iRow, 1, v1, flexAlignCenterCenter
    v1 = ""
    SetDataMfg MSFlexGrid1, iRow, 2, v2, flexAlignCenterCenter
    v2 = ""

    If MSFlexGrid1.Rows - 1 > 0 Then
        MSFlexGrid1.TopRow = MSFlexGrid1.Rows - 1
    End If
End Sub

Module1
Public strname As String

Module2
Public Nooke As Boolean
Public Const SORT_NONE = flexSortNone
Public Const SORT_GEN_ASC = flexSortGenericAscending
Public Const SORT_GEN_DESC = flexSortGenericDescending
Public Const SORT_NUM_ASC = flexSortNumericAscending
Public Const SORT_NUM_DESC = flexSortNumericDescending
Public Const SORT_NOCASE_ASC = flexSortStringNoCaseAscending
Public Const SORT_NOCASE_DESC = flexSortStringNoCaseDescending
Public Const SORT_STRING_ASC = flexSortStringAscending
Public Const SORT_STRING_DESC = flexSortStringDescending

Public Sub DeleteRowFlexgrid(ByVal objFlexgrid As MSFlexGrid, ByVal iRow As Integer)
    Dim R As Integer, C As Integer

    If iRow < objFlexgrid.Rows - 1 Then
        For R = iRow To objFlexgrid.Rows - 2
            For C = 0 To objFlexgrid.Cols - 1
                objFlexgrid.TextMatrix(R, C) = objFlexgrid.TextMatrix(R + 1, C)
```

```
        Next C
      Next R
      objFlexgrid.Rows = objFlexgrid.Rows - 1
    Else
      If iRow = objFlexgrid.Rows - 1 Then
        objFlexgrid.Rows = objFlexgrid.Rows - 1
      End If
    End If
End Sub

Public Sub ResizeColumnFlexgrid(ByVal objFlexgrid As MSFlexGrid, Optional ByVal iSubCol
As Integer = 10)
    On Error Resume Next
    Dim i As Integer, iWidth As Double, persentWidth As Double, resultWidth As Double
    Dim pieceWidth As Double

    For i = 0 To objFlexgrid.Cols - 1
      iWidth = iWidth + objFlexgrid.ColWidth(i)
    Next i
    For i = 0 To objFlexgrid.Cols - 1
      persentWidth = (100 * objFlexgrid.ColWidth(i)) / iWidth
      resultWidth = ((persentWidth * objFlexgrid.Width) / 100)
      If resultWidth > 0 Then
        objFlexgrid.ColWidth(i) = resultWidth
      Else
        objFlexgrid.ColWidth(i) = 0
      End If
    Next i
    iWidth = 0
    For i = 0 To objFlexgrid.Cols - 1
      iWidth = iWidth + objFlexgrid.ColWidth(i)
    Next i
```

```
    If objFlexgrid.Appearance = flex3D Then
        iWidth = (objFlexgrid.Width - 100) - iWidth
    Else
        iWidth = objFlexgrid.Width - iWidth
    End If
    pieceWidth = iWidth / objFlexgrid.Cols
    For i = 0 To objFlexgrid.Cols - 1
        resultWidth = objFlexgrid.ColWidth(i) + pieceWidth
        If resultWidth > 0 Then
            objFlexgrid.ColWidth(i) = resultWidth
        Else
            objFlexgrid.ColWidth(i) = 0
        End If
    Next i
End Sub

Public Sub SortColmfg(mfgR As MSFlexGrid, intCol As Long, intSort As Integer)
    With mfgR
        .Col = intCol
        .FixedRows = 1
        .Sort = intSort
        .Row = 1
    End With
End Sub

Public Function SetHeadMfg(ControlName As MSFlexGrid, intCol As Integer, StrDetial As
String, IntFixAlm As AlignmentSettings, IntColWidth As Integer)
    With ControlName
        .TextMatrix(0, intCol) = StrDetial
        .Col = intCol: .Row = 0
        .CellAlignment = IntFixAlm
        .CellPictureAlignment = IntFixAlm
```

```
    .ColWidth(intCol) = IntColWidth
    .Col = intCol
  End With
End Function

Public Sub SetHeaderFlexgrid(ControlName As MSFlexGrid, intCol As Integer, StrDetial As
String, IntFixAlm As AlignmentSettings, IntColWidth As Integer)
  With ControlName
    .TextMatrix(0, intCol) = StrDetial
    .CellAlignment = IntFixAlm
    .CellPictureAlignment = IntFixAlm
    .ColWidth(intCol) = IntColWidth
  End With
End Sub

Public Function SetDataMfg(ControlName As MSFlexGrid, intRow As Long, intCol As Integer,
ByVal strData As String, ByVal IntColAlm As Integer)
  With ControlName
    .TextMatrix(intRow, intCol) = strData
    .ColAlignment(intCol) = IntColAlm
  End With
End Function

Public Function SetPictureMfg(ControlName As MSFlexGrid, intRow As Long, intCol As
Integer, strPicture As Variant, ByVal IntColAlm As Integer)
  With ControlName
    .Row = intRow
    .Col = intCol
    If Trim(strPicture) = "" Then
      Set .CellPicture = LoadPicture("")
    Else
      Set .CellPicture = strPicture
```

```
      End If
      .CellPictureAlignment = IntColAlm
    End With
End Function

Public Sub SetSelFocus(ControlName As Control, StrDataCode As Variant, IntFindCol As
Integer)
    With ControlName
      For i = 1 To .Rows - 1
        If Trim(.TextMatrix(i, IntFindCol)) = Trim(StrDataCode) Then
          .TopRow = i
          .Row = i
          .Col = 0: .ColSel = .Cols - 1
          Exit For
        End If
      Next
    End With
End Sub

Private Sub SetPictureGrid(curForm As Form, curGrid As Control, curRow As Integer, curCol
As Integer, curData As String)
    On Error Resume Next
    curGrid.Col = curCol
    curGrid.Row = curRow
    Select Case UCase(curData)
      Case "ZEMPTY"
        Set curGrid.CellPicture = Nothing                                    'Empty
Sign
      Case "ZUNCHECK"
        Set curGrid.CellPicture = curForm.ilsCheckIcon.ListImages(2).Picture    'UnCheck
Sign
      Case "ZCHECKGREEN"
```

```
            Set curGrid.CellPicture = curForm.ilsCheckIcon.ListImages(3).Picture      'Check Sign
        Case "ZCHECKBLUE"
            Set curGrid.CellPicture = curForm.ilsCheckIcon.ListImages(4).Picture      'Check Sign
        Case "ZCHECKRED"
            Set curGrid.CellPicture = curForm.ilsCheckIcon.ListImages(5).Picture      'Check Sign
        Case "ZCHECKPURPLE"
            Set curGrid.CellPicture = curForm.ilsCheckIcon.ListImages(6).Picture      'Check Sign
        Case "ZCHECKGRAY"
            Set curGrid.CellPicture = curForm.ilsCheckIcon.ListImages(7).Picture      'Check Sign
        Case Else
            curGrid.TextMatrix(curRow, curCol) = curData
    End Select
End Sub

Public Sub HighLightGirdRow(curGrid As Control)
    Dim i As Long
    Dim j As Long
    If curGrid.Rows > 1 Then
        For i = 1 To curGrid.Rows - 1
            curGrid.Row = i
            For j = 0 To curGrid.Cols - 1
                curGrid.Col = j
                If curGrid.Row Mod 2 = 0 Then
                    curGrid.CellBackColor = &HE0E0E0
                Else
                    curGrid.CellBackColor = vbWhite
                End If
            Next j
        Next i
    End If
End Sub
```

```
Public Sub Call_PopupRClickMenu(myForm As Object, myControl As Control, Button As
Integer, Y As Single, myMenu As Menu)
  On Error GoTo ErrorHandler
    If Button = 2 And Y > myControl.RowHeight(0) Then
      Button = 99
      myControl.Row = Y \ myControl.RowHeight(1) + myControl.TopRow - 1        'Paint
Highlight of Row in another row differ from current selected row
      myControl.Col = 0
      myControl.ColSel = myControl.Cols - 1
      myForm.PopupMenu myMenu                                                  'Pop up
Menu
      Button = 98
    End If
ErrorHandler:
  If Button = 99 Then myForm.PopupMenu myMenu
  If Err.Number = 30090 Then Exit Sub
End Sub

Public Sub AlignTextboxInFlexgrid(ByVal objTextbox As TextBox, ByVal objFlexgrid As
MSFlexGrid, ByVal nCol As Integer)
  ' *** This sub is align Textbox into Flexgrid by specify the column in the last row.
  objTextbox.BorderStyle = 0
  objFlexgrid.Row = objFlexgrid.Rows - 1
  objFlexgrid.Col = nCol
  objTextbox.Top = objFlexgrid.Top + objFlexgrid.CellTop
  objTextbox.Left = objFlexgrid.CellLeft + objFlexgrid.Left
  objTextbox.Width = objFlexgrid.CellWidth
  objTextbox.Height = objFlexgrid.CellHeight
End Sub
```

**Module3**

```
Public Function DateStr() As String
```

```
    Dim filSave0$, filSave1$, filSave2$, filSave3$, filSave4$
    Dim PosTXT1%, PosTXT2%
    filSave0$ = Date
    PosTXT1% = InStr(filSave0$, "/")
    filSave1$ = Format(Left(filSave0$, PosTXT1% - 1), "00")
    PosTXT2% = InStr(PosTXT1% + 1, filSave0$, "/")
    filSave2$ = Right(Left(filSave0$, PosTXT2% - 1), PosTXT2% - PosTXT1% - 1)
    filSave3$ = Format(Mid(filSave0$, PosTXT1% + 1, 2), "00")
    filSave4$ = Format(Mid(filSave0$, PosTXT2% + 1, 4), "0000")
    DateStr = filSave1$ + "_" + filSave2$ + "_" + filSave4$
End Function

Public Function MouthStr() As String
    Dim filSave0$, filSave1$, filSave2$, filSave3$, filSave4$
    Dim PosTXT1%, PosTXT2%
    filSave0$ = Date
    PosTXT1% = InStr(filSave0$, "/")
    filSave1$ = Format(Mid(filSave0$, 1, 2), "00")
    PosTXT2% = InStr(PosTXT1% + 1, filSave0$, "/")
    filSave2$ = Format(Mid(filSave0$, PosTXT2% + 1, 4), "0000")
    MouthStr = filSave1$ + filSave2$
End Function

Public Function TimeStr() As String
    Dim filSave0$, filSave1$, filSave2$, filSave3$, filSave4$, filSave5$
    Dim PosTXT1%, PosTXT2%, PosTXT3%

    filSave0$ = Time
    PosTXT1% = InStr(filSave0$, ":")
    filSave1$ = Format(Left(filSave0$, PosTXT1% - 1), "00")
    PosTXT2% = InStr(PosTXT1% + 1, filSave0$, ":")
```

```
    filSave2$ = Format(Right(Left(filSave0$, PosTXT2% - 1), PosTXT2% - PosTXT1% - 1),
"00")
    PosTXT3% = InStr(PosTXT2% + 1, filSave0$, " ")
    filSave3$ = Format(Right(Left(filSave0$, PosTXT3% - 1), PosTXT3% - PosTXT2% - 1),
"00")
    filSave4$ = Right(filSave0$, 2)
    If filSave4$ = "PM" Then
       filSave1$ = filSave1$ + 12
    End If
    filSave5$ = filSave1$ + filSave2$ + filSave3$
    TimeStr = filSave5
End Function
```

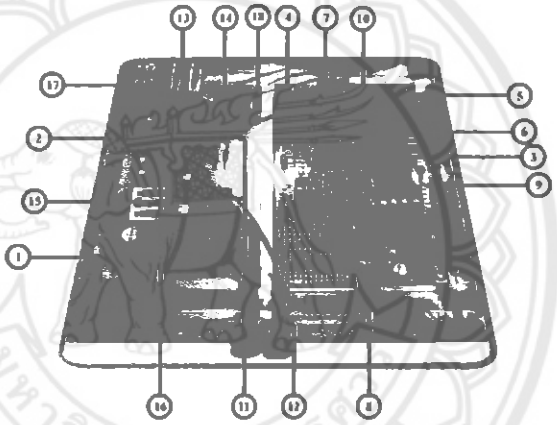
# ภาคผนวก ค
# คู่มือการใช้งาน

## คุณสมบัติของ Computer Based Data Logger

Computer Based Data Logger จะควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ PIC16F877 โดยจะใช้สำหรับเก็บข้อมูลแรงดันไฟฟ้าไฟตรง 0 – 5 โวลต์และกระแสไฟฟ้าไฟตรง 0 - 20 มิลลิแอมป์ จำนวน 2 ช่องสัญญาณ ข้อมูลที่ได้จะถูกเก็บในรูปไฟล์ Text ซึ่งจะประกอบด้วย ชื่อไฟล์ วัน เวลา และแรงดันไฟฟ้าไฟตรงและกรแสไฟฟ้าไฟตรง จำนวน 2 ช่องสัญญาณ โดยจะติดต่อผู้ใช้ผ่านหน้าจอคอมพิวเตอร์

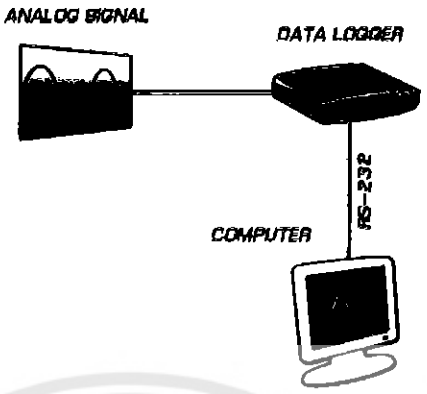
ในการบันทึกข้อมูลสามารถตั้งค่าหน่วงเวลาในการบันทึกข้อมูลได้ โดยค่าหน่วงเวลาจะมีค่า 1 มิลลิวินาที ถึง 5,000 มิลลิวินาที

## ส่วนประกอบของ Computer Based Data Logger



1. จุดเชื่อมต่อไฟฟ้าไฟตรง +/-15 โวลต์
2. จุดเชื่อมต่อไฟฟ้าไฟตรง 15 โวลต์
3. โมโครคอนโทรลเลอร์ 16F877
4. ไอซีฐานเวลาเบอร์ DS1307
5. พอร์ตขนาน DB25
6. ไฟแสดงผลแอลอีดี (LED)
7. จุดรับไฟฟ้าไฟตรง 5 โวลต์
8. จุดเชื่อมต่อไฟฟ้าไฟตรง 12 โวลต์
9. พอร์ตอนุกรม RS-232
10. ปุ่มรีเซตเครื่อง
11. ไอซีออปแอมป์ LM358
12. จุดวัดกระแส
13. ไอซีเรกกูเลเตอร์ 7805
14. ตัวต้านทานปรับค่าได้ 10 kΩ
15. ไอซีเรกกูเลเตอร์ 7805
16. ตัวต้านทานปรับค่าได้ 10 kΩ
17. จุดจ่ายไฟฟ้าไฟตรง 5 โวลต์
18. จุดจ่ายไฟฟ้าไฟตรง 5 โวลต์

## ลักษณะการทำงานของ Computer Based Data Logger

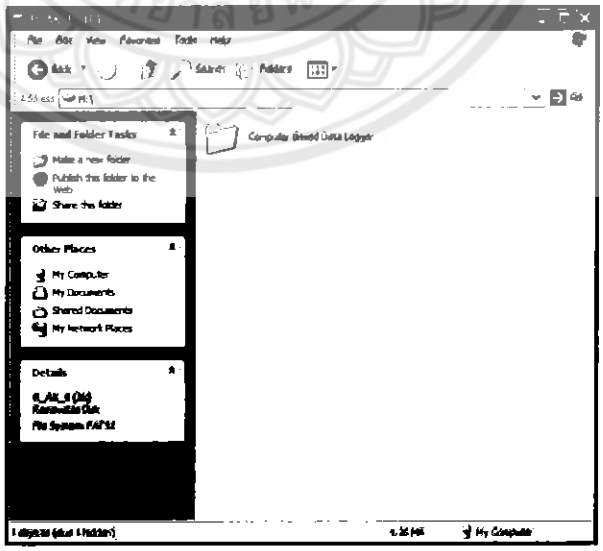



เริ่มจากการรับแรงดันไฟฟ้าไฟตรงเข้ามาผ่านวงจรแปลงสัญญาณอนาลอกเป็นข้อมูลดิจิตอลซึ่งอยู่ภายในไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะทำการส่งข้อมูลที่ได้จากการแปลง พร้อมทั้งข้อมูล วัน เวลา ผ่านการสื่อสารแบบอนุกรม (RS – 232) ไปยังคอมพิวเตอร์ คอมพิวเตอร์จะ มีโปรแกรมสำหรับรับข้อมูล Computer Based Data Logger โปรแกรมจะทำการแสดงข้อมูลในรูปแบบตัวเลข พร้อมทั้งบันทึกข้อมูลที่ได้เก็บไว้ในรูปแบบไฟล์ Text

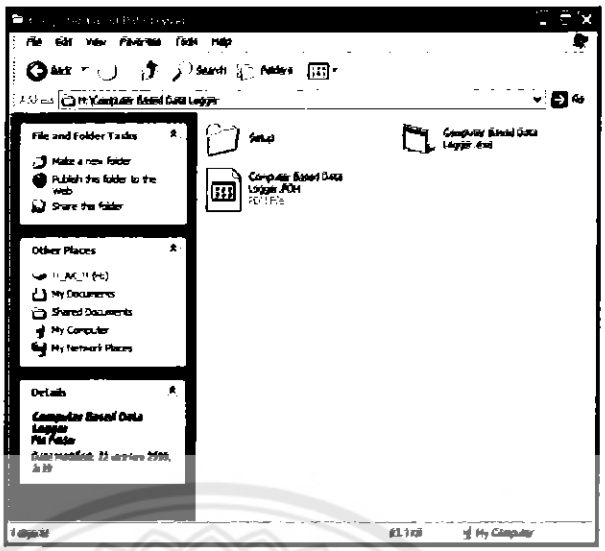
## ขั้นตอนการติดตั้งโปรแกรม

การติดตั้งโปรแกรม Computer Based Data Logger

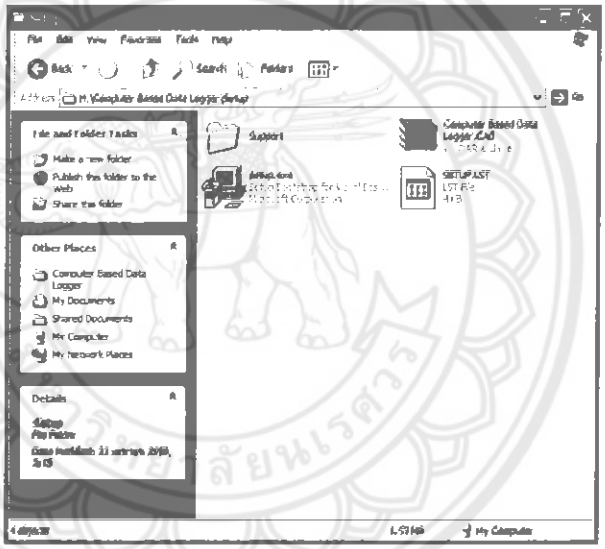
1. ใส่แผ่น ซีดี Computer Based Data Logger



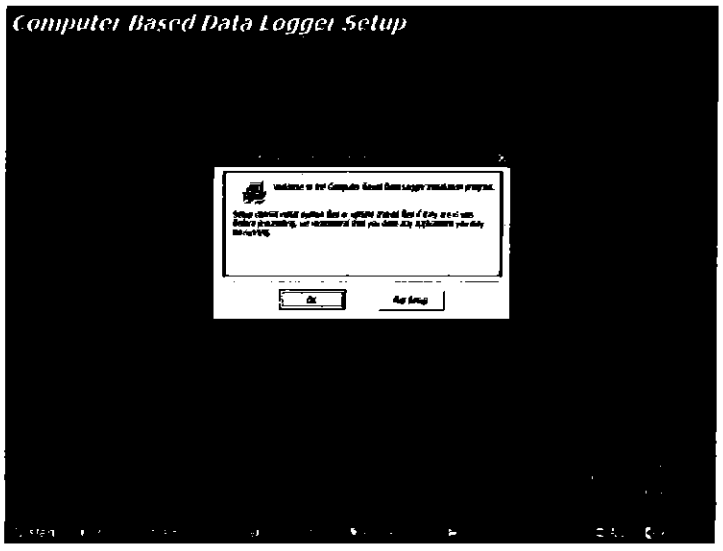
2. ดับเบิลคลิกโฟล์เดอร์ Setup



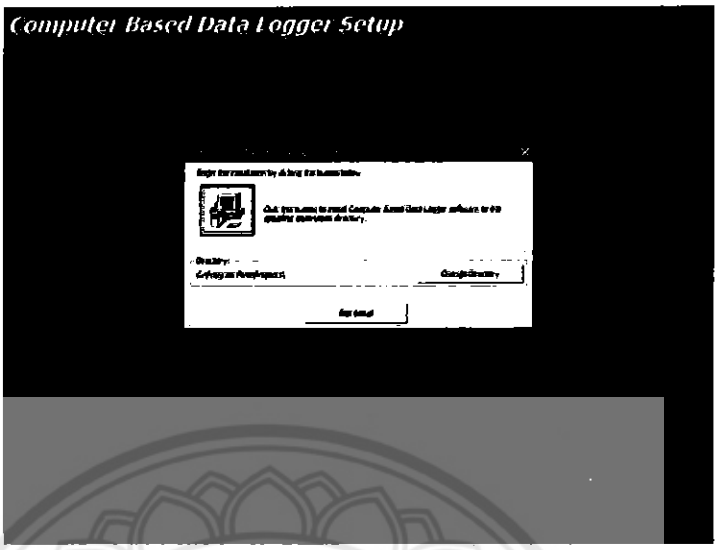
3. ดับเบิลคลิกไฟล์ Setup.exe



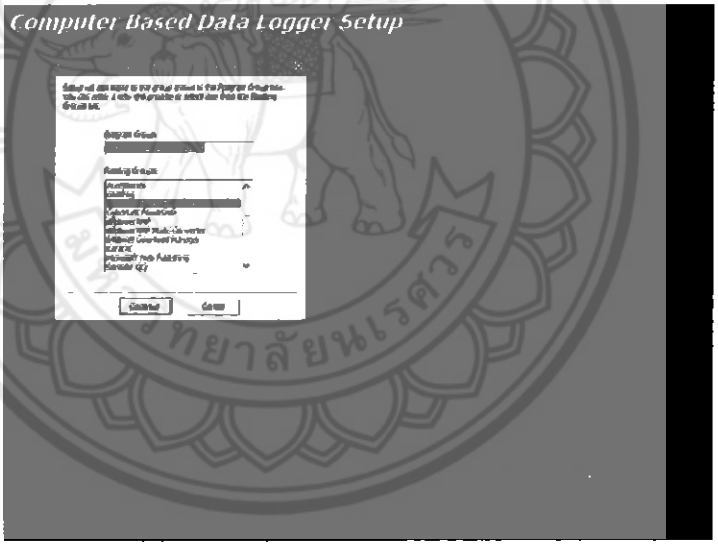
4.เริ่มการติดตั้งคลิก OK



5. เลือกไดเรกทอรี่ที่ต้องการ แล้วกด

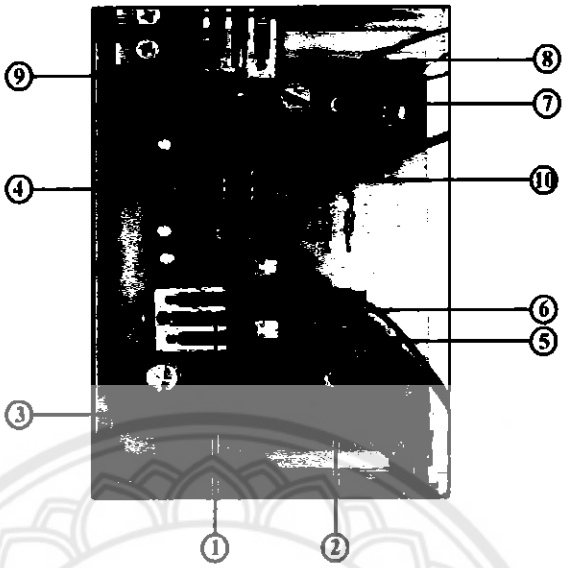



6. คลิก Continue



7. โปรแกรมติดตั้งจะทำงานโดยอัตโนมัติ เมื่อติดตั้งเรียบร้อยแล้วคลิก OK เป็นอันเสร็จสิ้นขั้นตอนการติดตั้งและพร้อมสำหรับการทำงาน

**การเชื่อมต่อจุดรับกระแสและแรงดันไฟฟ้าไฟตรง**

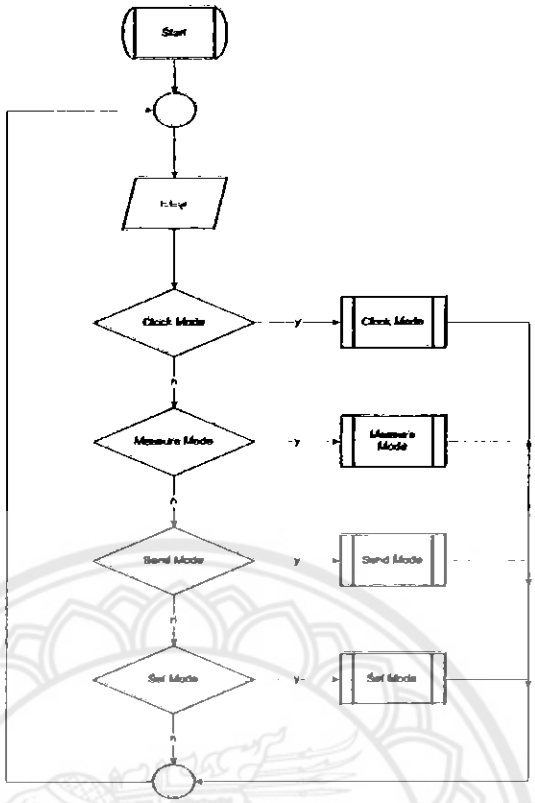



การรับสัญญาณแรงดันไฟฟ้าจากภายนอก สามารถรับได้ทั้งหมด 1 ช่องสัญญาณ(4) เมื่อเชื่อมต่อสัญญาณเข้ามาที่ช่องสัญญาณ(4) จากนั้นสัญญาณจะผ่านวงจรเรกกูเลเตอร์ซึ่งจะใช้ไอซีออปแอมป์เบอร์ 7805(9) ซึ่งจะทำหน้าที่ป้องกันแรงดันที่ป้อนเข้าไมโครคอนโทรลเลอร์ไม่ให้เกิน 5 โวลต์ ผ่านตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม(7) ไปยังเอาต์พุต(8) ต่อกับช่องรับสัญญาณตัวที่ 1 เพื่อเข้าวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ในไมโครคอนโทรลเลอร์ และจะใช้ไอซีออปแอมป์เบอร์ 7805(1) ซึ่งจะทำหน้าที่ป้องกันแรงดันที่ป้อนเข้าไมโครคอนโทรลเลอร์ไม่ให้เกิน 5 โวลต์ ผ่านตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม(2) ผ่านวงจรแปลงแรดันเป็นกระแสจะใช้ไอซีออปแอมป์เบอร์ LM 358(6) แล้วผ่านวงจรแปลงกระแสเป็นแรงดันจะใช้ไอซีออปแอมป์เบอร์ LM 358(10) ไปยังเอาต์พุต(5) ต่อกับช่องรับสัญญาณตัวที่ 2 เพื่อเข้าวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ในไมโครคอนโทรลเลอร์

**วิธีการใช้งาน Computer Based Data Logger**

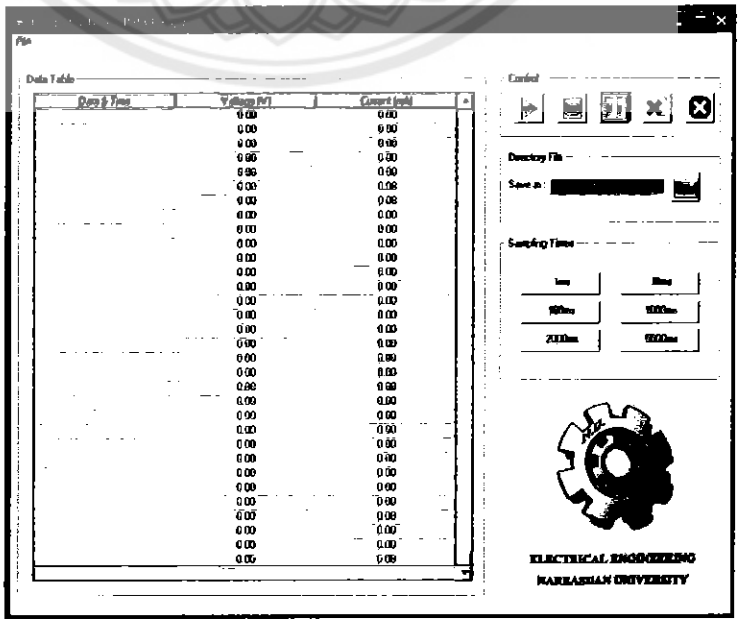
ในการใช้งาน Computer Based Data Logger จะแยกการทำงานออกเป็น 5 ส่วน คือ

- การแสดงแรงดันและกระแสไฟฟ้าไฟตรง
- การบันทึกข้อมูล
- การส่งข้อมูลไปคอมพิวเตอร์
- การแสดงวันเวลา
- การตั้งค่าการทำงาน




**การแสดงแรงดันและกระแสไฟฟ้าไฟตรง**

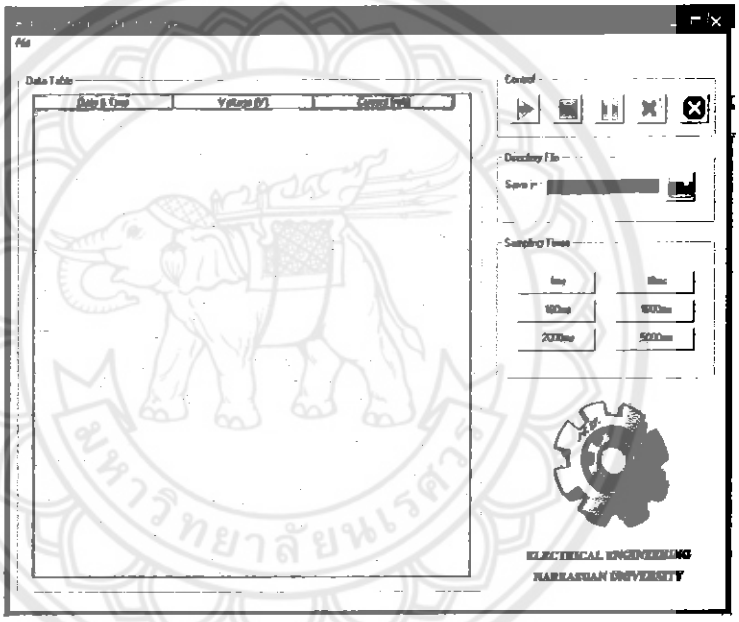
ส่วนของการแสดงผลเมื่อมีการรับข้อมูลเข้ามาจะแสดงค่าของวันที่ เวลา และแรงดันกับกระแส ข้อมูลที่รับมาแยกออกเป็นสามส่วน คือส่วนที่เป็น วันที่กับเวลา แรงดัน และกระแส วันที่และเวลาจะแสดงในช่อง Date&Time แรงดัน จะแสงในช่อง Voltage(V) และกระแสจะแสดงในช่อง Current(mA)



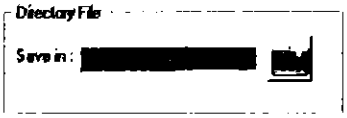
เริ่มต้น จะแสดงวันที่และเวลาเรียงลงมาในช่อง Date&Time ต่อจากนั้นแสดงแรงดันและกระแส ที่รับเข้ามา เรียงลงมาเรื่อยๆ ในรูปแบบทศนิยมสองตำแหน่ง โดยค่าแรงดันจะอยู่ระหว่าง 0 - 5 โวลต์ และกระแสอยู่ระหว่าง 0 - 20 มิลลิแอมป์ เมื่อครบทุกๆ 1 นาที จะแสดงค่าเวลา ต่อจากนั้นจะกลับไปแสดงแรงดันและกระแสเรียงลงมาเมื่อมีการหยุดการทำงานจะแสดงวันที่และเวลาเป็นลำดับสุดท้าย

ในการแสดงแรงดันไฟฟ้าจะเป็นการนำสัญญาณที่ได้จากการแปลงอนาลอกเป็นดิจิตอลมาแสดงผลบนหน้าจอคอมพิวเตอร์และจะบันทึกข้อมูลโดยอัตโนมัติ ข้อมูลที่ได้จะประกอบไปด้วยแรงดันไฟฟ้า และกระแสไฟฟ้า แสดงด้วยทศนิยม 2 ตำแหน่ง มีขั้นตอนการใช้งาน ดังนี้

1.เปิดโปรแกรม Computer Based Data Logger.exe

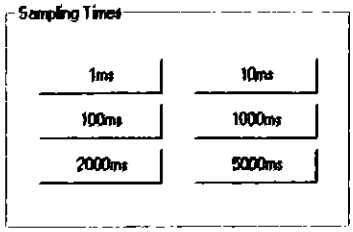



2.สร้างโฟล์เดอร์ที่ใช้ในการจัดเก็บข้อมูล โดยการตั้งชื่อใน Directory

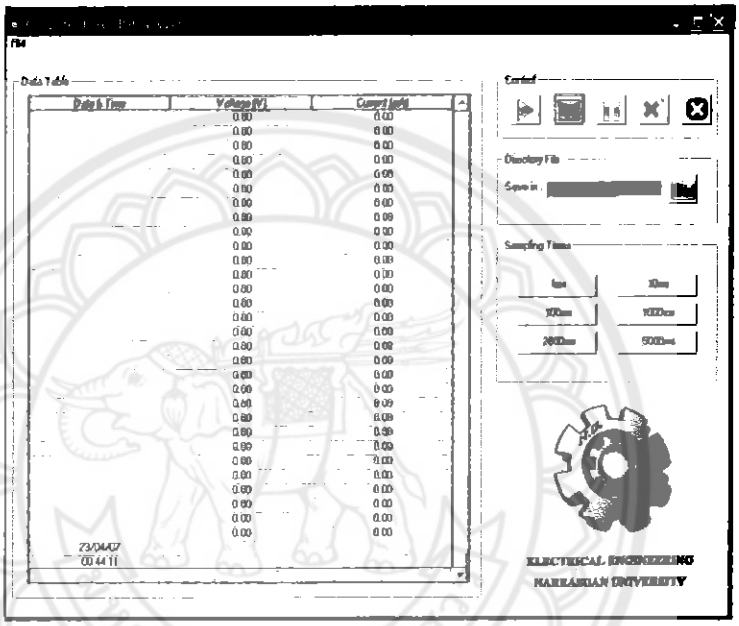



3. กดปุ่ม ในเฟรม Control การเปิดการเชื่อมต่อระหว่าง คอมพิวเตอร์กับไมโครคอนโทรลเลอร์

4. จากนั้นเลือกค่าหน่วงเวลาในเฟรม Sampling Time




5.การแสดงผลการทำงาน

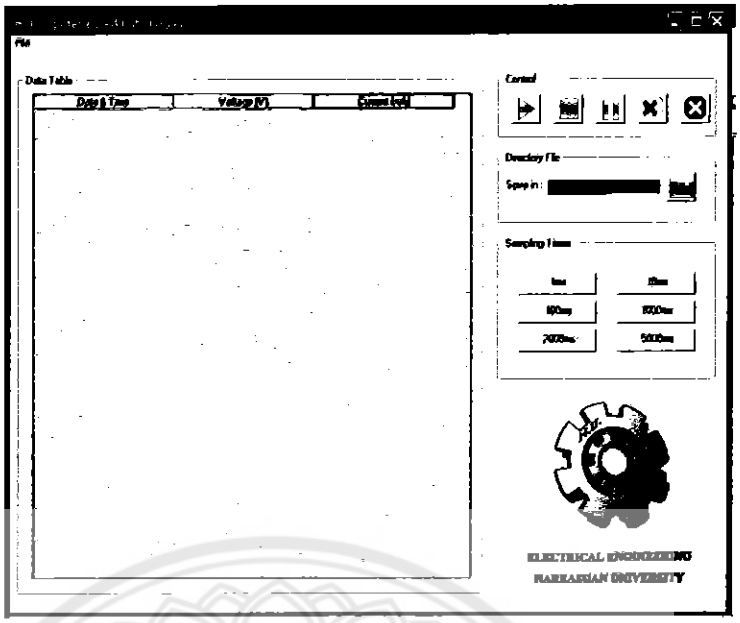



**การบันทึกข้อมูล**

การบันทึกค่าในรูปของ Text ไฟล์ ข้อมูลจะถูกเก็บให้อยู่ในรูปของตัวเลข เริ่มต้นบันทึกวันที่ บรรทัดถัดไป บันทึกเวลา ต่อจากนั้นบันทึกแรงดันและกระแสบรรทัดถัดไป เรียงลงมาเรื่อยๆ ในรูปแบบทศนิยมสองตำแหน่ง โดยค่าแรงดันจะอยู่ระหว่าง 0 - 5 โวลต์ และกระแสอยู่ระหว่าง 0 - 20 มิลลิแอมป์ เมื่อครบทุกๆ 1 นาที จะบันทึกค่าเวลา ต่อจากนั้นจะกลับไปบันทึกแรงดันและกระแสเรียงลงมา เมื่อมีการหยุดการทำงานจะบันทึกวันที่ บรรทัดถัดไปเวลาเป็นลำดับสุดท้าย

**การส่งข้อมูลเข้าคอมพิวเตอร์**

ในการส่งข้อมูลไปบันทึกในคอมพิวเตอร์ จะทำหน้าที่นำข้อมูลที่ต้องการบันทึก คือ แรงดันไฟฟ้าไฟตรงและกระแสไฟฟ้าไฟตรง 2 ช่องสัญญาณและข้อมูลของวันเวลาที่ทำการบันทึกส่งเข้าคอมพิวเตอร์ผ่านพอร์ตอนุกรม (RS – 232) ตามค่าหน่วงเวลา คอมพิวเตอร์จะมีโปรแกรมสำหรับบันทึกข้อมูลที่รับเข้ามา (Data Logger) ไปเก็บไว้ในรูปแบบของไฟล์ Text และจะแสดงข้อมูลของแรงดันไฟฟ้าและกระแสไฟฟ้า การส่งข้อมูลไปบันทึกในคอมพิวเตอร์



ในขณะทำการบันทึกข้อมูลถ้าต้องการหยุดการบันทึกชั่วคราวทำได้โดยการกดที่ปุ่ม "Pause" และถ้าต้องการให้เริ่มบันทึกต่อทำได้โดยการกดปุ่ม "Run" หรือกด "Pause" อีกครั้งถ้าต้องการเคลียร์ข้อมูลในตารางแสดงผลทำได้โดยการกดปุ่ม "Clear" และถ้าต้องการหยุดหรือเปลี่ยนค่าหน่วงเวลาในการบันทึกข้อมูลทำได้โดยการกดปุ่ม "Stop"

การแสดงวันเวลา

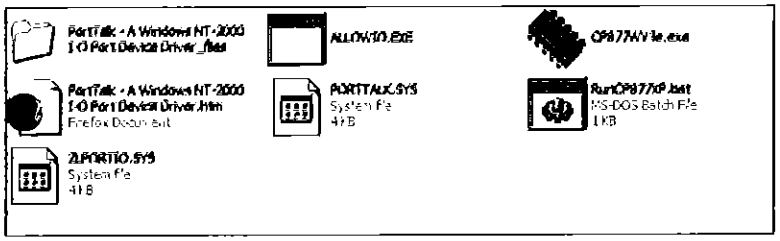
ในการแสดงวันเวลาจะเป็นการแสดงข้อมูลของวันและเวลาของComputer Based Data Logger เพื่อให้ผู้ใช้ตรวจสอบว่าข้อมูลของวันและเวลานั้นถูกต้องหรือไม่ โดยข้อมูลของวันและเวลาที่นำมาแสดงจะส่งมาจากไอซีฐานเวลาเบอร์ DS1307 ถ้าพบว่าข้อมูลของวันและเวลาไม่ถูกต้องเนื่องจากแบตเตอรี่ที่ใช้สำรองพลังงานไฟฟ้าให้กับไอซีฐานเวลาหมด ให้ทำการเปลี่ยนแบตเตอรี่แล้วตั้งค่าวันเวลาใหม่

**การตั้งค่าการทำงาน**

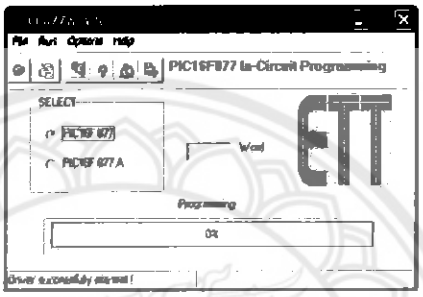
ในการใช้งานComputer Based Data Logger จะต้องมีการตั้งค่าการทำงานก่อนการใช้งาน ซึ่งจะประกอบไปด้วย การติดตั้งโปรแกรมวันเวลาของเครื่องและการติดตั้งโปรแกรมการทำงานเพื่อใช้เป็นข้อมูลในการบันทึก

- การติดตั้งโปรแกรมวันเวลา

1. เริ่มเปิดโปรแกรม cp877wve.exe

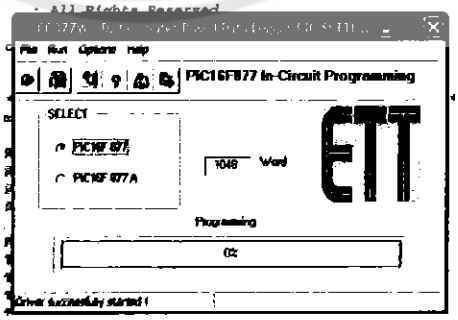



2. เลือกเบอร์ของไมโครคอนโทรลเลอร์ PIC16F877

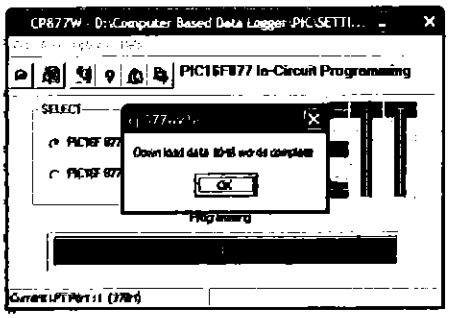



3. เปิดไฟล์ SETTIME.HEX

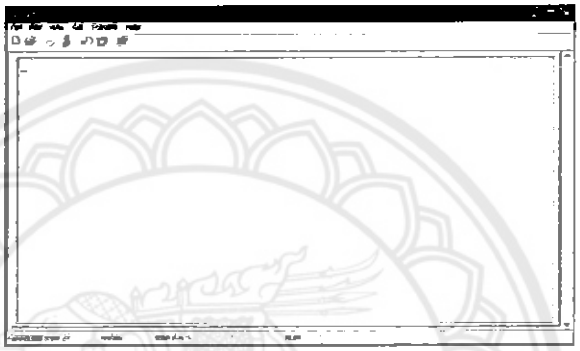
4. แสดงค่า Configuration word



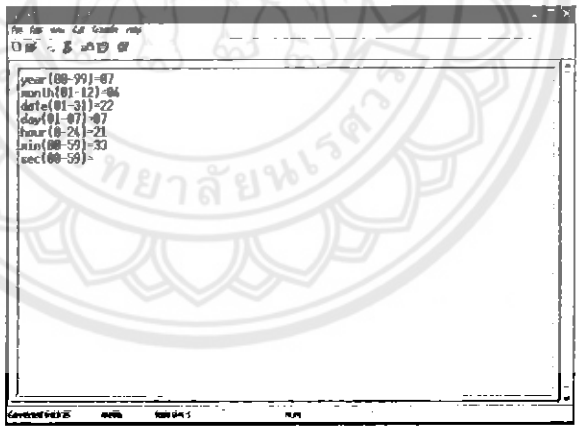
5. กด Erase + Program + Verify




6. เปิดโปรแกรม Hyper Terminal



7. กด "s" หรือ "S"เพื่อตั้งค่าเวลา

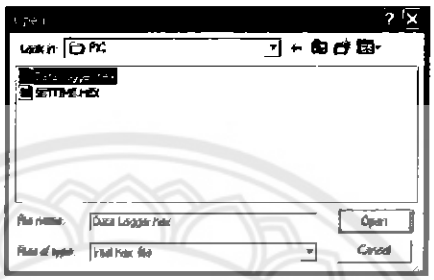



การตั้งวันเวลาของเครื่อง เป็นการกำหนดฐานเวลาให้กับComputer Based Data Logger เพื่อที่จะได้ระบุค่า วัน เวลา ลงในไฟล์ที่ทำการบันทึก เมื่อทำการเริ่มใช้งานครั้งแรกควรตั้งฐานเวลาให้กับเครื่องก่อน ซึ่งเมื่อตั้งค่าเสร็จแล้วในการทำงานครั้งต่อไปก็ไม่จำเป็นต้องมาตั้งค่าอีก นอกจากเมื่อใช้ไปนาน ๆ แล้ว วันเวลา เกิดคลาดเคลื่อนหรือแบตเตอรี่ที่ใช้จ่ายให้ไอซีฐานเวลาหมด จึงต้องมาตั้งค่าฐานเวลาใหม่ โดยจะเริ่มตั้งเวลาจากค่าวันในสัปดาห์ ซึ่งกำหนดให้วันอาทิตย์แทนด้วยเลข "07" วันจันทร์แทนด้วยเลข "01" และวันต่อ ๆ ไปให้แทนด้วยตัวเลขที่เพิ่มขึ้นเรื่อย ๆ จนถึงวันเสาร์ก็จะแทนด้วยเลข "06" และจากนั้นจะเป็นส่วนของวันที่ ซึ่งสามารถป้อนตัวเลขได้ในช่วง 01 – 31

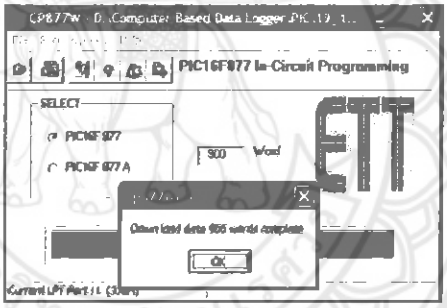
วัน ค่าของเดือนทำการป้อนตัวเลขได้ในช่วง 01 – 12 และค่าของปีทำการป้อนตัวเลขได้ในช่วง 00 – 99 ในรูปแบบของคริสต์ศักราช ในการกำหนดค่าของชั่วโมง สามารถป้อนตัวเลขในช่วง 00 – 23 ค่าของนาทีและวินาทีสามารถป้อนตัวเลขในช่วง 00 – 59 การตั้งวันเวลาของเครื่องมีขันตอนดังนี้

**- การติดตั้งโปรแกรมการทำงาน**

1. ทำตามขั้นตอนที่ 1 และ 2
2. เลือกไฟล์ Data Logger.hex

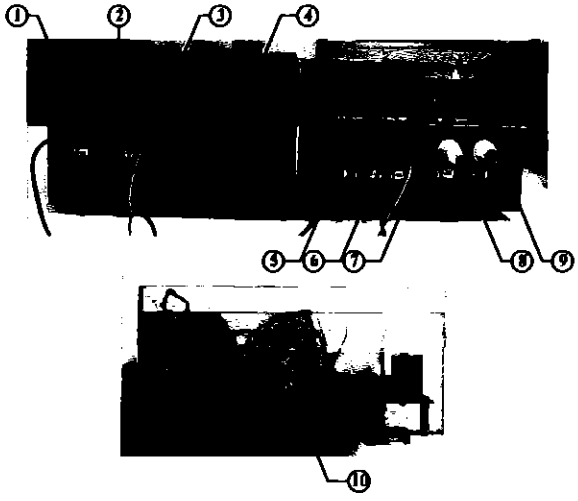



3. ทำตามขั้นตอนที่ 4 และ 5 เสร็จสิ้นการติดตั้ง

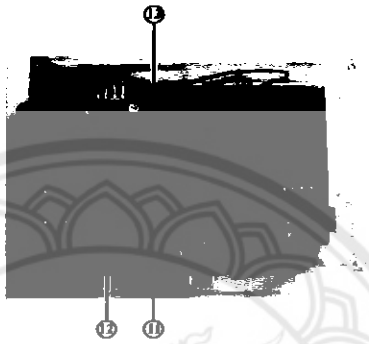



**- การติดตั้งและการเชื่อมต่ออุปกรณ์**

ในการทดลองจำเป็นต้องมีการเชื่อมต่ออุปกรณ์เพื่อจ่ายแรงดันไฟฟ้าไฟตรง +/- 15 Vdc เข้ากับวงจรเรกกูเลเตอร์ เพื่อใช้จำกัดค่าของแรงดันและกระแส ขั้นตอนในการติดตั้งมีดังนี้



เชื่อมต่อแหล่งจ่ายไฟฟ้าไฟตรงเข้าด้วยกัน จากรูปต่อแหล่งจ่ายไฟฟ้าไฟตรง +15 โวลต์(1) กับ จุดเชื่อมต่อไฟฟ้าไฟตรง +15 โวลต์(6) แล้วต่อกราวด์เข้าด้วยกันระหว่าง (2) กับ(5) และต่อแหล่งจ่ายไฟฟ้าไฟตรง -15 โวลต์(3) กับจุดเชื่อมต่อไฟฟ้าไฟตรง -15 โวลต์(7) จากนั้นต่อจุดวัดกระแส(10) เข้ากับ Multimeter ทำการเปิดสวิตซ์(4) ปรับค่า VR 10kΩ(11) ให้แรงดันอินพุตมีค่า 1 โวลต์ จากนั้นปรับ VR 500Ω(9) จนกระทั่งวัดกระแสได้ 4 มิลลิแอมป์




ปรับค่า VR 10kΩ(12) จนวัดแรงดันเอาต์พุตของวงจรได้ 1 โวลต์ ปรับกระแสอินพุตเป็น 20 มิลลิแอมป์ จากนั้นดูว่าแรงดันเอาต์พุตมีค่า 5 โวลต์หรือไม่ ถ้าไม่ให้ปรับค่า VR 500Ω(8) จนกระทั่งวัดแรงดันเอาต์พุตได้ 5 โวลต์ จากนั้นทำการเชื่อมต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์เพื่อทำการทดลองต่อไป

**คำแนะนำในการใช้งาน Computer Based Data Logger**

- ในการลบไฟล์ควรลบทั้งหมดทุกไฟล์ เพราะถ้าลบเป็นบางไฟล์อาจจะทำให้เกิดข้อผิดพลาดได้ในการใช้งานครั้งต่อไป
- ควรตรวจสอบว่าเวลาคลาดเคลื่อนหรือไม่ ถ้าคลาดเคลื่อนให้ทำการตั้งค่าของเวลาใหม่เพื่อความถูกต้องของข้อมูล

# ประวัติผู้เขียนโครงงาน

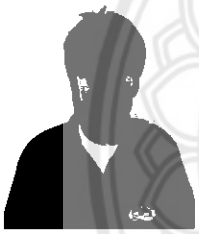
ชื่อ นายคมสัน อารีเอื้อ

ภูมิลำเนา 147 ม.19 ถ.สถิตย์นิมานกาล ต.กุดชมภู
อ.พิบูลมังสาหาร จ.อุบลราชธานี 34110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิบูลมังสาหาร
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

E-mail : ak1com@hotmail.com



ชื่อ นายวิทยา แสนธนโชคชัย

ภูมิลำเนา 105/2 ม.6 ต.ห้วยซ้อ อ.เชียงของ จ.เชียงราย 57140

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนห้วยซ้อวิทยาคม
  รัชมังคลาภิเษก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

E-mail : tatum501@hotmail.com