

เครื่องเก็บข้อมูลสำหรับงานอุตสาหกรรม

Industrial Data Logger



โดย
นายประทีป เฟื่องทอง
นายวาทัญญู สุขเสงี่ยม

เลขหมู่.....
เลขทะเบียน...42498
วัน, เดือน, ปี 24 พ.ค. 2545

| |
|---------|
| .b..... |
| .i..... |

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องเก็บข้อมูลสำหรับงานอุตสาหกรรม

Industrial Data Logger

โดย

นายประทีป เพ็งทอง 40010433

นายวชิรณัฐ สุขเสงี่ยม 40010666

อาจารย์ที่ปรึกษา

ผ.ศ. พรสุข รติโรจน์อนันต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

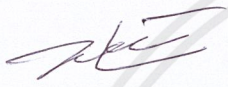
เรื่อง เครื่องบันทึกข้อมูลสำหรับงานอุตสาหกรรม

Industrial Data Logger

ผู้จัดทำ

1. นายประทีป เพ็งทอง 40010433

2. นายวาทัญญู สุขเสถียม 40010666


..... อาจารย์ที่ปรึกษา
(ผ.ศ. พรสุข รติโรจน์อนันต์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องเก็บข้อมูลสำหรับงานอุตสาหกรรม
(Industrial Data Logger)

โดย นายประทีป เฟื่องทอง 40010433
นายวาทัญญู สุขเสงี่ยม 40010666
อาจารย์ที่ปรึกษา ผศ.พรสุข รติโรจน์อนันต์

บทคัดย่อ

โครงการนี้เป็นการนำเสนอ เครื่องเก็บข้อมูลสำหรับระบบอุตสาหกรรม ซึ่งสามารถรับสัญญาณอินพุตที่มีความต่อเนื่องพร้อมกันได้ประมาณ 4 ช่องสัญญาณ โดยใช้ IC PCF8951 เป็นตัวแปลงสัญญาณจากอะนาลอกเป็นดิจิทัล ทำงานร่วมกับไมโครคอนโทรลเลอร์ตระกูล MCS - 51 ซึ่งทำหน้าที่ควบคุมและอินเตอร์เฟสกับคอมพิวเตอร์โดยผ่านทางพอร์ตอนุกรม ส่วนซอฟต์แวร์ที่ใช้จะเป็นโปรแกรม Delphi 5.0 ซึ่งจะช่วยในการวิเคราะห์และแสดงรูปสัญญาณ พร้อมทั้งสามารถบันทึกและพิมพ์รูปสัญญาณ พร้อมทั้งสามารถบันทึกและพิมพ์รูปสัญญาณออกจากเครื่องพิมพ์ได้

ABSTRACT

This project proposes data logger for Industrial System. It can receives 4-Channels analog signal at the same time from the out of temperature , flow ,weight and pressure transmitter. Each input signal is converted from analog by using IC PCF8951. The microcontroller family MCS-51 is used to communicate with computer by serial interface. The microsoft Delphi 5.0 is used for analyzing , displaying signal ,stores data and prints out to the printer.

สารบัญ

| | หน้าที่ |
|--|---------|
| บทที่ 1 บทนำ | 1 |
| บทที่ 2 ทฤษฎีและหลักการ | 2 |
| 2.1 หลักการของคานาต็อกเกอร์ | 2 |
| 2.2 พื้นฐานการสื่อสารแบบอนุกรม | 4 |
| 2.3 พื้นฐานการทำงานของ เอ/ดี คอนเวอร์เตอร์ | 6 |
| 2.4 โครงสร้างของโปรแกรม Delphi | 11 |
| 2.5 ขั้นตอนในการพัฒนาC-51 | 18 |
| บทที่ 3 การออกแบบ | 23 |
| 3.1 หลักการออกแบบฮาร์ดแวร์ | 23 |
| บทที่ 4 ผลการทดลอง | 30 |
| บทที่ 5 สรุปและวิจารณ์ผล | 40 |
| ภาคผนวก | 41 |
| กิตติกรรมประกาศ | 91 |
| บรรณานุกรม | 92 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

| | | หน้าที่ |
|-------------|--|---------|
| รูปที่ 2.1 | แสดงกระบวนการอบแห้ง | 2 |
| รูปที่ 2.2 | บล็อกไดอะแกรมของดาต้าล็อกเกอร์ | 3 |
| รูปที่ 2.3 | แสดงการส่งข้อมูลที่ใช้สัญญาณนาฬิกาตัวเดียวกัน | 5 |
| รูปที่ 2.4 | แสดงการเชื่อมต่อระหว่างตัวส่งกับตัวรับ | 5 |
| รูปที่ 2.5 | แสดงองค์ประกอบของเฟรม | 5 |
| รูปที่ 2.6 | แสดงการเลื่อนเฟสของสัญญาณนาฬิกา | 6 |
| รูปที่ 2.7 | แสดงกราฟคุณสมบัติของเอ็ดจ์ชิ่งขนาด 3 บิต | 7 |
| รูปที่ 2.8 | แสดงบล็อกของวงจร เอ็ดจ์ชิ่งขึ้นบันไดหรือตามรอย | 8 |
| รูปที่ 2.9 | แสดงรูปคลื่นของวงจร เอ็ดจ์ชิ่งขึ้นบันได | 9 |
| รูปที่ 2.10 | แสดงรูปคลื่นของวงจรเอ็ดจ์ชิ่งตามรอย | 9 |
| รูปที่ 2.11 | แสดงบล็อกของวงจร เอ็ดจ์ชิ่งที่ใช้ในการประมาณค่าโดยลำดับ | 10 |
| รูปที่ 2.12 | แสดงรูปคลื่น | 11 |
| รูปที่ 2.13 | แสดงหน้าต่างออปเจ็กอินสเปกเตอร์ (Object Inspector) | 13 |
| รูปที่ 2.14 | แสดงหน้าต่างซอร์สโค้ด (Source Code) | 14 |
| รูปที่ 2.15 | แสดงหน้าต่างการบันทึกโครงการ (Save Project) | 15 |
| รูปที่ 2.16 | ไฟล์เวิร์กชาร์ท ขั้นตอนการพัฒนาโปรแกรมภาษา C51 บนไมโครคอนโทรลเลอร์ | 19 |
| รูปที่ 2.17 | แสดงโครงสร้างหน่วยความจำในตระกูล MCS51 | 22 |
| รูปที่ 3.1 | แสดงการจัดขาของไอซี ADC/DAC ขนาด 8 บิต PCF8591 | 23 |
| รูปที่ 3.2 | แอดเดรสไบต์ของ PCF8591 | 25 |
| รูปที่ 3.3 | รายละเอียดไบต์ควบคุมของ PVF8591 | 25 |
| รูปที่ 3.4 | แสดงวงจรบนบอร์ด ISP-1000 | 29 |
| รูปที่ 4.1 | วงจรขยายสัญญาณ 625 เท่า | 30 |
| รูปที่ 4.2 | หน้าต่างแสดงผลแบบยังไม่ได้ทดลอง | 31 |
| รูปที่ 4.3 | การทดลอง โดยการป้อนสัญญาณ 1mV เข้าช่องที่ 1 | 32 |
| รูปที่ 4.3 | การทดลอง โดยการป้อนสัญญาณ 2mV เข้าช่องที่ 1 | 34 |
| รูปที่ 4.3 | การทดลอง โดยการป้อนสัญญาณ 3mV เข้าช่องที่ 1 | 36 |
| รูปที่ 4.3 | การทดลอง โดยการป้อนสัญญาณ 4mV เข้าช่องที่ 1 | 38 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

โครงการ Industrial Data Logger เป็นโครงการที่เป็นความร่วมมือระหว่างภาควิชาวิศวกรรมระบบควบคุม และภาควิชาวิศวกรรมอาหาร เนื่องจากเครื่องอบอาหารที่ใช้งานอยู่ในภาควิชาวิศวกรรมอาหาร ในการทำการทดลองแต่ละครั้งนั้นต้องเสียเวลาอย่างมากในการบันทึกผลการทดลอง โดยใช้เวลาในการบันทึกผล 4-5 ชั่วโมงโดยประมาณ เพราะการเปลี่ยนแปลงความชื้นของวัตถุดิบที่นำไปอบนั้นเป็นไปอย่างช้า ๆ ภาควิชาวิศวกรรมอาหารจึงต้องการให้ข้อมูลนี้สามารถเก็บบันทึกไว้ได้ในคอมพิวเตอร์ เพื่อความสะดวกในการทำการทดลอง และการนำผลการทดลองที่ได้มาวิเคราะห์ต่อไป

ด้วยเหตุนี้ โครงการนี้จึงมุ่งศึกษาถึงการตรวจวัดระดับสัญญาณต่าง ๆ ที่วัดได้จากเครื่องอบอาหาร อันได้แก่ อุณหภูมิ อัตราการไหลของลม ความดัน และน้ำหนักที่เปลี่ยนแปลงไปของวัตถุดิบที่อบ ซึ่งสัญญาณที่ได้จากเครื่องวัดจะเป็นสัญญาณไฟฟ้าที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง (Analog Signal) โดยคณะผู้จัดทำต้องนำสัญญาณเหล่านั้น ไปแปลงผันเป็นสัญญาณดิจิทัล (Digital signal) โดยการใช้เทหุติคอนเวอร์เตอร์ (A/D Converter) เพราะว่าสัญญาณดิจิทัลเป็นสัญญาณมาตรฐานที่สามารถส่งไปยังคอมพิวเตอร์เพื่อทำการประมวลผล และบันทึกผลที่ได้ ซึ่งกลุ่มผู้จัดทำได้เลือกใช้โปรแกรม Delphi 5.0 มาช่วยในการประมวลผล ซึ่งจะมีความสะดวกในการพัฒนาโปรแกรมต่อไปในอนาคต

อนึ่ง เครื่องอบอาหารของภาควิชาวิศวกรรมอาหาร ยังไม่มีเครื่องมือวัดที่สามารถให้สัญญาณไฟฟ้ามาตรฐานได้ เช่น โวลตเซลล์ ที่ใช้วัดความเปลี่ยนแปลงของน้ำหนัก ดังนั้นอาจารย์ที่ปรึกษาของกลุ่มผู้จัดทำ และภาควิชาวิศวกรรมอาหาร จึงตกลงร่วมกันว่าจะทำการจำลองสัญญาณมาตรฐานที่ได้จากเครื่องมือวัดขึ้นมาแทน โดยที่ภายหลังเมื่อภาควิชาวิศวกรรมอาหารได้จัดซื้อเครื่องมือวัดต่าง ๆ มาติดตั้งในเครื่องอบอาหารแล้ว ก็สามารถนำเอาเครื่องมือบันทึกทางอุตสาหกรรม (Industrial Data Logger) ไปใช้งานได้ต่อไป

บทที่ 2

ทฤษฎีและหลักการ

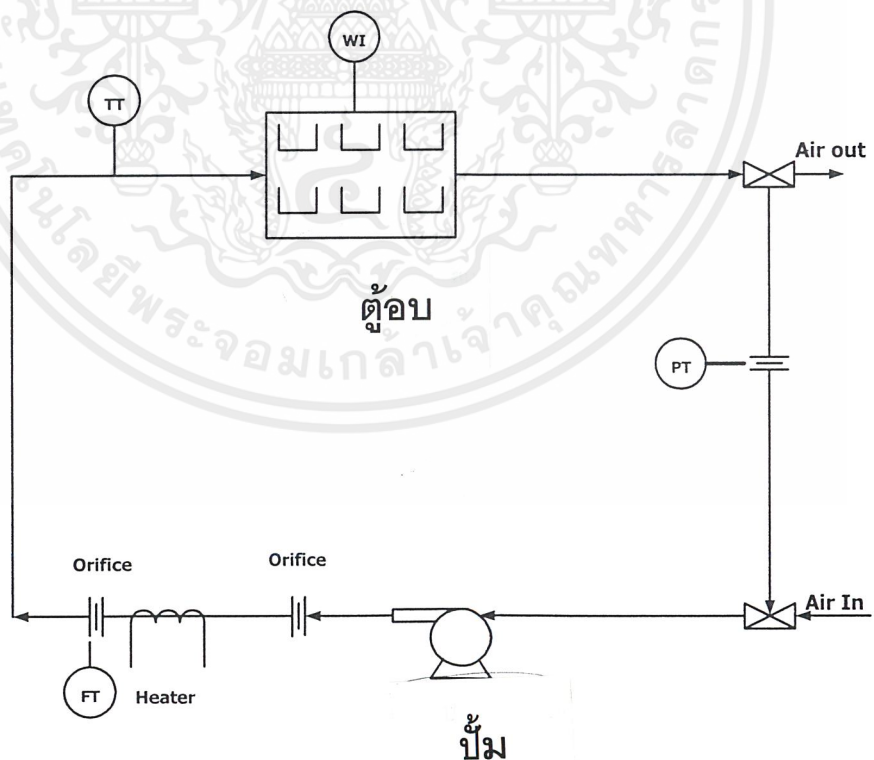
2.1 หลักการของเครื่องบันทึกบันทึกข้อมูล (Data Logger)

2.1.1 ภาพรวมกระบวนการ

ภาพรวมโครงการ

กระบวนการที่เราทำการศึกษาอยู่ เป็นกระบวนการอบแห้งอาหารโดยใช้ความสัมพันธ์ระหว่างความชื้นกับเวลา โดยความชื้นขึ้นกับเวลา โดยความชื้นนั้นเราใช้น้ำหนักในการคำนวณหา

เริ่มแรก ป้อนลมที่ทำให้ร้อนโดยใช้เครื่องทำความร้อน (Heater) แล้วลมร้อนก็ผ่านกระบวนการอบแห้งเป็นการถ่ายเทความร้อนให้กับอาหาร เมื่อถ่ายเทความร้อนแล้วลมที่ถูกทำให้ร้อนบางส่วนจะถูกระบายออกทิ้ง เป็นการนำความชื้นที่ติดมาด้วยบางส่วนทิ้งออกไป โดยนำลมที่เหลือมาผ่านกระบวนการเหมือนเดิมแต่บวกกับลมแห้งจากภายนอกเพิ่มเติม จากนั้นเราก็วัดสัญญาณที่ Sensor แต่ละจุดอยู่ ปรับสัญญาณให้เป็นสัญญาณมาตรฐาน 0-5 Volt เข้าสู่ A/D แล้วเข้าสู่ MCS-51 แล้วนำสัญญาณเก็บเข้าใน RAM แสดงผลใน LCD หากนำเข้าสู่ คอมพิวเตอร์ ก็ผ่านการส่งผ่านแบบอนุกรม

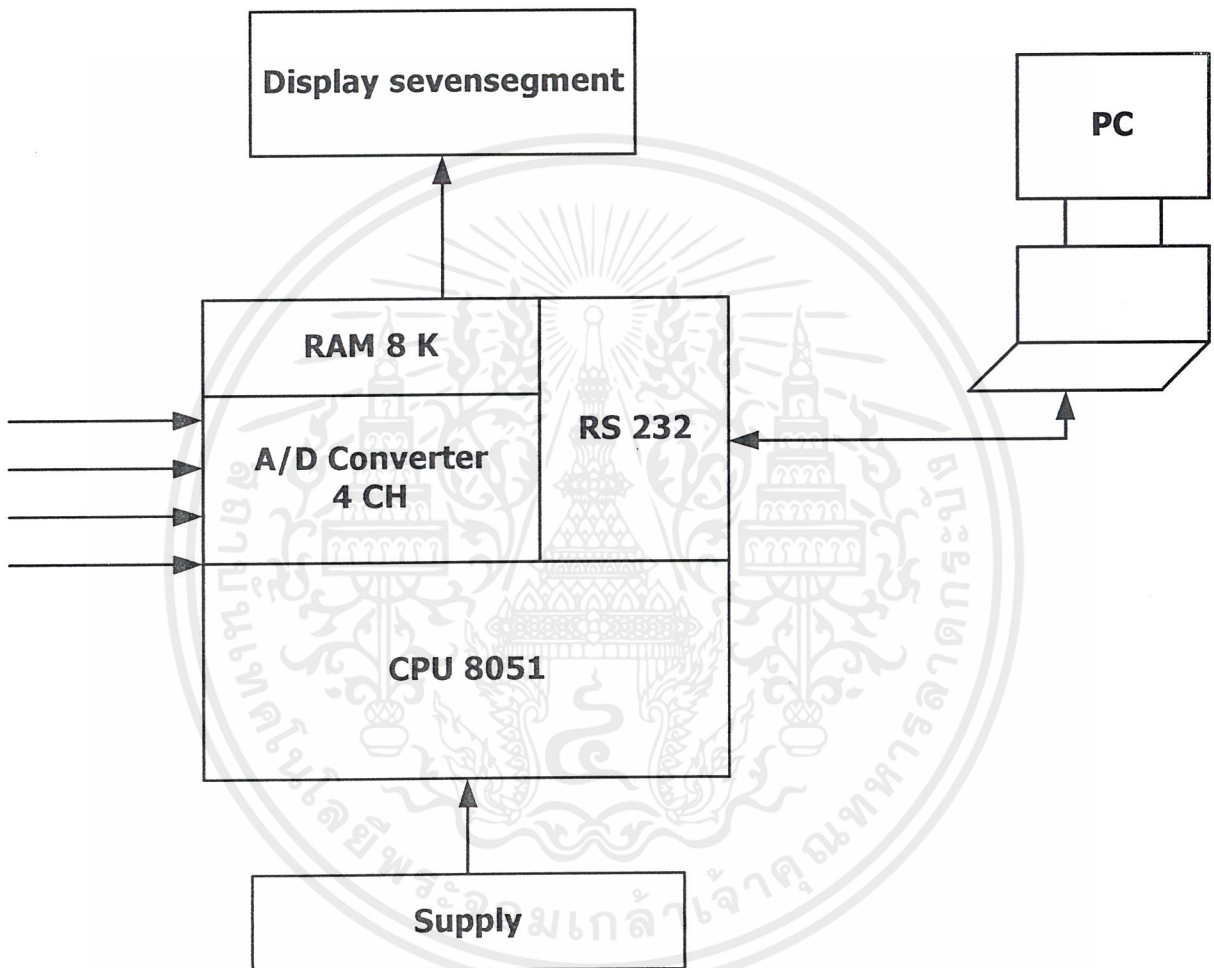


รูปที่ 2.1 แสดงกระบวนการอบแห้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดช่องสัญญาณ

เครื่องบันทึกทางอุตสาหกรรม (Data Logger) จากโครงการนี้แบ่งออกเป็น 4 ช่องสัญญาณ โดยใช้สัญญาณมาตรฐาน 0-5 Volt



รูปที่ 2.2 บล็อกไดอะแกรมของเครื่องบันทึกข้อมูล

- ช่องสัญญาณที่ 1 เป็นช่องสัญญาณที่ใช้วัดอุณหภูมิ (TRD)
- ช่องสัญญาณที่ 2 เป็นช่องสัญญาณที่ใช้วัดน้ำหนัก
- ช่องสัญญาณที่ 3 เป็นช่องสัญญาณที่ใช้วัดความเร็วลม (โดยใช้ DP cell)
- ช่องสัญญาณที่ 4 เป็นช่องสัญญาณวัดความดันลม (โดยใช้ DP cell)

2.1.2 ส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอล

วงจร เอดี ที่ใช้ในโครงการนี้จะใช้ไอซีเบอร์ PCF 8591 ทำการเปลี่ยนสัญญาณอนาลอกที่เข้ามาเป็นดิจิตอล โดยให้การประมาณค่าแบบต่อเนื่อง (Successive Approximation) มี 4 ช่องทางในตัวและมีความคล่องตัวในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ส่วนอินเตอร์เฟซและคอนโทรล

เป็นส่วนที่ทำหน้าที่ควบคุมการจัดระบบ ซึ่งใช้ไอซีไมโครคอนโทรลเลอร์ เบอร์ 89C51RD+ ในตระกูล MCS-51 ซึ่งเป็นหัวใจหลักในการทำงาน ไม่ว่าจะเป็นตัวควบคุมกระบวนการแปลงสัญญาณอนาลอกที่เข้ามาเป็นดิจิทัล การสวิตช์เพื่อเลือกสัญญาณแต่ละช่องสัญญาณการเชื่อมต่อชุด เอ/ดี คอนเวอร์เตอร์ โดยใช้ ไอซี MAX-232 เปลี่ยนระดับสัญญาณข้อมูลดิจิทัล(Digital Data)ไปเป็นระดับสัญญาณดิจิทัล(Digital signal) ตามมาตรฐาน RS -232

2.1.4 ส่วน PC Computer

ในการที่computerจะติดต่อกับอุปกรณ์ภายนอกได้นั้น จำเป็นต้องมีสื่อเข้ามาติดต่อ นั่นคือ โปรแกรมภาษาต่างๆ เช่นภาษาซี ภาษาปาสคาล ภาษาเบสิก เดลไฟล เป็นต้นแต่ในโครงงานนี้กลุ่มผู้จัดทำได้เลือกใช้โปรแกรม เดลไฟล(Delphi 5.0)ซึ่งเป็นโปรแกรมที่สามารถทำงานบนวินโดวส์ 98(windows 98) เป็นโปรแกรมขนาด 32 บิตโดยรหัสภาษา(Source code) ของภาษาเป็นโปรแกรมภาษาปาสคาล ตัวโปรแกรมมีความยืดหยุ่นในการออกแบบแอปพลิเคชัน(Application)ต่างๆโดยจะนำข้อมูลที่ได้จากชุดตรวจวัดมาทำการวิเคราะห์ แสดงผลและบันทึกเก็บไว้

2.2 พื้นฐานการสื่อสารแบบอนุกรม

การส่งข้อมูลจะแบ่งการส่งข้อมูลออกเป็น 2 ลักษณะ คือ การส่งแบบขนาน(Parallel)และการส่งแบบอนุกรม (Serial) การส่งข้อมูลแบบขนานเหมาะสำหรับการส่งที่ระยะทางไม่ไกลนัก เช่นการเชื่อมต่อระหว่างคอมพิวเตอร์กับเครื่องพิมพ์ แต่ถ้าเป็นการส่งข้อมูลในระยะทางไกลๆการส่งข้อมูลแบบขนานจะส่งข้อมูลแต่ละบิตต่อ 1 สายนำสัญญาณ เช่นเมื่อต้องการข้อมูลขนาด 8 บิตเราต้องเราต้องใช้สายนำสัญญาณจำนวน 8 เส้น จะเป็นการสิ้นเปลืองสายส่ง และเป็นการทำให้ค่าใช้จ่ายเพิ่มขึ้น นอกจากนี้สายส่งแต่ละเส้นพื้นที่หน้าตัดสายส่งแต่ละเส้นอาจไม่เท่ากันตลอดทุกเส้น เป็นผลทำให้ค่าความเหนี่ยวนำ(L)และค่าความจุไฟฟ้า (C) ที่อยู่ภายในสายส่งไม่เท่ากันเป็นผลทำให้การส่งข้อมูลอาจผิดพลาดได้ ด้วยเหตุผลนี้ การส่งข้อมูลแบบอนุกรม จึงถูกนำมาใช้งานแทน

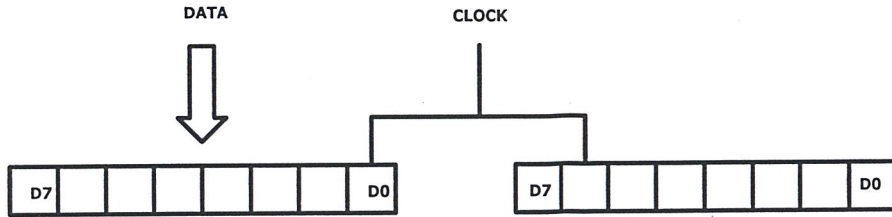
หลักการของการส่งข้อมูลแบบอนุกรม คือ นำข้อมูลแบบขนานไปเก็บไว้ในรีจิสเตอร์เลื่อนข้อมูล(Shift Register) แล้วทำการเลื่อน(Shift)ข้อมูลทีละบิตโดยอาศัยสัญญาณนาฬิกา(clock)

ปัญหาที่สำคัญอย่างหนึ่งของการส่งสัญญาณแบบอนุกรม คือ เราไม่สามารถทำการกำหนดสัญญาณนาฬิกาของตัวส่งและตัวรับให้เท่ากันได้ ในระยะการส่งไกลๆ

การส่งข้อมูลแบบอนุกรมนี้ ข้อมูลจากจุดส่งจะถูกส่งครั้งละ 1 บิต ไปยังจุดรับ ณ.จุดรับนี้ ต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาครั้งละ 1 บิตให้เป็นข้อมูลแบบขนานต่อไป การที่จะทำให้ตัวส่งและตัวรับข้อมูลเข้าใจในส่วนต่างๆของข้อมูลเหมือนกันเพื่อป้องกันความผิดพลาดในการรับส่งข้อมูล กลไกที่เวลานี้แบ่งออกเป็น 2 แบบ คือ ซิงโครนัส(Synchronous)และอะซิงโครนัส(Asynchronous)

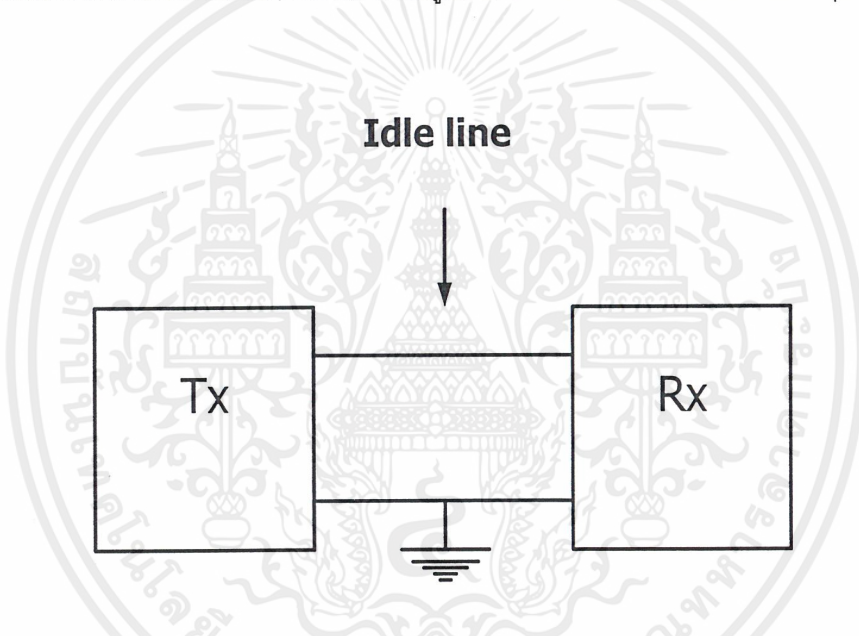
ซิงโครนัส(Synchronous)จะเป็นการส่งข้อมูลที่ใช้สัญญาณนาฬิกาตัวเดียวกันจะทำให้ตัวรับข้อมูลเหมือนกับตัวส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



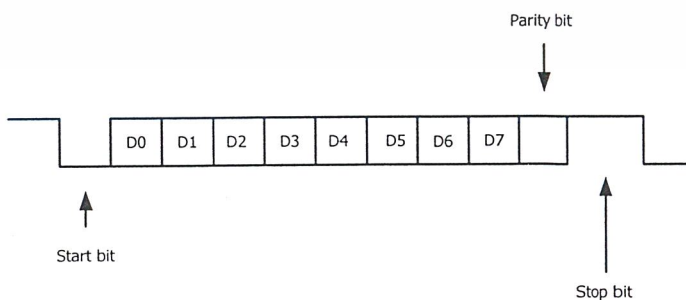
รูปที่ 2.3 แสดงการส่งข้อมูลที่ใช้สัญญาณนาฬิกาตัวเดียวกัน

แต่ในความเป็นจริงในการส่งข้อมูลในระยะไกลเราไม่สามารถทำให้สัญญาณนาฬิกาทั้งตัวรับตัวส่งเท่ากันได้แต่ทำให้ใกล้เคียงกันและสร้างเฟรม(Frame)ของข้อมูลมาเป็นตัวบอกให้ตัวรับทราบส่วนต่างๆของข้อมูล



รูปที่ 2.4 แสดงการเชื่อมต่อระหว่างตัวส่งกับตัวรับ

จากรูป 2.4 ระหว่างตัวรับและตัวส่งจะมีสายสองเส้นเชื่อมต่อกัน โดยเส้นหนึ่งจะเป็นไอดีไลน์ (Idle Line) จะเป็นสายที่มีสัญญาณไฮ (High) เมื่อพร้อมจะรับข้อมูล ส่วนอีกเส้นหนึ่งจะเป็นกราวด์ (ground)



รูปที่ 2.5 แสดงองค์ประกอบของเฟรม

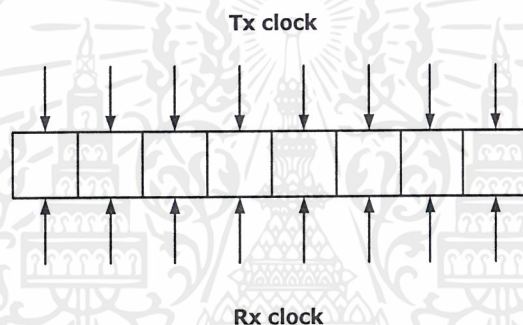
จากรูป 2.5 องค์ประกอบของเฟรม ประกอบด้วยบิตเริ่มต้น (Start Bit) เป็นส่วนที่บอกให้ตัวรับทราบว่าพร้อมที่จะส่งข้อมูล ซึ่งหลังจากบิตเริ่มต้น จะเป็นข้อมูล (Data) ที่ต้องการส่งและส่วนต่อไปเป็น พาริตีบิต (Parity bit) เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.5 องค์ประกอบของเฟรม ประกอบด้วยบิตเริ่มต้น (Start Bit) เป็นส่วนที่บอกให้ตัวรับทราบว่าพร้อมที่จะส่งข้อมูล ซึ่งหลังจากบิตเริ่มต้น จะเป็นข้อมูล(Data) ที่ต้องการส่งและส่วนต่อไปเป็น พาริตีบิต(Parity bit)เป็นส่วนที่ตรวจสอบ และส่วนสุดท้ายเป็นบิตสิ้นสุด (Stop bit)เป็นส่วนที่ทำให้ตัวรับทราบข้อมูลว่าสิ้นสุดเฟรมแล้ว โดยบิตสิ้นสุดจะมีความยาวเป็น 1,1.5,2 บิต

การส่งข้อมูลเริ่มจาก บิตเริ่มต้น ที่มีค่าเป็นโล (Low) เป็นการแจ้งให้ตัวรับทราบว่าจะมีการส่งข้อมูลให้ตัวรับเตรียมพร้อมที่จะรับข้อมูล โดยเริ่มเซตสัญญาณนาฬิกา ให้ใกล้เคียงกับตัวส่งแล้วจึง

เริ่มส่งข้อมูลในบิตต่อไป ในความเป็นจริงบิตแรกที่เข้ามาจะมีสัญญาณนาฬิกาเหมือนกันหรือใกล้เคียงกันมาก แต่ในบิตต่อไปจะเกิดการเลื่อนเฟส เนื่องจากตัวรับอาจใช้สัญญาณนาฬิกา ใกล้เคียงที่ช้ากว่าหรือเร็วกว่า แต่ไม่เกินขอบเขตของข้อมูลแต่ละบิตจนไปถึงค่า ๆ หนึ่ง การเลื่อน (Shift) จะเกินขอบเขตของข้อมูลได้ถูกต้อง ก็จะทำให้การหยุดส่งข้อมูล แล้วเริ่มส่งข้อมูลใหม่ ซึ่งเรียกวิธีนี้ว่า " รีซิงโครไนซ์ " (Resynchronize)



รูปที่ 2.6 แสดงการเลื่อนเฟสของสัญญาณนาฬิกา

อุปกรณ์ที่ทำหน้าที่ในแบบอะซิงโครนัส เรียกว่า UART (Universal Asynchronous Receive) เช่น Z80, 8250, 8251, 8450

2.3 พื้นฐานการทำงานของ เอ/ดี คอนเวอร์เตอร์

กระบวนการต่าง ๆ ที่เกิดขึ้นตามธรรมชาติส่วนใหญ่หากนำมาแปรค่าเป็นสัญญาณทางไฟฟ้ามักเป็นสัญญาณที่อยู่ในรูปของแรงดันหรือกระแส หรือมักเป็นลักษณะของค่าความต้านทานลักษณะที่ได้จะเป็นสัญญาณแอนะล็อก ซึ่งไม่สามารถนำไปใช้กับคอมพิวเตอร์โดยตรงได้ จึงจำเป็นต้องมีวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล เราเรียกว่าวงจร " เอ/ดี คอนเวอร์เตอร์ "

2.3.1 หลักการเบื้องต้นของการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล

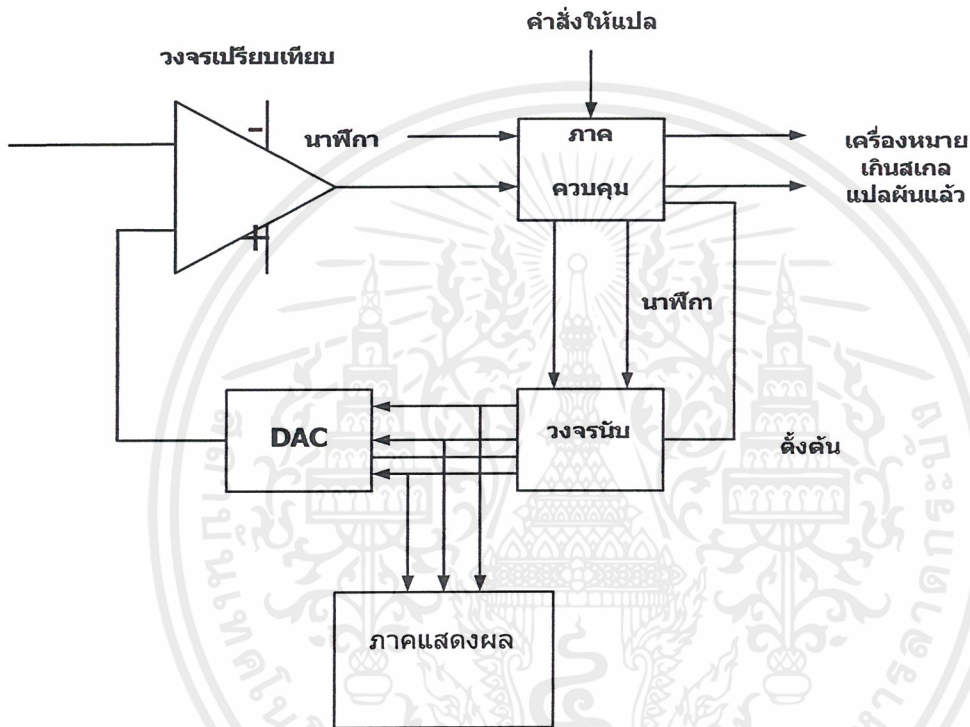
หากนำเอา เอ/ดี ขนาด 3 บิต มาเขียนกราฟคุณสมบัติระหว่างสัญญาณอินพุตกับเอาต์พุต สมมติว่าแรงดันอินพุต V_i เปลี่ยนค่าจาก 0 – 7 โวลต์ และได้สัญญาณเอาต์พุตที่เป็นสัญญาณดิจิทัลจาก 000 – 111 ดังแสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอดีซี ชนิดป้อนกลับขนาน

หลักการคือ การใช้วงจร ดีเอซี ในระบบป้อนกลับ วงจรเอดีซีที่เป็นวงจรประมวลผลสำเร็จรูปส่วนใหญ่จะ
ใช้หลักการนี้ เอดีซี ชนิดป้อนกลับขนานยังแบ่งออกเป็น

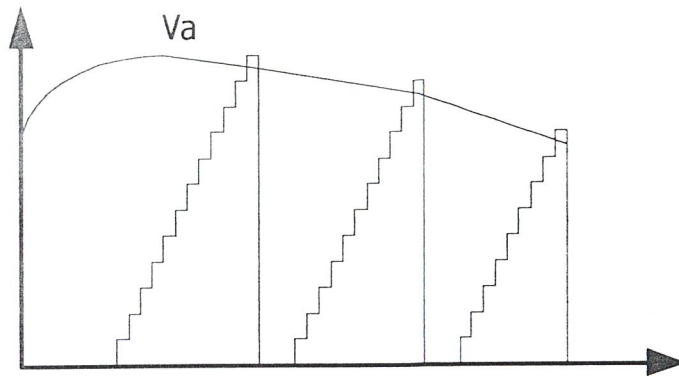
- เอดีซี ชั้นบันได
- เอดีซี ตามรอย (tracking)
- เอดีซี ที่ใช้การประมาณค่าโดยลำดับ (successive approximation)



รูปที่ 2.8 แสดงบล็อกของวงจร เอดีซี ชั้นบันไดหรือตามรอย

รูปที่ 2.8 แสดงแผนภาพของวงจร เอดีซี ชั้นบันไดหรือตามรอย แล้วแต่ว่าวงจรนับเป็นแบบรวมดาหรือแบบนับขึ้นลง

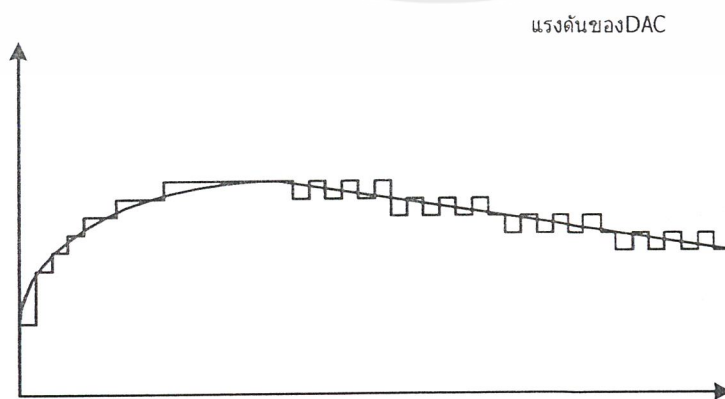
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงรูปคลื่นของวงจร เอดีซี ขึ้นบันได

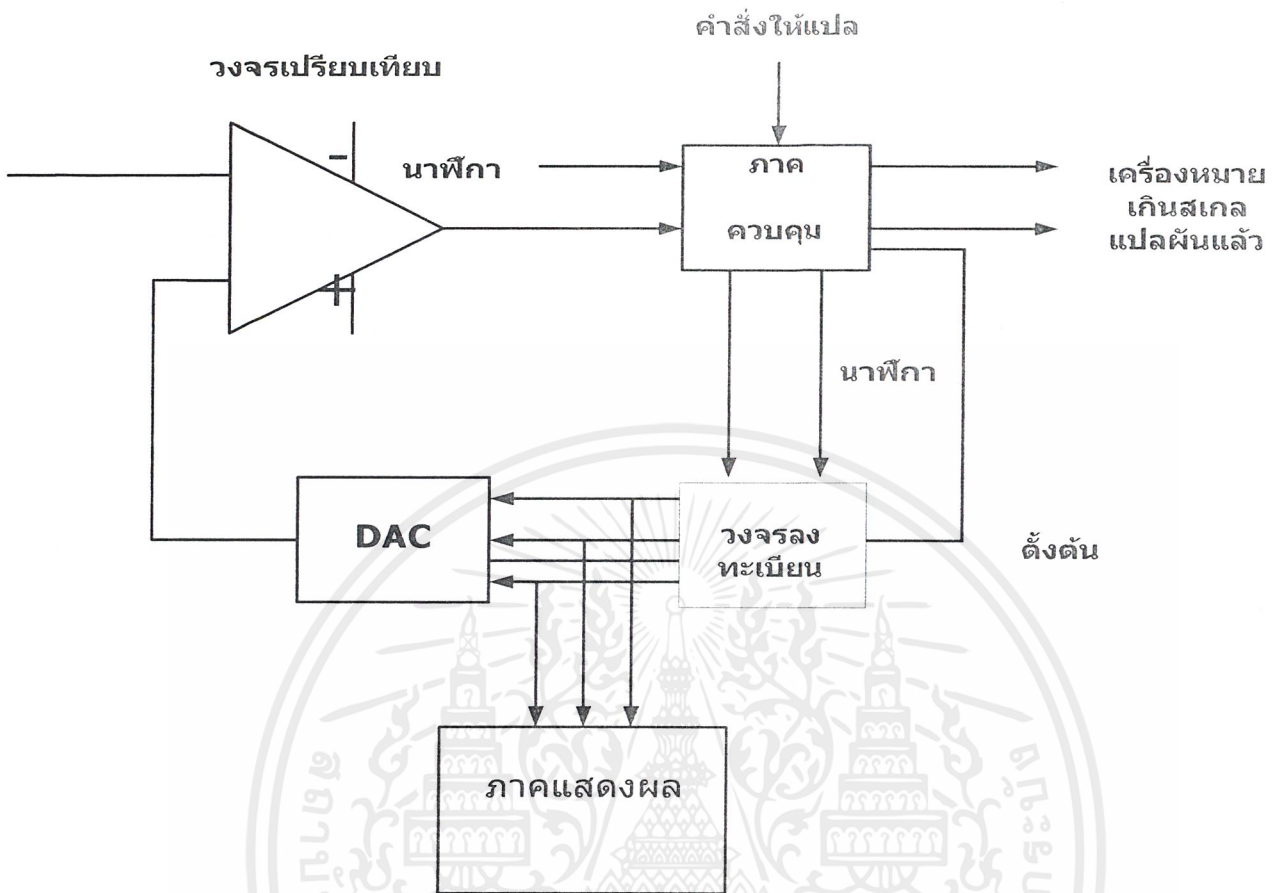
ในกรณีของวงจร เอดีซี ขึ้นบันได เมื่อมีคำสั่งให้แปลงผังวงจรนับจะตั้งต้นใหม่ (Reset) สัญญาณนาฬิกาจะทำให้วงจรนับ นับขึ้นไปเรื่อย ๆ ผลก็คือ แรงดันออกของดีเอซี จะเป็นรูปขึ้นบันได (ดูรูปที่ 2.7) จนกระทั่งขึ้นสูงกว่า V_a วงจรเปรียบเทียบจะบอกให้วงจรควบคุมหยุดสัญญาณนาฬิกาที่ส่งให้วงจรนับจนกว่าจะมีคำสั่งให้แปลงผังใหม่ หรือไม่วงจรเปรียบเทียบจะบอกให้วงจรควบคุมตั้งต้นวงจรนับใหม่ (กรณีรูปที่ 2.7) และวงจรจะเริ่มต้นกระบวนการแปลงผังใหม่ทันที

ถ้าเราใช้วงจรนับขึ้น - ลง และให้วงจรเปรียบเทียบทำหน้าที่ตัดสินใจว่าจะนับขึ้นหรือลงถ้าแรงดันออกของ ดีเอซี ต่ำกว่า V_a ให้นับขึ้น แรงดันออกของ ดีเอซี จะเพิ่มขึ้นตามรอย V_a ไป เมื่อแรงดันออกของดีเอซีสูงกว่า V_a จะนับลง และแรงดันออกของดีเอซีจะลดลงเช่นกัน สังเกตว่ากรณีที่ V_a มีค่าคงตัวหรือเกือบคงตัว แรงดันออกของ ดีเอซี จะขึ้นลงทุก ๆ จังหวะของสัญญาณนาฬิกา ทำให้เกิดความไม่แน่นอนในบิตต่ำสุดของสัญญาณเชิงเลข (digital signal) ขาออก (ดูรูป 2.8) อย่างไรก็ตามวงจรเอดีซี ตามรอยมีข้อบ่งชี้คือ เวลาการแปลงผัง (conversion time) จะสั้นและสัญญาณเชิงเลขที่ได้จะแทนรูปคลื่นของสัญญาณเชิงอุปมาน (analog signal) ได้ดีกว่า



รูปที่ 2.10 แสดงรูปคลื่นของวงจรเอดีซีตามรอย

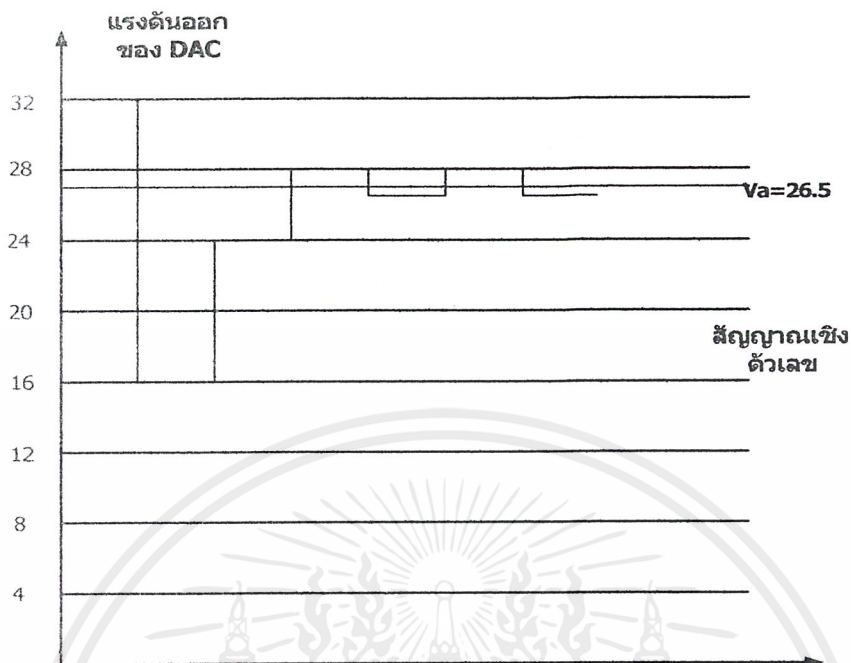
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงบล็อกของวงจร เอดีซี ที่ใช้ในการประมาณค่าโดยลำดับ

รูปที่ 2.10 แสดงแผนภาพบล็อกของวงจรเอดีซีที่ใช้การประมาณค่าโดยลำดับ จะสังเกตว่าเราใช้วงจรลงทะเบียน (Register) แทนวงจรนับ การควบคุมการทำงานของวงจรมีขั้นตอนที่ยู่งยากกว่า เอดีซี ตามรอย ขั้นตอนทำงานนี้แสดงอยู่ในภาพในรูปที่ 2.9 ตัวอย่างเช่น เราต้องการแปลงผัน $V_a = 26.5$ โวลต์เป็นสัญญาณเชิงเลข 6 บิต สมมติว่าในสำคัญต่ำสุด (LSB) สมัยกับ 1 ขวลิต ดังนั้นเลข 100000 จะสมัยกับ 32 ขวลิต ซึ่งเทียบแล้วใหญ่กว่า V_a ดังนั้น บิตในสำคัญสูงสุดเท่ากับศูนย์ (MSB =0) ให้บิตมีนัยสำคัญรองลงไปเป็น "1" เลข 010000 จะสมัยกับ 16 ขวลิต ซึ่งเล็กกว่า V_a ให้บิตนี้ไว้เป็น 1 และให้บิตรองลงไปเป็น 1 ด้วยเลข 011000 สมัยกับ 24 ขวลิต กับ V_a และเช่นนี้เรื่อย ๆ ไปจนครบทุกบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงรูปคลื่น

2.4 โครงสร้างโปรแกรมของเดลไฟ (The Structure of a Delphi Program)

การเขียนโปรแกรมเดลไฟให้ได้ประโยชน์มากขึ้น ถ้าพัฒนาโปรแกรมเข้าใจโครงสร้างของ Object Pascal ซึ่งเป็นรากฐานของเดลไฟ

IDE เป็นเครื่องมือที่ช่วยให้สามารถเขียนโปรแกรมได้เร็วขึ้น โดยที่ IDE ช่วยเขียนคำสั่งที่จำเป็นต้องใช้ และช่วยให้นักพัฒนาสามารถออกแบบจอภาพและเขียนคำสั่งที่ต้องการให้ทำงานเมื่อมีเหตุการณ์ (event) เกิดขึ้น

IDE จะช่วยเขียนโค้ดคำสั่งงานซึ่งจะอยู่ภายใต้กฎเกณฑ์ของ Object Pascal และช่วยสร้าง resource ที่ต้องใช้ตามกฎเกณฑ์ทำงานของระบบปฏิบัติการแบบ Windows

ในหัวข้อนี้จะได้ศึกษาถึงโครงงานของโปรแกรมและส่วนของคำสั่งงานในโปรแกรมที่จำเป็นต้องใช้

2.4.1 โพรซีเจอร์ (Procedure)

โพรซีเจอร์ เป็นกลุ่มคำสั่งงานที่ใช้การทำงานอย่างใดอย่างหนึ่ง อาจจะเรียกอีกอย่างว่า โปรแกรมย่อย เช่น การคำนวณภาษี การแสดงจอภาพเพื่อรับจากผู้ใช้งาน เป็นต้น ตัวอย่างของโพรซีเจอร์

1. procedure Button 1 Click ;
2. begin
3. Edit 1.text := 'Hello World';
4. End;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโพธิ์เตอร์ตัวอย่างนี้จะแสดงข้อความ Hello World ในคอนโทรล Edit 1 เมื่อผู้ใช้งานกดปุ่ม Bottom 1 โดยที่ชื่อโพธิ์เตอร์จะอยู่บรรทัดที่ 1 และมีคำสั่ง begin และ end คลุมคำสั่งที่ใช้ในการทำงานของโพธิ์เตอร์

- คำสั่งที่อยู่ในโพธิ์เตอร์อาจมีได้ตั้งแต่คำสั่งเดียวจนถึงหลาย ๆ หน้ากระดาษ
- เนื่องจากภาษา Object Pascal เป็นภาษาที่ Case - Insensitive ดังนั้น การตั้งชื่อต่าง ๆ ที่ใช้อักขระตัวใหญ่และอักขระตัวเล็กจะไม่แตกต่างกันเช่น Edit 1 หรือ edit 1 จะเป็นการอ้างชื่อเดียวกัน
- เพื่อความสะดวกในการอ่านโปรแกรมและความง่ายในการแก้ไขโปรแกรม ควรจะใช้อักขระตัวใหญ่กับตัวเล็ก

2.4.2 การสร้างโพธิ์เตอร์

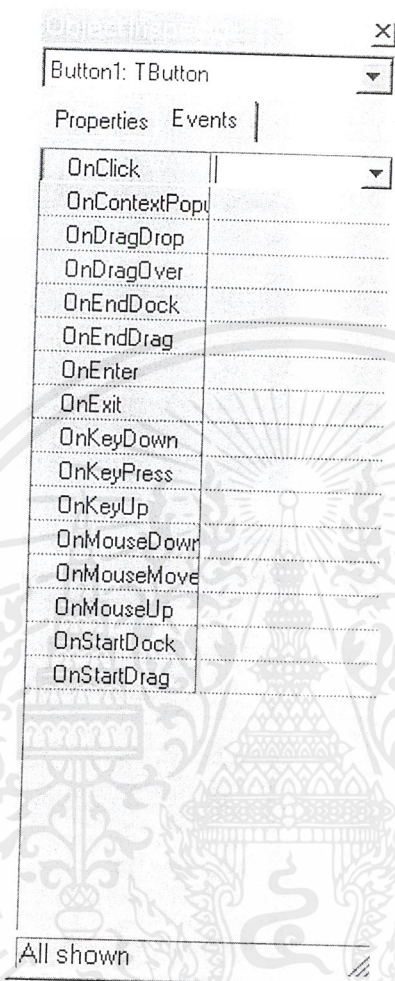
ปกติจะสร้างโดยใช้ IDE อย่างไรก็ตามเราสามารถสร้างขึ้นเองได้ถ้ามีความเข้าใจโครงสร้างของ Object Pascal อย่างดี

ตัวอย่างการสร้างโพธิ์เตอร์ใช้ IDE ให้ทำดังนี้

1. ใช้เมนู File/New Application เพื่อสร้างโปรเจ็คใหม่
2. เพิ่มคอนโทรล Edit และ Button 1
3. สร้างโพธิ์เตอร์เพื่อทำงานตามเหตุการณ์ ซึ่งมี 2 วิธีดังนี้
 - ดับเบิลคลิกที่คอนโทรลใดบนฟอร์ม IDE จะสร้างโพธิ์เตอร์เพื่อทำงานตาม event ที่ต้องการ วิธีนี้จะได้ default event สำหรับคอนโทรลนั้น
 - หรือ
 - คลิกที่คอนโทรลเพื่อให้คอนโทรลนั้นได้โฟกัส แล้วดูที่ Object Inspector ที่ Event Page จากนั้นดับเบิลคลิกที่ Event ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างเราต้องสร้างโปรซีเจอร์สำหรับ Button 1 เมื่อผู้ใช้คลิกที่ปุ่ม ดังนั้น ให้คลิกที่ Button 1 ให้ได้โฟกัสก่อนแล้วเลื่อนเมาส์ไปที่ Object Inspector คลิกที่ Event Page ซึ่งจะปรากฏภาพดังรูป

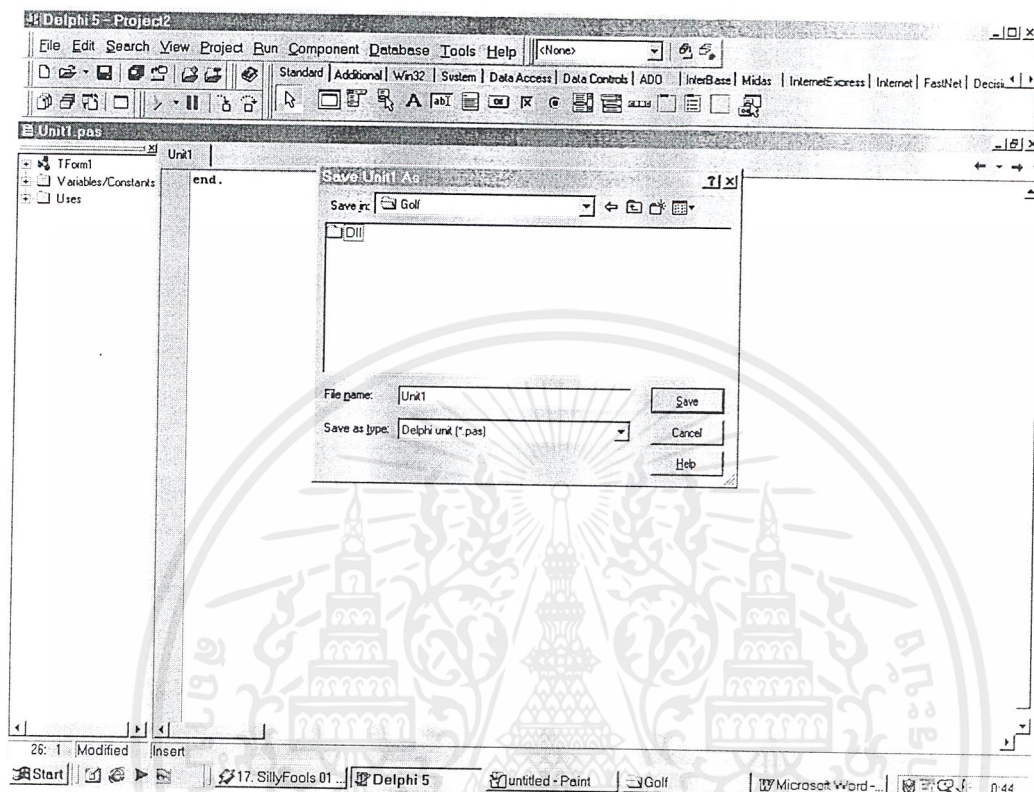


รูปที่ 2.13 แสดงหน้าต่างออปเจ็กอินสเปกเตอร์ (Object Inspector)

4. ดับเบิลคลิกบรรทัด OnClick จะทำให้ IDE การสร้างโค้ดคำสั่งพร้อมทั้งเปิด Editor Windows ขึ้นมาดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit 1.Text = 'Hello World'; การสลับหน้าต่างของเอดิเตอร์กับฟอร์มให้ใช้ปุ่ม
ใช้เมนู File/Save Project แล้วสร้างไคเร็คทอรีย่อย Control 1 ดังรูปที่ 2.14



รูปที่ 2.15 แสดงหน้าต่างการบันทึกโครงการ (Save Project)

ทำการตั้งชื่อยูนิต “ Main.Pas ” และชื่อโปรเจ็ค “ Control1.DPR ” เพื่อเก็บฟอร์มและโปรเจ็คไว้ใน
ไคเร็คทอรีที่สร้างขึ้นใหม่

การเก็บโปรแกรมที่เขียน โดยเคล ไฟล์นั้นควรจะเก็บแยกแต่ละ โปรเจ็คไว้ที่คนละไคเร็คทอรีกัน
และฟอร์มที่ถูกกำหนดให้เปิดขึ้นเมื่อเริ่มรัน โปรแกรมควรจะตั้งชื่อว่า Main
จากตัวอย่างจะได้ขอสไลด์ของ โปรเจ็คและยูนิตของฟอร์มดังนี้

ขอสไลด์ของโปรเจ็ค Control1.DPR

Program control;

Uses

Forms,

Unit1 in 'Unit1.pas' {Form1};

{SR *.RES}

begin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Application.Initialize;
Application.CreateForm(Tform1,Form1);
Application.Run;

End.

```

ซอสโค้ดของยูนิท Main.Pas เป็นดังนี้

```

unit main;
interface
uses
Windows , Messages , SysUtils , Classes , Graphics , Controls , Forms , Dialogs
StdCtrls;
Type
Tform1 = class(Tform)
Button1 : TButton;
Edit : TEdit;
procedure Button1Click (Sender : TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1 : TForm1;
implementation
{SR *.DFM}
procedure TForm1.Button1Click(Sender:TObject);
begin
Edit1.text := 'Hello World';
end;
end.

```

จากซอสโค้ดของโปรเจ็คข้างต้น เมื่อโปรแกรมเริ่มทำงาน คำสั่งแรกที่ทำงานคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application.CreateForm(TForm1,Form1);เป็นการสร้างฟอร์ม Form1 เพื่อเตรียมใช้งานต่อไป ส่วนคำสั่งต่อไปจะสั่งให้โปรแกรมเริ่มทำงานคือคำสั่ง Application.Run; โดยที่จะทำให้ Form1 แสดงขึ้นมาบนจอภาพแล้วงานก็จะดำเนินต่อไปโดยขึ้นอยู่กับเหตุการณ์ (Event) ที่เกิดขึ้นบนฟอร์ม Form1 นี้

จากข้อสโค้ดของยูนิท/ฟอร์ม “Main.pas” มีส่วนประกอบสำคัญ 3 ส่วน คือ

| | |
|----------------|--|
| Title | ส่วนที่เป็นชื่อยูนิท |
| Interface | ส่วนที่เป็นการติดต่อกับยูนิท/ฟอร์มอื่น เพื่อกำหนดขอบเขตของข้อมูลและฟังก์ชันที่ยอมให้ใช้ได้โดยใช้หลักการแบบ Object oriented programming |
| Implementation | เป็นส่วนที่ทำงานตามฟังก์ชันที่ได้กำหนดไว้ในส่วน Interface |

นอกจากนี้เราอาจเพิ่มส่วนของ initialization เพื่อกำหนดค่าเริ่มต้นครั้งแรกให้กับฟอร์ม/ยูนิท โดยคำสั่งภายใน initialization จะถูกเรียกทำงานครั้งเดียวเพื่อเริ่มทำงานเท่านั้น

2.4.3 โครงสร้างของยูนิท

จากยูนิท MAIN.PAS ส่วนของอินเตอร์เฟส (Interface) จะเริ่มจากบรรทัดที่ 2-17 ส่วนของ Implement จะเริ่มจากบรรทัดที่ 18 เป็นต้นไป รายละเอียดของส่วน Interface มีดังนี้

| | |
|----------------|---|
| คำสั่ง uses | ใช้สำหรับอ้างอิงถึงยูนิทที่ต้องการนำไปใช้ในขณะทำการคอมไพล์และลิงก์โปรแกรม |
| คำสั่ง type | ใช้สำหรับการประกาศรูปแบบของ Object ชื่อ TForm1 ซึ่งได้นำต้นแบบมาจาก Object ของ TForm โดยที่ TForm เป็น Object ของฟอร์มเปล่า สำหรับคำสั่ง Editt : Tedit; เป็นการกำหนดว่าบนฟอร์มนี้มีคอนโทรลชื่อ Editt ซึ่งได้ต้นแบบมาจาก Object Tedit คำสั่ง Button 1 : Tbutton; เป็นการกำหนดคอนโทรล Button 1 ซึ่งได้ต้นแบบมาจาก Tbutton คำสั่ง procedure Button 1 Click... เป็นการกำหนดเมทอด (method) หรือเป็นโพซีเตอร์ที่ต้องการสั่งให้ทำงาน เมื่อมีการคลิกที่ Button 1 โดยที่ IDE จะทำการเชื่อมส่วนนี้ให้ โดยผ่านทางฟอร์ม (Main.DEF) ซึ่งได้กำหนดไว้โดยใช้ Object Inspector โดยทั่วไปชื่อเมทอดจะถูกตั้งโดยใช้ชื่อคอนโทรลและตามด้วยเหตุการณ์ (event) เช่น Button 1 Click มาจาก Button 1 ซึ่งเป็นคอนโทรลกับ Click ซึ่งเป็นเหตุการณ์ที่เกิดกับ Button 1 อย่างไรก็ตามชื่อเมทอดนี้ นักพัฒนาระบบสามารถเปลี่ยนแปลงได้ |
| คำสั่ง Var | เป็นการกำหนดตัวแปรชื่อ Form1 ซึ่งได้ต้นแบบมาจาก TForm 1 ซึ่งได้กำหนดไว้ข้างต้น |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของส่วน Implementation

```

คำสั่ง
procedure TForm1.Button1Click(Sender : TObject);
begin
Edit1.Text:='Hello World';
End;

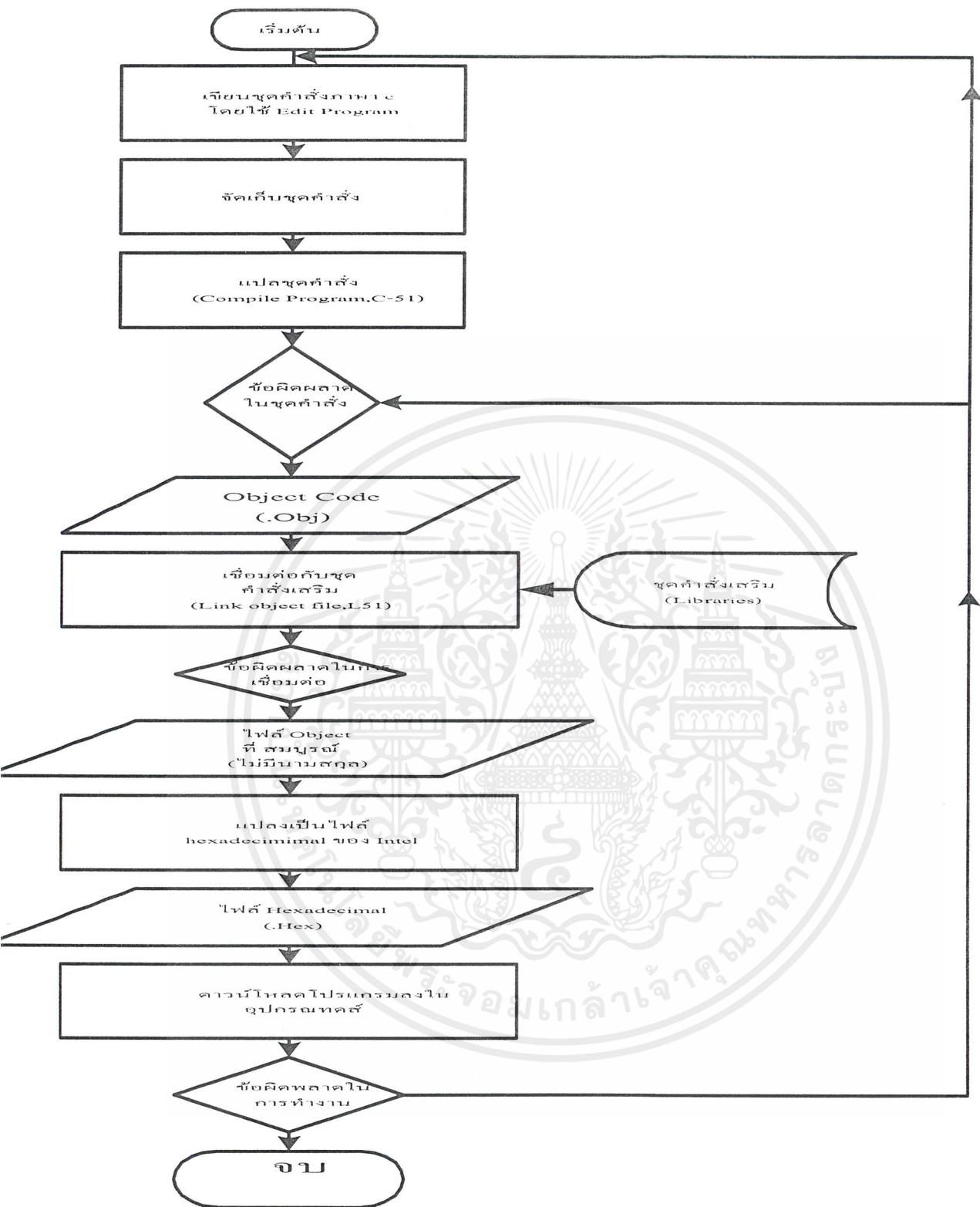
```

เป็นส่วนที่แสดงรายละเอียดการทำงานของเมทอดที่ได้กำหนดไว้ข้างต้น คือ Button1Click ส่วน TForm1 ที่อยู่ด้านหน้าก็จะเป็นการบอกว่า Button1Click เป็นเมทอดของ Object TForm1 สำหรับ(Sender ; TObject) จะเป็น Object ของคอนโทรลที่เป็นผู้ที่สั่งการทำงานนี้ซึ่งในที่นี้คือ Button1 โดยที่ Runtime 1 library (RTL) จะทำการส่ง Object นี้ผ่านทางพารามิเตอร์มาให้กับโพธิ์เตอร์นี้ ส่วนคำสั่งงานที่อยู่ระหว่าง begin....end ก็จะเป็นคำสั่งที่ถูกสั่งให้ทำงานต่อไป ซึ่งก็คือ Edit1.Text:='Hello World'; ก็จะเป็นการกำหนดค่าพริ๊อบเพอร์ตี้ Text ของ Edit1 ให้มีค่าเท่ากับ 'Hello World' ซึ่งจะมีผลทำให้แสดงค่า Hello World ออกมา

ตัวอย่างการส่งค่าผ่านพารามิเตอร์

2.5 ขั้นตอนในการพัฒนาโปรแกรมภาษา C-51

ขั้นตอนในการพัฒนาโปรแกรมภาษา C บน MCS-51 นั้นก็มีลักษณะคล้ายๆ กับภาษาระดับสูงอื่นๆ ที่มีลักษณะการ แปลภาษาโดยใช้คอมไพเลอร์ สำหรับขั้นตอนในการพัฒนานั้นสามารถเขียนเป็น Flow Chart ได้ดังรูป ที่ 2.16



รูปที่ 2.16 โฟลว์ชาร์ท ขั้นตอนการพัฒนาโปรแกรมภาษา C51 บนไมโครคอนโทรลเลอร์

ขั้นตอนในการเขียนชุดคำสั่งภาษา C : หลังจากที่เรารู้จักวิธีที่จะใช้ในการแก้ปัญหา (ซึ่งก็คือการเขียน Algorithm ซึ่งอาจจะเขียนในรูปของภาษาอังกฤษแบบง่ายๆ หรือ Flow Chart ก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาม) ขั้นตอนถัดมาก็คือการนำสิ่งที่เราคิดเอาไว้แล้วนั้นมาเขียนหรือ Code เป็นโปรแกรมขึ้นมา โดยการเขียนโปรแกรมนั้นก็อาจจะใช้ Editor ตัวใด ๆ ก็ได้ เช่น ในระบบปฏิบัติการ DOS อาจจะใช้ Editor ที่ชื่อ “Edit” หรือ “Qedit” เป็นต้น

ขั้นตอนการเขียนการจัดเก็บชุดคำสั่ง : หลังจากที่ได้อ่านโปรแกรมเสร็จแล้วเราจะต้องทำการจัดเก็บโปรแกรมลงในสื่อบันทึกข้อมูล ซึ่งการจัดเก็บนั้นก็ขึ้นอยู่กับระบบปฏิบัติการที่ใช้ ว่าในการจัดเก็บนั้นจะมีกฎเกณฑ์ในการตั้งชื่อไฟล์ข้อมูล เช่น ถ้าหากเป็นระบบปฏิบัติการ DOS ก็สามารถตั้งชื่อได้ไม่เกิน 8 ตัวอักษร เป็นต้น แต่สำหรับในส่วนที่เป็นนามสกุลนั้นจะต้องเป็น .C เท่านั้นและเราจะเรียกไฟล์ที่ได้ว่า Source Code หรือ Source Program

ขั้นตอนการเขียนแปลชุดคำสั่ง(คอมไพล์) : จาก Source Code ที่เราได้ นั้น เป็นเพียงชุดคำสั่งของภาษา C สิ่งที่เราจะต้องทำถัดมาก็คือ การแปลคำสั่งภาษา C ให้เป็นภาษาที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ ซึ่งเราเรียกว่า Machine Language โดยตัวแปลภาษาจะนำ Source Code ที่เราบันทึกเอาไว้แล้วมาทำการตรวจสอบความถูกต้องของภาษา และนำไปใช้งาน (หรือที่เรียกว่า Syntax Checking) ซึ่งถ้าหากตัวแปลตรวจสอบแล้วพบว่ามีส่วนใดส่วนหนึ่งของโปรแกรมมีการใช้รูปแบบที่ไม่เป็นไปตามกฎข้อบังคับของภาษา C แล้ว ตัวแปลของภาษาก็จะแสดงข้อความชี้แจงความผิดพลาดขึ้น แต่ถ้าหาก Source Program นั้นๆ มีการใช้งานที่ถูกต้องตามกฎของภาษา แล้วตัวแปลภาษาก็จะสร้างชุดคำสั่งขึ้นมาใหม่ สำหรับชุดคำสั่งใหม่ที่ตัวแปลภาษาสร้างขึ้นมาใหม่นี้จะมีการทำงานที่เหมือนกับ Source Program แต่จะแตกต่างกันตรงที่ ชุดคำสั่งชุดใหม่ที่ตัวแปลภาษาสร้างมานั้นจะอยู่ในรูปของภาษาเครื่อง (Machine Language) ซึ่งเป็นภาษาที่เครื่องคอมพิวเตอร์สามารถเข้าใจและปฏิบัติตามคำสั่งได้ โดยชุดคำสั่งที่ตัวแปลภาษาได้สร้างขึ้นมาใหม่นั้น จะใช้ชื่อที่เป็นชื่อเดียวกับ Source Program แต่จะมีการใช้นามสกุลเป็น “.obj” ซึ่งเราอาจจะเรียกชุดคำสั่งที่ได้ว่าเป็น object code และเรียกไฟล์ที่สร้างขึ้นมาใหม่นั้นว่าเป็น object file หรือ compiled file ก็ได้ ตัวอย่างเช่น ถ้าเรามี Source Program ที่ชื่อ test.c แล้ว object file ก็จะมีชื่อว่า test.obj

ขั้นตอนการเชื่อมต่อกับชุดคำสั่งเสริม : ในการเขียนโปรแกรมภาษา C นั้นจะมีลักษณะเป็นการเรียกใช้ฟังก์ชัน (Function call) เป็นหลัก ซึ่งฟังก์ชัน (หรือส่วนของโปรแกรม) นั้นอาจจะถูก Code เอาไว้ใน Source Program เดียวกันกับ Source Program ที่ต้องการนำไปใช้งานหรือไม่ก็ได้ ซึ่งถ้าหากว่า Source Program ของเรามีการเรียกใช้ฟังก์ชันที่อยู่ใน Source Program ตัวอื่น นั่นก็คือว่า object code ที่ได้มานั้น (อย่าลืมว่า object code ได้มาหลังจากการแปล Source Program) ยังขาดชุดคำสั่งที่เกี่ยวข้องอยู่ด้วย (ทั้งนี้เพราะชุดคำสั่งกลุ่มนั้น อยู่ในอีก object code หนึ่ง) ดังนั้นรายละเอียดในขั้นตอนนี้ก็คือ การนำเอา object code ที่เกี่ยวข้องทั้งหมดมาทำการเชื่อมต่อกัน (Linking) เพื่อให้ชุดคำสั่งต่างๆ สามารถอ้างอิง หรือเรียกใช้กันได้นั่นเอง ซึ่งถ้าหากว่าตัวทำหน้าที่ต่อเชื่อม (Linking) ไม่สามารถที่จะหาชุดคำสั่งที่เกี่ยวข้องได้ (ชุดคำสั่งเหล่านั้นจะถูกบรรจุอยู่ใน object file) ตัว Linker ก็แสดงข้อความชี้แจงความผิดพลาด ในการเชื่อมต่อให้ทราบ แต่ถ้าตัว Linker เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเชื่อมต่อกับคำสั่งต่างๆ ได้ครบแล้ว ตัว Linker ก็จะสร้างไฟล์ object ที่สมบูรณ์ขึ้นมาอีก ไฟล์หนึ่ง โดยไฟล์ที่สร้างขึ้นมาใหม่นั้น จะใช้ชื่อเดียวกับ Source Program และ Object File แต่จะไม่มีการใช้นามสกุล เช่น ถ้าหากเรามี Source Program ที่ชื่อ test.c แล้ว object file ที่ได้ ก็จะใช้ชื่อว่า test.obj และได้ Object File ที่สมบูรณ์ชื่อ test เป็นต้น

ขั้นตอนการแปลงไฟล์เป็น Hexadecimal File: เมื่อเราได้ไฟล์ Object ที่สมบูรณ์ แล้วถ้าต้องการนำไฟล์นี้ไปใช้ในการโปรแกรมลงอุปกรณ์ EPROM หรือ EEPROM จะต้องเปลี่ยนเป็น Hexadecimal ไฟล์ในรูปแบบของ บริษัท Intel โดยการเรียกโปรแกรม OHS51 ไฟล์ที่ได้จะมีนามสกุล .hex

ขั้นตอนการดาวน์โหลดอุปกรณ์ทดสอบ : ขั้นตอนสุดท้ายก็คือการนำไฟล์ .hex ดาวน์โหลดลงอุปกรณ์ทดสอบหรือโปรแกรม Simulator เพื่อทดสอบการทำงานของโปรแกรมว่าเป็นไปตามปัญหา (problems) ที่กำหนดเอาไว้ก่อนหน้านี้หรือไม่ ซึ่งถ้าหากพบข้อผิดพลาดในโปรแกรมเราต้องกลับไปแก้ไข Source Program และทำตามขั้นตอนต่างๆ ตามที่ได้กล่าวมาแล้ว อีกครั้งหนึ่งจนกว่าโปรแกรมที่ได้จะทำงานได้อย่างถูกต้อง ตรงตามที่ออกแบบไว้

โครงสร้างของหน่วยความจำในตระกูล MIC51

ในการจัดวางหน่วยความจำของ MSC-51 นั้น จะใช้การจัดวางหน่วยความจำแบบ Harvard ซึ่งจะแยกหน่วยความจำออกมาเป็น ส่วนของโปรแกรมและส่วนของข้อมูล ซึ่งทั้ง 2 ส่วน สามารถที่จะออกแบบให้มีขนาดอยู่ในขอบเขตสูงสุดที่ 64 Kbytes การติดต่อหน่วยความจำโปรแกรม ภายในกับภายนอกนั้น เราสามารถกำหนดให้มีการติดต่อภายนอกโดยการป้อนสัญญาณลอจิก 0 ที่ขาสัญญาณ EA ซึ่งถ้าเป็นภายนอกจะมีสัญญาณจากขา PSEN เป็นตัวติดต่อกับชิพที่ต่อภายนอก

ส่วน โครงสร้างสำหรับหน่วยความจำที่เป็นข้อมูลนั้น โดยทั่วไปของ MSC-51 ภายในจะมีแรมสำหรับใช้งานทั่วไปอยู่แล้วที่ตำแหน่ง 00H-7FH (128 ไบท์ ครอบคลุมตำแหน่งแบบบิท) วิธีการที่จะเข้าถึง จะเป็นแบบ Direct กับ Indirect และถ้าเพิ่มขยายไปอีก 128 ไบท์ สำหรับเบอร์ 8032 หรือ 8052 การเข้าถึงจะเป็นแบบ Indirect ตำแหน่งตั้งแต่ 80H-FFH ที่เพิ่มขยายมานี้โดยปกติจะเป็นตำแหน่งของ SFR (Special Function Register) จะเข้าถึงแบบ Direct โดยหน่วยความจำทั้งสองจะซ้อนกันอยู่เหมือนกระดากที่วางทับกัน

การจัดการหน่วยความจำนั้นตัวคอมพิวเตอร์ จะจัดการวางตัวแปรที่เราประกาศลงในตำแหน่งพื้นที่ที่สัมพันธ์กับชนิดของโมเดลหน่วยความจำที่ระบุ หรือเราสามารถที่จะกำหนดเป็นคำสั่งวงเล็บที่จะใช้ภายในบรรทัดของการประกาศตัวแปร ซึ่งรูปแบบของโมเดลและคำประกาศแบบเจาะจงมีดังนี้

Small ตัวแปรที่ถูกใช้งานจะถูกกำหนดอยู่ใน Internal Data memory ขนาดของแสดกจะถูกกำหนดตามการใช้งานจริงภายในโปรแกรม ทำให้มีความเร็วในการใช้งาน โดยปกติจะถือเป็นค่าที่ตั้งไว้เมื่อไม่มีการกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Compact ตัวแปรที่ถูกใช้งานจะถูกกำหนดอยู่ในพื้นที่หน่วยความจำข้อมูลภายนอก ขนาด 256 ไบต์ (@R0,@R1)

Large ตัวแปรที่ถูกกำหนดให้อยู่ในพื้นที่หน่วยความจำข้อมูลภายนอกทั้ง 64 Kbytes ซึ่งขนาดของตัวโปรแกรมจะมีขนาดใหญ่กว่าทุก โมเดล

การประกาศแบบเจาะจงโดยใช้คำสั่งวน

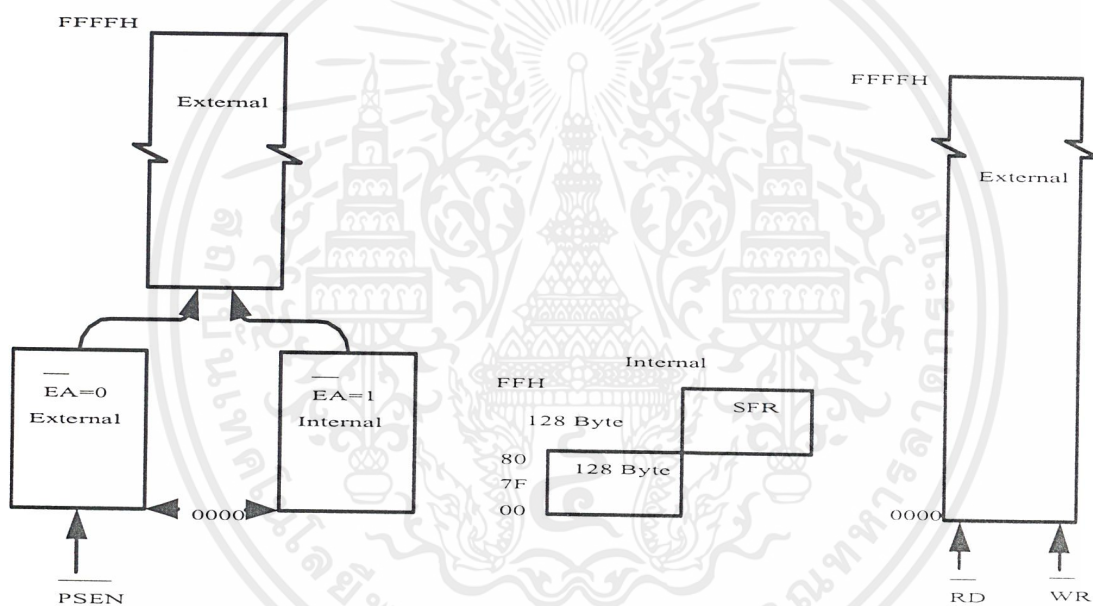
code แทนพื้นที่ 64 Kbytes ที่จัดเก็บในโปรแกรมเมมโมรี่ (move@A+DPTR)

data แทนพื้นที่ในดาต้าเมมโมรี่ 128 ไบต์ต่าง(00H-7FH) เข้าถึงแบบ Direct

idata แทน 256 ไบต์ ของดาต้าเมมโมรี่ภายใน(128 Byte + 128 Byte)

bdata แทนตัวแปรแบบบิตที่พื้นที่ของบิตแอดเดรส 128 บิต (20H-2FH)

xdata แทนตัวแปรในดาต้าเมมโมรี่ภายนอก 64 (movx@DPTR)



รูปที่ 2.17 แสดงโครงสร้างหน่วยความจำในตระกูล MCS51

การกำหนดโมเดลของหน่วยความจำให้กับตัวคอมไพเลอร์นั้นสามารถจะกำหนดให้ตรงกับฮาร์ดแวร์ที่มีการออกแบบไว้อยู่แล้ว โดยชนิดของโมเดลนั้น สามารถใช้ไคเรคทีฟช่วยประกาศในตอนทำการคอมไพล์ หรือจะกำหนดเป็น Preprocessor(#prama) ในตัวโปรแกรมก็ได้

บทที่ 3

การออกแบบ

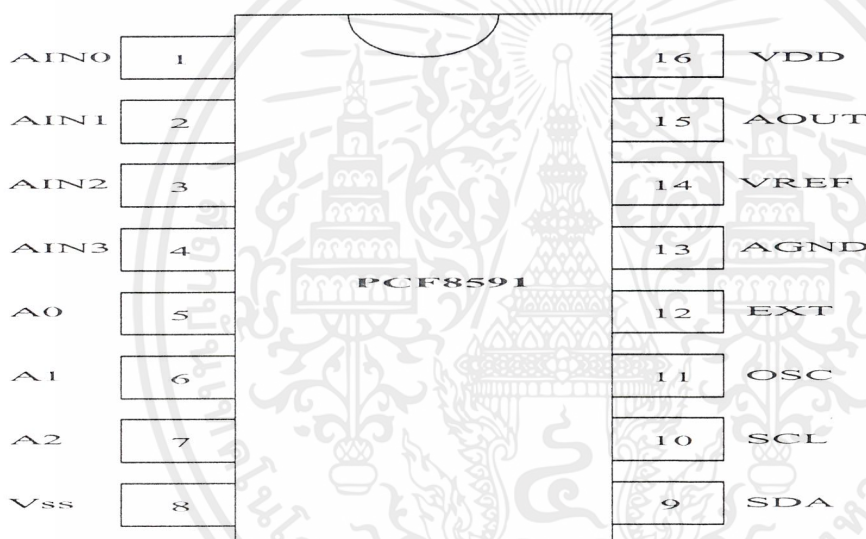
3.1 หลักการออกแบบฮาร์ดแวร์

ส่วนการแปลงสัญญาณอนาล็อกเป็นดิจิทัล

ไอซี PCF 8591 เป็นไอซีที่ทำหน้าที่รับสัญญาณอินพุตในรูปของสัญญาณอนาล็อกแล้วเปลี่ยนเป็นสัญญาณดิจิทัล เพื่อส่งให้CPUประมวลผล ซึ่งคุณสมบัติของไอซีเบอร์นี้จะเป็นแบบมอส

(CMOS)ขนาด 8 บิต สามารถรับอินพุตได้ 4 ช่องทาง ใช้หลักการประมาณค่าโดยลำดับ

(Successive approximation) ง่ายต่อการเชื่อมต่อเข้ากับ ไมโครคอนโทรลเลอร์ ตระกูลต่างๆ



รูปที่ 3.1 แสดงการจัดขาของไอซี ADC/DAC ขนาด 8 บิต PCF8591

การใช้งานเบื้องต้นPCF8591

ไอซี ADC/DAC เบอร์PCF8591เป็นADC แบบประมาณค่าโดยลำดับ (Successive Approximation)

ขนาด 8 บิตมีขาสัญญาณ 4 ช่องทางและมี DAC อีก 1 ช่องสัญญาณ การเชื่อมต่อแบบ I²C BUS และ

ยังสามารถเชื่อมต่อกันได้สูงสุด 8 ตัว ทำให้ได้วงจร ADC รวมสูงสุด 32 ช่องทาง และวง

จรDAC รวม 4 ช่อง ทำให้สามารถนำไปประยุกต์งานได้อย่างกว้างขวาง

คุณสมบัติทางเทคนิคของ PCF8591

1.ทำงาน โดยแหล่งจ่ายไฟชุดเดียว

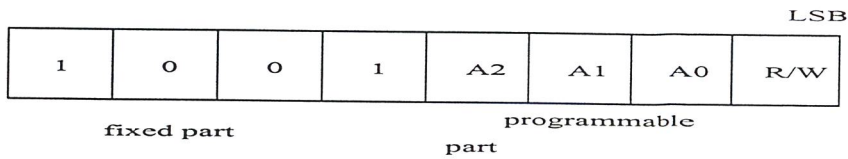
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กินกระแสไฟขณะอยู่ในสถานะ Standby ต่ำ
 4. การเชื่อมต่อเป็นแบบ I²C BUS
 5. สามารถเลือกตำแหน่งแอดเดรสทางฮาร์ดแวร์จากขา A0 A1 A2 ทำให้สามารถต่อได้สูงสุด 8 ตัว
 6. อัตราการสุ่มข้อมูล(Sampling)ขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาบน I²C BUS
 7. วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล มีช่องสัญญาณอินพุต 4 ช่องสัญญาณและยังเลือกการทำงานแยกช่องสัญญาณ หรือทำงานเป็นวงจรดิฟเฟอเรนเชียล
 8. การอ่านค่าสามารถกำหนดให้เลื่อนช่องอินพุตโดยอัตโนมัติได้
 9. สัญญาณอนาล็อกมีระดับแรงดัน Vss ถึง Vdd
 10. วงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลแบบประมาณค่าโดยลำดับ(Successive Approximation) ขนาด 8 บิต
 11. มีสัญญาณดิจิทัลเป็นอนาล็อกขนาด 8 บิต 1 ช่อง PCF 8591 สามารถทำหน้าที่เป็นไอซีแปลงสัญญาณดิจิทัลเป็นอนาล็อกในคราเดียวกัน
- รายละเอียดตำแหน่งขาสัญญาณต่างๆ
- | | |
|---------|---|
| AN0-AN3 | เป็นอนาล็อกอินพุต |
| A0-A2 | เป็นขาสำหรับแอดเดรสทางด้านฮาร์ดแวร์ |
| Vss | เป็นขา GND |
| SDA,SCL | เป็นขาเชื่อมต่อระบบ I ² C BUS |
| OSC | เป็นขาสำหรับสัญญาณนาฬิกาภายนอกเมื่อขา EXT ต่อลงมากับไฟเลี้ยง และจะเป็นขาเอาต์พุตเมื่อ EXT ต่อลง GND |
| EXT | เป็นขาสำหรับเลือกแหล่งกำเนิดสัญญาณนาฬิกา |
| ACND | เป็นขา GND ของสัญญาณอนาล็อก |
| Vref | เป็นป้อนแรงดันอ้างอิง |
| AOUT | เป็นขาเอาต์พุตของสัญญาณอนาล็อก |
| Vdd | เป็นขาต่อไฟเลี้ยง |

ฟังก์ชันต่างๆของ PCF 8591

ตำแหน่งแอดเดรส ในระบบ I²C BUS การติดต่ออุปกรณ์แต่ละตัวต้องระบุแอสแตรสของอุปกรณ์เหล่านั้นอย่างชัดเจน ซึ่งแอดเดรสของ PCF8591 แบ่งเป็น 2 ส่วน คือ Fixed part และ Programmable part แสดงดังรูป ในส่วนของ Programmable part จะเป็นส่วนระบุทางฮาร์ดแวร์ของ PCF8591 ในกรณีที่ต่อมากกว่า 1 ตัว และกำหนดการอ่านและเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แอดเดรสไบต์ของ PCF8591

ไบต์ควบคุม ไบต์ควบคุมเป็นไบต์ที่ต้องส่งให้กับ PCF 8591 ในการติดต่อแต่ละครั้งไบต์ควบคุมนี้เป็นการทำงานของPCF8591 รายละเอียดของไบต์ควบคุมแสดงดังรูปที่ 3.3บิตบนใช้สำหรับ Enable การทำงานของอนุลอกเอาต์พุต และใช้สำหรับเลือกโหมดการทำงานของสัญญาณอินพุตเป็นแบบแยกช่องสัญญาณ หรือแบบดิฟเฟอเรนเชียล

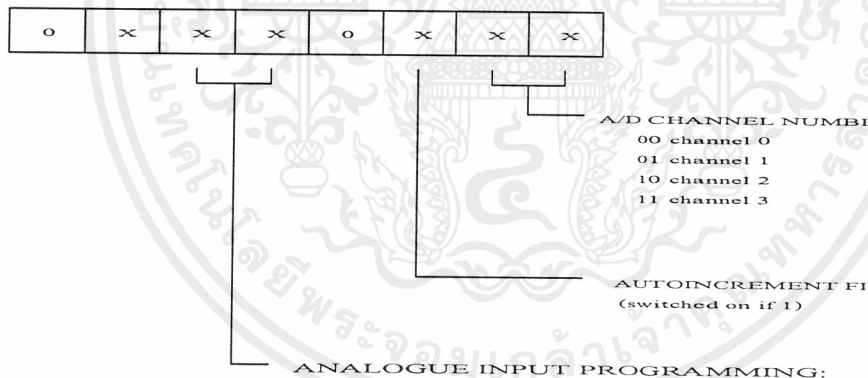
รายละเอียดของไบต์ควบคุม

บิตที่ 6 สำหรับ Enable ขาสัญญาณอนุลอกอินพุต

บิตที่ 4,5 ใช้สำหรับรูปแบบของขาสัญญาณอนุลอกอินพุต

บิตที่ 2 ใช้สำหรับเลือกรูปแบบการอ่านข้อมูลขาสัญญาณขาอินพุตอนุลอกว่าจะเป็นการอ่านอินพุตเดี่ยว หรือ อ่านแบบเรียงลำดับทุกอินพุต

บิต 0,1 ใช้สำหรับกำหนดช่องสัญญาณอินพุตที่ต้องการจะอ่าน



รูปที่ 3.3 รายละเอียดไบต์ควบคุมของ PVF8591

การอ่านข้อมูลอินพุตอนุลอกของPCF8591

หลังจากที่ส่งไบต์แอดเดรสและไบต์ควบคุมแล้วในไบต์ที่ 3 นี้จะเป็นแอดเดรสซึ่งในส่วนของ Programmable Part ต้องระบุการอ่าน(R/W เป็น 1) เมื่อส่งไบต์ที่ 3 เสร็จ PCF 8591 จะส่งมุลออกมาเมื่อส่งไบต์แอดเดรสและไบต์ควบคุม ไบต์ที่3 จะเป็นข้อมูลที่ต้องการแปลงเป็นสัญญาณอนุลอก ซึ่งการเชื่อมต่อ PCF8591กับไมโครคอนโทรลเลอร์ แล้วไปวิเคราะห์หาค่าไป เมื่อแปลงสัญญาณเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วชุดควบคุมก็จะส่งสัญญาณที่มีสถานะ “1” ไปยังบิต R/W ของ PCF ของ PCF 8591 เพื่อเป็นการ Enable ให้ PCF8591 ส่งข้อมูลออกมา ซีพียูก็จะทำการอ่านค่าที่เข้ามาเพื่อประมวลผลต่อไป ส่วนอินเทอร์เฟซและควบคุม

ส่วนที่เป็นหัวใจหลักคือ ไมโครคอนโทรลเลอร์ เบอร์ 89C51RD+ ซึ่งมีความคล่องตัวในการใช้งาน มีพอร์ต(Port) ให้ใช้งานในตัวได้ 4 พอร์ต มีตัวจับเวลา(timer) พร้อมกับรีจิสเตอร์หน้าที่พิเศษ(SFR) ให้ใช้งานอีกหลายตัวที่ทำให้สามารถที่จะนำไปประยุกต์ใช้งานในด้านต่างๆ ได้อย่างกว้างขวาง โดยมีหน้าที่และหลักการดังนี้

- 1.ควบคุมการเปลี่ยนแปลงสัญญาณของไอซี PCF 8591รวมทั้งควบคุมจังหวะการสวิทช์ เพื่อเลือกสัญญาณอินพุตแต่ละChannel
- 2.ควบคุมการเชื่อมโยงติดต่อสื่อสารข้อมูลระหว่างชุดตรวจจับสัญญาณกับเครื่องคอมพิวเตอร์ให้เป็นไปได้อย่างถูกต้องและเชื่อถือได้

ในส่วนของการติดต่อส่งผ่านข้อมูลจะใช้การสื่อสารแบบอนุกรม โดยจะใช้ ไอซี MAX 232 เพื่อเปลี่ยนระดับสัญญาณข้อมูลดิจิทัล(Digital data) ไปเป็นระดับสัญญาณดิจิทัล (Digital Signal) ตามมาตรฐาน RS - 232

กลุ่มผู้จัดได้เลือกใช้บอร์ด ISP-1000 ของบริษัท คิลารีเสิร์ช ซึ่งบอร์ดนี้มีทั้ง ไอซีไมโครคอนโทรลเลอร์ เบอร์ 8C51RD+ และ ไอซี เอ/ดี PCF8951 โดยวงจรรวมบนบอร์ด ISP-1000 แสดงดังรูป 3.5

ข้อมูลของบอร์ด ISP-1000

ISP-1000 เป็น Embedded controller board ที่มีทรัพยากรใช้อย่างครบครันในขนาดกระทัดรัด ใช้ไฟเลี้ยง 5 โวลต์เท่านั้น I/O Port เป็นแบบ TTL ไอซี 74HC374 เป็นแบบ Output Port และ 74HC245 เป็น Input port (User = 3 output port , Onboard = 2 output , 1input) โดย Onboard นั้นมีระบบเตือนแบบเตอรี่ย่อนโดยต่ออยู่ที่ P1.1(เมื่อเลือก Jumper ไว้ที่ En) และยังมีระย On/Off ไฟ Backlight ของ LCD โดยอยู่ที่ Use Pc.3

ด้วยคุณสมบัติของ CPU เบอร์ 89C51RD+ ซึ่งมีขนาด Flash Memory 64 K , Ram 1 K และสามารถดาวน์โหลดโปรแกรมจาก PC เพื่อโปรแกรมเข้า CPU ได้โดยตรงโดยไม่ต้องถอด CPU ออกจากบอร์ดมาโปรแกรม ทำให้การพัฒนาโปรแกรมเป็นไปด้วยความรวดเร็วและสะดวกสบายยิ่งขึ้น การดาวน์โหลดสามารถใช้ได้กับ R2D2-Monitor ของคิลารีหรือกระทำผ่าน Program Winisp โดยโหลดสามารถใช้ได้ อีกทั้งระบบ Watch Dog ในตัวไม่ต้องต่อวงจรเพิ่มเติม ดังนั้นบอร์ด ISP-1000 จึงเหมาะกับการพัฒนางานด้านต่างๆ เป็นอย่างมาก

คุณสมบัติของบอร์ด

CPU 9C51RD+ MICROCONTROLLER WITH 64 K FLASH AND 1K RAM ON CHIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|----------|---|
| CLOCK | 2.1184 MHz MODULE OSCILATOR |
| MEMOMORY | RAM 8/32 KBYTE ON BOARD (SELECTABLE) DEFAULT 8 KBYTE |
| PORT | 24 BIT OUTPUT PORT (74HC374) (PA=0A000H,PA=0B000H,PC=0C000H) 12 BIT I/O PORT (PCF8574A) 3 BIT I/O PORT (MCU) 1 BIT TOOL I/O PORT (MCU) 8*8 DISPLAY PORT (74HC374,74HC245)(D000H,E000H,F000H) LCD PORT (08000-08003H) |
| RS-232 | DS 275 |
| RS-435 | 75176,MAX3082(OPTION) |
| RESET | DS 1707(RESET AND LOWBATT CHECK) |
| RTC | DS1307 + BATTERY SOCKET(OPTION) |
| ADC/DAC | PCF8591 ADC 8 BIT 4 INPUT /DAC 8 BIT 1 OUTPUT (OPTION) |
| OTHER | 4 BIT DIP SWITCH (PB.4-PC.7) AT E000H 4 USER GREEN LED (PC.4-PC.7) AT F000H 2 INPUT SWICH (PCF8574) 2OUTPUT YELLOW LED(PCF8574) 1 BIT SPEAKER ONBOARD(P1.7) 1 BIT BACKLIGHT CONTROL FOR LCD (PC .3 F000H) 1 BIT ENABLE 74HC374(P1.0) 1 LED MONITER (RS-485) DIRECTION (P1.2) |

SUPPLY 5 VDC

SIZE 4 *5.5 INPUT PORT

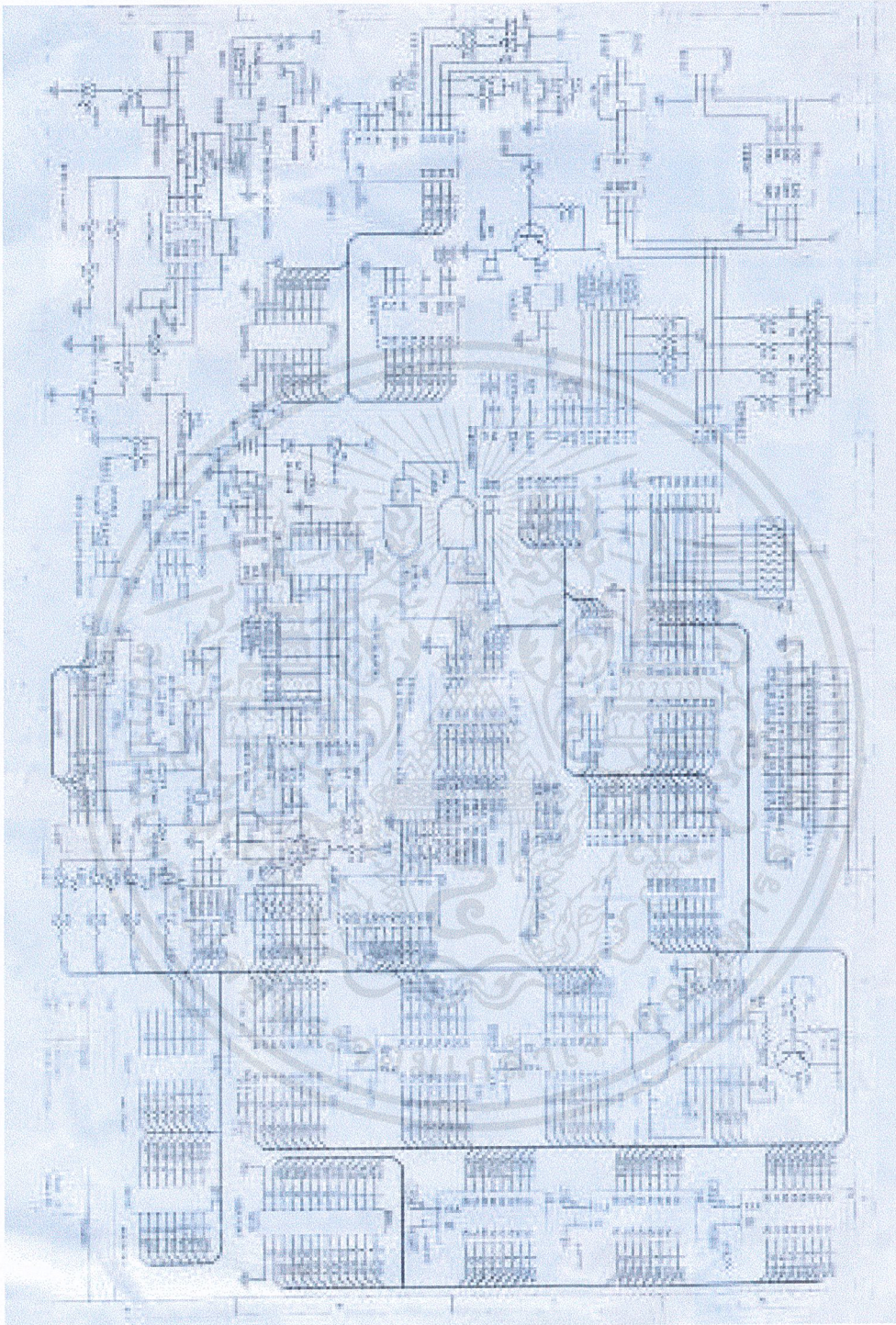
การใช้งาน User Port ของบอร์ด ISP-1000 ถูกออกแบบให้ เป็น Output port เท่านั้นและเป็นแบบTTL คือ ไอซี 74HC374 เมื่อต้องการให้ 74HC374 นี้ทำงาน จะต้องตั้งให้บิต 374EN(P1.0) เป็นลอจิก 0 (เสมือนเขียน Control Port ให้กับ 8255) ตัวอย่างโปรแกรมการใช้งาน User output port แบบภาษา C รวมทั้งส่วนอื่นๆด้วย ดูได้จากไฟล์ ISPSQC.C การใช้งานระบบรีเซตและ ระบบเตือนแบตเตอรี่ร่อนบอร์ด ISP-1000 ใช้ไอซี DS-1707 ในการ Reset ทั้งช่วง Power Up และ Down และสามารถใช้สวิทช์ Reset นอกจากนี้ยังเป็นระบบเตือนแบตเตอรี่ร่อนซึ่งจะต่ออยู่กับ P1.1 และ LEC แสดงผลสีเขียว (Active Low) หากต้องการใช้ระบบเตือนแบตเตอรี่ร่อนจะต้อง ปรับ Jumper ไปที่ Lowbatt.EN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน Serial Port Port สื่อสารอนุกรมของบอร์ด ISP-1000 สามารถเลือกใช้งานได้ 2 แบบ คือ RS-232 ให้ใช้ DS275 และใช้งานที่ขั้วต่อ 3 pin (Default) มี LED แสดงสถานะการรับส่งข้อมูล RS 232 หากต้องการใช้งาน RS 485 ให้ใช้ 7516 หรือ MAX 3082 และการใช้งานที่ขั้ว 2 pin โดยให้ P1.2 = 1 เป็นการส่งข้อมูล ซึ่งจะมี LED สีแดงแสดงสถานะการรับส่งข้อมูล RS-485 นี้ด้วยการใช้งาน RS-485 ซึ่งเหมาะสำหรับการสื่อสารที่เป็น Network ที่มีระยะการสื่อสารไม่เกิน 1.2 Km และจำนวนการต่อ Node ไม่เกิน 32 จุดสิ่งที่ขาดไม่ได้คือ การทำโปรแกรมควบคุม หรือ Protocol ในการควบคุมทิศทาง และการสื่อสารข้อมูลในแต่ละจุด ในระบบ ให้ทำงานได้อย่างมีระเบียบและประสิทธิภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงวงจรมนบอร์ด ISP-1000

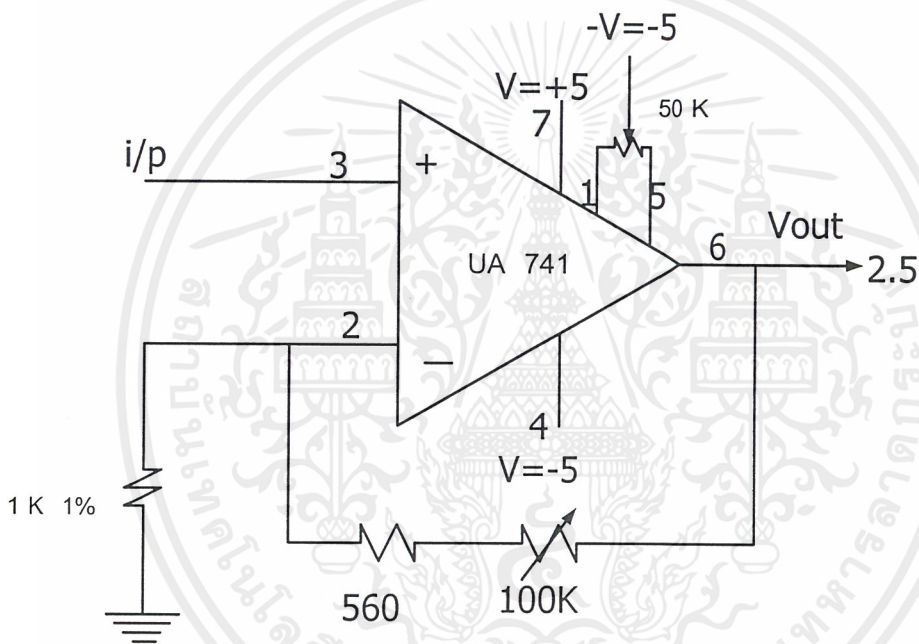
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

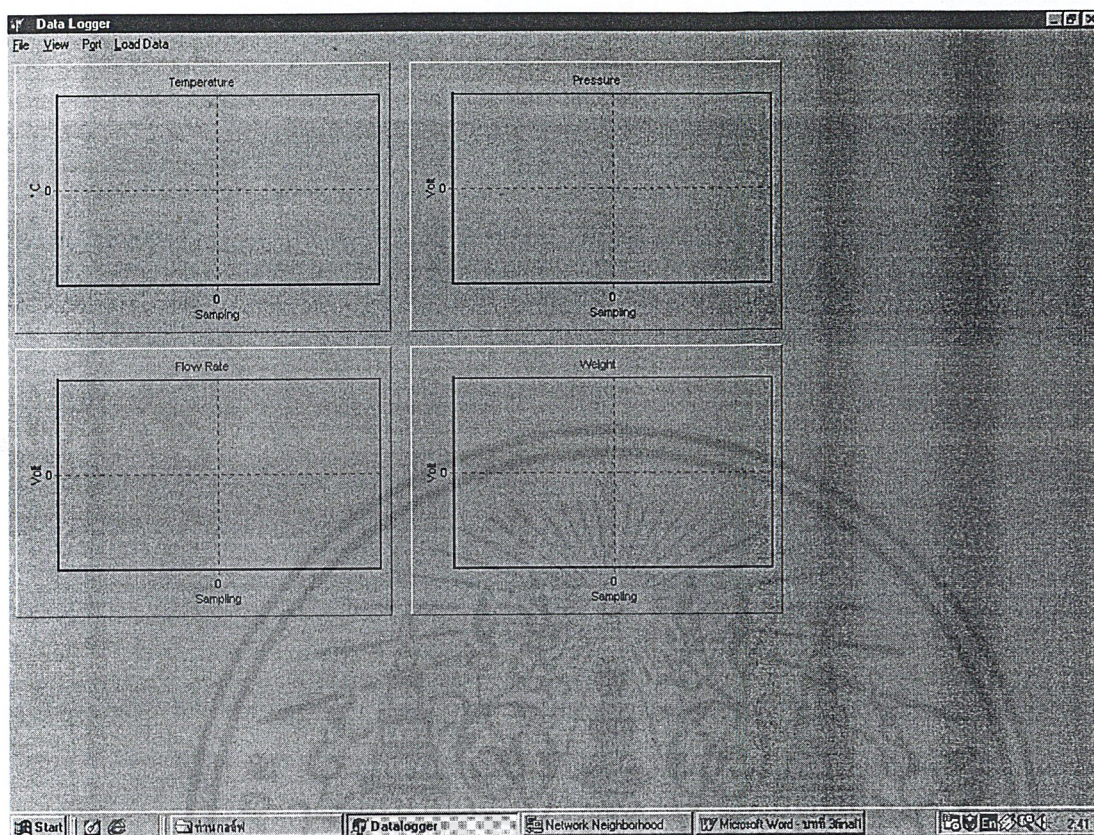
อธิบายผลการทดลอง

การทดลองใช้ไฟฟ้าป้อนเข้า A/D โดยปกติ A/D รับสัญญาณไฟฟ้า 0-2.5 V แต่สัญญาณของเซนเซอร์ มาตรฐาน 1-5 V และที่ช่องที่ 1 เป็นสัญญาณ ที่รับจาก Thermocouple Type K ช่วง 0-4mV ที่ตรงกับอุณหภูมิ 1-100 องศาเซลเซียส ดังนั้นจึงทำการออกแบบวงจรใหม่ เป็น ช่องที่ 1 แปลงสัญญาณ จาก 0-4mV เป็น 0-2.5 V และช่องที่ 2-4 แปลงสัญญาณ 0-5 V เป็น 0-2.5 V โดยช่องที่ 1 แสดงผลในหน้าจอเป็นองศาเซลเซียส และช่องที่ 2-4 แสดงผลเป็นแบบ 0-5V



รูปที่ 4.1 วงจรขยาย 625 เท่าสำหรับขยายสัญญาณ 0-4 mV เป็น 0-2.5 V

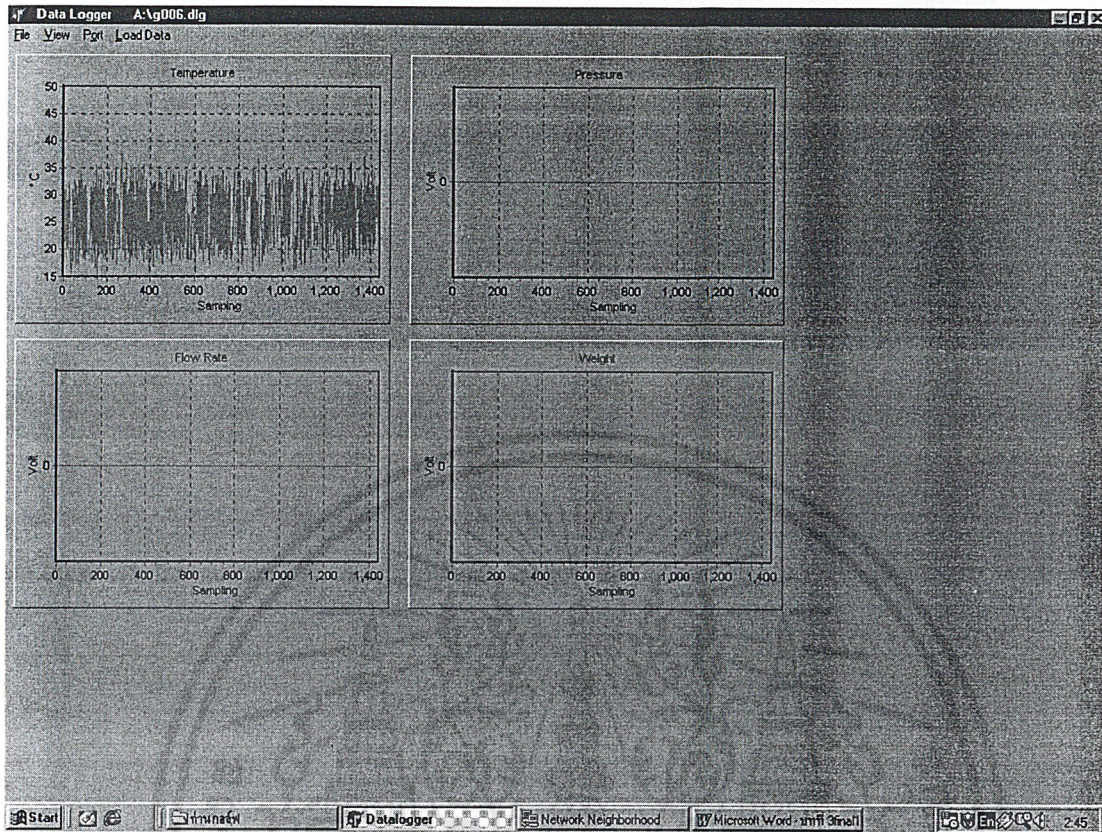
รูปแสดง หน้าจอปกติเป็นดังนี้



รูปที่ 4.2 ตารางกราฟปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การทดลองโดยการป้อนสัญญาณ 1mV เข้าช่องที่ 1



รูปที่ 4.3 ผลการป้อนสัญญาณ 1mV เข้าช่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้คือกราฟแสดงค่าประมาณ 20-30 องศา และค่าเฉลี่ยที่ 27 องศา

Table for Data Logger

Data Logger

| | Temperature | Pressure | Flow Rate | Weight |
|-----|-------------|----------|-----------|--------|
| 944 | 29.0196 | 0.0000 | 0.0000 | 0.0000 |
| 945 | 27.8431 | 0.0000 | 0.0000 | 0.0000 |
| 946 | 24.3137 | 0.0000 | 0.0000 | 0.0000 |
| 947 | 21.1764 | 0.0000 | 0.0000 | 0.0000 |
| 948 | 23.1372 | 0.0000 | 0.0000 | 0.0000 |
| 949 | 25.8823 | 0.0000 | 0.0000 | 0.0000 |
| 950 | 27.4509 | 0.0000 | 0.0000 | 0.0000 |
| 951 | 28.6274 | 0.0000 | 0.0000 | 0.0000 |
| 952 | 30.9803 | 0.0000 | 0.0000 | 0.0000 |
| 953 | 30.9803 | 0.0000 | 0.0000 | 0.0000 |
| 954 | 31.3725 | 0.0000 | 0.0000 | 0.0000 |
| 955 | 32.9411 | 0.0000 | 0.0000 | 0.0000 |
| 956 | 29.8039 | 0.0000 | 0.0000 | 0.0000 |
| 957 | 27.8431 | 0.0000 | 0.0000 | 0.0000 |
| 958 | 26.2745 | 0.0000 | 0.0000 | 0.0000 |

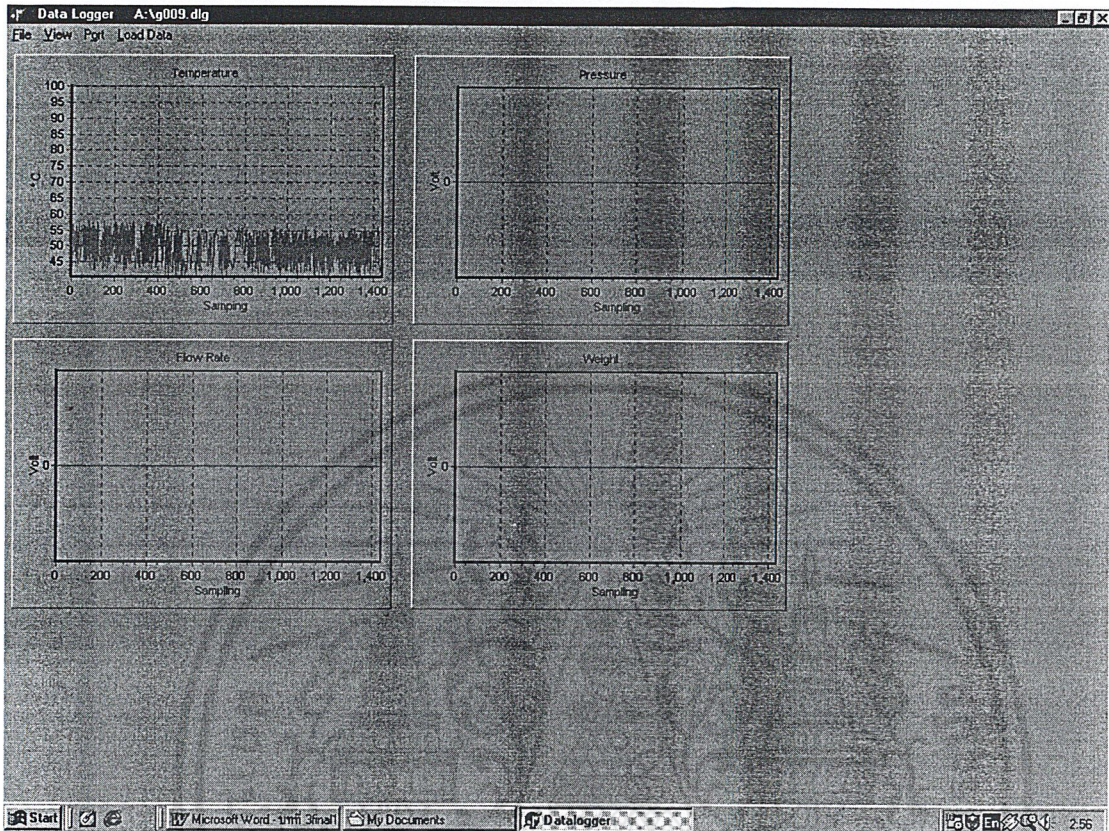
Save Close

Microsoft Word - บทที่ 3 final My Documents Datalogger 2.52

ตารางที่ 4.1 กราฟแสดงค่าประมาณ 20-30 องศา และค่าเฉลี่ยที่ 27 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดลองโดยการป้อนสัญญาณ 2mV เข้าช่องที่ 1



รูปที่ 4.3 การป้อนสัญญาณ 2mV เข้าช่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้คือกราฟแสดงค่าประมาณ 45-55 องศา และค่าเฉลี่ยที่ 50 องศา

Table for Data Logger

Date Logger

| | Temperature | Pressure | Flow Rate | Weight |
|-----|-------------|----------|-----------|--------|
| 944 | 56.0784 | 0.0000 | 0.0000 | 0.0000 |
| 945 | 45.0980 | 0.0000 | 0.0000 | 0.0000 |
| 946 | 51.3725 | 0.0000 | 0.0000 | 0.0000 |
| 947 | 48.2352 | 0.0000 | 0.0000 | 0.0000 |
| 948 | 44.3137 | 0.0000 | 0.0000 | 0.0000 |
| 949 | 54.5098 | 0.0000 | 0.0000 | 0.0000 |
| 950 | 47.4509 | 0.0000 | 0.0000 | 0.0000 |
| 951 | 47.8431 | 0.0000 | 0.0000 | 0.0000 |
| 952 | 52.9411 | 0.0000 | 0.0000 | 0.0000 |
| 953 | 43.9215 | 0.0000 | 0.0000 | 0.0000 |
| 954 | 46.6666 | 0.0000 | 0.0000 | 0.0000 |
| 955 | 54.5098 | 0.0000 | 0.0000 | 0.0000 |
| 956 | 50.1960 | 0.0000 | 0.0000 | 0.0000 |
| 957 | 49.8039 | 0.0000 | 0.0000 | 0.0000 |
| 958 | 47.0588 | 0.0000 | 0.0000 | 0.0000 |

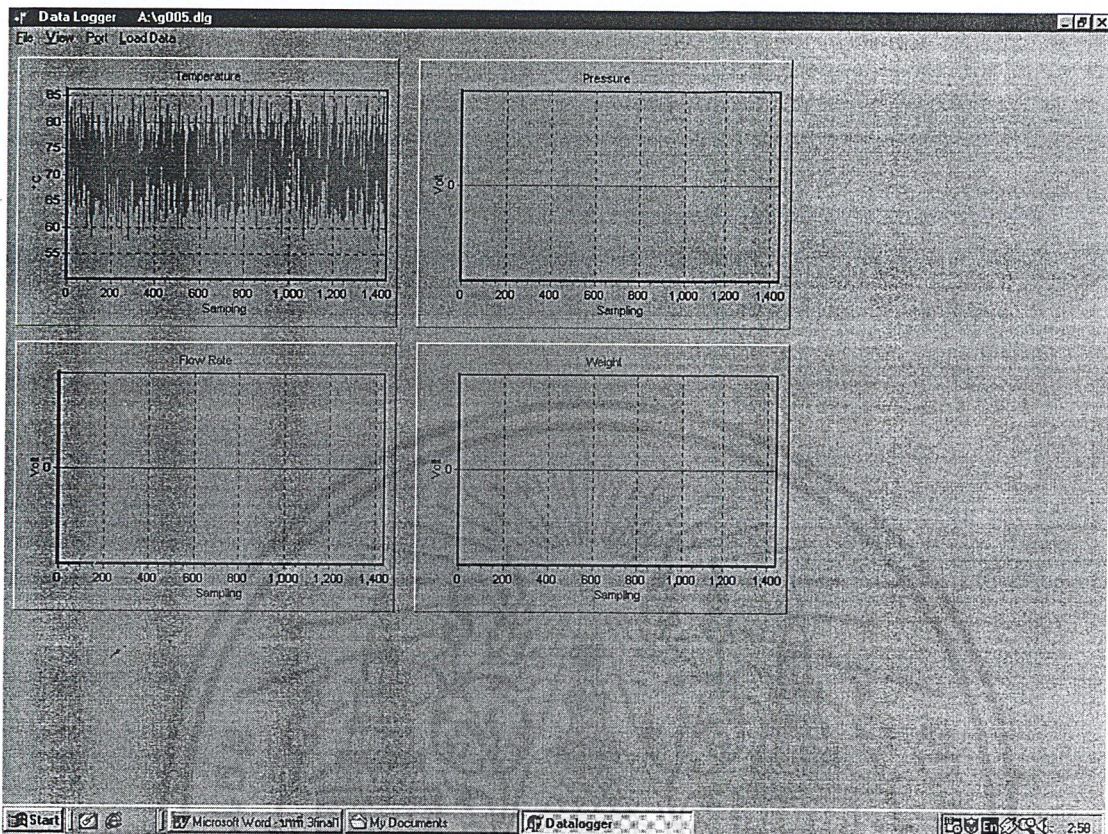
Save Close

Start Microsoft Word - มหัทธินิพนธ์ My Documents Datalogger 2:57

ตารางที่ 4.2 กราฟแสดงค่าประมาณ 45-55 องศา และค่าเฉลี่ยที่ 50 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดลองโดยการป้อนสัญญาณ 3mV เข้าช่องที่ 1



รูปที่ 4.4 การป้อนสัญญาณ 3mV เข้าช่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้คือกราฟแสดงค่าประมาณ 65-80 องศา และค่าเฉลี่ยที่ 75 องศา

Table for Data Logger

Data Logger

| | Temperature | Pressure | Flow Rate | Weight |
|-----|-------------|----------|-----------|--------|
| 944 | 77.6470 | 0.0000 | 0.0000 | 0.0000 |
| 945 | 62.3529 | 0.0000 | 0.0000 | 0.0000 |
| 946 | 70.1960 | 0.0000 | 0.0000 | 0.0000 |
| 947 | 75.2941 | 0.0000 | 0.0000 | 0.0000 |
| 948 | 80.0000 | 0.0000 | 0.0000 | 0.0000 |
| 949 | 67.0588 | 0.0000 | 0.0000 | 0.0000 |
| 950 | 76.8627 | 0.0000 | 0.0000 | 0.0000 |
| 951 | 73.7254 | 0.0000 | 0.0000 | 0.0000 |
| 952 | 78.4313 | 0.0000 | 0.0000 | 0.0000 |
| 953 | 73.7254 | 0.0000 | 0.0000 | 0.0000 |
| 954 | 67.0588 | 0.0000 | 0.0000 | 0.0000 |
| 955 | 69.0196 | 0.0000 | 0.0000 | 0.0000 |
| 956 | 72.1568 | 0.0000 | 0.0000 | 0.0000 |
| 957 | 77.2549 | 0.0000 | 0.0000 | 0.0000 |
| 958 | 69.0196 | 0.0000 | 0.0000 | 0.0000 |

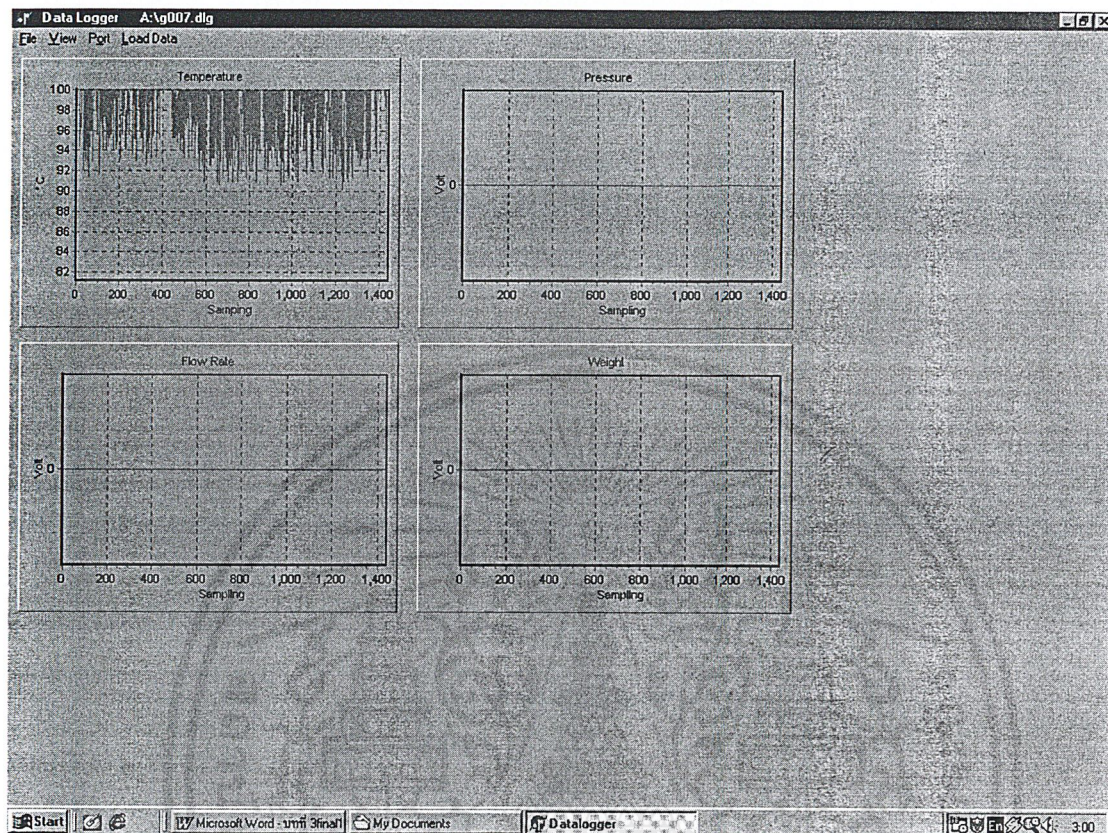
Save Close

Start Microsoft Word - บทที่ 3 (กมล) My Documents Datalogger 2:59

ตารางที่ 4.3 กราฟแสดงค่าประมาณ 65-80 องศา และค่าเฉลี่ยที่ 75 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ทดลอง โดยการป้อนสัญญาณ 4mV เข้าช่องที่ 1



รูปที่ 4.5 การป้อนสัญญาณ 4mV เข้าช่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้คือกราฟแสดงค่าประมาณ 80-100 องศา และค่าเฉลี่ยที่ 100 องศา

| | Temperature | Pressure | Flow Rate | Weight |
|-----|-------------|----------|-----------|--------|
| 966 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 967 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 968 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 969 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 970 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 971 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 972 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 973 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 974 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 975 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 976 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 977 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 978 | 100.0000 | 0.0000 | 0.0000 | 0.0000 |
| 979 | 98.4313 | 0.0000 | 0.0000 | 0.0000 |
| 980 | 94.5098 | 0.0000 | 0.0000 | 0.0000 |

ตารางที่ 4.4 กราฟแสดงค่าประมาณ 80-100 องศา และค่าเฉลี่ยที่ 100 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

เนื่องจากโครงงานนี้เป็นโครงงานร่วมกับภาควิชาอาหารและในระบบบอบแห้งของภาควิชาอาหารยังไม่มีเครื่องมือเซนเซอร์ มีแต่Thermocouple type K ที่รับสัญญาณ0-4mV ดังนั้นในการทดลองจึงต้องทำช่องสัญญาณ1ช่องสำหรับ0-4mVแปลงเป็น 0-2.5 V และช่องสัญญาณที่เหลือใช้ 0-5V แปลงเป็น 0-2.5 V เพื่อรองรับสัญญาณมาตรฐาน และในการทดลองได้ใช้สัญญาณจำลองจากเครื่องกำเนิดสัญญาณ

เครื่องมือวัดทางอุตสาหกรรมนี้สามารถตั้งได้โดยลำพังโดยไม่ต้องต่อเข้ากับคอมพิวเตอร์ เพียงแต่นำสัญญาณมาตรฐาน 0-5 V ต่อเข้าช่องที่ 2-4 และสำหรับช่องที่1 เป็นสัญญาณ0-4mV (Thermocouple type K)

แต่เนื่องจากผลการทดลองในส่วนช่องที่ 1 ไม่สามารถลดสัญญาณรบกวนได้เนื่องจากIC ยังไม่มีคุณภาพพอเนื่องจาก เป็นสัญญาณที่มีขนาดต่ำ ดังนั้นจึงต้องทำการหาIC ที่มีคุณภาพสูงกว่าที่มีอยู่คือUA741PC

ถ้าหากว่ามี IC เฉพาะงานก็สามารถนำไปใช้งานได้ในส่วนช่องที่1 ส่วนช่องที่2-4 สามารถใช้งานได้เป็นปกติ ดังนั้นในส่วนเครื่องม้วัดทางอุตสาหกรรมนี้จึงถือว่าสำเร็จตามจุดประสงค์แต่ยังไม่สามารถใช้งานได้เนื่องจากอุปกรณ์ไม่พร้อม

ภาคผนวก

Source code Delphi 5.0

```
unit DataLogger_1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
ExtCtrls, TeeProcs, TeEngine, Chart, DBChart, Display, wsc, Series, Menus;
```

```
type
```

```
Data_data = array[1..1440] of double;
```

```
TForm1 = class(TForm)
```

```
DBChart1: TDBChart;
```

```
DBChart2: TDBChart;
```

```
DBChart3: TDBChart;
```

```
DBChart4: TDBChart;
```

```
Series1: TLineSeries;
```

```
Series2: TLineSeries;
```

```
Series3: TLineSeries;
```

```
Series4: TLineSeries;
```

```
MainMenu1: TMainMenu;
```

```
File1: TMenuItem;
```

```
Data1: TMenuItem;
```

```
Open1: TMenuItem;
```

```
Save1: TMenuItem;
```

```
Exit1: TMenuItem;
```

```
OpenDialog1: TOpenDialog;
```

```
SaveDialog1: TSaveDialog;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Port1: TMenuItem;
COM11: TMenuItem;
COM21: TMenuItem;
COM31: TMenuItem;
COM41: TMenuItem;
Table1: TMenuItem;
Connect1: TMenuItem;
Disconnect1: TMenuItem;
N1: TMenuItem;
New1: TMenuItem;
N2: TMenuItem;
N3: TMenuItem;
Data11: TMenuItem;
procedure FormCreate(Sender: TObject);
procedure plotgraph;
procedure data_c;
procedure Exit1Click(Sender: TObject);
procedure Data11Click(Sender: TObject);
procedure New1Click(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure Save1Click(Sender: TObject);
procedure COM11Click(Sender: TObject);
procedure COM21Click(Sender: TObject);
procedure COM31Click(Sender: TObject);
procedure COM41Click(Sender: TObject);
procedure Table1Click(Sender: TObject);
procedure Connect1Click(Sender: TObject);
procedure Disconnect1Click(Sender: TObject);
private
  { Private declarations }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Port : Integer;
Baud : Integer;
Parity : Integer;
DataBits : Integer;
StopBits : Integer;
public
  { Public declarations }
end;

var
  Form1: TForm1;
  Data_1: Data_data;
  Data_2: Data_data;
  Data_3: Data_data;
  Data_4: Data_data;

implementation

uses table;

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
var code : integer;
begin
  (* initialize settings *)
  Port := COM1;
  Baud := Baud38400;
  Parity := NoParity;
  DataBits := WordLength8;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

StopBits := OneStopBit;

end;

procedure TForm1.data_c;
var temp,i : integer;
begin
    for i := 1 to 1440 do
        data_1[i] := data_1[i]*5/255;
    for i := 1 to 1440 do
        data_2[i] := data_2[i]*5/255;
    for i := 1 to 1440 do
        data_3[i] := data_3[i]*5/255;
    for i := 1 to 1440 do
        data_4[i] := data_4[i]*5/255;
end;

procedure TForm1.plotgraph;
begin
    DbChart1.Series[0].Clear;
    Dbchart1.Series[0].AddArray(Data_1);
    DbChart2.Series[0].Clear;
    Dbchart2.Series[0].AddArray(Data_2);
    DbChart3.Series[0].Clear;
    Dbchart3.Series[0].AddArray(Data_3);
    DbChart4.Series[0].Clear;
    Dbchart4.Series[0].AddArray(Data_4);
end;

procedure TForm1.Exit1Click(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    SioDone(Port);
    Application.Terminate;
end;

```

```

procedure TForm1.Data1Click(Sender: TObject);

```

```

var Code,i,ii,cheak : integer;

```

```

begin

```

```

// Cheak

```

```

SioPutc(Port,chr($A1));

```

```

for i := 1 to 5000 do

```

```

begin

```

```

    cheak := 1;

```

```

    Code := SioGetc(Port);

```

```

    if code = $A1 then

```

```

        begin

```

```

            cheak := 0;

```

```

            break;

```

```

        end;

```

```

    end;

```

```

if cheak = 1 then

```

```

begin

```

```

    showmessage('Error');

```

```

    SioDone(Port);

```

```

    Application.Terminate;

```

```

end;

```

```

// Read_Data_1

```

```

SioPutc(Port,chr($11));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for i := 1 to 1440 do
begin
  for ii := 1 to 5000 do
    begin
      Code := SioGetc(Port);
      if code >= 0 then break;
    end;
    Data_1[i] := code;
  end;

//  Read_Data_2
  SioPutc(Port,chr($12));

  for i := 1 to 1440 do
  begin
    for ii := 1 to 5000 do
      begin
        Code := SioGetc(Port);
        if code >= 0 then break;
      end;
      Data_2[i] := code;
    end;
  end;

//  Read_Data_3
  SioPutc(Port,chr($13));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for i := 1 to 1440 do
begin
for ii := 1 to 5000 do
begin
Code := SioGetc(Port);
if code >= 0 then break;
end;
Data_3[i] := code;
end;

// Read_Data_4
SioPutc(Port,chr($14));

for i := 1 to 1440 do
begin
for ii := 1 to 5000 do
begin
Code := SioGetc(Port);
if code >= 0 then break;
end;
Data_4[i] := code;
end;

data_c;
plotgraph;
end;

procedure TForm1.New1Click;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DbChart1.Series[0].Clear;
DbChart2.Series[0].Clear;
DbChart3.Series[0].Clear;
DbChart4.Series[0].Clear;
form1.Caption := ' Data Logger  ';
end;

procedure TForm1.Open1Click(Sender: TObject);
var F_file : file of data_data;
    F_name : string ;
begin
if opendialog1.Execute then
begin
F_name := opendialog1.FileName ;
assignfile(F_file,F_name);
reset(F_file);
read(F_file,data_1);
read(F_file,data_2);
read(F_file,data_3);
read(F_file,data_4);

closefile(F_file);

plotgraph;

form1.Caption := ' Data Logger  ' + F_name;
end;
end;

procedure TForm1.Save1Click(Sender: TObject);
var F_file : file of Data_data;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

F_name : string ;
begin
  F_name := opendialog1.FileName ;
  if F_name="" then
    savedialog1.FileName := 'nonamed'
  else
    savedialog1.FileName := F_name ;
  if savedialog1.Execute then
    begin
      F_name := savedialog1.FileName ;
      assignfile(F_file,F_name);
      Rewrite(F_file);

      write(F_file,Data_1);
      write(F_file,Data_2);
      write(F_file,Data_3);
      write(F_file,Data_4);

      closefile(F_file);
    end;
    form1.Caption := ' Data Logger ' + F_name;
end;

procedure TForm1.COM11Click(Sender: TObject);
begin
  Port := COM1;
  com11.Checked := true;
  com21.Checked := False;
  com31.Checked := False;
  com41.Checked := False;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
```

```
procedure TForm1.COM21Click(Sender: TObject);
```

```
begin
```

```
Port := COM2;
```

```
com11.Checked := False;
```

```
com21.Checked := true;
```

```
com31.Checked := False;
```

```
com41.Checked := False;
```

```
end;
```

```
procedure TForm1.COM31Click(Sender: TObject);
```

```
begin
```

```
Port := COM3;
```

```
com11.Checked := False;
```

```
com21.Checked := False;
```

```
com31.Checked := true;
```

```
com41.Checked := False;
```

```
end;
```

```
procedure TForm1.COM41Click(Sender: TObject);
```

```
begin
```

```
Port := COM4;
```

```
com11.Checked := False;
```

```
com21.Checked := False;
```

```
com31.Checked := False;
```

```
com41.Checked := true;
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.Table1Click(Sender: TObject);
begin
    form2.visible := true;
end;

```

```

procedure TForm1.Connect1Click(Sender: TObject);
var Code,i,ii,cheak : integer;
begin
    (* initialize WSC *)
    Code := SioReset(Port,2048,512);
    if Code < 0 then
    begin
        Showmessage('Port Error');
        SioDone(Port);
        Application.Terminate;
    end;
    SioBaud(Port,Baud);
    SioParms(Port, Parity, StopBits, DataBits);
    SioDTR(Port,'S');
    SioRTS(Port,'S');

```

```

// SioPute(Port,chr($AA));
    connect1.Checked := true;
    Disconnect1.Checked := false;
end;

```

```

procedure TForm1.Disconnect1Click(Sender: TObject);
begin
    Disconnect1.Checked := true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
connect1.Checked := false;  
SioDone(Port);  
end;  
  
end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code ภาษา C-51

```

/* Filename   ISPSQC.C
Description  Super QC for ISP-xxxx Board
Hardware     Any ISP-xxxx Board clock 22.1184 MHz
Compiler     Franklin v3.2
Engineer     Kriangsak B.
Company      Sila Research Co.,Ltd. */

```

```

#include <reg52.h>
#include <absacc.h>
#include <ctype.h>
#include <intrins.h>
#include <math.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/***** I/O PORT *****/

sbit  RAMA15 = P1^0;      /* ram address bit */
sbit  RAMA16 = P1^1;

sbit  IRSCL  = P1^3;      /* ds1307 RTC */
sbit  IRSDA  = P1^4;

sbit  IPSCL  = P1^5;      /* pcf8574 I/O Bit */
sbit  IPSDA  = P1^6;

sbit  SOUNDDB = P1^7;    /* Sound Bit */
sbit  WDBIT  = P3^4;     /* Watch dog bit */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sbit PBIT0 = P3^2;      /* 3B Port */
sbit PBIT1 = P3^3;
sbit PBIT2 = P3^5;

#define LCDWRC XBYTE [0x8000] /* lcd port */
#define LCDRDC XBYTE [0x8001]
#define LCDWRD XBYTE [0x8002]
#define LCDRDD XBYTE [0x8003]

#define ONBA XBYTE [0xa000] /* 8255 on board */
#define ONBB XBYTE [0xa001]
#define ONBC XBYTE [0xa002]
#define ONBP XBYTE [0xa003]

#define USRA XBYTE [0xb000] /* 8255 user */
#define USRB XBYTE [0xb001]
#define USRC XBYTE [0xb002]
#define USRP XBYTE [0xb003]

#define RESVEC XBYTE [0x7ffe] /* reset vector */
#define SPEEDM XBYTE [0x7fff] /* speed memory */

/* for ISP-1000 */

sbit EN374 = P1^0;      /* 74LS374 Enable */
sbit LOWB = P1^1;      /* Low Batt from DS1707 */
sbit D485 = P1^2;      /* RS485 Direction */
sbit TOOL = P3^4;      /* Tool I/O */

```

```
/* for I-400 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sbit PBIT20 = P3^4;      /* 3B Port (2) */
sbit PBIT21 = P3^6;
sbit PBIT22 = P3^7;
sfr  XLCDDA = 0x80;      /* lcd data d0-d7 */
sbit  XLCDCK = P1^0;     /* ls374 clock */
sbit  XLCDBL = P2^7;     /* back light */
sbit  XLCDRW = P2^4;     /* r/w */
sbit  XLCDRS = P2^5;     /* rs */
sbit  XLCDEN = P2^6;     /* enable */

/***** INT-RAM WORKING AREA *****/

unsigned char data LCDBUF[40]; /* lcd buffer */
unsigned char data TIMBUF[8]; /* ss,mm,hh,ww,dd,mm,yy,cc */
unsigned char data DISBUF[8]; /* 7 segment display buffer */
unsigned char data ATDBUF[4]; /* A/D buffer */

unsigned char bdata BITBUF; /* bit data for working */
sbit BIT0 = BITBUF^0;
sbit BIT1 = BITBUF^1;
sbit BIT2 = BITBUF^2;
sbit BIT3 = BITBUF^3;
sbit BIT4 = BITBUF^4;
sbit BIT5 = BITBUF^5;
sbit BIT6 = BITBUF^6;
sbit BIT7 = BITBUF^7;
bit PRESSF;
bit PRESSG;

/***** FUNCTION *****/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void dmsec (unsigned int count) { /* Delay mSec */
    unsigned char i;
    while (count) {
        for (i=1;i<=228;i++);
        count--;
    }
}

void sound (unsigned char freq,int time) {
    unsigned char i;
    while (time>0) {
        SOUNDB = 0; /* out signal low */
        for(i=1;i<=freq;i++)
            time--;
        SOUNDB = 1; /* out signal high */
        for(i=1;i<=freq;i++)
            time--;
    }
}

void beeph (void) {
    sound (40,20000);
}

void beepL (void) {
    sound (150,30000);
}

void lcdwi (unsigned char com) { // LCD write instruction

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bit flag;
LCDWRC = com;
flag = 1;
while (flag) flag = LCDRDC & 0x80;
}

void lcdwd (unsigned char dat) { // LCD write data
    bit flag;
    LCDWRD = dat;
    flag = 1;
    while (flag) flag = LCDRDC & 0x80;
}

void lcdset (void) { // LCD start process */
    lcdwi (0x38); // function set */
    lcdwi (0x0c); // display on */
}

void lcdclear (void) { // clear LCD & LCDBUF */
    unsigned char i;
    lcdwi (0x01);
    for(i=0;i<=39;i++)
        LCDBUF[i]=' ';
}

void lcdout (void) { // load LCDBUF to LCD */
    unsigned char i,j;
    i = 0;
    lcdwi (0x80); // line-1 */
    for (j=1;j<=20;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    lcdwd (LCDBUF[i++]);
lcdwi (0xc0);          /* line-2 */
for (j=1;j<=20;j++)
    lcdwd (LCDBUF[i++]);
}

/* for I-400 */
void xlcdwi (unsigned char com) { /* LCD write instruction */
    XLCDRS = 0; XLCDRW = 0; XLCDEN = 1;
    XLCDDA = com;
    XLCDCK = 0; _nop_ (); _nop_ ();
    XLCDCK = 1; _nop_ (); _nop_ (); _nop_ ();
    XLCDCK = 0; _nop_ (); _nop_ ();
    XLCDEN = 0; _nop_ (); _nop_ ();
    XLCDEN = 1; _nop_ (); _nop_ (); _nop_ ();
    XLCDEN = 0; _nop_ (); _nop_ ();
    dmsec(3);
}

void xlcdwd (unsigned char dat) { /* LCD write data */
    XLCDRS = 1; XLCDRW = 0; XLCDEN = 1;
    XLCDDA = dat;
    XLCDCK = 0; _nop_ (); _nop_ ();
    XLCDCK = 1; _nop_ (); _nop_ (); _nop_ ();
    XLCDCK = 0; _nop_ (); _nop_ ();
    XLCDEN = 0; _nop_ (); _nop_ ();
    XLCDEN = 1; _nop_ (); _nop_ (); _nop_ ();
    XLCDEN = 0; _nop_ (); _nop_ ();
    dmsec(3);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void xlcdset (void) {          /* LCD start process */
    xlcdwi (0x38);           /* function set */
    xlcdwi (0x0c);          /* display on */
}

```

```

void xlcdclear (void) {      /* clear LCD & LCDBUF */
    unsigned char i;
    xlcdwi (0x01);
    for(i=0;i<=39;i++)
        LCDBUF[i]=' ';
}

```

```

void xlcdout (void) {        /* load LCDBUF to LCD */
    unsigned char i,j;
    i = 0;
    xlcdwi (0x80);          /* line-1 */
    for (j=1;j<=20;j++)
        xlcdwd (LCDBUF[i++]);
    xlcdwi (0xc0);          /* line-2 */
    for (j=1;j<=20;j++)
        xlcdwd (LCDBUF[i++]);
}

```

```

void ipdel (void) {         /* I2C delay */
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

_nop_ ();
_nop_ ();
_nop_ ();
}

```

```

void ipchigh (void) {          /* I2C clock high */
    IPSCL = 1;
    ipdel ();
}

```

```

void ipclow (void) {         /* I2C clock low */
    IPSCL = 0;
    ipdel ();
}

```

```

void ipstart (void) {       /* start condition */
    IPSDA = 1;
    IPSCL = 1;
    IPSDA = 0;
    ipdel ();
    IPSCL = 0;
    IPSDA = 1;
}

```

```

void ipstop (void) {       /* stop condition */
    IPSDA = 0;
    IPSCL = 1;
    ipdel ();
    IPSDA = 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bit ipwrbyte (unsigned dat) {      /* write one byte */
    unsigned char i;              /* return 0 = ok */
    bit outbit;                   /* return 1 = error */
    for (i=1;i<=8;i++) {
        outbit = dat & 0x80;
        IPSDA = outbit;
        dat = dat << 1;
        ipchigh ();
        ipclow ();
    }
    IPSDA = 1;
    ipchigh ();
    outbit = IPSDA;
    ipclow ();
    return (outbit);
}

unsigned char iprdbyte () {      /* read one byte */
    unsigned char i,dat;         /* return 0xff = error */
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++) {
        ipchigh ();
        inbit = IPSDA;
        dat = dat << 1;
        dat = dat | inbit;
        ipclow ();
    }
    IPSDA = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ipchigh ();
inbit = IPSDA;
ipclow ();
if (~inbit) dat = 0xff;
return (dat);
}

```

```

bit ipwr (unsigned char addr,dat) { /* write port */
    bit err;
    ipstart ();
    addr = ((addr << 1) & 0x0e) | 0x70;
    err = ipwrbyte (addr);
    if (~err) err = ipwrbyte (dat);
    ipstop ();
    return (err);
}

```

```

unsigned char iprd (unsigned char addr) { /* read port */
    unsigned char dat; /* return 0xff = error */
    bit err;
    ipstart ();
    addr = ((addr << 1) & 0x0e) | 0x71;
    err = ipwrbyte (addr);
    if (~err) dat = iprdbyte (); else dat = 0xff;
    ipstop ();
    return (dat);
}

```

```

bit ipda (unsigned char addr,dat) { /* write D/A */
    bit err;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ipstart ();
addr = ((addr << 1) & 0x0e) | 0x90;
err = ipwrbyte (addr);
if (~err) {
    err = ipwrbyte (0x40);          /* control byte */
    if (~err) err = ipwrbyte (dat);
}
ipstop ();
return (err);
}

unsigned char ipad (unsigned char addr,ch) { /* read A/D */
    unsigned char dat;          /* return 0xff = error */
    bit err;
    if (ch==3) ch = 0; else ch++;
    ipstart ();                /**** set control ****/
    addr = ((addr << 1) & 0x0e) | 0x90;
    err = ipwrbyte (addr);
    if (~err) ipwrbyte (0x40 | ch); /* control byte */
    ipstop ();
    dmsec (1);
    ipstart ();                /**** read a/d ****/
    addr = addr | 0x1;          /* change r/w */
    err = ipwrbyte (addr);
    if (~err) dat = iprdbyte (); else dat = 0xff;
    ipstop ();
    dmsec (1);
    return (dat);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void irchigh (void) {          /* I2C clock high for ds1307 */
    IRSCL = 1;
    ipdel ();
}
```

```
void irclow (void) {         /* I2C clock low for ds1307 */
    IRSCL = 0;
    ipdel ();
}
```

```
void irstart (void) {       /* start condition for ds1307 */
    IRSDA = 1;
    IRSCL = 1;
    IRSDA = 0;
    ipdel ();
    IRSCL = 0;
    IRSDA = 1;
}
```

```
void irstop (void) {       /* stop condition for ds1307 */
    IRSDA = 0;
    IRSCL = 1;
    ipdel ();
    IRSDA = 1;
}
```

```
bit irwrbyte (unsigned dat) { /* write one byte for ds1307 */
    unsigned char i;          /* return 0 = ok */
    bit outbit;              /* return 1 = error */
    for (i=1;i<=8;i++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outbit = dat & 0x80;
    IRSDA = outbit;
    dat = dat << 1;
    irchigh ();
    irclow ();
}
IRSDA = 1;
irchigh ();
outbit = IRSDA;
irclow ();
return (outbit);
}

unsigned char irrdbyte () { /* read one byte for ds1307 */
    unsigned char i,dat; /* return 0xff = error */
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++) {
        irchigh ();
        inbit = IRSDA;
        dat = dat << 1;
        dat = dat | inbit;
        irclow ();
    }
    IRSDA = 1;
    irchigh ();
    inbit = IRSDA;
    irclow ();
    if (~inbit) dat = 0xff;
    return (dat);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

unsigned char irrdbytex () { /* read one byte ack=0 for DS1307 */
    unsigned char i,dat; /* return 0xff = error */
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++) {
        irchigh ();
        inbit = IRSDA;
        dat = dat << 1;
        dat = dat | inbit;
        irclow ();
    }
    IRSDA = 0;
    irchigh ();
    irclow ();
    IRSDA = 1;
    return (dat);
}

void rtset (void) { /* set RTC status bit */
    TIMBUF[0] = TIMBUF[0] & 0x7f; /* clear CH (clock halt) */
    TIMBUF[2] = TIMBUF[2] & 0xbf; /* set 24 hour */
    TIMBUF[7] = TIMBUF[7] & 0xec; /* b7=out b4=sqwe b1,b0=rs */
}

bit rtwr (void) { /* ds1307 rtc write */
    unsigned char i; /* return 0=ok 1=error */
    bit err;
    irstart ();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

err = irwrbyte (0xd0);      /* address & r/w bit */
if (~err) {
    err = irwrbyte (0x00);  /* start register addr=0 */
    if (~err) {
        for (i=0;i<=7;i++) {
            err = irwrbyte (TIMBUF[i]);
            if (err) break;
        }
    }
}
irstop ();
return (err);
}

```

```

bit rtrd (void) {          /* ds1307 rtc read */
    unsigned char i,dat;   /* return 0=ok 1=error */
    bit err;
    irstart ();
    err = irwrbyte (0xd0); /* address & r/w bit */
    if (~err) {
        err = irwrbyte (0x00); /* start register addr=0 */
        if (~err) {
            irstop ();
            ipdel ();
            irstart ();
            err = irwrbyte (0xd1); /* address & r/w bit */
            if (~err) {
                for (i=0;i<=6;i++) {
                    dat = irrdbytex ();
                    if (dat==0xff) break;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        TIMBUF[i] = dat;
    }
    dat = irrdbyte (); /* last byte */
    if (dat==0xff) err = 1;
    else TIMBUF[7] = dat;
}
}
}
irstop ();
return (err);
}

unsigned int autobaud (void) { /* auto RS232 baud-rate */
    unsigned int speed; /* start by SPACE character */
    /* auto 1200,2400,4800,9600,19200,38400 */
    TMOD = 0x10; /* timer-1 mode-1 16 bit counter */
    TH1 = 0x00;
    TL1 = 0x00;
    TR1 = 0;
    while (RXD); /* wait for first start bit */
    TR1 = 1; /* start count */
    while (~RXD);
    TR1 = 0; /* stop count */
    speed = TH1;
    speed <<= 8;
    speed |= TL1; /* speed = raw data */
    dmsec (100);

    SCON = 0x52; /* set RS232 parameter */
    TMOD = 0x20;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (speed>7050) {
    speed = 1200;
    SPEEDM = 12;
    TH1 = 0xd0;
}
else if (speed>3500) {
    speed = 2400;
    SPEEDM = 24;
    TH1 = 0xe8;
}
else if (speed>1850) {
    speed = 4800;
    SPEEDM = 48;
    TH1 = 0xf4;
}
else if (speed>860) {
    speed = 9600;
    SPEEDM = 96;
    TH1 = 0xfa;
}
else if (speed>430) {
    speed = 19200;
    SPEEDM = 19;
    TH1 = 0xfd;
}
else {
    speed = 38400;
    SPEEDM = 38;
    TH1 = 0xfd;
    PCON |= 0x80;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
TR1 = 1;
return (speed);
}

```

```

void setbaud (unsigned char speed) { /* set RS232 baud rate */
    SCON = 0x52;
    TMOD = 0x20;
    switch (speed) {
        case 12: TH1 = 0xd0; break;
        case 24: TH1 = 0xe8; break;
        case 48: TH1 = 0xf4; break;
        case 96: TH1 = 0xfa; break;
        case 192: TH1 = 0xfd; break;
        case 384: TH1 = 0xfd; PCON |= 0x80; break;
    }
    TR1 = 1;
}

```

```

unsigned char scan (void) { /* scan display & get key */
    unsigned char i,j,x;
    PRESSG = 0;
    for (i=0;i<=7;i++) {
        ONBB = (ONBB & 0xf8) | i;
        ONBA = DISBUF[i];
        dmsec(1);
        ONBA = 0;
        x = ONBC & 0x0f;
        if (x!=0x0f) {
            PRESSG = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (~PRESSF) {
        x = (x << 4) | i;
        PRESSF = 1;
        return(x);
    }
}

if (~PRESSG) PRESSF = 0;    /* No any key press , set PRESSF */
return(0xff);
}

unsigned char scanx (void) {    /* scan display & get key [374,245] */
    unsigned char i,j,x;
    PRESSG = 0;
    for (i=0;i<=7;i++) {
        XBYTE[0xf000] = i | 0xf8;
        XBYTE[0xd000] = DISBUF[i];
        dmsec(1);
        XBYTE[0xd000] = 0;
        x = XBYTE[0xe000] & 0x0f;
        if (x!=0x0f) {
            PRESSG = 1;
            if (~PRESSF) {
                x = (x << 4) | i;
                PRESSF = 1;
                return(x);
            }
        }
    }
}

if (~PRESSG) PRESSF = 0;    /* No any key press , set PRESSF */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(0xff);
}

void scany (void) {          /* scan display [P0,P2] */
    unsigned char i,x;      /* no key */
    for (i=0;i<=7;i++) {
        P2 = (P2 & 0xf0) | i;
        P0 = DISBUF[i];
        XLCDCK = 0; XLCDCK = 1; XLCDCK = 0;
        dmsec(1);
        P0 = 0;
        XLCDCK = 0; XLCDCK = 1; XLCDCK = 0;
    }
}

unsigned char code SEGTAB[16] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,
                                0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71};
    /* standard segment code */

unsigned char htosx (unsigned char hex) { /* change hex to segment */
    return (SEGTAB[hex]);
}

void ramtest (unsigned int start,end) {
    unsigned int run,block;
    unsigned char xdata *addr;
    unsigned char a,b;
    bit err;
    run = start;          /* write memory */
    a = 0x11;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (run<=end) {
    addr = run;
    *addr = a;
    a++;
    if (a==0x9a) a = 0x11;
    run++;
}

run = start;          /* read & check */
block = run;
a = 0x11;
err = 0;
while (run<=end) {
    addr = run;
    b = *addr;
    if (b!=a) err = 1;
    *addr = 0;        /* clear ram */
    a++;
    if (a==0x9a) {
        a = 0x11;
        if (err) printf ("%4X Error\n",block);
        else printf ("%4X\n",block);
        err = 0;
        block = run;
    }
    run++;
}
}

```

```

void start (void) {    /* start process */
    unsigned char i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned int speed,c,addr;

bit countflag;

speed = autobaud ();

beeph ();

countflag = 0;

c = 0;

for (addr=0x1fff;addr>0;addr--) { // find checksum (auto 8K)

    if (countflag==0 && CBYTE[addr]!=0xff) countflag = 1;

    if (countflag) {

        c = c + CBYTE[addr];

    }

}

c.= c + CBYTE[0];

for (i=1;i<=20;i++)

    printf ("\n");

printf ("ISPSQC Super QC Program v2.0 (checksum=%4X) ...\n",c);

printf ("2001 Sila Research Co.,Ltd. ... Speed = %u",speed);

}

void menu1 (void) {

    unsigned char i,j,k;

    printf ("\n\nTesting 24B Port [8255] with EX-LED ...");

    USRP = 0x80;

    for (k=1;k<=3;k++) {

        j = 1;

        for (i=1;i<=8;i++) {

            USRA = j;

            USRB = j;

            USRC = j;

            dmsec (400);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    j = j << 1;
  }
}
for (i=1;i<=3;i++) {
  j = 0x55;
  USRA = j;
  USRB = j;
  USRC = j;
  dmsec (600);
  j = 0xaa;
  USRA = j;
  USRB = j;
  USRC = j;
  dmsec (600);
}
j = 0;
USRA = j;
USRB = j;
USRC = j;
beeph ();
}

```

```

void menu2 (void) {
  unsigned char a,i,j,k;
  printf ("\n\nTesting 8x8 Display and 4x8,3x4 Key ...");
  ONBP = 0x89;
  for(i=1;i<=3;i++) {
    a = 1;
    for (j=1;j<=8;j++) {
      for (k=0;k<=7;k++) DISBUF[k] = a;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

k = 60;
while(k) {
    scan (); k--;
}
a = a << 1;
}
}
printf ("\nType any keys to exit");
for (i=0;i<=7;i++) DISBUF[i] = 0x3f;
while (~RI) {
    i = scan ();
    if (i != 0xff) {
        DISBUF[0] = htosx ((i & 0xf0) >> 4);
        DISBUF[1] = htosx (i & 0x0f);
        DISBUF[2] = DISBUF[0];
        DISBUF[3] = DISBUF[1];
        DISBUF[4] = DISBUF[0];
        DISBUF[5] = DISBUF[1];
        DISBUF[6] = DISBUF[0];
        DISBUF[7] = DISBUF[1];
    }
}
getkey ();
}

```

```

void menu3 (void) {
    unsigned char a,i;

    /* backlight on */

    /* ISP-2000 */

    ONBP = 0x88;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ONBB = 0x00;

        /* ISP-1000 */

EN374 = 0;          /* enable 74LS374 */

XBYTE[0xf000] = 0xf0;

printf ("\n\nTesting LCD Port ...");

lcdset ();

lcdclear ();

dmsec (1000);

for (a='0';a<='9';a++) {
    for (i=0;i<=39;i++) LCDBUF[i] = a;
    lcdout ();
    dmsec (400);
}

for (i=0;i<=39;i++) LCDBUF[i] = 0xff;
lcdout ();
beeph ();

        /* backlight off */

ONBB = 0x08;
XBYTE[0xf000] = 0xf8;
}

void menu4 (void) {
    char c;

    printf ("\n\nTesting Dip-SW & User LED ...");
    printf ("\nType any keys to exit");

    ONBP = 0x88;

    while (~RI) {
        c = ~ONBC;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    c &= 0xf0;
    ONBB = c;
}
getkey ();
ONBB = 0x00;
}

```

```

void menu5 (void) {
    unsigned char a;
    printf("\n\nTesting Opto-IN & Relat-OUT ...");
    printf("\nType any keys to exit");
    while (~RI) {
        BITBUF = iprd (1);
        BIT2 = BIT0;
        BIT3 = BIT1;
        BIT0 = 1;
        BIT1 = 1;
        ipwr (1,BITBUF);
    }
    getkey ();
}

```

```

void menu6 (void) {
    unsigned char a,i,j,k;
    printf("\n\nTesting 12B Port with EX-LED ...");
    for (k=1;k<=3;k++) {
        j = 1;
        for (i=1;i<=8;i++) {
            a = j | 0x0f;
            ipwr (0,j);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ipwr (1,a);
    dmsec (400);
    j = j << 1;
}
}
for (i=1;i<=3;i++) {
    j = 0x55;
    a = j | 0x0f;
    ipwr (0,j);
    ipwr (1,a);
    dmsec (600);
    j = 0xaa;
    a = j | 0x0f;
    ipwr (0,j);
    ipwr (1,a);
    dmsec (600);
}
ipwr (0,0x0);
ipwr (1,0x0f);
beeph ();
}

```

```

void menu7 (void) {
    unsigned char a;
    printf ("\n\nTesting D/A Port with METER ...");
    printf ("\nVolt step up every 0.5 second");
    printf ("\nType any keys");
    a = 0;
    while (~RI) {
        ipda (0,a);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    dmsec (500);

    a++;
}

getkey ();

printf ("\n\nTesting A/D Port with VR ...");
printf ("\nType any keys to exit");
while (~RI) {
    ATDBUF[0] = ipad (0,0);
    ATDBUF[1] = ipad (0,1);
    ATDBUF[2] = ipad (0,2);
    ATDBUF[3] = ipad (0,3);
    printf ("\nA/D channel 0-3 ... %3bu %3bu %3bu %3bu"
,ATDBUF[0],ATDBUF[1],ATDBUF[2],ATDBUF[3]);
    dmsec (200);
}
getkey ();
}

void menu8 (void) {
    unsigned char x;
    bit err;
    printf ("\n\nTesting RTC DS1307 ...");
    TIMBUF[0] = 0x00;    /* second */
    TIMBUF[1] = 0x44;    /* minute */
    TIMBUF[2] = 0x00;    /* hour */
    TIMBUF[4] = 0x27;    /* date */
    TIMBUF[5] = 0x03;    /* month */
    TIMBUF[6] = 0x01;    /* year */

    rtset ();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rtwr ();
while (TIMBUF[0]!=0x05) {
    dmsec (100);
    err = rtrd ();
    if (err) beeph ();
    if (TIMBUF[0]!=x) {
        printf ("time %2bx:%02bx:%02bx date %2bx/%02bx/%02bx\n",
            TIMBUF[2],TIMBUF[1],TIMBUF[0],TIMBUF[4],TIMBUF[5],TIMBUF[6]);
        x = TIMBUF[0];
    }
}
}

void menu9 (void) {
    printf ("\n\nTesting RAM 0000-7FFF ...\n");
    ramtest (0x0300,0x7fff);
}

void menua (void) {
    unsigned char i;
    printf ("\n\nTesting Watch Dog ...\n");
    printf ("\nEnable WD jumper ...");
    for (i='9';i>='0';i--) {
        printf ("%c",i);
        WDBIT = ~WDBIT;
        dmsec (400);
        WDBIT = ~WDBIT;
        dmsec (400);
    }
    printf ("\nWait for Reset !");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RESVEC = 0xa5;
while (1);
}

```

```

void menuax (void) {
    unsigned char i;
    RESVEC = 0;
    setbaud (SPEEDM);
    beeph ();
    printf ("\nReset OK , Disable WD Jumper ...");
    for (i='9';i>='0';i--) {
        printf ("%c",i);
        WDBIT = ~WDBIT;
        dmsec (400);
        WDBIT = ~WDBIT;
        dmsec (400);
    }
    printf ("\n");
}

```

```

void menub (void) {
    unsigned char a;
    printf ("\n\nTesting 3B Port with Logic probe ...");
    printf ("\nType any keys to exit");
    a = 0;
    while (~RI) {
        BITBUF = a;
        PBIT0 = BIT0;
        PBIT1 = BIT1;
        PBIT2 = BIT2;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    dmsec (200);
    a++;
}
getkey ();
PBIT0 = 1;
PBIT1 = 1;
PBIT2 = 1;
}

void outabc (unsigned char x) {
    XBYTE[0xa000] = x;
    XBYTE[0xb000] = x;
    XBYTE[0xc000] = x;
}

void menuc (void) {
    unsigned char i,j,k;
    outabc (0);          /* clear output first */
    EN374 = 0;          /* enable 74LS374 */

    printf ("\n\nTesting 24B Port [374] with EX-LED ...");
    for (k=1;k<=3;k++) {
        j = 1;
        for (i=1;i<=8;i++) {
            outabc (j);
            dmsec (400);
            j = j << 1;
        }
    }

    for (i=1;i<=3;i++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

j = 0x55;
outabc (j);
dmsec (600);
j = 0xaa;
outabc (j);
dmsec (600);
}
j = 0;
outabc (j);
beeph ();
}

void menud (void) {
    unsigned char a,i,j,k;
    EN374 = 0; /* enable 74LS374 */

    printf ("\n\nTesting 8x8 Display and 4x8 Key [374,245] ...");
    for(i=1;i<=3;i++) {
        a = 1;
        for (j=1;j<=8;j++) {
            for (k=0;k<=7;k++) DISBUF[k] = a;
            k = 60;
            while(k) {
                scanx (); k--;
            }
            a = a << 1;
        }
    }

    printf ("\nType any keys to exit");
    for (i=0;i<=7;i++) DISBUF[i] = 0x3f;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (~RI) {
    i = scanx ();
    if (i != 0xff) {
        DISBUF[0] = htosx ((i & 0xf0) >> 4);
        DISBUF[1] = htosx (i & 0x0f);
        DISBUF[2] = DISBUF[0];
        DISBUF[3] = DISBUF[1];
        DISBUF[4] = DISBUF[0];
        DISBUF[5] = DISBUF[1];
        DISBUF[6] = DISBUF[0];
        DISBUF[7] = DISBUF[1];
    }
}
getkey ();
}

void menue (void) {
    char c;
    EN374 = 0; /* enable 74LS374 */

    printf ("\n\nTesting Dip-SW & User LED [374,245] ...");
    printf ("\nType any keys to exit");
    while (~RI) {
        c = ~XBYTE[0xe000];
        c &= 0xf0;
        XBYTE[0xf000] = c;
    }
    getkey ();
    XBYTE[0xf000] = 0xff;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void menuf (void) {
    unsigned int i;
    printf ("\n\nTesting Tool I/O & 485 & Low Batt ...");
    printf ("\nif High Batt ... Tool I/O & 485 = Slow Pulse");
    printf ("\nif Low Batt ... Tool I/O & 485 = Fast Pulse");
    printf ("\nType any keys to exit");
    while (~RI) {
        if (LOWB) i = 500; else i = 200;
        TOOL = 0;
        D485 = 0;
        dmsec (i);
        TOOL = 1;
        D485 = 1;
        dmsec (i);
    }
    getkey ();
}

```

```

void menug (void) {
    unsigned char a,i;
    printf ("\n\nTesting LCD Port ...");
    XLCDBL = 0;
    xlcdset ();
    xlcdclear ();
    dmsec (1000);
    for (a='0';a<='9';a++) {
        for (i=0;i<=39;i++) LCDBUF[i] = a;
        xlcdout ();
        dmsec (400);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
for (i=0;i<=39;i++) LCDBUF[i] = 0xff;
xlcdout ();
beeph ();
XLCDBL = 1;
}

void menuh (void) {
    unsigned char a,i,j,k;
    printf ("\n\nTesting 8x8 Display [P0,P2] ...");
    for(i=1;i<=3;i++) {
        a = 1;
        for (j=1;j<=8;j++) {
            for (k=0;k<=7;k++) DISBUF[k] = a;
            k = 60;
            while(k) {
                scany (); k--;
            }
            a = a << 1;
        }
    }
    P2 = P2 | 0xf;           // No digit select
}

```

```

void menui (void) {
    unsigned char a;
    printf ("\n\nTesting 3B Port (2) with Logic probe ...");
    printf ("\nType any keys to exit");
    a = 0;
    while (~RI) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    BITBUF = a;
    PBIT20 = BIT0;
    PBIT21 = BIT1;
    PBIT22 = BIT2;
    dmsec (200);
    a++;
}

getkey ();
PBIT20 = 1;
PBIT21 = 1;
PBIT22 = 1;
}

void test () {
    unsigned int run,i;
    unsigned char xdata *addr;
    unsigned char data;
    bit err;

    run = 0x1ff8;
    addr = run;
    *addr = 0x80;
    run++;

    for (i=0x1ff9;i<=0x1fff;i++)
    {
        addr = i;
        data = *addr;
        printf ("%4X\n",data);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** MAIN *****/

void main (void) {
    unsigned char i;
    bit dflag;
    if (RESVEC==0xa5) menuax ();    /* menua recover */
    else start ();
        dflag = 1;
    while (1) {
        if (dflag) {
            printf ("\n\n");
            printf (" Main menu ...      ISP- 2000 1000 400\n");
            printf ("1. Test 24B Port [8255]          x \n");
            printf ("2. Test 8x8D 4x8K 3x4K [8255]      x \n");
            printf ("3. Test LCD Port (with Backlight) x  x \n");
            printf ("4. Test Dip-SW & LED(1) [8255]    x \n");
            printf ("5. Test Opto(Key) & Relay(LED)   x  x  x \n");
            printf ("6. Test 12B Port                   x  x  x \n");
            printf ("7. Test 4AD/1DA Port                x  x  x \n");
            printf ("8. Test RTC DS1307                  x  x  x \n");
            printf ("9. Test RAM 0000-7FFF                x  x \n");
            printf ("A. Test WD [1232]                   x \n");
            printf ("B. Test 3B Port                      x  x  x \n");
            printf ("C. Test 24B Port [374]                x \n");
            printf ("D. Test 8x8D 4x8K [374,245]          x \n");
            printf ("E. Test Dip-SW & LED(0) [374,245]    x \n");
            printf ("F. Test Tool I/O & 485 & Low Batt   x  x \n");
            printf ("G. Test LCD Port [P0+P2]             x \n");
            printf ("H. Test 8x8D [P0+P2]                 x \n");
            printf ("I. Test 3B Port (2)                  x \n");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf ("\nSelect (Test Sound) :");
}
dflag = 1;
i = getkey ();
if (i=='1') menu1 ();
else if (i=='2') menu2 ();
else if (i=='3') menu3 ();
else if (i=='4') menu4 ();
else if (i=='5') menu5 ();
else if (i=='6') menu6 ();
else if (i=='7') menu7 ();
else if (i=='8') menu8 ();
else if (i=='9') test ();
else if (i=='a' || i=='A') menua ();
else if (i=='b' || i=='B') menub ();
else if (i=='c' || i=='C') menuc ();
else if (i=='d' || i=='D') menud ();
else if (i=='e' || i=='E') menue ();
else if (i=='f' || i=='F') menuf ();
else if (i=='g' || i=='G') menug ();
else if (i=='h' || i=='H') menuh ();
else if (i=='i' || i=='I') menui ();
else { beep1 (); dflag = 0; }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Spec Industrial Data Logger

1. Input Signal Channel 1 (0-4 mV = 0-100 °C)
2. Input Signal Channel 2-4 (0-2.5 V)
3. Sampling Rate 15 seconds
4. Power Supply ± 5 V
5. Time Period 6 Hr. (1,441 measurement points)

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดี เนื่องจากความร่วมมือระหว่างภาควิชาวิศวกรรมระบบควบคุมและภาควิชาวิศวกรรมอาหาร และได้รับความอนุเคราะห์จาก ผศ.พรสุข รัตโรจน์อนันต์ (อาจารย์ที่ปรึกษาปริญญาานิพนธ์) และอาจารย์ทุกท่านในภาคที่ได้ให้คำแนะนำ ข้อคิด ตลอดจนการเอื้ออำนวยความสะดวกต่าง ๆ ในการทดลองโครงการครั้งนี้ และเพื่อนๆ ที่คอยช่วยเหลือและเป็นกำลังใจที่ดีมาโดยตลอด จึงถือโอกาสขอขอบคุณไว้ ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] รัชชัย อินทุโสและไทรภพ อินทุโส . ไมโครคอนโทรลเลอร์ 8051 . กรุงเทพฯ : ฟิสิกส์เซนเตอร์, 174หน้า.
- [2] สมยศ จุณณะปิยะ . การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 , พิมพ์ครั้งที่ 2 . กรุงเทพฯ , 2541.
- [3] วรพล ลีลาเกียรติสกุล,มารุต ตามศรีและพัฒนาพล สายกลั่น . C51LAB C Language Laboratory .กรุงเทพฯ : ศิลารี่เสิร์ช ,81หน้า.
- [4] ชาริน ลีทธิธรรมชารี และ ธนัญชัย จำนงศักดิ์ . Microsoft Visual Basic version 5.0 ,พิมพ์ครั้งที่ 2 . กรุงเทพฯ : :ซัคเซส มีเดีย ,427หน้า.
- [5] ชันวา ศรีประโมง . การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม , พิมพ์ครั้งที่สี่(เพิ่มเติม) . กรุงเทพฯ , 2539 .

