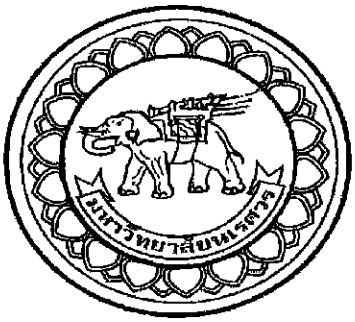


การวัดและแสดงค่าของสัญญาณไฟฟ้าทางคอมพิวเตอร์

Measure and display value of signal by computer

นายบุญชวน ปวนไชยดี รหัส 45380070

นายวสันต์ นันทะสี รหัส 45380107

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 W.A. 2553
เลขทะเบียน 150 5054
เลขเรียกหนังสือ วิ5
มหาวิทยาลัยนเรศวร 24220

2549

ค.ว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การวัดและแสดงค่าของสัญญาณไฟฟ้าทางคอมพิวเตอร์
ผู้ดำเนินโครงงาน	นายบุญชวน ปวนไชยดี รหัส 45380070
	นายวสันต์ นันทะสี รหัส 45380107
อาจารย์ที่ปรึกษา	ดร.สมพร เรืองสินชัยวานิช
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2549

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.สมพร เรืองสินชัยวานิช)

.....กรรมการ

(ดร.สมยศ เกียรติวนิชวิไล)

.....กรรมการ

(อาจารย์ปิยดนัย ภาชนะพรรณ์)

หัวข้อโครงงาน	การวัดและแสดงค่าของสัญญาณไฟฟ้าทางคอมพิวเตอร์		
ผู้ดำเนินโครงงาน	นายบุญชวน	ปวนไชยดี	รหัส 45380070
	นายวสันต์	นันทะสี	รหัส 45380107
อาจารย์ที่ปรึกษา	ดร.สมพร	เรืองสินชัยวานิช	
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2549		

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาเครื่องวัดสัญญาณไฟฟ้า ที่สามารถแสดงผลจากการวัดได้ตามเวลาจริง(real time)และเก็บบันทึกข้อมูลที่วัดได้ ไว้ด้วยคอมพิวเตอร์

ผลที่ได้จากการทำโครงงานนี้คือ ได้เครื่องวัดสัญญาณไฟฟ้าที่สามารถแสดงผลการวัดสัญญาณและบันทึกผลข้อมูลที่ได้ เพื่อนำไปประยุกต์ใช้ในงานจริงต่อไป

Project title	Measure and display value of signal by computer
Name	Mr.Boonchuan Puanchaidee ID. 45380070
	Mr.Wasan Nantasee ID. 45380107
Project Advisor	Dr.Somporn Ruangsinchaiwanich
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2006

Abstract

This project, we study to develop the meter which is a device used to measure and show that in real time. And recording measure data into computer.

The result of this project is all measurable electrical properties, the meter that could be showing. The fact our project to continuing applied in near future.

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร.สมพร เรืองสินชัยวานิช อาจารย์ที่ปรึกษา ที่คอยให้คำปรึกษาและให้ความช่วยเหลือตลอดจนคำแนะนำต่างๆ ในการทำโครงงานชิ้นนี้

สุดท้ายต้องขอขอบพระคุณคุณพ่อ และ คุณแม่ อาจารย์ทุกท่านและเพื่อนๆ พี่ ๆทุกคนที่ยังไม่ได้เอ่ยนามที่ให้คำแนะนำและให้การสนับสนุนผู้จัดทำโครงงานให้สามารถทำโครงงานชิ้นนี้จนสำเร็จลุล่วงไปได้ด้วยดี

นายบุญชวน ปวนไชยดี

นายวสันต์ นันทะสี

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ 1

1.3 ขอบข่ายโครงงาน 1

1.4 กิจกรรมดำเนินการ 2

1.5 ผลที่คาดว่าจะได้รับ 2

1.6 งบประมาณ 2

บทที่ 2 หลักการและทฤษฎี

2.1 ลักษณะสัญญาณในการวัดและแสดงค่าของสัญญาณทางคอมพิวเตอร์ 3

2.2 วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 6

2.3 ข้อพิจารณาเลือกใช้งานวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 8

2.4 การเลือกวงจรปรับสภาพสัญญาณ 11

2.5 ข้อมูลของ PCF8591 ไอซี ADC/DAC 12

บทที่ 3 วิธีการออกแบบ

3.1 มอเตอร์ดีซี 18

3.2 วงจรปรับสภาพสัญญาณ 18

3.3 ส่วนฮาร์ดแวร์เพื่อการวัดสัญญาณและเชื่อมต่อกับคอมพิวเตอร์ 20

3.4 โปรแกรมสำหรับการแสดงค่าของสัญญาณไฟฟ้าทางคอมพิวเตอร์ 23

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง

4.1 จุดประสงค์การทดลอง....27

4.2 ขั้นตอนการทดลอง....27

4.3 ผลการทดลอง....27

4.4 วิเคราะห์ผลการทดลอง....28

บทที่ 5 สรุปผลและการวิเคราะห์ปัญหาในการทดลอง

5.1 สรุปผลการทดลอง....29

5.2 ปัญหาในการทำงานและแนวทางแก้ไข....29

5.3 ข้อเสนอแนะ....29

5.4 แนวทางสำหรับการพัฒนา....29

เอกสารอ้างอิง....30

ภาคผนวก (ก) Source code ของโปรแกรม....32

ภาคผนวก (ข) ตัวอย่างการสร้างมิเตอร์....38

ประวัติผู้เขียนโครงงาน....44

สารบัญตาราง

ตารางที่		หน้า
4.1	ผลการวัดค่าแรงดันไฟฟ้า	28
4.2	ผลการวัดค่ากระแสไฟฟ้า	28

สารบัญรูป

รูปที่		หน้า
2.1	กระบวนการทำงานกับสัญญาณสถานะดิจิตอลในระบบ	3
2.2	กระบวนการทำงานกับสัญญาณสถานะดิจิตอลแบบขบวนพัลส์ในระบบ	4
2.3	กระบวนการทำงานกับสัญญาณไฟตรงอะนาลอกในระบบ	5
2.4	กระบวนการทำงานกับสัญญาณอะนาลอกในเชิงเวลาของระบบ	5
2.5	กระบวนการทำงานกับสัญญาณอะนาลอกในเชิงความถี่ของระบบ	6
2.6	กราฟแสดงความสัมพันธ์ระหว่างแรงดันอินพุตกับข้อมูลดิจิตอลเอาต์พุต	7
2.7	แสดงผลของการสุ่มสัญญาณด้วยอัตราที่แตกต่างกัน	8
2.8	แสดงผลของการวัดเมื่อเครื่องมือวัดมีข้อจำกัดด้านแบนด์วิดธ์	9
2.9	แสดงจังหวะของการสุ่มสัญญาณแบบต่างๆ ที่ใช้ในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล	10
2.10	แสดงตัวอย่างของการเลือกใช้วงจรปรับสภาพสัญญาณให้เข้ากับตัวตรวจจับ	12
2.11	การจัดขาและตารางแสดงชื่อขาสัญญาณของ PCF8591	13
2.12	แสดงบิตข้อมูลควบคุม	15
3.1	ไดอะแกรมการทำงานเบื้องต้นของระบบ	17
3.2	มอเตอร์ไฟฟ้ากระแสตรง	18
3.3	วงจรปรับสภาพแรงดันไฟฟ้า	19
3.4	วงจรปรับสภาพกระแสไฟฟ้า	20
3.5	แสดงส่วนประกอบของบอร์ด PDX-800	21
3.6	วงจรสมบูรณ์ของบอร์ด PDX-800	22
3.7	การเพิ่มคอนโทรล PDX800OCX ให้กับ VISUAL BASIC	23
3.8	รูปแบบการใช้งาน	26

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันคอมพิวเตอร์ได้กลายเป็นเทคโนโลยีที่ผู้คนให้ความนิยมมากที่สุด ซึ่งนับวันได้พัฒนาไปได้รวดเร็ว ทั้งทางด้านอุปกรณ์ฮาร์ดแวร์ และซอฟต์แวร์

การนำเอาคอมพิวเตอร์มาประยุกต์ใช้ในการวัดและแสดงผลอัตโนมัติเพื่องานในเชิงวิศวกรรมนั้น ด้วยความสามารถของคอมพิวเตอร์ในปัจจุบัน ประกอบกับการใช้งานที่ง่ายขึ้นของซอฟต์แวร์ระบบปฏิบัติการในลักษณะที่เป็นวินโดวส์หรือเป็นกราฟฟิก ทำให้การประยุกต์เพื่อนำคอมพิวเตอร์มาใช้ในงานด้านนี้มีความเป็นไปได้อย่างไม่ยุ่งยาก และในส่วนของฮาร์ดแวร์ที่เชื่อมต่อกับคอมพิวเตอร์อาจกำหนดให้ทำหน้าที่เป็นตัวรับค่าสัญญาณและแปลงสัญญาณจากอะนาลอกเป็นสัญญาณดิจิตอลเพื่อให้ซอฟต์แวร์ ทำการเก็บรวบรวมข้อมูลจากอุปกรณ์หรือฮาร์ดแวร์มาเก็บไว้เป็นฐานข้อมูล แล้วทำการแสดงผลข้อมูลที่ได้รับตามเวลาจริง (Real Time) ที่หน้าจอคอมพิวเตอร์ โดยแสดงผลในลักษณะเป็นมิเตอร์วัดค่าหรือภาพกราฟฟิก

1.2 วัตถุประสงค์

1. เพื่อศึกษาการเขียนโปรแกรมเกี่ยวกับการเชื่อมต่อระหว่างคอมพิวเตอร์กับฮาร์ดแวร์และการแสดงลักษณะของสัญญาณผ่านทางคอมพิวเตอร์
2. เพื่อพัฒนาเครื่องมือวัดสัญญาณไฟฟ้าให้มีราคาถูกลงและมีประสิทธิภาพใกล้เคียง
3. เพื่อศึกษาเกี่ยวกับวงจรและการออกแบบเครื่องมือวัด

1.3 ขอบข่ายโครงงาน

โครงงานนี้เป็นการวัดและแสดงลักษณะสัญญาณไฟฟ้า โดยใช้บอร์ดทดลอง แปลงสัญญาณอะนาลอกเป็นสัญญาณดิจิตอล แล้วส่งข้อมูลที่ได้ทำการแปลงและประมวลผลไปยังคอมพิวเตอร์เพื่อให้ซอฟแวร์ที่เขียนขึ้นมาแสดงลักษณะสัญญาณที่ได้จากการวัด ซึ่งมีหัวข้อหลัก ที่จะทำการศึกษาเพื่อสร้างชิ้นงาน ดังนี้

1. ศึกษาการทำงานของบอร์ดทดลอง PDX-800 และการเขียนโปรแกรมวิชวลเบสิค
2. ออกแบบเครื่องวัดสัญญาณ ที่ใช้เชื่อมต่อกับคอมพิวเตอร์
3. เขียนโปรแกรมเพื่อใช้สำหรับวัดและแสดงลักษณะของสัญญาณไฟฟ้าทางคอมพิวเตอร์

1.4 กิจกรรมดำเนินการ

กิจกรรม	ปี 2549		ปี 2550				
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. เขียนโครงการทำงาน	←						
2. รวบรวมข้อมูลและเอกสาร	←						
3. วิเคราะห์และออกแบบชิ้นงาน		←	→				
4. สร้างและทดสอบชิ้นงาน		←	→	→			
5. ปรับปรุงและแก้ไขชิ้นงาน					←	→	
6. จัดทำเอกสารและคู่มือการใช้งาน							←

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถออกแบบ และสร้างระบบวัดและแสดงลักษณะสัญญาณไฟฟ้า ซึ่งใช้งานได้จริง
2. สามารถนำความรู้จากการศึกษาบอร์ดทดลอง ที่ใช้ในโครงงานไปประยุกต์ใช้งานอื่นๆ ได้

1.6 งบประมาณ

1. ค่าวัสดุคอมพิวเตอร์	1,500	บาท
2. ค่าถ่ายเอกสาร งานพิมพ์	300	บาท
3. ค่าวัสดุอื่น ๆ	200	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท (สองพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

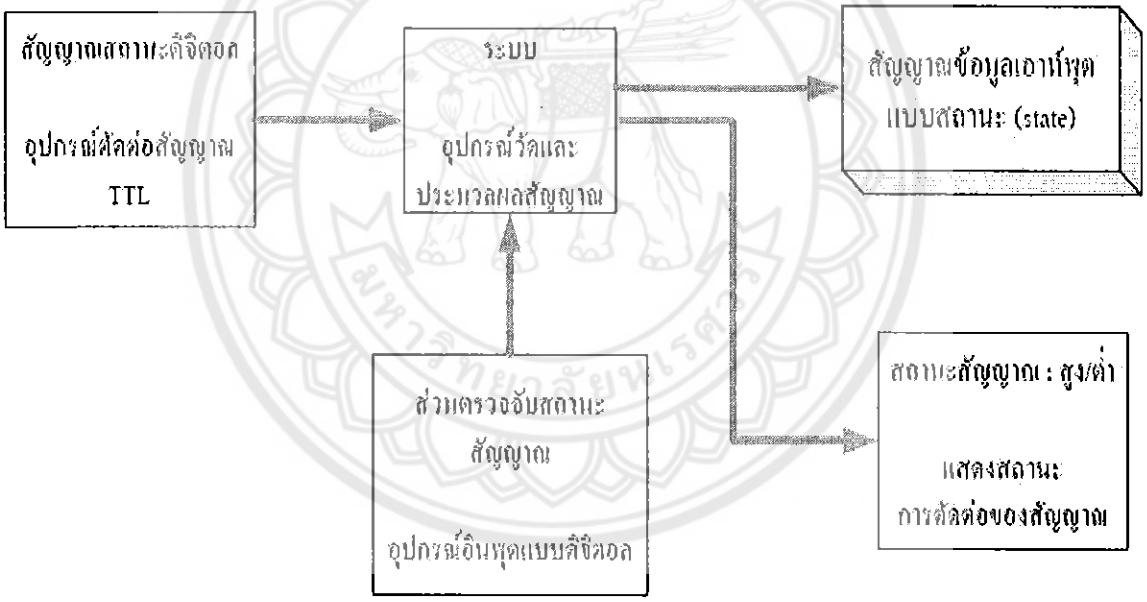
บทที่ 2
หลักการและทฤษฎี

2.1 ลักษณะสัญญาณในการวัดและแสดงค่าของสัญญาณทางคอมพิวเตอร์

ในการวัดและแสดงค่าของสัญญาณได้กำหนดลักษณะสัญญาณทั้งด้านอินพุตและเอาท์พุตเป็น 2 ลักษณะใหญ่ๆ คือ สัญญาณแบบดิจิตอลและสัญญาณแบบอะนาลอก

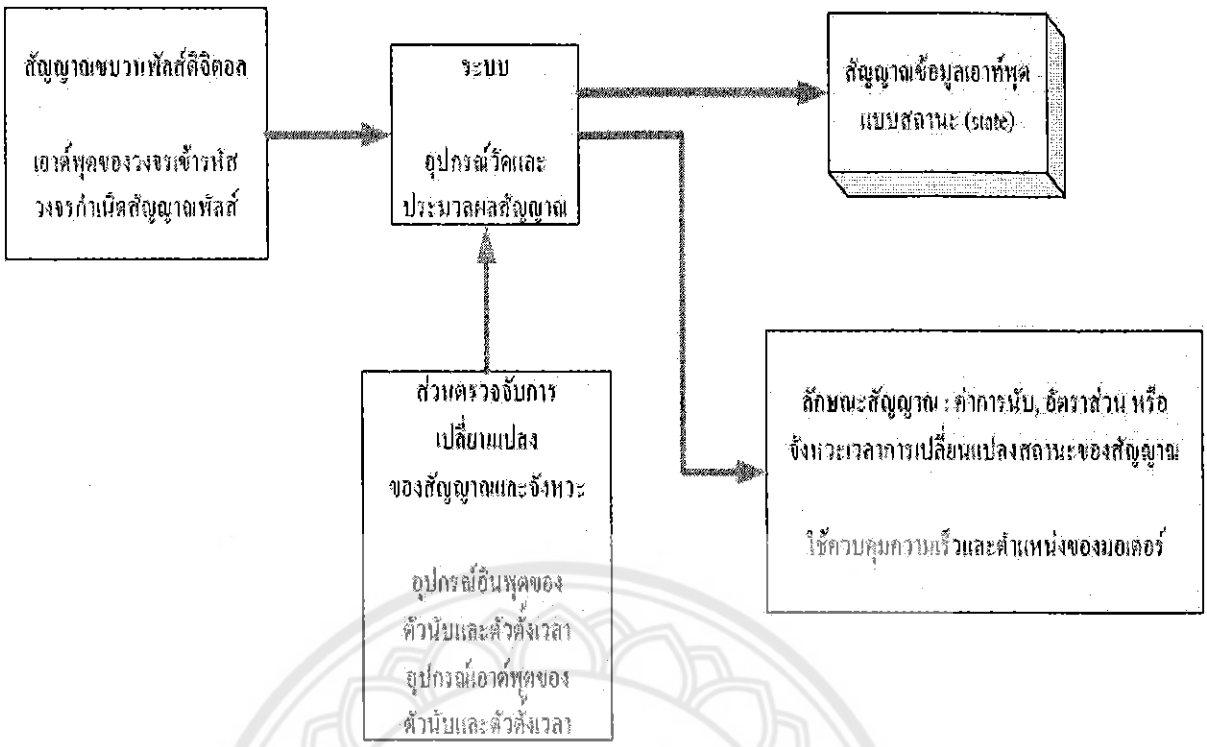
2.1.1 สัญญาณแบบดิจิตอล

สัญญาณดิจิตอลในระบบนี้มี 2 แบบคือ ค่าสถานะ (state) และ อัตราส่วน (rate) โดยค่าสถานะด้านอินพุตคือ การอ่านค่าสถานะเปิดหรือปิด (อาจพิจารณาเป็น "1" หรือ "0" แทนการ "เปิด" และ "ปิด" ก็ได้) จากอุปกรณ์ประเภทสวิทช์ และถ้าพิจารณาทางด้านเอาต์พุตก็จะเป็นการส่งค่าเปิด/ปิดไปยังอุปกรณ์อย่างรีเลย์หรือวาล์ว ดังแสดงไดอะแกรมในการใช้งานสัญญาณแบบนี้ในรูปที่ 2.1



รูปที่ 2.1 กระบวนการทำงานกับสัญญาณสถานะดิจิตอลในระบบ

สัญญาณอีกแบบหนึ่งคือ สัญญาณแบบอัตราส่วน มักเป็นการอ่านค่าพัลส์จากอุปกรณ์จำพวกเอ็นโค้ดเดอร์ (encoder) หรือตัวเข้ารหัสและอ่านค่าสัญญาณนาฬิกาจากอุปกรณ์หรือวงจรกำเนิดสัญญาณภายนอก ถ้าพิจารณาทางด้านเอาต์พุตก็จะเป็นสัญญาณควบคุมความกว้างพัลส์และความถี่เพื่อใช้ในการควบคุมการเคลื่อนตำแหน่งของมอเตอร์ ดังแสดงไดอะแกรมในการใช้งานสัญญาณแบบนี้ในรูปที่ 2.2

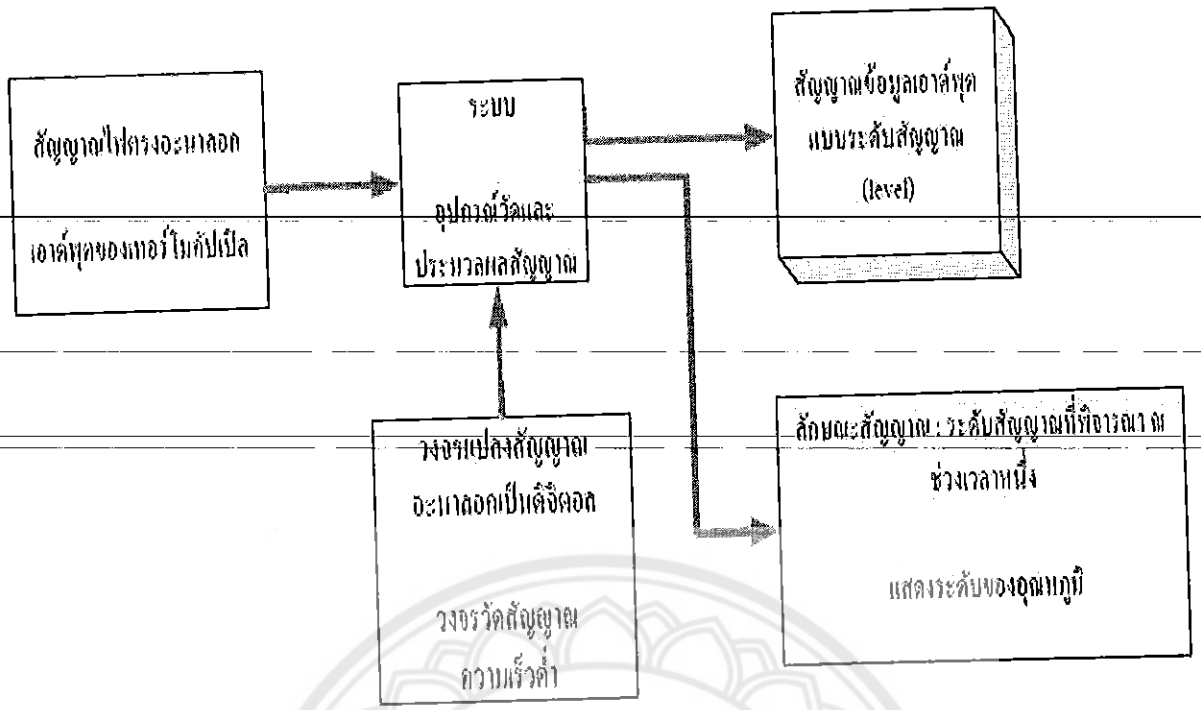


รูปที่ 2.2 กระบวนการทำงานกับสัญญาณสถานะดิจิตอลแบบขบวนพัลส์ในระบบ

2.1.2 สัญญาณอะนาลอก

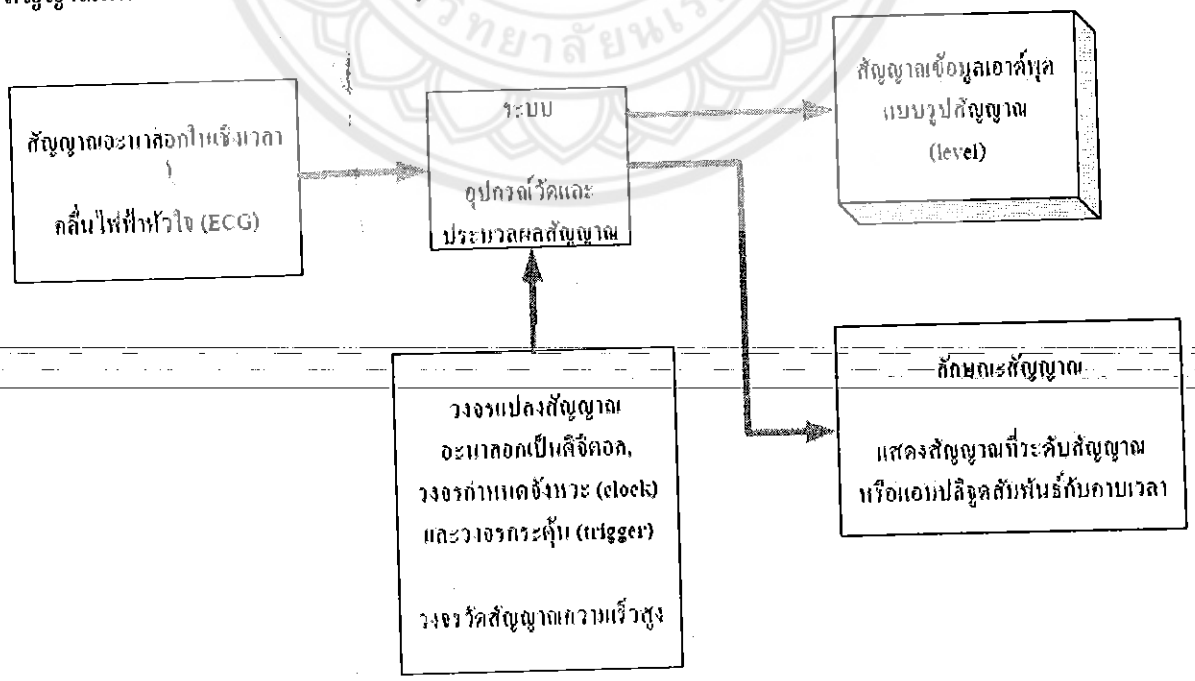
แบ่งได้ 3 แบบ คือ แบบสัญญาณไฟตรง (analog DC signals), แบบเปลี่ยนค่าตามเวลา (time-domain) และ แบบเปลี่ยนค่าตามความถี่ (frequency content) สำหรับสัญญาณอะนาลอกที่เปลี่ยนค่าตามความถี่

สัญญาณอะนาลอกไฟตรง มักเป็นค่าที่ได้จากการวัดขนาดหรือระดับของสัญญาณ ซึ่งมีการเปลี่ยนแปลงค่าสัญญาณในเวลาที่ไม่เร็วมากนัก อาทิ ค่าอุณหภูมิ, ระดับของไหล, ความดัน, อัตราการไหล, น้ำหนัก เป็นต้น สามารถใช้วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล(ADC : Analog to Digital Converter) ที่มีอัตราการสุ่มสัญญาณไม่เร็วมากได้ ดังแสดงในรูปที่ 2.3



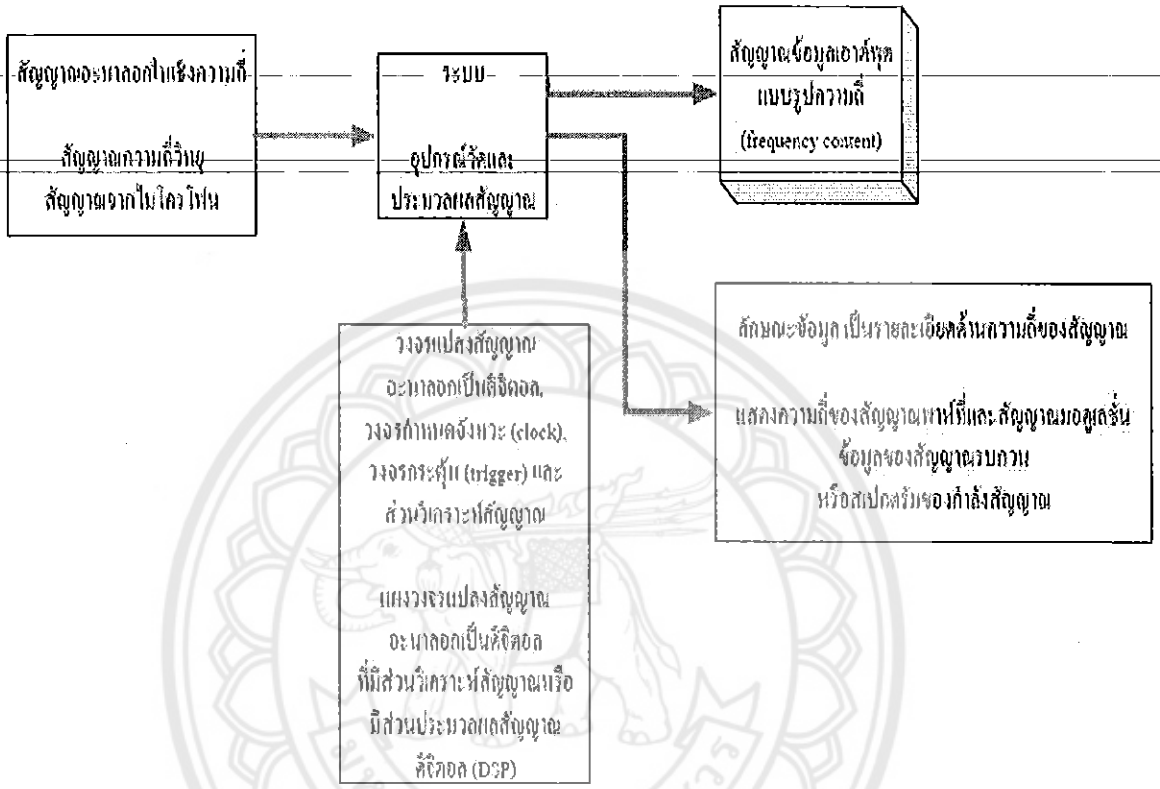
รูปที่ 2.3 กระบวนการทำงานกับสัญญาณไฟตรงอะนาลอกในระบบ

สัญญาณอะนาลอกแบบเปลี่ยนค่าตามเวลา เป็นสัญญาณที่วัดเพื่อพิจารณาลักษณะรูปสัญญาณเป็นหลัก อาทิ สัญญาณคลื่นหัวใจมนุษย์(ECG) ซึ่งมีความจำเป็นต้องใช้วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลที่มีความเร็วในการสุ่มสัญญาณสูง และอาจต้องมีฮาร์ดแวร์พิเศษเพื่อให้ค่าเวลาที่ได้จากรูปสัญญาณมีความเที่ยงตรงดังแสดงในรูปที่ 2.4



รูปที่ 2.4 กระบวนการทำงานกับสัญญาณอะนาลอกในเชิงเวลาของระบบ

สัญญาณอะนาลอกแบบเปลี่ยนค่าตามความถี่ ได้แก่ สัญญาณความถี่วิทยุ (radio frequency : RF) และสัญญาณคลื่นเสียง เป็นต้น ในการวิเคราะห์จำเป็นต้องมีฮาร์ดแวร์พิเศษเพื่อช่วยวิเคราะห์อย่าง DSP (digital signal processing) ทำงานร่วมกับวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลด้วย ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 กระบวนการทำงานกับสัญญาณอะนาลอกในเชิงความถี่ของระบบ

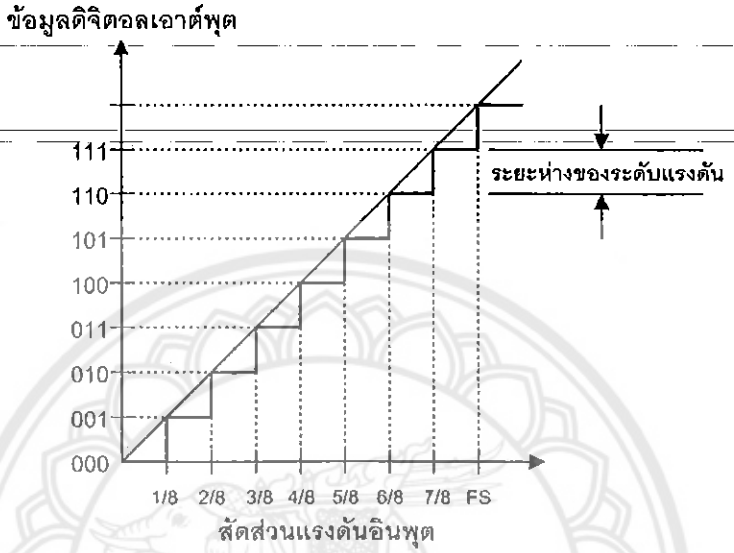
2.2 วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล

กระบวนการแปลงสัญญาณอะนาลอกเป็นดิจิตอลเป็นการแสดงความสัมพันธ์ระหว่างปริมาณของสัญญาณไฟฟ้าที่เป็นสัญญาณอะนาลอกกับข้อมูลตัวเลขที่ใช้แทนสัญญาณดิจิตอล ความแม่นยำของการแปลงจะขึ้นอยู่กับจำนวนบิตของข้อมูลดิจิตอล วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด n บิต จะเกิดข้อมูลดิจิตอลจำนวน 2^n ข้อมูล ยกตัวอย่างวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 3 บิต จะเกิดข้อมูลดิจิตอลทางเอาต์พุตทั้งสิ้น 8 ข้อมูล ดังแสดงในรูปที่ 2.6 ระยะห่างของแต่ละข้อมูลจะเป็นตัวกำหนดความแม่นยำของการแปลงสัญญาณ กระบวนการที่ทำหน้าที่ตีความระดับสัญญาณอะนาลอกว่าตรงกับข้อมูลดิจิตอลใดเรียกว่า กระบวนการควอนไตซิ่ง (quantizing)

ระยะห่างของระดับข้อมูลดิจิตอลในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล สามารถคำนวณได้จากความสัมพันธ์ทางคณิตศาสตร์ดังนี้

$$\text{ระยะห่างของระดับแรงดัน} = V_{LSB} = \frac{V_{FS}}{2^n} \tag{2.1}$$

โดยที่ V_{FS} คือ แรงดันเต็มสเกลหรือ แรงดันสูงสุดที่สามารถเกิดขึ้นได้ในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ปกติมีค่าเท่ากับไฟเลี้ยง



รูปที่ 2.6 กราฟแสดงความสัมพันธ์ระหว่างแรงดันอินพุตกับข้อมูลดิจิตอลเอาต์พุต

ถ้าหาก V_{FS} ของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 3 บิต มีค่าเท่ากับ 5 V ระยะห่างของระดับข้อมูลดิจิตอลเท่ากับ 5/8 = 0.625 V ข้อมูลดิจิตอลสูงสุดในรูปที่ 2.6 คือ 111_2 ซึ่งมีค่าเท่ากับ 7_{10} ดังนั้นที่ข้อมูลดิจิตอลสูงสุดของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 3 บิตจะมีค่าเทียบกับแรงดันอะนาลอกทางอินพุตเท่ากับ

$$\frac{7}{8} \times 5V = 4.375V \tag{2.2}$$

เมื่อเป็นเช่นนี้จึงสามารถที่จะกำหนดความสัมพันธ์ของแรงดันอะนาลอกอินพุตกับข้อมูลดิจิตอลสูงสุดในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ไม่ว่าจะเป็นกี่บิตก็ตามดังนี้

แรงดันอะนาลอกอินพุตที่ทำให้เกิดข้อมูลสูงสุด $= V_{FS} - V_{LSB}$

โดยที่ V_{FS} คือ แรงดันเต็มสเกล หรือ แรงดันสูงสุดที่สามารถเกิดขึ้นได้ในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ปกติมีค่าเท่ากับไฟเลี้ยง

V_{LSB} คือระยะห่างของระดับแรงดันที่ข้อมูล 1 บิต หรือค่าแรงดันที่ข้อมูลดิจิตอลเท่ากับ 1

ในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 3 บิต ค่าแรงดัน V_{LSB} เท่ากับ 0.625 V ถ้าหากจำนวนบิตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลมีมากขึ้นค่าของแรงดัน V_{LSB} จะลดลง ทำให้ความแม่นยำของการแปลงสัญญาณมีมากขึ้น และส่งผลให้ที่ข้อมูลดิจิตอลสูงสุดเมื่อเทียบกับแรงดันอะนาลอกทางอินพุตจะมีค่าใกล้เคียงกันมากขึ้น ยกตัวอย่าง วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล 8 บิตจะมีค่า V_{LSB} เท่ากับ

$$V_{LSB} = \frac{5}{2^8} \times 5 = \frac{5}{256} \times 5 = 0.0195V \qquad (2.3)$$

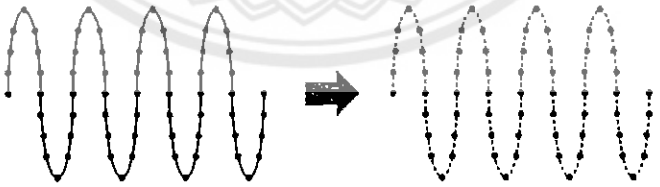
ดังนั้น ที่ข้อมูลดิจิตอลสูงสุดคือ 11111111_2 หรือ FFH จะมีค่าเท่ากับแรงดันอะนาลอกเป็น 5-0.0195 = 4.9805 V ถ้าเพิ่มเป็น 10 บิต แรงดันอะนาลอกที่ข้อมูลดิจิตอลสูงสุดจะเป็น 4.9951V

2.3 ข้อพิจารณาเลือกใช้งานวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล

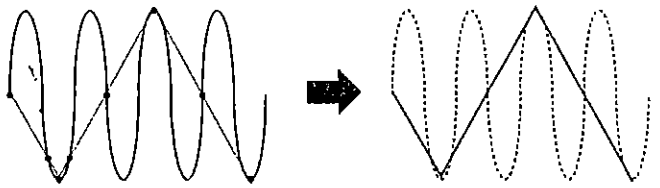
จะเห็นได้ว่าสัญญาณอินพุตในระบบอะนาลอกมีอยู่หลายลักษณะ ในการวัดสัญญาณด้วยวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล จึงต้องเลือกใช้ให้เหมาะสมกับลักษณะสัญญาณ มีข้อพิจารณาดังนี้

2.3.1 อัตราการสุ่มสัญญาณ

การสุ่มสัญญาณอะนาลอกตามทฤษฎีบทของไนควิสต์นั้น(Nyquist Theorem) กล่าวไว้ว่าอัตราการสุ่มสัญญาณจะต้องมีค่าสูงกว่าความถี่สูงสุดของสัญญาณวัดอย่างน้อย 2 เท่า เช่น การวัดความถี่เสียงซึ่งมีย่านความถี่สูงสุดที่ 20kHz ดังนั้นต้องใช้วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลที่อัตราสุ่มสัญญาณอย่างน้อย 2 เท่าของสัญญาณวัดคือ มากกว่า 40kHz ซึ่งหากการสุ่มของสัญญาณไม่เร็วพอก็อาจทำให้การอ่านข้อมูลที่เกิดขึ้นผิดพลาดได้ ดังรูปที่ 2.7



(ก) ถ้าอัตราการสุ่มสัญญาณสูงมากเพียงพอสัญญาณที่ได้จะใกล้เคียงกับสัญญาณอินพุตมาก

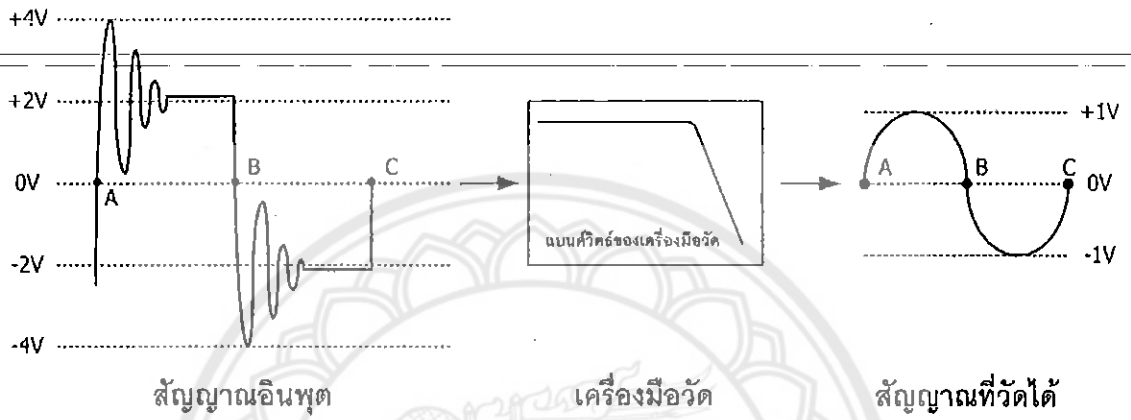


(ข) แต่ถ้าอัตราการสุ่มสัญญาณต่ำ สัญญาณที่ได้จะผิดเพี้ยนไปจากสัญญาณอินพุตมาก

รูปที่ 2.7 แสดงผลของการสุ่มสัญญาณด้วยอัตราที่แตกต่างกัน

2.3.2 แบนด์วิดธ์สัญญาณ

ในภาครับอะนาลอกที่มีความสามารถรับแบนด์วิดธ์รับสัญญาณได้ไม่เพียงพอนั้นจะทำให้ความถี่และขนาดสัญญาณที่วัดได้นั้นสูญหายหรือผิดเพี้ยนไป ดังในรูปที่ 2.8 ดังนั้น ซึ่งในภาครับอะนาลอกมีโครงสร้างของตัวเก็บประจุ (capacitor) และตัวเหนี่ยวนำไฟฟ้า (inductor) ที่สามารถตอบสนองกรองสัญญาณย่านความถี่สูงและต่ำมากออกไปได้ จึงควรพิจารณาข้อกำหนดทางฮาร์ดแวร์จากผู้ผลิตในจุดนี้ด้วย



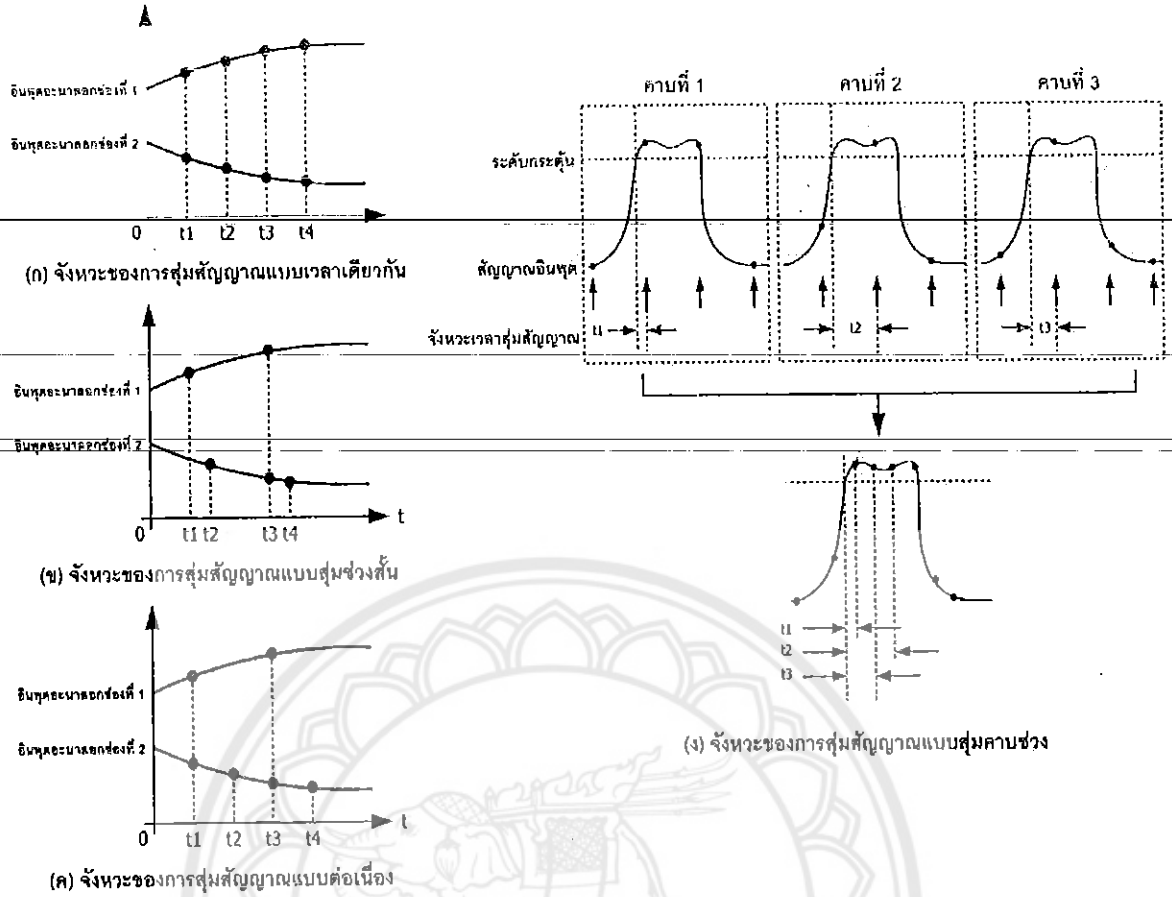
รูปที่ 2.8 แสดงผลของการวัดเมื่อเครื่องมือวัดมีข้อจำกัดด้านแบนด์วิดธ์

2.3.3 วิธีการสุ่มสัญญาณ

ปกติในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล จะสามารถรับสัญญาณได้หลายสัญญาณ อาจจะ 8 ช่อง สัญญาณ 16 หรือ 32 ช่องสัญญาณ โดยอาศัยหลักการมัลติเพล็กซ์ซิ่ง (Multiplexing) เลือกรับสัญญาณอะนาลอกหลายช่อง ไปยังวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลชุดเดียวกัน ในเทคนิควิธีการสุ่มสัญญาณของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล มีอยู่ 4 ลักษณะ ดังนี้

1. การสุ่มในเวลาเดียวกัน (Simultaneous Sampling) เป็นการสุ่มสัญญาณอะนาลอกหลายช่องสัญญาณเข้าสู่วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล มาพร้อมกันในเวลาเดียวกัน ดังในรูป 2.9 (ก) ซึ่งมักต้องมีการออกแบบส่วนวงจรการสุ่มและค้างสัญญาณ (Sample-and-hold) หรือภาครับของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลแยกต่างหากเฉพาะแต่ละช่องสัญญาณ การสุ่มลักษณะนี้ที่เห็นชัดเจน ได้แก่ การวัดเฟสสัญญาณไฟฟ้าสลับ 3 เฟส ที่ต้องให้ความสำคัญความสัมพันธ์ขนาดสัญญาณแต่ละเฟสเทียบเวลาเดียวกัน

2. การสแกนสุ่มช่วงสั้น (Interval Scanning) เป็นการสุ่มสัญญาณอะนาลอกหลายช่องที่ให้ผลใกล้เคียงกับการสุ่มในเวลาเดียวกัน ดังในรูปที่ 2.9 (ข) โดยใช้เทคนิคการมัลติเพล็กซ์ซิ่งความเร็วสูง ซึ่งการสุ่มแบบสแกนนี้เหมาะสำหรับสัญญาณอะนาลอกที่มีความถี่ไม่สูงมากนัก เช่น สัญญาณอุณหภูมิ, ความดัน ไม่เน้นความสำคัญในการเปรียบเทียบเฟสของสัญญาณ



รูปที่ 2.9 แสดงจังหวะของการสุ่มสัญญาณแบบต่างๆ ที่ใช้ในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล

3. การสแกนสุ่มต่อเนื่อง (Continuous Scanning) มักใช้งานกับการวัดสัญญาณอะนาลอกจำนวนมากและมีอัตราการเปลี่ยนค่าช้า มักมีการเพิ่มบอร์ดรีเลย์มัลติเพล็กซ์เซอร์เพื่อขยายจำนวนช่องรับสัญญาณของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ทำให้อาจสามารถรับสัญญาณ อะนาลอกได้มากเป็นร้อยช่องสัญญาณได้ โดยใช้วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลเพียงชุดเดียวความเที่ยงตรงของการวัดสัมพันธ์กับเวลาและแปรผกผันกับจำนวนช่องที่เพิ่มออกไป ยิ่งมีจำนวนช่องสัญญาณมากความเที่ยงตรงจะน้อยลง จึงต้องแน่ใจว่า สัญญาณที่วัดมีอัตรา การเปลี่ยนแปลงค่าที่ช้ามากพอเพียงในการสแกนสุ่มสัญญาณต่อเนื่องในลักษณะนี้ได้ ในรูปที่ 2.9 (ค) แสดงจังหวะของการสุ่มสัญญาณแบบนี้

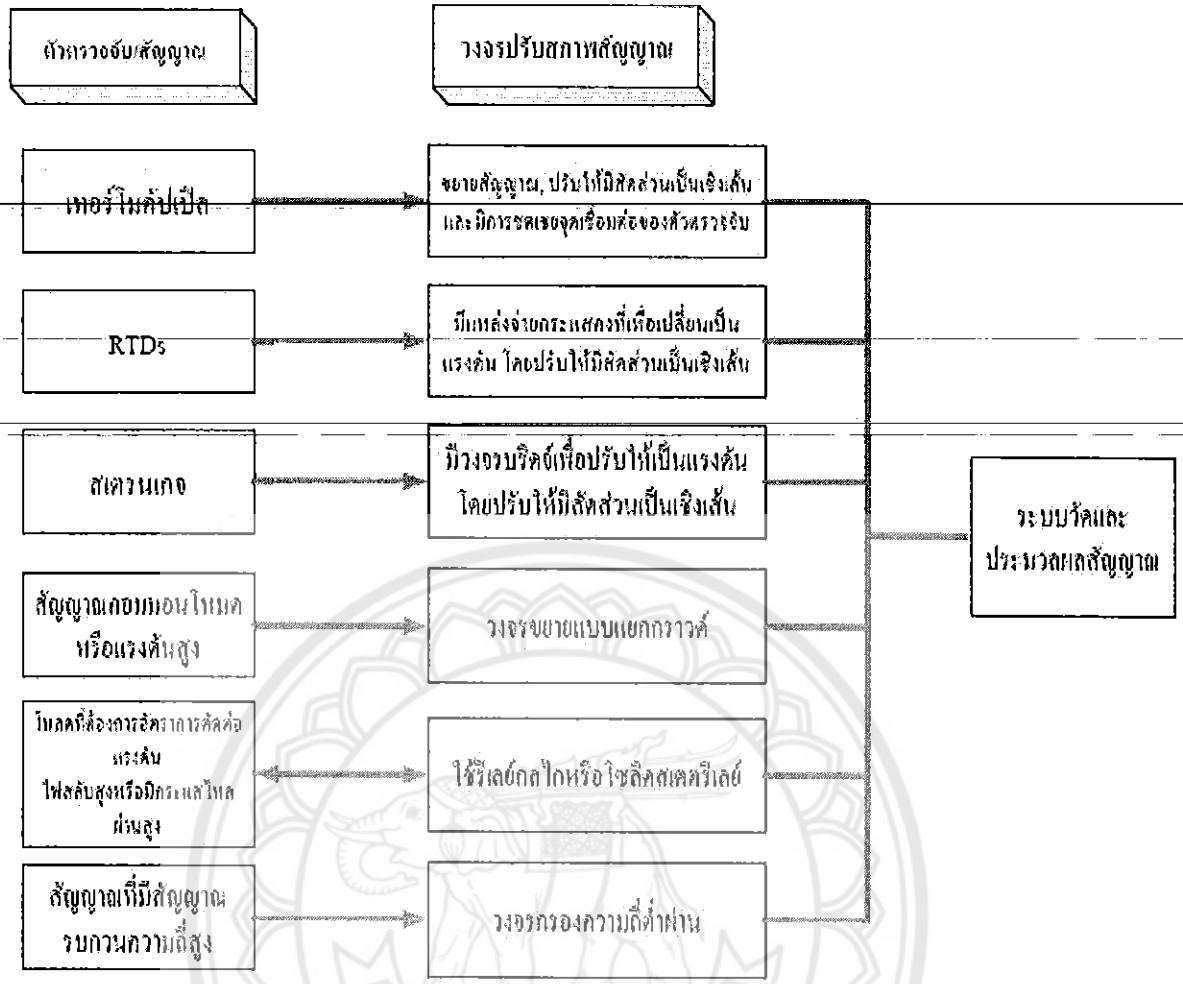
4. การสุ่มทายคาบช่วง (Random Interleaved Sampling : RIS) มักพบวิธีการวัดค่าสัญญาณนี้กับอุปกรณ์วัดสัญญาณความเร็วสูง เทคนิค RIS เป็นการเพิ่มความเร็วในการอ่านสัญญาณให้มากขึ้น อาทิ ตามปกติในวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลสามารถสุ่มสัญญาณได้ที่ 100MHz แต่เมื่อใช้เทคนิค RIS ในการสุ่มวัดค่าสัญญาณอะนาลอกที่มีรูปคลื่นซ้ำๆ กันทุกช่วงคาบของคลื่น จะสามารถวัดสัญญาณได้เร็วถึง 2.5GHz ดังแสดงในรูปที่ 2.9 (ง) การสุ่มวัดสัญญาณแบบนี้สามารถใช้ได้กับสัญญาณที่มีรูปคลื่นซ้ำๆกันทุกช่วงคาบเท่านั้น เนื่องจากเป็นการเก็บจำค่าสัญญาณในแต่ละคาบ ซึ่งมีจุดกระตุ้นวัดค่าหรือจุดทริกเกอร์ (trigger point) ต่างเวลากัน แล้วมาปะติดปะต่อเป็นรูปคลื่นรวม

2.4 การเลือกวงจรปรับสภาพสัญญาณ

การเลือกวงจรปรับสภาพสัญญาณก็เป็นสิ่งที่ต้องให้ความสนใจมากด้วยเช่นกัน เนื่องจากตัวตรวจจับแต่ละแบบจะให้สัญญาณไฟฟ้าออกมาแตกต่างกันทั้งชนิดของสัญญาณ และขนาดของสัญญาณ ในตัวตรวจจับบางแบบให้สัญญาณไฟฟ้าออกมาในรูปของกระแสไฟฟ้า แรงดันไฟฟ้า หรือความต้านทานไฟฟ้า และให้ขนาดของสัญญาณในหน่วยของมิลลิแอมป์ มิลลิโวลต์ โวลต์ หรือ โอห์มขึ้นอยู่กับชนิดของสัญญาณไฟฟ้าที่ได้จากตัวตรวจจับ ดังนั้น วงจรปรับสภาพจะมีหน้าที่หลักในการแปลงสัญญาณไฟฟ้าที่แตกต่างกันเหล่านั้นเป็นแรงดันไฟฟ้าที่เหมาะสมกับวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล ซึ่งโดยปกติจะสามารถรับสัญญาณอินพุตในรูปของแรงดันไฟฟ้ากระแสตรง 0-5 V เป็นส่วนใหญ่ แต่ในบางวงจรจะรับแรงดัน 0-2.5 V, 0-10 V, -5 ถึง +5 V, -2.5 ถึง +2.5 V หรือ -10 ถึง +10 V

ดังนั้นวงจรปรับสภาพสัญญาณจึงมีทั้งเป็นวงจรขยายในกรณีที่สัญญาณจากตัวตรวจจับมีระดับสัญญาณที่ไม่สูงมากนัก วงจรลดทอนสัญญาณในกรณีที่ระดับสัญญาณจากตัวตรวจจับสูงเกินกว่าที่วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลจะรับได้ วงจรแปลงกระแสไฟฟ้าเป็นแรงดันไฟฟ้า (I to V converter) วงจรแปลงความถี่เป็นแรงดัน (F to V converter) เป็นต้น และที่สำคัญคือ สัญญาณที่ออกจากวงจรปรับสภาพสัญญาณจะต้องมีสัดส่วนของการปรับเปลี่ยนตามการเปลี่ยนแปลงของสัญญาณจากตัวตรวจจับอย่างถูกต้อง อาจจะมีสัดส่วนสัมพันธ์กันในแบบเชิงเส้น (linear) หรือแบบลอการิธึ่ม (logarithm) ก็ได้

ในรูปที่ 2.10 แสดงตัวอย่างของการเลือกใช้วงจรปรับสภาพสัญญาณให้เหมาะสมกับตัวตรวจจับก่อนที่จะนำสัญญาณที่เหมาะสมส่งต่อไปยังแผงวงจรวัดและประมวลผลสัญญาณ ต่อไป



รูปที่ 2.10 แสดงตัวอย่างของการเลือกใช้วงจรปรับสภาพสัญญาณให้เข้ากับตัวตรวจจับ

2.5 ข้อมูลของ PCF8591 ไอซี ADC/DAC

PCF8591 เป็นไอซีที่ทำหน้าที่แปลงสัญญาณอะนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลเป็นอะนาลอกในตัวเดียวกัน เป็นผลงานของฟิลิปส์ เซมิคอนดักเตอร์ ผู้พัฒนาระบบบัส I^2C คุณสมบัติทางเทคนิคที่สำคัญของ PCF8591 มีดังนี้

1. ทำงานด้วยความละเอียดของข้อมูลดิจิตอลขนาด 8 บิต
2. มีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ADC) ขนาด 8 บิต 4 ช่อง
3. มีวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอก (DAC) ขนาด 8 บิต 1 ช่อง
4. ทำงานโดยใช้แหล่งจ่ายไฟชุดเดียว ตั้งแต่ 2.5-6 V
5. กินกระแสไฟฟ้าขณะอยู่ในสภาวะสแตนบายต์ต่ำ
6. ติดต่อกับไมโครคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ผ่านระบบบัส I^2C

7. สามารถเลือกตำแหน่งแอดเดรสทางฮาร์ดแวร์จากขา A0, A1, A2 ทำให้สามารถต่อพ่วง กันได้สูงสุดถึง 8 ตัว จึงขยายจำนวนต่อช่องอินพุตของสัญญาณอะนาลอกได้สูงถึง 32 ช่อง และช่องเอาต์พุตของสัญญาณอะนาลอกจากวงจร DAC สูงถึง 8 ช่อง

8. อัตราการสุ่มข้อมูล (sampling) ขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาบนบัส I^2C

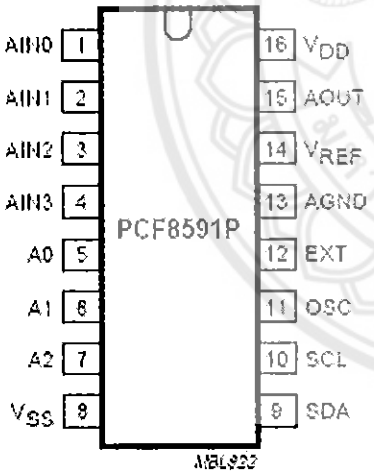
9. วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล (ADC) สามารถเลือกการทำงานเป็นแบบแยกช่องหรือ ทำงานเป็นวงจรดิฟเฟอเรนเชียลได้ โดยจำนวนช่องอินพุตจะลดลงเหลือ 2 ช่อง

10. การอ่านค่าสามารถกำหนดให้เลื่อนช่องอินพุตโดยอัตโนมัติได้

11. สามารถรับสัญญาณอะนาลอกระดับแรงดันตั้งแต่ V_{SS} ไปจนถึง V_{DD}

12. วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลเป็นแบบซักเซสซีฟแอปพร็อกซิเมชั่น

PCF8591 สามารถทำหน้าที่เป็นไอซีแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 8 บิต 4 ช่อง และเป็นไอซีแปลงสัญญาณดิจิตอลเป็นอะนาลอกได้ในคราวเดียวกัน ด้วยการควบคุมผ่านระบบบัส I^2C ทำให้สามารถต่อพ่วง PCF8591 ได้สูงสุด 8 ตัว รองรับการอ่านค่าสัญญาณอะนาลอกอินพุต ได้สูงสุดถึง 32 ช่อง และส่งสัญญาณอะนาลอกเอาต์พุตสูงสุดได้ 8 ช่องด้วยการกำหนดแอดเดรสจากขา A0, A1 และ A2 การจัดขาและรายละเอียดตำแหน่งขาของ PCF8591 แสดงในรูปที่ 2.11



SYMBOL	PIN	DESCRIPTION
AIN0	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V_{SS}	8	negative supply voltage
SDA	9	I²C-bus data input/output
SCL	10	I²C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V_{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V_{DD}	16	positive supply voltage

รูปที่ 2.11 การจัดขาและตารางแสดงชื่อขาสัญญาณของ PCF8591

2.5.1 รายละเอียดฟังก์ชั่นต่างๆ ของ PCF8591

2.5.1.1 ตำแหน่งแอดเดรส

ในระบบบัส I^2C การติดต่อกับอุปกรณ์แต่ละตัวต้องระบุแอดเดรสของอุปกรณ์เหล่านั้นอย่างชัดเจน สำหรับไอซี PCF8591 จะมีค่าเท่ากับ 1001 (ฐานสอง) ข้อมูล 3 บิตถัดมาจะเป็นค่าของแอดเดรสที่ผู้ใช้งานสามารถกำหนดได้ทางฮาร์ดแวร์เพื่อเลือกไอซี PCF8591 ที่ต้องการติดต่อด้วยในกรณีที่มีการต่อใช้งานมากกว่า 1 ตัว ส่วนบิต LSB ใช้ในการกำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับไอซีตัวนั้นๆ ตัวอย่าง ถ้าต้องการอ่านข้อมูลจากชิปที่กำหนดแอดเดรสไว้เป็น 000 จะต้องป้อนข้อมูลแอดเดรสเท่ากับ &H91 เป็นต้น

2.5.1.2 ข้อมูลควบคุม

หลังจากส่งข้อมูลกำหนดแอดเดรสให้แก่ PCF8591 แล้ว ต้องส่งข้อมูลควบคุมตามไปด้วย เพื่อกำหนดคุณสมบัติของวงจร ADC และ DAC ภายใน PCF8591 โดยมีรายละเอียดตามรูปที่ 2.12 ดังนี้

บิต 6 ของข้อมูลควบคุมใช้สำหรับการเอ็นเอเบิลขาอะนาลอกเอาต์พุต เมื่อต้องการเอ็นเอเบิลต้องกำหนดให้ขานี้เป็น "1"

บิต 4 และ บิต 5 ของข้อมูลควบคุมใช้สำหรับการกำหนดรูปแบบของสัญญาณอะนาลอกอินพุตที่ป้อนให้แก่ PCF8591

บิต 2 ใช้สำหรับเลือกรูปแบบการอ่านข้อมูลจากขาอินพุตอะนาลอกว่าจะเป็นการอ่านจากเพียงอินพุตเดียวหรืออ่านแบบเรียงลำดับทุกอินพุต ถ้าต้องการเลือกให้อ่านแบบเรียงลำดับ ต้องกำหนดให้บิตนี้เป็น "1"

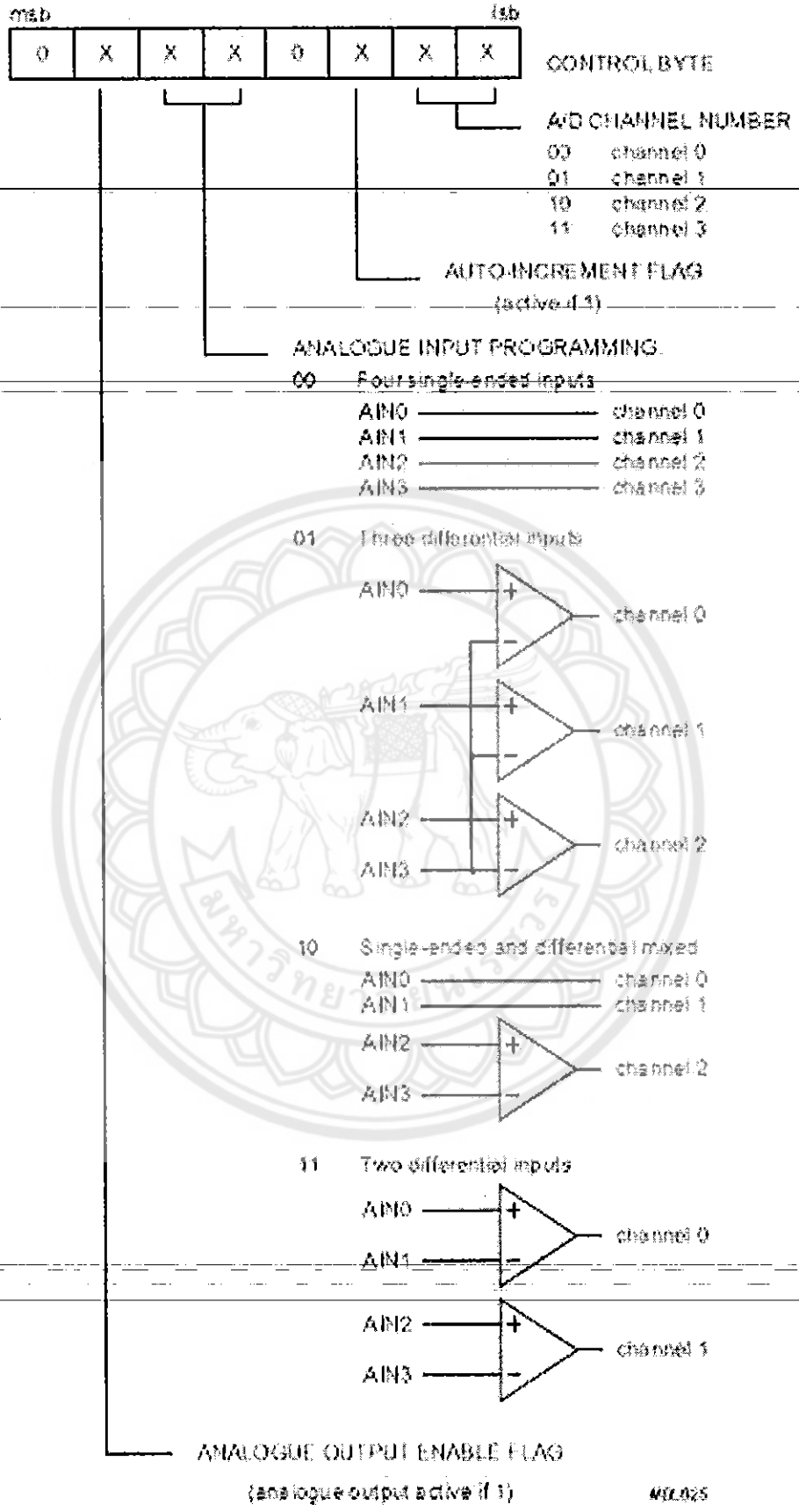
บิต 0 และ บิต 1 ใช้สำหรับกำหนดช่องของอินพุตอะนาลอกที่ต้องการอ่าน ถ้ากำหนดให้บิต 2 เป็น "1" หลังจากอ่านค่าของบิต "0" และบิต "1" แล้ว ในการอ่านค่าครั้งต่อไปจะเป็นการ อ่านค่าอินพุตจากช่องที่ 1

ข้อมูลควบคุมทั้งหมดจะถูกเก็บไว้ในรีจิสเตอร์ควบคุมภายใน PCF8591

เมื่อจ่ายไฟให้แก่ PCF8591 ครั้งแรก บิตต่างๆ ข้อมูลภายในรีจิสเตอร์ควบคุมจะถูกกำหนดให้เป็น "0"

2.5.1.3 ออสซิลเลเตอร์

วงจรออสซิลเลเตอร์ภายใน PCF8591 จะสร้างสัญญาณนาฬิกาสำหรับการแปลงสัญญาณอะนาลอกเป็นดิจิตอล เมื่อต้องการใช้วงจรออสซิลเลเตอร์ภายในขา EXT ต้องต่อ ลงกราวด์ ถ้าต้องการใช้ออสซิลเลเตอร์จากภายนอกขา EXT ต้องต่อเข้ากับไฟบวก แล้วป้อนสัญญาณนาฬิกาเข้าที่ขา OSC ของ PCF8591 โดยความถี่ของสัญญาณนาฬิการสูงสุดที่ป้อนให้กับออสซิลเลเตอร์เท่ากับ 1.25 MHz



รูปที่ 2.12 แสดงบิตข้อมูลควบคุม

2.5.2 การเขียนโปรแกรมเพื่ออ่านข้อมูลอินพุตอะนาลอกแบบต่อเนื่องจาก PCF8591

ในการติดต่อกับ PCF8591 มีรูปแบบสัญญาณตามมาตรฐานของระบบบัส I^2C โดยการติดต่อมีขั้นตอนดังนี้

1. ส่งสัญญาณ START
2. ส่งข้อมูลกำหนดแอดเดรส ในที่นี้กำหนดแอดเดรสของ PCF8591 ไว้ที่ 000 (ขา A0, A1, A2 ต่อลงกราวด์ทั้งหมด) และให้ทำงานในโหมดเขียนข้อมูล (ป้อนข้อมูล "0" ให้แก่ บิต R/W)
3. รอรับสัญญาณตอบกลับจาก PCF8591
4. ส่งข้อมูลควบคุมไปยัง PCF8591 โดยใช้ค่า 45H ซึ่งเป็นการเอ็นเอเบิลอะนาลอกเอาต์พุต, กำหนดให้อินพุตอะนาลอกทำงานในโหมดซิงเกิล, กำหนดให้ใช้การเลื่อนอ่านข้อมูลแบบต่อเนื่อง และเริ่มอ่านข้อมูลจากช่องที่ 1
5. รอรับสัญญาณตอบกลับจาก PCF8591
6. ส่งสัญญาณ STOP
7. ส่งสัญญาณ START
8. ส่งข้อมูลกำหนดแอดเดรสอีกครั้ง โดยครั้งนี้กำหนดให้เป็นโหมดอ่านข้อมูล (ส่งลอจิก "1" ให้แก่บิต R/W) เพื่อเริ่มต้นอ่านค่าข้อมูลจากช่องสัญญาณอะนาลอกอินพุต
9. รอรับสัญญาณตอบกลับจาก PCF8591
10. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 1
11. ส่งสัญญาณรับรู้จากคอมพิวเตอร์ไปยัง PCF8591
12. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 2
13. ส่งสัญญาณรับรู้จากคอมพิวเตอร์ไปยัง PCF8591
14. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 3
15. ส่งสัญญาณรับรู้จากคอมพิวเตอร์ไปยัง PCF8591
16. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลช่องที่ 4
17. ส่งสัญญาณรับรู้จากคอมพิวเตอร์ไปยัง PCF8591
18. ส่งสัญญาณ STOP

บทที่ 3
วิธีการออกแบบ

ขั้นตอนนี้จะทำการศึกษา ค้นคว้าข้อมูลเกี่ยวกับการทำงานของวงจรและการออกแบบระบบวัดสัญญาณไฟฟ้าซึ่งทำงานร่วมกับคอมพิวเตอร์ผ่านทางพอร์ตขนาน และการเขียนโปรแกรมแสดงลักษณะของสัญญาณโดยใช้โปรแกรม Visual Basic

ขั้นตอนและการออกบบวงจร

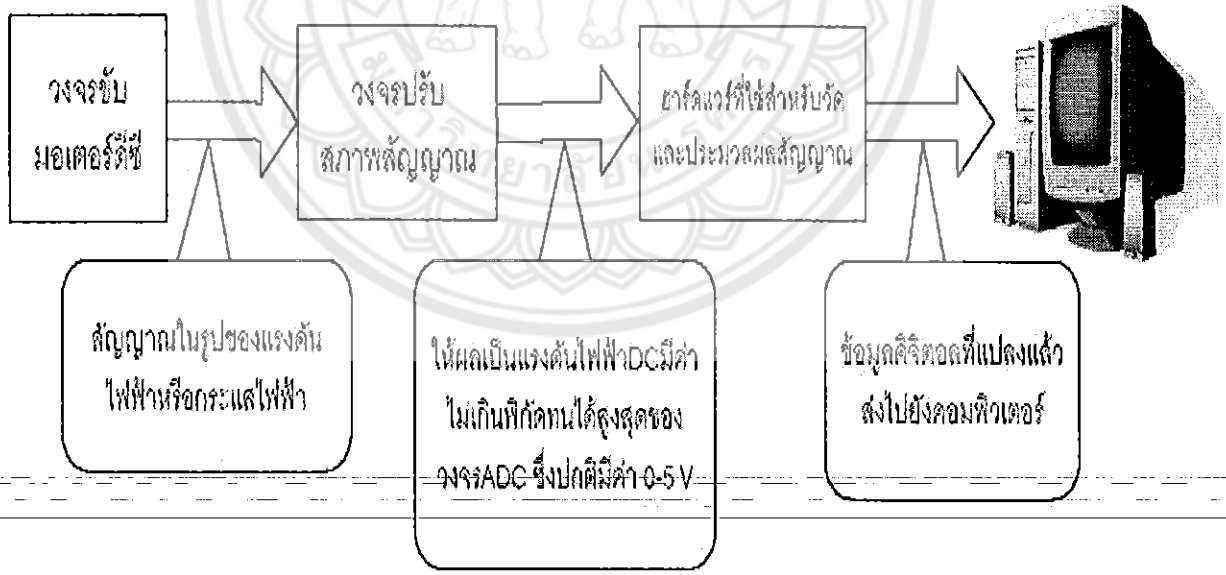
ส่วนประกอบสำคัญของระบบวัดและแสดงลักษณะของสัญญาณไฟฟ้าทางคอมพิวเตอร์ในรูปที่ 3.1 แสดงไดอะแกรมการทำงานเบื้องต้นของระบบ มีส่วนประกอบสำคัญ 4 ส่วน คือ

3.1 มอเตอร์ดีซี (DC Motor)

3.2 วงจรปรับสภาพสัญญาณ (Signal conditioning)

3.3 ส่วนฮาร์ดแวร์เพื่อการวัดสัญญาณและเชื่อมต่อกับคอมพิวเตอร์

3.4 โปรแกรมสำหรับการแสดงค่าของสัญญาณไฟฟ้า (Application software)

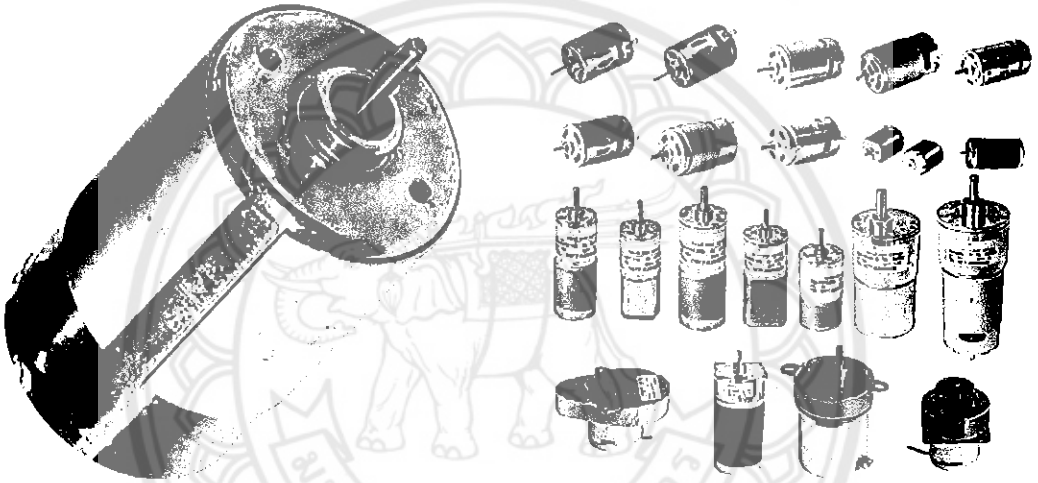


รูปที่ 3.1 ไดอะแกรมการทำงานเบื้องต้นของระบบ

ซึ่งในแต่ละส่วนมีหลักการและขั้นตอนออกแบบดังนี้

3.1 มอเตอร์ดีซี (DC Motor)

มอเตอร์ไฟฟ้ากระแสตรง จะทำหน้าที่เปลี่ยนพลังงาน จากพลังงานไฟฟ้าเป็นพลังงานกล โดยประกอบด้วย 2 ส่วน คือส่วนที่อยู่กับที่ (Stator) และส่วนที่เคลื่อนที่ (Rotor) ลักษณะการหมุนของมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) กับมอเตอร์สเต็ปปิ้ง (Stepping Motor) จะมีความแตกต่างกัน คือ DC Motor จะหมุนอย่างต่อเนื่อง ถ้ากลับทิศทางก็หมุนอย่างต่อเนื่องเช่นกัน ซึ่งการเคลื่อนที่ขณะหมุนจะสม่ำเสมอแต่ไม่สามารถควบคุมตำแหน่งได้



รูปที่ 3.2 มอเตอร์ไฟฟ้ากระแสตรง

3.2 วงจรปรับสภาพสัญญาณ (Signal conditioning)

วงจรปรับสภาพสัญญาณเป็นวงจรที่ทำขึ้นมาเพื่อปรับแรงดันและกระแสที่เข้ามาให้มีค่าแรงดันที่ไม่เกิน 5 โวลท์ และค่ากระแสก็ไม่เกิน 10 มิลลิแอมป์

เช่น เมื่อมีแรงดันสูงสุดในย่านนี้ คือ 250 โวลท์ เข้ามาเราจะต้องทำการปรับสภาพสัญญาณเข้าจาก 250 โวลท์ ให้สัญญาณออกเหลือเพียงค่าไม่เกิน 5 โวลท์ โดยอาศัยหลักการตามทฤษฎีแบ่งแรงดันเพื่อสัญญาณออกที่ได้เราจะส่งเข้า A/D

ส่วนในย่านอื่นๆ ก็มีหลักการแบบเดียวกันแต่จะมีความแตกต่างกันที่ตรงค่าความต้านทานที่ใช้เท่านั้น

โดยในการออกแบบวงจรจะใช้โปรแกรม PSpice ออกแบบและวิเคราะห์วงจร ซึ่งวงจรที่ออกแบบได้ดังรูปที่ 3.3 และ รูปที่ 3.4

และวงจรที่ได้ออกแบบ มีหลักการคำนวณค่าแรงดัน ดังนี้

กำหนด ให้มีกระแสไฟฟ้า : $I = 2A$

และแรงดันไฟฟ้า : $V = 20V$

ค่าความต้านทาน : $R = 1k\Omega$

เนื่องจาก บอร์ดที่ทำการทดลองสามารถรับแรงดันไฟฟ้าสูงสุดได้ 5 V

จึงต้องทำการลดทอนแรงดัน โดยที่ $R_1 = 1k\Omega$

$R_2 = 5k\Omega$

$R_3 = 1k\Omega$

$R_4 = 5k\Omega$

$R_5 = 1k\Omega$

$R_6 = 5k\Omega$

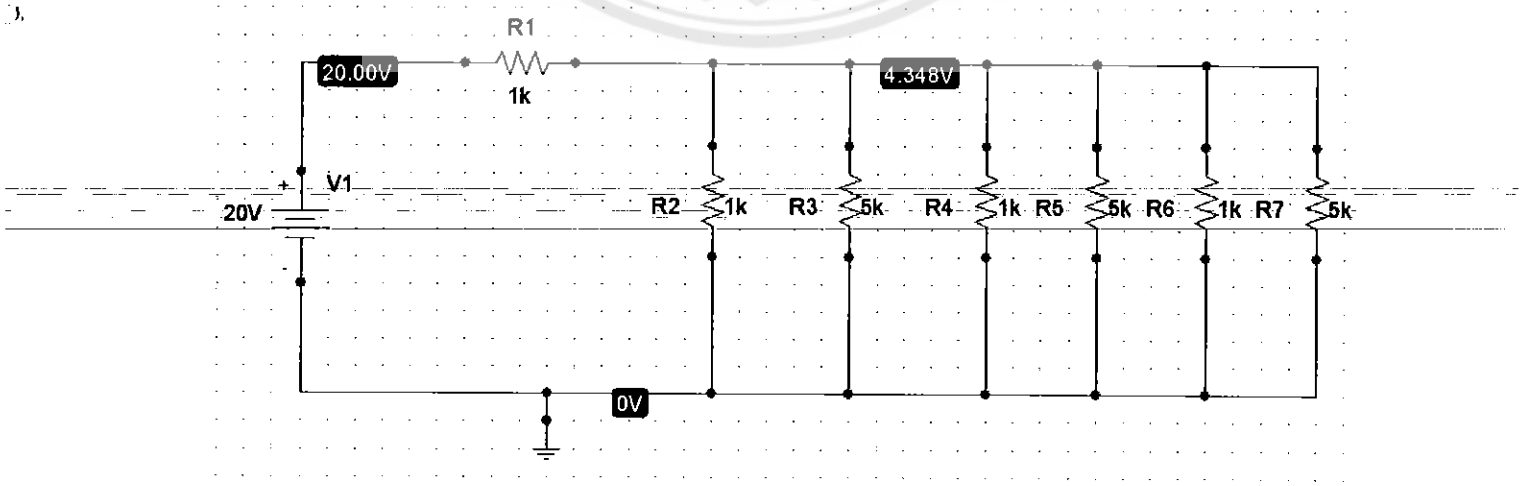
ดังนั้น ค่าความต้านทานรวม $= \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4} + \frac{1}{R_5} + \frac{1}{R_6}$

$= 274.725\Omega$

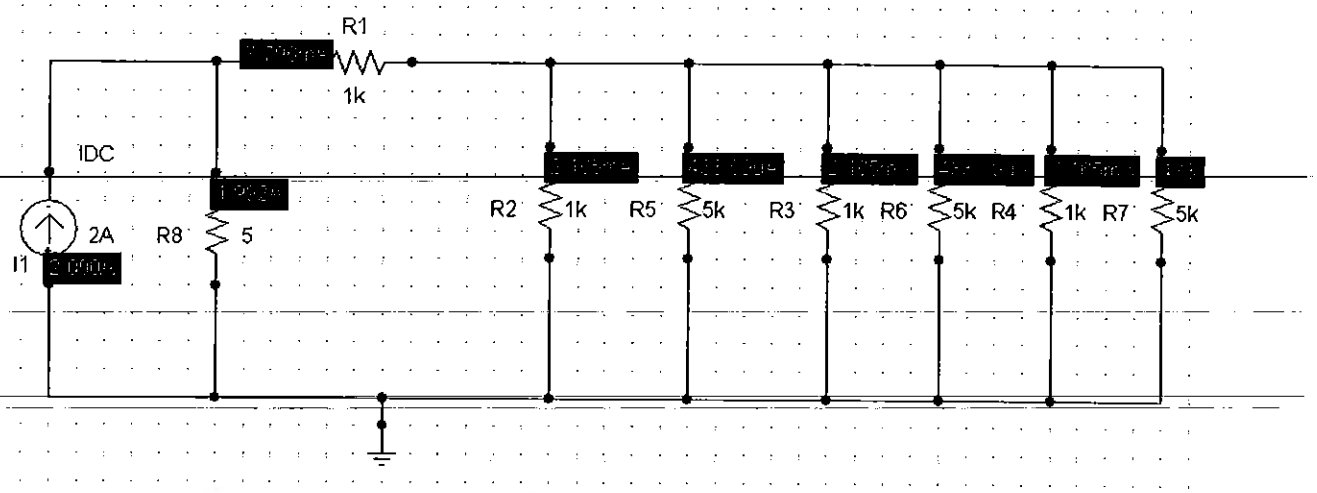
จะได้กระแส ไฟฟ้า จาก $V_o = (\frac{R_{eq}}{R + R_{eq}}) \times V_i$

$= (\frac{274.725\Omega}{1k\Omega + 274.725\Omega}) \times 20V = 4.31V$

ดังนั้น แรงดันที่ตกคร่อม $R_{eq} = 4.31V$



รูปที่ 3.3 วงจรปรับสภาพแรงดันไฟฟ้า



รูปที่ 3.4 วงจรปรับสภาพกระแสไฟฟ้า

3.3 ส่วนฮาร์ดแวร์เพื่อการวัดสัญญาณและเชื่อมต่อกับคอมพิวเตอร์

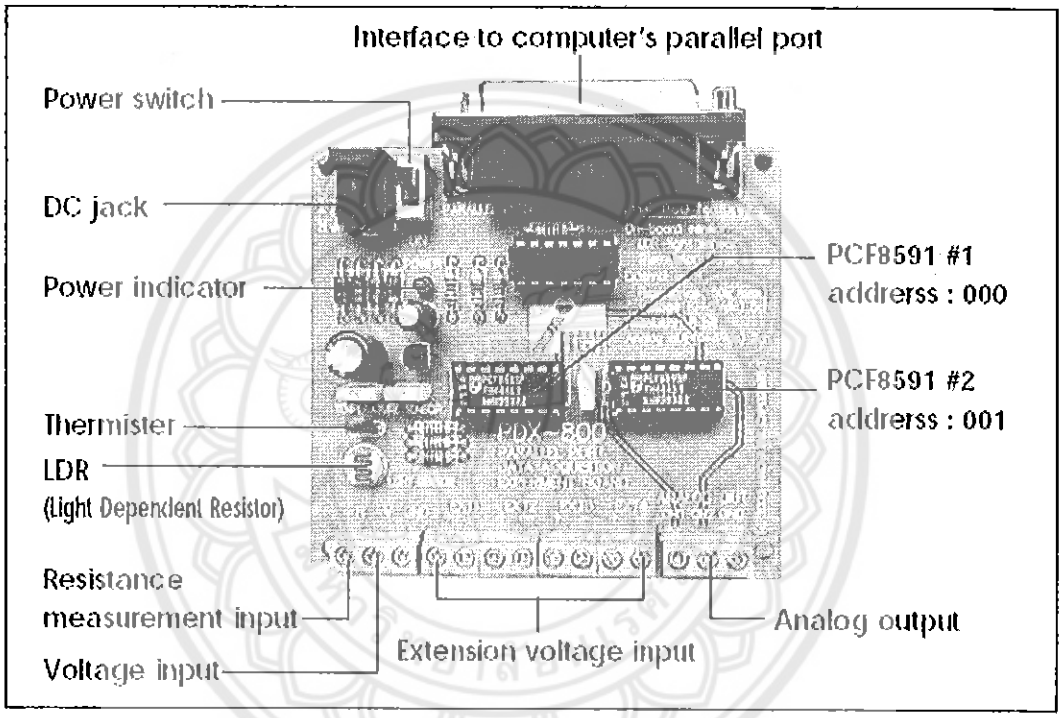
ส่วนของฮาร์ดแวร์ที่ใช้ในการเชื่อมต่กับคอมพิวเตอร์ คือ บอร์ดทดลองPDX-800 เพื่อใช้ในการวัดและประมวลผลสัญญาณ

คุณสมบัติทางเทคนิค

- มีตัวตรวจจับอุณหภูมิติดตั้งบนบอร์ด เป็นเทอร์มิสเตอร์ชนิด NTC ค่าความต้านทาน 1kohm ที่อุณหภูมิห้อง
- มีตัวตรวจจับแสงติดตั้งบนบอร์ด เป็นตัวต้านทานแปรค่าตามแสงหรือ LDR
- อินพุตสำหรับวัดค่าความต้านทาน (resistance) วัดได้สูงสุด 150 kohm
- อินพุตสำหรับวัดแรงดัน รับแรงดันได้สูงสุด 5 V
- จำนวนช่องอินพุตรับสัญญาณอะนาลอกจากภายนอก : 4 ช่อง
- วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 8 บิต โดยใช้ไอซี PCF8591
- ระดับแรงดันอะนาลอกอินพุต : 5 V
- วงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกขนาด 8 บิต 2 ช่องให้แรงดันไฟตรงเอาต์พุต 0-5 V
- เชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตขนาน
- มีวงจรบัฟเฟอร์เพื่อป้องกันความเสียหายต่อพอร์ตขนาน
- มีวงจรแปลงสัญญาณพอร์ตอนุกรมเป็นสัญญาณในระบบบัส I^2C
- ใช้ไฟเลี้ยง +9V มีวงจรเรกูเลเตอร์ +5V บนบอร์ด พร้อมวงจรป้องกันการจ่ายไฟกลับขั้ว
- ขนาดของบอร์ด : $5\frac{1}{2}\times 5$ นิ้ว

วงจรและการทำงาน

ในรูปที่ 3.5 แสดงส่วนประกอบทั้งหมดของบอร์ดที่ใช้ประกอบในการทดลอง ส่วนวงจรสมบูรณ์แสดงในรูปที่ 3.6 หัวใจสำคัญของวงจรคือ IC3 และ IC4 เบอร์ PCF8591 อันเป็นไอซีแปลงสัญญาณอะนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลเป็นอะนาลอกขนาด 8 บิตที่ใช้การติดต่อผ่านระบบบัส I^2C โดยภายใน PCF8591 มีอินพุตสำหรับรับแรงดันไฟตรง 0-5 V มากถึง 4 ช่อง และมีเอาต์พุตของวงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกที่ให้แรงดัน 0-5 V อีก 1 ช่อง และสามารถต่อพ่วงกันเพื่อขยายจำนวนช่องสัญญาณได้อีก 7 ตัว โดยยังคงใช้สายสัญญาณในการติดต่อและควบคุมการทำงานเพียง 2 เส้นตามข้อกำหนดของระบบบัส I^2C

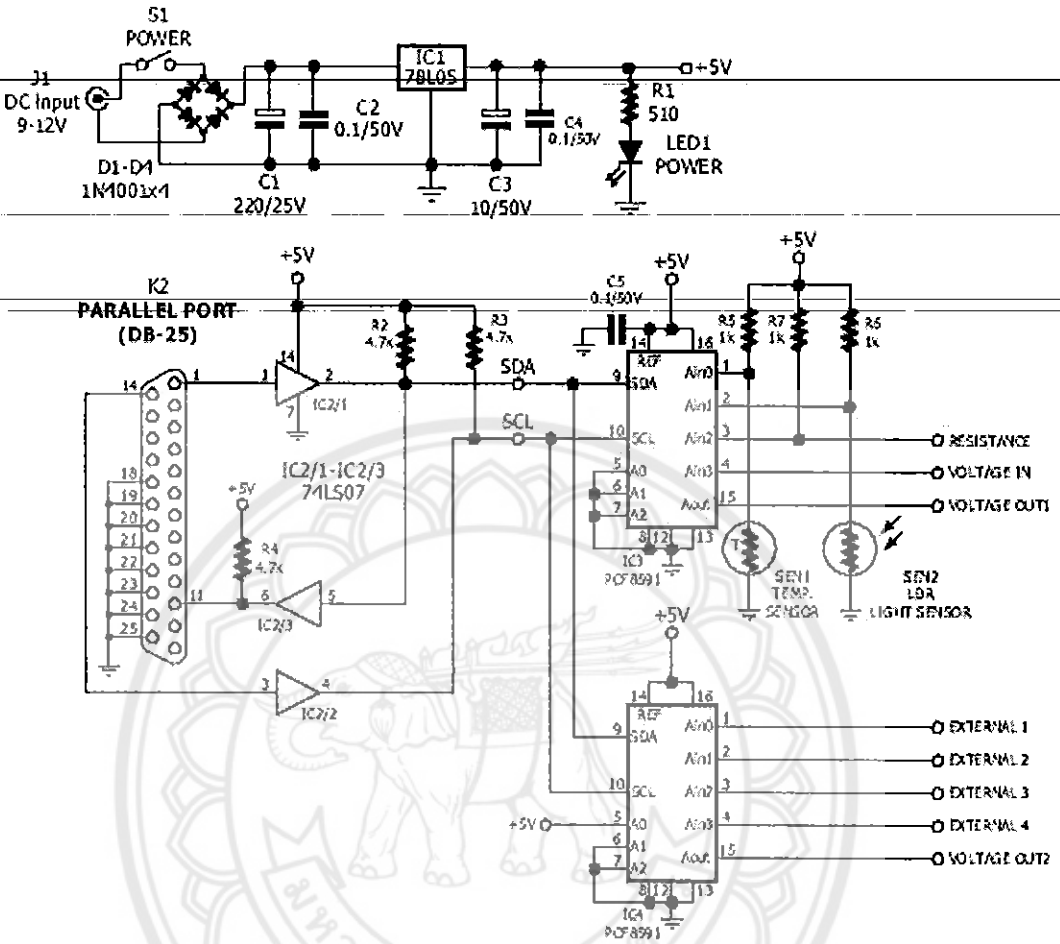


รูปที่ 3.5 แสดงส่วนประกอบของบอร์ด PDX-800

การควบคุม IC3 และ IC4 มาจากพอร์ตขนานของคอมพิวเตอร์ผ่านทางคอนเน็กเตอร์ K2 โดยมี IC2 เบอร์ 7407 ทำหน้าที่ในการสร้างสัญญาณให้เข้ากับข้อกำหนดของระบบบัส I2C นั่นคือ สร้างสัญญาณข้อมูลอนุกรม SDA (Serial Data) และสัญญาณนาฬิกาอนุกรม SCL (Serial Clock) สายสัญญาณ SDA และ SCL จะต่อเข้ากับขา SDA และ SCL ของ IC3 และ IC4 การต่อพ่วงกันของ IC3 และ IC4 สามารถทำได้โดยการกำหนดขาแอดเดรสทางฮาร์ดแวร์ A0-A2 ของ IC3 และ IC4 จากวงจรของบอร์ด กำหนดให้ IC3 มีแอดเดรสเท่ากับ 000 ในขณะที่ IC4 มีแอดเดรสเท่ากับ 001

เนื่องจาก PCF8591 มีอินพุตสำหรับรับแรงดันอะนาลอก 4 ช่องต่อตัว เมื่อนำมาต่อพ่วงกัน 2 ตัว จึงได้อินพุตสำหรับรับแรงดันอะนาลอกมากถึง 8 ช่อง และเอาต์พุตสำหรับส่งแรงดันอะนาลอกจาก

วงจรแปลงสัญญาณดิจิตอลเป็นอะนาลอกภายในไอซีอีก 2 ช่อง ในบอร์ดจึงทำการจัดสรรอินพุตอะนาลอกเพื่อให้เหมาะสมกับการใช้งานดังนี้



รูปที่ 3.6 วงจรสมบูรณ์ของบอร์ด PDX-800

ช่อง Ain0 ของ PCF8591 ตัวที่ 1 (IC3) ต่อกับเทอร์มิสเตอร์เพื่อใช้ในการวัดอุณหภูมิ

ช่อง Ain1 ของ PCF8591 ตัวที่ 1 (IC3) ต่อกับ LDR หรือตัวต้านทานแปรค่าตามแสงเพื่อใช้ในการตรวจจับแสง

ช่อง Ain2 ของ PCF8591 ตัวที่ 1 (IC3) ต่อกับวงจรแบ่งแรงดันเพื่อใช้ในการวัดค่าความต้านทานไฟฟ้า

ช่อง Ain3 ของ PCF8591 ตัวที่ 1 (IC3) เป็นอินพุตสำหรับวัดแรงดันไฟตรง 0-5 V

ช่อง Ain0-Ain3 ของ PCF8591 ตัวที่ 2(IC4) เป็นอินพุตเพิ่มเติมสำหรับวัดแรงดันไฟตรง 0-5 V

แรงดันไฟเลี้ยงของบอร์ดได้มาจากอะแดปเตอร์ขนาด 9-16 V ต่อเข้าที่ J1 โดยมีสวิตช์ S1 ใช้ในการตัดต่อไฟเลี้ยง แรงดันจะผ่านเข้ามายังวงจรเรียงกระแสแบบบริดจ์เพื่อจัดขั่วแรงดันใหม่ทำให้สามารถใช้งานบอร์ดกับอะแดปเตอร์ที่มีการจัดขั้วแรงดันอย่างไรก็ได้ ตัวเก็บประจุ C1 ทำหน้าที่กรองแรงดันให้เรียบขึ้น และ C2 ทำหน้าที่ลดสัญญาณรบกวนความถี่สูงของไฟเลี้ยงวงจรจากนั้นแรงดันจะ

ถูกควบคุมให้คงที่ที่ +5 V สำหรับเลี้ยง IC2-IC4 อุปกรณ์ที่ทำหน้าที่นี้คือ IC1 เบอร์ 7805 ส่วน LED1 ใช้ในการแสดงสถานะการทำงานของวงจร

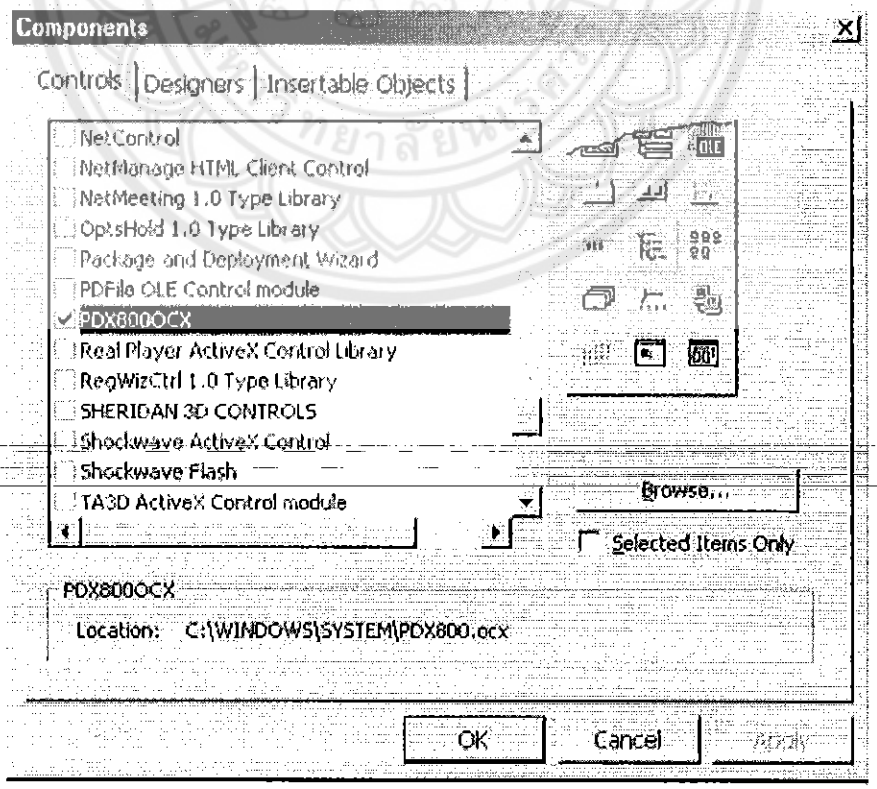
3.4 โปรแกรมสำหรับการแสดงค่าของสัญญาณไฟฟ้าทางคอมพิวเตอร์

ก่อนอื่นต้องทำการติดตั้งไฟล์ซึ่งจำเป็นการเชื่อมต่อระหว่างบอร์ดทดลองกับคอมพิวเตอร์ทางด้านโปรแกรมที่ใช้ในโครงงานนี้พัฒนาขึ้นด้วย Visual Basic

การติดตั้งไฟล์ที่จำเป็นต้องใช้ในการเขียนโปรแกรม มีด้วยกัน 3 ไฟล์คือ pdx_i2c.dll, pdx800.ocx และ io.dll ซึ่งสามารถดาวน์โหลดได้โดยไม่ต้องเสียค่าใช้จ่ายที่ http://www.inex.co.th เมื่อดาวน์โหลดไฟล์มาอยู่บนเครื่องเรียบร้อยแล้วขั้นตอนต่อไปคือทำการคัดลอกไฟล์ทั้งสามจากไดเรกตอรี่ที่เราได้เก็บไฟล์เหล่านี้ไว้ ไปลงในไดเรกตอรี่ System ของ Window โดยทั่วไปไดเรกตอรี่System ของวินโดวส์จะอยู่ที่ C:\WINDOWS\ System ในกรณีใช้วินโดวส์ 2000 และ XP จะอยู่ที่ C:\WINDOWS\ System32

ขั้นตอนการเชื่อมต่อบอร์ดทดลองกับเครื่องคอมพิวเตอร์

1) ต่อบอร์ด PDX-800 เข้ากับพอร์ตขนานของคอมพิวเตอร์
2) เปิดโปรแกรม Visual Basic เลือกเป็น Standard EXE กดปุ่ม Open
3) ไปที่เมนู Project เลือก Components เลือกแท็ป Control จะมีหน้าจอให้เพิ่มเติมคอนโทรลที่ต้องการดังรูป 3.7



รูปที่ 3.7 การเพิ่มคอนโทรล PDX800OCX ให้กับ Visual Basic

4) กดปุ่ม Browse แล้วเลือกไฟล์ pdx800.ocx ในไดเรกตอรี่ C:\WINDOWS\ System กดปุ่ม Open แล้วจะเห็นเครื่องหมายถูกที่หน้าคอนโทรล PDX800OCX แล้วกดปุ่ม OK

ขั้นตอนการออกแบบโปรแกรม

1. การรับสัญญาณ

การรับค่าของสัญญาณจากบอร์ดโดยการวางคอนโทรล PDX800 ลงบนฟอร์ม และสามารถกำหนดคุณสมบัติของคอนโทรลต่างๆ ที่หน้าต่าง Properties ซึ่งจะสังเกตเห็นว่าค่าที่อ่านได้จากบอร์ดยังไม่ใช่ค่าที่แท้จริง แก้ไขโปรแกรมโดยนำค่าที่อ่านได้จากบอร์ดหารด้วย 51.2 แล้วให้โปรแกรมทำงานอ่านค่าจากบอร์ดทุกๆ 1 วินาที ตามค่าคุณสมบัติ Interval ที่ตั้งไว้ในคอนโทรล Timer ซึ่งมีการเขียนโปรแกรมดังต่อไปนี้

```
Private Sub Timer3_Timer()
Dim V_Ch As Single
Dim A  As Single
Timer3.Interval = 1000
V_Ch = PDX1.Ch_V / 51.2
A = PDX1.Ch_Ext1 / 51.2
Text7.Text = Round(V_Ch * 4.61, 3)
 Text8.Text = Round((A * 4.61) / 166.44, 3)
 B = Text8.Text
 Text3.Text = Round(B * (V_Ch * 4.61), 3)
End Sub
```

2. การแสดงผลของสัญญาณในรูปของกราฟ

การแสดงผลในรูปของกราฟแสดงโดย MSChart ซึ่งเป็น Application Object ของโปรแกรม Visual Basic ลักษณะของกราฟที่แสดงเป็นกราฟเส้น โดยการกำหนดค่าเริ่มต้นให้กับ MSChart ดังนี้

```
Private Sub Form_Load()
   MSChart1.ColumnCount = 1
   MSChart1.RowCount = 10
   MSChart1.chartType = VtChChartType2dLine
   MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.Auto = True
```

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร T5615054 C.2

ฮร.
ย142ว
2549

```
   MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.Minimum = 1
   MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.Maximum = 12
   MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.MajorDivision = 10

MSChart2.ColumnCount = 1
   MSChart2.RowCount = 10
   MSChart2.chartType = VtChChartType2dLine
   MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.Auto = True
   MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.Minimum = 0
   MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.Maximum = 10
   MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.MajorDivision = 10

MSChart3.ColumnCount = 1
   MSChart3.RowCount = 10
   MSChart3.chartType = VtChChartType2dLine
   MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.Auto = True
   MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.Minimum = 1
   MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.Maximum = 20
   MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.MajorDivision = 10
   Timer1.Interval = 1000
End Sub
```

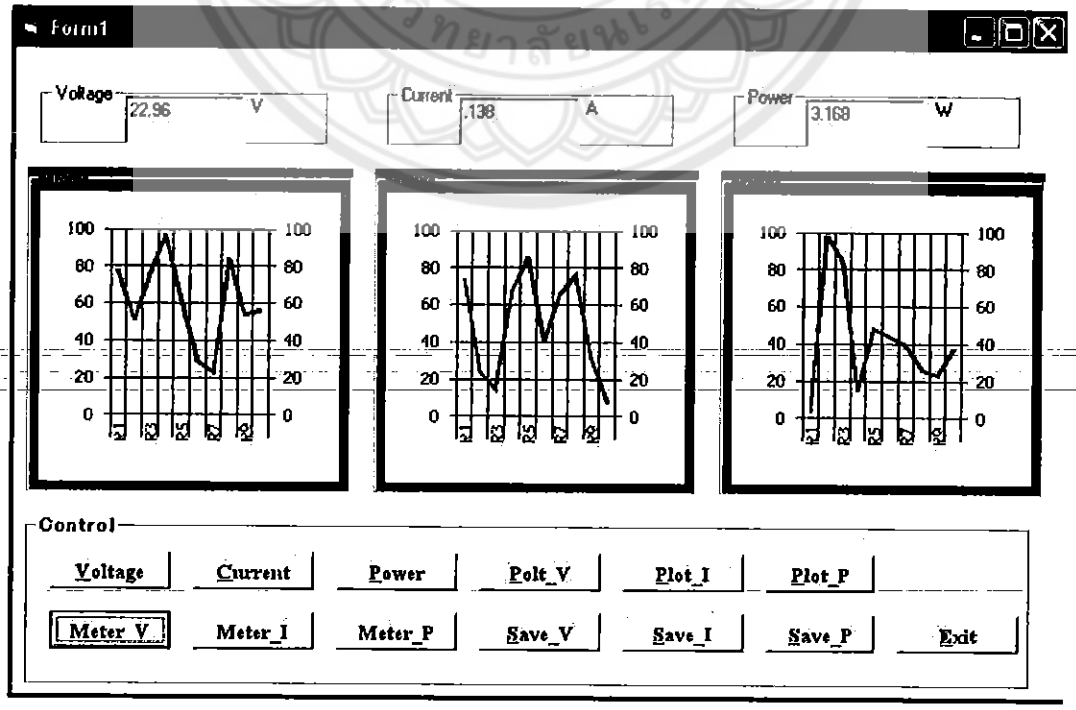
และฟังก์ชันในการแสดงผล โดยเขียนโปรแกรมเพิ่มในเหตุการณ์ Timer ให้นำค่าที่อ่านได้มาเขียนกราฟ โดยทำการเลื่อนข้อมูลทั้งหมดออกไป 1 จุดแล้วนำข้อมูลใหม่มาแทนที่จุดสุดท้ายของกราฟ มีการเขียนโปรแกรมดังนี้

```
Private Sub Timer6_Timer()
Dim V_Ch As Single
Dim A  As Single
Timer3.Interval = 1000
V_Ch = PDX1.Ch_V / 51.2
A = PDX1.Ch_Ext1 / 51.2
```

```
Text7.Text = Round(V_Ch * 4.61, 3)
Text8.Text = Round((A * 4.61) / 166.44, 3)
B = Text8.Text
Text9.Text = Round(B * (V_Ch * 4.61), 3)

For i = 1 To MSChart3.RowCount - 1
    MSChart3.Row = i + 1
    tmp = MSChart3.Data
    MSChart3.Row = i
    MSChart3.Data = tmp
  Next i
  MSChart3.Column = 1
  MSChart3.Row = MSChart3.RowCount
  MSChart3.Data = Text9.Text
End Sub
```

ลักษณะของโปรแกรม

รูปแบบของโปรแกรมแสดงดังรูปที่ 3.8



รูปที่ 3.8 รูปแบบการใช้งาน

บทที่ 4

ผลการทดลอง

4.1 จุดประสงค์การทดลอง

4.1.1 เพื่อทดสอบวงจรปรับสภาพสัญญาณ

4.1.2 เพื่อทดสอบการวัดและการแสดงลักษณะของสัญญาณไฟฟ้าทางคอมพิวเตอร์

4.1.3 เพื่อหาจุดบกพร่อง สาเหตุ วิธี ในการแก้ไขและปรับปรุง

4.2 ขั้นตอนการทดลอง

4.2.1 การทดสอบวงจรปรับสภาพสัญญาณ

วิธีการทดลอง

การทดสอบวงจรปรับสภาพสัญญาณเป็นการทดลองวงจรที่ได้ทำการออกแบบไว้ว่า สามารถทำการปรับแรงดันที่เข้ามาให้มีค่าอยู่ระหว่าง 0-5 โวลท์ และค่ากระแสไฟฟ้าไม่เกิน 10 มิลลิแอมป์

1. ทดลองจ่ายสัญญาณไฟฟ้าขนาด 12 โวลท์ 500 มิลลิแอมป์ แล้ววัดขนาดของสัญญาณไฟฟ้าเป็น 2.609 โวลท์ 1.949 มิลลิแอมป์ หรือไม่
2. ทดลองจ่ายสัญญาณไฟฟ้าขนาด 20 โวลท์ 1 แอมป์ แล้ววัดขนาดของสัญญาณไฟฟ้าเป็น 4.348 โวลท์ 3.898 มิลลิแอมป์ หรือไม่

4.2.2 การทดสอบการวัดและการแสดงลักษณะของสัญญาณไฟฟ้าทางคอมพิวเตอร์

ในขั้นตอนนี้จะทำการวัดหาค่าแรงดันไฟฟ้าและกระแสไฟฟ้าเปรียบเทียบระหว่างค่าที่วัดได้จากมัลติมิเตอร์และค่าที่วัดได้จากระบบวัดที่ทำการออกแบบ แล้วทำการหาค่าเปอร์เซ็นต์ ความผิดพลาด

4.3 ผลการทดลอง

4.3.1 การทดสอบวงจรปรับสภาพสัญญาณ

ผลของการทดสอบวงจรปรับสภาพสัญญาณ มีดังนี้

1) หลังจากทดลองจ่ายสัญญาณไฟฟ้าขนาด 12 โวลท์ 500 มิลลิแอมป์ แล้ววัดขนาดของสัญญาณไฟฟ้า ได้ประมาณ 2.875 โวลท์ 2.051 มิลลิแอมป์
2) หลังจากทดลองจ่ายสัญญาณไฟฟ้าขนาด 20 โวลท์ 1 แอมป์ แล้ววัดขนาดของสัญญาณไฟฟ้าได้ประมาณ 4.298 โวลท์ 3.794 มิลลิแอมป์

4.3.2 การทดสอบการวัดและการแสดงลักษณะของสัญญาณไฟฟ้าทางคอมพิวเตอร์

ตารางที่ 4.1 ผลการวัดค่าแรงดันไฟฟ้า

ค่าที่วัดได้จาก มัลติมิเตอร์	ค่าที่วัดได้จาก เครื่องวัดที่ออกแบบ	ผลต่าง	ค่าเปอร์เซ็นต์ความ ผิดพลาด
12.95 V	13.2 V	0.25	1.93
12.78 V	12.98 V	0.2	1.56
12.95 V	13.0 V	0.05	0.39

ตารางที่ 4.2 ผลการวัดค่ากระแสไฟฟ้า

ค่าที่วัดได้จาก มัลติมิเตอร์	ค่าที่วัดได้จาก เครื่องวัดที่ออกแบบ	ผลต่าง	ค่าเปอร์เซ็นต์ความ ผิดพลาด
109.8 mA	110.0 mA	0.2	0.18
110.0 mA	112.3 mA	2.3	2.09
109.5 mA	109.6 mA	0.1	0.09

4.4 วิเคราะห์ผลการทดลอง

4.4.1 การทดสอบวงจรปรับสภาพสัญญาณ

จากการทดสอบวงจรปรับสภาพสัญญาณ สามารถที่จะทำการลดระดับแรงดันไฟฟ้าและกระแสไฟฟ้าได้ตรงตามที่ได้ออกแบบไว้ แต่อาจคลาดเคลื่อนบางเล็กน้อย เป็นผลจากค่าสูญเสียที่ตัวต้านทาน

4.4.2 การทดสอบการวัดและการแสดงลักษณะของสัญญาณไฟฟ้าทางคอมพิวเตอร์

จากตารางที่ 4.1 และ4.2 จะพบว่าเมื่อทำการวัดค่าแรงดันไฟฟ้าและกระแสไฟฟ้าด้วยมัลติมิเตอร์และเปรียบเทียบกับค่าที่ได้จากโปรแกรม ผลปรากฏว่าเกิดความผิดพลาดน้อยมาก และโปรแกรมสามารถแสดงค่าสัญญาณทางไฟฟ้าได้

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

เครื่องวัดสัญญาณไฟฟ้าโดยแสดงค่าของสัญญาณทางคอมพิวเตอร์ที่ได้ทำการออกแบบมีลักษณะและประสิทธิภาพดังนี้คือ

1) สามารถทำการวัดสัญญาณไฟฟ้าได้ 2 ช่องสัญญาณ
2) สามารถวัดสัญญาณไฟฟ้ากระแสตรงได้
3) สามารถวัดแรงดันไฟฟ้าได้ถึง 20 โวลท์ และกระแสไฟฟ้า 2 แอมป์

5.2 ปัญหาในการทำงานและแนวทางแก้ไข

- ปัญหา หาค่าความต้านทานที่ต้องการใช้ไม่ได้จากร้านผู้จำหน่าย
สาเหตุ ข้อจำกัดของการสั่งสินค้า
ผล ค่าแรงดันไฟฟ้าที่นำไปใช้ไม่ตรงตามที่ได้คำนวณไว้
วิธีแก้ไข สำรวจค่าตัวต้านทานที่จะนำมาใช้ว่ามีขายในร้านหรือไม่

5.3 ข้อเสนอแนะ

5.3.1 การวัดสัญญาณไฟฟ้า

1) ต้องแน่ใจว่าขนาดแรงดันไฟฟ้าที่จะทำการวัดนั้นมีค่าไม่เกินพิกัดที่ได้ตั้งไว้ เพราะจะทำให้อุปกรณ์ภายในเสียหายและใช้การไม่ได้

5.4 แนวทางสำหรับการพัฒนา

5.4.1 ออกแบบให้สามารถวัดสัญญาณได้ทั้งสัญญาณไฟฟ้ากระแสตรงและกระแสสลับ

5.4.2 ออกแบบวงจรให้สามารถวัดแรงดันและกระแสไฟฟ้าที่พิกัดสูงกว่าเดิม

เอกสารอ้างอิง

[1] ฉันทวุฒิ พืชผล และ พิชิต สันติกุลานนท์. **คู่มือเรียน Visual Basic 6**. พิมพ์ครั้งที่ 5. 2544

[2] อภิชาติ ภู่พลับ. **สนุกกับการประยุกต์ใช้ Visual Basic**. พิมพ์ครั้งที่ 1. พฤษภาคม 2546

[3] อภิชาติ ภู่พลับ. **รวมคอนโทรลฟรี [สำหรับ VB6]**. พิมพ์ครั้งที่ 1. ธันวาคม 2547

[4] บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด **"Basic data acquisition experiment board."** [Online]Available: http://www.inex.co.th/electronic/computerinterface.html

ภาคผนวก

ภาคผนวก (ก)

Source code ของโปรแกรม

```
Dim V_Ch As Single
Dim A As Single
Private Sub Command1_Click()
If Timer1.Enabled Then
Timer1.Enabled = False
    Command1.Caption = "&Pause"
  Else
    Timer1.Enabled = True
    Command1.Caption = "&Voltage"
  End If
End Sub
Private Sub Command10_Click()
If Timer5.Enabled Then
Timer5.Enabled = False
    Command10.Caption = "&Pause"
  Else
    Timer5.Enabled = True
    Command10.Caption = "&Plot_I"
    End If
End Sub
Private Sub Command11_Click()
MSChart1.EditCopy
End Sub
Private Sub Command12_Click()
MSChart2.EditCopy
End Sub
Private Sub Command13_Click()
MSChart3.EditCopy
```

```
End Sub
Private Sub Command2_Click()
If Timer2.Enabled Then
Timer2.Enabled = False
    Command2.Caption = "&Pause"
  Else
    Timer2.Enabled = True
    Command2.Caption = "&Current"
  End If
End Sub
Private Sub Command3_Click()
If Timer3.Enabled Then
Timer3.Enabled = False
    Command3.Caption = "&Pause"
  Else
    Timer3.Enabled = True
    Command3.Caption = "&Power"
  End If
End Sub
Private Sub Command4_Click()
If Timer4.Enabled Then
Timer4.Enabled = False
    Command4.Caption = "&Pause"
  Else
    Timer4.Enabled = True
    Command4.Caption = "&Plot_V"
  End If
End Sub
Private Sub Command5_Click()
End
End Sub
Private Sub Command6_Click()
```

```
Form2.Show
End Sub
Private Sub Command7_Click()
Form3.Show
End Sub
Private Sub Command8_Click()
Form4.Show
End Sub
Private Sub Command9_Click()
If Timer6.Enabled Then
Timer6.Enabled = False
    Command9.Caption = "&Pause"
  Else
    Timer6.Enabled = True
    Command9.Caption = "&Plot_P"
  End If
End Sub
Private Sub Form_Load()
  MSChart1.ColumnCount = 1
  MSChart1.RowCount = 10
  MSChart1.chartType = VtChChartType2dLine
  MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.Auto = True
  MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.Minimum = 1
  MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.Maximum = 12
  MSChart1.Plot.Axis(VtChAxisIdY).ValueScale.MajorDivision = 10
MSChart2.ColumnCount = 1
  MSChart2.RowCount = 10
  MSChart2.chartType = VtChChartType2dLine
  MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.Auto = True
  MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.Minimum = 0
  MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.Maximum = 10
```

```
    MSChart2.Plot.Axis(VtChAxisIdY).ValueScale.MajorDivision = 10


MSChart3.ColumnCount = 1
    MSChart3.RowCount = 10
    MSChart3.chartType = VtChChartType2dLine
    MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.Auto = True
    MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.Minimum = 1
    MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.Maximum = 20
    MSChart3.Plot.Axis(VtChAxisIdY).ValueScale.MajorDivision = 10
    Timer1.Interval = 1000


End Sub
Private Sub Timer1_Timer()
Dim V_Ch As Single
Timer1.Interval = 1000
V_Ch = PDX1.Ch_V / 51.2
Text1.Text = Round(V_Ch * 4.61, 3)
End Sub
Private Sub Timer2_Timer()
Dim A As Single
Timer2.Interval = 1000
A = PDX1.Ch_Ext1 / 51.2
Text2.Text = Round((A * 4.61) / 4.98, 3)
End Sub
Private Sub Timer3_Timer()
Dim V_Ch As Single
Dim A  As Single
Timer3.Interval = 1000
V_Ch = PDX1.Ch_V / 51.2
A = PDX1.Ch_Ext1 / 51.2
Text7.Text = Round(V_Ch * 4.61, 3)
 Text8.Text = Round((A * 4.61) / 4.98, 3)
```

```
B = Text8.Text
Text3.Text = Round(B * (V_Ch * 4.61), 3)
End Sub
Private Sub Timer4_Timer()
Dim V_Ch As Single
Dim tmp As Single
Dim i As Single
Timer4.Interval = 1000
V_Ch = PDX1.Ch_V / 51.2
Text4.Text = Round(V_Ch * 4.61, 3)
For i = 1 To MSChart1.RowCount - 1
    MSChart1.Row = i + 1
    tmp = MSChart1.Data
    MSChart1.Row = i
    MSChart1.Data = tmp
  Next i
  MSChart1.Column = 1
  MSChart1.Row = MSChart1.RowCount
  MSChart1.Data = Text4.Text
End Sub
Private Sub Timer5_Timer()
Dim A  As Single
Dim tmp As Single
Dim i As Single
Timer5.Interval = 1000
A = PDX1.Ch_Ext1 / 51.2
Text5.Text = Round((A * 4.61) / 4.98, 3)
For i = 1 To MSChart2.RowCount - 1
    MSChart2.Row = i + 1
    tmp = MSChart2.Data
    MSChart2.Row = i
    MSChart2.Data = tmp
```

```
    Next i
    MSChart2.Column = 1
    MSChart2.Row = MSChart2.RowCount
    MSChart2.Data = Text5.Text
End Sub
Private Sub Timer6_Timer()
Dim V_Ch As Single
Dim A As Single
Timer3.Interval = 1000
V_Ch = PDX1.Ch_V / 51.2
A = PDX1.Ch_Ext1 / 51.2
Text7.Text = Round(V_Ch * 4.61, 3)
 Text8.Text = Round((A * 4.61) / 4.98, 3)
 B = Text8.Text
 Text9.Text = Round(B * (V_Ch * 4.61), 3)

For i = 1 To MSChart3.RowCount - 1
    MSChart3.Row = i + 1
    tmp = MSChart3.Data
    MSChart3.Row = i
    MSChart3.Data = tmp
  Next i
  MSChart3.Column = 1
  MSChart3.Row = MSChart3.RowCount
  MSChart3.Data = Text9.Text
End Sub
```

ภาคผนวก (ข)

ตัวอย่างการสร้างมิเตอร์

การเขียนโปรแกรมแสดงผลในรูปของมิเตอร์ มีขั้นตอนดังนี้

1. ให้ทำการเปิด Project ใหม่โดยคลิกที่ File > New Project ให้เลือก Standard EXE เพื่อเปิดฟอร์มขึ้นมา
2. นำคอนโทรล Line 1 ตัว และคอนโทรล Timer 1 ตัว มาวางลงบนฟอร์ม
3. เขียนโค้ดเพื่อควบคุมการทำงานของโปรแกรมดังนี้

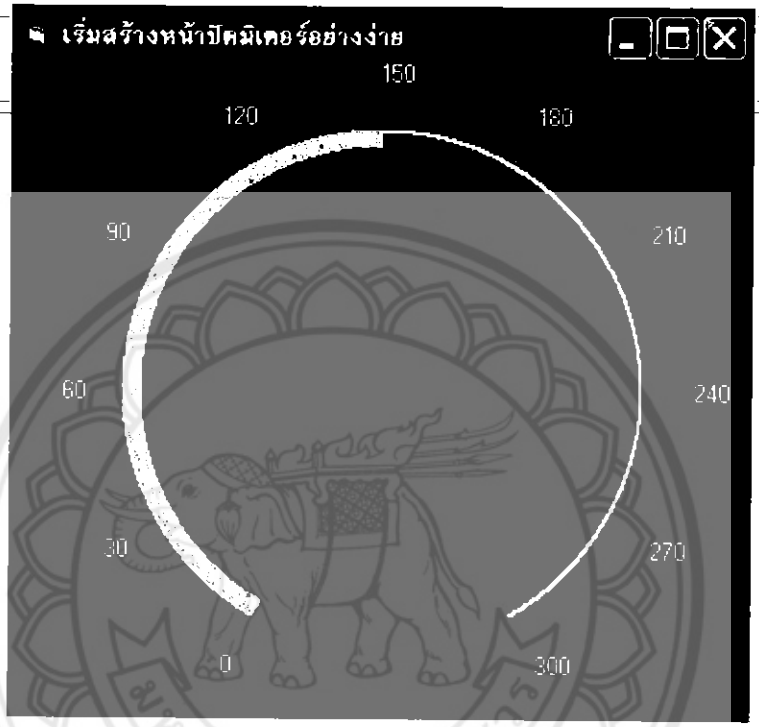
```
Dim d, DialLength, DialLength1 As Integer
Dim n, minr, maxr As Integer
Dim MidX As Integer, MidY As Integer
Const PI = 3.14159

Sub Centre()
   Me.DrawWidth = 10
   MidX = Me.ScaleWidth \ 2
   MidY = Me.ScaleHeight \ 2
   Line1.X1 = MidX
   Line1.Y1 = MidY
   Me.PSet (MidX, MidY), vbRed
End Sub

Private Sub DrawDial()
   Me.Cls
   CurrentX = MidX
   CurrentY = MidY

   If Me.ScaleWidth < Me.ScaleHeight Then
      DialLength = Me.ScaleWidth * 90 / 200
      DialLength1 = Me.ScaleWidth * 80 / 200
```

```
    DialLength2 = Me.ScaleWidth * 75 / 200
Else
    DialLength = Me.ScaleHeight * 90 / 200
    DialLength1 = Me.ScaleHeight * 80 / 200
    DialLength2 = Me.ScaleHeight * 75 / 200
End If

For I = 0 To 10
    Me.DrawWidth = 2
    DialX = DialLength * Cos(PI / 180 * (30 * I - 240)) + MidX
    DialY = DialLength * Sin(PI / 180 * (30 * I - 240)) + MidY
    DialX1 = DialLength1 * Cos(PI / 180 * (30 * I - 240)) + MidX
    DialY1 = DialLength1 * Sin(PI / 180 * (30 * I - 240)) + MidY
    Line (DialX, DialY)-(DialX1, DialY1), vbBlue
    Me.ForeColor = vbWhite
    Me.CurrentX = (DialLength + 180) * Cos(PI / 180 * (30 * I - 240)) + MidX - 80
    Me.CurrentY = (DialLength + 180) * Sin(PI / 180 * (30 * I - 240)) + MidY - 80
    Print I * 30
Next I

For I = 0 To 50 Step 0.01
    Me.DrawWidth = 2
    DialX = DialLength1 * Cos(PI / 180 * (6 * I - 240)) + MidX
    DialY = DialLength1 * Sin(PI / 180 * (6 * I - 240)) + MidY
    PSet (DialX, DialY), vbWhite
Next I

For I = 0 To 25 Step 0.1
    Me.DrawWidth = 2
    DialX = DialLength1 * Cos(PI / 180 * (6 * I - 240)) + MidX
    DialY = DialLength1 * Sin(PI / 180 * (6 * I - 240)) + MidY
    DialX1 = DialLength2 * Cos(PI / 180 * (6 * I - 240)) + MidX
```

```
    DialY1 = DialLength2 * Sin(PI / 180 * (6 * I - 240)) + MidY
    Line (DialX1, DialY1)-(DialX, DialY), vbYellow
  Next I

  For I = 0 To 150
  Circle (MidX, MidY), I, vbRed
  Next I

End Sub

Private Sub Form_Load()
  Me.BackColor = vbBlack
  Me.AutoRedraw = True
  Line1.BorderWidth = 3
  Line1.BorderColor = vbRed
  Timer1.Interval = 1
  Call Centre
'  Call Timer1_Timer
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  Call Centre
  Call DrawDial
End Sub

'Private Sub Timer1_Timer()
'  minr = 145
'  maxr = 160
'  n = minr + Fix(Rnd * (maxr - minr + 1))
'  Line1.X2 = DialLength1 * Cos(PI / 180 * (n - 240)) + MidX
'  Line1.Y2 = DialLength1 * Sin(PI / 180 * (n - 240)) + MidY
```

```
'   Form1.Caption = n
'End Sub
```

4. เมื่อรันโปรแกรม โปรแกรมจะได้ดังรูปที่ ผนวก.1 สังเกตได้ว่าเข็มจะกระดิกตามค่าตัวเลขที่อยู่บน Caption ของฟอร์มซึ่งเกิดจากการสุ่มตัวเลขนั่นเอง



รูปที่ ผนวก.1 ลักษณะของโปรแกรม

คำอธิบาย

จากโค้ดควบคุมโปรแกรม โปรแกรมจะทำงานตามคำสั่งที่อยู่ในอีเวนต์ Form_Load ก่อนซึ่งเป็นการกำหนดคุณสมบัติต่างๆ ให้กับคอนโทรลที่เราสร้างไว้บนฟอร์ม

ส่วนคำสั่งที่อยู่ในอีเวนต์ From_Resize นั้นจะเป็นการจัดสัดส่วนให้กับโปรแกรมเมื่อมีการย่อหรือขยายฟอร์ม

ส่วนคำสั่งที่อยู่ใน Sub_Center() จะเป็นการกำหนดจุดกึ่งกลางของฟอร์มโดยใช้หลักการของความกว้างของฟอร์มหารสอง และความยาวของฟอร์มหารสอง

สำหรับคำสั่งต่างๆ ที่อยู่ใน Sub DrawDial() นั้นจะเป็นการแสดงค่าสเกลต่างๆ บนหน้าจอ ซึ่งเป็นขั้นตอนดังนี้

```
If Me.ScaleWidth < Me.ScaleHeight Then
    DialLength = Me.ScaleWidth * 90 / 200
```

```
    DialLength1 = Me.ScaleWidth * 80 / 200
    DialLength2 = Me.ScaleWidth * 75 / 200
  Else
    DialLength = Me.ScaleHeight * 90 / 200
    DialLength1 = Me.ScaleHeight * 80 / 200
    DialLength2 = Me.ScaleHeight * 75 / 200
  End If
```

คำสั่งข้างต้นเป็นการกำหนดค่าให้กับตัวแปรให้มีขนาด 90 เปอร์เซ็นต์, 80 เปอร์เซ็นต์ และ75 เปอร์เซ็นต์ โดยจะทำการตรวจสอบขนาดความกว้างและความยาวของฟอร์มก่อนดูว่าด้านใดสั้นหรือยาวมากกว่ากัน แล้วจึงทำการกำหนดค่าให้กับตัวแปรต่างๆ เพื่อใช้ในการแสดงผลต่อไป

```
For I = 0 To 10
    Me.DrawWidth = 2
    DialX = DialLength * Cos(PI / 180 * (30 * I - 240)) + MidX
    DialY = DialLength * Sin(PI / 180 * (30 * I - 240)) + MidY
    DialX1 = DialLength1 * Cos(PI / 180 * (30 * I - 240)) + MidX
    DialY1 = DialLength1 * Sin(PI / 180 * (30 * I - 240)) + MidY
    Line (DialX, DialY)-(DialX1, DialY1), vbBlue
    Me.ForeColor = vbWhite
    Me.CurrentX = (DialLength + 180) * Cos(PI / 180 * (30 * I - 240)) + MidX - 80
    Me.CurrentY = (DialLength + 180) * Sin(PI / 180 * (30 * I - 240)) + MidY - 80
    Print I * 30
  Next I
```

คำสั่งต่างๆ ข้างต้นจะเป็นการขีดเส้นแบ่งสเกล โดยจะแบ่งออกเป็น 10 ส่วนและกำหนดค่าตัวเลขให้กับสเกลด้วย (0-300)

```
For I = 0 To 50 Step 0.01
    Me.DrawWidth = 2
    DialX = DialLength1 * Cos(PI / 180 * (6 * I - 240)) + MidX
    DialY = DialLength1 * Sin(PI / 180 * (6 * I - 240)) + MidY
```

```
    PSet (DialX, DialY), vbWhite
  Next I
```

คำสั่งข้างต้นจะเป็นการขีดส้นรอบวงจากสเกลแรกจนถึงสเกลสุดท้าย (0-300)

```
For I = 0 To 25 Step 0.1
    Me.DrawWidth = 2
    DialX = DialLength1 * Cos(PI / 180 * (6 * I - 240)) + MidX
    DialY = DialLength1 * Sin(PI / 180 * (6 * I - 240)) + MidY
    DialX1 = DialLength2 * Cos(PI / 180 * (6 * I - 240)) + MidX
    DialY1 = DialLength2 * Sin(PI / 180 * (6 * I - 240)) + MidY
    Line (DialX1, DialY1)-(DialX, DialY), vbYellow
  Next I
```

คำสั่งข้างต้นจะเป็นการแสดงแถบสีซึ่งเป็นสีเหลือง แสดงจากสเกล 0-150

```
For I = 0 To 150
  Circle (MidX, MidY), I, vbRed
  Next I
```

คำสั่งข้างต้นเป็นการแสดงจุดวงกลมสีแดงที่กึ่งกลางฟอร์ม และ ส่วนคำสั่งที่ใช้สำหรับแสดงตำแหน่งของเข็มและการสุ่มตัวเลขจะอยู่ในอีเวนต์ของ Timer 1

ประวัติผู้เขียนโครงงาน

ชื่อ นายบุญชวน ป่วนไชยดี

ภูมิลำเนา 60/6 ตบ้านเหล่า อ.แม่ใจ จ. พะเยา 56130

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนแม่ใจวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : **phehu@hotmail.com**

ชื่อ นายวสันต์ นันทะสี

ภูมิลำเนา 157/1 หมู่ 4 ต.หลวงเหนือ อ.งาว จ.ลำปาง 52110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนประชารัฐธรรมคุณ
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 5
  สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : **wasan_no.1@hotmail.com**