

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมไร้สาย

Wireless Control System



โดย  
นายบุญมาก เพ็งกัต  
นายพรชัย แหงมมา

เลขหมู่.....  
เลขทะเบียน.....72691  
วัน,เดือน,ปี...2...1...สิ.ย...2550

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมระบบควบคุม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมไร้สาย

Wireless Control System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมไร้สาย  
Wireless Control System

ผู้จัดทำ

- |              |           |              |          |
|--------------|-----------|--------------|----------|
| 1. นายบุญมาก | เพ็งกัตต์ | รหัสประจำตัว | 47015293 |
| 2. นายพรชัย  | แหงมมา    | รหัสประจำตัว | 47015295 |



.....อาจารย์ที่ปรึกษา  
( รศ.ดร. วรพงศ์ ตั้งศรีรัตน์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ระบบควบคุมไร้สาย

นายบุญมาก เพ็งกลัด  
นายพรชัย แหงมมา  
อาจารย์ที่ปรึกษา  
รศ.ดร. วรพงศ์ ตั้งศรีรัตน์  
ปีการศึกษา 2549

## บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบระบบควบคุมกระบวนการแบบไร้สาย ซึ่งได้นำมาประยุกต์ใช้กับระบบควบคุมความเร็วของมอเตอร์ ให้มีความเร็วตามที่กำหนดและมีการทำงานเป็นไปตามที่ต้องการ โดยสามารถแสดงผลการทำงานแบบเวลาจริง ผ่านทางหน้าจอคอมพิวเตอร์ และยังสามารถเก็บบันทึกค่าสถานะการทำงานของมอเตอร์ไว้ในรูปแบบของฐานข้อมูล ทำให้สามารถเรียกดูผลการทำงานย้อนหลังได้ สำหรับการควบคุมและประมวลผลในระบบจะใช้ไมโครคอนโทรลเลอร์ PIC16F877 ซึ่งจะติดต่อสื่อสารข้อมูลกับคอมพิวเตอร์ผ่านทางคลื่นวิทยุความถี่ 2.4 GHz

# Wireless Control System

Mr. Boonmak Pengklad

Mr. Pornchai Ngaemma

Advisor

Assoc. Prof. Dr. Worapong Tangsirat

Academic Year 2006

## Abstract

The wireless control system is proposed in this project. The designed system is applied to the motor speed control system. The status of the process parameters can be displayed real-time through the monitor and also recorded in the database for future access. Microcontroller PIC16F877 is used as a processor to implement the system. The communication between computer and microcontroller PIC is base on the use of radio frequency 2.4GHz.

## กิตติกรรมประกาศ

โครงการในครั้งนี้ที่สำเร็จลุล่วงไปได้ด้วยดีนั้น เพราะได้รับความช่วยเหลือเป็นอย่างดี จึงต้องขอขอบพระคุณ รศ.ดร. วรพงศ์ ตั้งศรีรัตน์ อาจารย์ที่ปรึกษา เป็นอย่างสูง ที่คอยให้คำแนะนำที่เป็นประโยชน์ต่างๆ ในโครงการ รวมทั้งตรวจแก้ไขข้อบกพร่อง ทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี ภาควิชาวิศวกรรมระบบควบคุมที่สนับสนุนค่าใช้จ่ายทั้งหมด ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

อีกทั้งขอขอบคุณ นายคณูชา ประเสริฐสม นักศึกษาปริญญาโท ภาควิชาวิศวกรรมระบบควบคุม ที่ให้คำปรึกษาอย่างเต็มที่ทุกครั้งที่ผู้จัดทำได้ซักถาม และแนะนำแหล่งข้อมูลที่เป็นประโยชน์ในการทำโครงการนี้และเพื่อนๆ ทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ติดลอคมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้



ผู้จัดทำ

นายบุญมาก

นายพรชัย

เพ็งก๊ัด

แหงมมา

# สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 ขอบเขตโครงการ	2
บทที่ 2 ทฤษฎีและหลักการพื้นฐาน	3
2.1 โปรแกรม Visual Basic	3
2.1.1 จุดเด่นของ Visual Basic 6.0	3
2.1.2 การเริ่มใช้งาน โปรแกรม Visual Basic	4
2.1.3 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม	7
2.1.4 คุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้	9
2.2 สถาปัตยกรรม ADO ของ 6.0	10
2.2.1 องค์ประกอบของกลุ่มออบเจกต์ ADO	10
2.3 โปรแกรม Microsoft Office Access 2003	11
2.3.1 ส่วนประกอบต่างๆ ของไฟล์ฐานข้อมูลใน Access 2003	12
2.4 ไมโครคอนโทรลเลอร์	13
2.5 มอเตอร์กระแสตรง	13
2.5.1 โมเดลอิเล็กทรอนิกส์โทรแมคคานิคอล	14
2.6 การควบคุมความเร็วของมอเตอร์ขั้นพื้นฐาน	17
2.6.1 การควบคุมด้วยตัวต้านทานที่ปรับค่าได้	17
2.6.2 การควบคุมด้วยวิธีเปลี่ยนค่าแรงดัน	19
2.6.3 การควบคุมแบบ PWM (Pulse Width Modulation)	20
2.7 วงจรขับมอเตอร์	21
2.8 เอ็นโค้ดเดอร์	22
2.9 วงจรนับ 74HC393	25

## สารบัญ (ต่อ)

	หน้า
2.10 การอนุรักษ์ตัวควบคุม พีไอดี ด้วยไมโครคอนโทรลเลอร์	27
2.10.1 องค์ประกอบ P	27
2.10.2 องค์ประกอบ I	27
2.10.3 องค์ประกอบ D	28
2.11 การรับส่งข้อมูลแบบไร้สาย	29
2.11.1 TRW 2.4G	29
<b>บทที่ 3 การออกแบบและการทำงาน</b>	31
3.1 ภาพรวมของโครงการ	31
3.1.1 ส่วนควบคุมและสั่งงาน	31
3.2 การสร้างและออกแบบจอภาพ	31
3.3 การออกแบบหน้าต่างในการใช้งาน	33
3.4 การส่งผ่านข้อมูล	38
3.5 การควบคุมการทำงาน	38
3.5.1 การออกแบบและทดสอบ	40
3.5.2 การอ่านค่าความเร็วรอบของมอเตอร์จากเอ็นโค้ดเดอร์และไอซีเคาน์เตอร์	41
<b>บทที่ 4 การทดลอง</b>	43
4.1 การสั่งงานการทำงาน	43
4.1.1 การเลือกค่าอัตราบอดทของข้อมูล	43
4.1.2 การเลือกค่าความเร็วของมอเตอร์	44
4.1.3 การตั้งค่าสถานะต่างๆของการทำงาน	45
4.2 ผลการทดสอบการเปลี่ยนค่าพารามิเตอร์ของตัวควบคุม	47
<b>บทที่ 5 บทวิจารณ์และสรุป</b>	53
5.1 บทวิจารณ์	53
5.2 ปัญหาในการทำงาน	54
5.3 สรุปผลการทำงาน	54
<b>ภาคผนวก</b>	55
ภาคผนวก ก คู่มือไอซี ( Data sheet )	56
ภาคผนวก ข วงจรการทำงาน	66
<b>เอกสารอ้างอิง</b>	72

## สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางการทำงานของไอซีเบอร์ 74HC393	26
3.1 การกำหนดการเพิ่มโหลดด้วยบิท 13-15	41
4.1 ผลการทดลองที่ความรอบ 800 รอบต่อนาที เมื่อ $K_p = 10$ และ $K_i = 80$	47
4.2 ผลการทดลองที่ความรอบ 800 รอบต่อนาที เมื่อ $K_p = 2$ และ $K_i = 80$	49
4.3 ผลการทดลองเปลี่ยนความรอบ 1600 เป็น 1000 รอบต่อนาที	51



# สารบัญรูป

รูปที่	หน้า
1.1 บล็อกไดอะแกรมของโครงการระบบควบคุมไร้สาย	2
2.1 หน้าต่าง New Project	5
2.2 โปรแกรม Visual Basic	6
2.3 หน้าต่าง Components	7
2.4 การใช้งาน Serial Port	8
2.5 Provider แต่ละชนิดของ ADO	10
2.6 แบบจำลองของออบเจกต์ 3 ตัว ของ ADO	11
2.7 ส่วนประกอบต่างๆ ของโปรแกรม	12
2.8 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น	14
2.9 แรงบิดที่เกิดขึ้นต่อ โทลคของมอเตอร์	16
2.10 บล็อกไดอะแกรมของดีซีมอเตอร์	17
2.11 วงจรควบคุมความเร็วของมอเตอร์กระแสตรงแบบใช้ตัวต้านทานอนุกรม และกราฟแสดงคุณสมบัติ	18
2.12 การควบคุมความเร็ว โดยเปลี่ยนค่าแรงดัน	19
2.13 สัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆ กัน	20
2.14 โครงสร้าง H bridge	21
2.15 การควบคุมแบบ Sign/Magnitude	22
2.16 Rotary Encoder	23
2.17 การสร้าง pulse ของ Rotary Encoder	23
2.18 พัลส์ของ Rotary แบบสองเฟส	24
2.19 การตรวจเช็คสัญญาณพัลส์ของ Rotary Encoder	24
2.20 ลักษณะพัลส์ของ Rotary Encoder ทั้ง สองเฟส	25
2.21 การจัดวางขาของไอซีวงจรมับ	25
2.22 บล็อกไดอะแกรมของตัวควบคุม PID	29
3.1 หน้าต่างการทำงานเบื้องต้น	32
3.2 การกำหนดพอร์ตอนุกรม RS 232	33
3.3 การเลือกโหมดการทำงาน	34
3.4 หน้าต่างโปรแกรมหลักในการควบคุม	35

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.5 โพลีชาร์ตการทำงานการส่งค่าของหน้าต่างควบคุมหลัก	36
3.6 โพลีชาร์ตการรับค่าการทำงานเพื่อแสดงผล	37
3.7 บล็อกไดอะแกรมของการรับและส่งข้อมูล	38
3.8 โพลีชาร์ตแสดงการทำงานของไมโครคอนโทรลเลอร์	39
3.9 ส่วนควบคุมกระบวนการ	40
3.10 รูปแบบการส่งข้อมูล 16 บิต	40
4.1 การเลือกค่าอัตราบอดเรต	43
4.2 การเลือกค่าความเร็วของมอเตอร์	44
4.3 ผลของการวัดค่าการทำงานเวลาปัจจุบัน	44
4.4 เลือกโหมดการบันทึกค่าสถานะ	45
4.5 การบันทึกค่าสถานะต่างของมอเตอร์	45
4.6 กำหนดวันที่ต้องการดูสถานะการทำงาน	46
4.7 สถานะการทำงานที่เก็บในฐานข้อมูล Access 2003	46
4.8 ผลการตอบสนองของระบบ เมื่อ $K_p = 10$ และ $K_i = 80$	48
4.9 กราฟผลตอบสนองของระบบ เมื่อ $K_p = 2$ และ $K_i = 80$	50
4.10 กราฟผลตอบสนองของระบบที่เปลี่ยนความรอบ 1600 เป็น 1000 รอบต่อนาที	52

## VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

ปัญหาอย่างหนึ่งที่เกิดขึ้นมาก ในการเชื่อมโยง ติดต่อสื่อสารระหว่าง จุดสองจุด หรือหลายๆจุด ซึ่งมีระยะทางที่ไกลกัน นั่นก็คือการที่จะต้องมีการเดินสาย เพื่อเป็นสายสัญญาณ ในการติดต่อสื่อสาร ระหว่างกันและกัน การควบคุม ไร้สายจึงเป็นส่วนหนึ่ง ที่เข้ามาช่วยอำนวยความสะดวก และจัดการกับ ปัญหาที่เกี่ยวกับการที่ต้องเดินสายสัญญาณ ได้

ในโครงการนี้ได้นำเสนอการออกแบบระบบควบคุมแบบ ไร้สาย เพื่อทำการควบคุมการทำงานของมอเตอร์ โดยจะส่งค่าเป้าหมาย (ความเร็วที่ต้องการ) ทิศทางการหมุน และ โหลดที่ต้องการ (เพื่อทดสอบว่ามอเตอร์สามารถหมุนด้วยความเร็วตามเป้าหมายได้) ผ่านทางหน้าจอกอมพิวเตอร์ และมีการบันทึกการทำงานเก็บเป็นฐานข้อมูล และสามารถที่จะเรียกดูข้อมูลการทำงานของมอเตอร์ย้อนหลังได้ การติดต่อระหว่างคอมพิวเตอร์ที่ใช้ส่งข้อมูลและรับข้อมูลมาแสดงผลจะถูกแยกจากส่วนควบคุม ซึ่งจะ เป็นการติดต่อแบบ ไร้สาย โดยจะส่งข้อมูลผ่านทางตัวกลางอากาศ เช่นเดียวกันส่วนควบคุมเมื่อได้รับ ข้อมูลและทำการประมวลผลแล้วจะส่งข้อมูลไปแสดงผลที่คอมพิวเตอร์โดยผ่านตัวกลางอากาศเช่นกัน

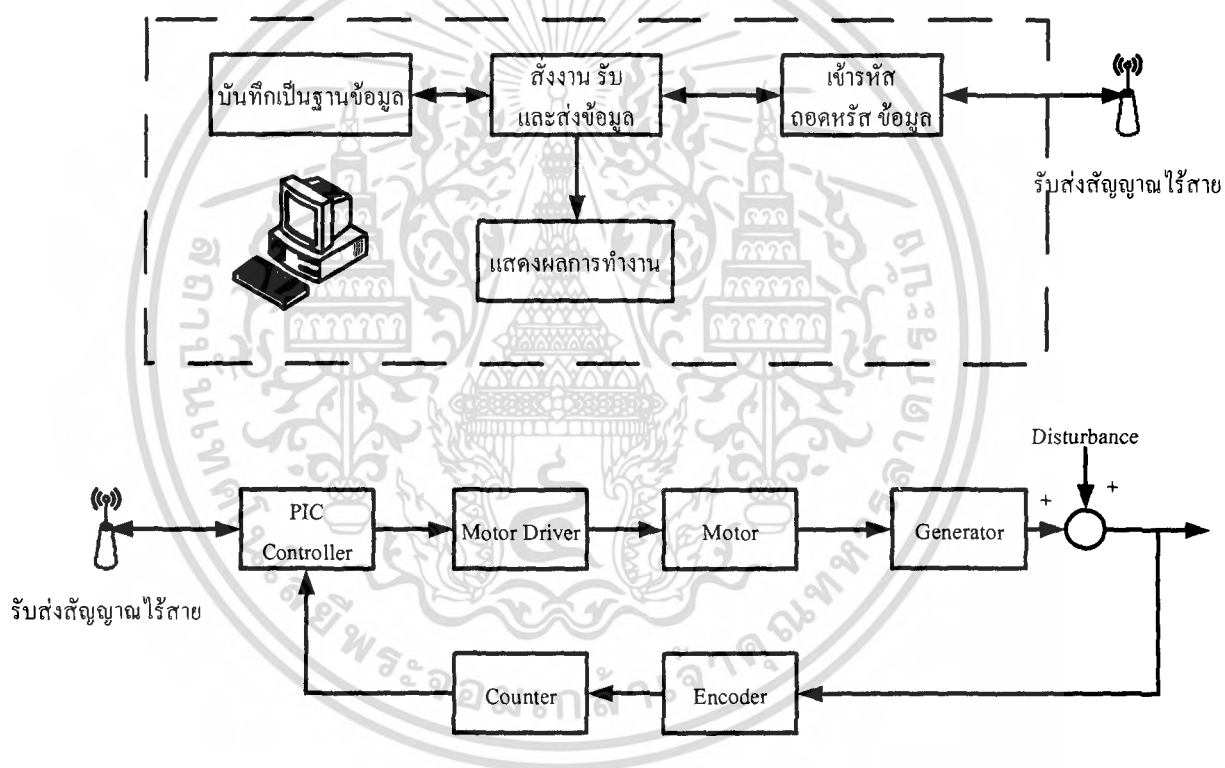
### 1.1 วัตถุประสงค์

- 1) เพื่อศึกษาการส่ง รับ สัญญาณผ่านทางคลื่นวิทยุ
- 2) เพื่อศึกษาการเก็บบันทึกข้อมูล ไว้เป็นฐานข้อมูล
- 3) เพื่อให้ได้ชุดควบคุมมอเตอร์ ไร้สาย
- 4) เป็นการศึกษาการควบคุมวงปิด
- 5) เป็นการศึกษาการควบคุมดิจิทัล
- 6) เพื่อนำความรู้และทฤษฎีที่ได้ศึกษาไปแล้วมาประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 ขอบเขตโครงการ

- 1) การควบคุมติดต่อสื่อสารรับ ส่งข้อมูลผ่านทางพอร์ตอนุกรม และส่งผ่านตัวกลางเป็นอากาศ โดยผ่านทางความถี่วิทยุ 2.4 GHz
- 2) สามารถควบคุมความเร็วของมอเตอร์ให้เป็นไปตามค่าเป้าหมายที่รับมา
- 3) เมื่อมีการรบกวนทำให้ความเร็วของมอเตอร์เปลี่ยนไป สามารถที่จะทำให้ความเร็วกลับมาที่ความเร็วที่ต้องการได้
- 4) สามารถสั่งงานและแสดงผลการทำงาน แบบเวลาจริง
- 5) ทำการบันทึกค่าการทำงาน สถานะต่างๆ เก็บไว้เป็นฐานข้อมูล และสามารถเรียกดูข้อมูลการทำงานต่างๆ ย้อนหลังได้



รูปที่ 1.1 บล็อกไดอะแกรมของโครงการระบบควบคุมไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและหลักการพื้นฐาน

### 2.1 โปรแกรม Visual Basic

ในโครงการนี้เลือกใช้โปรแกรม Visual Basic 6.0 เนื่องจาก เป็นเครื่องมือที่ใช้เขียนโปรแกรมบน Windows ที่ได้รับความนิยมสูง และนอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic 6.0 ยังมีเครื่องมือช่วยในการเขียนโปรแกรมเป็นเรื่องที่ไม่ยุ่งยาก เพราะมีเครื่องมือช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยากตรวจสอบได้อัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์ลดลงได้มาก

#### 2.1.1 จุดเด่นของ Visual Basic 6.0

Visual Basic 6.0 เป็นเครื่องมือและภาษาสำหรับการเขียนโปรแกรมที่ได้รับความนิยมสูงสุด มีผู้ใช้งานมากที่สุด ซึ่งจุดเด่นที่ Visual Basic 6.0 มีเหนือกว่าภาษาเขียนโปรแกรมอื่นๆ คือ

1) **สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว** Visual Basic 6.0 ได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยให้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอปพลิเคชันให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Repaid Application Development หรือ RAD

ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ซากๆ ออกไป, ขจัดสิ่งที่ไม่จำเป็นต้องรู้เกี่ยวกับการควบคุมฮาร์ดแวร์ และการจัดการภายในของ Windows ออกไปให้เหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาของงานจริงๆ แล้วเขียนโปรแกรมจัดการกับปัญหานั้นๆ ส่วนเรื่องอื่นๆ ปลดปล่อยให้ Visual Basic จัดการ

2) **ภาษาเขียนโปรแกรมที่ง่ายต่อการเริ่มเรียนรู้** ถ้าได้มีโอกาสเขียนโปรแกรมด้วย Visual Basic 6.0 แล้วจะเห็นว่า ภาษา Basic ใน Visual Basic 6.0 นั้นอ่านง่ายคือ อ่านแล้วใกล้เคียงภาษาที่เราใช้งานปกติ อ่านแล้วสื่อความหมายได้เข้าใจง่ายกว่าภาษาโปรแกรมอื่นๆ ทำให้ผู้เริ่มต้นเขียนโปรแกรมทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

แม้บรรดานักโปรแกรมหลายๆ คนจะค่อนข้างคิดว่าภาษา Basic ใน Visual Basic 6.0 มีโครงสร้างไม่ค่อยดีนักเมื่อเทียบกับภาษาปาสคาล (Pascal) หรือภาษาจาวา (Java) แต่ต้องไม่ลืมว่าภาษา Basic ใช้เวลาในการเรียนรู้สั้นกว่า จึงง่ายต่อการเรียนรู้และการเข้าใจของคนทั่วไป

3) **รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม** นอกจากจะง่ายต่อการเรียนรู้ แล้วโปรแกรม ยังมีเครื่องมือช่วยในการเขียนโปรแกรมเป็นสิ่งที่ไม่ยุ่งยาก เพราะจะมีเครื่องมือช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยาก ตรวจสอบได้อัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์ลดลงได้มาก

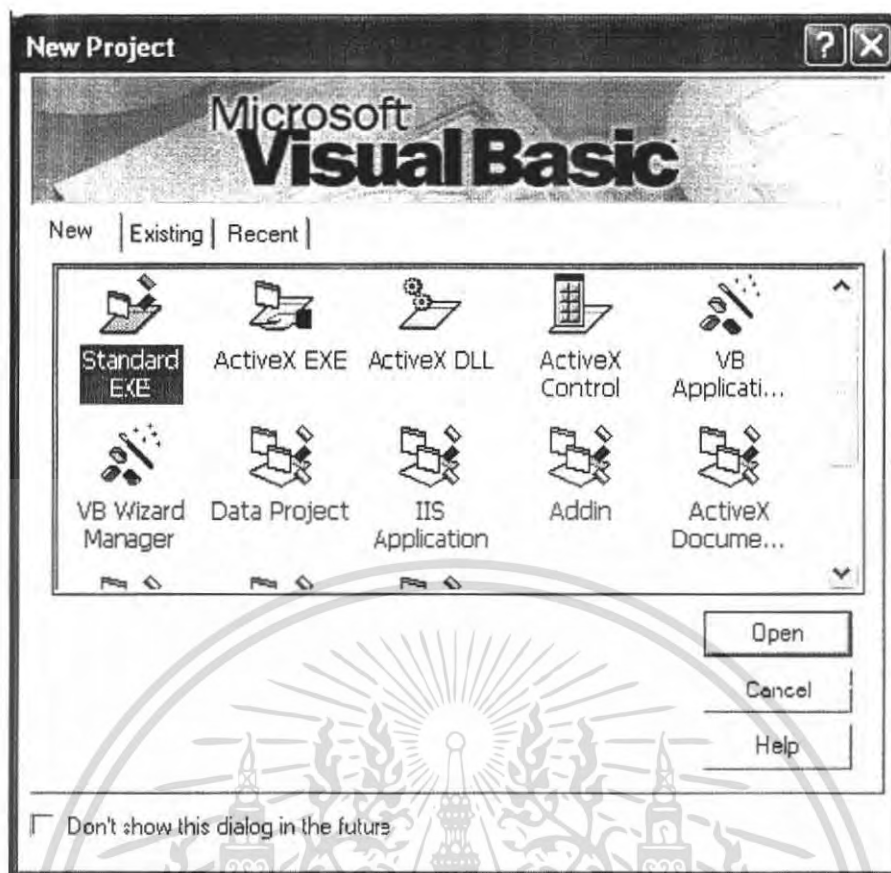
นอกจากจะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้ว ยังมีเครื่องมือใช้ทดสอบและแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิง และขอความช่วยเหลือในจุดที่เราข้องใจ

เครื่องมือที่กล่าวมาทั้งหมดที่กล่าวมาจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อว่า IDE ซึ่งมาจาก Integrated Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม , ทดสอบ , แก้ไข , สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปได้เรื่อยๆ หรือถอดเครื่องมือที่ไม่จำเป็นออกเพื่อประหยัดเนื้อที่ฮาร์ดดิสก์ก็ได้เช่นกัน

4) **มีการพัฒนา ActiveX Control ให้มีการใช้งานหลายด้าน** เช่น การที่ Visual Basic ได้รับความนิยมนอย่างสูง เนื่องจากมีบุคคลหรือบริษัททางด้านพัฒนาซอฟต์แวร์ได้สร้าง ActiveX Control ขึ้นมาใช้งานกับ Visual Basic มากมาย ครอบคลุมการทำงานหลายด้าน เช่น ฮาร์ดแวร์ กราฟิกฐานข้อมูล การจัดการเครือข่าย อินเทอร์เน็ต ฯลฯ ซึ่งก็มีทั้งในรูปของฟรีแวร์ แชร์แวร์ และซอฟต์แวร์ที่ขายกันจริงๆ ซึ่งถือเป็นเหตุผลสำคัญที่ทำให้ Visual Basic ยังคงถูกใช้งานแพร่หลายมายาวนาน

### 2.1.2 การเริ่มใช้งานโปรแกรม Visual Basic

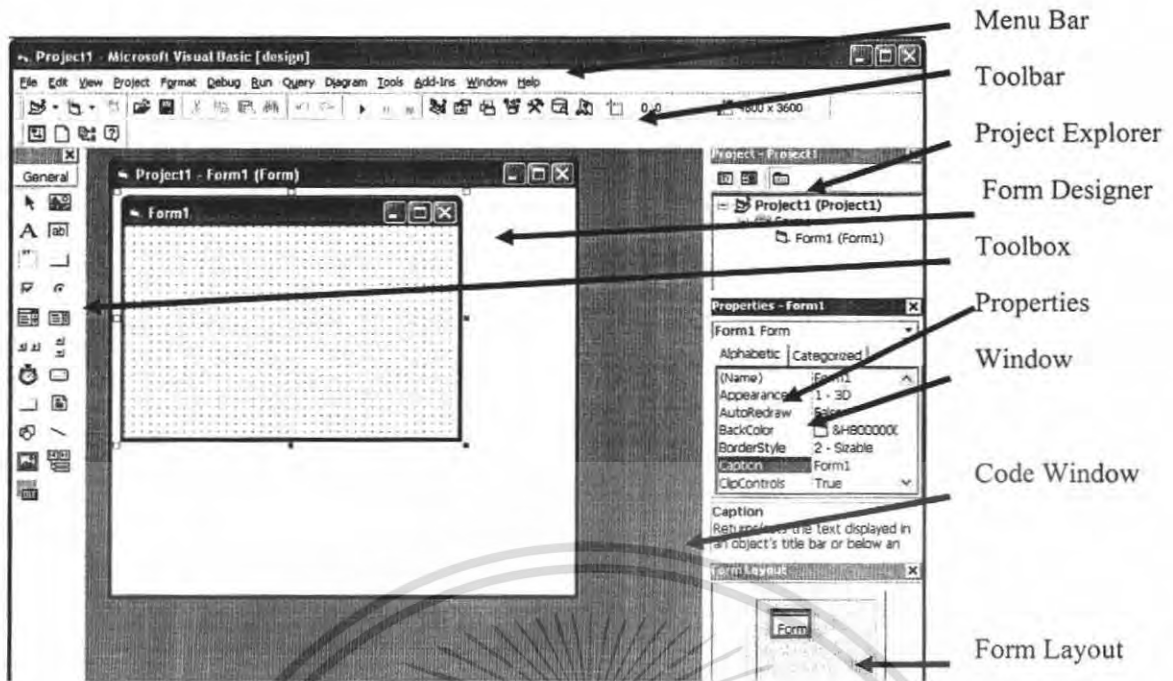
สิ่งแรกเมื่อเข้าสู่โปรแกรม Microsoft Visual Basic 6.0 ได้แก่ จอภาพที่ใช้สำหรับเปิด Project (Project จะเป็นชื่อเรียกแทนระบบงานที่พัฒนาขึ้นด้วย Visual Basic)



รูปที่ 2.1 หน้าต่าง New Project

ในเบื้องต้นให้คลิก Icon “Standard EXE” ใน Tab “New” เพื่อเข้าสู่หน้าจอของ Visual Basic ที่ใช้ในการพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โปรแกรม Visual Basic

**-Menu Bar**

เป็นส่วนที่รับข้อมูลในแบบเมนู เมื่อเราสร้างแอปพลิเคชันด้วย Visual Basic เป็นเสมือนศูนย์กลางที่ควบคุมการสร้างแอปพลิเคชัน

**- Toolbar**

ในการใช้งานเมนูบาร์สั่งงานอาจมีขั้นตอนที่ยุ่งยาก เพื่อลดขั้นตอนลงเราจะคลิกที่ทูลบาร์เพียงครั้งเดียว ก็สามารถสั่งงานที่เราต้องการได้ (เป็นเหมือนลัดในการทำงาน)

**- Toolbox**

เป็นกล่องเก็บ ActiveX Control ซึ่งเราจะนำมาประกอบเป็นส่วนต่างๆ ของแอปพลิเคชัน

**- Project Explorer**

เป็นเครื่องมือที่ใช้ควบคุมการทำงานของโปรเจกต์

**- Properties Window**

เป็นส่วนที่กำหนดหรือเพอร์ตีให้กับออบเจกต์ต่างๆ ในแอปพลิเคชัน

**- Form Layout**

เป็นหน้าต่างร่างๆ ของฟอร์มที่ได้จากการรันแอปพลิเคชัน ทำให้เราสามารถทราบตำแหน่งที่ปรากฏบนจอภาพเมื่อแอปพลิเคชันทำงาน

**-Form Designer**

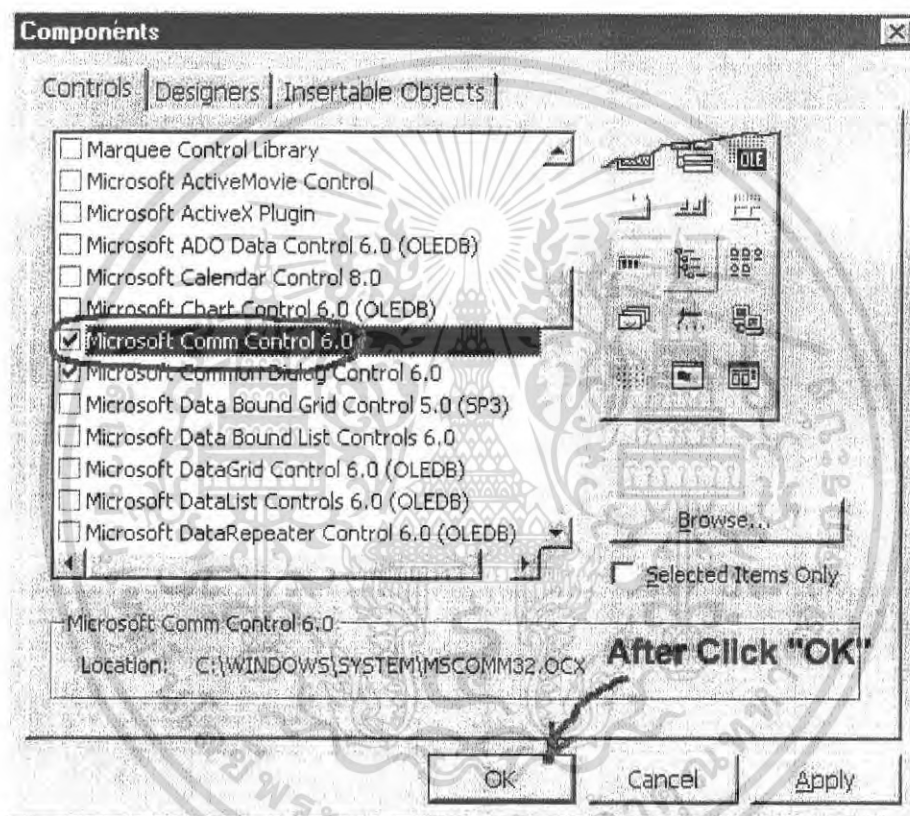
เป็นส่วนที่เรามองเห็นในขณะที่ออกแบบแอปพลิเคชันของ Visual Basic ซึ่งเราจะออกแบบหน้าต่างของแอปพลิเคชันผ่านฟอร์มดีไซน์เนอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**-Code Window** เป็นส่วนที่เราเขียนโปรแกรม (เรียกสั้นๆ ว่าเขียนโค้ด) เพื่อควบคุมการทำงานของแอปพลิเคชัน

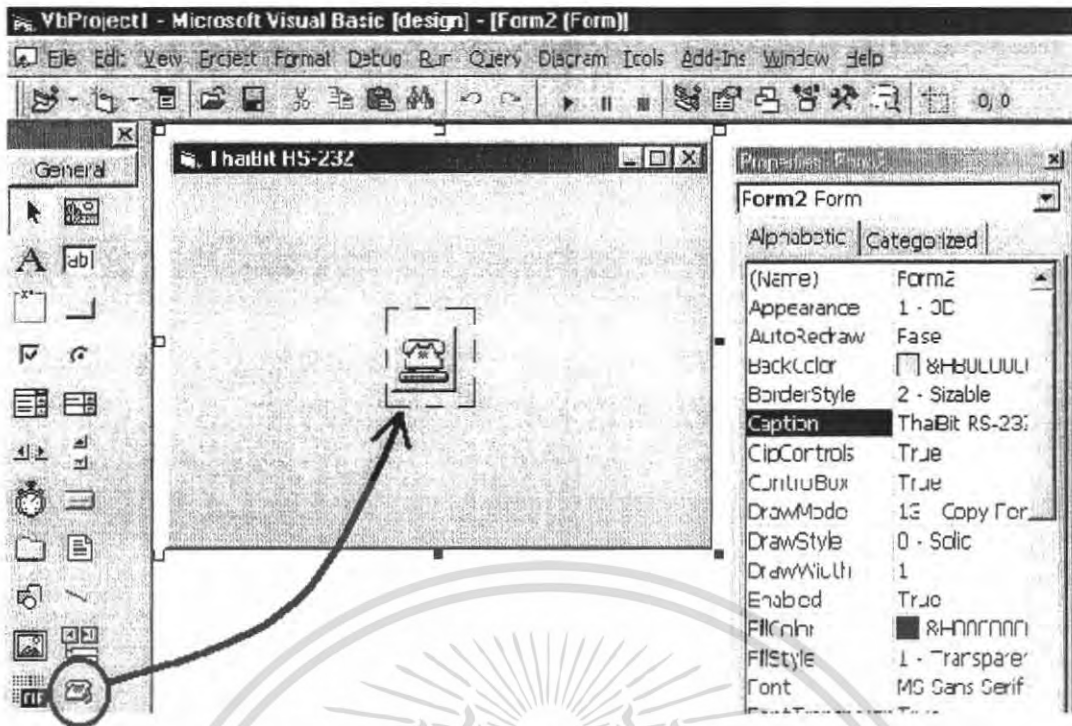
### 2.1.3 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอเนกตรม

สำหรับการใช้งาน Visual Basic ตั้งแต่เวอร์ชัน 2 เป็นต้นมา Visual Basic จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอเนกตรมผ่านทางพอร์ตอเนกตรมของคอมพิวเตอร์มาให้ โดยทำการเลือกเมนู Project → Components แล้วจะปรากฏหน้าต่าง Components ขึ้นมา จากนั้นทำการเลือกไปที่ .Microsoft Comm Control 6.0



รูปที่ 2.3 หน้าต่าง Components

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การใช้งาน Serial Port

การเขียนโปรแกรมติดต่อกับSerial Port สามารถทำได้ 2 วิธี คือ

1) การติดต่อแบบอินเทอร์รัพต์ ขบวนการอินเทอร์รัพต์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่อุปกรณ์อื่น ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงการอุปกรณ์ จะส่งเป็นรหัสแอสกี เราจะเขียน โปรแกรมอินเทอร์รัพต์ โดยเมื่อที่ข้อมูลเข้ามาที่จะทำให้มีCommEventกับOnCommEvent

2) การติดต่อแบบโพลลิง ในระบบพีซี การโพลมบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ ต้องส่งเข้ามา เรียกว่า "Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ในVBเราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียน โปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ มิลลิวินาทีหรือจะใช้ Do.....Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพต์ โดยการเขียนโปรแกรมติดต่อกับ Serial Port แบบธรรมดาจะใช้ comEvent เพียง comEvReceive,comEvSend ถ้าเป็นการติดต่อสื่อสารแบบ โมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ

#### 2.1.4 คุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

1. Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน

ตัวอย่าง MSComm1.CommPort=1

2. Property ชื่อ Settings คือ การตั้งค่าของการรับส่งข้อมูลซึ่งจะต้องรู้ด้วยว่าอัตราบอดเรต ของอุปกรณ์ที่จะติดต่อก็คือเป็นเท่าไร โดยมีรายละเอียดการใส่ต่างๆค่าดังนี้  
MSComm1.Settings="Baud (อัตราการรับส่งข้อมูล) Parity (ถ้าไม่ใช่ N,จำนวนบิตข้อมูล,บิตสต๊อป"

ตัวอย่าง MSComm1.Settings="1200,N,8,1"

3. Property ชื่อ InputLen คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่าน ข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์

ตัวอย่าง MSComm1.InputLen=1

4. Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด =True ถ้าปิด =False

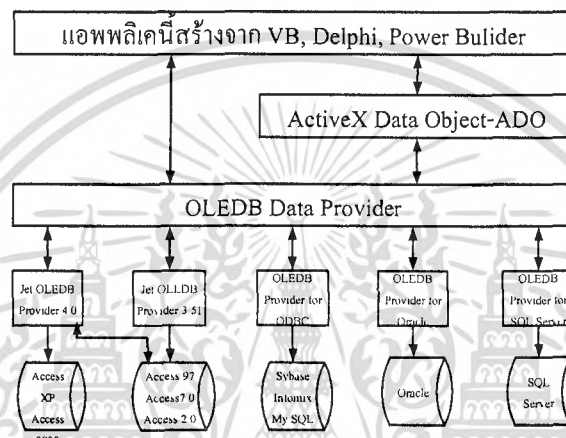
ตัวอย่าง MSComm1.PortOpen=True

5. Property ชื่อ Rthreshold คือทำให้เกิดการกระตุ้นด้วย Event-drivenเมื่อมีข้อมูลบัฟเฟอร์รับข้อมูล(Comport)มันทำให้เกิดCommEvent ใน OnComm Event

ตัวอย่าง MSComm1.Rthreshold =1

## 2.2 สถาปัตยกรรม ADO ของ 6.0

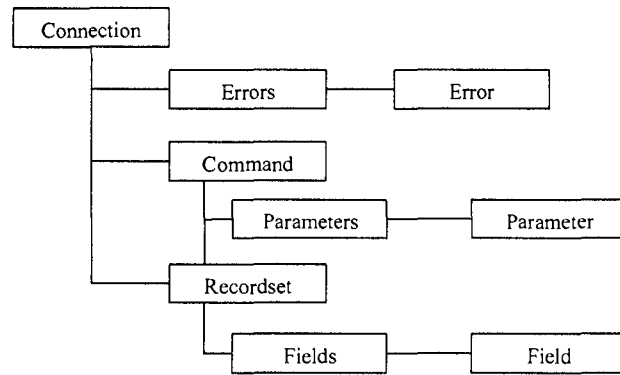
การเขียนฐานข้อมูลของ VB 6.0 จะอาศัยสถาปัตยกรรม ActiveX Data Object (ADO) ทำหน้าที่เชื่อมต่อแหล่งข้อมูลประเภทต่างๆ โดยอาศัย Provider แต่ละชนิดเชื่อมต่อกับแหล่งข้อมูลแต่ละประเภท ในการเก็บข้อมูลในฐานข้อมูลเราใช้งานกลุ่มออบเจกต์ ADO โดยตรง สำหรับการใช้งานแบบนี้ถือเป็นวิธีที่เหมาะสมในการพัฒนาระบบงานฐานข้อมูล แต่ว่าจะต้องมีการที่จะต้องเขียนโค้ดเพื่อควบคุมการทำงานของออบเจกต์แต่ละตัว แต่มีข้อดีคือสามารถควบคุม แก้ไข ได้สะดวก และยืดหยุ่น ในการที่จะติดต่อกับฐานข้อมูลนั้นจะต้องทำการเขียน ชุดคำสั่ง SQL ที่จะเป็นการติดต่อกับฐานข้อมูลที่เราต้องการ ซึ่งจะสามารถ เพิ่ม ลบ แสดง ข้อมูลตามต้องการ



รูปที่ 2.5 Provider แต่ละชนิดของ ADO

### 2.2.1 องค์ประกอบของกลุ่มออบเจกต์ ADO

ออบเจกต์ของสถาปัตยกรรม ADO จะมีออบเจกต์ที่เป็นแกนหลักอยู่ 3 ตัว คือ ออบเจกต์ Connection, Command และออบเจกต์ Recordset ลักษณะที่สำคัญที่สุดของออบเจกต์ทั้ง 3 ตัวก็คือ การทำงานของออบเจกต์แต่ละตัว สามารถทำงานแยกอิสระจากกัน หรือทำงานด้วยกันได้ ดังรูป 2.6



รูปที่ 2.6 แบบจำลองของออบเจกต์ 3 ตัว ของ ADO

ADO โดยที่ออบเจกต์แต่ละตัวรับผิดชอบหน้าที่ของตัวเอง กล่าวคือ

- 1) เชื่อมต่อกับแหล่งข้อมูล หรือฐานข้อมูลด้วยออบเจกต์ Connection
- 2) คิวรีข้อมูลออกมาด้วยออบเจกต์ Command
- 3) เก็บข้อมูลที่คิวรีออกมาไว้ที่ออบเจกต์ Recordset

แต่เราสามารถใช้งานออบเจกต์แต่ละตัว ให้แตกต่างจากหน้าที่ที่มันรับผิดชอบอยู่ได้เช่นกัน ซึ่งเป็นคุณสมบัติของออบเจกต์ที่มีอยู่ในสถาปัตยกรรม ADO ข้อมูลที่ทำการบันทึกลงไปนั้นจะถูกจัดเก็บโดยโปรแกรม Microsoft Office Access

### 2.3 โปรแกรม Microsoft Office Access 2003

Microsoft Office Access 2003 เป็นโปรแกรมฐานข้อมูลที่นิยมใช้กันอย่างแพร่หลาย เนื่องจาก Access 2003 เป็นโปรแกรมฐานข้อมูลที่มีความสามารถในหลายๆ ด้าน ใช้งานง่าย ซึ่งผู้ใช้สามารถเริ่มทำได้ตั้งแต่การออกแบบฐานข้อมูลจัดเก็บข้อมูล เขียน โปรแกรมควบคุม ตลอดจนการทำรายงานแสดงผลของข้อมูล

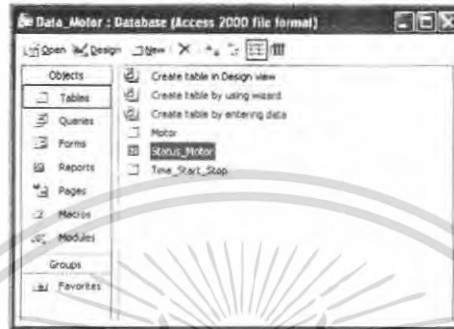
Access 2003 เป็นโปรแกรมฐานข้อมูลที่ใช้งานง่าย โดยผู้ใช้ไม่จำเป็นต้องมีความเข้าใจ ในการเขียนโปรแกรมก็สามารถใช้งาน ได้โดยที่ไม่จำเป็นต้องศึกษารายละเอียด ในการเขียนโปรแกรมให้ยุ่งยาก และสำหรับนักพัฒนาโปรแกรมมีอาชีพนี้ Access 2003 ยังตอบสนองความต้องการในระดับสูงขึ้นไปอีกเช่น การเชื่อมต่อระบบฐานข้อมูลกับฐานข้อมูลอื่นๆ เช่น SQL SERVER, ORACLE หรือแม้แต่การนำข้อมูลออกสู่เครือข่ายอินเทอร์เน็ตก็สามารถทำได้

ในโครงการนี้ได้นำ Access 2003 มาใช้ในการเก็บบันทึกข้อมูลต่างๆ ของมอเตอร์ที่เราได้ทำการควบคุมเพื่อเก็บไว้เป็นข้อมูลดิบ และสามารถที่จะดึงข้อมูลดิบนั้นๆ ออกมาแสดงได้ตามความ

ต้องการ ในการที่จะเก็บข้อมูลต่างๆ นั้นจะต้องทำการออกแบบตารางที่ใช้เก็บข้อมูล ในโปรแกรม Access 2003 เสียก่อน

### 2.3.1 ส่วนประกอบต่างๆ ของไฟล์ฐานข้อมูลใน Access 2003

ในโปรแกรม Access จะมีไอคอนล๊อกซ์บ็อกซ์ที่สำคัญดังรูปที่ 2.7 ซึ่งไอคอนล๊อกซ์บ็อกซ์นี้จะใช้ในการสร้างฐานข้อมูล Access และไอคอนล๊อกซ์บ็อกซ์นี้มีส่วนประกอบต่างๆ ดังต่อไปนี้



รูปที่ 2.7 ส่วนประกอบต่างๆ ของโปรแกรม

ซึ่งส่วนประกอบที่สำคัญ จะประกอบไปด้วย Tables, Queries, Forms, Reports, Pages, Macros, Modules ในโครงการนี้ได้ใช้ส่วนประกอบที่สำคัญ 2 ส่วน ด้วยกันคือ

1. **Tables** ในที่นี้จะเรียกส่วนนี้ว่า ตาราง เป็นส่วนที่เก็บโครงสร้างของฐานข้อมูล และข้อมูลต่างๆ ที่เรามี เช่น ตารางเก็บข้อมูลเกี่ยวกับความเร็วรอบของมอเตอร์และวัน เวลา ที่ทำงาน ตารางนี้จะเก็บข้อมูลอยู่ในรูปแถวและคอลัมน์ โดยข้อมูลในแต่ละแถวจะเรียกว่า เรคอร์ด (Record) ซึ่งจะเป็ข้อมูลของรายละเอียดต่างๆ ของมอเตอร์ และข้อมูลในแต่ละคอลัมน์จะเรียกว่า (Field)

2. **Queries** ในที่นี้ส่วนนี้เรียกว่า คิวรี เป็นเครื่องมือที่ใช้ในการดึงข้อมูลดิบมากมายที่เก็บอยู่ในฐานข้อมูลออกมาใช้อย่างเป็นระเบียบ โดยเราสามารถเรียกดูทั้งหมด หรือเฉพาะข้อมูลที่เจาะจง และสามารถเรียงลำดับข้อมูลจากมากไปน้อย จากน้อยไปมาก รวมกลุ่มข้อมูลและสามารถหาค่ารวม (Summary) ของข้อมูลแต่ละกลุ่มได้ เราสามารถใช้คิวรีเป็นตัวดึงข้อมูลดิบมากมายออกมาเป็นข้อมูลพร้อมใช้ได้อย่างเต็มประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ที่ใช้จะเป็นไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877 มีโครงสร้าง 40 ขา มีพอร์ตใช้งานทั้งสิ้น 5 พอร์ต คือ พอร์ต A 6 บิต พอร์ต B 8 บิต พอร์ต C 8 บิต พอร์ต D 8 บิต และ พอร์ต E 3 บิต เป็นพอร์ตแบบสองทิศทางสามารถเป็นได้ทั้งอินพุตและเอาต์พุตและยังเป็นพอร์ตที่สามารถแปลงสัญญาณอนาล็อกเป็นดิจิทัลได้อีกด้วย

### คุณสมบัติของ PIC16F877 มีดังนี้

- หน่วยความจำแบบแฟลช 8 กิโลไบต์
- หน่วยความจำข้อมูล(Data Memory) 368 byte
- หน่วยความจำอีพรอม (EEPROM Data Memory) 256 byte
- ไทมเมอร์/เคาน์เตอร์ 3 ตัว
  - ไทมเมอร์ 0 ขนาด 8 บิต
  - ไทมเมอร์ 1 ขนาด 16 บิต
  - ไทมเมอร์ 2 ขนาด 8 บิต
- มีกระแสซิงค์และซอร์ส 25 mA
- วงจรแปลงสัญญาณ Analog to Digital Converter 10บิต
- มีวงจรสื่อสารแบบอนุกรมทั้ง SPI และ I2C
- มีวงจร Pulse Width Modulate (PWM) 10 บิต

หน่วยความจำแบบแฟลช สามารถเขียนและลบได้ 100,000 ครั้ง

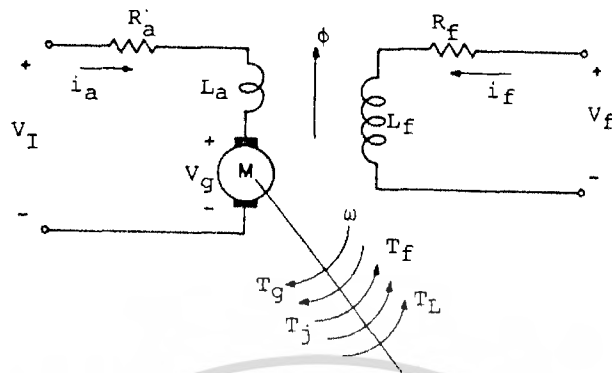
## 2.5 มอเตอร์กระแสตรง

เมื่อมีการผ่านกระแสไฟฟ้าเข้าไปยังขดลวดในสนามแม่เหล็ก จะทำให้เกิดแรงแม่เหล็ก ซึ่งมีสัดส่วนของแรงขึ้นอยู่กับการเสถียรของสนามแม่เหล็ก โดยแรงจะเกิดขึ้นเป็นมุมฉากกับกระแส และสนามแม่เหล็ก ขณะที่ทิศทางของแรงกลับมีทิศตรงกันข้ามกัน ถ้าหากกระแสของสนามแม่เหล็กไหลย้อนกลับ จึงมีผลทำให้เกิดการเปลี่ยนแปลงของกระแส และสนามแม่เหล็กเป็นผลทำให้ทิศทางของแรงเปลี่ยนไป ด้วยคุณสมบัตินี้ทำให้มอเตอร์กระแสตรงกลับทิศทางการหมุนได้

สนามแม่เหล็กของมอเตอร์ส่วนหนึ่งเกิดขึ้นจากแม่เหล็กถาวรซึ่งจะถูกยึดติดกับแผ่นเหล็ก หรือเหล็กกล้า โดยปกติส่วนนี้จะเป็นส่วนที่ยึดอยู่กับที่ และ ขดลวดเหนี่ยวนำจะพันอยู่กับส่วนที่เป็นแกนหมุนของมอเตอร์

### 2.5.1 โมเดลอิเล็กทรอนิกส์ทรแมคคานิคอล

ส่วนสำคัญของดีซีมอเตอร์แบบฟีดแบกกระตุ้นมีโมเดลดังแสดงในรูป 2.8



รูปที่ 2.8 โมเดลของดีซีมอเตอร์แบบฟีดแบกกระตุ้น

$R_a$	คือ	ความต้านทานของอาร์เมเจอร์
$L_a$	คือ	อินดักแตนซ์ของอาร์เมเจอร์
$V_g$	คือ	แรงดันกำเนิดในอาร์เมเจอร์ (แรงดันย้อนกลับ)
$R_f$	คือ	ความต้านทานของฟีดแบก
$L_f$	คือ	อินดักแตนซ์ของฟีดแบก
$\phi$	คือ	ช่องว่างอากาศของเส้นแรงสนามแม่เหล็ก
$\omega$	คือ	ความเร็วของเพลอาร์เมเจอร์
$T_g$	คือ	แรงบิดที่พัฒนาขึ้นในมอเตอร์
$T_f$	คือ	แรงบิดเสียดทานของมอเตอร์
$T_j$	คือ	แรงเฉื่อยของมอเตอร์
$T_L$	คือ	แรงบิดโหลดบนเพลของมอเตอร์

ขั้นแรกเราจะหาสมการพื้นฐาน โมเดลของดีซีมอเตอร์ได้จากกฎของอาร์เมเจอร์

$$V_i(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + V_g(t) \quad (1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $V_g(t)$  คือ แรงดันย้อนกลับของมอเตอร์ซึ่งเกิดขึ้น เมื่อเส้นลวดตัวนำของอาร์มาเจอร์หมุนตัดเส้นแรงแม่เหล็กซึ่งเกิดขึ้นในกระแสของฟลักซ์ ( $i_f$ ) ตามกฎของฟาราเดย์รูปของเส้นลวดตัวนำหมุนในฟลักซ์แม่เหล็กคงที่จะมีการเหนี่ยวนำแรงดัน ขึ้นในขดลวดนั้น

$$V_g(t) = \frac{d\lambda(t)}{dt} \quad (2)$$

เมื่อ  $\lambda(t)$  คือ เส้นแรงแม่เหล็กที่ลิงเคจ (linkages) ไปยังขดลวด และ  $t$  คือเวลาในการหมุนของคอมมิวเตเตอร์ของมอเตอร์ การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลต์เต็มขึ้นในส่วนของตัวนำนั้นตามสมการ (2) เมื่อ  $\frac{d\lambda(t)}{dt}$  จะเป็นสัดส่วนต่อเส้นแรงแม่เหล็กในช่องว่างอากาศและความเร็วเชิงมุม  $\omega(t)$  เราจะได้ว่า

$$V_g(t) = K\phi(t)\omega(t) \quad (3)$$

สมมติให้กระแสของฟลักซ์มีค่าคงที่ และไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟลักซ์เนื่องจากอาร์มาเจอร์รีแอกชัน เส้นแรงฟลักซ์ก็จะมีค่าคงที่ดังนั้นสมการ (3) ก็จะเป็น

$$V_g(t) = K_e\omega(t) \quad (4)$$

เมื่อเราสมมติให้เส้นแรงของฟลักซ์มีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้น แก่โรเตอร์ของมอเตอร์ จะเป็นสัดส่วน กับกระแสของอาร์มาเจอร์

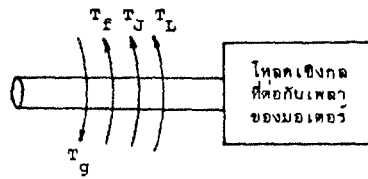
$$T_g(t) = K_t i_a(t) \quad (5)$$

เมื่อ  $K_t$  คือ ค่าคงที่ของแรงบิดของมอเตอร์

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

$$P_g(t) = T_g(t)\omega(t) \quad (6)$$

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะหายไปยังโหลดที่ต่ออยู่กับเพลาของมอเตอร์แต่กำลังงานนี้บางส่วนจะมีการสูญเสียไปในมอเตอร์ การสูญเสียจากแรงเสียดทาน หมายถึงความหน่วงเนื่องจากลมที่มีต่อโรเตอร์ แรงเสียดทานตัวรองรับโรเตอร์ กระแสที่ไหลวนในเหล็กของโรเตอร์และฮีตเทริซิส โดยแรงบิดต่างๆแสดงดังนี้



รูปที่ 2.9 แรงบิดที่เกิดขึ้นต่อ โหลดของมอเตอร์

$T_g(t)$  คือ แรงบิดของมอเตอร์

$T_r(t)$  คือ แรงบิดที่ต้องชนะการสูญเสียเนื่องจากการเสียดทาน

$T_L(t)$  คือ แรงบิดเพื่อใช้เพิ่มอัตราเร่งแก่ความเฉื่อยของโหลด

$T_L(t)$  คือ แรงบิดโหลด

ในช่วงเวลาใดๆก็ตาม แรงบิดของมอเตอร์จะต้องเท่ากับ และมีทิศทางตรงข้าม กับผลรวมของแรงบิด  $T_r(t)$ ,  $T_L(t)$  และ  $T_g(t)$  ดังนั้น

$$T_g(t) = T_r(t) + T_L(t) + J \frac{d\omega(t)}{dt} \quad (7)$$

เมื่อ  $J$  คือผลรวมของโมเมนต์แรงเฉื่อยของโรเตอร์และ โหลดที่ต่ออยู่ที่เพลาของมอเตอร์

ผลรวมของแรงบิดเสียดทานที่ประกอบกันขึ้นมาที่เพลาของมอเตอร์ ซึ่งเป็นลิเนียร์ฟังก์ชันกับความเร็วเชิงมุมของโรเตอร์เรียกว่า ส่วนประกอบของแรงเสียดทานเนื่องจากความเฉื่อย และมักจะอยู่ในเทอมที่แยกออกจากแรงเสียดทานอื่นๆ ซึ่งแสดงได้ด้วยสมการต่อไปนี้

$$T_g(t) = T_r(t) + T_L(t) + J \frac{d\omega(t)}{dt} + B\omega(t) \quad (8)$$

เมื่อ  $B$  คือสัมประสิทธิ์ของแรงเสียดทานเนื่องจากความเฉื่อยของมอเตอร์และ โหลดที่ต่ออยู่กับเพลาของมอเตอร์  $T_r(t)$  คือผลรวมของฟริกชันของโหลดและมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของเพลามอเตอร์ยกเว้นวิสคอสฟริกชัน

สมการ (1) (4) (5) และ (8) เป็นชุดสมการพื้นฐานของดีซีมอเตอร์โมเดลและสมการเหล่านี้เราสามารถจะหาทรานสเฟอร์ฟังก์ชันของดีซีมอเตอร์ได้ โดยใส่ลาปลาซทรานสฟอร์มทั้งสองข้างของชุดสมการพื้นฐานได้เป็น

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

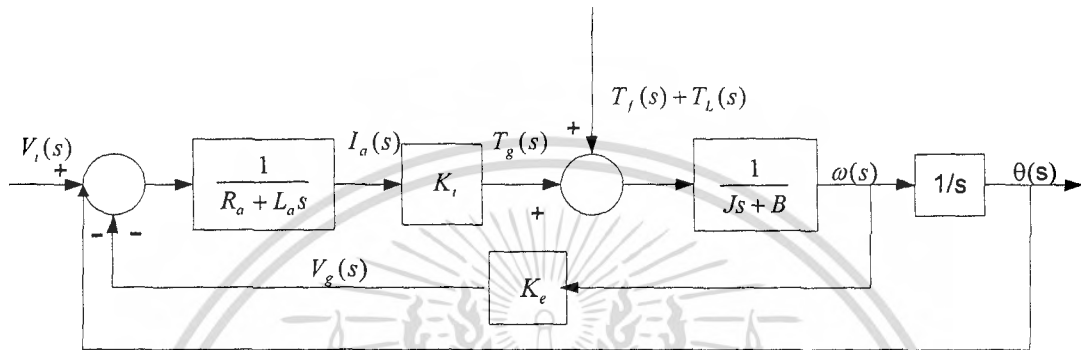
$$V_i(s) - V_g(s) = (R_a + sL_a) I_a(s) \quad (9)$$

$$V_g(s) = K_e \omega(s) \quad (10)$$

$$T_g(s) = K_t I_a(s) \quad (11)$$

$$T_g(s) - T_f(s) - T_L(s) = (B + sJ) \omega(s) \quad (12)$$

สามารถเขียนเป็นบล็อกไดอะแกรมที่แสดงสมการพื้นฐานเหล่านี้ได้ดังนี้



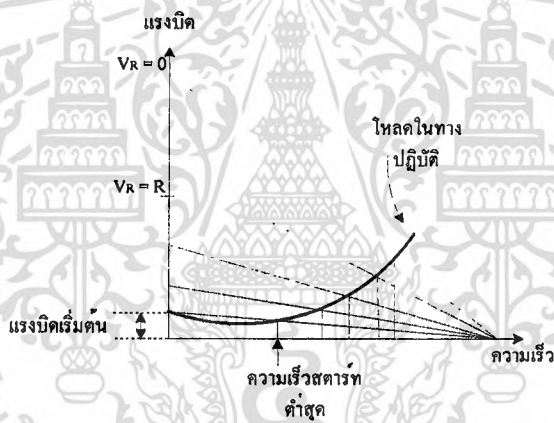
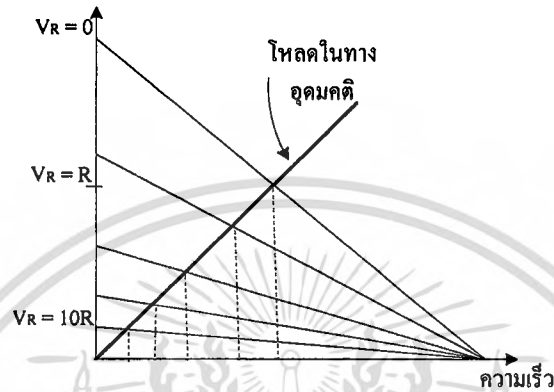
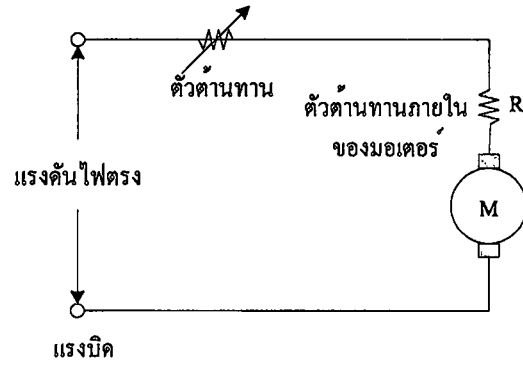
รูปที่ 2.10 บล็อกไดอะแกรมของดีซีมอเตอร์

## 2.6 การควบคุมความเร็วของมอเตอร์ขั้นพื้นฐาน

### 2.6.1 การควบคุมด้วยตัวต้านทานที่ปรับค่าได้

การควบคุมมอเตอร์รูปแบบพื้นฐานที่สุดคือ ใช้ตัวต้านทานปรับค่าได้ต่ออนุกรมกับมอเตอร์ โดยตัวต้านทานที่ปรับค่าได้จะเป็นตัวกำหนดความเร็วในการหมุนของมอเตอร์ การบังคับแบบนี้ไม่มีประสิทธิภาพ เพราะกำลังไฟสูญเสียไปในตัวความต้านทาน มักนิยมใช้กับมอเตอร์ตัวเล็กๆ การบังคับแบบนี้ให้คุณสมบัติการสตาร์ทดี (ให้แรงบิดสูงที่ความเร็วต่ำ) แต่จะให้ความเร็วสูงมากเมื่อมอเตอร์อยู่ในสถานะที่มีโหลดน้อยๆ ดังนั้นการบังคับแบบนี้มีประโยชน์เฉพาะสถานะที่แรงต้านคงที่ เช่น การบังคับความเร็วของเครื่องจักรเย็บผ้า เป็นต้น

72691

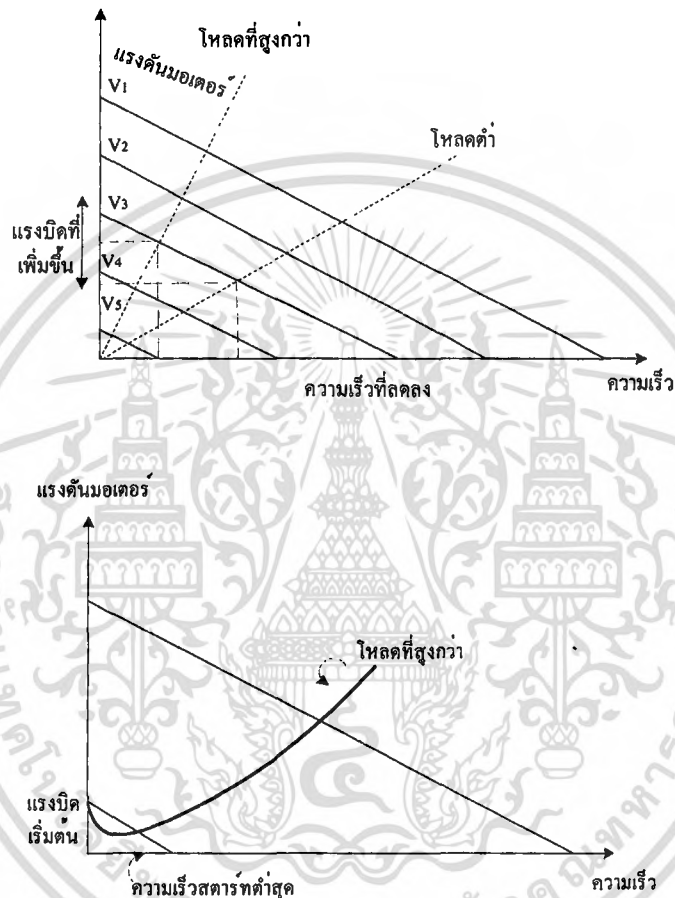


รูปที่ 2.11 วงจรควบคุมความเร็วของมอเตอร์กระแสตรงแบบใช้ตัวต้านทานอนุกรม และกราฟแสดงคุณสมบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.2 การควบคุมด้วยวิธีเปลี่ยนค่าแรงดัน

วิธีการนี้ดีกว่าวิธีการแรก แต่จะซับซ้อนกว่าต้องใช้อุปกรณ์อิเล็กทรอนิกส์ ที่ที่อัตราขยายกำลังสูง และ มอเตอร์จะถูกป้อนด้วยแรงดันที่เปลี่ยนแปลงค่าได้ จากแหล่งจ่ายที่มีอิมพีแดนซ์ต่ำ ข้อดีของการควบคุมวิธีนี้คือ ถ้าความเร็วลดลงจากผลของแรงบิด แรงดันที่ป้อนให้กับมอเตอร์จะเพิ่มขึ้นเพื่อรักษาระดับความเร็ว ส่วนข้อเสียจากการควบคุมวิธีนี้คือ เมื่อมอเตอร์มีความเร็วต่ำแรงดันที่ป้อนให้กับมอเตอร์จะมีค่าต่ำเช่นกัน



รูปที่ 2.12 การควบคุมความเร็ว โดยเปลี่ยนค่าแรงดัน

### 2.6.3 การควบคุมแบบ PWM (Pulse Width Modulation)

Pulse width modulation (PWM) คือ เทคนิคสำหรับควบคุมวงจรทางด้านฮาร์ดแวร์โดยใช้สัญญาณเอาต์พุตแบบดิจิทัลของไมโครโปรเซสเซอร์ควบคุม

#### การทำงานของสัญญาณ PWM

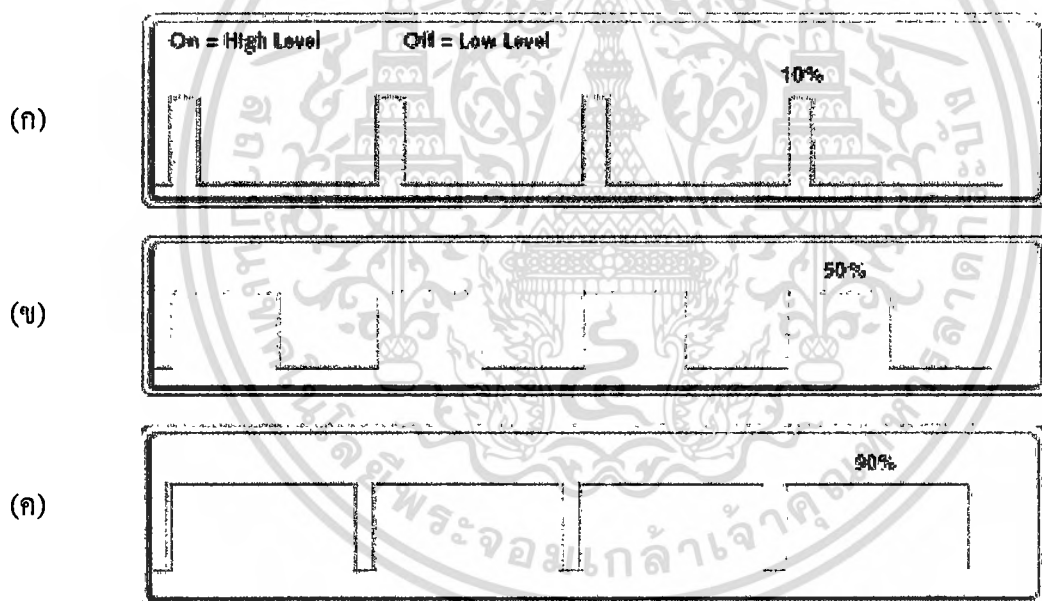
รูปที่ 2.31 แสดงสัญญาณ PWM ที่แตกต่างกัน 3 สัญญาณ โดยที่

รูปที่ 2.13 (ก) แสดงสัญญาณ PWM ที่ 10% duty cycle คือ สัญญาณในการอนจะเป็น 10% ของคาบสัญญาณ และ จะออฟเป็น 90% ของคาบสัญญาณ

รูปที่ 2.13 (ข) แสดงสัญญาณ PWM ที่ 50% duty cycle คือ สัญญาณในการอนจะเป็น 50% ของคาบสัญญาณ และ จะออฟเป็น 50% ของคาบสัญญาณ

รูปที่ 2.13 (ค) แสดงสัญญาณ PWM ที่ 90% duty cycle คือ สัญญาณในการอนจะเป็น 90% ของคาบสัญญาณ และ จะออฟเป็น 10% ของคาบสัญญาณ

เช่น ถ้า Power Supply มี 9V และ duty cycle เป็น 10% จะได้อาต์พุต 0.9V



รูปที่ 2.13 สัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆ กัน

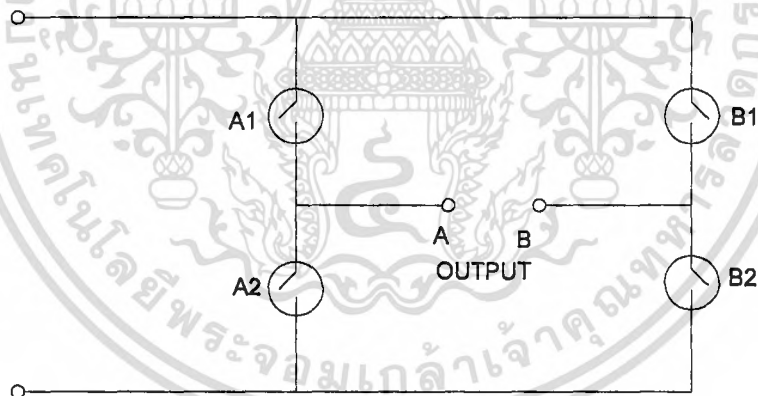
## ทำไมถึงใช้ PWM ในการควบคุมความเร็วมอเตอร์

มีหลายเหตุผลว่าทำไม PWM ถึงถูกเลือกใช้ในการควบคุมความเร็วของมอเตอร์ เช่น:

- PWM ง่ายในการอินเทอร์เฟสกับไมโครคอนโทรลเลอร์ และ ใช้เพียงแค่เอาท์พุตสัญญาณเดียวในการควบคุมความเร็ว
- PWM มีประสิทธิภาพ คือ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF(FULL ON and FULL OFF)
- PWM ทำให้ได้ค่า ทอร์ค และ ความเร็วสูงสุดของมอเตอร์ เป็นเพราะ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF (FULL ON and FULL OFF)

ซึ่งในโครงการนี้ที่ใช้ทรานซิสเตอร์เป็นตัวตัดต่อวงจร เราสามารถจะควบคุมจังหวะในการจ่ายกระแสได้ การ on และ off ในสัดส่วนต่างๆกัน ด้วยความถี่ที่เหมาะสมก็จะทำให้มอเตอร์ หมุนที่ความเร็วต่างๆ กันตามความต้องการได้ ถ้าความถี่ต่ำไปมอเตอร์ก็จะหมุนแบบกระตุกๆ ไม่เรียบ และอาจจะได้ยินเสียง จากการสั่นของ ขดลวดทองแดง ถ้าความถี่สูงกว่า 20 kHz เราก็จะไม่ได้ยินเสียง ขดลวดสั่นอีกต่อไป และ ที่ความถี่สูงขึ้นไป มากๆ ก็จะเกิดการสูญเสีย พลังงาน ในวงจรมากเกินความจำเป็น

## 2.7 วงจรขับมอเตอร์



รูปที่ 2.14 โครงสร้าง H bridge

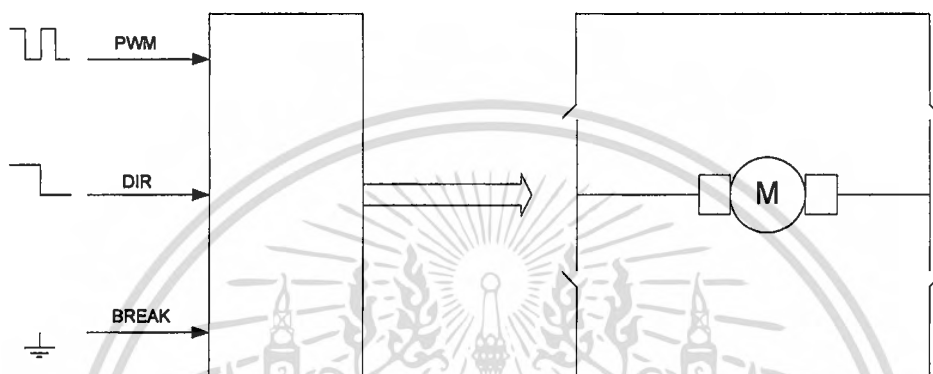
วงจรควบคุมทิศทางของมอเตอร์โดยการเปิด-ปิด สวิตช์ A1,B2 หรือ B1,A2 เช่นถ้าเปิด A1,B2 มอเตอร์จะหมุนไปทางหนึ่งและจะหมุนอีกทางหนึ่งเมื่อปิดสวิตช์ B1,A2 นอกจากนี้เราต้องการควบคุมทิศทางแล้วเรายังต้องควบคุมความเร็วของมอเตอร์อีกด้วยโดยส่วนมากจะใช้วิธีการ PWM (Pulse

Width Modulate) ซึ่งสามารถสร้างแรงดันเฉลี่ย (average Voltage) ที่มากกว่า 0 ได้ซึ่งก็หมายความว่าเราสามารถทำให้มอเตอร์เร็วขึ้นได้นั่นเอง

### Sign/Magnitude PWM

สัญญาณที่ใช้ควบคุมทิศทางและกำลังนั้นแยกจากกัน โดยกระแสที่ไหลลง ได้รับจะเป็นสัดส่วนโดยตรง กับความกว้างของ PWM ที่ส่งเข้ามา

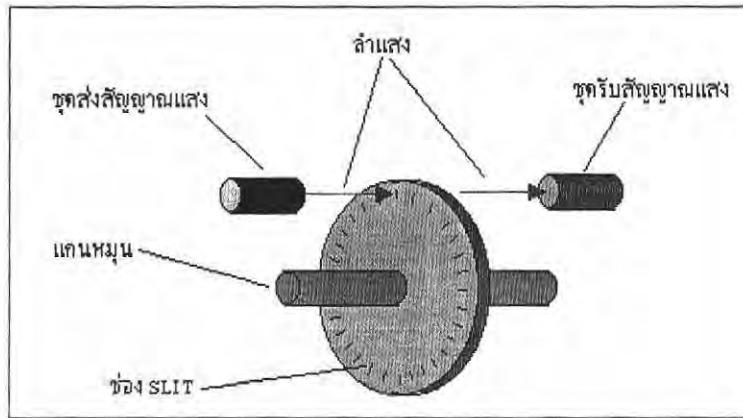
ถ้าสัญญาณ DIR ที่ Pin 5 เป็น low มอเตอร์จะหมุนอีกทางหนึ่งและเช่นเดียวกันกำลังจะขึ้นกับการปรับ Duty cycle ถ้ามากมอเตอร์จะหมุนเร็ว



รูปที่ 2.15 การควบคุมแบบ Sign/Magnitude

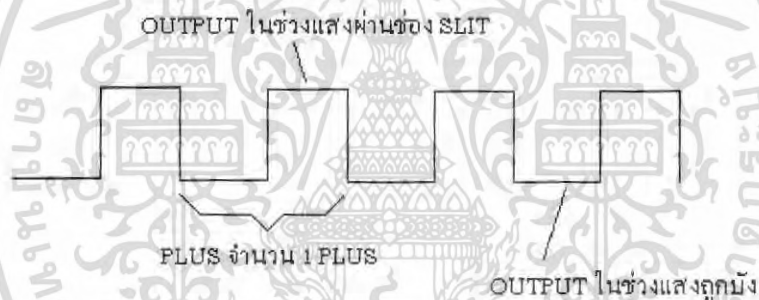
## 2.8 เอ็นโค้ดเดอร์

Rotary Encoder มีลักษณะเป็นแผ่นกลมมีแกนอยู่ตรงกลาง และ ที่แผ่นกลม จะมีช่องเล็ก ที่แสงสามารถ ส่องผ่านได้ เป็นจำนวนมากเราเรียกช่องนี้ว่า ช่อง slit ซึ่งที่ด้านหนึ่งของ แผ่นกลม นี้ จะมีตัวส่งแสงอินฟราเรด ไปยังตัวรับสัญญาณแสงอินฟราเรด ซึ่งจะอยู่ในด้านตรงกันข้าม



รูปที่ 2.16 Rotary Encoder

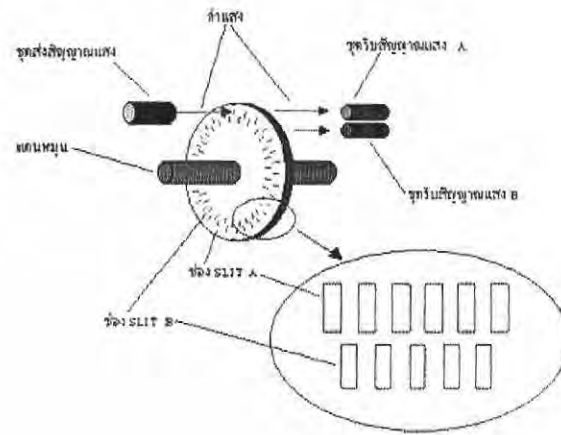
เมื่อหมุนแกนหมุนทำให้แผ่นกลมหมุนไปตัดลำแสงอินฟราเรด ดังนั้นชุดรับแสงอินฟราเรด จึงมีแสงมากระทบเป็นช่วงๆ ตามจังหวะที่แสงส่องผ่านช่อง slit จึงทำให้สัญญาณเอาต์พุตของชุดรับแสงอินฟราเรด มีลักษณะเป็นพัลส์



รูปที่ 2.17 การสร้างพัลส์ของ Rotary Encoder

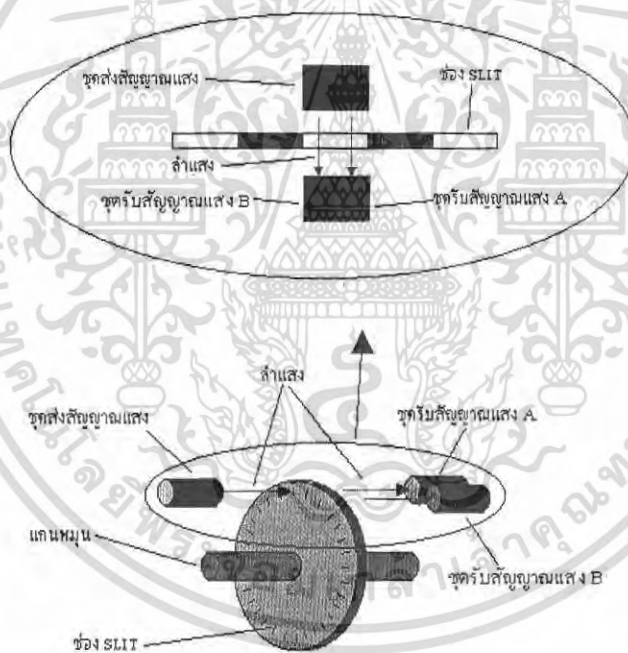
จำนวนพัลส์ที่ได้ออกมาจะเป็นตัวที่ชี้บ่งว่ามอเตอร์หมุนไปกี่องศา หรือกี่รอบซึ่งเราสามารถคำนวณได้จาก สูตร จำนวนรอบที่ มอเตอร์หมุนไป = จำนวน plus ค่าความละเอียดของ rotary encoder โดยที่ ค่าความละเอียดของ rotary encoder นั้นนิยมใช้หน่วยเป็น plus/ round หรือ ppr เช่น 1000 ppr ก็หมายถึงว่าเมื่อ มอเตอร์หมุนไป 1 รอบจะมีพัลส์ออกมา 1000 plus เป็นต้น ส่วนในเรื่องที่ว่าเราจะทราบได้อย่างไรว่า มอเตอร์หมุนไปทิศทางใดนั้น ส่วนใหญ่แล้วจะใช้วิธีสร้างช่องเป็น 2 ชุดเหลื่อมกัน 90 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 พัลส์ของ Rotary แบบสองเฟส

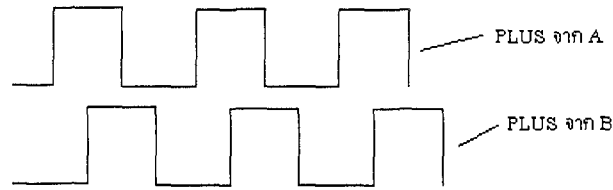
หรืออาจจะใช้ช่องเพียง 1 ชุดแต่มีการจัดวางชุดรับสัญญาณแสงดังรูปข้างล่างแต่ข้อสำคัญคือจะต้องมีเฟสต่างกัน 90 องศา



รูปที่ 2.19 การตรวจเช็คสัญญาณพัลส์ของ Rotary Encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

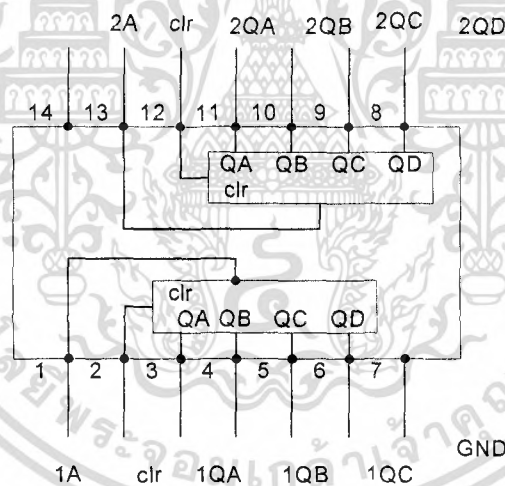
ดังนั้นสัญญาณเอาต์พุต จากมอเตอร์ จึงมี 2 ชุด คือ A และ B โดยที่สัญญาณพัลส์จาก A และ B ก็จะเหลื่อมกัน 90 องศาด้วย อาจกล่าว ได้ว่าสัญญาณเอาต์พุต จากมอเตอร์ มีค่าเท่ากับ 2 บิต คือ หนึ่งบิต มาจาก A และอีกหนึ่งบิต มาจาก B



รูปที่ 2.20 ลักษณะพัลส์ของ Rotary Encoder ทั้ง สองเฟส

## 2.9 วงจรนับ 74HC393

คือไอซีที่ใช้สำหรับนับจำนวนพัลส์ของเอ็นโค้ดเดอร์ ซึ่งมีการจัดวางขา และตารางการ เป็นดังนี้



รูปที่ 2.21 การจัดวางขาของไอซีวงจรรนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางการทำงานของไอซีเบอร์ 74HC393

COUNT	QA	QB	QC	QD
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.10 การอเนว้ตตัวควบคุม พีไอดี ด้วยไมโครคอนโทรลเลอร์

ตัวควบคุมพีไอดี มีแบบจำลองทางคณิตศาสตร์ ในรูปของทรานสเฟอร์ฟังก์ชันก้เป็นดังสมการที่ (13)

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (13)$$

$$Gc(s) = K_p + \frac{K_i}{s} + K_D \quad (14)$$

$K_p$  คือ proportion gain

$K_i$  คือ integral gain

$K_d$  คือ derivative gain

ซึ่งค่า  $K_p$ ,  $K_i$  และ  $K_d$  คือ ค่าพารามิเตอร์ของตัวควบคุม PID นั้นเอง

### 2.10.1 องค์ประกอบ P

เมื่อพิจารณาสมการที่ (13) องค์ประกอบ P สามารถแสดงได้ในสมการที่ (15) เมื่อทำการแปลงลาปลาซ (Laplace Transform) สมการที่ (15) จะได้เป็นสมการที่ (16) จากนั้นแปลง Z (Z-Transform) สมการที่ (16) และแปลงเป็นสมการผลต่างจะได้เป็นสมการที่(17) และ (18) ตามลำดับ

$$u_p(t) = K_p e(t) \quad (15)$$

$$U_p(s) = K_p E(s) \quad (16)$$

$$U_p(z) = K_p E(z) \quad (17)$$

$$u_p(i) = K_p e(i) \quad (18)$$

### 2.10.2 องค์ประกอบ I

เมื่อพิจารณาสมการที่ (13) องค์ประกอบ I สามารถแสดงได้ในสมการที่ (19) เมื่อทำการแปลงลาปลาซ (Laplace Transform) สมการที่ (19) จะได้เป็นสมการที่ (20) สมการที่ (21) แสดงการอินทิเกรตสี่เหลี่ยมคางหมู

$$u_i(t) = K_i \int e(t) dt \quad (19)$$

$$U_I(s) = \frac{K_I}{s} E(s) \quad (20)$$

$$s = \frac{T(1+Z^{-1})}{2(1-Z^{-1})} \quad (21)$$

การประมาณค่าของทუსติน (Tustin's approximation) ได้เป็นสมการที่ (22)

$$s = \frac{2(1-Z^{-1})}{T(1+Z^{-1})} \quad (22)$$

แทนค่า  $s$  จากสมการที่ (22) ลงในสมการที่ (20) เพื่อทำการแปลง  $Z$  หลังจากนั้นแปลงให้อยู่ในรูปสมการผลต่างได้ดังสมการที่ (23), (24) และ (25) ตามลำดับ

$$(1-Z^{-1})U_I(z) = \frac{K_I T}{2} (1+Z^{-1})E(z) \quad (23)$$

$$U_I(z) = U_I(z)U_I(Z^{-1}) - \frac{K_I T}{2} (E(z) + E(z)z^{-1}) \quad (24)$$

$$u_I(i) = u_I(i-1) + \frac{K_I T}{2} (e(i) + e(i-1)) \quad (25)$$

### 2.10.3 องค์ประกอบ D

เมื่อพิจารณาสมการที่ (1) องค์ประกอบ D สามารถแสดงได้ในสมการที่ (26) เมื่อทำการแปลงลาปลาซ (Laplace Transform) สมการที่ (26) จะได้เป็นสมการที่ (27) สมการที่ (28) แสดงการอนุพันธ์ทำการแปลง Z-Transform และแปลงเป็นสมการผลต่างได้เป็นสมการที่ (29) และ (30) ตามลำดับ

$$u_D(t) = K_D \frac{de(t)}{dt} \quad (26)$$

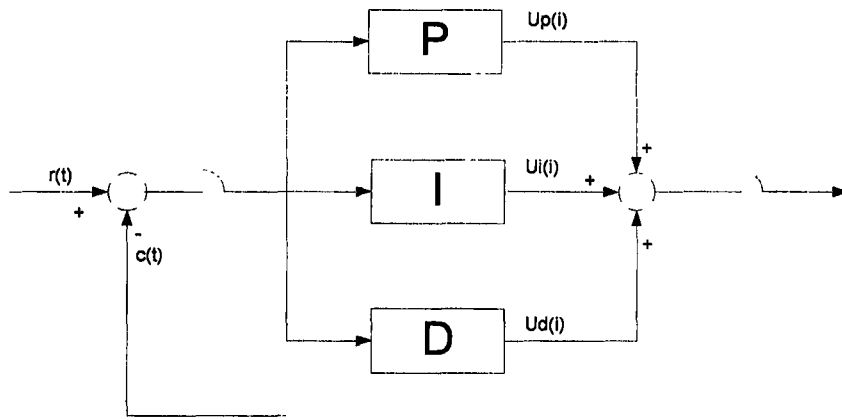
$$U_D(s) = K_D s E(s) \quad (27)$$

$$s = \frac{(1-Z^{-1})}{T} \quad (28)$$

$$U_D(z) = \frac{K_D}{T} (1-Z^{-1})E(z) \quad (29)$$

$$u_D(i) = \frac{K_D}{T} (e(i) - e(i-1)) \quad (30)$$

ดังนั้นจะได้ตัวควบคุมพีไอดี ในรูปของตัวควบคุมพีไอดี แบบดิจิทัล ซึ่งจะได้ออกมาสมการข้างต้นเป็นดังบล็อกไดอะแกรมในรูปที่ 2.22



รูปที่ 2.22 บล็อกไดอะแกรมของตัวควบคุม PID

## 2.11 การรับส่งข้อมูลแบบไร้สาย

การรับส่งข้อมูลโดยใช้อากาศเป็นตัวกลางนั้นการสื่อสารในแบบนี้มีทั้งข้อดี แต่ก็มีข้อจำกัดอยู่บ้างบางประการเหมือนกัน โดยเฉพาะอย่างยิ่งเรื่องของความน่าเชื่อถือของข้อมูลที่รับและส่งกันซึ่งมีโอกาสผิดพลาดหรือสูญหายได้เหมือนกัน เนื่องจากการลำเลียงข้อมูลนั้นไม่ได้ใช้สายสัญญาณเป็นตัวกลาง ในการรับส่งข้อมูล แต่ใช้อากาศเป็นตัวกลางในการรับส่งข้อมูลแทน ซึ่งมีโอกาสที่ข้อมูลจะเกิดการรบกวนจากสัญญาณอื่นๆ ที่มีย่านความถี่ใกล้เคียงกัน แล้วทำให้ข้อมูลผิดเพี้ยนไปได้บ้างเหมือนกัน แต่ก็ได้มีการแก้ไขปัญหานี้โดยมีระบบการป้องกัน โดยการเข้ารหัส และถอดรหัสของข้อมูลให้มีความน่าเชื่อถือ โดยข้อมูลแต่ละไบต์ ที่มีการรับส่งกันนั้น จะมีการตรวจสอบความถูกต้องของข้อมูลให้ด้วยแล้ว โดยข้อมูลที่รับได้จากด้าน RF นั้นรับประกันได้ว่าเป็นข้อมูลที่มีความถูกต้องแน่นอน แต่อย่างไรก็ตามการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกัน จึงมีการตรวจสอบข้อมูลทุกไบต์ ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีความผิดพลาดเกิดขึ้นจะทิ้งข้อมูลไบต์ นั้นไป ซึ่งผู้ใช้ควรมีกลไกในการตรวจสอบข้อมูลที่รับส่งกันว่าครบถ้วนหรือไม่ด้วย ซึ่งหากพบว่ามีการสูญหายของข้อมูลเกิดขึ้นก็ให้ร้องขอให้มีการส่งข้อมูลนั้นซ้ำนั้นๆ ใหม่อีกครั้งหนึ่ง ก็จะสามารแก้ไขปัญหาดังกล่าวได้

### 2.11.1 TRW 2.4G

ในโครงการนี้ได้เลือกใช้ตัวรับ ส่งสัญญาณแบบไร้สาย คือ TRW 2.4G ซึ่งมีขนาดเล็ก ราคาไม่แพง และในตัว TRW 2.4G เพียงตัวนั้นยังสามารถที่เป็นได้ทั้ง ตัวส่งสัญญาณ (Transmit) และตัวรับสัญญาณ (Receive) ในตัวเดียวได้อีกด้วย ซึ่งจะสะดวกและประหยัด แต่ถ้าให้ตัว TRW 2.4G เป็นทั้ง

ตัวรับและตัวส่งสัญญาณในตัวเดียวกัน การรับและการส่งข้อมูลนั้นจะเป็นการทำงานแบบ Half Duplex แต่เราก็สามารถใช้งานแบบ Full Duplex ได้เช่นกัน เพียงแค่ใช้ตัว TRW 24G เพิ่มขึ้นเท่านั้น คุณสมบัติพิเศษของตัวรับส่งสัญญาณมีดังนี้

- ความถี่ที่ใช้งาน: 2.4~2.524 GHz
- โหมดที่ใช้ในการมอดคูเลต: GFSK
- แรงดันไฟฟ้าที่ใช้งาน: 3V
- Output Power: +4dBm
- อัตราความเร็วในการส่งผ่านข้อมูล: 1Mbps; 250Kbps
- ขนาด: 20.0\*36.7\*2.4mm
- ระยะทางที่ใช้งานได้ไกลสุด: 280m (250Kbps); 150m (1Mbps)
- เสออากาศในตัว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบและการทำงาน

#### 3.1 ภาพรวมของโครงการ

ในโครงการนี้เป็นการรวมการ สั่งงาน และการแสดงผลค่าต่างๆ ของการควบคุมความเร็วของมอเตอร์ มารวมกันบนหน้าจอกอมพิวเตอร์ เพื่อความสะดวก และรวดเร็วในการดูผลการทำงาน การสั่งงาน และการเก็บบันทึกค่าสถานะการทำงาน ในการออกแบบนั้นจึงได้แยกออกเป็น 2 ส่วนด้วยกัน คือ ส่วนควบคุมและสั่งงาน ส่วนปฏิบัติการ

##### 3.1.1 ส่วนควบคุมและสั่งงาน

ในส่วนนี้จะอยู่บนหน้าจอกอมพิวเตอร์ เพื่อความสะดวกในการสั่งงาน และการดูผลการทำงาน จึงเลือกใช้โปรแกรม Visual Basic ในการออกแบบหน้าต่างควบคุมการทำงาน ซึ่งจะมีหน้าต่างๆ หลายๆ หน้าต่างเพื่อควบคุมการทำงานที่แตกต่างกันออกไป ดังนั้นจึงต้องมีการเขียนโค้ดโปรแกรม เพื่อที่จะเชื่อมโยงส่วนต่างๆ เข้าด้วยกัน

#### 3.2 การสร้างและออกแบบจอภาพ

Visual Basic เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งที่มี Tool สำหรับสร้างจอภาพด้วยเทคนิคแบบ Visual Basic กล่าวคือ ในการสร้างจอภาพด้วย Visual Basic จะเป็นเพียงการนำ Control ต่างๆที่ Visual Basicเตรียมไว้ให้ ไปวางไว้บน Form ตามความต้องการซึ่งด้วยวิธีนี้การสร้างจอภาพจึงเป็นเรื่องที่ง่าย และใช้เวลาน้อยซึ่งต่างจากโปรแกรมในลักษณะเดิม ที่ต้องใช้เวลาค่อนข้างมาก ในการสร้างแต่ละจอภาพขึ้นใช้งาน Control ต่างๆใน Toolbox ซึ่งใช้ในการสร้างจอภาพ จะมีชื่อเรียกและการทำงานแตกต่างกันไป การจะเลือก Control ใดมาใช้นั้นขึ้นอยู่กับจอภาพออกแบบจอภาพของนักพัฒนาโปรแกรม และความเหมาะสมในการใช้งาน หลังจากที่วาง Control ลงบน Form แล้วขั้นตอนต่อไปในการพัฒนาโปรแกรมด้วย Visual Basic ได้แก่ การกำหนดคุณสมบัติ (ต่อไปนี้จะเรียกว่า “Property”) ให้กับแต่ละ Object ที่ปรากฏอยู่บน Formซึ่งแต่ละ Object จะมี Propertyทั้งที่เหมือนกับ Object อื่นๆ และ Propertyประจำตัว การกำหนด Propertyให้กับ Objectจะกระทำตั้งแต่เริ่มวาง Objectลงบน Form หรืออาจเปลี่ยนแปลงโดยการเขียนโปรแกรมก็ได้

### การ Run และเลิกงาน Project

ในการ Run Project ที่พัฒนาขึ้นด้วย Visual Basic สามารถ Run ทั้งโดยใช้ Interpreter และการใช้ Compiler กล่าวคือ เราสามารถทดลอง Run สิ่งต่างๆที่เราทำขึ้นพร้อมๆกับการแก้ไขโปรแกรม จนกระทั่งเสร็จสมบูรณ์ แล้วจึง Compile โปรแกรมให้อยู่ในรูปของ Executed Program เพื่อนำไปใช้งาน

ในการ Run Project ทำได้ 3 วิธี คือ

วิธีที่ 1 โดยการกด F5

วิธีที่ 2 คลิกที่ Icon “Run” ใน Toolbar

วิธีที่ 3 เลือกจากเมนู Run และ Start ตามลำดับ

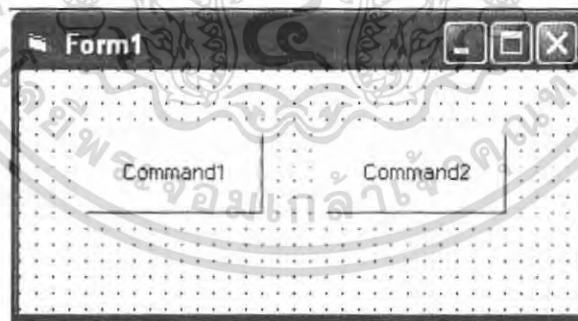
วิธีการเลิกงาน Project ทำได้ 2 วิธี คือ

วิธีที่ 1 คลิกที่ Icon “End” ใน Toolbar

วิธีที่ 2 เลือกจากเมนู Run และ End ตามลำดับ

### Procedure และ Function ใน Visual Basic

Visual Basic จะแบ่งโปรแกรมออกเป็นส่วนๆที่เรียกว่า “โมดูล” โดยจะแบ่งโปรแกรมออกเป็น “Procedure” และ “Function” แต่เนื่องจากการกำหนดการทำงานของโปรแกรมจะกำหนดตาม Event ที่เกิดขึ้นกับ Object ดังนั้นการแบ่ง Procedure ใน Visual Basic จึงแบ่งตามชื่อ Object และ Event เช่น การเขียนโปรแกรมให้กับ Object สมมุติชื่อ “Command1” ซึ่งเป็นปุ่มบนจอภาพภายใต้ Event กดปุ่ม ก็จะเขียนเป็น Procedure หนึ่งกับอีก Object หนึ่งเป็น Object ประเภทเดียวกันแต่คนละชื่อ สมมุติชื่อ “Command2” ภายใต้ Event กดปุ่มเหมือนกันก็ต้องเขียนเป็นอีก Procedure หนึ่งถึงแม้จะเป็น Event ที่เกิดกับ Object คนละตัว



รูปที่ 3.1 หน้าต่างการทำงานเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sub Command1\_Click()

Action1

End Sub

Sub Command2\_Click()

Action2

End Sub

ชื่อของ Procedure ใน Visual Basic จะอยู่ในรูปแบบดังนี้

[Public | Private] Sub name [(arglist)]

...

End Sub

โดยที่ Private Sub จะเป็นคำเฉพาะที่ใช้บอกว่าเป็น Procedure

name คือ ชื่อของ Procedure

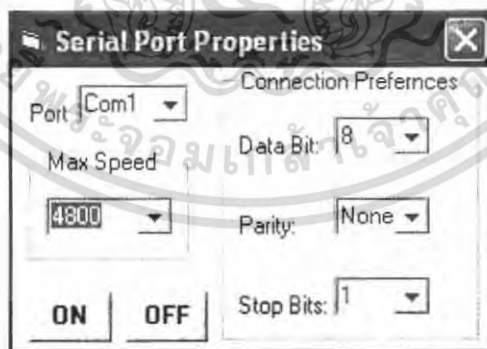
arglist คือ รายชื่อ Argument ที่ใช้ภายใน Procedure นั้น

End Sub คือ เป็นคำเฉพาะที่ใช้สำหรับจบการทำงานของแต่ละ Procedure

คำสั่ง Public ใช้ในกรณีที่ต้องการให้ Procedure นั้นสามารถถูกเรียกใช้โดย Procedure อื่นที่สร้างขึ้นใน Form อื่น ส่วนคำสั่ง Private จะใช้ในกรณีที่ต้องการให้ Procedure นั้นสามารถถูกเรียกใช้ได้ เฉพาะ Procedure ที่สร้างขึ้นใน Form เดียวกับ Procedure นั้น

### 3.3 การออกแบบหน้าต่างในการใช้งาน

ในการติดต่อกับอุปกรณ์ภายนอกของโปรแกรม Visual Basic นั้นจะใช้การติดต่อแบบพอร์ตอนุกรม RS 232 ซึ่งจะต้องกำหนด พอร์ต อัตราความเร็วของข้อมูล จำนวนบิต สามารถกำหนดได้จากหน้าต่างควบคุมได้ และเปลี่ยนแปลงได้สะดวก



รูปที่ 3.2 การกำหนดพอร์ตอนุกรม RS 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสั่งงานเพื่อที่จะนำไปควบคุมมอเตอร์นั้น ได้มีการแบ่งออกเป็น 4 โหมดการทำงาน คือ

- 1) ไม่ตั้งเวลาในการทำงาน ไม่บันทึกข้อมูล
- 2) ไม่ตั้งเวลาในการทำงาน บันทึกข้อมูล
- 3) ตั้งเวลาในการทำงาน ไม่บันทึกข้อมูล
- 4) ตั้งเวลาในการทำงาน บันทึกข้อมูล

เพื่อความสะดวกในการใช้งานแต่ละแบบที่แตกต่างกันออกไป ซึ่งในแต่ละโหมดนั้นสามารถเลือกได้จากหน้าต่างควบคุมหลัก โดยการคลิกเลือกในช่องที่เราต้องการ และทำเลือกเวลาที่ต้องการได้จากช่องที่กำหนด



Setting Motor

Set Time

Start Time: 10:06:31

Stop Time: 10:20:00

Run Now

Run Time

Time Interval:

- 5 นาที
- 10 นาที
- 30 นาที
- 1 นาที
- 2 นาที
- 5 นาที
- 10 นาที
- 30 นาที
- 1 ชั่วโมง
- ไม่บันทึก

รูปที่ 3.3 การเลือกโหมดการทำงาน

หน้าต่างที่ออกแบบให้เป็นตัวที่สั่งการทำงาน ทำหน้าที่ควบคุมการทำงาน แสดงผลการทำงาน บันทึกค่าต่างๆบนคอมพิวเตอร์ ซึ่งสามารถที่จะสั่งงาน หยุดการทำงาน ควบคุมความเร็ว ได้ในส่วนของโปรแกรมนี้ รูปร่างลักษณะของโปรแกรมควบคุม ดังรูปที่ 3.4

**Wireless Motor Control** 10:00:35

Setting Comport: SET COM | Setting Motor: Motor

Motor 1 สถานะการทำงาน ของมอเตอร์

เวลาเริ่มทำงาน : 10:01:08 ความเร็วที่ตั้งไว้ : 980 RPM บันทึกค่าทุกๆ 5 วินาที

เวลาหยุดทำงาน : 10:03:00 Load1 Off Load2 Off Load3 Off Load4 Off

SPEED 0000 RPM

LOAD1 LOAD2 LOAD3 LOAD4

Send Stop

Motor Speed 0000 RPM 1200 Max

ตารางบันทึกค่า

Time Start	Time Stop	Time Total	Time	Set Speed	Speed	Date	Load
10:03:00			19:53:02	769	980	12/11/2549	2
10:16:00	10:17:00	0:01:00	19:53:02	769	980	12/11/2549	2
13:02:11	13:03:00	0:01:11	19:53:02	769	980	12/11/2549	2
19:51:31	19:53:00	0:01:29	19:53:02	769	980	12/11/2549	2

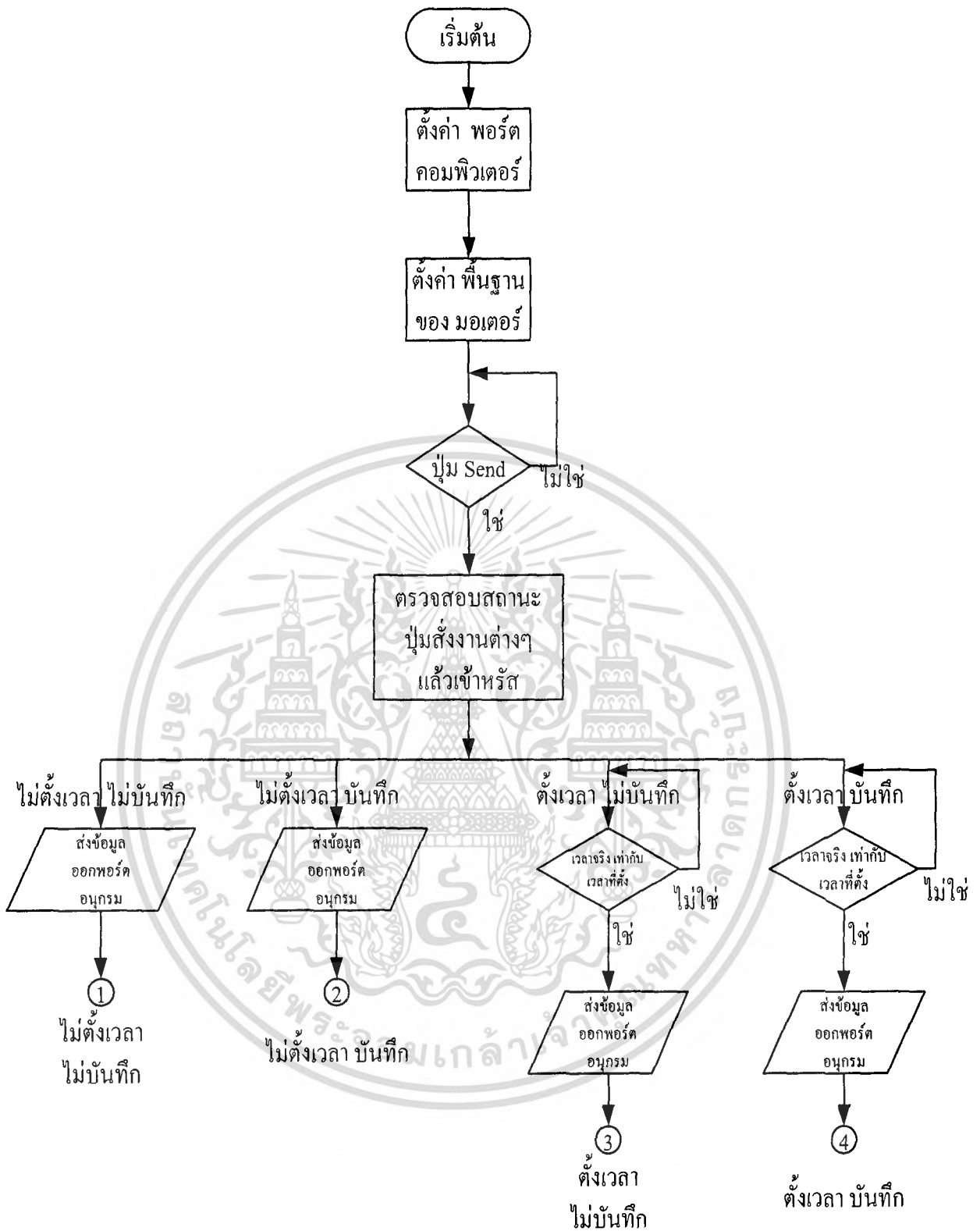
Report Graph

Exit

รูปที่ 3.4 หน้าต่าง โปรแกรมหลักในการควบคุม

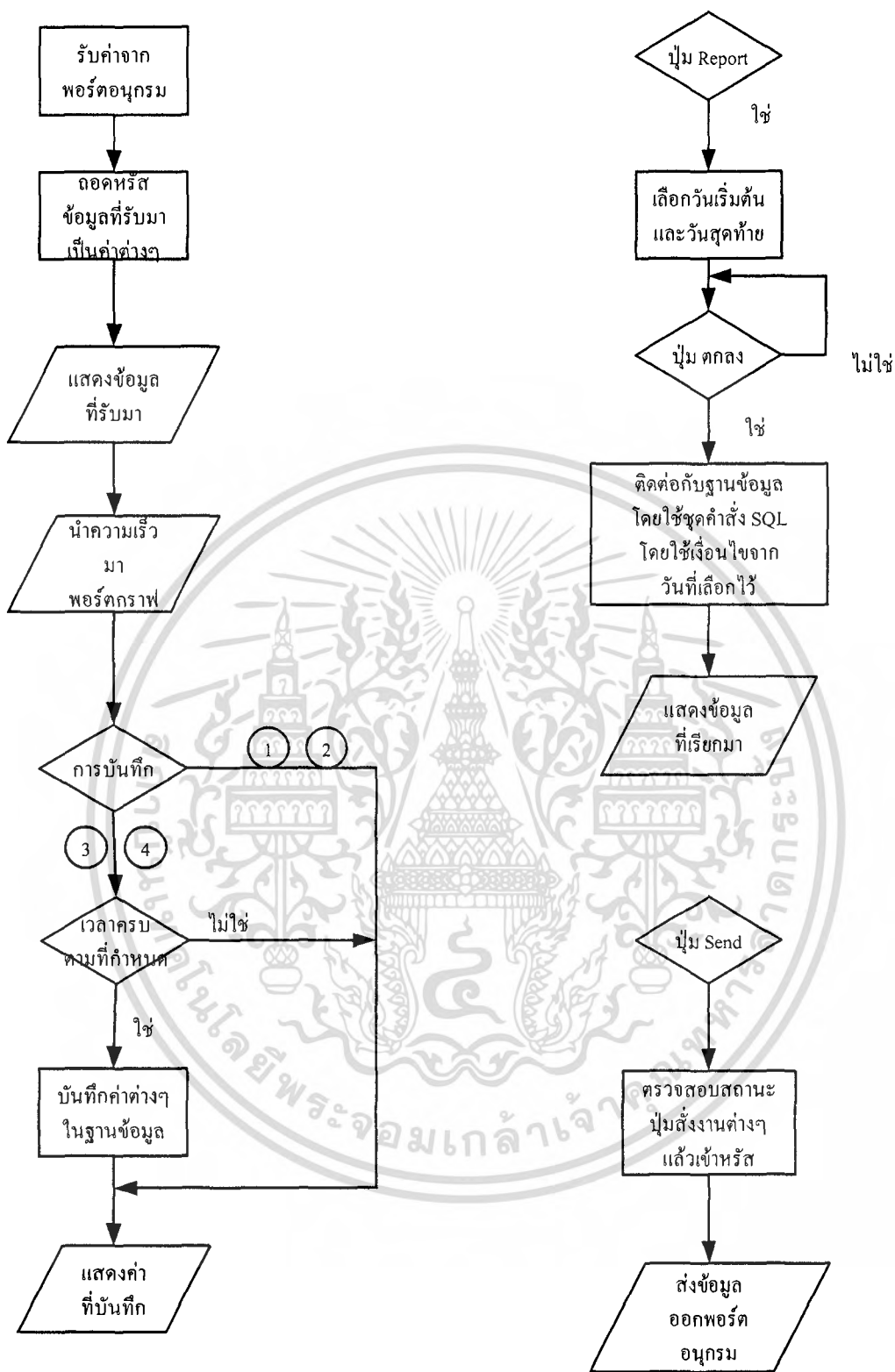
เมื่อมีการสั่งงานจากส่วนควบคุมออกไปนั้น ส่วนของตัวควบคุมนั้นก็ทำการแสดงข้อมูลที่ส่งกลับมาให้ นั่นก็คือค่าที่เราต้องการที่จะควบคุมนั่นเอง และจะนำค่านั้นมาทำการพรีออร์ทเป็นกราฟ เพื่อที่จะแสดงการเปรียบเทียบ ระหว่างค่าที่เราตั้งไว้ กับค่าที่ทำงาน ได้จริง ให้ดูเปรียบเทียบ ได้ทันที และทำการบันทึกค่าต่างๆ ในช่วงเวลาต่างๆ เพื่อที่จะเก็บเป็นข้อมูลการทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 โพลีชาร์ตการทำงานการตั้งค่าของหน้าต่างควบคุมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

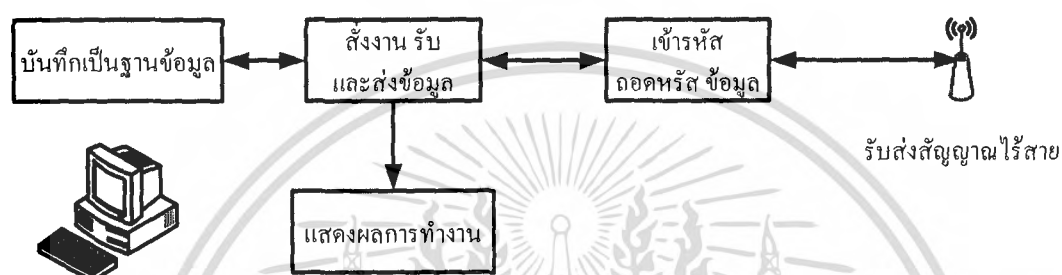


รูปที่ 3.6 โพลีชาร์ตการรับค่าการทำงานเพื่อแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การส่งผ่านข้อมูล

เมื่อส่งข้อมูลจากการสั่งงานของผู้ใช้แล้ว ตัวข้อมูลนั้นก็จะถูกจัดส่ง ไปยังตัวคอนโทรลเลอร์ เพื่อที่จะทำการควบคุมความเร็วของมอเตอร์ให้เป็นไปตามต้องการ แล้วจากนั้นก็ทำการวัดค่าต่างๆ ของมอเตอร์เพื่อที่จะนำกลับมาแสดงที่หน้าจอคอมพิวเตอร์ ซึ่งกระบวนการทั้งหมดนั้นจะทำการส่งข้อมูลทั้งหมด ผ่านตัวกลางเป็นอากาศ คือใช้ตัวรับ ส่งสัญญาณ TRW 2.4G สามารถเป็นทั้งตัวรับ ตัวส่ง หรือว่าเป็นทั้งรับและส่งในตัวเดียวกัน ซึ่งในโครงการนี้เลือกใช้ให้ตัว TRW 2.4G เป็นทั้งตัวรับและตัวส่งในตัวเดียวกัน คือการรับการส่งจะเป็นแบบ Half Duplex คือ สลับหน้าที่กันระหว่างการเป็นตัวรับ และตัวส่ง

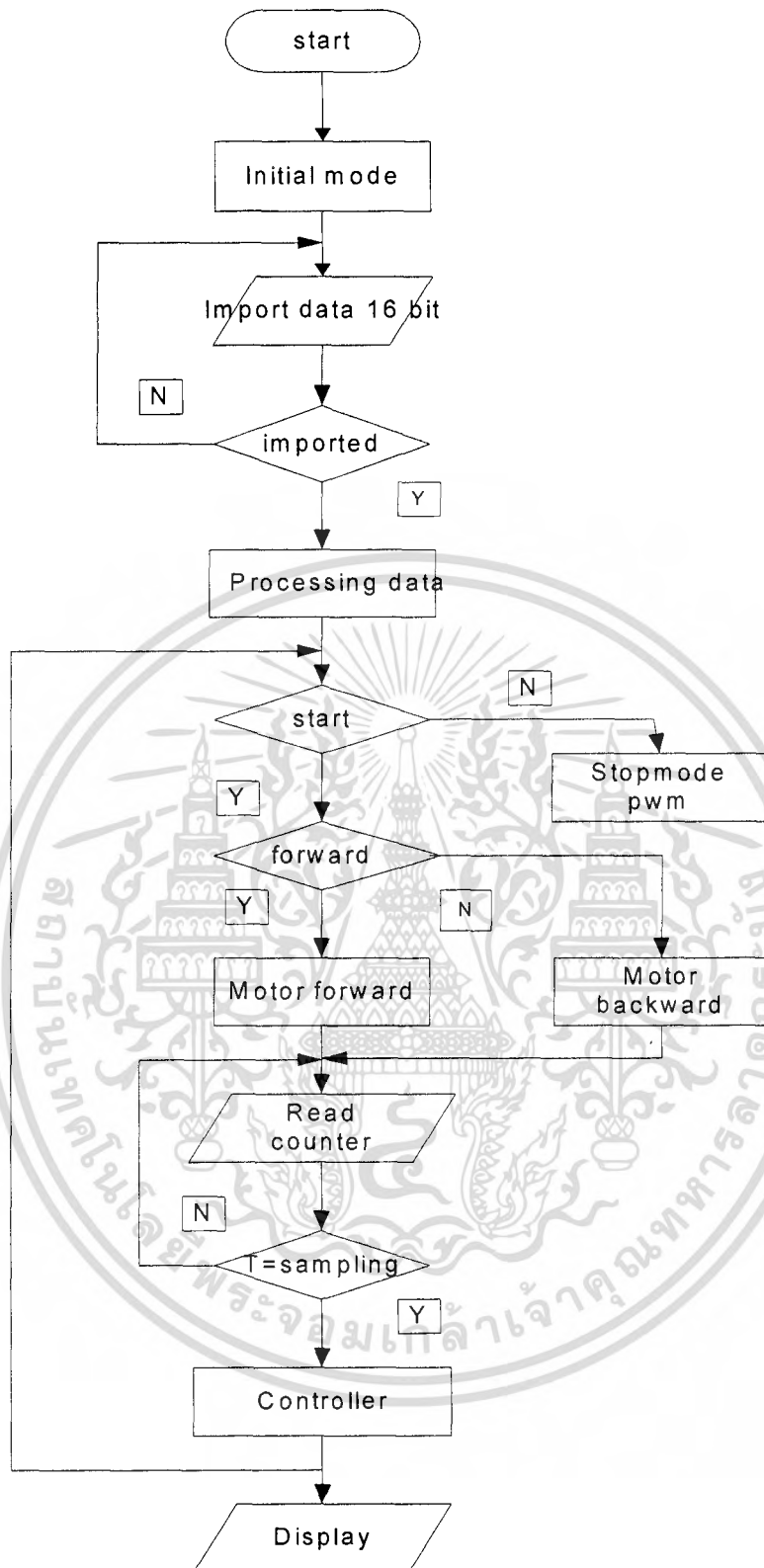


รูปที่ 3.7 บล็อกไดอะแกรมของการรับและส่งข้อมูล

### 3.5 การควบคุมการทำงาน

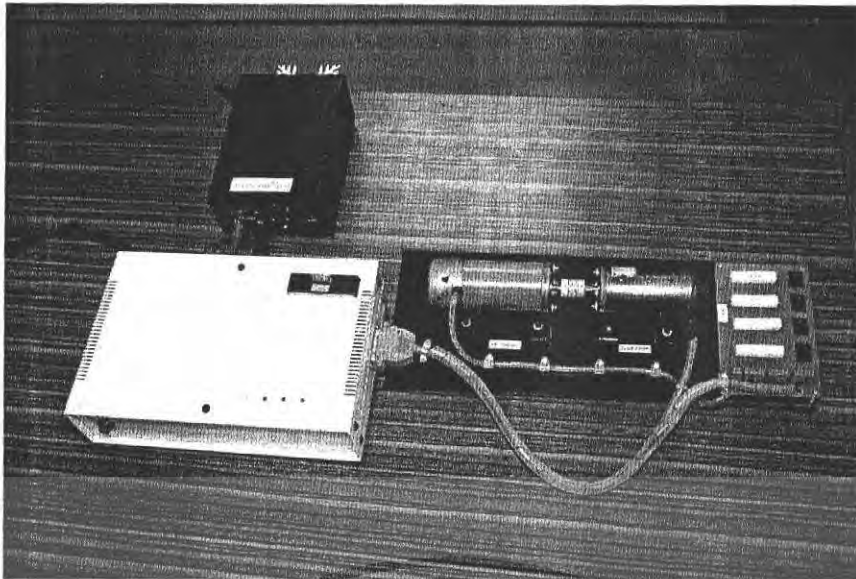
เริ่มต้นโปรแกรมจะรอรับข้อมูลจากคอมพิวเตอร์ ซึ่งจะส่งผ่านอากาศเป็นตัวกลางผ่านมา เป็นข้อมูลขนาด 16 บิต โดย 10 บิตล่างเป็นข้อมูลความเร็วและอีก 6 บิตที่เหลือเป็นข้อมูลคำสั่ง ทำงานและหยุดทำงาน และคำสั่งเพิ่ม โหลด

เมื่อได้รับคำสั่งมาแล้วตัวไมโครคอนโทรลเลอร์ จะประมวลผลตามคำสั่งที่ได้รับ และในขณะที่ไมโครคอนโทรลเลอร์ทำงานอยู่นั้น ได้มีการเปิดบริการอินเตอร์รัพต์ จากพอร์ตอนุกรมตลอดเวลาเพื่อรอรับค่าข้อมูลมาประมวลผล ซึ่งก็คือการรับค่าเป้าหมายค่าใหม่ และคำสั่งต่างๆ ที่ได้กล่าวข้างต้นนั่นเอง ขณะที่ส่วนควบคุมกระบวนการทำงานอยู่นั้นจะมีการติดต่อกับส่วนคอมพิวเตอร์ตลอดเวลาเพื่อส่งข้อมูลไปแสดงผลที่หน้าจอคอมพิวเตอร์และรับข้อมูลมาประมวลผล โดยส่งข้อมูลผ่านอากาศที่ความถี่ 2.4 GHz ใช้อุปกรณ์รับส่งข้อมูล TRW 2.4G



รูปที่ 3.8 โพล์วชาร์ตแสดงการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ส่วนควบคุมกระบวนกร

### 3.5.1 การออกแบบและทดสอบ

รูปแบบการรับส่งข้อมูลระหว่างส่วนคอมพิวเตอร์และส่วนควบคุมกระบวนกรจะส่งข้อมูลที่มีขนาด 16 บิต ซึ่งแต่ละบิตมีหน้าที่เป็นดังนี้



รูปที่ 3.10 รูปแบบการส่งข้อมูล 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 0-10 กำหนดความเร็วของมอเตอร์ที่ต้องการ ซึ่งมีค่าจำนวนเต็ม 0-2048

บิต 11 กำหนดการทำงานและหยุดทำงานของมอเตอร์ ถ้าเป็น“1” หมายความว่าให้มอเตอร์ทำงาน ถ้าเป็น “0” หมายความว่าให้มอเตอร์หยุดทำงาน

บิต 12 กำหนดทิศทางการหมุนของมอเตอร์ ถ้าเป็น “1” หมายความว่าให้มอเตอร์หมุนตามเข็มนาฬิกา ถ้าเป็น “0” หมายความว่าให้มอเตอร์หมุนทวนเข็มนาฬิกา

บิต 13-15 กำหนดการเพิ่มโหลด ซึ่งมีโหลดเป็นความต้านทานสี่ตัวต่อขนานกัน การเพิ่มโหลดด้วยบิต 13-15 อธิบายเป็นตารางดังนี้

ตารางที่ 3.1 การกำหนดการเพิ่มโหลดด้วยบิต 13-15

บิต 15	บิต 14	บิต 13	จำนวนโหลด
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

เมื่อกำหนดค่าตามที่ต้องการ ในแต่ละบิตแล้ว ค่าประจำหลักในแต่ละบิตจะถูกนำมาบวกกัน ซึ่งจะเป็นจำนวนเต็มที่อยู่ระหว่าง 0-65535 ไมโครคอนโทรลเลอร์ในส่วนควบคุมกระบวนการนี้จะนำข้อมูลนี้ไปประมวลผล

### 3.5.2 การอ่านค่าความเร็วรอบของมอเตอร์จากเอ็นโค้ดเดอร์และไอซีเคาน์เตอร์

จำนวนพัลส์ที่เอาต์พุตของเอ็นโค้ดเดอร์ถูกนับโดยไอซีเคาน์เตอร์ซึ่งมีขนาด 16 บิต จำนวนพัลส์ในไอซีเคาน์เตอร์จะเพิ่มขึ้นทุกๆ ขอบขาลงของสัญญาณเอาต์พุตของเอ็นโค้ดเดอร์ที่ต่อเป็น อินพุตให้ขาล็อกของไอซีเคาน์เตอร์ จากนั้นจึงทำการต่อเอาต์พุตของไอซีเคาน์เตอร์นำมาต่อร่วมกับพอร์ตของไมโครคอนโทรลเลอร์ และเขียนโปรแกรมอ่านข้อมูลจากพอร์ตอินพุต แล้วนำมาเข้าสมการเพื่อคำนวณแปลงเป็นความเร็วรอบต่ออนาที ตามสมการต่อไปนี้

$$\text{รอบ/นาที} = (\text{จำนวนพัลส์/Slit}) * N * 60 \quad (31)$$

$$\text{Slit} = 200$$

N คือ ค่านำมาคูณกับเวลาที่ใช้นับพัลส์แล้วมีค่าเท่ากับ 1 นาที เช่น ถ้าเวลาที่ใช้ในการนับเท่ากับ 200 ms เพราะฉะนั้น  $N=5$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลอง

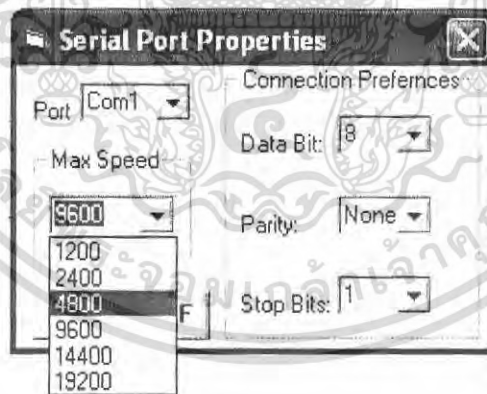
ในการทดลองนี้เป็นการทดสอบกระบวนการ โดยที่เราได้ตั้งค่าความเร็วรอบของมอเตอร์ที่ต้องการควบคุม ผ่านทางหน้าจอคอมพิวเตอร์ที่ความเร็วค่าต่างๆ จากนั้นก็จะมีการทำงานและส่งค่าความเร็วกลับมาแสดงที่หน้าจอคอมพิวเตอร์และมีการเก็บบันทึกค่าสถานะต่างเก็บไว้ เพื่อเป็นข้อมูลของการทำงานของมอเตอร์ และสามารถที่ดึงข้อมูลของการทำงานของตัวมอเตอร์นี้ขึ้นมาดูเพื่อนำไปวิเคราะห์การทำงานต่อไป

#### 4.1 การสั่งงานการทำงาน

เริ่มต้นโดยการที่กำหนดค่าอัตราบอดเรต ของข้อมูลที่เราต้องการส่งผ่าน จากนั้นเลือกโหมดการทำงาน ของการตั้งเวลา การบันทึกข้อมูล และเลือกค่าความเร็วรอบของมอเตอร์

##### 4.1.1 การเลือกค่าอัตราบอดของข้อมูล

การเลือกค่าอัตราบอด (baud rate) นั้น จะมีหน้าต่างการทำงานใหม่ขึ้นมาซึ่งในหน้าต่างนี้สามารถที่จะเลือก พอร์ตของ RS-232 ว่าต้องการใช้พอร์ตที่เท่าไร ค่าอัตราบอดเรต จำนวนบิตของข้อมูล พาริตีบิต บิตหยุด ว่าต้องการอย่างไร ได้ตามต้องการ

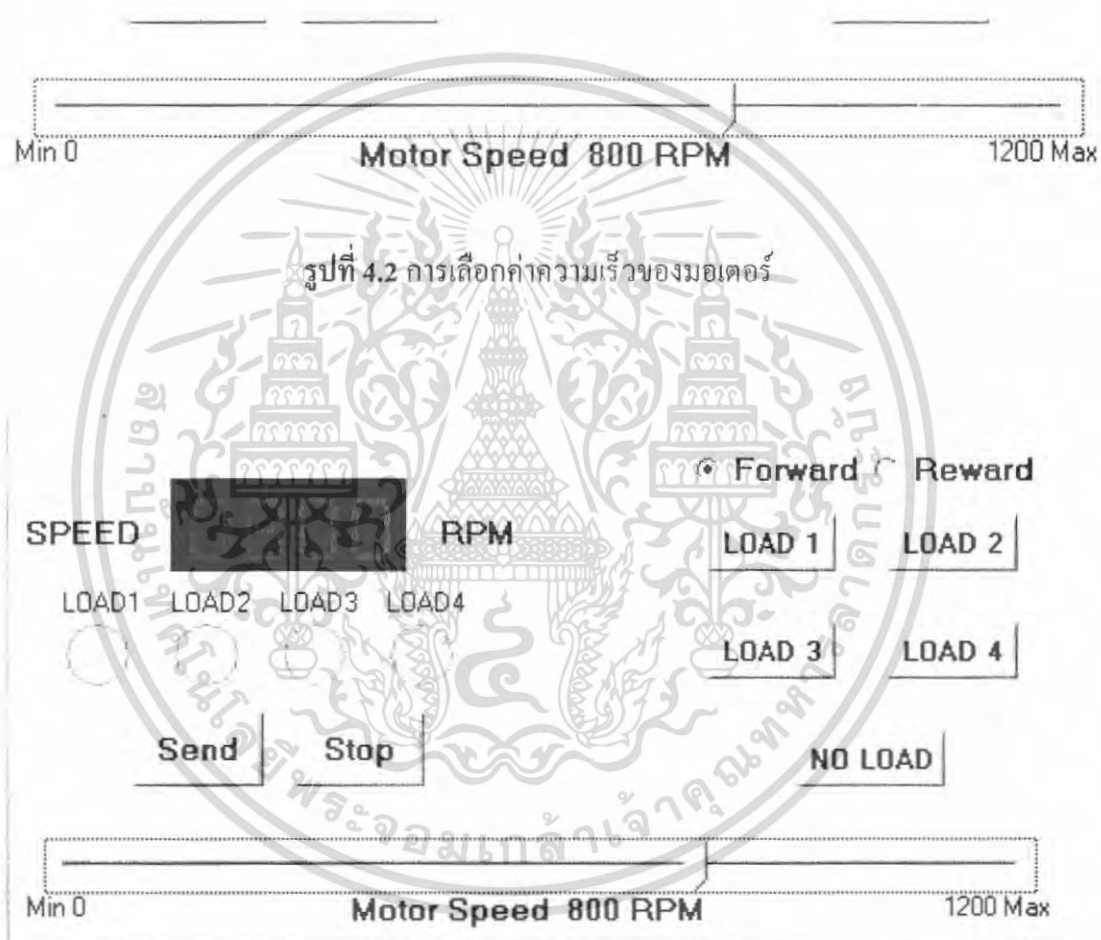


รูปที่ 4.1 การเลือกค่าอัตราบอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.2 การเลือกค่าความเร็วของมอเตอร์

การเลือกค่าความเร็วนี้ สามารถเลือกค่าความเร็วรอบของมอเตอร์ออกมาเป็นหน่วยความเร็วรอบต่อนาที (rpm) เลือกได้โดยการคลิกเลื่อนแถบค่าความเร็วที่กำหนดไว้ให้ และค่าความเร็วที่เลือกไว้จะปรากฏอยู่ที่ด้านล่างของแถบค่าความเร็ว และเมื่อเลือกค่าความเร็ว จากนั้นทำการส่งค่าความเร็วที่ต้องการออกไปนั้น โปรแกรมก็จะรอการส่งค่าการทำงานกลับมาเพื่อที่จะนำค่าที่วัดได้มาแสดงที่หน้าจอกอมพิวเตอร์ เพื่อให้ผู้ใช้งาน ได้ทราบว่าค่าความเร็วที่ทำงานได้ในขณะปัจจุบันเป็นเท่าไร

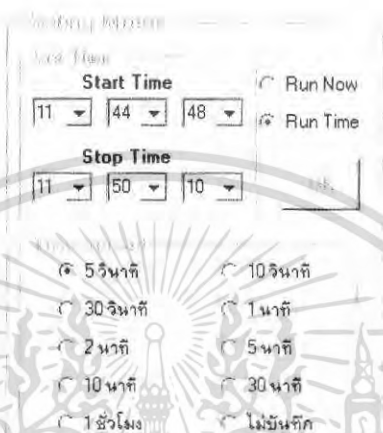


รูปที่ 4.3 ผลของการวัดค่าการทำงานเวลาปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.3 การตั้งค่าสถานะต่างๆของการทำงาน

ในการทำงานของมอเตอร์นั้นได้มีการเลือกการ กำหนดการเก็บค่าสถานะการทำงานต่างๆไว้ตามความต้องการของผู้ใช้งาน ซึ่งการเก็บข้อมูลนี้ไว้ นั้น จะถือว่าข้อมูลนี้เป็นข้อมูลดิบของมอเตอร์ ซึ่งผู้ใช้งานนั้นสามารถที่จะดึงข้อมูลเฉพาะ ข้อมูลที่ผู้ใช้งานต้องการทราบได้ ขึ้นมาดูได้สะดวก การเลือกการเก็บบันทึกข้อมูลนั้นสามารถเลือกได้ตามช่องที่กำหนดไว้ได้



รูปที่ 4.4 เลือกโหมดการบันทึกค่าสถานะ

เมื่อมีการเลือกค่าสถานะการบันทึกข้อมูลแล้ว โปรแกรมจะบันทึกเก็บข้อมูลการทำงาน ของมอเตอร์เก็บไว้ในฐานข้อมูล ตามช่วงเวลาที่ได้ถูกกำหนดไว้ในตอนต้น ผู้ใช้สามารถดูค่าการทำงานของมอเตอร์ได้จากหน้าต่างควบคุมหลัก ซึ่งในหน้าต่างนี้จะแสดงค่าการทำงาน เฉพาะ 4 ค่าหลังสุดของการเก็บบันทึกเท่านั้น

ตารางบันทึกค่า

Time Start	Time Stop	Time Total	Time	Set Speed	Speed	Date	Load
9:56:56	9:57:00	0:00:04	19:52:47	980	980	12/11/2549	0
10:16:00	10:17:00	0:01:00	19:52:52	980	980	12/11/2549	0
13:02:11	13:03:00	0:00:49	19:52:57	980	980	12/11/2549	0
19:51:31	19:53:00	0:01:29	19:53:02	980	980	12/11/2549	0

รูปที่ 4.5 การบันทึกค่าสถานะต่างของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## 4.2 ผลการทดสอบการเปลี่ยนค่าพารามิเตอร์ของตัวควบคุม

ทำการทดสอบกระบวนการควบคุมความเร็วของมอเตอร์ โดยการทำการปรับค่าพารามิเตอร์ต่างๆ ของตัวควบคุมกระบวนการ ซึ่งตัวควบคุมที่ใช้ในการ ควบคุมความเร็วของมอเตอร์นี้ ใช้ตัวควบคุมกระบวนการคือ ตัวควบคุมแบบ พีไอ ( PI Controller ) ซึ่งแปรที่ต้องทำการปรับตั้ง หรือพารามิเตอร์ต่างๆ มีอยู่ 2 ตัว คือ  $K_p$  และ  $K_i$

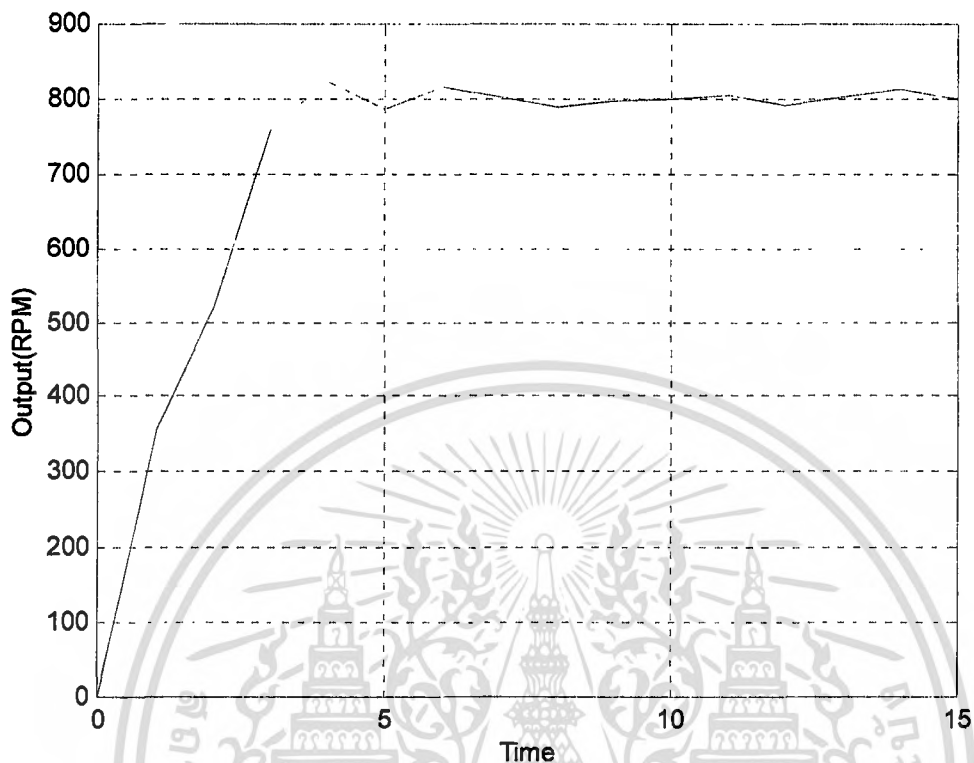
$$\begin{aligned} K_p &= 10 & K_i &= 80 \\ K_d &= 0 & T &= 200\text{ms} \end{aligned}$$

ตารางที่ 4.1 ผลการทดลองที่ความรอบ 800 รอบต่อนาที เมื่อ  $K_p = 10$  และ  $K_i = 80$

เวลา	ค่าเป้าหมาย	ค่าที่วัดได้
T	800	357
2T	800	520
3T	800	760
4T	800	823
5T	800	786
6T	800	815
7T	800	802
8T	800	788
9T	800	798
10T	800	800
11T	800	805
12T	800	793
13T	800	803
14T	800	814
15T	800	799

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่าที่ได้จากตารางที่ 4.1 นำมาพล็อตกราฟ เพื่อดูผลตอบสนองของการทำงานของตัวมอเตอร์ เพื่อดูค่าของความเร็วในการตอบสนองของระบบ



รูปที่ 4.8 กราฟผลตอบสนองของระบบ เมื่อ  $K_p = 10$  และ  $K_i = 80$

จากกราฟการทำงานในช่วง ค่าพารามิเตอร์  $K_p = 10$  และ  $K_i = 80$  จะเห็นได้ว่า ผลการทำงานของระบบยังคงช้าอยู่มาก จึงต้องทำการปรับแต่งค่าพารามิเตอร์ของระบบใหม่ เพื่อที่จะให้การทำงานของระบบเข้าสู่เป้าหมายได้เร็วที่สุด

ทดลองเปลี่ยนค่าพารามิเตอร์ คือ

$$K_p = 2$$

$$K_i = 80$$

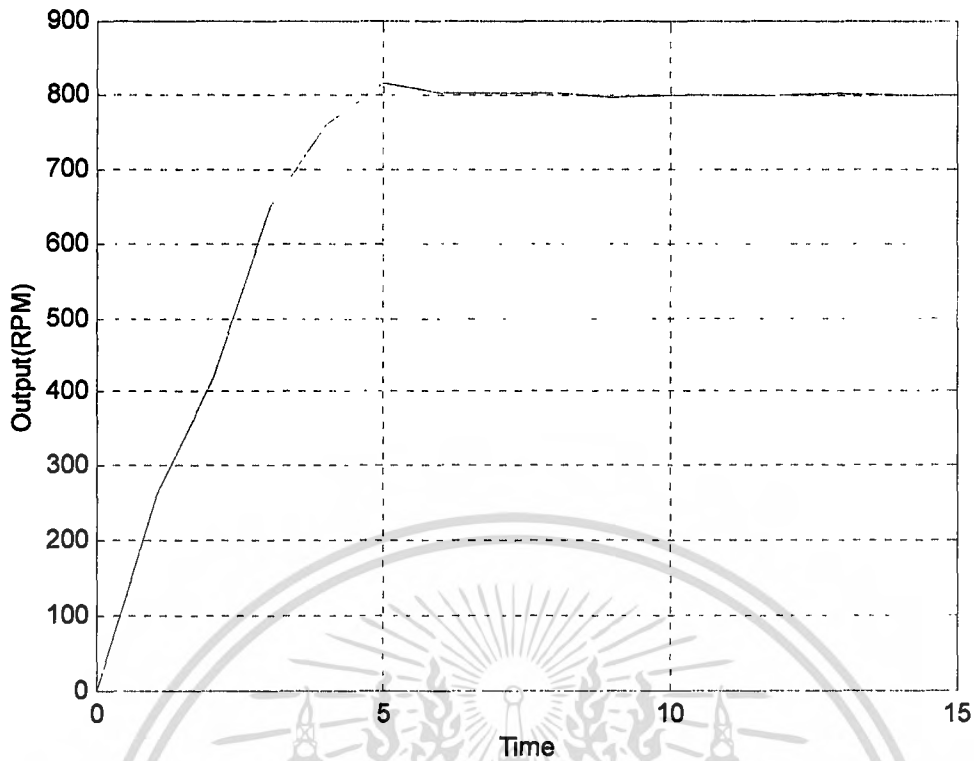
$$K_d = 0$$

$$T = 200\text{ms}$$

ตารางที่ 4.2 ผลการทดลองที่ความรอบ 800 รอบต่อนาที เมื่อ  $K_p = 2$  และ  $K_i = 80$

เวลา	ค่าเป้าหมาย	ค่าที่วัดได้
T	800	257
2T	800	420
3T	800	650
4T	800	760
5T	800	815
6T	800	803
7T	800	802
8T	800	802
9T	800	798
10T	800	800
11T	800	801
12T	800	801
13T	800	803
14T	800	800
15T	800	799

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 กราฟผลตอบสนองของระบบ เมื่อ  $K_p = 2$  และ  $K_i = 80$

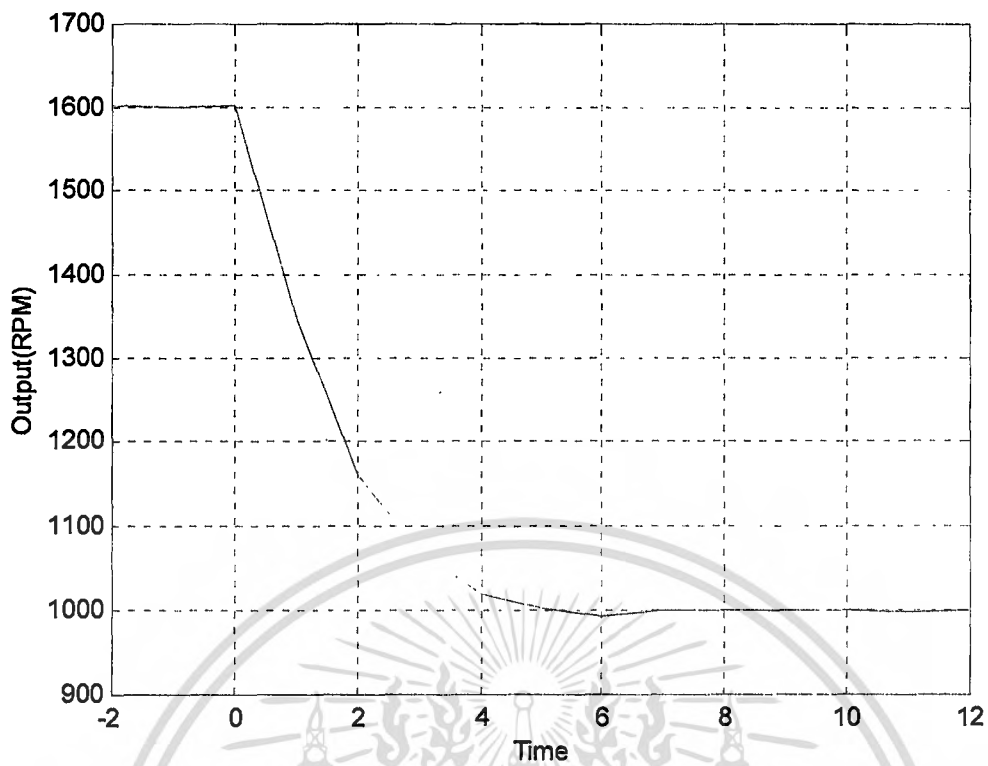
จากรูปที่ 4.9 เป็นผลตอบสนองการทำงานของมอเตอร์ ซึ่งตั้งความเร็วรอบที่ 800 รอบต่อนาที โดยที่ไม่มีการต่อโหลดให้กับมอเตอร์ ในสภาวะคงตัวผลตอบสนอง หรือความเร็วของมอเตอร์สามารถเข้าสู่เป้าหมายได้ ในขณะที่เดียวกันเมื่อทำการรบกวนระบบ คือการเพิ่มโหลดให้กับมอเตอร์ ซึ่งจะทำให้ความเร็วรอบของมอเตอร์ตกลงไปชั่วขณะหนึ่ง แล้วกับเข้าสู่เป้าหมาย ตามหลักการของระบบควบคุมวงปิด

ทำการทดลองปรับเปลี่ยนความเร็วของมอเตอร์ โดยให้ความเร็วลดลงจาก 1600 รอบต่อนาที เหลือ 1000 รอบต่อนาที ผลการทดลองแสดงโดยตารางและกราฟต่อไปนี้

ตารางที่ 4.3 ผลการทดลองเปลี่ยนความรอบ 1600 เป็น 1000 รอบต่อนาที

เวลา	ค่าเป้าหมาย	ค่าที่วัด
t-2T	1600	1602
t-T	1600	1600
t	1600	1601
t+T	1000	1345
t+2T	1000	1160
t+3T	1000	1070
t+4T	1000	1020
t+5T	1000	1003
t+6T	1000	1995
t+7T	1000	1002
t+8T	1000	1001
t+9T	1000	1000
t+10T	1000	1002
t+11T	1000	999
t+12T	1000	1000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 กราฟผลตอบสนองของระบบที่เปลี่ยนความเร็วรอบ 1600 เป็น 1000 รอบต่อนาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

## บทวิจารณ์และสรุป

### 5.1 บทวิจารณ์

ในการทำปริญญาานิพนธ์ฉบับนี้ ได้จัดทำขึ้นเพื่อทำการควบคุมกระบวนการ คือ การควบคุมความเร็วของมอเตอร์ ซึ่งสามารถทำการควบคุม สั่งงาน และดูผลการทำงาน รวมถึงการเก็บบันทึกค่าการทำงาน ในเวลาต่างๆ ได้ เพื่อที่สามารถเรียกดูข้อมูลการทำงาน ในช่วงเวลาต่างๆ ได้อย่างสะดวก ซึ่งในการส่งผ่านข้อมูลของการสั่งงาน และการแสดงผลนั้นทำการส่งข้อมูลแบบไร้สาย โดยผ่านตัวส่งสัญญาณ TRW 2.4G การสั่งงานและการแสดงผลนั้น ทำการสั่งงานผ่านทางหน้าจอคอมพิวเตอร์

จากผลการทดลองและสิ่งที่ได้ทั้งหมด คือการนำเสนอการควบคุมความเร็วของมอเตอร์ โดยควบคุมผ่านทางคอมพิวเตอร์ ในส่วนควบคุมกระบวนการได้ใช้ไมโครคอนโทรลเลอร์ PIC 16F877 เป็นหัวใจหลักในการควบคุมและประมวลผลการทำงานทั้งหมด เช่น

- 1 การอ่านค่าความเร็วรอบของมอเตอร์ผ่าน ไอซีเคาน์เตอร์
- 2 การอนูวัตตัวควบคุม PI แบบดิจิตอล
- 3 การสร้างสัญญาณ PWM
- 4 การติดต่อสื่อสารแบบอนุกรม (RS 232)

ซึ่งสามารถทำงานได้เป็นอย่างดี และแสดงให้เห็นว่า การควบคุมแบบดิจิตอลนั้นมีข้อดีคือสามารถปรับแต่งการทำงานของระบบได้ง่ายกว่าเพียงแค่แก้ไขในส่วนของซอฟต์แวร์ แต่ขั้นตอนการออกแบบจะมีความซับซ้อนมากกว่าการควบคุมแบบอนาล็อก

นอกจากไมโครคอนโทรลเลอร์ที่เป็นส่วนประกอบสำคัญแล้วส่วนประกอบที่สำคัญที่จะทำให้ระบบควบคุมไม่ว่าจะเป็น การควบคุมความเร็วหรือปริมาณทางฟิสิกส์ใดๆก็ตาม ให้สามารถทำงานได้อย่างมีประสิทธิภาพตามที่เรากำลังต้องการก็คือ ส่วน Actuator หรือ Amplifier และ ระบบการวัด ซึ่งเป็นสิ่งที่ต้องคำนึงถึงในการพิจารณานำมาใช้ งาน เช่น Amplifier สามารถขับพลังงานและทนต่อสภาพการทำงานของกระบวนการได้ดีหรือไม่ หรือระบบการวัด ต้องพิจารณาให้สอดคล้องกับปริมาณที่จะวัด วิธีการวัด การปรับแต่งสัญญาณ และการเชื่อมต่อสัญญาณที่ได้จากการตรวจวัดจากเซนเซอร์ เมื่อพิจารณาถึงส่วนประกอบในทุกด้านของระบบควบคุมแล้ว จะทำให้มีแนวทางในการออกแบบสร้าง และทำให้ผลที่ได้มีประสิทธิภาพ

## 5.2 ปัญหาในการทำงาน

1. การจัดส่งข้อมูลของการทำงาน การแสดงผล นั้นมีหลายค่า ทำให้เกิดความยุ่งยากในการส่งข้อมูล จึงจำเป็นต้องทำการนำข้อมูลทั้งหมด มาทำการเข้ารหัส และถอดรหัส เพื่อความสะดวก
2. การส่งผ่านข้อมูล ที่เป็นความถี่ อาจเกิดการรบกวน จากความถี่ใกล้เคียง สภาพแวดล้อม สภาพอากาศ ระยะทางระหว่างจุดรับและจุดส่งสัญญาณ ซึ่งจะมีผลต่อการส่งผ่านข้อมูล
3. การนับจำนวนพัลส์ของเอน โค้ดเดอร์ ไม่สามารถใช้ไมโครคอนโทรลเลอร์นับได้โดยตรง ต้องใช้ไอซีเคอร์เตอร์เป็นตัวนับแล้วส่งค่าจำนวนพัลส์ให้ไมโครคอนโทรลเลอร์
4. ในการส่งรหัสคำสั่งจากคอมพิวเตอร์ให้ส่วนควบคุมกระบวนการใช้อัตราบอดเท่ากับ 9600 ไม่ได้ เนื่องจากตัวรับและตัวส่งสัญญาณ TRW 2.4G ไม่สามารถรับส่งข้อมูลได้ทัน จึงใช้อัตราบอดเท่ากับ 4800 ในการรับและส่งข้อมูล

## 5.3 สรุปผลการทำงาน

โครงการนี้เป็น การควบคุมกระบวนการแบบไร้สาย ใช้การส่งผ่านข้อมูลผ่านทางคลื่นวิทยุ สามารถควบคุมกระบวนการได้ หรือว่าควบคุมระบบที่ต้องการให้มีค่าเป็นไปตามต้องการ โดยมีความสะดวก โดยการใช้ที่ไม่ต้องมีการเดินสายสัญญาณ เพื่อทำการติดต่อระหว่างส่วนควบคุม และกระบวนการ จึงไม่เกิดความยุ่งยากในส่วนนี้ เพราะใช้การติดต่อทางคลื่นวิทยุแทน จึงมีความคล่องตัว ในการที่จะเคลื่อนย้าย ตัวกระบวนการไปได้ หลากหลายที่ แต่ต้องอยู่ในรัศมีการทำงานของตัว ส่ง และรับสัญญาณ TRW 2.4G ด้วย จากหลักการนี้สามารถที่จะนำไปใช้ได้หลากหลาย และเกิดประโยชน์ ในการควบคุมในระบบอื่นๆ ต่อไป

## เอกสารอ้างอิง

1. สัจจะ จรัสรุ่งรวีร์, “คู่มือเขียนโปรแกรม Visual Basic ฉบับผู้เริ่มต้น”, บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซนเตอร์ จำกัด, 432 หน้า, 2548
2. ศุภชัย สมภานิช, “Advanced Database Programming ด้วย VB 6+VB .NET”, บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซนเตอร์ จำกัด, 384 หน้า, 2548
3. บัณฑิต จามรภูติ, “คู่มือการใช้งาน Protel 99”, บัณฑิตเพรส, 290 หน้า, 2544
4. ชาริน สิทธิธรรมชารี, “สร้างระบบฐานข้อมูลอย่างมืออาชีพ Access 2003”, บริษัท ชัคเชส มีเดีย จำกัด, 330 หน้า, 2549
5. โยธิน เปรมปราณีรัชต์, “SOLID-STATE DC MOTOR CONTROL”, ภาควิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
6. เดชฤทธิ์ มณีธรรม, “คัมภีร์ไมโครคอนโทรลเลอร์ PIC (Microcontroller PIC)”, กรุงเทพฯ :เคทีพี คอมพ์ แอนด์ คอนซัลท์, 2549, 368 หน้า.
7. ณัฐพล วงศ์สุนทรชัย “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC 16F877”, อินโนเวตีฟ เอ็ดจิวอร์เมนท์ จำกัด.
8. ประจัน พลังสันติกุลม, “เรียนรู้และใช้งาน CCS C คอมไพเลอร์”, อินโนเวตีฟ เอ็ดจิวอร์เมนท์ จำกัด.
9. วรพงษ์ ตั้งศรีรัตน์, “เซนเซอร์ และทรานสดิวเซอร์”, สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 292 หน้า, กรุงเทพฯ, 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## RF-MODULE TRW-2.4G

Ensemble list

### Transmitter

TRW-2.4G(250kbps)

TRW-2.4G(1Mbps)

Surface And Size View

### Receiver

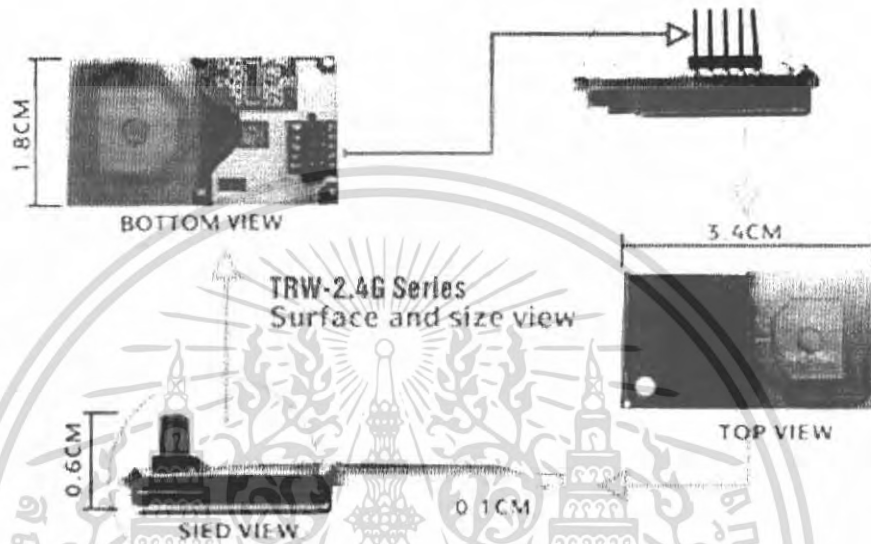
TRW-2.4G(250kbps)

TRW-2.4G(1Mbps)

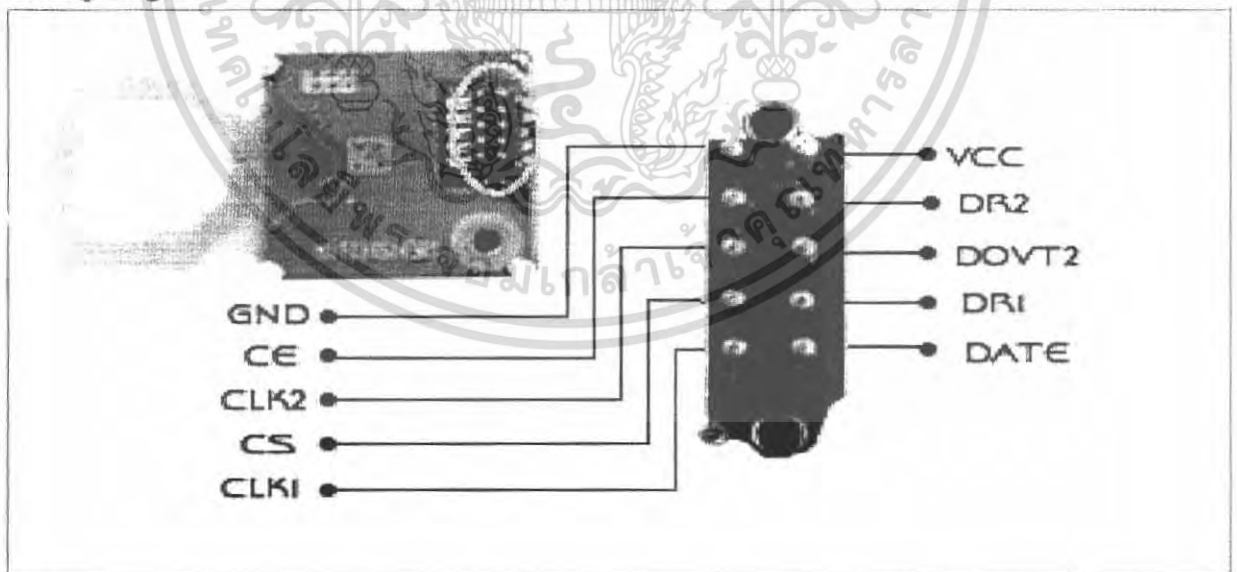
### Distance

280m

150m



Wiring Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# WENSHING

## TRW -2.4GHz Radio Transceiver

Conditions: VDD = +3V, VSS = 0V, T<sub>A</sub> = -40°C to +85°C

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
<b>Operating conditions</b>						
VDD	Supply voltage		1.9	3.0	3.6	V
TEMP	Operating Temperature		-40	+27	+85	°C
<b>Digital input pin</b>						
V <sub>IH</sub>	HIGH level input voltage		VDD-0.3		VDD	V
V <sub>IL</sub>	LOW level input voltage		V <sub>SS</sub>		0.3	V
<b>Digital output pin</b>						
V <sub>OH</sub>	HIGH level output voltage (I <sub>OH</sub> =-0.5mA)		VDD-0.3		VDD	V
V <sub>OL</sub>	LOW level output voltage (I <sub>OL</sub> =0.5mA)		V <sub>SS</sub>		0.3	V
<b>General RF conditions</b>						
f <sub>OP</sub>	Operating frequency	1)	2400		2524	MHz
Δf	Frequency deviation			±156		kHz
R <sub>GENK</sub>	Data rate ShockBurst™		±0		1000	kbps
f <sub>CHANNEL</sub>	Channel spacing			1		MHz
<b>Transmitter operation</b>						
P <sub>RF</sub>	Maximum Output Power	4)		0	+4	dBm
P <sub>RFc</sub>	RF Power Control Range		16	20		dB
P <sub>RFcR</sub>	RF Power Control Range Resolution				±3	dB
P <sub>BW</sub>	20dB Bandwidth for Modulated Carrier				1000	kHz
P <sub>RF2</sub>	2 <sup>nd</sup> Adjacent Channel Transmit Power 2MHz				-20	dBm
P <sub>RF3</sub>	3 <sup>rd</sup> Adjacent Channel Transmit Power 3MHz				-40	dBm
I <sub>VDD</sub>	Supply current @ 0dBm output power	5)		13		mA
I <sub>VDD</sub>	Supply current @ -20dBm output power	5)		8.8		mA
I <sub>VDD</sub>	Average Supply current @ -5dBm output power, ShockBurst™	6)		0.8		mA
I <sub>VDD</sub>	Average Supply current in stand-by mode	7)		12		μA
I <sub>VDD</sub>	Average Supply current in power down			1		μA
<b>Receiver operation</b>						
I <sub>VDD</sub>	Supply current one channel 250kbps			18		mA
I <sub>VDD</sub>	Supply current one channel 1000kbps			19		mA
I <sub>VDD</sub>	Supply current two channels 250kbps			23		mA
I <sub>VDD</sub>	Supply current two channels 1000kbps			25		mA
RX <sub>SENS</sub>	Sensitivity at 0.1%BER (@250kbps)			-90		dBm
RX <sub>SENS</sub>	Sensitivity at 0.1%BER (@1000kbps)			-80		dBm
C/I <sub>CO</sub>	C/I Co-channel			6		dB
C/I <sub>1S1</sub>	1 <sup>st</sup> Adjacent Channel Selectivity C/I 1MHz			-1		dB
C/I <sub>2ND</sub>	2 <sup>nd</sup> Adjacent Channel Selectivity C/I 2MHz			-16		dB
C/I <sub>3RD</sub>	3 <sup>rd</sup> Adjacent Channel Selectivity C/I 3MHz			-26		dB
RX <sub>II</sub>	Blocking Data Channel 2			-41		dB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## WENSHING

### TRW -2.4GHz Radio Transceiver

#### ShockBurst™

The ShockBurst™ technology uses on-chip FIFO to clock in data at a low data rate and transmit at a very high rate thus enabling extremely power reduction.

When operating the TRW-2.4G in ShockBurst™, you gain access to the high data rates (1 Mbps) offered by the 2.4 GHz band without the need of a costly, high-speed micro controller (MCU) for data processing.

By putting all high speed signal processing related to RF protocol on-chip, the TRW-2.4G offers the following benefits:

- Highly reduced current consumption
- Lower system cost (facilitates use of less expensive micro controller)
- Greatly reduced risk of 'on-air' collisions due to short transmission time

The TRW-2.4G can be programmed using a simple 3-wire interface where the data rate is decided by the speed of the micro controller.

By allowing the digital part of the application to run at low speed while maximizing the data rate on the RF link, the nRF ShockBurst™ mode reduces the average current consumption in applications considerably.

#### ShockBurst™ principle

When the TRW-2.4G is configured in ShockBurst™, TX or RX operation is conducted in the following way (10 kbps for the example only).

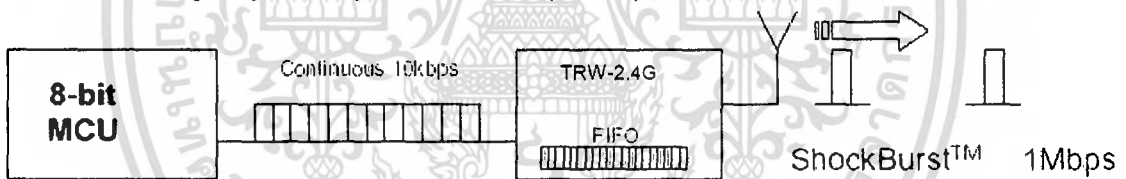


Figure 4 Clocking in data with MCU and sending with ShockBurst™ technology

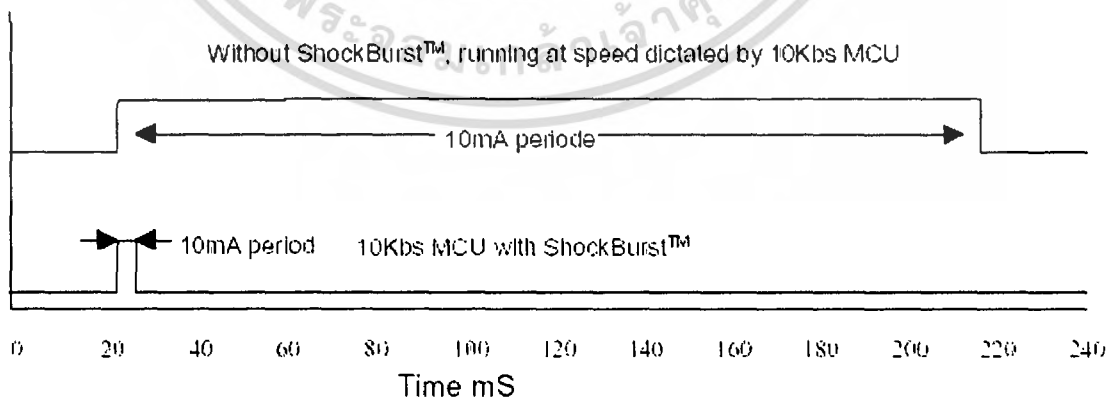


Figure 4 Current consumption with & without ShockBurst™ technology

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# WENSHING

## TRW -2.4GHz Radio Transceiver

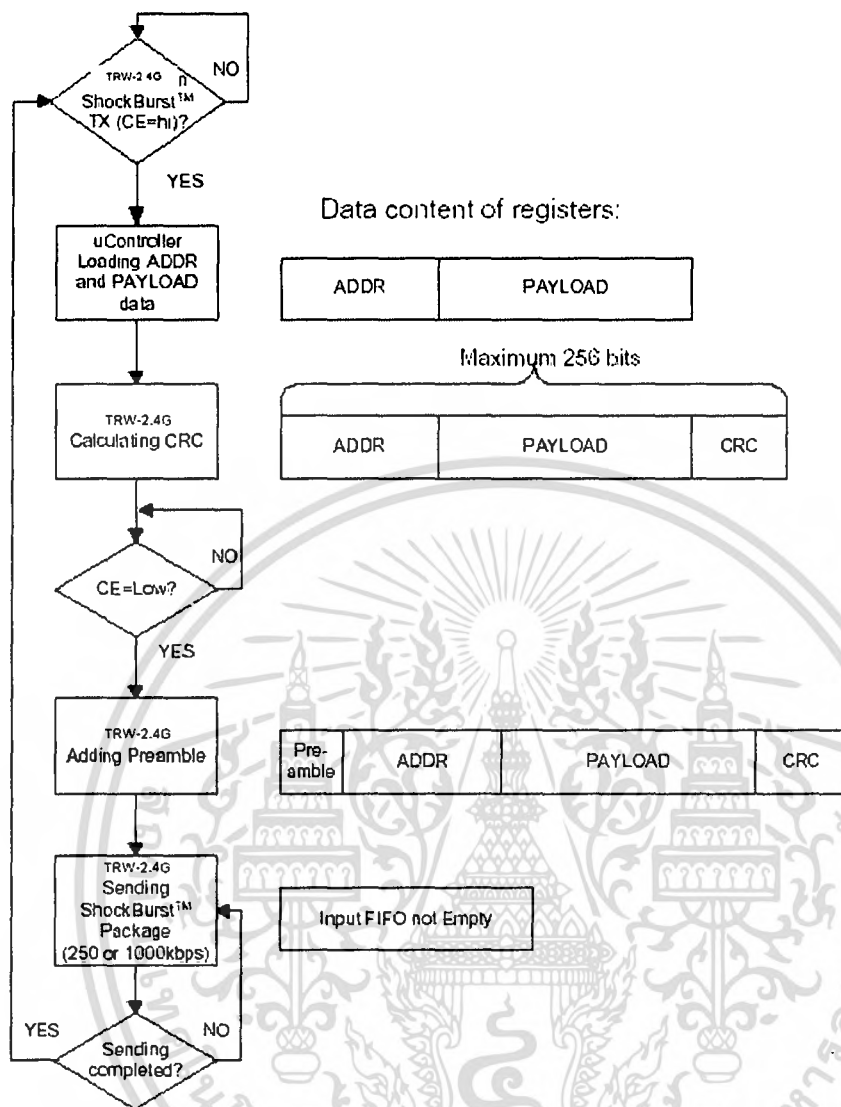


Figure 2 Flow Chart ShockBurst™ Transmit of TRW-2.4G

### nRF2401 ShockBurst™ Transmit:

MCU interface pins: CE, CLK1, DATA

1. When the application MCU has data to send, set CE high. This activates TRW-2.4G on-board data processing.
2. The address of the receiving node (RX address) and payload data is clocked into the TRW-2.4G. The application protocol or MCU sets the speed <math>< 1\text{Mbps}</math> (ex: 10kbps).
3. MCU sets CE low, this activates a TRW-2.4G ShockBurst™ transmission.
4. TRW-2.4G ShockBurst™:
  - RF front end is powered up
  - RF package is completed (preamble added, CRC calculated)
  - Data is transmitted at high speed (250 kbps or 1 Mbps configured by user).
  - TRW-2.4G return to stand-by when finished

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# WENSHING

## TRW-2.4GHz Radio Transceiver

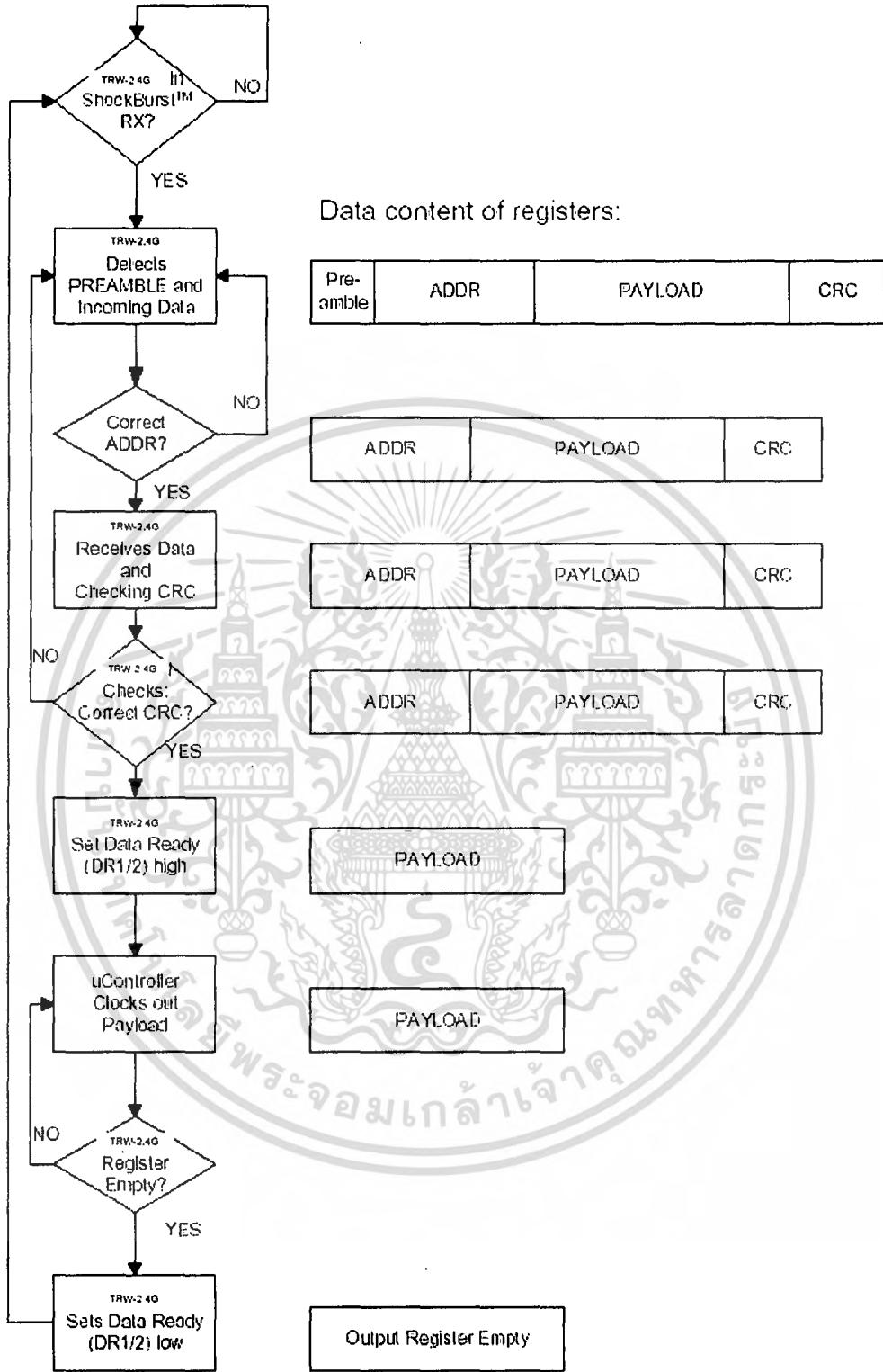


Figure 3 Flow Chart ShockBurst™ Receive of TRW-2.4G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

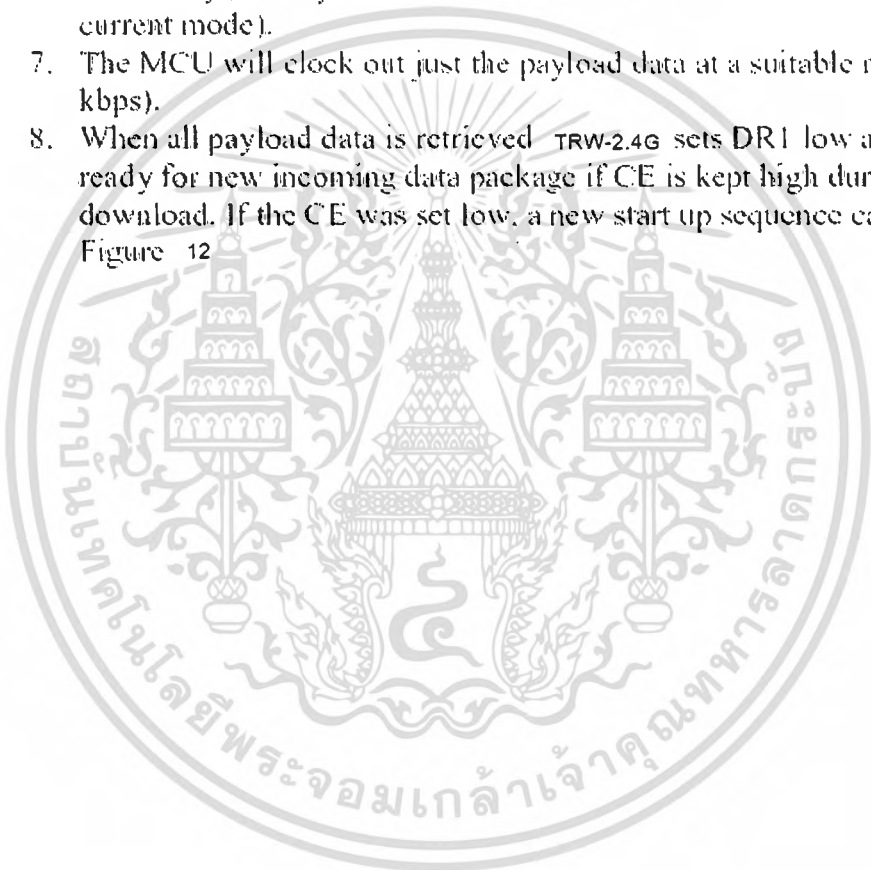
# WENSHING

## TRW-2.4GHz Radio Transceiver

### TRW-2.4G ShockBurst™ Receive:

MCU interface pins: CE, DR1, CLK1 and DATA (one RX channel receive)

1. Correct address and size of payload of incoming RF packages are set when TRW-2.4G is configured to ShockBurst™ RX.
2. To activate RX, set CE high.
3. After 200  $\mu$ s settling, TRW-2.4G is monitoring the air for incoming communication.
4. When a valid package has been received (correct address and CRC found), TRW-2.4G removes the preamble, address and CRC bits.
5. TRW-2.4G then notifies (interrupts) the MCU by setting the DR1 pin high.
6. MCU may (or may not) set the CE low to disable the RF front end (low current mode).
7. The MCU will clock out just the payload data at a suitable rate (ex. 10 kbps).
8. When all payload data is retrieved TRW-2.4G sets DR1 low again, and is ready for new incoming data package if CE is kept high during data download. If the CE was set low, a new start up sequence can begin, see Figure 12



## WENSHING

TRW -2.4GHz Radio Transceiver

### DuoCeiver™ Simultaneous Two Channel Receive Mode

In both ShockBurst™ . . . modes the TRW-2.4G can facilitate simultaneous reception of two parallel independent frequency channels at the maximum data rate. This means:

- TRW-2.4G can receive data from two 1 Mbps transmitters (ex: TRW-2.4G or TRW-2.4G\_) 8 MHz (8 frequency channels) apart through one antenna interface.
- The output from the two data channels is fed to two separate MCU interfaces.
  - Data channel 1: CLK1, DATA, and DR1
  - Data channel 2: CLK2, DOUT2, and DR2
  - DR1 and DR2 are available only in ShockBurst™.

The TRW-2.4G DuoCeiver™ technology provides 2 separate dedicated data channels for RX and replaces the need for two, stand alone receiver systems.

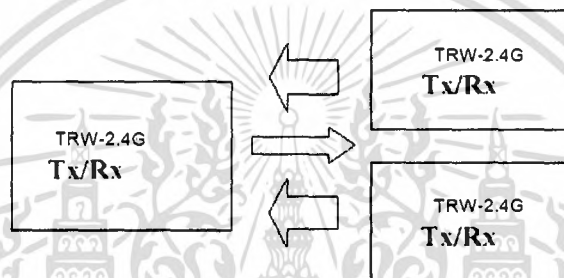


Figure 4 Simultaneous 2 channel receive on TRW-2.4G

There is one absolute requirement for using the second data channel. For the TRW-2.4G to be able to receive at the second data channel the frequency channel must be 8MHz higher than the frequency of data channel 1. The TRW-2.4G must be programmed to receive at the frequency of data channel 1. No time multiplexing is used in TRW-2.4G to fulfil this function. In direct mode the MCU must be able to handle two simultaneously incoming data packets if it is not multiplexing between the two data channels. In ShockBurst™ it is possible for the MCU to clock out one data channel at a time while data on the other data channel waits for MCU availability, without any lost data packets, and by doing so reduce the needed performance of the MCU.

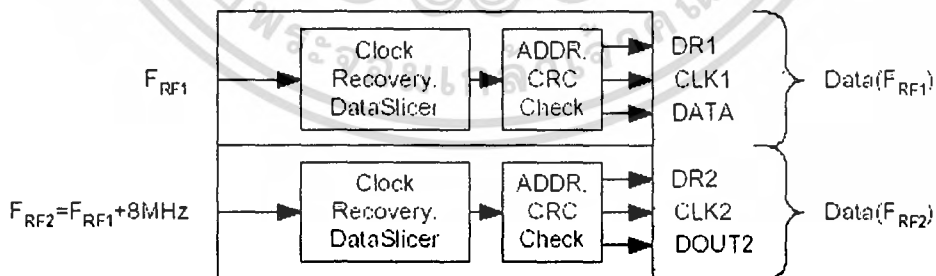


Figure 5 DuoCeiver™ with two simultaneously independent receive channels.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## WENSHING

TRW -2.4GHz Radio Transceiver

### DEVICE CONFIGURATION

All configuration of the TRW-2.4G is done via a 3-wire interface to a single configuration register. The configuration word can be up to 15 bytes long for ShockBurst™

#### Configuration for ShockBurst™ operation

The configuration word in ShockBurst™ enables the TRW-2.4G to handle the RF protocol. Once the protocol is completed and loaded into TRW-2.4G only one byte, bit[7:0], needs to be updated during actual operation.

The configuration blocks dedicated to ShockBurst™ is as follows:

- Payload section width: Specifies the number of payload bits in a RF package. This enables the TRW-2.4G to distinguish between payload data and the CRC bytes in a received package
- Address width: Sets the number of bits used for address in the RF package. This enables the TRW-2.4G to distinguish between address and payload data.
- Address (RX Channel 1 and 2): Destination address for received data.
- CRC: Enables nRF2401 on-chip CRC generation and de-coding.

#### NOTE:

These configuration blocks, with the exception of the CRC, are dedicated for the packages that a TRW-2.4G is to receive.

In TX mode, the MCU must generate an address and a payload section that fits the configuration of the TRW-2.4G that is to receive the data.

When using the TRW-2.4G on-chip CRC feature ensure that CRC is enabled and uses the same length for both the TX and RX devices.

PRE-AMBLE	ADDRESS	PAYLOAD	CRC
-----------	---------	---------	-----

Figure 10 Data packet set-up

# WENSHING

TRW -2.4GHz Radio Transceiver

## Configuration Word overview

	Bit position	Number of bits	Name	Function	
ShockBurst™ configuration	143:120	24	TEST	Reserved for testing	
	119:112	8	DATA2_W	Length of data payload section RX channel 2	
	111:104	8	DATA1_W	Length of data payload section RX channel 1	
	103:64	40	ADDR2	Up to 5 byte address for RX channel 2	
	63:24	40	ADDR1	Up to 5 byte address for RX channel 1	
	23:18	6	ADDR_W	Number of address bits (both RX channels).	
	17	1	CRC_1	8 or 16 bit CRC	
	16	1	CRC_EN	Enable on-chip CRC generation/checking.	
General device configuration		15	1	RX2_EN	Enable two channel receive mode
		14	1	CM	Communication mode (Direct or ShockBurst™)
		13	1	RFDR_SB	RF data rate (1Mbps requires 16MHz crystal)
		12:10	3	XO_F	Crystal frequency
		9:8	2	RF_PWR	RF output power
		7:1	7	RF_CH#	Frequency channel
		0	1	RXEN	RX or TX operation

Table 1 Table of configuration words.

The configuration word is shifted in MSB first on positive CLK1 edges. New configuration is enabled on the falling edge of CS.

### NOTE.

On the falling edge of CS, the TRW-2.4G updates the number of bits actually shifted in during the last configuration.

Ex:

If the TRW-2.4G is to be configured for 2 channel RX in ShockBurst™, a total of 120 bits must be shifted in during the first configuration after VDD is applied.

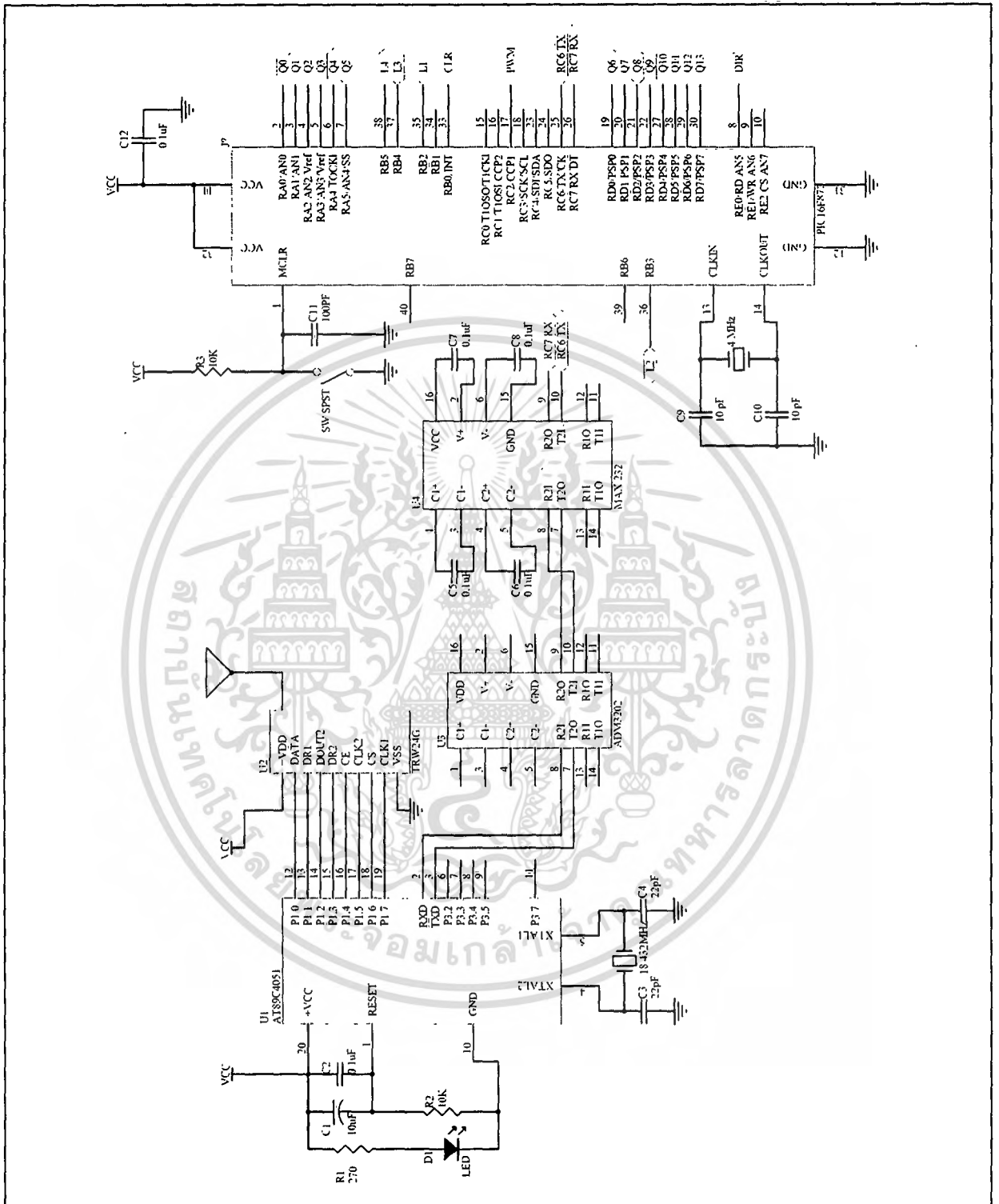
Once the wanted protocol, modus and RF channel are set, only one bit (RXEN) is shifted in to switch between RX and TX.



ภาคผนวก ข  
วงจรถวายงาน

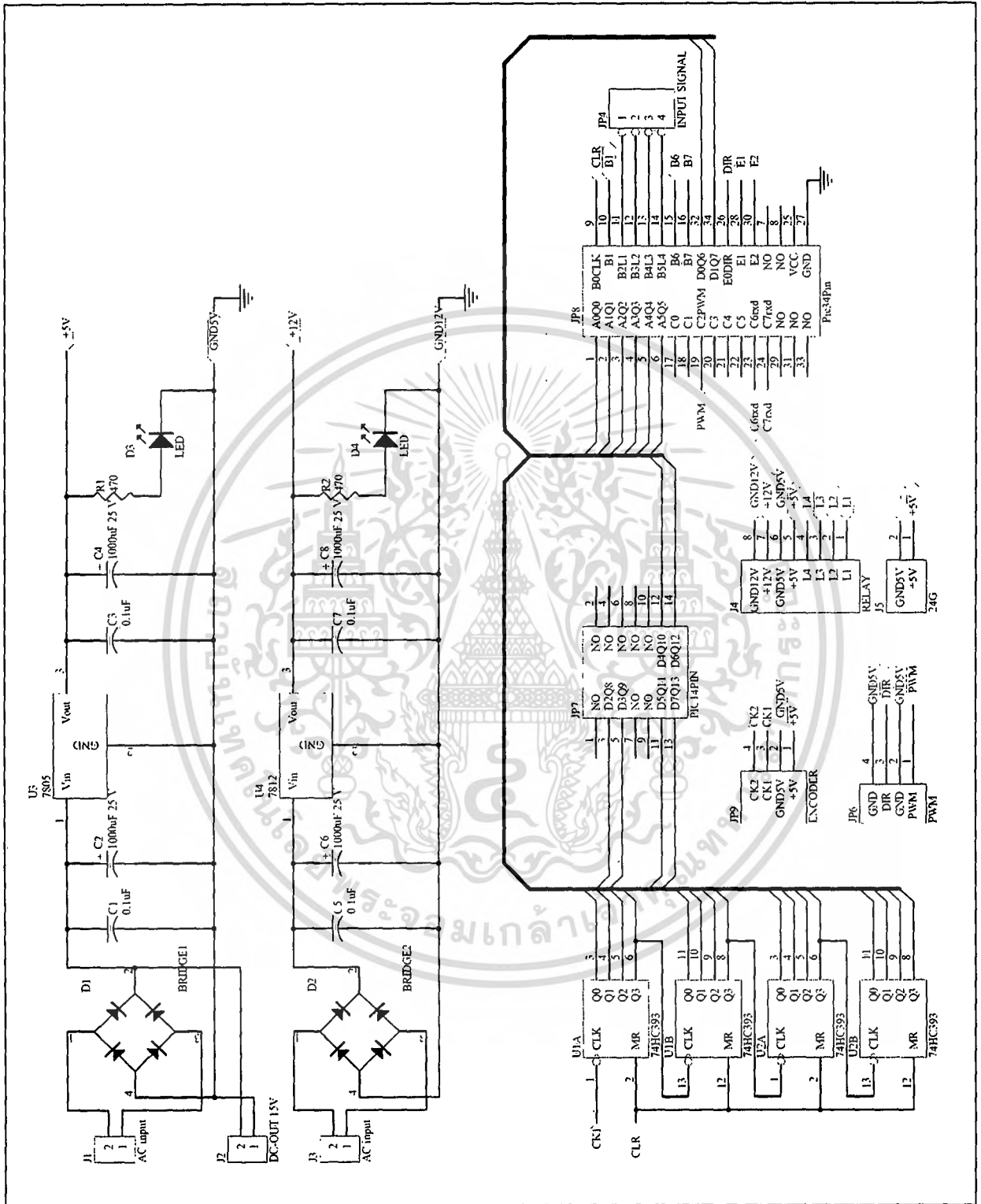
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# วงจรควบคุมกระบวนการทำงาน



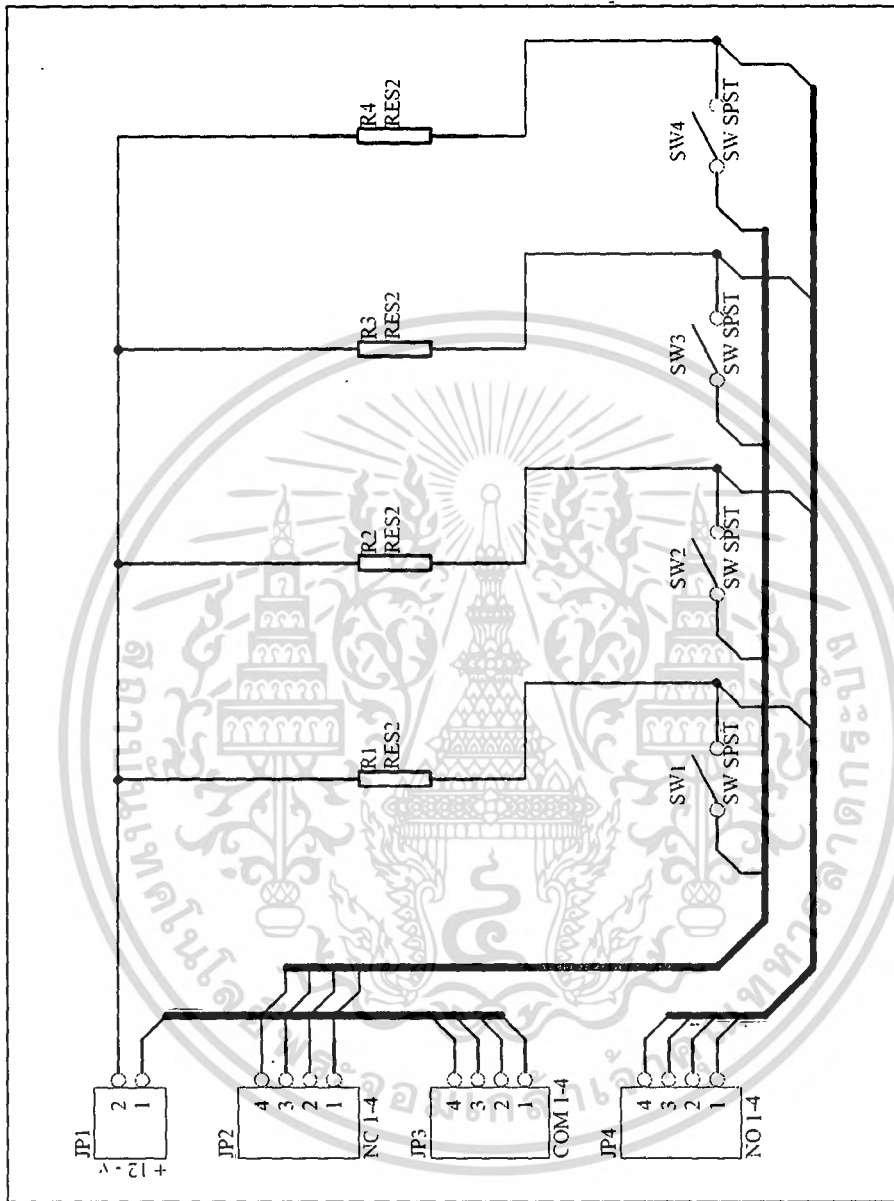
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรวมสายสัญญาณ



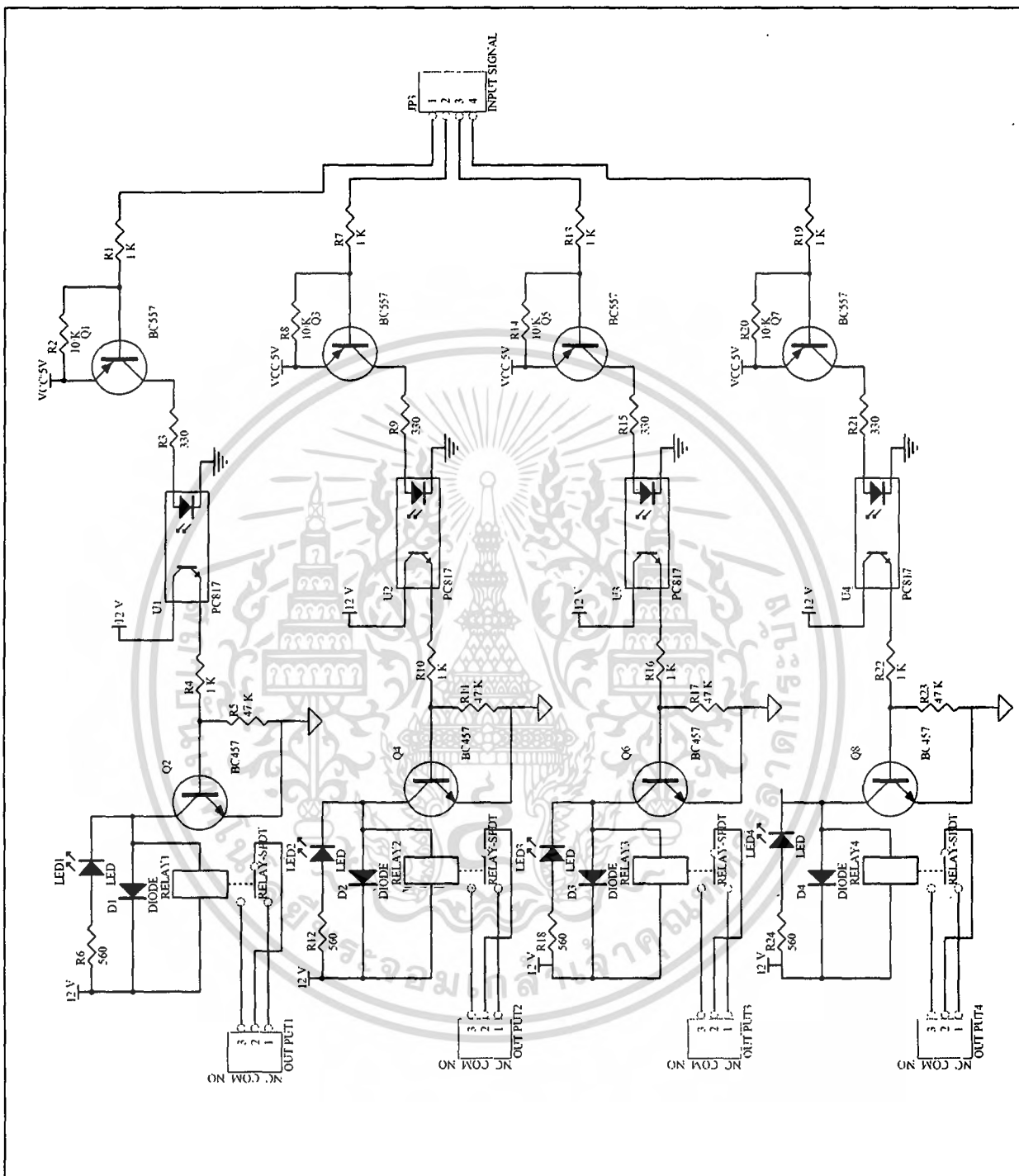
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## วงจรโหลด



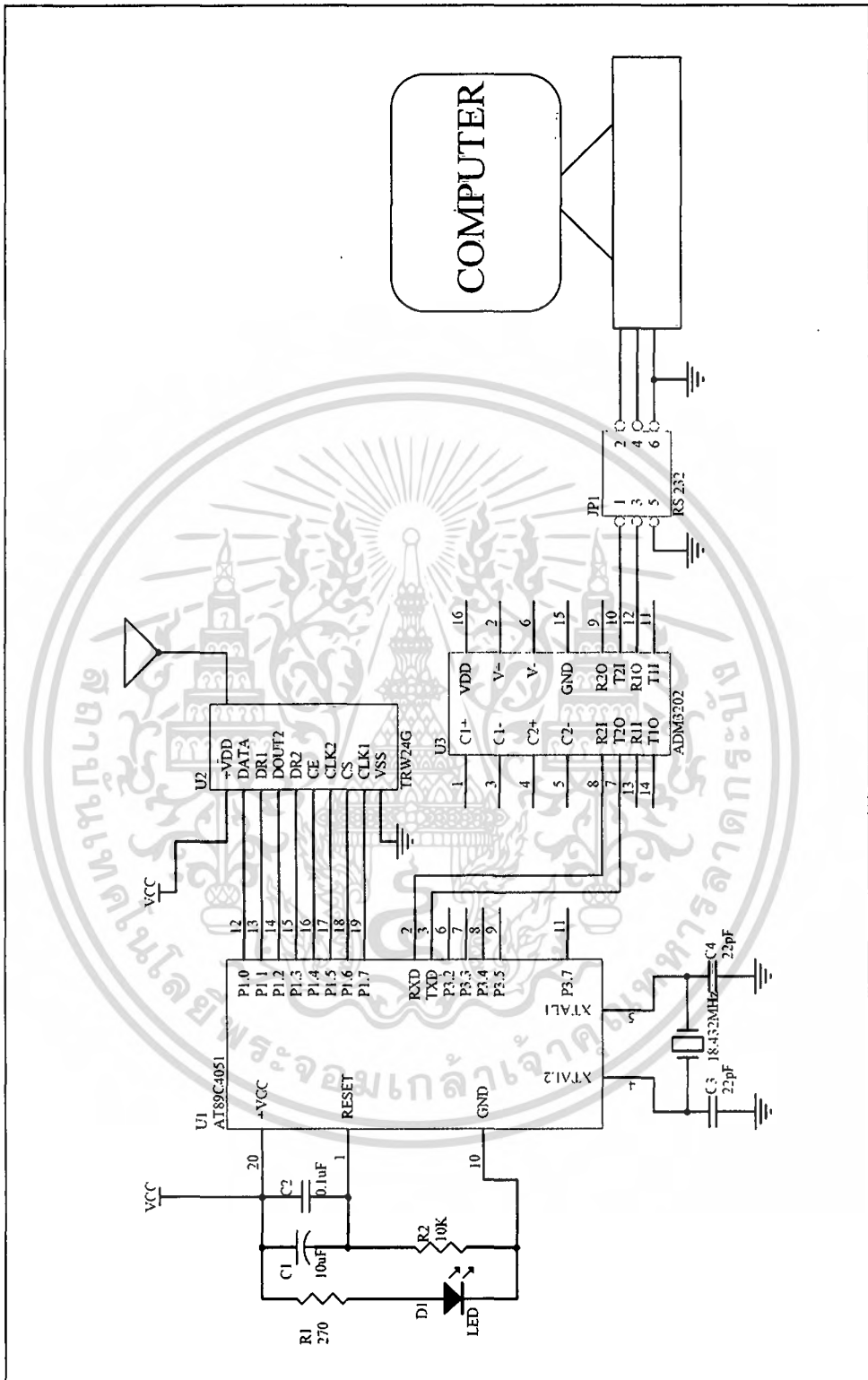
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# วงจรสวิทช์ รีเลย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# วงจรการส่ง รับข้อมูลทางคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้