

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การขับเคลื่อนดีซีมอเตอร์ แบบ 4 -ควอดแดรนต์โดยใช้ วิชาล เบสิก

ควบคุมผ่านทาง ไมโครคอนโทรลเลอร์

4 – Quadrant DC Motor Drive Using Visual Basic Supervised

Microcontroller

นาย ธนะศักดิ์ โล่ห์สุวรรณ

นาย มหัทธ เสมอเหมือน

นาย อนุสรณ์ ก้อนทอง

นาย อธิชา เอื้องสูง

อาจารย์ที่ปรึกษา

ผศ.ดร. สุพัฒน์ กิตติรัตน์สัจจา

ดร. สมภพ ผลไม้

เลขหมู่.....

เลขทะเบียน..... 62569

วัน,เดือน,ปี..... 19 ส.ค. 2549

b. 11626276
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาบัตรปีการศึกษา 2548

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การขับเคลื่อนดีซีมอเตอร์ แบบ 4 -คอร์ตแดรนท์ โดยใช้ วิชาล เบสิก ควบคุม  
ผ่านทางไมโครคอนโทรลเลอร์

ผู้จัดทำ

นาย ชนะศักดิ์ โล่ห์สุวรรณ  
นาย มนัส เสมอเหมือน  
นาย อนุสรณ์ ก้อนทอง  
นาย อัจฉา เสีงสูง



(ผศ.ดร. สุตพัฒน์ กิตติรัตน์สัจจา)

อาจารย์ที่ปรึกษา



(ดร. สมภพ ผลไม้)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขับเคลื่อนดีซีมอเตอร์ แบบ 4 -คอร์ตแดรนท์โดยใช้ วิชาล เบสิก ควบคุมผ่านทาง  
ไมโครคอนโทรลเลอร์

นาย ชนะศักดิ์ โล่ห์สุวรรณ  
นาย มนัส เสมอเหมือน  
นาย อนุสรณ์ ก้อนทอง  
นาย อัจฉา เอื้องสูง

ผศ.ดร. สุพัฒน์ กิตติรัตน์สัจจา อาจารย์ที่ปรึกษา  
ดร. สมภพ ผลไม้ อาจารย์ที่ปรึกษา  
ปีการศึกษา 2548

บทคัดย่อ

ในปัจจุบันการขับเคลื่อนและควบคุมมอเตอร์ไฟฟ้ากระแสตรงมีมากมายหลากหลายวิธี ซึ่งการจะใช้วิธีใดนั้นขึ้นอยู่กับลักษณะของงานที่ทำรวมไปถึงความชำนาญของผู้ควบคุมด้วย เช่นเดียวกันเราปฏิเสธไม่ได้ว่าคอมพิวเตอร์เข้ามามีบทบาทในชีวิตประจำวันของเรามากขึ้น การใช้คอมพิวเตอร์ร่วมในการทำงานรูปแบบต่างๆ รวมไปถึงในงานอุตสาหกรรมด้วย

ปฏิญญาฉบับนี้เป็นการนำเสนอการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงด้วยไมโครคอนโทรลเลอร์ มีวงจรชุดขับที่มี ไอจีบีที เป็นตัวสวิตช์ มีอุปกรณ์ตรวจจับตำแหน่งของมอเตอร์ หน้าที่รับผลที่เกิดจากการขับเคลื่อนมอเตอร์ส่งกลับมาแสดงผลให้เห็น การควบคุมการทำงานจะเป็นแบบลูปปิดโดยใช้ตัวควบคุมแบบ PI ในการควบคุมความเร็ว รวมถึงการแสดงผลของค่าพารามิเตอร์ต่างๆ ผ่านทางหน้าจอดีซีคอมพิวเตอร์ โดยมีหน้าตาโปรแกรม วิชาล เบสิก เป็นตัวควบคุม โดยสามารถที่จะสั่งให้มอเตอร์ทำงานหรือดูผลของการทำงานผ่านทางตัวโปรแกรมนี้ได้

## 4 – Quadrant DC Motor Drive Using Visual Basic Supervised Microcontroller

Tanasak Losuwan

Manut Samermean

Anusorn Konthong

Autcha Haunghoong

Asst. Prof. Dr. Supat Kittiratsatcha Advisor

Dr. Sompob Polmai Advisor

2005

### ABSTRACT

Nowadays, there are lots of methods of driving and supervising DC motor and the most efficient way normally depends on each application and also operator's skills. We cannot deny that computers play an important role in every day lives comparing to the past. Computer is used widely in every field including industrial fields and it tends to be used more and more because the computer's efficiency develops all the time.

This thesis proposes how to drive and supervise a DC motor by using a microcontroller. IGBTs are used for the switching devices. Speed and position of the motor are collected by an incremental encoder and displayed on the computer screen. The feedback control is implemented in Visual Basic Program using PI Controller. Several parameters are shown on the screen including user interface.

## กิตติกรรมประกาศ

ในการทำงานอย่างใดอย่างหนึ่งนั้น การที่จะพยายามทำให้งานสำเร็จออกมาด้วยดีนั้น จะต้องประกอบไปด้วยองค์ประกอบหลาย ๆ ส่วน ซึ่งส่วนที่สำคัญที่สุดเป็นที่รู้จักกันดีก็คือความมุ่งมั่น และความพยายามศึกษาและค้นคว้าของตนเองหรือในกลุ่ม แต่ก็มีอีกสิ่งหนึ่งที่สำคัญไม่แพ้กัน คือความร่วมมือและคำปรึกษาจากบุคคลรอบข้าง ซึ่งหากบุคคลหรือกลุ่มคนใดมีการประสานงานที่ดี หรือได้รับคำแนะนำ ดีชม รวมไปถึงกำลังใจที่ดีจากบุคคลรอบข้างแล้วนั้น งานที่กระทำก็จะประสบความสำเร็จด้วยดีเสมอมา

ดังนั้นปริญญาบัตรฉบับนี้คงจะไม่สมบูรณ์เป็นแน่ หากขาดคำแนะนำที่ดีและข้อคิดดี ๆ ในการทำงานจากบุคคลที่จะกล่าวถึงในต่อไปนี้

อันดับแรกคือท่านอาจารย์ที่ปรึกษาทั้ง 2 ท่าน คือ ผศ.ดร.สุพัฒน์ กิตติรัตนัสัจจา และ ดร.สมภาพ ผลไม้ ที่กรุณาช่วยเหลือและชี้แนะ เอาใจใส่ รวมไปถึงให้คำแนะนำที่ดีเสมอมา

ที่ลืมไม่ได้คือ พี่ ๆ นักศึกษาปริญญาโท ที่น่ารักทุกท่าน โดยเฉพาะพี่เอก พี่แจ๊ค และ พี่เอกวัฒน์ ที่คอยช่วยเหลือ ให้คำปรึกษา และความเป็นกันเองอย่างดีที่สุด ทำให้บรรยากาศในการทำงานเป็นไปอย่างราบรื่น ซึ่งในส่วนนี้รวมไปถึงเพื่อน ๆ คนอื่น ๆ ที่ร่วมทำโครงการอยู่ในห้องเดียวกันที่มีส่วนทำให้บรรยากาศในการทำงานเป็นไปอย่างยอดเยี่ยม

ต่อมาคือกลุ่มเพื่อนและบุคคลอื่น ๆ ที่ทางกลุ่มเคารพและขอความช่วยเหลือ ซึ่งได้รับความร่วมมือเป็นอย่างดีเสมอมา ขอขอบคุณเป็นอย่างสูง

สุดท้ายที่สำคัญที่สุดก็คือ บิดา มารดา และครอบครัว อันเป็นที่เคารพรักของข้าพเจ้าทุกคน ที่คอยให้กำลังใจและสอบถามในการทำงานอยู่เสมอ รวมทั้งการดูแลเอาใจใส่และสนใจในการดำเนินชีวิตของพวกเขาทุกคนในทุก ๆ ด้านของชีวิต ข้าพเจ้าทุกคนขอระลึกถึงในพระคุณอันหาสิ่งตอบแทนมิได้ จึงขอ กราบขอบพระคุณเป็นอย่างสูงที่สุด มา ณ ที่นี้

นายชนะศักดิ์	โล่ห์สุวรรณ
นายมนัส	เสมอเหมือน
นายอนุสรณ์	ก้อนทอง
นายอัฐชา	เสียงสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	II
สารบัญ.....	III
สารบัญรูป.....	VI
สารบัญตาราง.....	XI
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตและข้อกำหนดในการวิจัย.....	2
1.4 ขั้นตอนและวิธีการดำเนินงาน.....	2
1.5 ประโยชน์ที่ได้รับจากการวิจัย.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 มอเตอร์ไฟฟ้ากระแสตรง ( DC Motor).....	4
2.1.1 ประเภทของมอเตอร์ไฟฟ้ากระแสตรง.....	6
2.1.2 มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กคงที่.....	6
2.1.3 มอเตอร์ไฟฟ้ากระแสตรงแม่เหล็กถาวร.....	7
2.2 ไอจีบีที.....	10
2.3 ไมโครคอนโทรลเลอร์.....	13
2.3.1 ไมโครคอนโทรลเลอร์ AT89C51AC2.....	13
2.3.2 Programmable Counter Array (PCA).....	15
2.3.3 รีจิสเตอร์ที่เกี่ยวข้องกับการสร้างรูปคลื่นสี่เหลี่ยมด้วยพีซีเอ.....	16
2.4 โปรแกรม วิซวล เบสิก.....	19
2.5 การสื่อสารผ่านทางพอร์ทอนุกรม.....	21
2.5.1 รูปแบบการสื่อสารแบบอนุกรม.....	21
2.5.2 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม.....	22
2.5.3 การเชื่อมต่อพอร์ทอนุกรมมาตรฐานRS-232.....	24
2.5.4 ไอซี MAX 232.....	25
2.5.5 การสื่อสารผ่าน serial port ของโปรแกรม visual basic.....	26
2.5.6 การสื่อสารผ่าน พอร์ทอนุกรม ของไมโครคอนโทรลเลอร์.....	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## IV

2.6	อุปกรณ์ตรวจจับตำแหน่งของโรเตอร์แบบ incremental.....	36
2.6.1	ความละเอียดของอุปกรณ์ตรวจจับตำแหน่งของโรเตอร์แบบ incremental.....	38
2.6.2	เอาต์พุตของอุปกรณ์ตรวจจับตำแหน่งของโรเตอร์แบบ incremental.....	39
บทที่ 3	หลักการงานและการออกแบบวงจร.....	40
3.1	วงจรควบคุม.....	40
3.1.1	การควบคุมป้อนกลับเบื้องต้น.....	40
3.1.2	คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพี.....	42
3.1.3	คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดไอ.....	43
3.1.4	คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีไอ.....	43
3.1.5	คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีดี.....	45
3.1.6	คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีไอดี.....	47
3.1.7	วงจรเดดไทม์.....	48
3.1.8	วงจรขับไอจีบีที.....	49
3.2	วงจรกำลัง.....	50
3.2.1	DC – DC CONVERTER FULL-BRIDGE.....	50
3.3	การออกแบบในส่วนของโปรแกรม วิชาการ เบสิก.....	54
3.3.1	Form Object.....	54
3.3.2	Frame Object.....	55
3.3.3	Option Object.....	56
3.3.4	Text Box Object.....	57
3.3.5	Label Object.....	57
3.3.6	Command Button Object.....	58
3.3.7	Picture Object.....	59
3.3.8	Timer Object.....	60
บทที่ 4	การทดลอง.....	61
4.1	การทดสอบจับสัญญาณของวงจรเดดไทม์.....	61
4.2	การทดสอบจับสัญญาณที่ออกจากวงจรชุดขับและวงจรกำลัง.....	62
4.3	การทดลองแบบวงรอบปิด.....	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 สรุปและวิจารณ์โครงการ.....	67
5.1 สรุปโครงการ.....	67
5.2 ปัญหา อุปสรรค และการแก้ไข.....	68
5.3 แนวทางการพัฒนา.....	69
ภาคผนวก	
เอกสารอ้างอิง	
กิตติกรรมประกาศ	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
1.1	บล็อกไดอะแกรมแสดงการทำงานและการควบคุมโดยรวม.....2
2.1	มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร.....5
2.2	มอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนาม.....5
2.3	การทำงานของมอเตอร์ไฟฟ้ากระแสตรงทั้ง 4 ย่านการทำงาน.....7
2.4	แม่เหล็กถาวรที่ใช้เป็นสเตเตอร์ในการสร้างสนามแม่เหล็ก.....8
2.5	วงจรสมมูลของมอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร.....8
2.6	คุณลักษณะ ของ แรงบิด-ความเร็วรอบ $V_{t1} > V_{t2} > V_{t3} > V_{t4}$ โดยที่ $V_{T4}$ คือแรงดันไฟฟ้าที่พิกัด.....9
2.7	สมรรถภาพของแรงบิด-ความเร็วแบบต่อเนื่อง.....9
2.8	มอเตอร์แบบแม่เหล็กถาวรที่ใช้ในโรงงาน.....10
2.9	สัญลักษณ์ของไอจีบีที.....11
2.10	ภาพตัดขวางโครงสร้างพื้นฐานของไอจีบีที.....11
2.11	ลักษณะทั่วไปภายนอกของไอจีบีที.....12
2.12	ไอจีบีทีรูปแบบต่างๆ.....12
2.13	ลักษณะภายนอกของ ไมโครคอนโทรลเลอร์ AT89C51AC2.....14
2.14	ลักษณะการทำงานของขาต่างๆ ของ ไมโครคอนโทรลเลอร์ AT89C51AC2....14
2.15	บอร์ดพีซีเอ โมดูล ของ ETT พร้อมอุปกรณ์.....15
2.16	บล็อกการทำงานของ 8 bit pulse width modulator.....18
2.17	หน้าตาพื้นฐานของ วิซวล เบสิก ให้เลือกโหมดการทำงาน.....19
2.18	หน้าตาพร้อมอุปกรณ์และเครื่องมือพื้นฐานของ วิซวล เบสิก.....20
2.19	การส่งข้อมูล ในหนึ่งจังหวะของช่วงเวลาจะส่งข้อมูลออกพร้อมกันทั้ง 8 บิต.....21
2.20	การส่งข้อมูลอนุกรม ในหนึ่งจังหวะของช่วงเวลาจะส่งข้อมูลออก 1 บิต.....22
2.21	แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที.....23
2.22	ส่งข้อมูล 8 บิตพร้อมบิตเริ่มต้น,บิตพาริตี,บิตหยุด ความเร็ว 9600 บิต/วินาที...24
2.23	แรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะ"1"และ "0".....24
2.24	แสดงตำแหน่งขาของไอซี MAX232 และการต่อใช้งาน.....25
2.25	ขั้นตอนการเลือกฟังก์ชัน MSComm มาใช้งาน.....26
2.26	เลือกฟังก์ชัน Microsoft Comm Control 6.0 ใน Components.....27
2.27	ลาก Control ชื่อ Microsoft Comm จาก ToolBox มาไว้บน Form.....27

## VII

2.28	แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0.....	30
2.29	แสดงสัญญาณการส่งข้อมูลในโหมด 1.....	31
2.30	แสดงสัญญาณการส่งข้อมูลในโหมด 2.....	31
2.31	แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register).....	32
2.32	แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล.....	32
2.33	อัตรารับส่งโหมด 0 = ความถี่สัญญาณนาฬิกา ทหาร12.....	32
2.34	รับส่งโหมด 2 = ความถี่สัญญาณนาฬิกา ทหาร 32 หรือทหาร 64.....	33
2.35	การแยกประเภทของเอนโค้ดเดอร์.....	36
2.36	โครงสร้างและสัญญาณของอินทรีเมนท์เอนโค้ดเดอร์.....	37
2.37	ลักษณะชนิดของแผ่นจานหมุนที่แบ่งช่องให้แสงผ่าน.....	38
2.38	จานหมุนของเอนโค้ดเดอร์ที่ใช้ในโครงการนี้ มีความละเอียด 400.....	38
2.39	การติดตั้งจานหมุนและตัวเซ็นเซอร์แสง.....	39
3.1	บล็อกไดอะแกรมการทำงานและการควบคุมโดยรวม.....	40
3.2	การควบคุมแบบลูปเปิด.....	41
3.3	การควบคุมแบบลูปปิด.....	42
3.4	แผนภาพกรอบของตัวควบคุมชนิดพี.....	42
3.5	แผนภาพกรอบของตัวควบคุมชนิดไอ.....	43
3.6	การใช้ตัวควบคุมแบบ PI.....	44
3.7	สัญญาณเข้าฟังก์ชันหนึ่งหน่วย.....	44
3.8	สัญญาณออกของตัวควบคุม.....	44
3.9	บล็อกไดอะแกรมของโครงการที่ใช้แบบ PI.....	45
3.10	การใช้ตัวควบคุมแบบ PD.....	46
3.11	สัญญาณที่เข้าฟังก์ชันแรมปีหนึ่งหน่วย.....	46
3.12	สัญญาณออกของตัวควบคุม.....	46
3.13	การใช้ตัวควบคุมแบบ PID.....	47
3.14	สัญญาณที่เข้าฟังก์ชันแรมปีหนึ่งหน่วย.....	47
3.15	สัญญาณออกของตัวควบคุม.....	48
3.16	วงจรถัดไทม์.....	48
3.17	วงจรถัดขั้วของไอจีบีที.....	49
3.18	การควบคุม 4 คอร์ดแดรนท์.....	50
3.19	ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q1 , Q2 ทำงาน.....	51
3.20	ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q3 , Q4 ทำงาน พร้อม D1 , D2.....	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## VIII

3.21	ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q3 , Q4 ทำงาน .....	52
3.22	ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q1 , Q2 ทำงาน พร้อม D3 , D4.....	53
3.23	แถบเครื่องมือ (Tool Box) และอุปกรณ์พื้นฐานบนหน้าต่างโปรแกรม.....	54
3.24	ฟอร์มที่ปรากฏขึ้นมาเมื่อเริ่มเปิดโปรแกรม.....	55
3.25	การเลือกทิศทางการหมุนของมอเตอร์โดยใช้ Option ร่วมกับ Frame.....	56
3.26	Text Box สำหรับใส่ค่าความเร็วรอบที่ต้องการโดยวางอยู่ใน Frame.....	57
3.27	ปุ่มกด RUN และ STOP โดยใช้ Command Button Object.....	59
3.28	กราฟแสดงค่าความเร็วรอบที่สร้างจาก Picture Object.....	60
4.1	ส่วนช่วงเดดไทม์ที่อยู่ประมาณ 3 $\mu$ S.....	61
4.2	สัญญาณที่ออกจากชุดขับและช่วงเดดไทม์.....	62
4.3	การจับสัญญาณที่ออกจากชุดวงจรกำลังที่พร้อมไปขับมอเตอร์.....	63
4.4	กราฟแสดงผลตอบซ้ำครู่ของความเร็รรอบมอเตอร์ เมื่อตั้งค่าความเร็วที่ 300 rpm.....	64
4.5	กราฟความเร็วรอบของมอเตอร์เมื่อเปลี่ยนค่าความเร็วจาก 150 rpm ไปยัง 320 rpm.....	64
4.6	กราฟความเร็วรอบของมอเตอร์เมื่อเปลี่ยนค่าความเร็วจาก 300 rpm ไปยัง 150 rpm.....	65
4.7	กราฟความเร็วรอบที่ลดลงสู่ศูนย์.....	66
4.8	ลักษณะของกระแสเทียบกับแรงดันขณะปรับความเร็วมอเตอร์.....	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1	อัตรารับส่งข้อมูลที่กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้บอตเรทมาตรฐาน.....35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญของปัญหา

ปัจจุบันในภาคของงานอุตสาหกรรมนั้น มอเตอร์ไฟฟ้าเป็นส่วนประกอบที่สำคัญเป็นอย่างยิ่ง โดยทั่วไปประเภทของมอเตอร์ที่ใช้ในงานอุตสาหกรรมทั่วไปนั้น มีอยู่หลัก ๆ 2 ประเภท คือ มอเตอร์กระแสตรงและมอเตอร์กระแสสลับ ซึ่งการนำไปใช้งานก็จะแตกต่างกันไป ซึ่งงานแต่ละงานและมอเตอร์แต่ละประเภทก็มีการควบคุม ที่แตกต่างกันไป

เทคนิคการควบคุมแบบ DC – DC CONVERTER FULL- BRIDGE ซึ่งใช้สวิตช์เป็น ไอจีบีที 4 ตัว เป็นอีกวิธีหนึ่งที่นิยมใช้กัน โดยให้สวิตช์ไอจีบีทีทำงานเป็นคู่ มีโหมดการทำงานเป็น 4 คอร์ดแตรนต์ ซึ่งสามารถที่จะสั่งหรือกำหนดให้มอเตอร์หมุนไปในทิศทางที่ต้องการ รวมไปถึงการเบรกและเพิ่ม – ลด ความเร็วรอบตามความต้องการหรือตามขนาดของภาระไฟฟ้าหรือภาระทางกลที่ต่ออยู่กับมอเตอร์ได้

การควบคุมมอเตอร์ไฟฟ้ากระแสตรง โดยให้ไมโครคอนโทรลเลอร์เป็นตัวประมวลคำสั่ง นั้นก็เป็นแนวทางที่ใช้นกันมาก เพราะว่าส่วนของไมโครคอนโทรลเลอร์มีประสิทธิภาพในการทำงานค่อนข้างสูง มีขนาดเล็กกะทัดรัด มีระบบการรับ – ส่งข้อมูลค่อนข้างรวดเร็ว การใช้ไมโครคอนโทรลเลอร์เข้ามาร่วมกับชุดขับที่เป็นวงจรถูกบูรณาการ ซึ่งใช้ไอจีบีทีเป็นสวิตช์ นั้นจะช่วยให้การควบคุมมอเตอร์ไฟฟ้ากระแสตรง ทำได้สะดวกมากขึ้น

โปรแกรม วิชาล เบสิก (visual basic) เป็นอีกโปรแกรมหนึ่งในปัจจุบันที่มีการใช้นกันอย่างแพร่หลาย แต่โดยทั่วไปจะพบว่าในการใช้โปรแกรมมาาร่วมกับการควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ยังมีไม่มากนัก ทั้งที่การนำมาใช้ร่วมกันนั้น จะทำให้เกิดความสะดวกในการควบคุมและขับเคลื่อนมอเตอร์มากขึ้นไปอีก

ในระบบการตรวจสอบหรือการตรวจวัดตำแหน่งในงานที่ใช้มอเตอร์เป็นตัวทำงานนั้น การใช้เอนโค้ดเดอร์ เซ็นเซอร์ เป็นตัวตรวจจับเป็นที่นิยมใช้นกันมากในภาคอุตสาหกรรม ซึ่งชนิดของมันนั้นมีหลากหลายชนิด แต่ชนิดที่เหมาะสมกับโครงการนี้จะเป็นแบบตรวจสอบด้วย เซ็นเซอร์ และจานหมุน ซึ่งจากการตรวจสอบจะทำให้เราทราบว่าค่าต่าง ๆ ที่มอเตอร์ทำงานอยู่มีลักษณะอย่างไร ทำให้เรารู้ถึงการทำงานและประสิทธิภาพ ความเหมาะสมในการทำงานนั้น ๆ ได้

ดังนั้นในโครงการนี้จึงนำส่วนหลัก ๆ คือ มอเตอร์ไฟฟ้ากระแสตรง วงจรถูกขับ ไมโครคอนโทรลเลอร์ เซ็นเซอร์ เอนโค้ดเดอร์ และโปรแกรม วิชาล เบสิก ทั้ง 5 ส่วน มาประยุกต์ใช้ร่วมกันโดยการเชื่อมต่อและสื่อสารผ่านทาง พอร์ทอนุกรม (serial port) ของคอมพิวเตอร์ ด้วยมาตรฐาน RS-232 เพื่อที่จะเพิ่มประสิทธิภาพในการขับเคลื่อนและควบคุมมอเตอร์ไฟฟ้ากระแสตรงให้มากขึ้น

## 1.2 วัตถุประสงค์ของการวิจัย

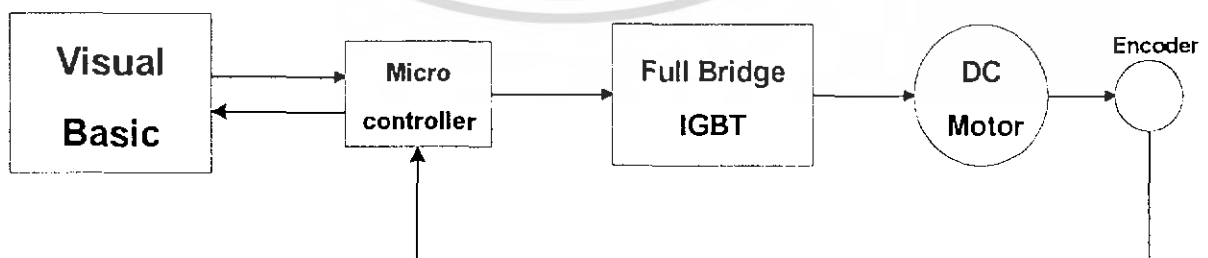
1. เพื่อศึกษาระบบการเชื่อมต่อ รับ-ส่ง ควบคุม และแสดงผลข้อมูล ระหว่างคอมพิวเตอร์กับชิ้นงานโดยผ่านทางพอร์ทอนุกรม (serial port)
2. เพื่อศึกษาระบบควบคุมแบบมีการป้อนกลับด้วยเอ็นโค้ดเดอร์ โดยมีไมโครคอนโทรลเลอร์เป็นตัวควบคุมและประมวลผลรับ – ส่งค่าไปแสดงที่คอมพิวเตอร์
3. เพื่อศึกษาการใช้งานโปรแกรม วิชาล เบสิก (visual basic) เพื่อใช้ในการควบคุมอุปกรณ์ไฟฟ้า
4. เพื่อศึกษาการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบ 4 คอร์ดแดรนจ์ โดยใช้วงจร DC – DC CONVERTER FULLBRIDGE ซึ่งใช้ไอจีบีที เป็นสวิตช์

## 1.3 ขอบเขตและข้อกำหนดในการวิจัย

ขอบเขตการทำงานคือการศึกษาการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง โดยใช้ชุดขับที่เป็น DC – DC CONVERTER FULL-BRIDGE ซึ่งใช้ไอจีบีที 4 ตัวเป็นตัวสวิตช์ ควบคุม รับ-ส่งค่าด้วยไมโครคอนโทรลเลอร์กำหนดทิศทางหมุน ความเร็วรอบ ด้วยการป้อนข้อมูลทางโปรแกรม วิชาล เบสิก และแสดงผลที่ได้ด้วยการป้อนกลับโดยใช้เอ็นโค้ดเดอร์ ป้อนกลับมาทางไมโครคอนโทรลเลอร์และไปแสดงผลที่ตัวโปรแกรม วิชาล เบสิก อีกครั้ง ด้วยการสื่อสารผ่านทาง พอร์ทอนุกรม (serial port)

## 1.4 ขั้นตอนและวิธีการดำเนินงาน

1. เลือกหัวข้อในการทำโครงการงาน
2. ศึกษาวัตถุประสงค์และขอบเขตการทำงานของโครงการงาน
3. ศึกษาและค้นคว้าข้อมูลที่เกี่ยวข้องกับโครงการงาน
4. ออกแบบและสร้างวงจรทดลองเพื่อศึกษาการทำงาน
5. ศึกษาผลการทดลองที่ได้และนำผลการทดลองมาทำการวิเคราะห์
6. สรุปผลการทดลอง รวมถึงปัญหาและอุปสรรคในการทำงาน



รูปที่ 1.1 บล็อกไดอะแกรมแสดงการทำงานและการควบคุมโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.5 ประโยชน์ที่ได้รับจากการวิจัย

1. เข้าใจถึงหลักการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบ DC – DC CONVERTER FULL-BRIDGE โดยใช้ไอจีบีทีเป็นสวิตช์
2. ได้รับความรู้ในเรื่องการเชื่อมต่อของไมโครคอนโทรลเลอร์กับโปรแกรมคอมพิวเตอร์โดยผ่านทางพอร์ทอนุกรม (serial port)
3. ทำให้รู้หลักการทำงานของระบบป้อนกลับโดยเอ็นโค้ดเดอร์และการเชื่อมต่อกับไมโครคอนโทรลเลอร์และการแสดงผลที่คอมพิวเตอร์
4. ได้ศึกษาและเรียนรู้เกี่ยวกับโปรแกรม วิชาล เบสิก (visual basic)
5. สามารถนำประสบการณ์และความรู้ที่ได้จากโครงการไปประยุกต์ใช้ได้ในงานจริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

### 2.1 มอเตอร์ไฟฟ้ากระแสตรง ( DC Motor)

เครื่องจักรกลไฟฟ้ากระแสตรง เป็นเครื่องจักรที่ทำหน้าที่แปลงรูปพลังงานไฟฟ้า เป็นพลังงาน ทางกล โดยที่เครื่องจักรทำงานโดยการหมุนตัวของการหมุน (Rotor) เมื่อเครื่องจักรแปลงรูป พลังงานกลเป็นพลังงานไฟฟ้ากระแสตรงเรียกว่า เครื่องกำเนิดไฟฟ้ากระแสตรง (DC Ganeator ) และถ้าเครื่องจักรกลแปลงรูปพลังงานไฟฟ้ากระแสตรง ไปเป็นพลังงานทางกล เรียกว่า มอเตอร์ไฟฟ้ากระแสตรง (DC motor)

มอเตอร์ไฟฟ้ากระแสตรงจะมีแรงบิดในขณะหมุนสูง เพราะตำแหน่งที่มุมระหว่าง สนามแม่เหล็กจากขั้วแม่เหล็กกับสนามแม่เหล็กจากอาร์เมเจอร์ทำมุมเท่ากับ 90 องศาทางไฟฟ้า ซึ่งมอเตอร์ ไฟฟ้ากระแสสลับทั่วไปไม่สามารถทำได้ในตำแหน่งนี้

ในเครื่องจักรไฟฟ้ากระแสตรงมีซีคอมมิวเตเตอร์ (Commutator) แปรงถ่าน (Carbon Brush) ทำหน้าที่จัดเรียงกระแสจากไฟฟ้าสลับ ในขดลวดอาร์เมเจอร์ (Armature) ให้เป็นไฟฟ้า กระแสตรง ออกจากตัวเครื่องจึงเป็นแหล่งจ่ายไฟฟ้ากระแสตรง แต่เมื่อคัทตาไฟฟ้าที่ขั้วมีค่าสูงกว่าคัทตาไฟฟ้าที่เกิดขึ้นภายใน จะทำให้กระแสจากภายนอกเข้าตัวเครื่องจักร ซึ่งซีคอมมิวเตเตอร์จะทำหน้าที่จัดเรียงกระแสที่อยู่ภายใต้ขั้วแม่ที่แตกต่างกัน ก็จะมีกระแสไหลในทิศทางตรงข้ามกันจะทำให้เกิดแรงบิดเสริมกัน จะเห็นได้ว่าในเครื่องจักรไฟฟ้ากระแสตรงตัวหนึ่งนั้นสามารถเป็นได้ทั้ง มอเตอร์และเครื่องกำเนิดไฟฟ้า

#### 2.1.1 ประเภทของมอเตอร์ไฟฟ้ากระแสตรง

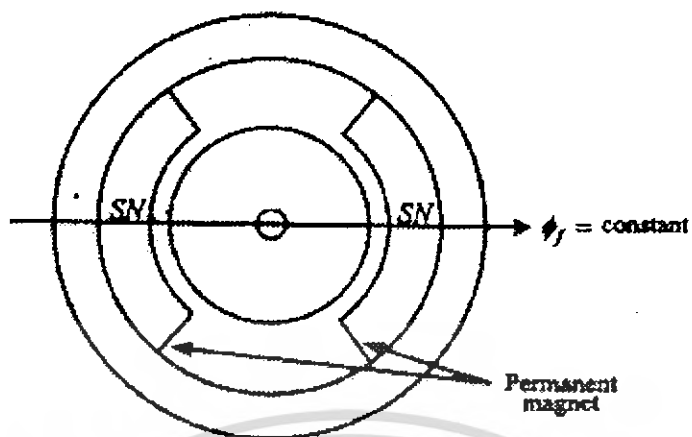
มอเตอร์กระแสตรงสามารถแบ่งออกได้หลายประเภท ขึ้นอยู่กับลักษณะการสร้างสนามแม่เหล็กของตัวมอเตอร์ และขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาร์เมเจอร์

การแบ่งประเภทตามการจ่ายสนามแม่เหล็กแยกออกได้เป็น 2 แบบคือ

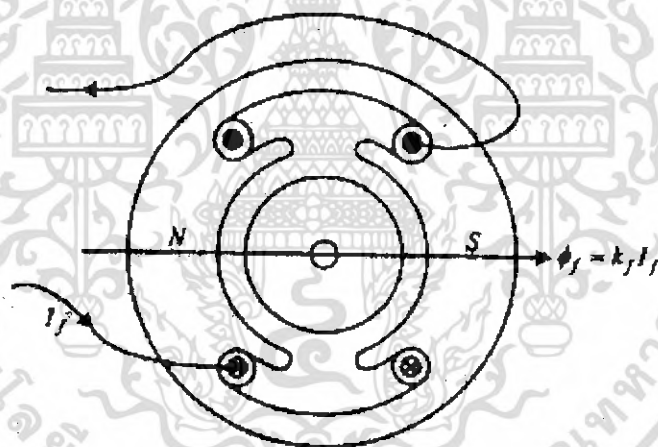
1. มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กคงที่
2. มอเตอร์ไฟฟ้ากระแสตรงแบบปรับเส้นแรงแม่เหล็กได้

#### 2.1.2 มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กคงที่

มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กคงที่ หรือแม่เหล็กถาวร (Permanent Magnet) ดังรูปที่ 2.1 ซึ่งค่าเส้นแรงแม่เหล็กที่เกิดจากแม่เหล็กถาวรจะมีค่าคงที่สม่ำเสมอหรือจากขดลวดสนาม (fieldwinding) ดังรูปที่ 2.2



รูปที่ 2.1 มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร



รูปที่ 2.2 มอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนาม

กระแสฟีลด์ (Field current,  $I_f$ ) จะเป็นตัวควบคุมเส้นแรงแม่เหล็ก ถ้ากำหนดให้สเตเตอร์ที่ทำหน้าที่ในการสร้างเส้นแรงแม่เหล็กไม่เกิดการอิ่มตัวจะได้

$$\phi_f = k_f \cdot I_f \quad (2.1)$$

โดยที่  $k_f$  หมายถึง ค่าคงที่ของเส้นแรงแม่เหล็ก

$\phi_f$  หมายถึง ค่าฟลักซ์ แม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโรเตอร์ (Rotor) จะมีขดลวดพันรอบตัวโรเตอร์ซึ่งขดลวดดังกล่าว เรียกว่า ขดลวดอาร์เมเจอร์ (Armature Winding) ซึ่งจะทำหน้าที่ในการรับพลังงานไฟฟ้าเข้าสู่มอเตอร์ ไฟฟ้ากระแสตรงโดยผ่านขดลวดอาร์เมเจอร์

แรงบิดของมอเตอร์ไฟฟ้ากระแสตรงจะเกิดจากความสัมพันธ์ระหว่าง เส้นแรงแม่เหล็ก และกระแสอาร์เมเจอร์ ( $i_a$ ) ดังสมการที่ 2.2

$$T_{em} = k_t \cdot \phi_r \cdot i_a \quad (2.2)$$

โดยที่  $k_t$  หมายถึงค่าคงที่แรงบิดของมอเตอร์ไฟฟ้ากระแสตรง

ค่าแรงเคลื่อนไฟฟ้าย้อนกลับ (back - emf) ในส่วนของ อาร์เมเจอร์จะเกิดจากการ หมุนตัวนำของอาร์เมเจอร์ด้วยความเร็ว  $\omega_m$  ตัดผ่านเส้นแรงแม่เหล็ก  $\phi_r$

$$e_a = k_e \cdot \phi_r \cdot \omega_m \quad (2.3)$$

โดยที่  $k_e$  หมายถึงค่าคงที่ของแรงดันไฟฟ้ากระแสตรง

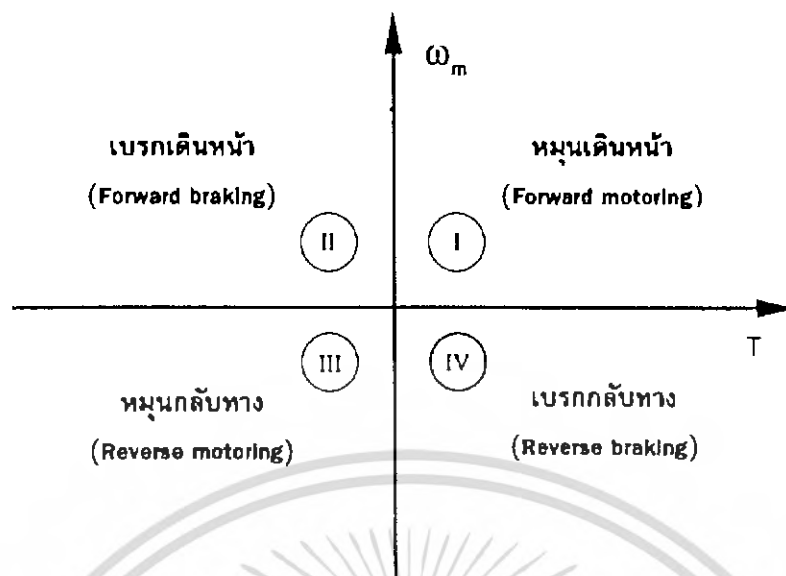
อย่างไรก็ตาม ในหน่วย SI ค่า  $k_t$  และ  $k_e$  จะมีค่าเท่ากันซึ่งสามารถพิสูจน์ได้จากสมการสมดุลพลังงาน โดยกำลังไฟฟ้า ( $e_a \cdot i_a$ ) เท่ากับกำลังไฟฟ้า ทางกล ( $\omega_m \cdot T_{em}$ ) จะได้

$$P_e = e_a \cdot i_a = k_e \cdot \phi_r \cdot \omega_m \cdot i_a \quad (2.4)$$

และกำลังไฟฟ้าทางกลเท่ากับ

$$P_m = \omega_m \cdot T_{em} = \omega_m \cdot k_t \cdot \phi_r \cdot i_a \quad (2.5)$$

เครื่องจักรกลไฟฟ้ากระแสตรงนั้น สามารถทำงานหรือเป็นเครื่องกำเนิดไฟฟ้ากระแสตรงได้ แต่โดยทั่วไปจะไม่นิยมนำมาใช้งานเป็นเครื่องกำเนิดไฟฟ้า แต่อย่างไรก็ตามในขณะทำการเบรกมอเตอร์ไฟฟ้ากระแสตรงนั้น ก็จะมีลักษณะเสมือนเป็นเครื่องกำเนิดไฟฟ้ากระแสตรง ในขณะที่ความเร็วลดลง ดังนั้นขณะที่มอเตอร์เบรกจำเป็นต้องพิจารณาเป็นเครื่องกำเนิดไฟฟ้า โดยกำหนดให้ค่าเส้นแรงแม่เหล็กคงที่และมอเตอร์กำลังขับโหลดคงที่ที่มีความเร็วเท่ากับ  $\omega_m$  และเพื่อที่จะลดความเร็วมอเตอร์เพื่อที่จะทำให้ค่าแรงดันไฟฟ้า  $V_t$  มีค่าต่ำกว่าค่าแรงเคลื่อนเหนี่ยวนำย้อนกลับ  $e_a$  จะส่งผลให้  $i_a$  ไหลกลับทิศทาง ค่าแรงบิด  $T_{em}$  จะกลับทางเช่นกัน นอกจากนั้นพลังงานจลน์ที่เกิดจากแรงเฉื่อยของโหลดก็จะแปลงเป็นพลังงานไฟฟ้า โดยพลังงานที่ได้จะจ่ายไปแหล่งจ่ายแรงดันไฟฟ้า  $V_t$  หรืออยู่ในรูปของความร้อนโดยใช้ตัวต้านทาน



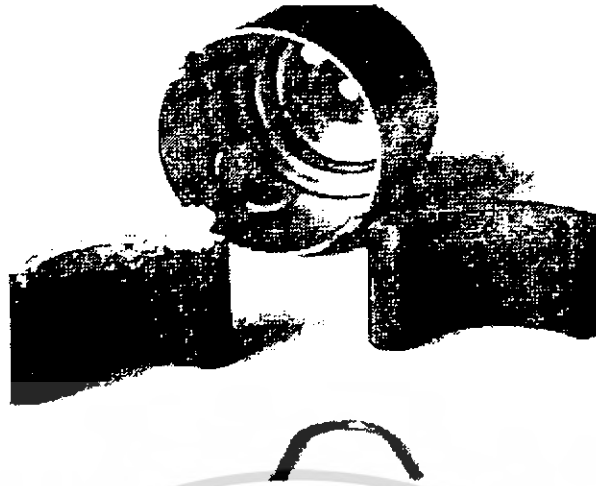
รูปที่ 2.3 การทำงานของมอเตอร์ไฟฟ้ากระแสตรงทั้ง 4 ย่านการทำงาน

ขณะที่มอเตอร์ทำการเบรคนั้น ขั้วของแรงเคลื่อนเหนี่ยวนำย้อนกลับ  $e_a$  ไม่เปลี่ยนแปลง เพราะทิศทางการหมุนยังคงเหมือนเดิม ยังคงใช้สมการเดิมได้ ในการหาค่าแรงเคลื่อนไฟฟ้าเหนี่ยวนำย้อนกลับได้ และเมื่อของโรเตอร์ลดลง เป็นผลให้แรงเคลื่อนเหนี่ยวนำย้อนกลับลดลง ( $\phi_f$  มีค่าคงที่) จนกระทั่งมอเตอร์หยุดหมุนและถ้าขั้วของแหล่งจ่ายเปลี่ยนไปก็จะมีผลทำให้ทิศทางการหมุนของมอเตอร์เปลี่ยนไป

ดังนั้นมอเตอร์ไฟฟ้ากระแสตรงจะสามารถทำงานได้ในทิศทางใดทิศทางหนึ่ง และเมื่อทำการเบรก แรงบิดของตัวมอเตอร์ไฟฟ้ากระแสตรงก็สามารถกลับทิศทางได้ จากรูปที่ 2.3 จะแสดงความสัมพันธ์ระหว่างแรงบิดกับความเร็วรอบสำหรับย่านการทำงานของมอเตอร์ไฟฟ้ากระแสตรงทั้ง 4 ย่านการทำงาน

### 2.1.3 มอเตอร์ไฟฟ้ากระแสตรงแม่เหล็กถาวร ( Permanent Magnet DC motor )

มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร นิยมใช้เป็นมอเตอร์กระแสตรงขนาดเล็ก โดยที่สเตเตอร์จะใช้แม่เหล็กถาวรในการสร้างสนามแม่เหล็ก ซึ่งจะทำให้เส้นแรงแม่เหล็กคงที่ ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 แม่เหล็กถาวรที่ใช้เป็นสเตเตอร์ในการสร้างสนามแม่เหล็ก

สำหรับสภาวะอยู่ตัวกำหนดให้ค่าเส้นแรงแม่เหล็ก มีค่าคงที่ พิจารณาวงจรสมมูลของมอเตอร์ไฟฟ้า กระแสตรงและสมการ

$$T_{em} = k_T \cdot I_a \quad (2.6)$$

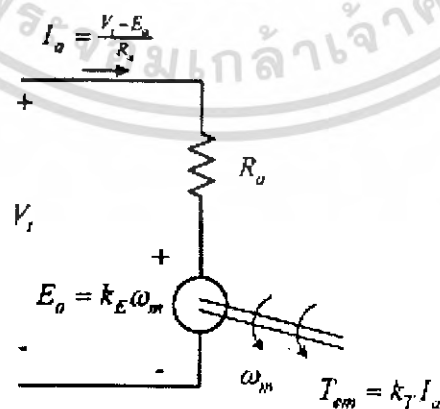
$$E_a = k_E \cdot \omega_m \quad (2.7)$$

$$V_t = E_a + R_a \cdot I_a \quad (2.8)$$

โดยที่  $k_T = k_t \cdot \phi_f$  และ  $k_E = k_e \cdot \phi_f$

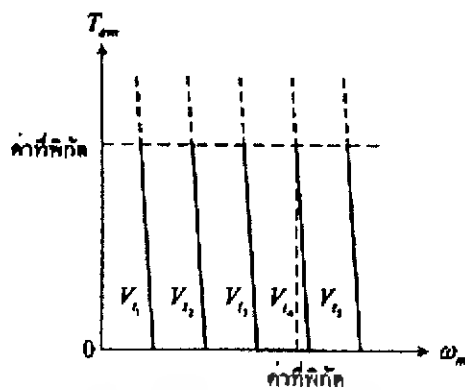
และจากสมการที่ 2.6 – 2.8 จะสามารถหาค่าความเร็วรอบ  $\omega_m$  ที่สภาวะอยู่ตัวเมื่อทราบค่าแรงดัน  $V_t$  และเป็นฟังก์ชันของ  $T_{em}$  จะได้สมการที่ 2.9

$$\omega_m = 1/k_E [ V_t - (R_a/k_T) \cdot T_{em} ] \quad (2.9)$$

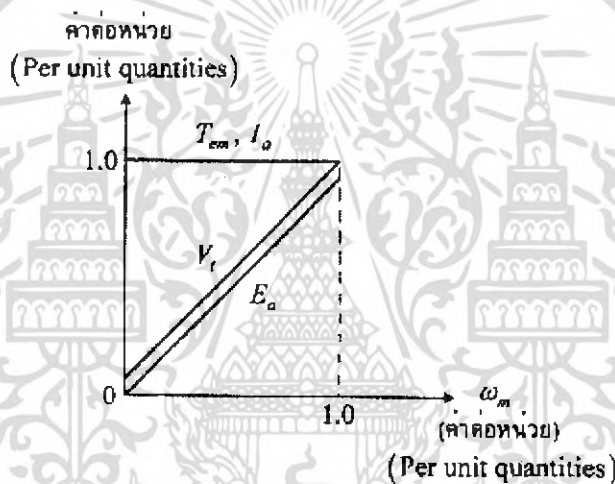


รูปที่ 2.5 วงจรสมมูลของมอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

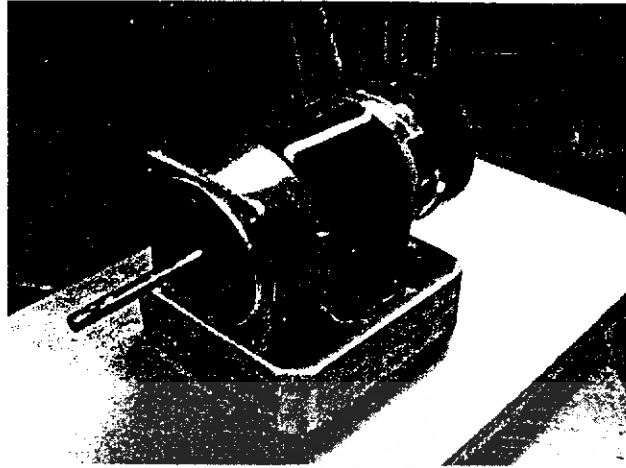


รูปที่ 2.6 คุณสมบัติของ แรงบิด-ความเร็วรอบ  $V_{11} > V_{12} > V_{13} > V_{14}$   
โดยที่  $V_{T4}$  คือแรงดันไฟฟ้าที่พิกัด



รูปที่ 2.7 สมรรถนะของแรงบิด-ความเร็วแบบต่อเนื่อง

สมการที่นำมาวาดกราฟได้ดังรูปที่ 2.6 ซึ่งจะเห็นได้ว่าขณะที่แรงบิดมีค่าเพิ่มขึ้น คุณสมบัติของแรงบิด-ความเร็วรอบที่ค่าความเร็วรอบที่ค่าแรงดันคงที่  $V_r$  ที่กำหนดจะมีรูปร่างเกือบเป็นเส้นตรงขนานกับแรงบิด  $T_{em}$  โดยความแตกต่างเพียงเล็กน้อยที่เกิดขึ้น จะเกิดจากแรงดันไฟฟ้าดกคร่อมค่าความต้านของอาร์เมเจอร์ในมอเตอร์ไฟฟ้ากระแสตรง  $I_a \cdot R_a$  นอกจากนั้นคุณสมบัติของ แรงบิด- ความเร็วสามารถเลื่อนไปมาได้ในแนวแกนนอนได้โดยการควบคุมค่าแรงดันไฟฟ้า  $V_r$  ดังนั้น ความเร็วรอบของโหลดที่คุณสมบัติของแรงบิดต่อความเร็วรอบใดๆ โดยจะสามารถควบคุมค่าแรงดันไฟฟ้า ที่ป้อนให้แก่มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร

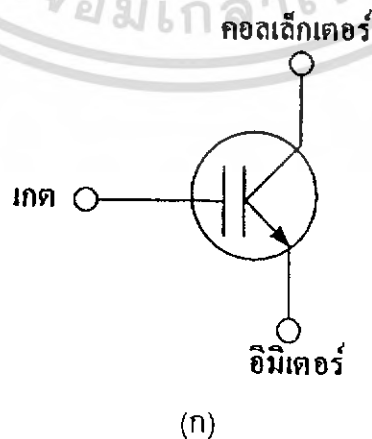


รูปที่ 2.8 มอเตอร์แบบแม่เหล็กถาวรที่ใช้ในโรงงาน

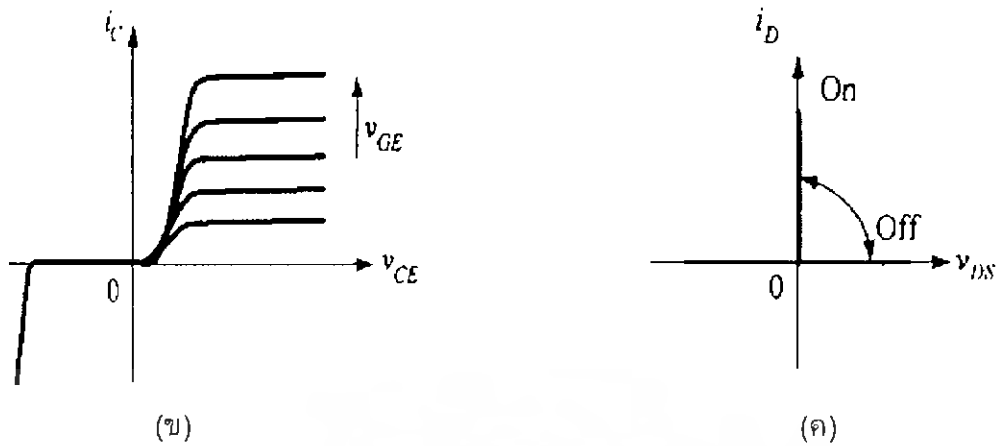
อย่างไรก็ตาม ค่ากระแสอาร์เมเจอร์และแรงบิดไม่ควรสูงเกินกว่าค่ากระแสและแรงบิดที่พิกัดค่าที่เกิดกว่าค่าที่ได้แสดงไว้ด้วยเส้นประ ดังแสดงในรูปที่ 2.6 จะเห็นได้ว่าเมื่อทำการเพิ่มความเร็วของมอเตอร์ไฟฟ้าให้สูงกว่าค่าที่พิกัด จะเป็นผลให้ต้องเพิ่มพิกัดของมอเตอร์ไฟฟ้าสูงกว่าค่าที่พิกัดจะเป็นผลให้ต้องเพิ่มค่าแรงดันไฟฟ้า  $V_s$  สูงซึ่งกว่าค่าที่พิกัดของมอเตอร์ไฟฟ้า

## 2.2 ไอจีบีที (IGBT: Insulated Gate Bipolar Transistor)

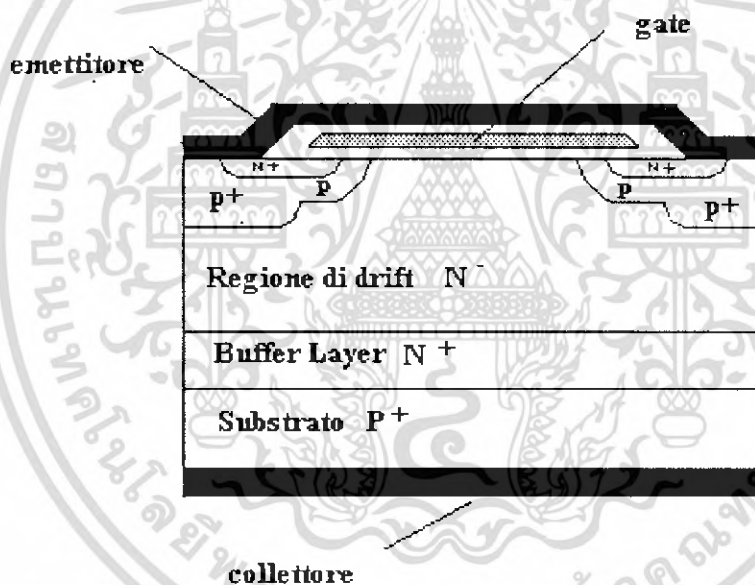
ไอจีบีทีจัดเป็นอุปกรณ์อิเล็กทรอนิกส์กำลังที่สามารถควบคุมให้นำกระแส และควบคุมให้หยุดนำกระแสได้ โดยการควบคุมจากแรงดันไฟฟ้าที่ขาเกต ไอจีบีทีมีสัญลักษณ์ รูปที่ 2.9 (ก) และคุณลักษณะของกระแสและแรงดันไฟฟ้าขณะสภาวะอยู่ตัว รูปที่ 2.9 (ข) คุณลักษณะของกระแสและแรงดันไฟฟ้าในอุดมคติ รูปที่ 2.9 (ค) แสดงในรูปที่ 2.9 ตามลำดับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 (ก) สัญลักษณ์ของไอจีบีที (ข) คุณลักษณะของกระแสและแรงดันไฟฟ้าขณะสภาวะอยู่ตัว (ค) คุณลักษณะของกระแสและแรงดันไฟฟ้าในอุดมคติ



รูปที่ 2.10 ภาพตัดขวางโครงสร้างพื้นฐานของไอจีบีที

โครงสร้างของไอจีบีที แสดงเป็นภาพตัดขวางดังรูปที่ 2.10 โครงสร้างรวมส่วนใหญ่จะเป็นไปในลักษณะของมอสเฟต แต่จะแตกต่างกันตรงที่ไอจีบีทีนั้นจะมีชั้น P<sup>-</sup> หรือชั้น อินเจ็คติง (Injecting) ต่ออยู่ระหว่างขาเตรน ซึ่งจะไม่มีในมอสเฟต ซึ่งการมีชั้นเพิ่มขึ้นจะทำให้ความต้านทานอินพุทที่ขาเกตมีค่าสูง โดยทั่วไปประมาณ 10<sup>9</sup> โอห์ม

ดังนั้นจะเห็นได้ว่าไอจีบีทีได้รวมข้อดีของมอสเฟตกำลัง ทรานซิสเตอร์กำลัง และจีทีไอไทรสเตอร์ มารวมอยู่ด้วยกัน คือ

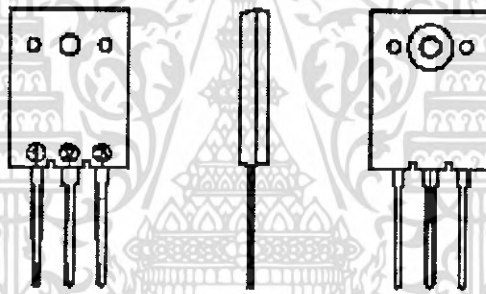
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอจีบีทีจะมีค่าอิมพีแดนซ์ขาเกตที่มีค่าสูงคล้ายมอสเฟตกำลังตั้งที่กล่าวมาข้างต้น ซึ่งต้องการพลังงานน้อยมากในการควบคุมการสวิตช์ ควบคุมการนำและหยุดกระแสด้วยแรงดันไฟฟ้าที่ขาเกต รวมทั้งสามารถใช้งานการสวิตช์ที่ความถี่สูงได้อีกด้วย

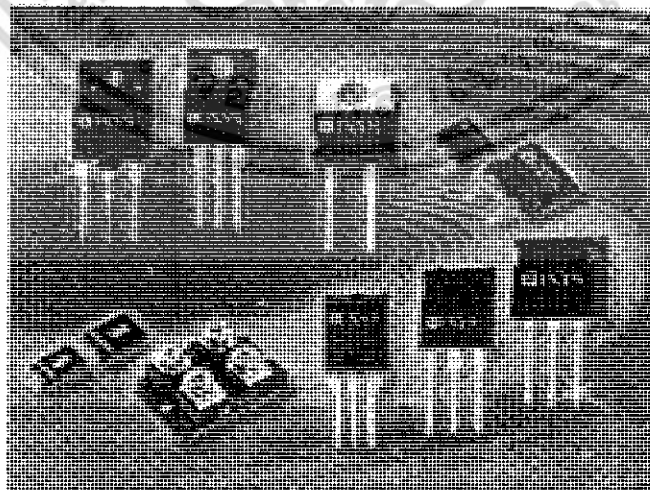
ไอจีบีทีจะมีค่าแรงดันไฟฟ้าตกคร่อมขณะนำกระแสต่ำคล้ายกับตัวทรานซิสเตอร์กำลังซึ่งมีค่าประมาณ 2-3 โวลต์ สำหรับพิกัดแรงดันไฟฟ้า 1,000 โวลต์ นอกจากนั้นไอจีบีทีจะออกแบบให้ทนแรงดันไฟฟ้าด้านลบได้คล้ายกับจีทีโอไทรสเตอร์

ไอจีบีทีจะมีช่วงเวลานำกระแสและหยุดนำกระแสมีค่าประมาณ 1 ไมโครวินาที มีขนาดพิกัดกระแสไฟฟ้าถึง 2,000 แอมป์ และพิกัดแรงดันไฟฟ้าถึง 5,000 โวลต์

ทั้งนี้จากการศึกษาและทดลองจะพบว่า ข้อดีของไอจีบีทีนั้นมีมากมาย สามารถนำไปใช้ ประโยชน์ในงานด้านต่าง ๆ ได้หลากหลาย ซึ่งจากคุณสมบัติเหล่านั้น มันจึงมีผลทำให้ตัวของไอจีบีทีนั้นมีราคาแพงเมื่อเทียบกับอุปกรณ์ตัวอื่น ซึ่งถือเป็นข้อเสียที่สำคัญข้อใหญ่ของไอจีบีที ดังนั้นการเลือกใช้งานควรจะใช้กับงานที่เหมาะสมและต้องคำนึงถึงความเสียหายที่อาจจะเกิด รวมถึงปัจจัยทางด้านต้นทุนอีกด้วย



รูปที่ 2.11 ลักษณะทั่วไปภายนอกของไอจีบีที



รูปที่ 2.12 ไอจีบีทีรูปแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ประเภทสารกึ่งตัวนำที่รวบรวมฟังก์ชันการทำงานต่างๆ ไว้ภายในตัวของมันเอง โดยมีโครงสร้างใกล้เคียงกับคอมพิวเตอร์ คือ ภายในประกอบด้วยหน่วยรับข้อมูลและโปรแกรม หน่วยประมวลผล หน่วยความจำ หน่วยแสดงผล ซึ่งส่วนประกอบเหล่านี้มีความสมบูรณ์ในตัวของมันเอง ทำให้มันมีขนาดเล็กและสามารถที่จะเขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ต่างๆ ที่เชื่อมต่อกับตัวมัน ง่ายต่อการนำไปประยุกต์ใช้

ความแตกต่างของ Microcontroller กับ Microcomputer คือ Microcontroller นั้นมีสมบูรณ์ภายในตัวของมันเอง คือ มีส่วนประกอบต่างๆ ครบถ้วน ส่วน Microcomputer นั้นต้องทำงานร่วมกับอุปกรณ์ข้างเคียงที่เชื่อมต่อจากภายนอก เช่น แป้นพิมพ์ เครื่องอ่านเขียนแผ่นบันทึก หน่วยความจำ I/O ฯลฯ

ส่วนประกอบของไมโครคอนโทรลเลอร์ มีส่วนประกอบหลักๆ ดังนี้

1. หน่วยประมวลผลกลาง CPU (Central Processing Unit)
2. หน่วยความจำ (Memory Unit)
  - o RAM (Random Access Memory)
  - o EPROM/PROM/ROM (Erasable Programmable Read Only Memory)
3. หน่วยรับและแสดงผลข้อมูล I/O (Input/Output) - serial and parallel
4. Timers
5. Interrupt Controller

นอกจากนี้ยังมีส่วนประกอบอื่นๆ เช่น Analog to Digital Converter, Pulse Width Modulator ฯลฯ ซึ่งขึ้นกับจุดประสงค์ในการใช้งาน

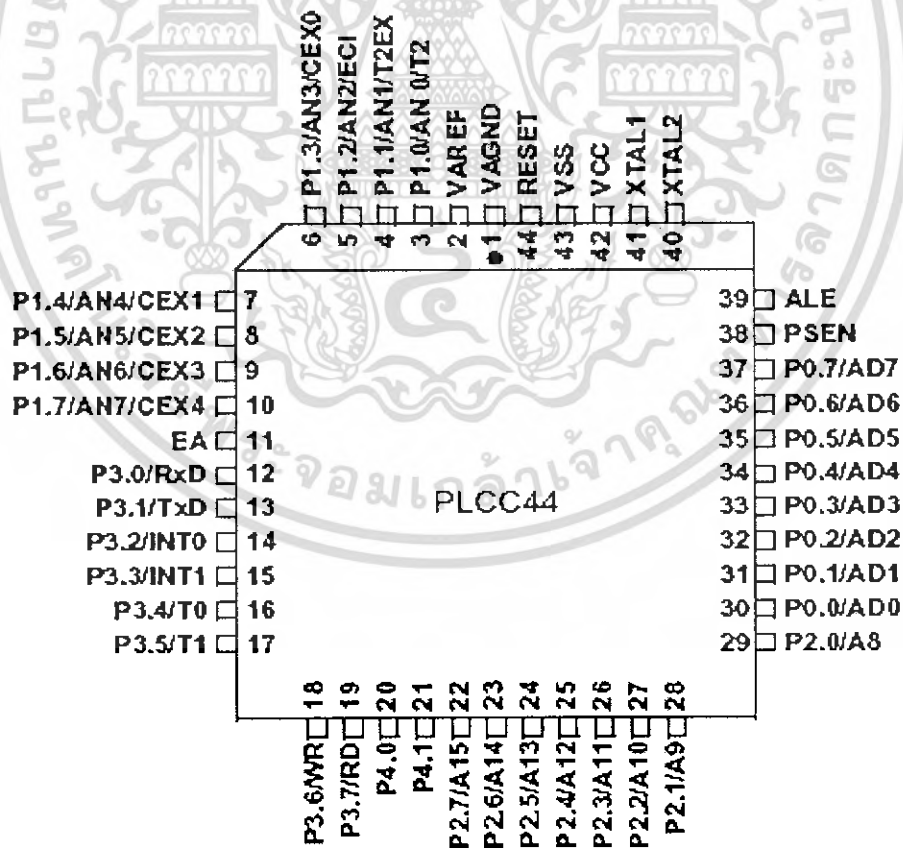
### 2.3.1 ไมโครคอนโทรลเลอร์ AT89C51AC2

AT89C51AC2 คือ ไมโครคอนโทรลเลอร์ 8 บิตความสามารถสูงหนึ่งในตระกูล 80C51 AT89C51AC2 ยังคงรูปแบบเหมือน 80C51 ประกอบด้วยแรมภายใน 256 ไบต์ พอร์ทอินเทอร์รัพท์ ไทม์เมอร์/เคาน์เตอร์ เอทิวติคอนเวอร์เตอร์ 10 บิต และพีซีเอ ทั้งยังกินพลังงานต่ำกว่า 80C51 อีกด้วย

AT89C51AC2 สามารถนำมาประยุกต์ใช้ในงาน เช่น เอทิวติคอนเวอร์เตอร์ PWM และงานควบคุมมอเตอร์



รูปที่ 2.13 ลักษณะภายนอกของ ไมโครคอนโทรลเลอร์ AT89C51AC2



เอกสารนี้รูปที่ 2.14 ลักษณะการทำงานของขาต่างๆ ของไมโครคอนโทรลเลอร์ AT89C51AC2  
 ไม่ว่าการมีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 Programmable Counter Array (PCA)

พีซีเอสามารถใช้งานเป็นไทม์เมอร์แต่มีข้อดีกว่า คือ การเขียนโปรแกรมไม่ยุ่งยากและมีความเที่ยงตรงสูง โดยที่ฐานเวลาของพีซีเอสามารถโปรแกรมได้จาก 4 แหล่ง คือ

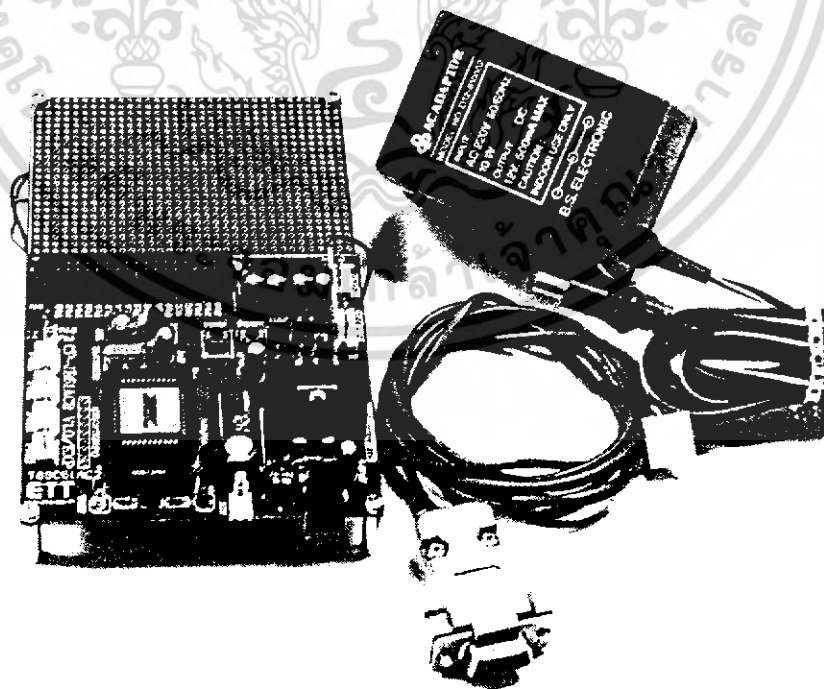
- PCA Clock frequency/6
- PCA Clock frequency/2
- Timer 0 overflow
- External input ECI(P1.2)

พีซีเอ มีการทำงานทั้งหมด 6 ฟังก์ชัน คือ

- 16 bit capture Positive edge trigger
- 16 bit capture Negative edge trigger
- 16 bit capture both Positive and Negative edge trigger
- 16 bit software timer
- 16 bit high speed output
- 8 bit pulse width modulator

Pulse Width Modulator (ในโครงการนี้สนใจเฉพาะฟังก์ชัน 8 bit pulse width modulator)

พีซีเอสามารถใช้งานให้เข้าออกของ พีดับเบิลยูเอ็มโมดูลทั้ง 4 เป็นรูปคลื่นสี่เหลี่ยมที่มีค่าดีวีดีเป็นอิสระต่อกัน แต่มีความถี่เดียวกันเนื่องจากแต่ละโมดูลใช้ฐานเวลาจากพีซีเอร่วมกัน



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอน เมื่อผู้ใดเห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.3 รีจิสเตอร์ที่เกี่ยวข้องกับการสร้างรูปคลื่นสี่เหลี่ยมด้วยพีซีเอ

#### CKCON(Clock control Register)

-	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2
---	------	-------	------	------	------	------	----

#### WDX2 Watchdog Clock

- 0 เลือกโหมด 6 คล็อก
- 1 เลือกโหมด 12 คล็อก

#### PCAX2 Programmable counter array

- 0 เลือกโหมด 6 คล็อก
- 1 เลือกโหมด 12 คล็อก

#### SIX2 UART Clock

- 0 เลือกโหมด 6 คล็อก
- 1 เลือกโหมด 12 คล็อก

#### T2X2 Timer2

#### T1X2 Timer1

#### T0X2 Timer0

- 0 เลือกโหมด 6 คล็อก
- 1 เลือกโหมด 12 คล็อก

#### X2 CPU Clock

- 0 เลือกโหมด 12 คล็อก (โหมดมาตรฐาน)
- 1 เลือกโหมด 6 คล็อก (โหมด X2)

#### CMOD (PCA Counter Mode)

CIDL	WDTE	-	-	-	CPS1	CPS0	ECF
------	------	---	---	---	------	------	-----

#### CIDL PCA idle control mode

- 0 เลือกเปิดให้พีซีเอทำงานใน Idle mode
- 1 เลือกปิดให้พีซีเอไม่ทำงานใน Idle mode

#### WDTE Watchdog timer enable

- 0 เลือกให้ Watchdog Clock ไม่ทำงาน
- 1 เลือกให้ Watchdog Clock ทำงาน

#### CPS1: CPS0 EWC Count pulse select bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 00 Internal clock FPca/6
- 01 Internal clock FPca/2
- 10 Timer0 overflow
- 11 External clock

ECF Enable PCA counter overflow interrupt bit

- 0 เลือกให้ปิดอินเตอร์รัพท์ที่ ECF ใน CMOD
- 1 เลือกให้เปิดอินเตอร์รัพท์ที่ ECF ใน CMOD

CCON(PCA Counter Control)

CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0
----	----	---	------	------	------	------	------

CF เซตโดยฮาร์ดแวร์เมื่อพีซีเอเกิดโอเวอร์โฟลล์ และจะเกิดอินเตอร์รัพท์ถ้าบิต ECF ใน CMOD ถูกเซต

CR เซตเพื่อ เปิดการทำงานของพีซีเอ เคลียร์ปิดการทำงานของพีซีเอ

CCF4 -

CCF3 -

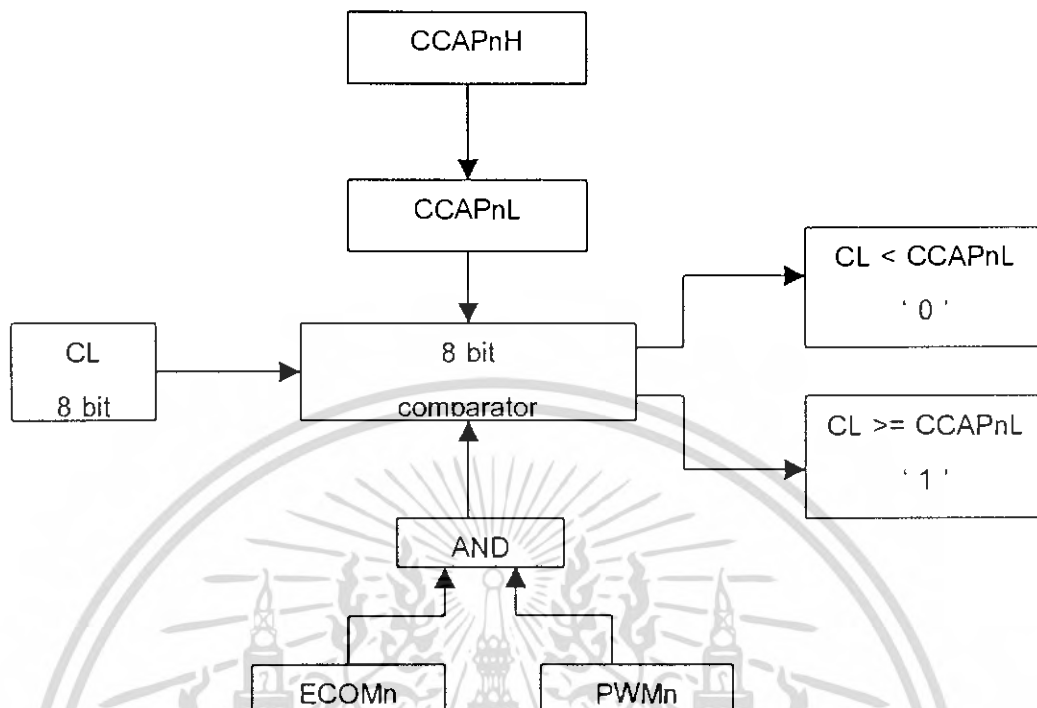
CCF2 -

CCF0 -

CCAPnH (n = 0,1,2,3,4) PCA High byte compare/capture

CCAPnL (n = 0,1,2,3,4) PCA Low byte compare/capture

### PWM Mode



รูปที่ 2.16 บล็อกการทำงานของ 8 bit pulse width modulator

ตัวอย่างโปรแกรมสร้างคลื่นสี่เหลี่ยมโดยใช้ PCA

```

CKCON EQU 08FH ; Clock Control Register
CCON EQU 0D8H ; PCA Timer/Counter Control
CMOD EQU 0D9H ; PCA Timer/Counter Mode
CL EQU 0E9H ; PCA Timer/Counter Low Byte
CH EQU 0F9H ; PCA Timer/Counter High Byte
CCAPM0 EQU 0DAH ; PAC Timer/Counter Mode0
ORG 0000H
MOV CKCON, #11111110B
MOV CMOD, #11000001B
MOV CCAP0H, #7FH
MOV CCAP0L, #7FH
MOV CCAPM0, #01000010B
MOV CCON, #01000000B
LJMP $
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 โปรแกรม วิวอล เบสิก (Visual Basic)

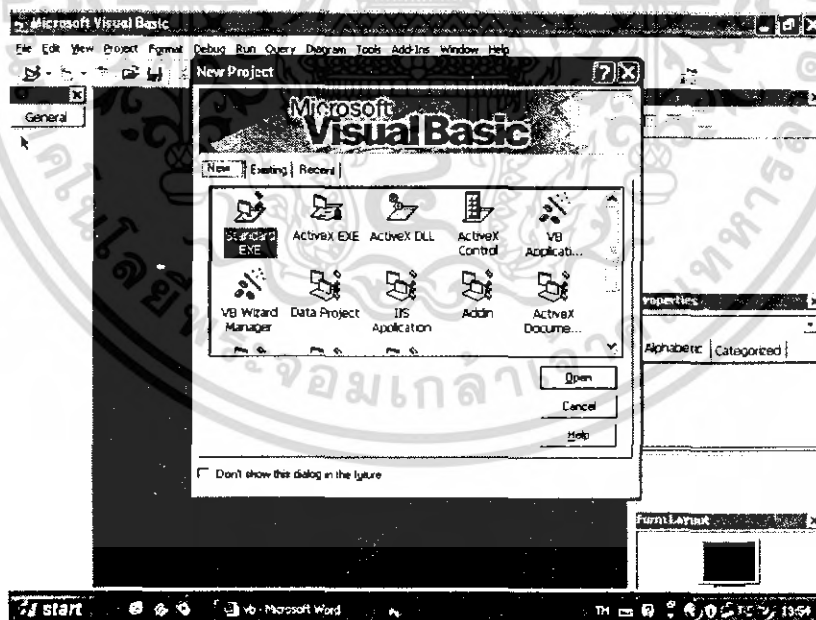
โปรแกรมวิวอล เบสิกเป็นโปรแกรมที่ใช้กันอย่างกว้างขวางในปัจจุบัน สามารถรองรับกับโปรแกรมต่างๆ ของบริษัทไมโครซอฟท์ได้เป็นอย่างดี แม้แต่ไมโครซอฟท์ ออฟฟิศ เองก็ยังต้องใช้ วิวอล เบสิก สำหรับเขียนโปรแกรมต่อเติมเพื่อการทำงานในระดับสูง หรือแก้ไขส่วนที่บกพร่องเล็กๆน้อยๆ ของโปรแกรม

ในส่วนของภาษาที่ใช้ในการเขียนโปรแกรมนั้น พัฒนามาจากภาษาเบสิก ( Basic ) ซึ่งเป็นภาษาที่มีมาแต่ดั้งเดิมในสมัยที่ยังใช้ ดอส ( Dos ) ในการพัฒนาโปรแกรม ในตอนนั้นมีภาษาเบสิกหลายตัว อาทิ QBasic, MBasic เป็นต้น

ส่วนสำคัญที่ทำให้ วิวอล เบสิก กลายมาเป็นโปรแกรมที่นิยมใช้กันมากในปัจจุบันก็เพราะตัวภาษาเบสิกถูกนำมาใช้เขียนโปรแกรมกับวินโดวส์ ทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้อย่างสะดวก และมีความหลากหลายมากขึ้น

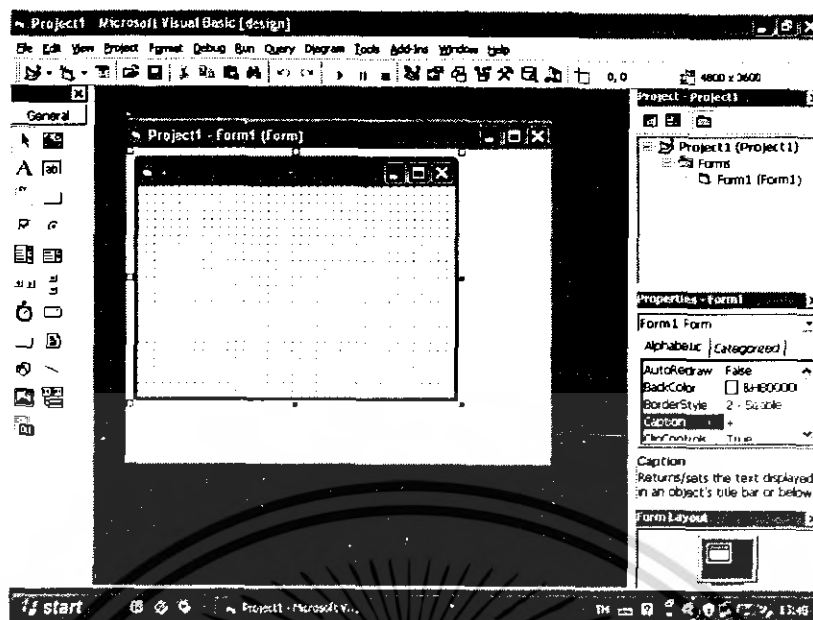
ในโครงการนี้เลือกใช้โปรแกรมวิวอล เบสิก เวอร์ชัน 6.0 ซึ่งเป็นเวอร์ชันที่นิยมใช้กันมากในปัจจุบัน ทั้งในงานด้านกราฟฟิคต่างๆ รวมไปถึงการนำมาใช้ในการควบคุมในงานอุตสาหกรรมประเภทต่างๆ อีกด้วย

นอกจากนั้นยังรวมไปถึงแหล่งความรู้ทางด้านบุคคลหรือเอกสาร หนังสือ ซึ่งมีอยู่มากมาย สำหรับเวอร์ชัน 6.0 นี้ ทำให้เมื่อเกิดปัญหาจะสามารถแก้ไขได้โดยไม่ลำบากนัก รวมถึงการเพิ่มทักษะในการเขียนโค้ดต่างๆ ให้งานออกมาดียิ่งขึ้น



รูปที่ 2.17 หน้าต่างพื้นฐานของ วิวอล เบสิก ให้เลือกโหมดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 หน้าต่างพร้อมอุปกรณ์และเครื่องมือพื้นฐานของ วิซวล เบสิก

#### คุณสมบัติเด่นๆ ของวิซวล เบสิก

นอกจากความง่ายและความสะดวกสบายในการใช้และศึกษาโปรแกรมแล้ว วิซวล เบสิก มีข้อดีเด่นๆ ที่พอสรุปได้ ดังนี้

1. ใช้เขียนโปรแกรมได้ทั้งวินโดวส์และเว็บ วิซวล เบสิก ในเวอร์ชันหลังๆ ได้รับการพัฒนาให้สามารถเขียนโปรแกรมให้ทำงานได้ทั้งบนวินโดวส์และบนอินเทอร์เน็ต โดยที่ผู้สนใจเขียนโปรแกรมบนวินโดวส์สามารถประยุกต์ใช้ความรู้ที่ตนมีอยู่ในการพัฒนาเว็บแอปพลิเคชันได้ง่าย โดยสามารถเรียนรู้ VB Script ซึ่งเป็นภาษาในตระกูลวิซวล เบสิก ได้อย่างรวดเร็ว
2. แก้ไขโปรแกรมที่เขียนขึ้นได้ง่าย ในบางครั้งการเขียนโปรแกรมอาจจะมีข้อผิดพลาด การแก้ไขหรือปรับปรุงในภายหลังของวิซวล เบสิก นั้นทำได้ง่ายโดยมีเครื่องมือต่างๆ คอยช่วยเหลืออยู่
3. มีเอกสารให้ศึกษาอ้างอิงมาก จากที่กล่าวมาแล้วว่าโปรแกรม วิซวล เบสิก เป็นที่นิยมใช้มากในปัจจุบัน เมื่อมีปัญหาหรือข้อสงสัยเกิดขึ้นจะหาคำตอบได้ง่ายจากผู้มีความรู้รวมทั้งเอกสารในรูปแบบต่างๆ รวมถึงจากเว็บไซต์ของไมโครซอฟท์และเว็บไซต์อื่นๆ ด้วย
4. เป็นโปรแกรมที่มีอนาคต วิซวล เบสิก นั้นเป็นภาษาเขียนโปรแกรมที่ไมโครซอฟท์ถือว่าเป็นภาษาที่สำคัญมาก จึงถูกพัฒนามาอย่างต่อเนื่อง คาดว่าเวอร์ชันใหม่ๆ จะถูกเพิ่มความสามารถอย่างมาก ทั้งตัวภาษาและตัวผลงานที่ได้จากการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 การสื่อสารผ่านทางพอร์ทอนุกรม (Serial Port Communication)

ในอดีตนั้นการเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์มีความเฉพาะตัวเป็นอย่างยิ่ง การควบคุมในอดีตควบคุมผ่านภาษา Assembly ซึ่งเป็นภาษาที่ผูกติดกับฮาร์ดแวร์มาก แม้จะให้ผลการทำงานดี แต่การเขียนและการแก้ไขโปรแกรมทำได้ยากมากเพราะ Assembly เป็นภาษาที่ทำความเข้าใจยากมากและเป็นภาษาที่ไม่เป็นโครงสร้าง

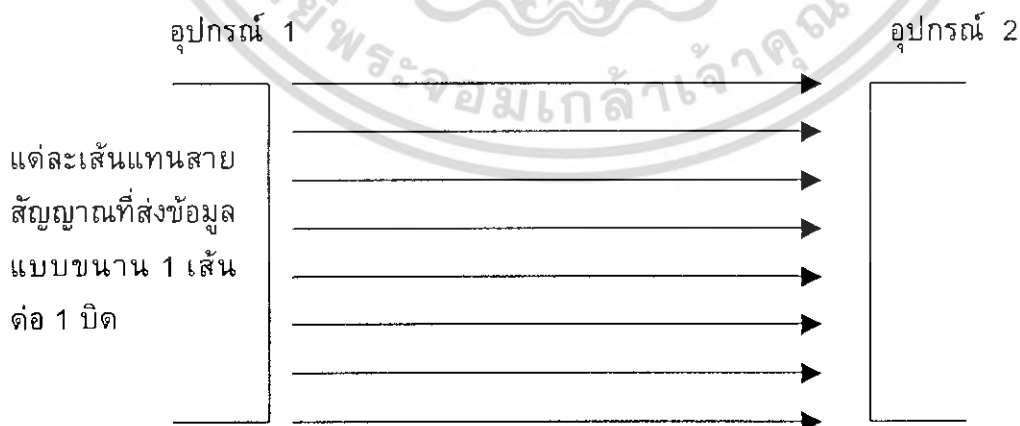
ต่อมาเราได้ใช้งานภาษาโปรแกรมยุคใหม่ ๆ เพื่อใช้ในการเขียนโปรแกรม ไม่ว่าจะเป็นภาษา C, Pascal ซึ่งภาษา C จะได้รับความนิยมมากเพราะมีประสิทธิภาพใกล้เคียงกับ Assembly และเข้าใจได้ง่ายกว่า

แต่ในยุคปัจจุบัน การเขียนโปรแกรมส่วนใหญ่จะเป็นการเขียนโปรแกรมแบบติดต่อกับผู้ใช้ซึ่งจะเขียนโปรแกรมได้ง่ายและรวดเร็ว ทำให้ภาษาโปรแกรมอย่าง Visual Basic, Visual C++, Delphi เป็นที่นิยมอย่างมาก ทำให้มีความพยายามที่จะนำเอาภาษาโปรแกรมยุคใหม่ ๆ เหล่านี้มาใช้ในการติดต่อและควบคุมฮาร์ดแวร์ แม้ว่าในระยะแรกจะลำบากแต่ปัจจุบันได้รับการพัฒนาให้ครอบคลุมเกือบทั้งหมดแล้ว

ในโครงการนี้จะใช้การติดต่อสื่อสารระหว่างโปรแกรมกับฮาร์ดแวร์ผ่านทางพอร์ทอนุกรม โดยอ้างอิงมาตรฐาน RS - 232 ใช้โปรแกรมวิซวล เบสิก เป็นหลัก

### 2.5.1 รูปแบบการสื่อสารแบบอนุกรม

ข้อมูลในการสื่อสารที่เราใช้อยู่นี้ จะเป็นข้อมูลที่มีความยาวขนาด 1 ไบต์ ที่ประกอบด้วย 8 บิตซึ่งโดยปกติถ้าเราจะให้ส่งข้อมูลพร้อม ๆ กันไป 8 บิตจะเป็นวิธีการส่งข้อมูลแบบขนาน แสดงได้ดังรูป 1 จะเป็นการส่งข้อมูลขนาด 8 บิตพร้อมกันไปยังอุปกรณ์ภายนอก และจะต้องมีจำนวนของสายสัญญาณจำนวน 8 เส้น เพื่อให้พอดีกับจำนวนของบิตที่ต้องการจะส่ง การส่งข้อมูลแบบขนานจึงทำให้มีการส่งข้อมูลที่มีความรวดเร็ว แต่ถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้จำนวนของสาย และระยะทางของสายมากขึ้นจึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง

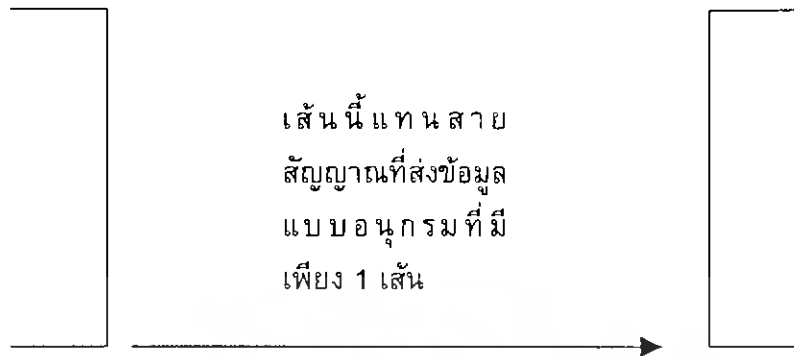


รูปที่ 2.19 การส่งข้อมูลแบบขนาน ในหนึ่งจังหวะของช่วงเวลาจะส่งข้อมูลออกพร้อมกัน 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## อุปกรณ์ 1

## อุปกรณ์ 2

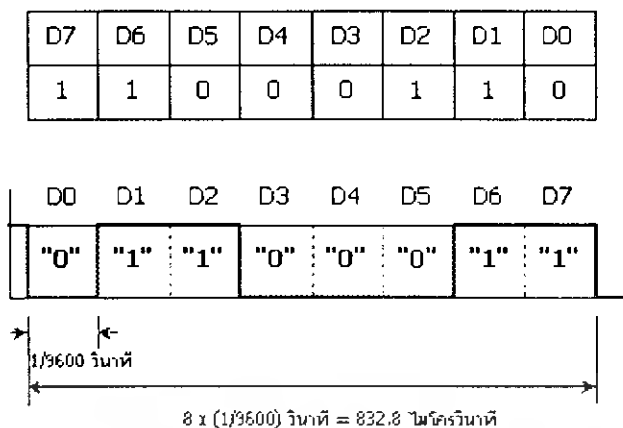


รูปที่ 2.20 การส่งข้อมูลอนุกรม ในหนึ่งจังหวะของช่วงเวลาจะส่งข้อมูลออก 1 บิต

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ ในการสื่อสาร โดยจะใช้สายเพียงเส้นเดียวในการส่งข้อมูล หรือรับข้อมูล (คำว่าเส้นเดียวหมายความว่าสายส่ง(TxD) 1 เส้น สายรับ(RxD) 1 เส้น และสายกราวด์ รวม(Ground) 1 เส้น ) นำมาใช้สื่อสารข้อมูลกับอุปกรณ์ภายนอกในระยะทางที่ไกล ดังนั้น ถ้าหากต้องการส่งข้อมูลขนาด 1 ไบต์ ก็จะทำให้การส่งข้อมูลออกไปที่ละบิตเป็นลำดับไป จนกว่าจะครบจำนวนทั้ง 8 บิต แล้วก็จะเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม ข้อมูลจะถูกส่งไปตามสายสัญญาณที่ละบิตตามจังหวะเวลาที่กำหนด เป็นความกว้างของพัลส์ โดยจังหวะเวลาที่กล่าวนี้จะต้องมีมาตรฐานและเท่ากันของทั้งฝ่ายส่ง และฝ่ายรับด้วย ในการรับสัญญาณที่ส่งมาที่ละบิต จะทำการตรวจสอบระดับแรงดันของสัญญาณที่เข้ามาเพื่อแปลงเป็นลอจิก "0" หรือ "1" เมื่อรับข้อมูลเข้ามาครบใน 1 ไบต์ที่กำหนดไว้ ก็จะถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนเดิม

#### 2.5.2 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

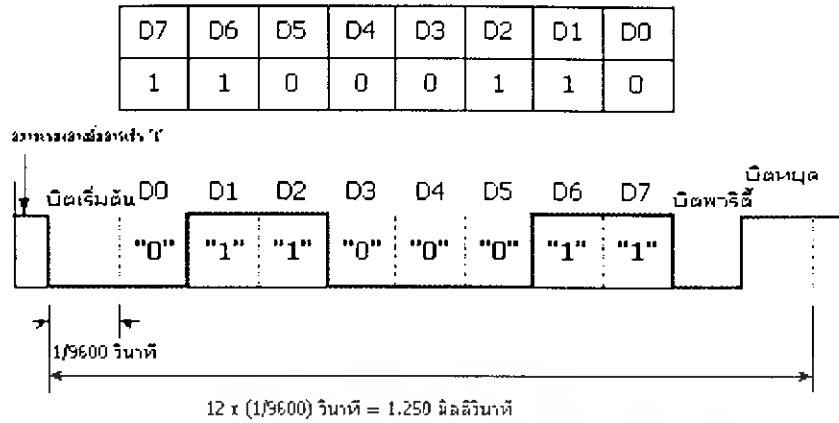
ในการสื่อสารข้อมูลแบบอนุกรม เพื่อรับหรือส่งข้อมูล จะเป็นลักษณะของกลุ่มข้อมูลที่มาตรฐาน ดังนั้นอัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่งโดยทั่วไปเราจะระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 และ 9,600 บิตต่อวินาที ในรูป 3 ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ  $1/9600$  หรือ  $104.1$  ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่าเท่ากับ  $8 \times 104.1$  หรือ  $832.8$  ไมโครวินาที



รูปที่ 2.21 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส เป็นวิธีการรับและส่งข้อมูลโดยไม่ต้องอาศัยสัญญาณนาฬิกาส่งร่วมไปด้วย แต่จะใช้อัตราความเร็วของจำนวนข้อมูลต่อวินาที และจะทำการเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมากยิ่งขึ้น แสดงดังรูปที่ 4 ซึ่งประกอบด้วยกัน 4 ส่วนคือ

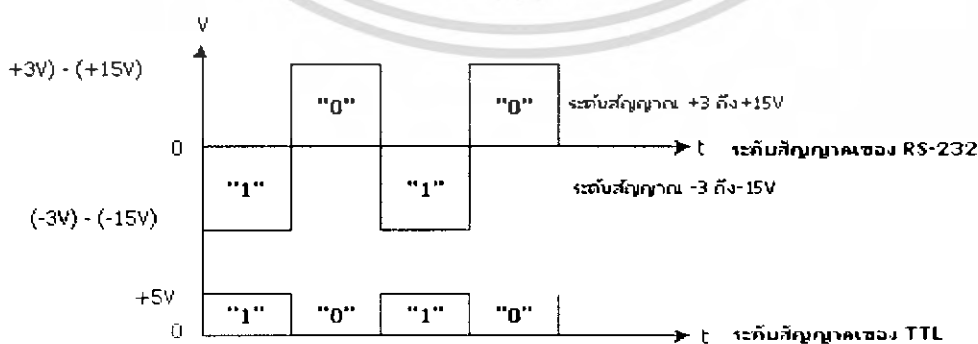
1. บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิต จะเป็นระดับลอจิกตรงกันข้ามกับระดับลอจิก ของสถานะสายสื่อสาร ขณะที่ยังไม่มีการส่งข้อมูล
  2. บิตข้อมูล (Data bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อนหรือ บิต LSB ก่อน โดยข้อมูลที่จะส่งอาจจะมีขนาด 5, 6, 7 หรือ 8 บิตก็ได้
  3. บิตแสดงสถานะเลขคู่หรือเลขคี่ (Parity bit) มีขนาด 1 บิตโดยบิตนี้จะนำไปต่อท้ายกับบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น "1" โดยเลือกการส่งข้อมูลเป็นแบบ พาริตีคู่ หรือ พาริตีคี่ ตัวอย่าง ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตีคู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะทำให้บิตพาริตีนี้เป็น "1" เพื่อจะได้จำนวนเลข "1" เป็นคู่นั่นเอง ทำนองเดียวกันทางด้านรับเองก็ต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น "1" รวมทั้งบิตพาริตี 1 บิต ถ้ามีค่า "1" เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง
- \*\*\*\* สามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตีบิตก็ได้
4. บิตสุดท้ายหรือบิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล โดยจะทำให้ขาข้อมูลมีสถานะ ลอจิกเป็น "1" ซึ่งอาจมีจำนวนมากกว่า หนึ่งบิตก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต



รูปที่ 2.22 ส่งข้อมูล 8 บิตพร้อมด้วย บิตเริ่มต้น,บิตพาริตี,บิตหยุด ความเร็ว 9600 บิต/วินาที

### 2.5.3 การเชื่อมต่อพอร์ทอนุกรมมาตรฐานRS-232

การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรมEIA RS-232 เป็นมาตรฐานอุตสาหกรรม โดย คณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบ อะซิงโครนัส 2 ทิศทาง เพื่อให้มีการใช้งาน ในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณจะกำหนด ความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุต โดยจะมีระดับสัญญาณเริ่มตั้งแต่ 3 โวลท์ จนถึง 15 โวลท์ สำหรับลอจิก "0" และมีระดับแรงดันที่ -3 โวลท์ จนถึง -15 โวลท์ สำหรับลอจิก "1" ดังแสดงใน รูปที่ 2.22 ดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่าง ออกไปจากระบบไอซีดิจิทัลทั่วไป การต่อใช้งานจึงต้อง มีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรง ดันจาก 0 - 5 โวลท์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า - 3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งาน

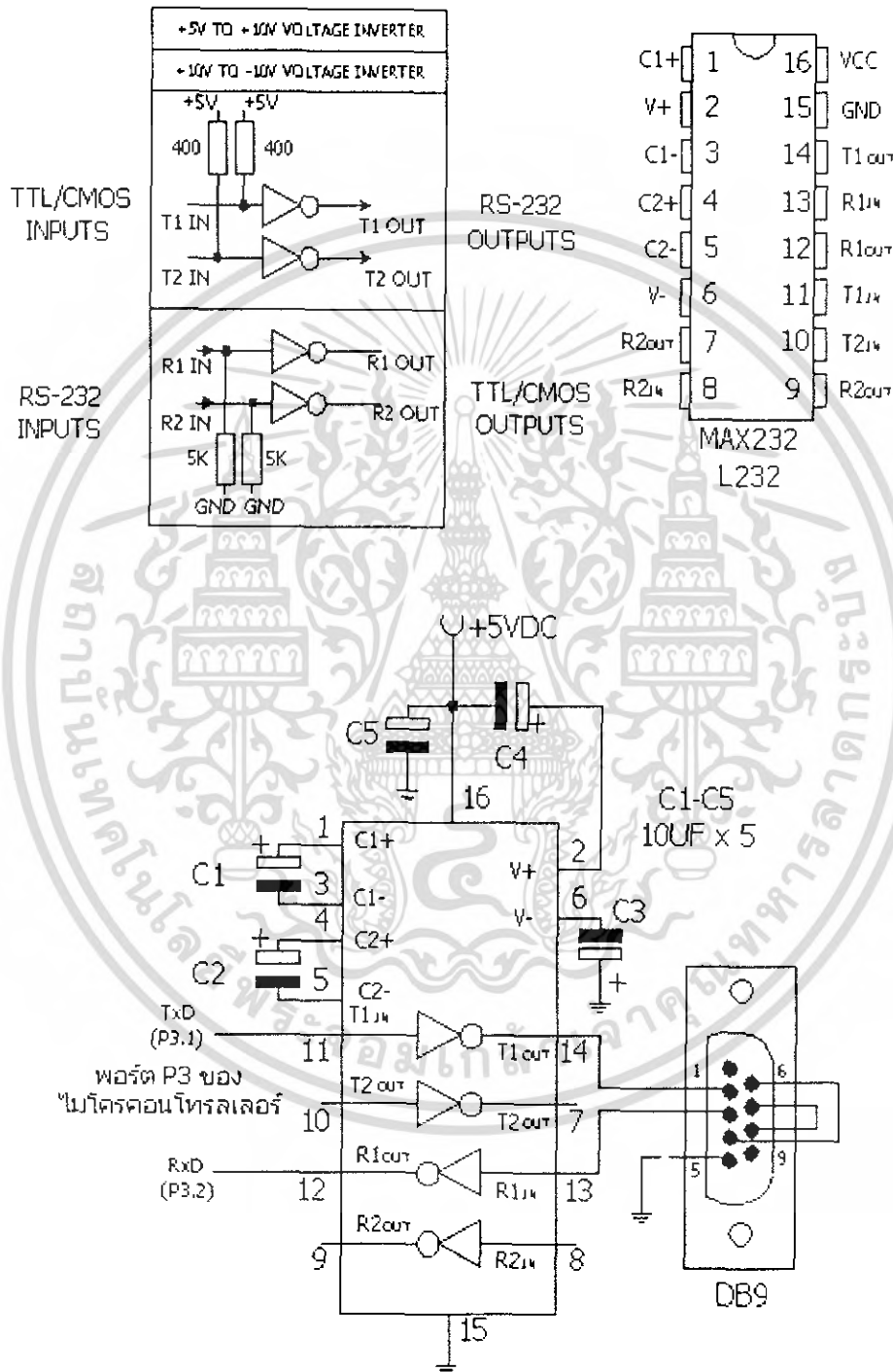


รูปที่ 2.23 แรงดันสัญญาณของพอร์ทอนุกรม RS-232 กับ TTL ในสถานะลอจิก"1"และ "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.4 ไอซี MAX 232

ไอซี MAX232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232 เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้



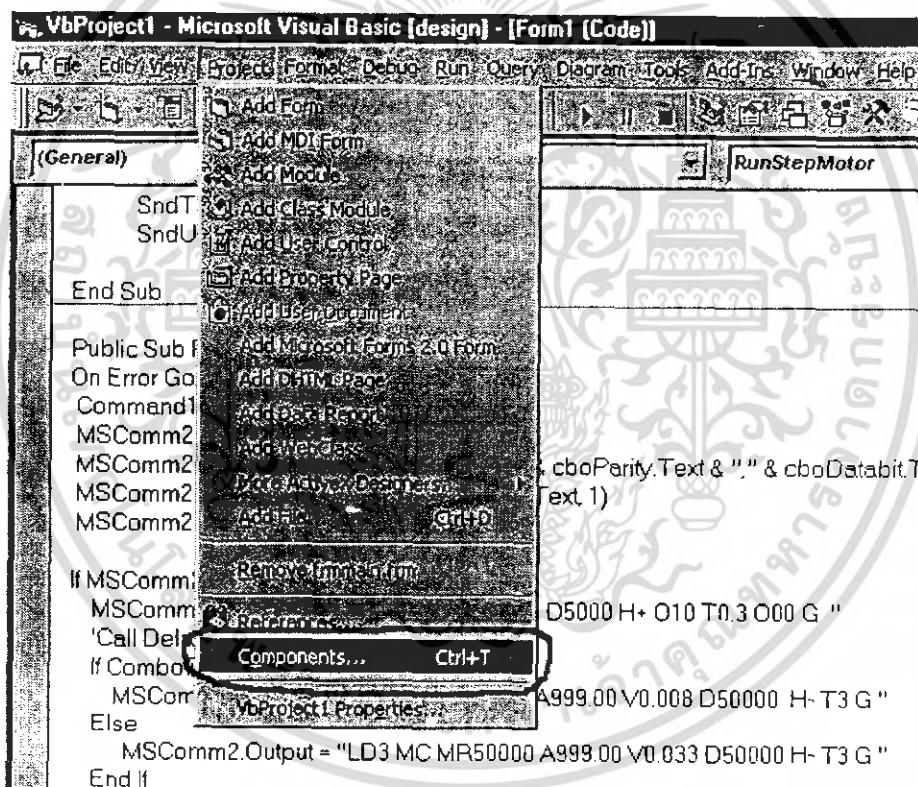
รูปที่ 2.24 แสดงตำแหน่งขาของไอซี MAX232 และการต่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.5 การสื่อสารผ่าน serial port ของโปรแกรม visual basic

ด้านการติดต่อ I/O ผ่านพอร์ตจะเน้นในลักษณะที่นำไปประยุกต์ใช้การติดต่อกับไมโครคอนโทรลเลอร์ การเขียนโปรแกรมติดต่อกับฮาร์ดแวร์โดย Visual Basic บนวินโดวส์เราไม่จำเป็นต้องเขียนโค้ดที่ค่อนข้างยาว เราสามารถเขียนโปรแกรมได้ง่ายขึ้นซึ่งจะมีตัวคอนโทรลชื่อ MSComm ที่สามารถติดต่อผ่าน RS-232 ได้

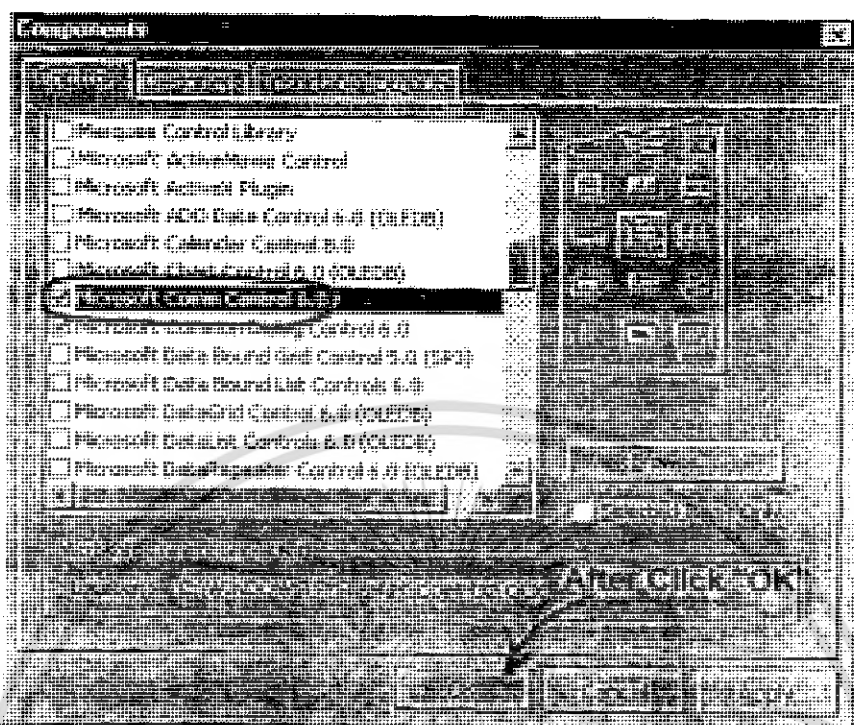
สามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดยที่ต้อง กำหนด Custom Control เข้าไปที่ เมนู Project--->Components เลือกที่ช่อง MSComm จะปรากฏ เป็นรูปไอคอนโทรศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของโปรแกรม โดยสามารถทำตามวิธีที่กล่าวมา



รูปที่ 2.25 ขั้นตอนการเลือกฟังก์ชัน MSComm มาใช้งาน

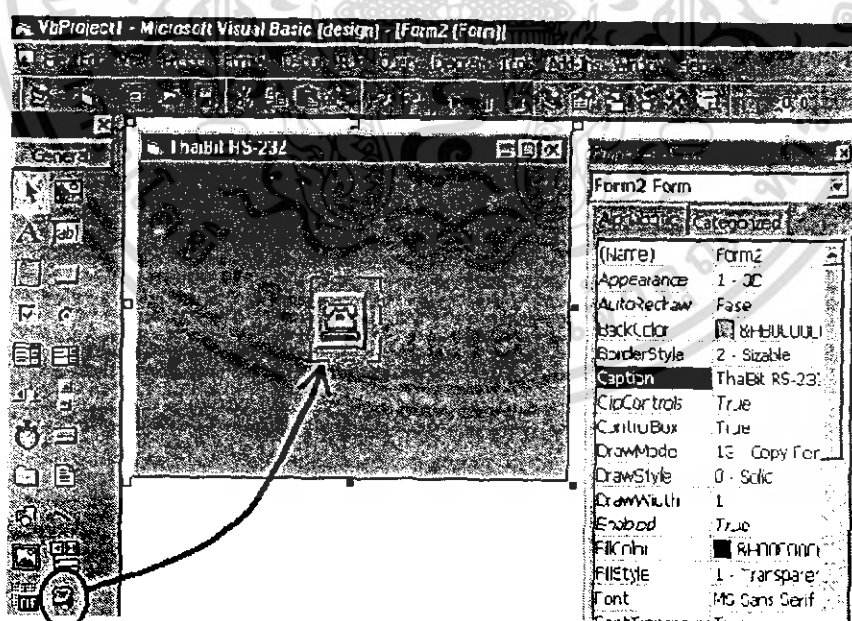
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกชื่อ Control ชื่อ Microsoft Comm Control 6.0 ดังรูปด้านล่าง



รูปที่ 2.26 เลือกฟังก์ชัน Microsoft Comm Control 6.0 ใน Components

ลาก Control ชื่อ Microsoft Comm จาก Toolbox มาไว้บน Form ดังรูปด้านล่าง



รูปที่ 2.27 ลาก Control ชื่อ Microsoft Comm จาก Toolbox มาไว้บน Form

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ

1. การติดต่อแบบอินเทอร์รัพต์ ขบวนการอินเทอร์รัพต์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง จะส่งเป็นรหัสแอสกี เราจะเขียนโปรแกรมอินเทอร์รัพต์ โดยเมื่อข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnComm Event

2. การติดต่อแบบโพลลิ่ง ในระบบพีซี การโพลลิ่งมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบคีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่ได้รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทาง คีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามา เรียกว่า "Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การโพลแบบ "Round Robin" แทน แต่ใน VB เราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do....Loop ก็ได้

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้น ก็คือ OnComm Event ซึ่งจะใช้ในการติดต่อแบบอินเทอร์รัพต์ การเขียนโปรแกรมติดต่อกับ Serial Port แบบธรรมดาจะใช้ comEvent เพียง comEvReceive, comEvSend

### องค์ประกอบในการใช้ MSComm

- ComPort คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อ RS-232 (Com1, Com2)
- Setting คือ เราต้องกำหนดอัตรา Baud, Parity, Data ( จำนวนบิต ), Stop ตัวอย่าง 1200, n, 8, 1
- HandShaking คือ เราจะกำหนดได้ 4 แบบ 1.comNone 2.comXonXoff 3. comRTS 4.comTRSXonXoff
- \*\*\* โครงการนี้เน้นที่การติดต่อกับไมโครคอนโทรลเลอร์ ที่มีแค่ 3 สาย จึงไม่ใช้งาน HandShaking

### การใช้ Buffer ในการรับส่งข้อมูล

- InBufferSize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา
- OutBufferSize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป
- Rthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา
- Sthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลออกไป
- InputLen คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล

เอกสาร EofEnable คือ การที่บอกว่าสิ้นสุดของไฟล์ (EOF) End of File อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

1. Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่จะต่อใช้งาน

ตัวอย่าง MSComm1.CommPort=1

ในที่นี้เลือกจะใช้ Com1อยู่ที่ด้านหลังเครื่องคอมพิวเตอร์

2. Property ชื่อ Settings คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอด ของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ต่าง ๆ ค่าดังนี้

MSComm1.Settings="Baud(อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช่ใส่ N,จำนวนบิตข้อมูล,บิต สต๊อป"

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

3. Property ชื่อ InputLen คือ กำหนดขนาดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์

ตัวอย่าง MSComm1.InputLen=1

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

ตัวอย่าง MSComm1.PortOpen=True

5. Property ชื่อ Rthreshold คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล(Comport)มันทำให้เกิดCommEvent ใน OnComm Event

ตัวอย่าง MSComm1.Rthreshold =1

จากรายละเอียดที่กล่าวมา เราจะมาเขียนใน โปรซีเจอร์ VB ซึ่งจะไว้ที่ Sub Form\_Load ( )

Private Sub Form\_Load()

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

MSComm1.CommPort=1

MSComm1.InputLen=1

MSComm1.PortOpen=True

MSComm1.Rthreshold =1

End Sub

#### 2.5.6 การสื่อสารผ่าน พอร์ตอนุกรม (serial port) ของไมโครคอนโทรลเลอร์

พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ จะมีโครงสร้างเป็นแบบฟูลดูเพล็กซ์ ซึ่งรับและส่งข้อมูลในเวลาเดียวกันได้ โดยจะมีรีจิสเตอร์ SBUF ( Serial Data Buffer ) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรม โดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการโดยวิธีทางฮาร์ดแวร์ในการเลื่อนบิต เพื่อส่งสัญญาณออกไปภายนอก หลังจากที่มีการส่งข้อมูลออกไปจนครบแล้ว จึงจะทำการเซตบิตโดยกำหนดค่าของแฟล็ก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ "1" เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้ว และพร้อมที่จะส่งข้อมูลไบต์ต่อไปได้

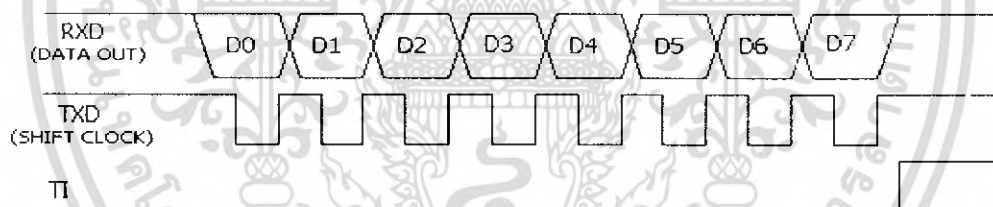
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้น โดยการกำหนดค่าของบิต REN ที่อยู่ในรีจิสเตอร์ SCON ให้มีค่าเป็นสถานะ " 1 " หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอกก็จะทำการเลื่อนข้อมูลไปโดยอัตโนมัติและเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ในรีจิสเตอร์ SBUF และจะทำการเซตที่บิต RI ให้เป็นสถานะ"1" ซึ่งส่งผลให้เกิดการอินเตอร์รัพท์โปรแกรมขึ้น

### โหมดการติดต่อทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์จะแบ่งออกได้เป็น 4 โหมดด้วยกัน และใน แต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้

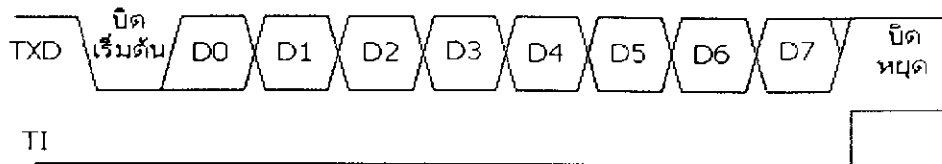
**โหมด 0 :** จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรม การส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RXD เพียงขาเดียว และจะไม่มีการส่งบิตเริ่มต้น (Start bit ) ส่วนขา TXD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อน ข้อมูลกับวงจรภายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิพรีจิสเตอร์ (Shift Register) จุดประสงค์เพื่อใช้ในการขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุต ให้มีจำนวนมาก แต่ในโหมด 0 เรามักจะไม่ค่อยนิยมนำมาใช้งาน เพราะไอซี ไมโครคอนโทรลเลอร์ในปัจจุบันที่เราใช้อยู่ นั้น มีจำนวนพอร์ตที่มากพอ และมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันให้เลือกจำนวนพอร์ตใช้งานมากมายอยู่แล้ว



รูปที่ 2.28 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0

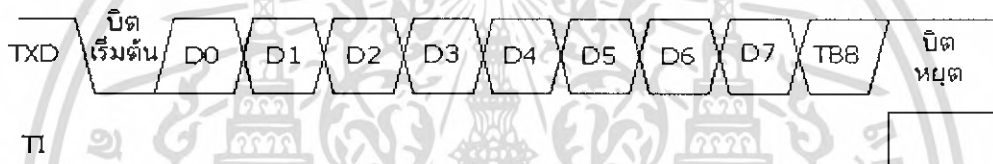
**โหมด 1 :** จะเป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ในการติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิต จะเข้ามาทางขา RXD และส่งข้อมูลออกแบบอนุกรมทางขา TXD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0) , 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน ) และ บิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุด (Stop bit) ที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1 จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โพล์ของไทม์เมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 แสดงสัญญาณการส่งข้อมูลในโหมด 1

โหมด 2 : จะเป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD ซึ่งข้อมูล 11 บิตประกอบด้วย บิตแรกจะเป็นบิตเริ่มต้น (Start bit ค่า 0) , 9 บิตต่อมาจะเป็นบิตของข้อมูล และบิตสุดท้ายจะเป็นบิตหยุด 1 บิต (Stop bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity bit )



รูปที่ 2.30 แสดงสัญญาณการส่งข้อมูลในโหมด 2

โหมด 3 : เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับและส่งข้อมูลได้ตามต้องการ

#### รีจิสเตอร์ที่ใช้ในการควบคุมและรับส่งข้อมูลของพอร์ตอนุกรม

รีจิสเตอร์ที่ใช้งานในการติดต่อสื่อสารทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จะประกอบด้วยรีจิสเตอร์ SCON ที่ทำหน้าที่ควบคุมการทำงาน รีจิสเตอร์ SBUF จะใช้ในการเก็บข้อมูลที่จะรับหรือส่ง และรีจิสเตอร์ PCON ซึ่งจะใช้ ในการกำหนดอัตรารับส่ง โดยรีจิสเตอร์ แต่ละตัวจะมีหน้าที่และการทำงานในแต่ละบิตดังต่อไปนี้

รีจิสเตอร์ SCON (Serial Port Control Register) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในส่วนของรีจิสเตอร์พิเศษ (Special Function Register) ในตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบต์ และแบบบิตได้ โดยจะทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงาน และเก็บข้อมูลในบิตที่ 9 ( ซึ่งโดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล(TB8) รายละเอียดของแต่ละบิตมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

รูปที่ 2.31 แสดงบิตต่างๆของรีจิสเตอร์ SCON (Serial Port Control Register)

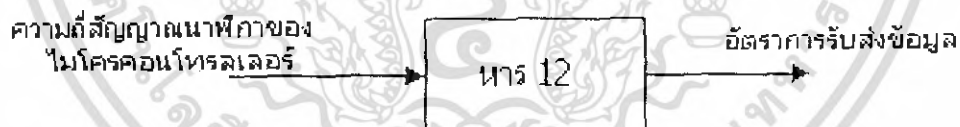
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

รูปที่ 2.32 แสดงรีจิสเตอร์ PCON เพื่อกำหนดอัตราการรับส่งข้อมูล

รีจิสเตอร์ PCON (Power Control) เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดี่ยวนั้น โดยจะประกอบด้วยบิตดังรูปที่ 2.31

#### การกำหนดอัตรารับและส่งข้อมูล (Baud Rate Generator)

โหมด 0 : อัตรารับส่ง (Baud rate) ในโหมด 0 ไม่สามารถกำหนดอัตรารับส่งเองได้ แต่จะมีค่าเท่ากับความเร็ว สัญญาณนาฬิกาไมโครคอนโทรลเลอร์หารด้วย 12 หรืออาจกล่าวได้ว่า ขึ้นกับค่าความถี่ของคริสตอลที่นำมาต่อใช้งาน แล้วหารด้วย 12 นั้นเอง



รูปที่ 2.33 อัตรารับส่งโหมด 0 = ความถี่สัญญาณนาฬิกา หาร 12

โหมด 2 : อัตรารับส่ง (Baud rate) ในโหมด 2 เลือกได้ 2 อัตราความเร็วในการรับส่งข้อมูล โดยหาได้จากความถี่ของสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์หารด้วย 32 หรือหารด้วย 64 เรียกว่า SMOD0 และ SMOD1 ซึ่งจะขึ้นอยู่กับค่าสถานะของบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON เป็นตัวเลือก



ให้มีการร้องขออินเทอร์รัพท์ที่เกิดจากไทม์เมอร์1 ในกรณีใดๆ ในระหว่างนั้นอีก (โดยการควบคุมที่ รีจิสเตอร์ IE) อัตราการรับและส่งข้อมูลจะมาจากอัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์1 และค่าของบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON สามารถที่จะเขียนเป็นสูตรสำหรับการคำนวณหาค่าอัตรารับส่ง (Baudrate) ได้ดังสมการต่อไปนี้

$$\text{อัตรารับส่งโหมด 1 และ 3} = 2^{\text{SMOD}} / 32 \times (\text{อัตราการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 1})$$

ในการใช้งานสื่อสารข้อมูลแบบอนุกรมนั้น เราจะนิยมใช้งานไทม์เมอร์1 ในลักษณะของโหมด 2 (Auto-reload กำหนดค่าการควบคุมที่รีจิสเตอร์ TMOD = 0010XXXX) การคำนวณหาค่าอัตรารับส่ง (Baud rate) จะได้สมการดังต่อไปนี้

$$\text{อัตรารับส่งโหมด 1 และ 3} = 2^{\text{SMOD}} / 32 \times [\text{Oscillator} / 12 \times (256 - (\text{TH1}))]$$

เราสามารถที่จะให้ไทม์เมอร์1 มีอัตราการรับส่งต่างๆ ได้โดยใช้ ไทม์เมอร์1 ในโหมด1 ทำงานลักษณะของตัวจับเวลาแบบ 16 บิต (มีค่าของการควบคุมที่รีจิสเตอร์ TMOD = 0001XXXX) และให้มีการอินเทอร์รัพท์จากไทม์เมอร์1 โดยให้โปรแกรมตอบสนองการอินเทอร์รัพท์ของไทม์เมอร์1 แล้วจะทำการกำหนดค่าเริ่มต้นใหม่ (Reload) ให้กับตัวจับเวลา ซึ่งเป็นการทำงานแบบ 16 บิตทางซอฟต์แวร์ (Software reload) เนื่องจากการทำงานในโหมด 1 ของ ไทม์เมอร์1 ไม่สามารถโหลดค่าใหม่เองด้วยฮาร์ดแวร์ได้

ตารางที่ 2.1 จะแสดงตารางอัตรารับส่งข้อมูลที่กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้ บอดเรทค่ามาตรฐานต่างๆ และการกำหนดค่าของรีจิสเตอร์ TMOD ที่บิต C/T และรีจิสเตอร์ PCON ที่บิต SMOD เพื่อใช้งาน

ตารางที่ 2.1 อัตรารับส่งข้อมูลที่กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้บอดเรทค่ามาตรฐาน

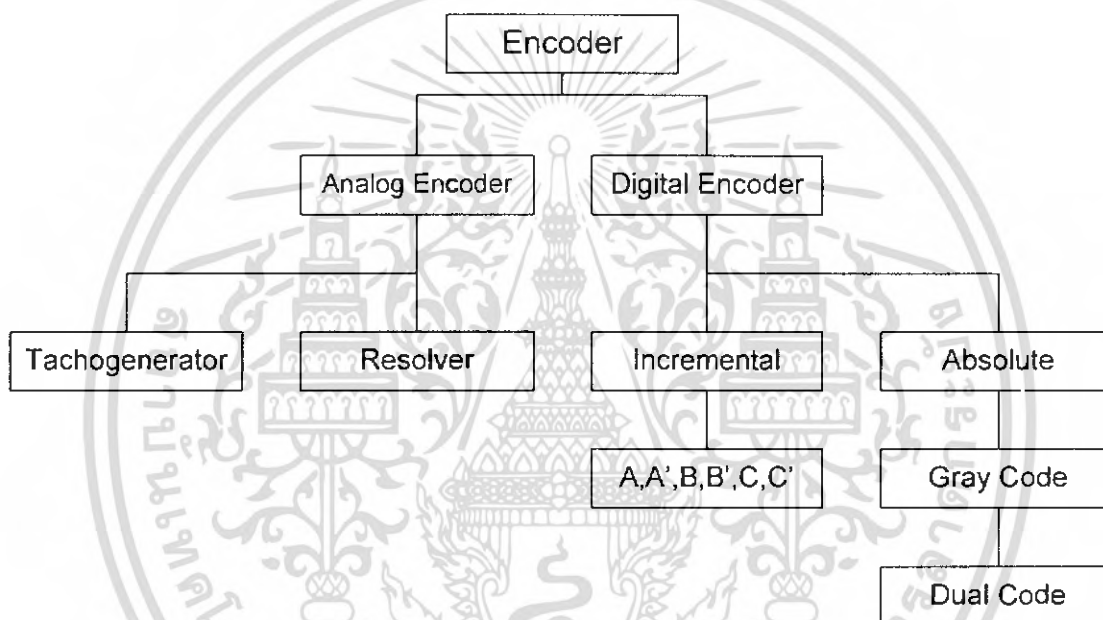
Baud rate	F (MHz)	บิต SMOD	Timer 1		
			บิต C/T	MODE	Reload Value
82.5 k	12.000	1	0	2	FFH
19.25 k	11.059	1	0	2	FDH
9600	11.059	0	0	2	FDH
4800	11.059	0	0	2	FAH
2400	11.059	0	0	2	F4H
1200	11.059	0	0	2	E8H
137.5	11.059	0	0	2	1DH
110	6.000	0	0	2	72H
110	12.000	0	0	1	FEEDH

จะสังเกตได้ว่าการใช้งานจากความถี่คริสตอลเรามักจะนิยมจะเลือกใช้ค่าความถี่ที่มีค่า 11.0592 MHz เนื่องจากการกำหนดค่าอัตราการรับส่งข้อมูลจะสามารถกำหนดค่าบอดเรทใน โหมด 1 และโหมด3 ได้ และเป็นค่าที่มาตรฐานเช่น 1200,2400,4800, 9600,19200 ดังนั้นจึงเป็นเหตุผลที่เราเลือกใช้คริสตอลค่าความถี่ 11.0592 MHz มากกว่าค่า 12 MHz ในการกำหนดอัตรารับส่งข้อมูลยังสามารถที่จะใช้ไทม์เมอร์ 2 กำหนดค่าได้ แต่จะยังไม่ขอกกล่าวในที่นี้เพราะไอซีที่เรานำมาทดลองนั้นไม่มีไทม์เมอร์ 2 ใช้งาน

## 2.6 อุปกรณ์ตรวจจับตำแหน่งของโรเตอร์แบบ incremental (Incremental Encoder)

ในระบบบังคับตำแหน่งหรือตรวจสอบความเร็วของมอเตอร์ ส่วนมากแล้วจะนิยมใช้อินคริเมนทอลเอนโค้ดเดอร์มาใช้ในกาสรทำงาน โดยที่เอนโค้ดเดอร์จะสร้างพัลส์ที่แปรผันตรงกับการหมุนของเพลลา ซึ่งสามารถนำไปใช้ในการรับรู้ความเร็วของเพลลามอเตอร์ขณะทำงานในรูปของอัตราจำนวนพัลส์ได้ แล้วนำผลที่ได้ป้อนกลับมาแสดงผลโดยผ่านไมโครคอนโทรลเลอร์และไปแสดงผลที่ระบบควบคุมต่อไป

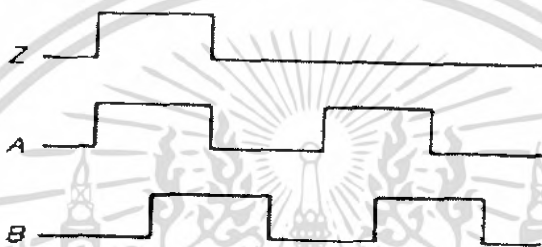
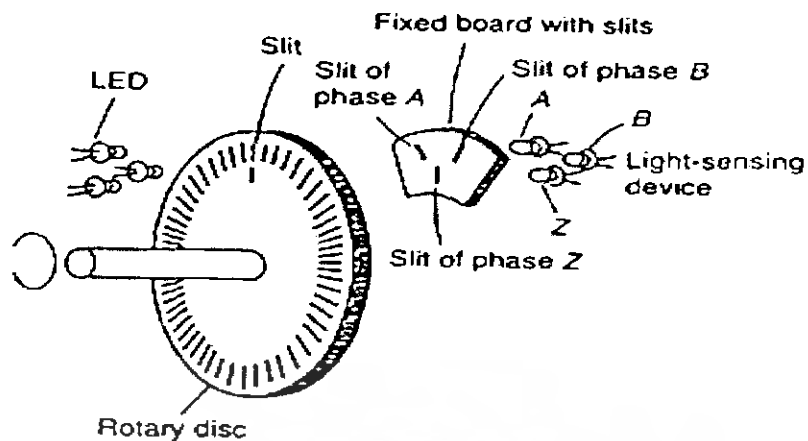
จากบทบาทของเอนโค้ดเดอร์ซึ่งทำหน้าที่ตรวจวัดความเร็ว (Speed) และตำแหน่ง (Position) ของมอเตอร์ จึงทำให้อุปกรณ์ชนิดนี้ถูกเรียกชื่อตามบทบาทและหน้าที่ว่า Speed Sensor หรือไม่ก็เรียกว่า Shaft Position Sensor โดยประกอบด้วยชนิดต่างๆ ดังนี้



รูปที่ 2.35 การแยกประเภทของเอนโค้ดเดอร์

จากรูปที่ 2.35 เอนโค้ดเดอร์ (Encoder) สามารถแยกประเภทตามหลักการได้ 2 กลุ่มคือชนิดที่ทำงานโดยอาศัยหลักการเหนี่ยวนำหรือเรียกว่าอะนาลอก เอนโค้ดเดอร์ (Analog Encoder) และชนิดที่ทำงานโดยอาศัยหลักการดิจิตอล (Digital Encoder)

ชนิดที่ทำงานโดยอาศัยหลักการเหนี่ยวนำหรือเรียกว่า อะนาลอกจะประกอบไปด้วยเทคโนโลยีเจนเนอเรเตอร์(TachoGenerator) และรีโซลเวอร์ (Resolver) ส่วนชนิด ที่ทำงานโดยอาศัยหลักการออปติคอลล หรือแบบดิจิตอลจะแยกเป็นแบบ Incremental Encoder และ Absolute Encoder



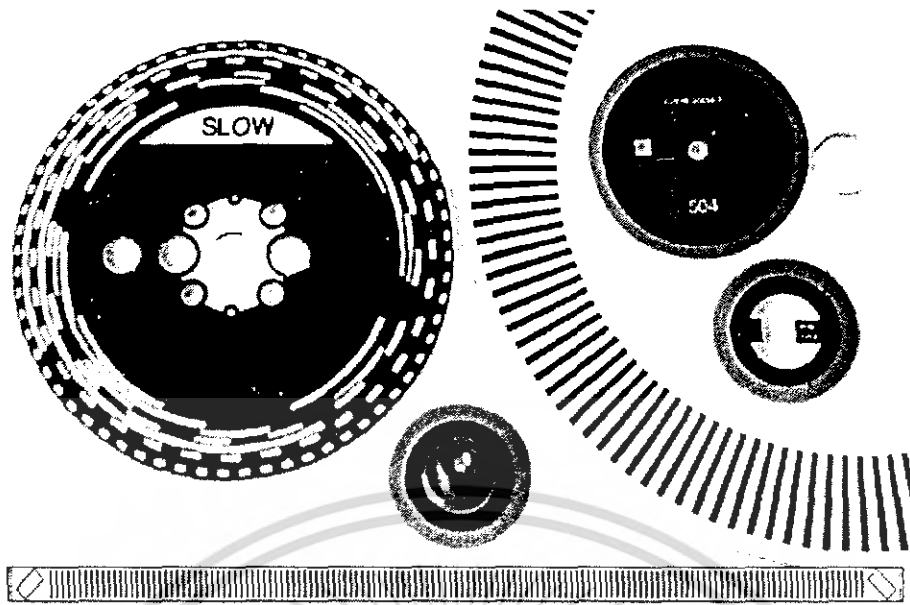
รูปที่ 2.36 โครงสร้างและสัญญาณของอินครีเมนทัลเอ็นโค้ดเดอร์

โครงสร้างจะประกอบด้วยตัวกำเนิดแสง, ตัวจับแสงซึ่งถูกคั่นกลางด้วยแผ่นจานกลมๆที่มีการทำรูเจาะไว้รอบๆแผ่น ( จำนวนรูจะขึ้นอยู่กับความละเอียดของ incremental encoder ) และ หน้ากากแยกช่องของสัญญาณพัลส์ A ,B และ Z

สัญญาณพัลส์ที่ได้มาจากเอ็นโค้ดเดอร์ชนิดนี้จะประกอบด้วย 3 แทรค คือ A ,B และ Z ดังรูปที่ 2.36

พัลส์ที่เกิดจาก แทรค A และ B จะเกิดการเลื่อนกัน มีความต่างเฟสกัน 90 องศา เพื่อทำหน้าที่รายงานผลของความเร็วและทิศทางการหมุนของมอเตอร์ให้คอนโทรลเลอร์ ดังนี้ กรณีพัลส์ A เกิดขึ้นก่อน B คอนโทรลเลอร์จะรับรู้ว่ามีมอเตอร์กำลังหมุนด้วยทิศทางตามเข็มนาฬิกา แต่ถ้าหากพัลส์ B เกิดขึ้นก่อน A คอนโทรลเลอร์จะรับรู้ว่ามีมอเตอร์กำลังหมุนด้วยทิศทางทวนเข็มนาฬิกา

ส่วนแทรค Z หรือพัลส์อ้างอิง จะเกิดขึ้น 1 พัลส์ในการหมุน 1 รอบ ทำหน้าที่ใช้อ้างอิงตำแหน่งของโรเตอร์



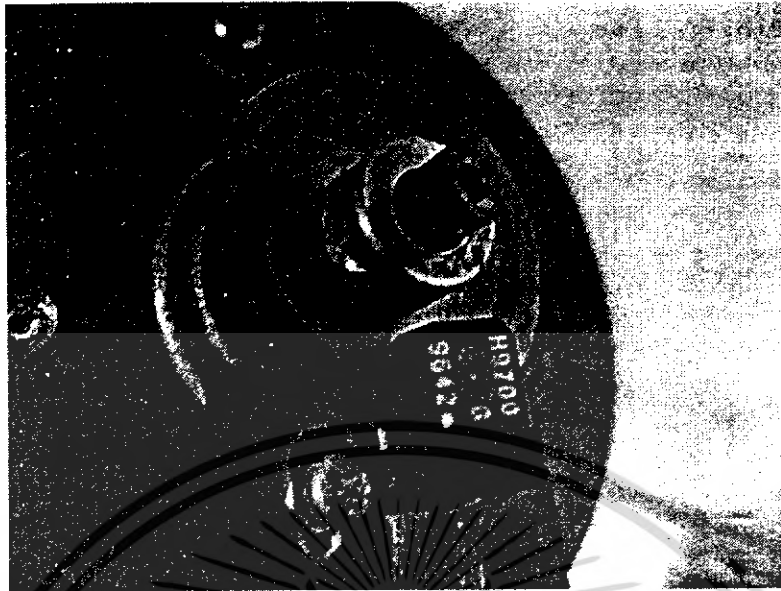
รูปที่ 2.37 ลักษณะชนิดของแผ่นจานหมุนที่แบ่งช่องให้แสงผ่าน

2.6.1 ความละเอียดของอุปกรณ์ตรวจจับตำแหน่งของโรเตอร์แบบ incremental ความละเอียดของเอนโค้ดเดอร์ คือ จำนวนคาบของสัญญาณเอาต์พุตต่อการหมุนของเพลลา 1 รอบ ซึ่งจะบอกมาเป็นจำนวนพัลส์ต่อรอบ หรือจำนวนไซเคิล ต่อ 360 องศา มุมทางกล หรือไซเคิล ต่อ องศา ของเอนโค้ดเดอร์ที่ใช้กันทั่วไป จะมีค่าความละเอียดตั้งแต่ 15 ถึง 10,000 พัลส์



รูปที่ 2.38 จานหมุนของเอนโค้ดเดอร์ที่ใช้ในโรงงานนี้ มีความละเอียด 400 พัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 การติดตั้งจานหมุนและตัวเซ็นเซอร์แสง

#### 2.6.2 เอาท์พุทของอุปกรณ์ตรวจจับตำแหน่งของโรเตอร์แบบ incremental

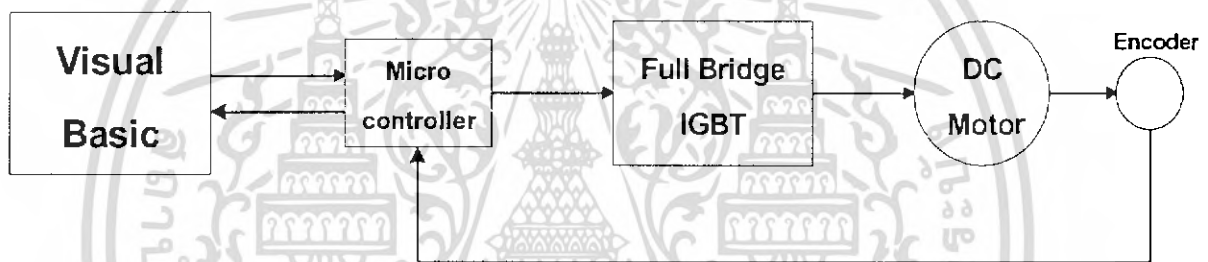
โดยทั่วไปแล้วสัญญาณเอาท์พุทที่ออกมาจากเอนโค้ดเดอร์โดยตรง มักจะมีระดับไม่เพียงพอในการควบคุมหรือประมวลผล ดังนั้นจึงต้องมีวงจรถ่าย และแปลงสัญญาณต่อไว้ในตัวเอนโค้ดเดอร์ด้วยเสมอ สัญญาณลูกคลื่นที่ได้รับจากตัวเซ็นเซอร์ปกติแล้วจะเป็นรูปสามเหลี่ยมหรือซายน์ จะเป็นอย่างไรขึ้นอยู่กับความละเอียดที่ต้องการ รูปสัญญาณเหล่านี้สามารถทำให้เปลี่ยนเป็นสัญญาณรูปสี่เหลี่ยมได้โดยการต่อ คอมพาราเตอร์เข้ากับลิเนียร์แอมพลิไฟร์ของเอนโค้ดเดอร์ จึงจะได้เอาท์พุทเป็นรูปสี่เหลี่ยมตามต้องการ

## บทที่ 3

### หลักการทํางานและการออกแบบวงจร

สำหรับในโครงงานนี้จะศึกษาเกี่ยวกับการควบคุมมอเตอร์ไฟฟ้ากระแสตรง โดยให้ไมโครคอนโทรลเลอร์เป็นตัวประมวลคำสั่ง ส่งออกไปที่ชุดขับซึ่งเป็นวงจร DC – DC CONVERTER FULL-BRIDGE ซึ่งใช้ไอจีบีที เป็นสวิตช์ ทั้งหมด 4 ตัว โดยเป็นโหมดการทํางานคือ 4 คอร์ตแตรนท์ โดยการทำงานที่ละคู่ เมื่อมอเตอร์เริ่มทํางานและเปลี่ยนโหมดตามต้องการ จะมีการป้อนกลับของระบบโดยใช้ตัว เอนโค้ดเตอร์ ป้อนกลับไปแสดงผลที่คอมพิวเตอร์

ในส่วนของคอมพิวเตอร์นั้น จะใช้โปรแกรม Visual Basic ในการรับคำสั่งจากผู้ใช้งาน เพื่อสั่งให้มอเตอร์ทํางานตามต้องการและรับการป้อนกลับจากมอเตอร์มาเพื่อแสดงค่าต่างๆ ที่จำเป็นต่อการทำงานหรือตรวจสอบประสิทธิภาพการทํางานของมอเตอร์ตัวนั้นๆ



รูปที่ 3.1 บล็อกไดอะแกรมการทํางานและการควบคุมโดยรวม

### 3.1 วงจรควบคุม

ในส่วนของการควบคุมทั้งหมดของระบบของโครงงานจะใช้รูปแบบของ การควบคุมป้อนกลับเบื้องต้น (Basic Feedback Control) ซึ่งรูปแบบนี้จะเป็นหลักการควบคุมที่นิยมใช้ในงานที่เกี่ยวกับมอเตอร์ และเป็นการควบคุมที่เหมาะสมกับวัตถุประสงค์ของโครงงาน ซึ่งในการควบคุมแบบป้อนกลับนี้ จะมีคุณลักษณะในการควบคุมต่างๆ ไปอยู่ 5 ลักษณะ คือ การควบคุมแบบ P, I, PI, PD และ PID ซึ่งรายละเอียดและคุณลักษณะของแต่ละชนิดจะอธิบายต่อไป

นอกจากนี้ยังมีวงจรควบคุมการทํางานอื่นๆ อีก คือ วงจรเดดไทม์ วงจรควบคุมการทํางานแต่ละตัวของไอจีบีทีในชุดวงจรกำลัง ซึ่งการควบคุมเหล่านี้จะต้องทํางานประสานกันเพื่อให้การทํางานของมอเตอร์และระบบทั้งหมดเป็นไปอย่างมีประสิทธิภาพ

#### 3.1.1 การควบคุมป้อนกลับเบื้องต้น (Basic Feedback Control)

ระบบควบคุมที่ตื้นนั้นจะต้องมีผลตอบสนองเวลาต่อสัญญาณด้านเข้าแบบต่าง ๆ มีเสถียร

ภาพที่ดี ดังนั้นต้องมีตัวควบคุมที่ดีที่สามารถทำให้ระบบสามารถทํางานให้มีสมรรถนะตามเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักศึกษาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการ โดยทั่วไปเราจะใช้ตัวควบคุมแบบตัวควบคุมชนิดพี (Proportional Control) ตัวควบคุมชนิดไอ (Integral Control) และตัวควบคุมชนิดดี (Derivative Control)

### คุณสมบัติของการควบคุมแบบป้อนกลับ (Characteristics of Feedback)

1. การควบคุมแบบป้อนกลับสามารถทำให้ระบบเกิดเสถียรภาพได้
2. การควบคุมแบบป้อนกลับยังสามารถทำให้สมรรถนะในช่วงทรานเซียนของระบบดีขึ้น
3. สามารถลดหรือกำจัดค่าผิดพลาดในสภาวะคงตัวได้
4. สามารถกำจัดสัญญาณรบกวนได้
5. สามารถลดความไวของระบบต่อการเปลี่ยนแปลงของพารามิเตอร์ได้

สมมุติถ้าระบบใด ๆ มีฟังก์ชันถ่ายโอนดังนี้

$$Y(s) = \frac{1}{s(s+1)}$$

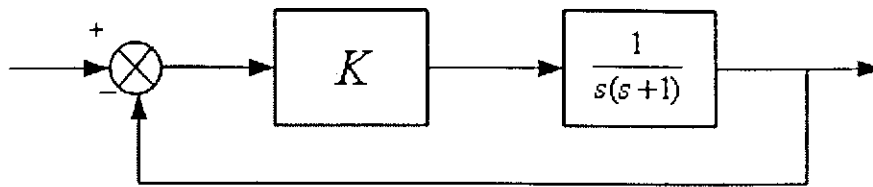
เมื่อทำการควบคุมระบบดังกล่าวด้วยวิธีแบบลูปเปิด



รูปที่ 3.2 การควบคุมแบบลูปเปิด

$$T(s) = \frac{K}{s(s+1)}$$

พบว่าระบบจะไม่เสถียร ไม่สามารถควบคุมได้ เนื่องจากเทอมอินทิกรัล ( $j\omega$ ) จากนั้นเมื่อใช้หลักการของการควบคุมแบบป้อนกลับ



รูปที่ 3.3 การควบคุมแบบลูปปิด

$$T(s) = \frac{\frac{K}{s(s+1)}}{1 + \frac{K}{s(s+1)}} = \frac{K}{s^2 + s + K}$$

จะเห็นได้ว่า เมื่อ  $K > 0$  ระบบจะเสถียรภาพ

จากการศึกษาจะเห็นได้ว่าหลักการควบคุมแบบป้อนกลับจะทำให้ระบบที่ไม่เสถียรสามารถเสถียรภาพได้ซึ่งจะมีผลทำให้ระบบดีขึ้น

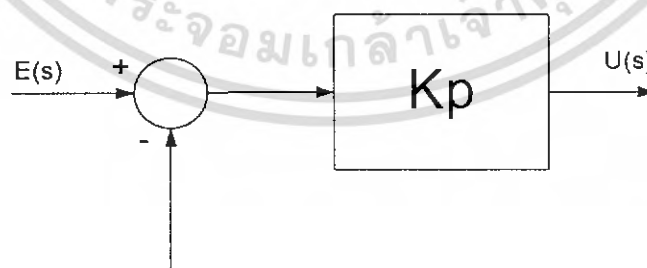
3.1.2 คุณสมบัติของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพี (P-controller)  
ตัวควบคุมนี้มีสัญญาณควบคุมออกมาเป็นสัดส่วนโดยตรงกับสัญญาณเข้า

หรือ

$$U(t) = K_p e(t)$$

$$U(s) = K_p E(s)$$

โดยค่า  $K_p$  คืออัตราขยายสัดส่วนมีค่าคงที่



รูปที่ 3.4 แผนภาพกรอบของตัวควบคุมชนิดพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.3 คุณสมบัติของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดไอ (I-controller)

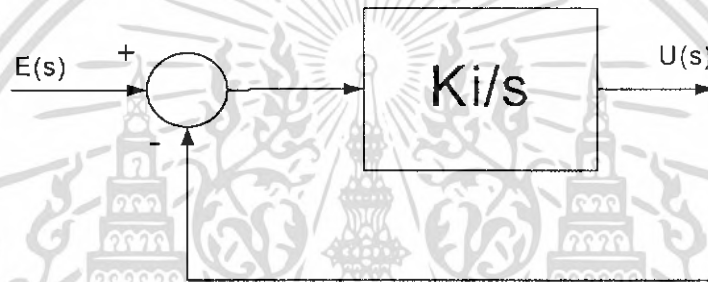
ตัวควบคุมชนิดนี้มีสัญญาณควบคุมออกมาด้วยอัตราที่เป็นสัดส่วนกับสัญญาณคลาดเคลื่อนเข้า

$$D U(t) = K_i e(t)$$

$$U(t) = K_i \int e(t) dt$$

โดย  $K_i$  คือค่าคงที่ที่ปรับค่าได้

$$U(s) = K_i/s \cdot E(s)$$



รูปที่ 3.5 แผนภาพกรอบของตัวควบคุมชนิดไอ

### 3.1.4 คุณสมบัติของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีไอ (PI-controller)

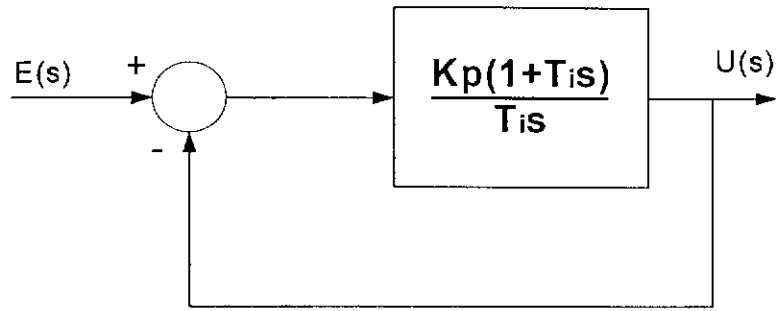
ตัวควบคุมแบบนี้เป็นการรวมตัวควบคุมแบบ P กับ I เข้าด้วยกัน และเป็นการควบคุมแบบที่โครงการนี้เลือกใช้

$$U(t) = K_p e(t) + (K_p/T_i) \int e(t) dt$$

เขียนเป็นฟังก์ชันถ่ายโอนได้

$$\frac{U(s)}{E(s)} = K_p(1+1/T_i s)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

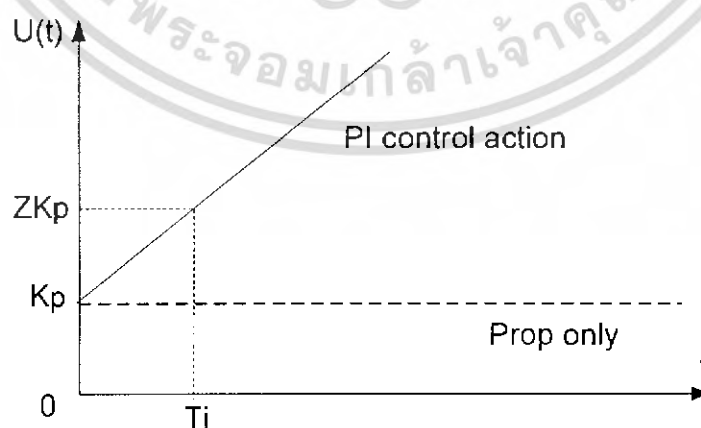


รูปที่ 3.6 การใช้ตัวควบคุมแบบ PI

โดย  $K_p$  คืออัตราขยายส่วน และ  $T_i$  เรียกว่าช่วงเวลาอินทิกรัล (Integral time) ค่า  $K_p$  และ  $T_i$  เป็นค่าคงที่ที่ปรับค่าได้ ส่วนกลับของค่า  $T_i$  เรียก reset rate ซึ่งหมายถึงจำนวนครั้งต่อวินาทีที่สัญญาณควบคุมมีค่าเป็น 2 เท่า



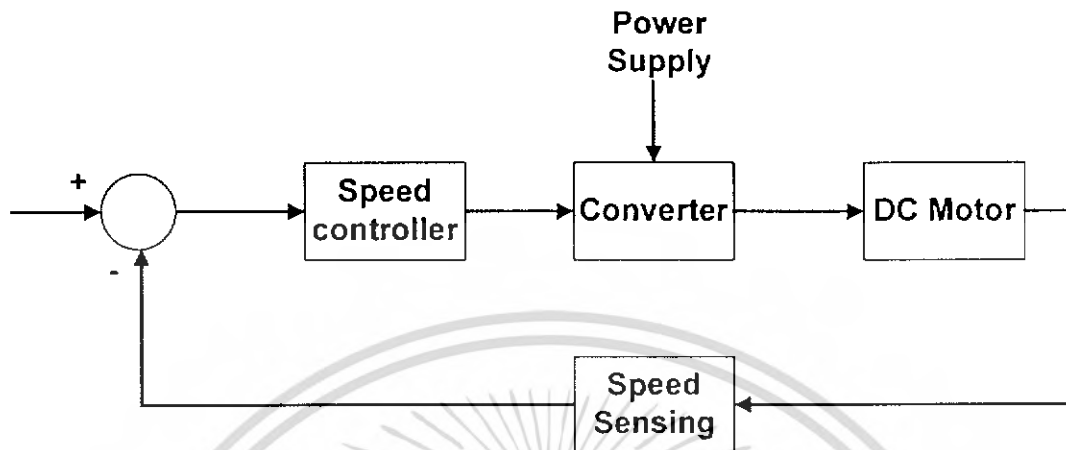
รูปที่ 3.7 สัญญาณเข้าฟังก์ชันหนึ่งหน่วย



รูปที่ 3.8 สัญญาณออกของตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการควบคุมของโครงการจะเป็นการรวมตัวควบคุมแบบ P เข้ากับแบบ I



รูปที่ 3.9 บล็อกไดอะแกรมของโครงการ

3.1.5 คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีดี (PD-controller) ตัวควบคุมแบบนี้เป็นการรวมตัวควบคุมแบบ P กับ D เข้าด้วยกัน

$$U(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

ฟังก์ชันถ่ายโอนคือ

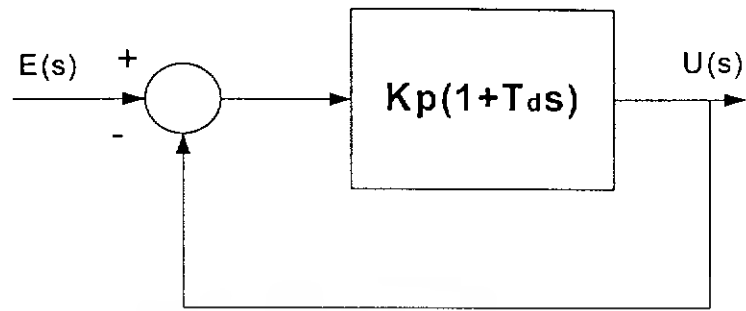
$$\frac{U(s)}{E(s)} = K_p(1+T_d s)$$

โดย  $K_p$  คืออัตราขยายส่วน และ  $T_d$  เรียกว่าช่วงเวลาอนุพันธ์ (Derivative time) ค่า  $K_p$  และ  $T_d$  เป็นค่าคงที่ที่ปรับค่าได้ การกระทำควบคุมแบบ D ให้สัญญาณออกมีค่าเป็นสัดส่วนกับอัตราเปลี่ยนแปลงของสัญญาณคลาดเคลื่อนเข้า

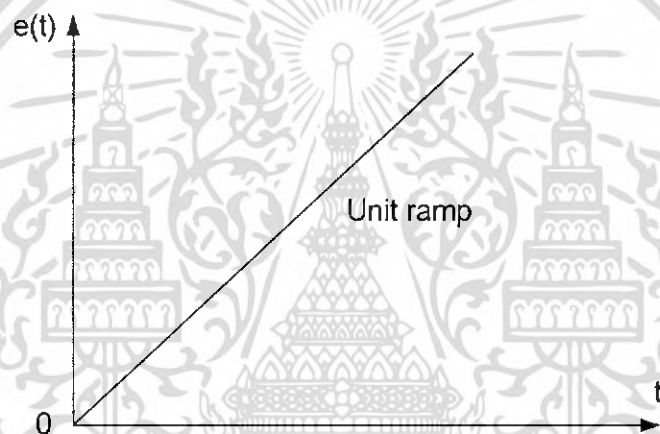
รูปที่ 3.10 แสดงแผนภาพกรอบของตัวควบคุม PD ถ้าสัญญาณคลาดเคลื่อนเข้ามีลักษณะเป็นแรมป์หนึ่งหน่วยดังรูป 3.11 จะได้สัญญาณควบคุมดังรูปที่ 3.12

จะสังเกตได้ว่าการกระทำควบคุมแบบ D ให้ผลต่อระบบกว่าแบบ P ซึ่งเป็นข้อดีของแบบ D แต่มีข้อเสียคือ ขยายสัญญาณรบกวนและอาจทำให้เกิดการอิ่มตัว (Saturation) ขึ้นในตัวขับเคลื่อน (Actuator) ได้

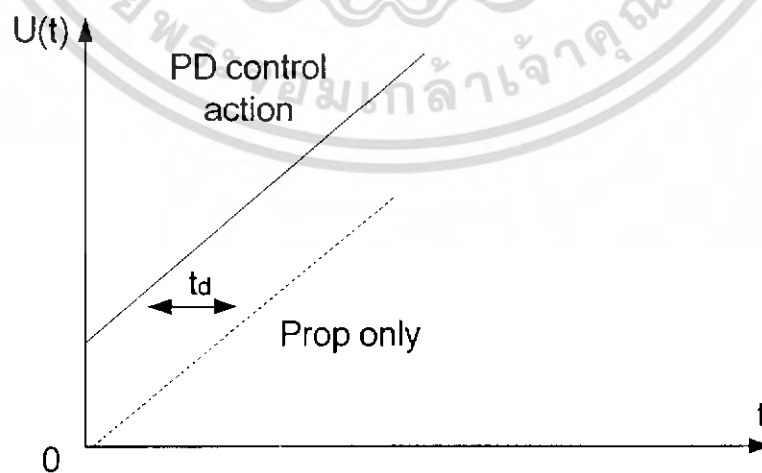
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การใช้ตัวควบคุมแบบ PD



รูปที่ 3.11 สัญญาณที่เข้าฟังก์ชันแรมป์หนึ่งหน่วย



รูปที่ 3.12 สัญญาณออกของตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

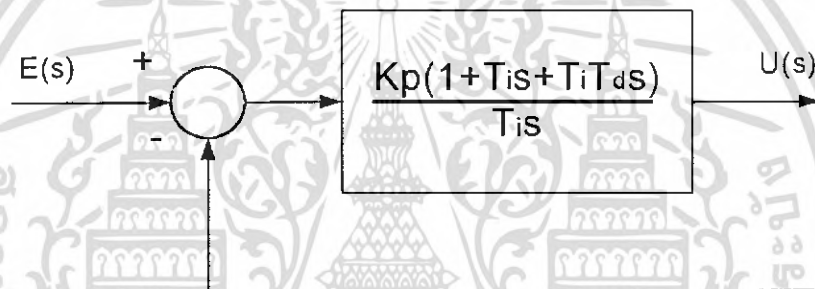
### 3.1.6 คุณสมบัติของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีไอดี (PID-controller)

$$U(t) = K_p \cdot e(t) + (K_p/T_i) \int e(t)dt + K_p T_d D e(t)$$

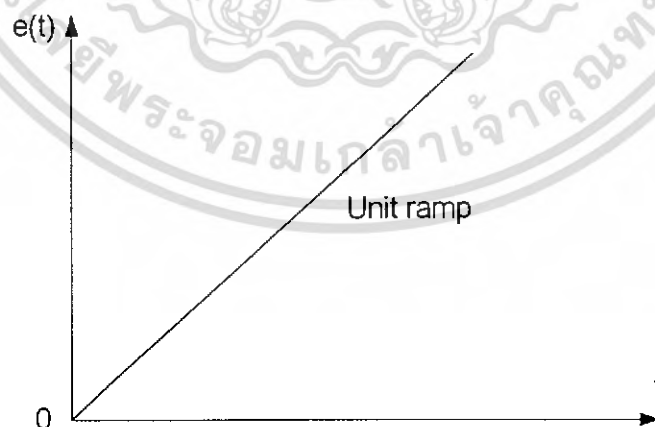
ฟังก์ชันถ่ายโอนคือ

$$\frac{U(s)}{E(s)} = K_p(1+1/T_i s + T_d s)$$

โดย  $K_p$  คืออัตราสัดส่วน  $T_i$  เรียกว่าช่วงเวลาอินทิกรัล และ  $T_d$  เรียกว่าช่วงเวลาอนุพันธ์

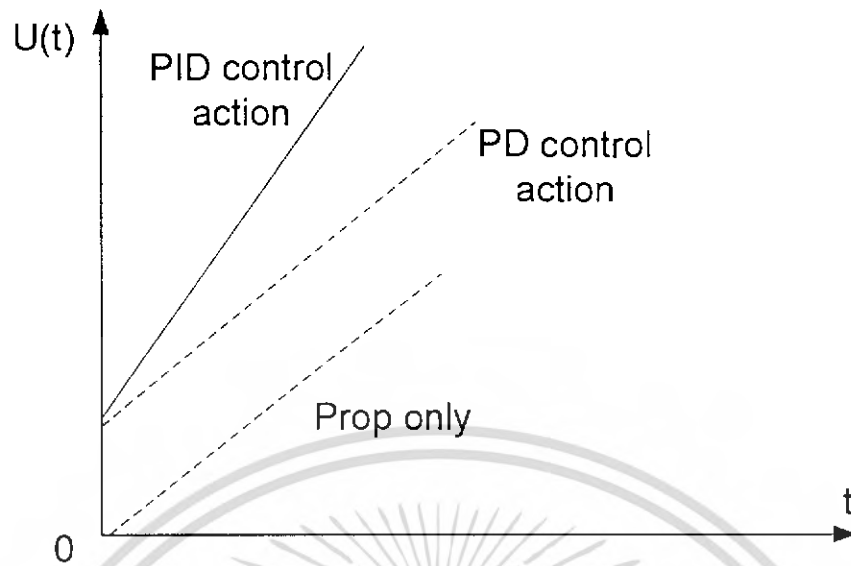


รูปที่ 3.13 การใช้ตัวควบคุมแบบ PID



รูปที่ 3.14 สัญญาณที่เข้าฟังก์ชันแรมปีหนึ่งหน่วย

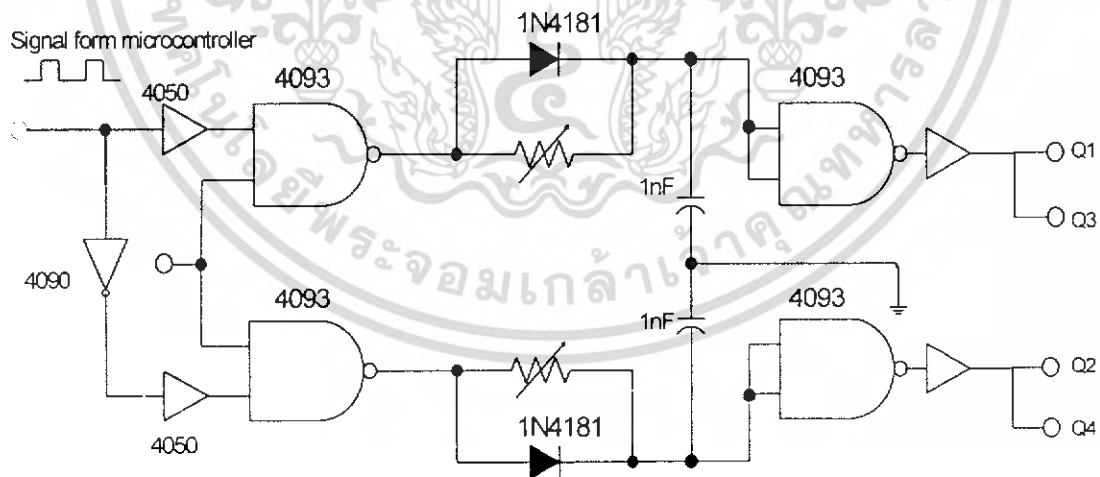
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 สัญญาณออกของตัวควบคุม

### 3.1.7 วงจรเดดไทม์

วงจรเดดไทม์มีไว้เพื่อป้องกันสวิตช์ในกิ่งเดียวกันทำงานพร้อมกัน ซึ่งหากเกิดลัดวงจรขึ้นจะทำให้สวิตช์เสียหาย และส่งผลไปถึงความเสียหายโดยรวมของระบบได้



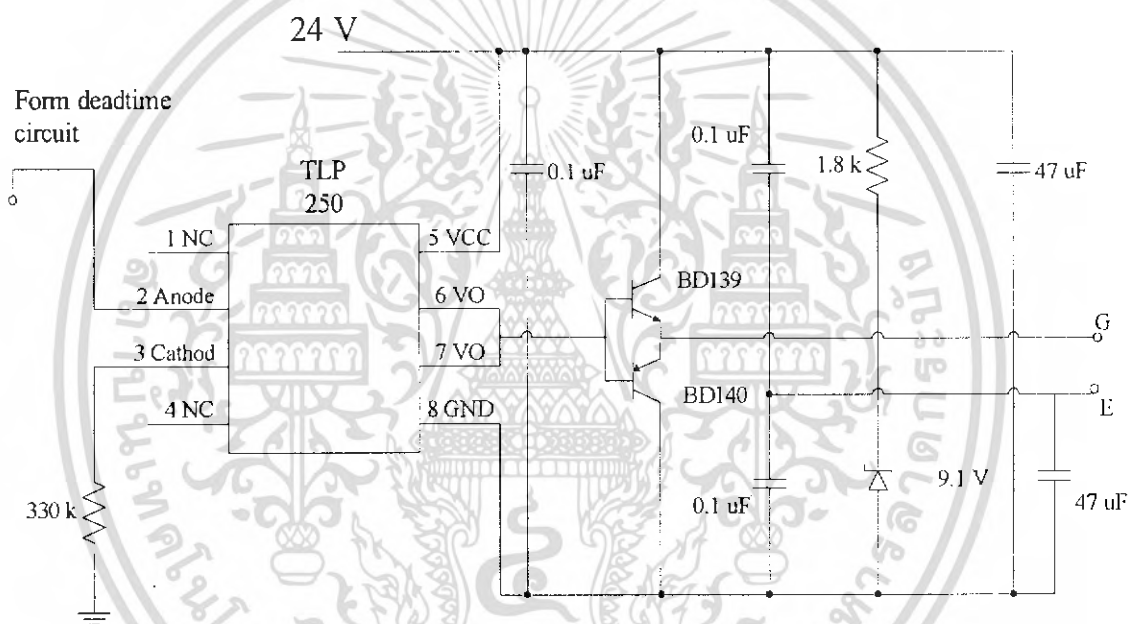
รูปที่ 3.16 วงจรเดดไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่าเดดไทม์สามารถกำหนดได้โดยใช้การปรับเปลี่ยนค่าความต้านทาน และค่าตัวเก็บประจุ แต่โดยทั่วไปนั้นนิยมใช้แบบความต้านทานปรับค่าได้ ดังนั้นในโครงงานนี้จึงใช้วิธีปรับค่าเดดไทม์โดยใช้การปรับที่ความต้านทานปรับค่าได้เช่นเดียวกัน โดยจะใช้ที่ค่าเวลาประมาณ 3  $\mu$ s

### 3.1.8 วงจรขับไอจีบีที

วงจรขับไอจีบีทีที่ใช้ในโครงงานนี้ตัวหลักอยู่ที่ ไอซีทีแอลพี 250 ซึ่งเป็น ออปโตคัปเปอเรอร์ ซึ่งไอซีทีแอลพี 250 ตัวนี้สามารถตอบสนองต่อความถี่ที่ต้องการได้ดี และสัญญาณเอาท์พุทที่ได้ สามารถนำไปขับได้เลย ไม่จำเป็นต้องมีวงจรปรับแต่งสัญญาณแต่อย่างใด



รูปที่ 3.17 วงจรชุดขับของไอจีบีที

จากวงจรที่แสดงในรูปที่ 3. จะเป็นเพียง 1 โหมต การทำงานเท่านั้น ซึ่งความเป็นจริงจะมีวงจรในลักษณะนี้ ทั้งหมด 4 วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

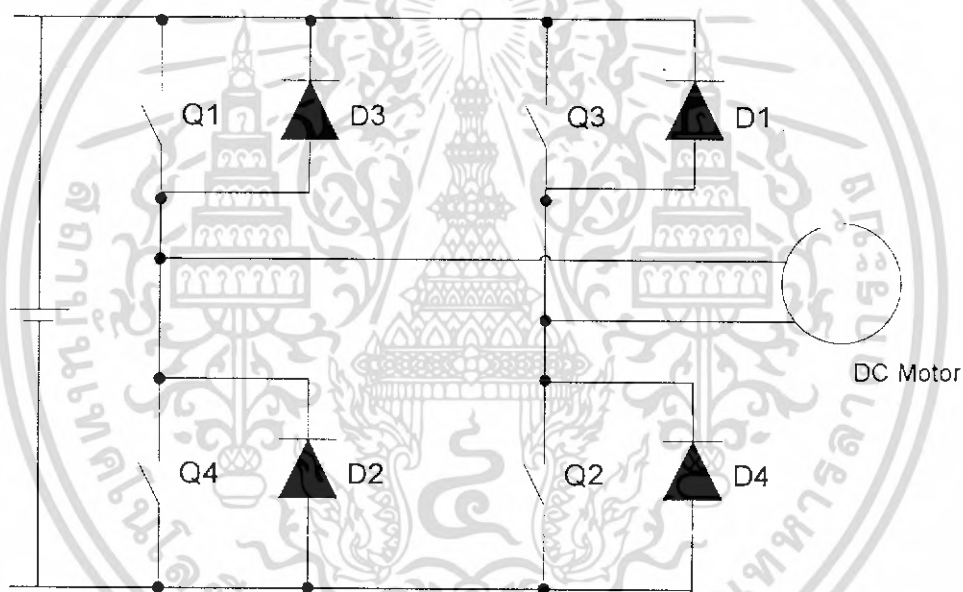
### 3.2 วงจรกำลัง

การต้องการให้มอเตอร์กระแสตรงทำงานได้ทั้ง 4 ควอเตอร์นั้น เพื่อที่จะทดสอบว่า การใช้ วิชาล เบสิก เป็นตัวสั่งและควบคุม จะทำงานได้อย่างสมบูรณ์ทุกๆ ยานการทำงานที่มอเตอร์สามารถทำงานได้หรือไม่

ดังนั้นโครงงานนี้จึงเลือกใช้การขับเคลื่อนมอเตอร์ แบบ DC - DC CONVERTER FULL - BRIDGE โดยใช้ ไอจีบีที เป็นตัวสวิตช์

#### 3.2.1 DC - DC CONVERTER FULL-BRIDGE

การทำงาน 4 ควอเตอร์ของมอเตอร์ไฟฟ้ากระแสตรงด้วยการควบคุมของวงจร DC - DC CONVERTER FULL-BRIDGE สามารถกระทำได้ดังวงจรรูปที่ 3.18



รูปที่ 3.18 การควบคุม 4 ควอเตอร์

จากรูปที่ 3.18 แสดงวงจรถูกกำลังที่จะนำไปขับมอเตอร์ ใช้รูปแบบของวงจร DC - DC CONVERTER FULL-BRIDGE โดยใช้ ไอจีบีทีเป็น ตัวสวิตช์

โดยทั่วไปแล้วการควบคุมสามารถทำได้หลายวิธีด้วยกัน เช่น การให้สวิตช์ Q ตัวใดตัวหนึ่งทำงานอย่างต่อเนื่อง หรือว่า การทำงานเป็นคู่ๆ สลับ กับไดโอด และการควบคุมกระแสที่ไหลอีก ซึ่งสามารถประยุกต์ใช้ได้หลายวิธี

แต่ในโครงงานนี้จะขออธิบายในวิธีที่ใช้ในการควบคุมโครงงานจริง ซึ่งคือวิธี การควบคุมให้ Q1 และ Q2 ทำงานเป็นคู่ พร้อมกับไดโอด D1 และ D2

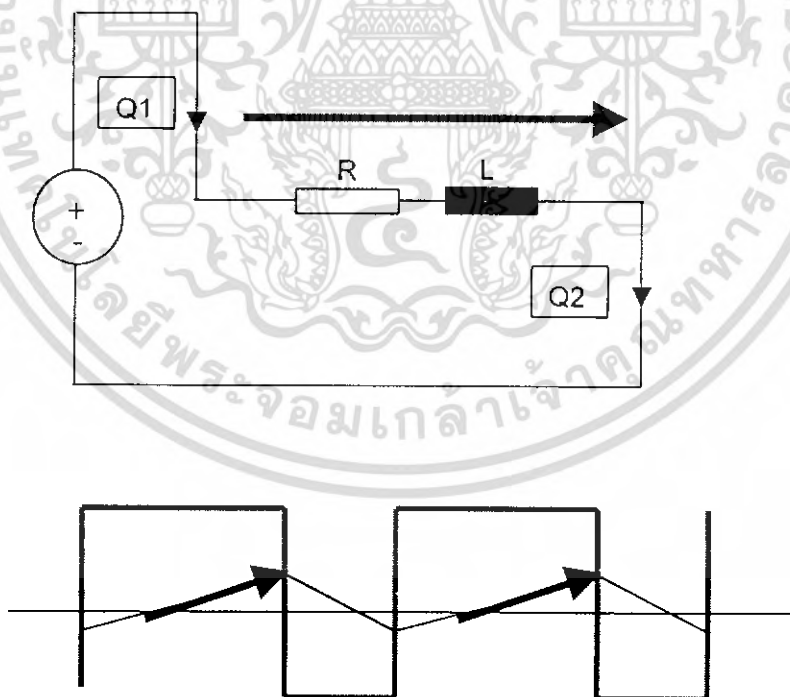
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทํางานคือ เมื่อเราใช้สวิตช์ Q1 และ Q2 ทํางานเป็นคู่ พร้อมด้วยไดโอด D1 และ D2 วงจรดังกล่าวจะให้กระแสที่เป็นบวกและแรงดันที่ปรับค่าได้ในแต่ละทิศทาง ดังนั้นจึงเป็นการควบคุมมอเตอร์ใน คอร์ดแตรนที่ 1 และ 4 ส่วนสวิตช์ Q3 และ Q4 พร้อมด้วยไดโอด D3 และ D4 จะเป็นอีกวงจรหนึ่งที่ให้กระแสเป็นลบและให้แรงดันที่ปรับค่าได้ในแต่ละทิศทาง ดังนั้นจึงหมายถึงการควบคุมใน คอร์ดแตรนที่ 2 และ 3 นั้นเอง

การเปลี่ยนการทํางานจากคอร์ดแตรนที่ 1 ไปเป็น คอร์ดแตรนที่ 3 สามารถกระทำได้โดย ใช้ขั้นตอนดังต่อไปนี้

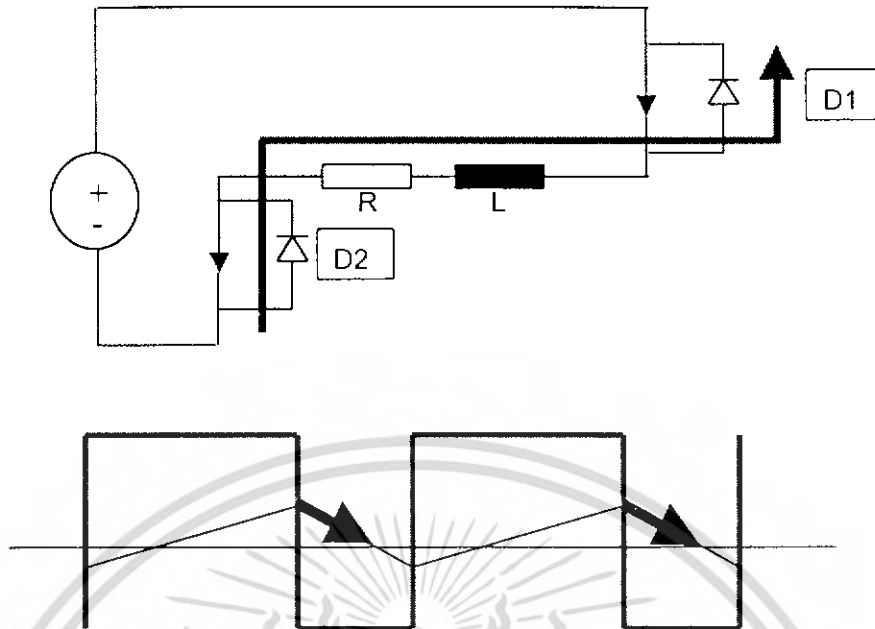
ใน คอร์ดแตรนที่ 1 สวิตช์ Q1 และ Q2 จะถูกควบคุมด้วย  $0.5 < \delta < 1.0$  ในช่วงนี้ สวิตช์ Q1 และ Q2 จะถูกสั่งให้ถูกทํางาน กระแสอาร์เมเจอร์ไหลผ่านไดโอด D1 แหล่งจ่าย V และไดโอด D2 กระแสจะตกลงสู่ศูนย์อย่างรวดเร็ว แรงดันไฟฟ้าต้านกลับของมอเตอร์จะมีขั้วทางด้านซ้ายมือเป็นบวก หลังจากนั้นสวิตช์ Q3 และ Q4 จะถูกควบคุมด้วย  $0 < \delta < 0.5$  แต่โดยทั่วไปจะใช้ที่ประมาณใกล้เคียงกับ 0.5 กระแสมอเตอร์จะลดลงที่แรงบิดสูงสุดและอัตราเร็วจะค่อยๆ ตกลงเป็นศูนย์

หลังจากนั้นปรับค่า  $\delta$  ให้สอดคล้องกับอัตราเร็วที่ต้องการ  $0.5 < \delta < 1.0$  มอเตอร์มีอัตราเร่งที่เพิ่มขึ้นที่แรงบิดสูงสุด และเป็นการหมุนกลับทิศทางตามที่ต้องการ

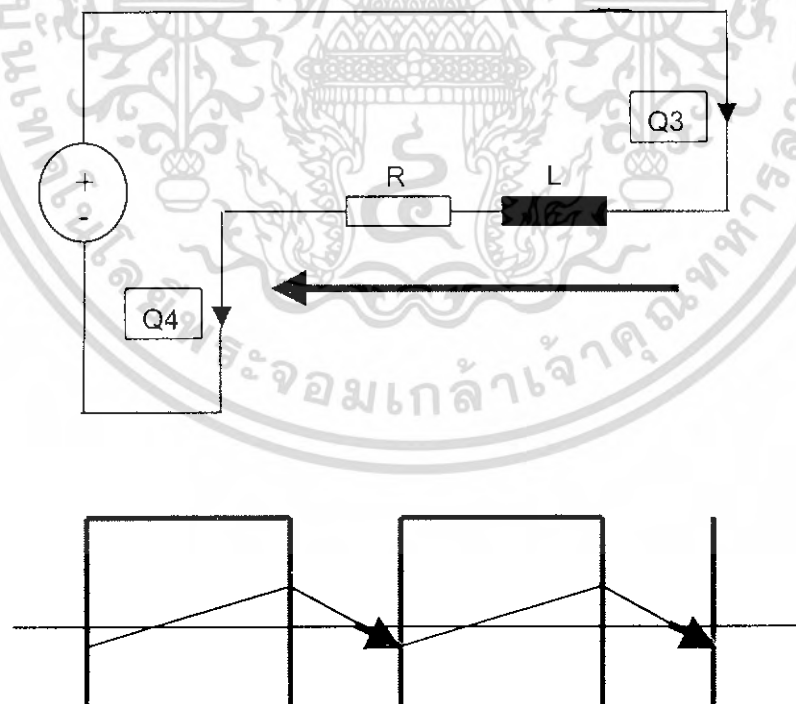


รูปที่ 3.19 ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q1 และ Q2 ทํางาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

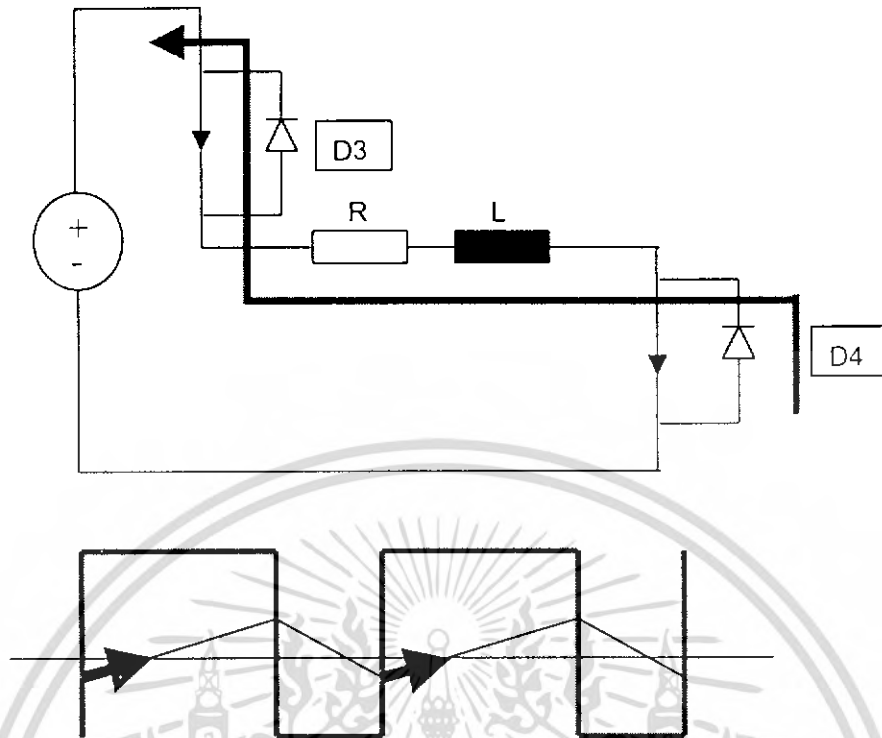


รูปที่ 3.20 ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q3 และ Q4 ทำงาน พร้อม D1 และ D2



รูปที่ 3.21 ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q3 และ Q4 ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

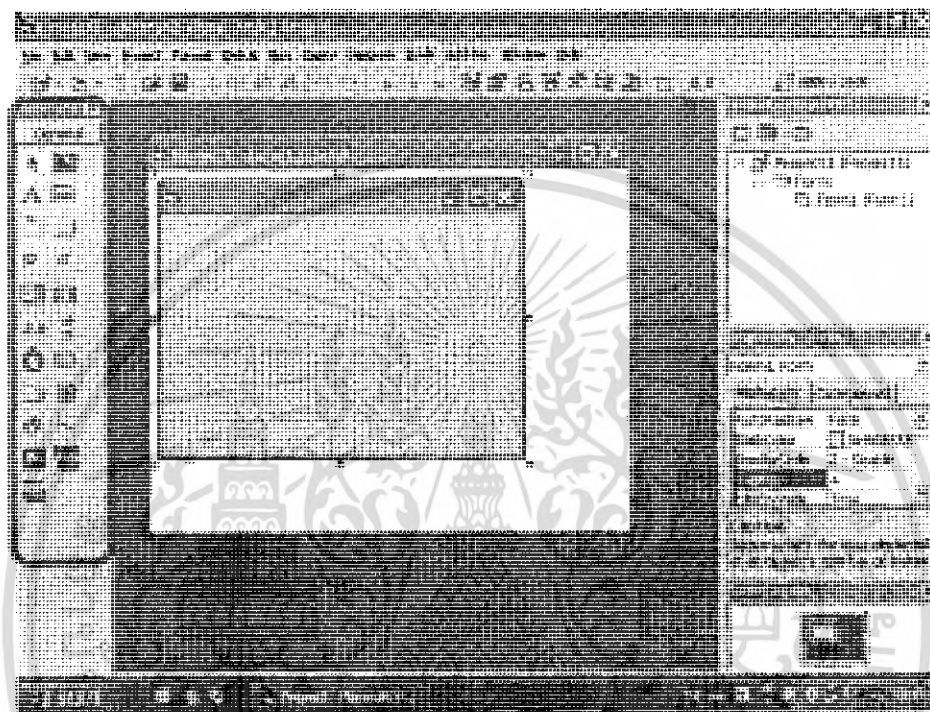


รูปที่ 3.22 ทิศทางการไหลของกระแสเมื่อ สวิตช์ Q1 และ Q2 ทำงาน พร้อม D3 และ D4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การออกแบบในส่วนของโปรแกรม วิชาล เบสิก

ในส่วนของการออกแบบหน้าต่างของโปรแกรมวิชาล เบสิก ซึ่งเป็นส่วนที่สำคัญที่จะใช้ในการออกคำสั่งรวมถึงการแสดงผลที่ได้กระทำให้เราเห็นนั้น จะใช้ตัวอุปกรณ์ในการออกแบบเป็นแบบพื้นฐานโดยทั่วไป ซึ่งมีบรรจุอยู่แล้วในแถบเครื่องมือ (Tool Box) บนหน้าต่างของโปรแกรมวิชาล เบสิก



รูปที่ 3.23 แถบเครื่องมือ (Tool Box) และอุปกรณ์พื้นฐานบนหน้าต่างโปรแกรม

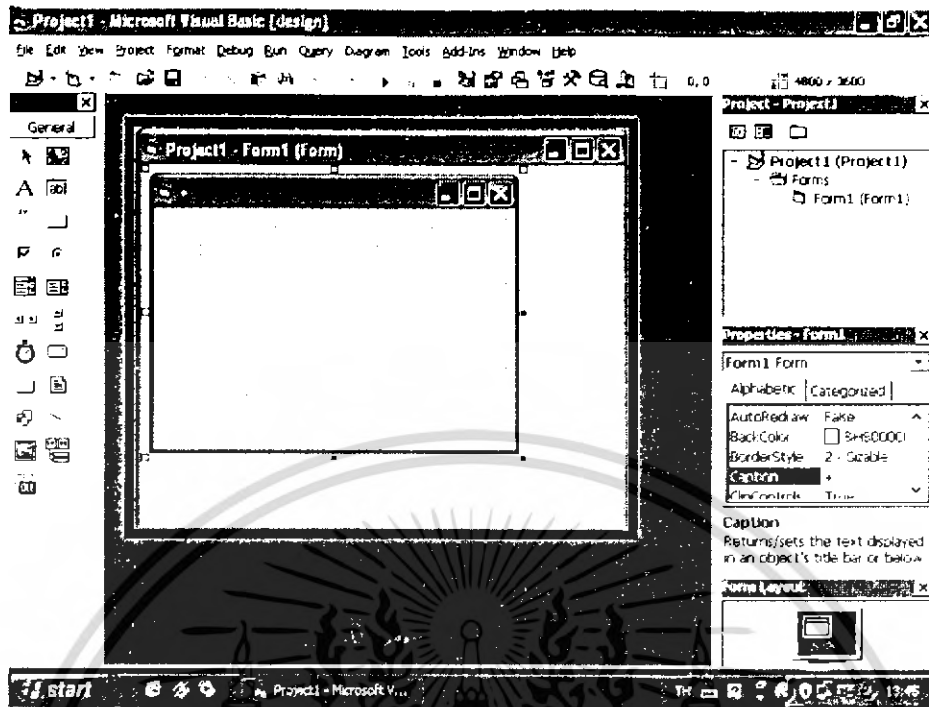
จากวัตถุประสงค์หลักที่ต้องการจะส่งคำสั่งที่สั่งมอเตอร์ในลักษณะต่างๆ และรวมถึงรับค่าการกระทำของมอเตอร์ตั้งที่ได้กล่าวมาแล้วนั้น จากการทำโครงการก็จะพบว่ามียุอุปกรณ์และเครื่องมือหลักๆ ที่อยู่ในแถบเครื่องมือของ วิชาล เบสิก ที่เลือกมาใช้งานเพื่อให้ตรงวัตถุประสงค์ดังต่อไปนี้

#### 3.4.1 Form Object

Form Object เป็นอ็อบเจกต์แรกสุดที่โปรแกรม วิชาล เบสิก ติดตั้งให้เองโดยอัตโนมัติ และเป็นอ็อบเจกต์ที่ไม่ได้อยู่บน Tool Box แต่จะนำขึ้นมาหลังจากได้เรียกโปรแกรม วิชาล เบสิก ขึ้นมาทำงานเป็นที่เรียบร้อยแล้ว

Form Object ก็จะประกอบไปด้วย คุณสมบัติ (Property), กฎเกณฑ์ (Method), และ เหตุการณ์ (Event) ต่างๆ ที่ช่วยให้เราเขียนโปรแกรมเพื่อควบคุมการทำงานของฟอร์มตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 ฟอรัมที่ปรากฏขึ้นมาเมื่อเริ่มเปิดโปรแกรม

ฟอรัมเป็นเหมือนกับแผ่นกระดาษเปล่าๆ แผ่นหนึ่งที่จะแต่งเติมอะไรลงไปก็ได้ ไม่ว่าจะเป็นข้อความ ช่องกรอกข้อมูล ปุ่มกด หรือการแสดงรูปหรือผลการทำงานต่างๆ ซึ่งสิ่งต่างๆ เหล่านี้ จะเป็นสิ่งที่เป็นส่วนโต้ตอบหรือแสดงผลกับผู้ใช้โปรแกรม ดังนั้นการทำงานบนโปรแกรม วิชาการ เบสิก อันดับแรกต้องเป็นส่วนของฟอรัมเสมอ

### 3.3.2 Frame Object

Frame Object เป็นอ็อบเจกต์สำหรับจัดกลุ่มของอ็อบเจกต์ต่างๆ ที่เราออกแบบไว้บนฟอรัม เพื่อให้ดูเป็นหมวดหมู่ และมีชื่อกำกับโดยไม่จำเป็นว่าในเฟรมนั้นจะมีอ็อบเจกต์กี่ตัว อีกประการคือเพื่อที่เวลาเกิดความผิดพลาด การแก้ไข หรือย้ายตำแหน่งของอ็อบเจกต์ต่างๆ ที่ได้ออกแบบไว้ก็จะกระทำได้ง่าย

การเลือก Frame Object ออกมาใช้งานนั้นสามารถลากออกมาจาก แถบเครื่องมือพื้นฐาน (Tool Box) มาวางบนฟอรัมและลากกำหนดขนาดได้ตามต้องการ

พรอเพอร์ตี้ที่สำคัญสำหรับ Frame Object ประกอบด้วย Caption, Font และ ForeColor, BackColor และ BorderStyle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Caption :** เป็นพรอเพอร์ตี้สำหรับเก็บข้อความเอาไว้แสดงเป็นหัวข้อของ เฟรม ตามปกติแล้ว ถ้าไม่มีการเปลี่ยนแปลงจะเป็นข้อความที่ชื่อ Frame1 แต่ถ้าเราต้องการเปลี่ยนจาก Frame1 ก็ให้พิมพ์ข้อความลงไปในพรอเพอร์ตี้ Caption ที่ Property Box

**Font and ForeColor :** เป็นพรอเพอร์ตี้สำหรับปรับแต่งตัวหนังสือที่แสดงบนพรอเพอร์ตี้ Caption อย่างเดียว

**BackColor :** เป็นพรอเพอร์ตี้สำหรับกำหนดสีจากหลังแก่ เฟรม ที่สามารถกำหนดจาก Property Box หรือจากโปรแกรม

**BorderStyle :** เป็นพรอเพอร์ตี้สำหรับกำหนดเส้นขอบของ เฟรม โดยจะมีค่าเป็นตัว เลข คือ 0 หรือ 1 ที่สามารถกำหนดจาก Property Box ถ้าเลือก 0 จะไม่แสดงเส้นขอบของเฟรม รวมทั้งข้อความที่แสดงที่ Caption แต่ถ้าเลือก 1 จะแสดงเส้นขอบตามปกติ

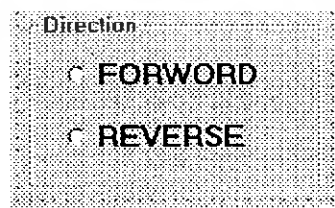
### 3.4.3 Option Object

Option Object เป็นอ็อบเจกต์ที่มีลักษณะเป็นปุ่มสำหรับเลือกอย่างใดอย่างหนึ่ง ดังนั้น Option จะต้องมีย่างน้อย 2 ตัวหรือมากกว่า ถ้า Option ตัวใดถูกเลือก ตัวที่เหลือก็จะถูกเลือกไม่ได้อีกโดยอัตโนมัติ

การเลือก Option Object ออกมาใช้งานนั้นสามารถลากออกมาจาก แถบเครื่องมือพื้นฐาน (Tool Box) ทั่วไปจะใช้ร่วมกับ Frame ที่ถูกวางบนฟอร์มอยู่ก่อนแล้ว โดยการกำหนดชื่อของสิ่งที่ต้องการให้เลือก

ค่าของ Option จะมีค่าเป็น True หรือ False เท่านั้น จะเป็น True เมื่อเราคลิกที่ปุ่มกลมจนมีจุดดำอยู่ที่กึ่งกลางวงกลม และจะเป็น False โดยอัตโนมัติเมื่อไปคลิกที่ปุ่มกลมอันอื่น

โดยในโครงการนี้ใช้ Option ร่วมกับ Frame ในการเลือกทิศทางหมุนของมอเตอร์ ซึ่งจะมีให้เลือก 2 ทิศทาง คือใช้ 2 Option คือ FORWARD และ REVERSE โดยอยู่ใน Frame ที่มีชื่อว่า Direction ดังรูปที่ 3.25



รูปที่ 3.25 การเลือกทิศทางหมุนของมอเตอร์โดยใช้ Option ร่วมกับ Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.4 Text Box Object

Text Box เป็นอ็อบเจกต์สำหรับบันทึกข้อความในรูปแบบตัวอักษรขณะที่โปรแกรมทำงาน ผู้ใช้งานสามารถจะพิมพ์ข้อความหรือตัวเลข ลงใน Text Box ได้

การเลือก Text Box ออกมาใช้งานนั้นสามารถลากออกมาจาก แถบเครื่องมือพื้นฐาน (Tool Box) ทั่วไปจะใช้ร่วมกับ Frame ที่ถูกวางบนฟอร์มอยู่ก่อนแล้วเช่นเดียวกับ Option Object โดยการกำหนดหัวข้อที่ Frame ที่ต้องการจะให้ใส่ข้อความหรือตัวเลขลงใน Text Box



รูปที่ 3.26 Text Box สำหรับใส่ค่าความเร็วรอบที่ต้องการโดยวางอยู่ใน Frame

โดยในโครงการนี้ใช้ Text Box ร่วมกับ Frame ในการใส่ค่าความเร็วรอบที่ต้องการลงไป เพื่อให้มอเตอร์หมุนตามรอบที่ต้องการ โดยจะมีพรอเพอร์ตี้ที่ใช้ในโครงการดังนี้

**BorderStyle :** เป็นพรอเพอร์ตี้ที่เกี่ยวกับการตีกรอบของอ็อบเจกต์ที่ใช้ ซึ่งมีใช้ในอ็อบเจกต์ต่อไปนี้คือ Form, Picture, Text Box และ Label โดยค่าของ BorderStyle จะมีอยู่ 2 ค่าคือ 0 และ 1 ถ้าเลือก 0 จะไม่แสดงเส้นกรอบ ถ้าเป็น 1 จะแสดงเส้นกรอบ

**Enable :** เป็นพรอเพอร์ตี้สำหรับให้อ็อบเจกต์นั้นทำงานได้หรือไม่ โดยมีค่าเป็น True หรือ False ถ้าเป็น True ก็ทำงานเป็น False ก็ไม่ทำงาน พรอเพอร์ตี้นี้จะใช้มากในอ็อบเจกต์ Timer แต่ในส่วนของ Text Box ก็มีความสำคัญคือถ้าเป็น False โปรแกรมก็จะไม่สามารถนำค่าตัวเลขหรือข้อความที่ใส่ไปกระทำต่อได้ จะเป็นเพียงแค่การแสดงให้เห็นว่าใส่อะไรลงไปเท่านั้น

**Name :** เป็นพรอเพอร์ตี้ที่ใช้ในเกือบทุกอ็อบเจกต์ โดยปกติใน Text Box จะแสดงคำว่า Text1 หากต้องการเปลี่ยน ไปเปลี่ยนได้ที่พรอเพอร์ตี้ Name ที่ Property Box ซึ่งการเปลี่ยนนั้นจะเปลี่ยนเป็นไม่มีข้อความเลยเพื่อที่จะรอข้อความที่ผู้ใช้งานจะใส่ลงไปก็ได้ แต่หากข้อความที่จะใส่ลงไปเป็นตัวเลข ให้เปลี่ยนจากเดิมเป็นเลข 0

### 3.3.5 Label Object

Label Object เป็นอ็อบเจกต์ ที่ใช้ในการนำเสนอข้อความบนฟอร์มคล้ายกับ Text Box แต่ว่า Label จะไม่สามารถรับข้อมูลที่ต้องการใส่ลงไปได้ มันจะแสดงข้อความหรือตัวเลขที่ใส่ไว้ตอนเขียนโปรแกรมหรือข้อมูลที่ไ้จากการกระทำของโปรแกรมมาแสดงให้เห็น

การเลือก Label Object ออกมาใช้งานนั้นสามารถลากออกมาจาก แถบเครื่องมือพื้นฐาน (Tool Box) มาวางบนฟอร์มและลากกำหนดขนาดได้ตามต้องการ  
 พรอเพอร์ตี้ที่สำคัญของ Label มีดังต่อไปนี้ คือ

*Caption* : มีหน้าที่กำหนดข้อความที่ใช้บรรยายอ็อบเจกต์นั้น ซึ่งสามารถจะบรรยายได้ใน Property Box ที่ Caption ของ Label ข้อความที่พิมพ์ลงไปก็จะปรากฏใน Label นั้นๆ

*Alignment* : เป็นคำสั่งสำหรับจัดตัวอักษรให้ชิดซ้าย ชิดขวา หรือกึ่งกลาง เมื่อคลิกที่พรอเพอร์ตี้ Alignment ใน Property Box ก็จะพบตัวเลือก 3 ตัวเลือก คือ ถ้าเป็น 0 จะชิดซ้าย 1 จะชิดขวา และ 2 จะอยู่ตรงกลาง

*BorderStyle* : เป็นพรอเพอร์ตี้ที่เกี่ยวกับการตีกรอบของอ็อบเจกต์ที่ใช้ ซึ่งมีใช้ในอ็อบเจกต์ต่อไปนี้คือ Form, Picture, Text Box และ Label โดยค่าของ BorderStyle จะมีอยู่ 2 ค่าคือ 0 และ 1 ถ้าเลือก 0 จะไม่แสดงเส้นกรอบ ถ้าเป็น 1 จะแสดงเส้นกรอบ

สำหรับในโครงการนี้การทำงานของ Label จะขึ้นอยู่กับเหตุการณ์ที่เกิดขึ้นตามที่เราต้องการ คือค่าที่ Label แสดงจะเป็นค่าที่เกิดขึ้นจากการทำงานของระบบไปแล้ว ที่เห็นได้ชัดคือค่าความเร็วรอบของมอเตอร์ที่จับมาจากการใช้เอนโค้ดเดอร์แล้วมาแสดงที่คอมพิวเตอร์ ก็จะแสดงผ่านที่ตัว Label นี้

#### 3.4.6 Command Button Object

Command Button Object เป็นอ็อบเจกต์ที่มีลักษณะเป็นปุ่มสี่เหลี่ยม ออกแบบมาเพื่อให้ใช้เมาส์คลิกตัวหนังสือที่อยู่บนปุ่มที่คลิก

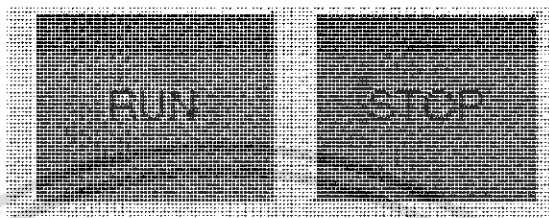
พรอเพอร์ตี้ของ Command Button ที่สำคัญได้แก่ พรอเพอร์ตี้ที่ใช้ในการแก้ไขเปลี่ยนแปลงข้อความบนปุ่ม พรอเพอร์ตี้ที่ใช้ในการเปลี่ยนรูปแบบตัวอักษรที่แสดงข้อความ รวมทั้งพรอเพอร์ตี้ที่ใช้ในการเปลี่ยนสีของ Command Button ซึ่งมีดังต่อไปนี้

*Caption* : มีหน้าที่กำหนดข้อความต่างๆ บนปุ่ม โดยปกติจะมีค่าชื่อเป็น Command Button การเปลี่ยนแปลงสามารถทำได้ที่ Property Box ที่ Caption ของ Command Button ข้อความที่พิมพ์ลงไปก็จะปรากฏใน Command Button นั้นๆ

*Font and ForeColor* : เป็นพรอเพอร์ตี้สำหรับปรับแต่งตัวหนังสือที่แสดงบนพรอเพอร์ตี้ Caption สามารถปรับปรุงขนาดและสีของตัวหนังสือได้

*Style and BackColor* : เป็นพรอเพอร์ตี้ที่ใช้ในการเปลี่ยนสีของ Command Button โดยเราจะใช้พรอเพอร์ตี้ 2 ตัวในการเปลี่ยนสี เริ่มจากการไปที่ พรอเพอร์ตี้ Style ซึ่งจะมีค่าให้เลือก ระหว่าง 0 คือไม่ปรับเปลี่ยน กับ 1 คือการปรับเปลี่ยน เลือก 1 แล้วไปที่ พรอเพอร์ตี้ที่ชื่อ BackColor แล้วทำการเลือกสีของ Command Button ตามต้องการ

ในโครงการนี้ใช้ Command Button เป็นตัวปุ่มกด RUN และ STOP การทำงานของระบบ ดังแสดงในรูปที่ 3.27



รูปที่ 3.27 ปุ่มกด RUN และ STOP โดยใช้ Command Button Object

### 3.3.7 Picture Object

Picture Object เป็นอ็อบเจกต์ที่ใช้แสดงรูปภาพบนโปรแกรมวิซวล เบสิก ซึ่งรูปภาพนั้นจะเกิดจากการสร้างขึ้นหรือเป็นไฟล์รูปภาพก็ได้

ในโครงการนี้จะใช้อ็อบเจกต์นี้ในลักษณะของกราฟแสดงการทำงานของค่าความเร็วรอบ โดยเราจะลากเส้นทำเป็นตารางกราฟและกำหนดจุดที่เส้นกราฟสามารถจะวิ่งไปได้ ให้ได้มากที่สุด และนำกราฟที่สร้างจาก Picture Object ไปวางไว้ใน Frame

โดยจะมีคำสั่งที่สำคัญที่ใช้ในการเขียนโปรแกรม ดังนี้

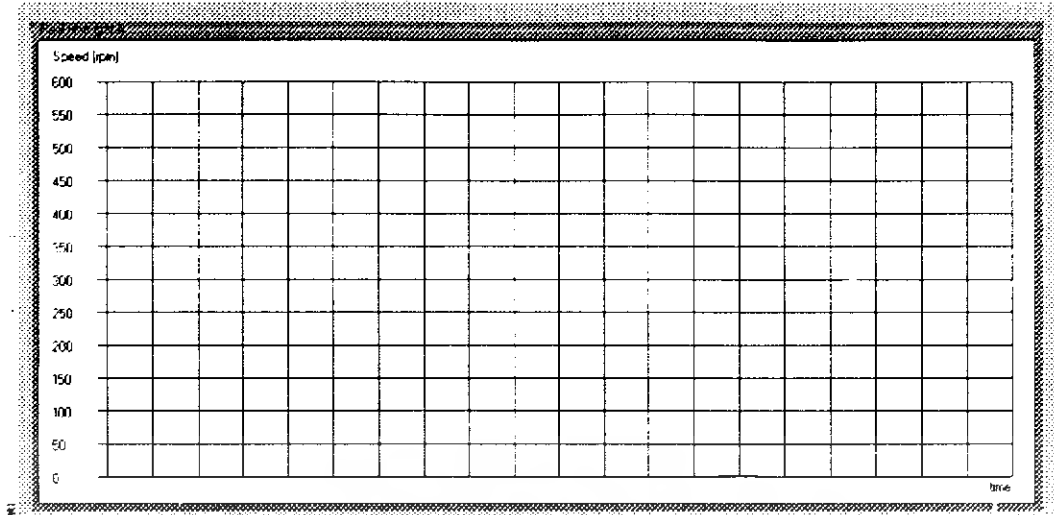
*Picture.Line (X1,Y1)-(X2,Y2),สี* : เป็นคำสั่งสำหรับลากเส้นตรงพร้อมทั้งกำหนดสีให้กับเส้นที่ลากด้วย ถ้าไม่กำหนดสี ถือว่าเป็นสีดำ

*Picture.Line -(X1,Y1)* : เป็นคำสั่งสำหรับลากเส้น โดยลากต่อจากจุดสุดท้ายของคำสั่งเดิม มายังจุด X1,Y1 เพื่อให้เส้นมีการต่อเนื่อง ถ้าใช้คำสั่งนี้แล้ว ไม่มีจุดเริ่มต้น ก็จะเป็นจุด (0,0)

*Picture.Pset (X1,Y1)* : เป็นคำสั่งสำหรับกำหนดสร้างจุดลงใน Picture ที่ตำแหน่ง X1,Y1 หลังจากใช้คำสั่งนี้แล้วจะได้จุด 1 จุด เราใช้คำสั่งนี้ในการสร้างกราฟ

การทำงานของ Picture Object ในกรณีนี้จะขึ้นอยู่กับสัญญาณที่รับมาจากตัวเอนโค้ดเดอร์ผ่านไมโครคอนโทรลเลอร์แปลงผลเข้ามาที่ วิซวล เบสิก ซึ่งจะแสดงออกมาเป็นการวิ่งของกราฟเส้นที่ตอบสนองอย่างทันทีทันใด (Real Time) ดังแสดงในรูปที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 กราฟแสดงค่าความเร็วรอบที่สร้างจาก Picture Object

### 3.3.8 Timer Object

Timer Object เป็นอ็อบเจกต์สำหรับตั้งรอบเวลาทำงาน โดยจะทำงานเป็นรอบ เมื่อจบรอบหนึ่งก็จะเริ่มรอบใหม่ในเวลาที่ตั้ง หรืออาจจะให้จบการทำงานเลยก็ได้ ซึ่งสามารถจะกำหนดได้ในพรอเพอร์ตี้

สำหรับ Timer Object เมื่อติดตั้งลงบนฟอร์มแล้ว เวลาโปรแกรมทำงานจะไม่แสดงอ็อบเจกต์ Timer ให้เห็น ดังนั้นสามารถนำ Timer ไปติดตั้งที่ใดบนฟอร์มก็ได้ พรอเพอร์ตี้ที่สำคัญของ Timer มีดังนี้

**Interval :** เป็นพรอเพอร์ตี้สำหรับตั้งเวลาแก่ Timer มีค่าเป็นตัวเลข โดยที่หนึ่งหน่วยจะมีค่าหนึ่งในพันของวินาที (1 หน่วย=1/1000 วินาที) ดังนั้นหากต้องการตั้งเวลา 1 วินาที จะต้องกำหนดค่า Interval เป็น 1000 การกำหนดสามารถกำหนดได้ใน Property Box หรือเขียนเป็นโปรแกรม

**Enable :** เป็นพรอเพอร์ตี้สำหรับให้อ็อบเจกต์นั้นทำงานได้หรือไม่ โดยมีค่าเป็น True หรือ False ถ้าเป็น True นั้น Timer ก็ทำงานถ้าเป็น False Timer ก็ไม่ทำงาน

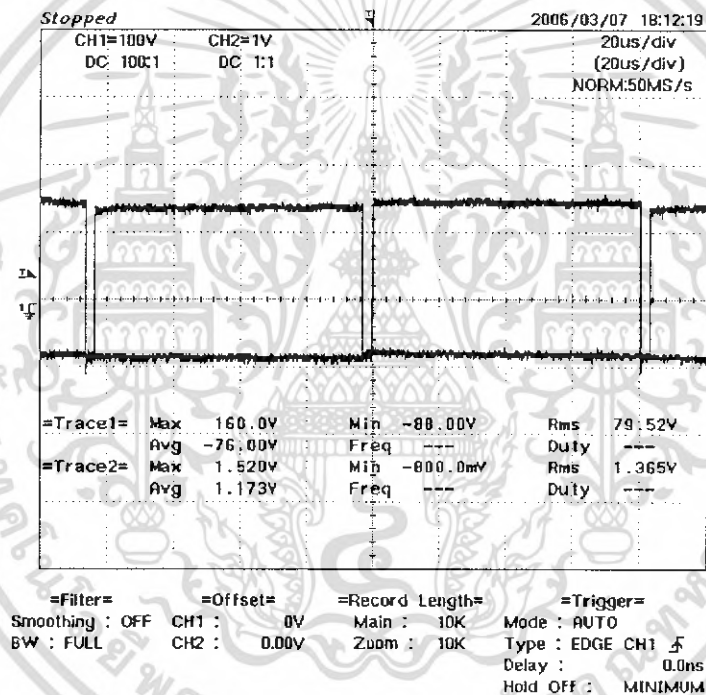


จากข้างต้นที่กล่าวว่าการกำหนดค่าเดดไทม์สามารถกำหนดได้โดยใช้การปรับเปลี่ยนค่าความต้านทาน โดยใช้แบบความต้านทานปรับค่าได้ ( R-trim pot )

จากวงจรและการทดลองจะพบว่าเมื่อปรับค่าความต้านทาน จนมีผลให้ค่า Duty Cycle อยู่ที่ประมาณ 50 % แล้วนั้น เราจะพบว่าช่วงของเดดไทม์จะอยู่ที่ประมาณ 3  $\mu$ s โดยแสดงการจับสัญญาณในรูปที่ 4.1

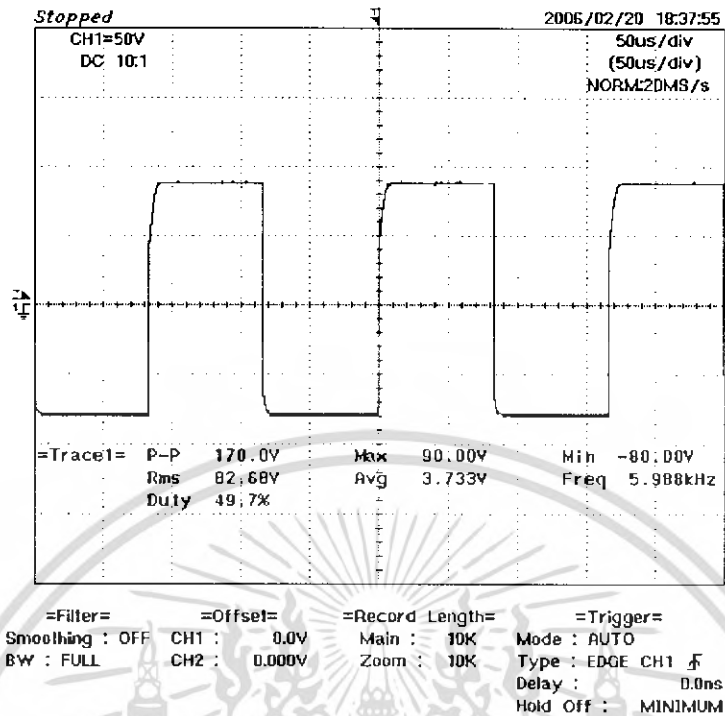
#### 4.2 การทดสอบจับสัญญาณที่ออกจากวงจรชุดขับและวงจรกำลัง

วงจรชุดขับที่จะไปสั่งให้วงจรกำลังทำงาน จะใช้ ไอซีทีแอลพี 250 ซึ่งเป็น ออปโตคัปเปิลอร์ ประกอบกับวงจรส่วนอื่นๆ เพื่อไปขับ ไอจีบีที ที่ชุดวงจรกำลังเพื่อนำไปขับมอเตอร์



รูปที่ 4.2 สัญญาณที่ออกจากชุดขับและช่วงเดดไทม์

จากการทดลองในการจับสัญญาณ 1 โหมดแล้วนั้น จึงได้ทดลองต่อว่าเมื่อจับสัญญาณ 2 โหมดแล้วสัญญาณจะออกมาในลักษณะใด เมื่อทดลองจึงพบว่าสัญญาณที่ออกมาใน 2 โหมดที่ต่างกันเป็นไปในอย่างที่ต้องการ คือ มีช่วงเดดไทม์ที่เหมาะสม ทำให้ระบบเกิดความปลอดภัย และน่าจะสามารถนำไปสั่งให้ชุด วงจรกำลังทำงานได้ ดังรูปที่ 4.2

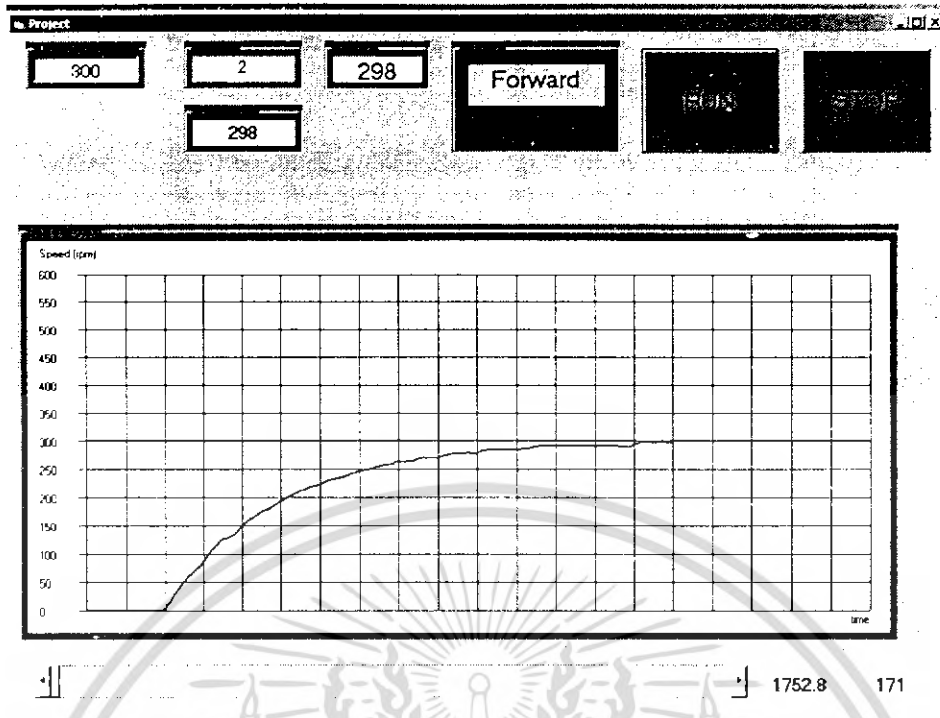


รูปที่ 4.3 การจับสัญญาณที่ออกจากชุดวงจรกำลังที่พร้อมไปขับมอเตอร์

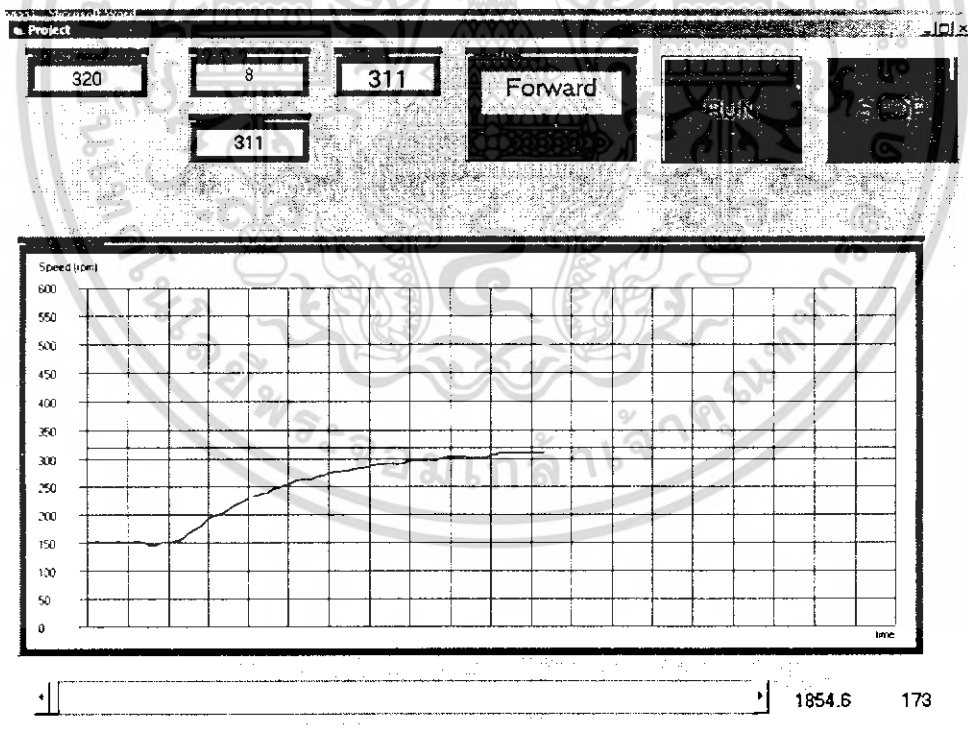
จากสัญญาณที่ออกมาจากชุดวงจรกำลังจะพบว่า แรงดันที่ออกมาใกล้เคียงกับค่าแรงดันที่มอเตอร์ต้องการ คือ 90 โวลท์ ซึ่งวงจรกำลังที่ใช้วงจร DC – DC CONVERTER FULL-BRIDGE โดยใช้ไอซีบีทีเป็นตัวสวิทช์นั้น การที่จะแรงดันออกมาตามที่ต้องการจะต้องใช้หม้อแปลง แปลงแรงดันออกมาให้ได้ใกล้เคียงกับค่าพิกัดมอเตอร์ ซึ่งแรงดันที่ออกมาก่อนที่จะไปขับมอเตอร์ สามารถจับสัญญาณได้ดังรูปที่ 4.3

#### 4.3 การทดลองแบบวงจรปิด

การทดลองในขั้นตอนนี้จะเป็นการทดลองการทำงานโดยรวมของโครงการ โดยจะใช้หน้าต่างโปรแกรม วิซวล เบสิก เป็นตัวสั่งและควบคุม รวมถึงแสดงผลที่ได้ในการทำงาน ประกอบกับชุดวงจรทุกส่วนในการทำงาน เพื่อที่จะทดสอบว่าโครงการโดยรวมเป็นไปตามวัตถุประสงค์ที่วางไว้หรือไม่



รูปที่ 4.4 กราฟแสดงผลตอบชั่วคราวของความเร็วรอบมอเตอร์เมื่อตั้งค่าความเร็วที่ 300 rpm



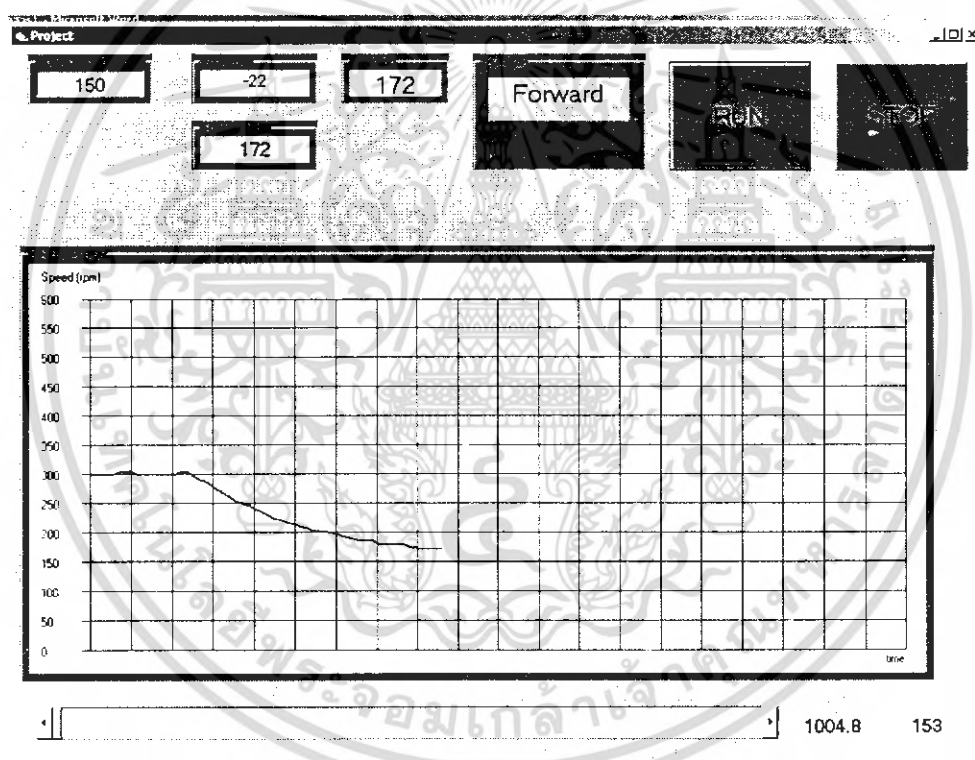
รูปที่ 4.5 กราฟความเร็วรอบของมอเตอร์เมื่อเปลี่ยนค่าความเร็วจาก 150 rpm ไปยัง 320 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.5 จะเห็นว่ากราฟของความเร็วยังจะพุ่งขึ้นไปหาค่าที่ต้องการ โดยการสั่งผ่านทางหน้าต่างของโปรแกรม เมื่อกด RUN กราฟก็จะพุ่งขึ้นไปตามค่าที่กำหนดไว้ในช่อง Speed Setpoint โดยค่าที่ได้จะแสดงในรูปของตัวเลขในช่อง Actual Speed ส่วนต่างที่ผิดพลาดไปจากค่าที่ตั้งไว้จะแสดงในช่อง Error

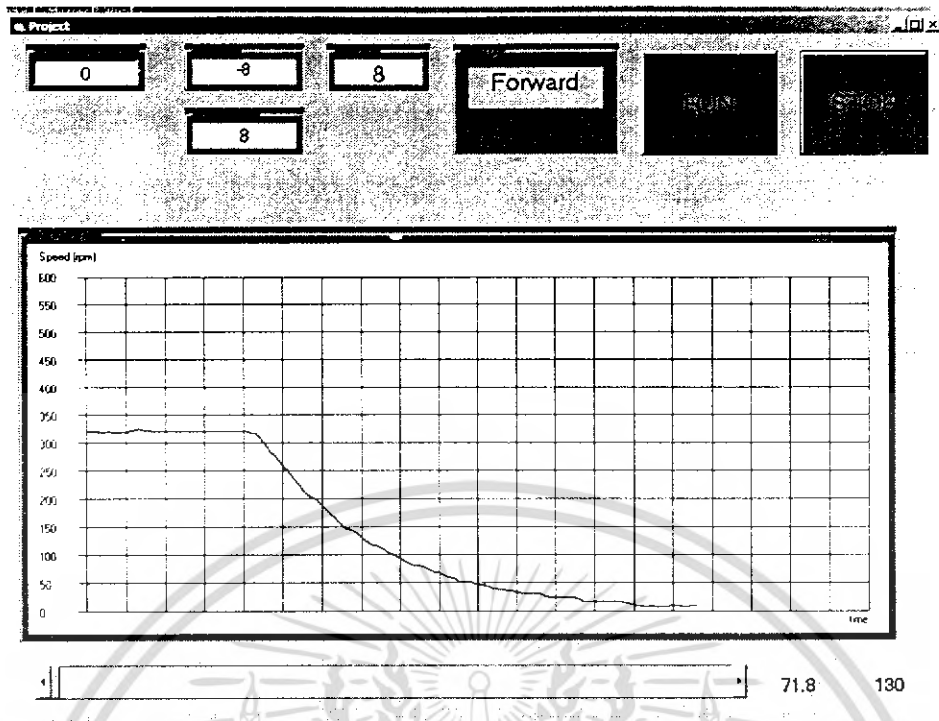
ในรูปที่ 4.6 เป็นการเพิ่มค่าของความเร็วยหลังจากที่มอเตอร์หมุนที่ค่าความเร็วรอบค่าใดค่าหนึ่งแล้ว ซึ่งการสั่งการทำงานก็จะสามารถทำได้เหมือนกับการเริ่มสตาร์ทตอนแรก ค่าความเร็วรอบก็จะพุ่งขึ้นไปตามที่ต้องการ

จากการทดลองทั้งสองขั้นตอน ดังที่ปรากฏในกราฟทั้งสองนั้น ทำให้รู้ว่ระบบสามารถทำงานในการเพิ่มความเร็วยรอบให้กับมอเตอร์ให้ได้ความเร็วตามต้องการได้ และใช้การควบคุมและแสดงผลผ่านทางโปรแกรมวิชวล เบสิก ได้ตามวัตถุประสงค์



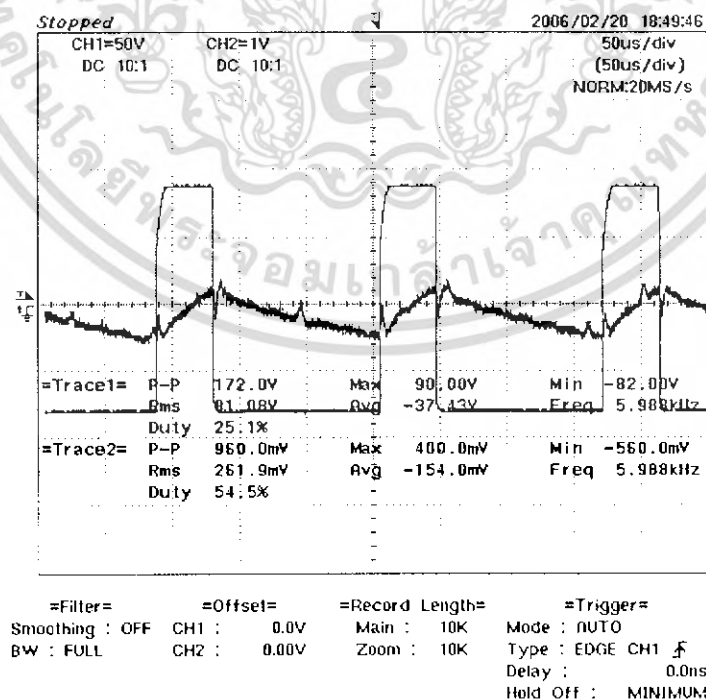
รูปที่ 4.6 กราฟความเร็วรอบของมอเตอร์เมื่อเปลี่ยนค่าความเร็วจาก 300 rpm ไปยัง 150 rpm

ในรูปที่ 4.6 แสดงให้เห็นถึงการลดลงของค่าความเร็วรอบ มาสู่ค่าความเร็วรอบใหม่ที่ต้องการ ในขณะที่มอเตอร์หมุนอยู่ที่ความเร็วเดิม ซึ่งแสดงให้เห็นว่าการควบคุมสามารถทำให้มอเตอร์ทำงานตามที่ต้องการได้



รูปที่ 4.7 กราฟความเร็วรอบที่ลดลงสู่ศูนย์

จากรูปที่ 4.7 แสดงให้เห็นถึงการทำงานในช่วงที่ต้องการให้มอเตอร์หยุดการทำงาน โดยเราจะกดปุ่ม STOP เพื่อให้มอเตอร์หยุดหมุน กราฟจะตกลงสู่ศูนย์และมอเตอร์หยุดการทำงาน เพื่อรอคำสั่งใหม่ต่อไป



รูปที่ 4.8 ลักษณะของกระแสเทียบกับแรงดันขณะปรับความเร็วมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปและวิจารณ์โครงงาน

#### 5.1 สรุปโครงงาน

การควบคุมในลักษณะของโครงงานที่ทำนี้ จะช่วยในเรื่องของความสะดวกในการทราบคุณลักษณะของมอเตอร์ในด้านของความเร็วรอบ ในขณะที่มอเตอร์ทำงาน รวมไปถึงการปรับค่าความเร็วรอบต่างๆ ตามที่ต้องการ ซึ่งสามารถกระทำได้ในโครงงานนี้

การออกแบบวงจรการขับเคลื่อนมอเตอร์ โครงงานนี้ใช้เทคนิคการควบคุมแบบ DC – DC CONVERTER FULL- BRIDGE ซึ่งใช้สวิตช์เป็น ไอจีบีที 4 ตัว ซึ่งเป็นวิธีหนึ่งที่นิยมใช้กัน โดยให้สวิตช์ไอจีบีทีทำงานเป็นคู่ มีโหมตการทำงานเป็น 4 คอร์ดแตรนท์ ซึ่งสามารถที่จะตอบสนองความต้องการในการทดสอบการควบคุมให้มอเตอร์ทำงานได้หลากหลาย โดยส่งผ่านทางหน้าต่างโปรแกรม วิชวล เบสิก ได้เป็นอย่างดี

จากการทดลองต่อเนื่องมา ทางกลุ่มสามารถที่จะทำให้มอเตอร์หมุนได้ด้วยการส่งจากหน้าต่างโปรแกรม วิชวล เบสิก ที่ได้สร้างขึ้น โดยการส่งคำสั่งไปที่ชุดไมโครคอนโทรลเลอร์ที่ไปสั่งให้ชุดขับทำงานให้มอเตอร์หมุน และมีเอนโค้ดเดอร์คอยตรวจสอบค่าการทำงานของมอเตอร์มาแสดงที่หน้าต่างของ โปรแกรม วิชวล เบสิก เพื่อทราบผลการทำงาน

ในส่วนของการควบคุมมอเตอร์ โครงงานนี้ได้ใช้การควบคุมแบบ PI ซึ่งจากการทดลองจะพบว่าระบบจะตอบสนองต่อการสั่งการได้รวดเร็วยิ่งขึ้นจากการควบคุมแบบ P และการผิดพลาดของการตอบสนองหรือค่า error ของระบบลดน้อยลงเมื่อได้รับการควบคุมแบบ I ซึ่งเมื่อนำการควบคุมทั้ง 2 มาใช้ร่วมกัน ก็เป็นผลให้ระบบมีประสิทธิภาพในการทำงานมากขึ้น

เมื่อมองโครงงานโดยรวมจะได้ว่าจากการศึกษาและค้นคว้าเนื้อหาของโครงงานและการประกอบชิ้นงานและทำการทดลองนั้น ผลที่ได้สามารถที่จะยืนยันแนวความคิดของการขับเคลื่อนมอเตอร์กระแสตรง โดยควบคุมผ่านทางโปรแกรม วิชวล เบสิก ได้เป็นอย่างดี เป็นไปตามวัตถุประสงค์และขอบเขตของโครงงานที่วางไว้ทุกประการ

สุดท้าย จากการเริ่มต้นทำโครงงานและทดลองในส่วนต่างๆ นั้น ทำให้ผู้ทำได้รับความรู้และประสบการณ์ในการทำงานใหม่ๆ ได้มองเห็นถึงคุณสมบัติของอุปกรณ์ต่างๆ และความผิดพลาดรวมไปถึงการแก้ปัญหาในการทำงาน ซึ่งจะนำไปปรับแก้และประยุกต์ใช้ในการทำงานและการดำเนินชีวิตประจำวันต่อไปในอนาคต

## 5.2 ปัญหา อุปสรรค และการแก้ไข

จากการทำงานที่ผ่านมาสามารถที่จะสรุปปัญหาและอุปสรรค รวมถึงการแก้ไขได้ดังนี้

ในส่วนของชิ้นงาน ปัญหาและอุปสรรคที่สำคัญคือ ในส่วนของวงจรชุดขับที่ใช้ ออปโตคัปเปลอร์ TLP 250 เป็นตัวไอโซเลต ซึ่งในการทดลองและต่อวงจรนั้นเกิดการผิดพลาดในการต่อวงจร ทำให้อุปกรณ์และวงจรเกิดการเสียหาย

ส่วนต่อมาที่สำคัญที่สุดคือ ส่วนของวงจรถูกำลังที่ใช้ ไอจีบีที 4 ตัวเป็นตัวสวิทช์ ซึ่งจากการทดลองที่ผ่านมาจะประสบปัญหาความเสียหายจากไอจีบีทีบ่อยครั้งมาก ซึ่งเกิดมาจากการต่อวงจรที่ผิดพลาดอย่างหนึ่ง รวมไปถึงการทดลอง โดยการถอดชุดแหล่งจ่ายที่มาจ่ายให้กับชุดวงจรถูกำลังหลังสุด ซึ่งในความเป็นจริงจะต้องปิดแหล่งจ่ายนี้เป็นอันดับแรก จึงเป็นผลให้เกิดการช็อตกันระหว่างขาของไอจีบีที ทำให้เกิดการเสียหายกับตัวไอจีบีที ไม่สามารถที่จะใช้งานต่อไปได้ ทำให้การทดลองการทำงานของโครงงานต้องหยุดชะงักไป

แนวทางการแก้ไขที่ทางกลุ่มได้จัดทำคือการแยกสวิทช์ที่ทำการตัดต่อแหล่งจ่ายนี้ ออกมาให้เห็นเด่นชัด ไม่รวมกับแหล่งจ่ายชุดอื่นๆ ทำเครื่องหมายและบอกลำดับการตัดต่อการทำงานอย่างชัดเจน เพื่อลดความผิดพลาดในการทำงาน ซึ่งก็สามารถรองรับกับความผิดพลาดได้เป็นอย่างดี

อุปสรรคที่สำคัญอีกอย่างหนึ่งคือการใช้เครื่องมือวัดต่างๆ ที่ใช้ประกอบในการทดลอง ซึ่งเครื่องมือบางชนิด ทางกลุ่มยังไม่มี ความชำนาญในการใช้งาน จึงต้องศึกษาการใช้งานอย่างละเอียด เพื่อความถูกต้องในการทดลองและที่สำคัญคือป้องกันความเสียหายที่อาจจะเกิดได้กับเครื่องมือวัดนั้นๆ ซึ่งในส่วนนี้ก็ถือว่าเป็นส่วนหนึ่งที่ทำให้การทดลองล่าช้าไปพอสมควร แต่ก็มีประโยชน์ที่ได้ความรู้จากการได้ใช้เครื่องมือหลากหลายประเภท

ปัญหาที่สำคัญอีกส่วนหนึ่งก็คือ การเลือกซื้ออุปกรณ์มาทำการต่อวงจร เพราะอุปกรณ์บางตัวเมื่อไปซื้อกลับพบว่าของหมด หรือของยังไม่มา อีกข้อคือการซื้ออุปกรณ์หรือไอซี บางตัวผิดเบอร์ทำให้เกิดการล่าช้าในการจัดซื้อใหม่หรือเปลี่ยนใหม่

ทางด้านปัญหาของการจัดทำปริญาานิพนธ์นั้นนั้น ส่วนใหญ่จะมาจากการรวบรวมข้อมูล การคัดเลือกข้อมูลที่ต้องการและเหมาะสมกับโครงงาน ซึ่งข้อมูลของอุปกรณ์หรือวงจรบางจุด มีหลากหลายเนื้อหาที่ต่างกันจึงต้องนำมาประยุกต์ให้ได้ข้อมูลที่ใส่ใจความและเหมาะสมที่สุด และความผิดพลาดทางด้านกราฟและการจัดวางรูปแบบ ซึ่งทั้งหมดนี้ก็เป็นปัญหาอีกทางหนึ่งที่ทำให้โครงงานเกิดความล่าช้า

จากปัญหาอุปสรรคต่างๆ ที่มีผลทำให้โครงงานสะดุดหรือล่าช้า ตามที่ได้กล่าวมานั้น แนวทางการแก้ไขของทางกลุ่ม อันดับแรกคือการศึกษาค้นคว้าเพิ่มเติมในขอบเขตที่สามารถศึกษาและค้นคว้าด้วยตนเองได้ แต่หากเป็นปัญหาหรืออุปสรรคที่ไม่สามารถแก้ไขได้ ก็จะทำให้การขอคำปรึกษา คำอธิบาย หรือการแก้ไขปัญหาจากอาจารย์ที่ปรึกษา รวมไปถึงคำแนะนำจากนักศึกษาปริญญาโท ซึ่งจากการปรึกษาและขอคำแนะนำ ทำให้ข้อผิดพลาดของโครงงานถูกแก้ไขอย่างถูกต้องทำให้โครงงานสามารถดำเนินการต่อไปได้ และประสบผลสำเร็จเป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 แนวทางการพัฒนา

สามารถที่จะนำไปใช้ควบคุมการทำงานของมอเตอร์ในงานอุตสาหกรรมได้ โดยการใช้การควบคุมแบบควบคุมความเร็วรอบ สามารถจะทำงานกับมอเตอร์หลาย ๆ ตัวได้ในหน้าต่างโปรแกรมเดียวกัน ในส่วนของการสื่อสารแบบ RS-232 ที่ใช้สายเชื่อมต่อผ่านทาง Serial Port ก็อาจจะเปลี่ยนมาใช้ในการสื่อสารไร้สายโดยใช้ Wire Less แทน และอาจจะพัฒนาไปถึงการควบคุมมอเตอร์เหนี่ยวนำ และเครื่องจักรกลทางอุตสาหกรรมอื่นๆ อีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] N. Mohan, T .M. Underland, and W.P. Robins, **Power Electronics Converter, Application and Design**, John Wiley & Sons, Inc., 1989.
- [2] Jan Axelson, **Serial port complete : programming and circuits for RS-232 and RS-485 links and networks**, Madison WI : Lakeview Research, 2000.
- [3] Muhammad H. Rashid, **Power electronics : circuits, devices, and applications**, 2nd Ed., London : Prentice-Hall International, 1993.
- [4] B.K. Bose, **Power electronics and AC drives**, Englewood Cliffs, NJ : Prentice - Hall, 1986.
- [5] วีระเชษฐ ชันเงิน และวุฒิพล ธาราธิรเศรษฐ์, อิเล็กทรอนิกส์กำลัง, กรุงเทพฯ, 2547.
- [6] ไวกพ วัฒนะ และคณะ, " การขับมอเตอร์ไฟฟ้ากระแสตรงแบบ 4 คอร์ตแดรนท์ ควบคุมโดยไมโครคอนโทรลเลอร์ ", วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต, สาขาวิศวกรรมไฟฟ้า, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2544
- [7] ทิวา ชำนาญถิ่นเทียน และคณะ, " เอซีชอปเปอร์ชนิดกลับขั้วได้สำหรับปรับสภาพแรงดันไฟฟ้า ", วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต, สาขาวิศวกรรมไฟฟ้า, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2547
- [8] อภิชาติ ภูพลับ,สนุก! กับการประยุกต์ใช้ Visual Basic, นนทบุรี : อินโฟเพรส, 2546
- [9] อภิชาติ ภูพลับ, เริ่มต้นการเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic, นนทบุรี : อินโฟเพรส, 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรม วิชาล เบลิก

Dim speed(0 To 200) As Single

Dim Index As Single

Dim data As String

Dim counter As Single

Dim error, U\_PCA, error\_Old, sum, Ui, Up, U, txtSpeeds As Single

Dim status\_For, status\_Rev As Boolean

Private Sub cmdRUN\_Click()

txtSpeeds = Val(txtSpeed.Text)

pic(0).DrawWidth = 2

pic(0).Line (0, Val(txtSpeed.Text)-(200, Val(txtSpeed.Text)), vbRed

If txtSpeeds < 3000 Then

trmComp.Enabled = True

Else

MsgBox ("Speed can't more than 3000 rpm"), , "command error"

txtSpeed.Text = " "

End If

End Sub

Private Sub cmdStop\_Click()

txtSpeeds = 0

txtSpeed.Text = "0"

Iblhsc.Caption = U\_PCA

IblError.Caption = error

IblPCA.Caption = U

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Activate()
```

```
pic(0).DrawWidth = 1
```

```
pic(0).Scale (-15, 660)-(205, -60)
```

```
pic(0).Line (0, 0)-(200, 0)
```

```
pic(0).Line (0, 0)-(0, 600)
```

```
For i = 0 To 600 Step 50
```

```
pic(0).Line (-2, i)-(200, i)
```

```
pic(0).CurrentX = -9
```

```
pic(0).CurrentY = i + 9
```

```
pic(0).Print i
```

```
Next i
```

```
For k = 0 To 200 Step 20
```

```
pic(0).CurrentX = k - 2
```

```
pic(0).CurrentY = -6
```

```
pic(0).Print k * 0.025
```

```
Next k
```

```
For j = 10 To 200 Step 10
```

```
pic(0).Line (j, 600)-(j, 0)
```

```
Next j
```

```
pic(0).CurrentX = -12
```

```
pic(0).CurrentY = 650
```

```
pic(0).Print "Speed (rpm)"
```

```
pic(0).CurrentX = 190
```

```
pic(0).CurrentY = -35
```

```
pic(0).Print "time (sec)"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub
Private Sub Form_Initialize()
    Index = 0
    For i = 1 To 4 Step 1
        pic(i).Visible = False
    Next i
End Sub
Private Sub Form_Load()
    msc.CommPort = 2
    msc.Settings = "9600,n,8,1"
    msc.PortOpen = True
    msc.RThreshold = 1
    msc.InputLen = 0
    If Not msc.PortOpen Then
        msc.PortOpen = True
    Else
        MsgBox ("com port already open"), , "comport error"
    End If
    hsc.Min = 128
    hsc.Max = 200
    hsc.Value = 128
    trmComp.Enabled = False
    error_Old = 0
    lblStatus_F.Visible = False
    lblStatus_R.Visible = False
End Sub
Private Sub hsc_Change()
    msc.Output = Chr(hsc.Value)
    lblhsc.Caption = (hsc.Value)
End Sub
Private Sub hsc_Scroll()
    msc.Output = Chr(hsc.Value)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    lblhsc.Caption = (hsc.Value)
End Sub
Private Sub msc_OnComm()
Select Case msc.CommEvent
    Case comEvReceive

        data = msc.Input          'get data from 8051

        lbl6.Caption = Asc(Right(data, 1)) * 256 + Asc(Left(data, 1))
        lbl1.Caption = Round(Val(lbl6.Caption))          ' sampling 100
mS if mul by 1.5
        counter = Round(Val(lbl1.Caption))
        Call plotgraph
End Select
End Sub
Sub plotgraph()

If Index < 200 Then
    If counter < 600 Then
        pic(1).Visible = False
        pic(2).Visible = False
        pic(3).Visible = False
        pic(4).Visible = False
        Call scale_0
    ElseIf counter < 1200 Then
        pic(1).Visible = True
        pic(2).Visible = False
        pic(3).Visible = False
        pic(4).Visible = False
        Call scale_1
    ElseIf counter < 1800 Then
        pic(1).Visible = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pic(2).Visible = True
pic(3).Visible = False
pic(4).Visible = False
Call scale_2
Elseif counter < 2400 Then
pic(1).Visible = False
pic(2).Visible = False
pic(3).Visible = True
pic(4).Visible = False
Call scale_3
Elseif counter < 3000 Then
pic(1).Visible = False
pic(2).Visible = False
pic(3).Visible = False
pic(4).Visible = True
Call scale_4
End If
Else
Index = 0
speed(0) = speed(200)
pic(0).Cls
pic(1).Cls
pic(2).Cls
pic(3).Cls
pic(4).Cls

End If
End Sub
Sub scale_0()
pic(0).DrawWidth = 1
pic(0).Scale (-15, 660)-(205, -40)

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pic(0).Line (0, 0)-(200, 0)    'X axis
pic(0).Line (0, 0)-(0, 600)    'Y axis
```

```
For i = 0 To 600 Step 50        ' scale Y axis
pic(0).Line (-2, i)-(200, i)
pic(0).CurrentX = -9
pic(0).CurrentY = i + 9
pic(0).Print i
Next i
```

```
For k = 0 To 200 Step 20
pic(0).CurrentX = k - 2
pic(0).CurrentY = -6
pic(0).Print k * 0.025
Next k
```

```
For j = 10 To 200 Step 10
pic(0).Line (j, 600)-(j, 0)
Next j
```

```
pic(0).CurrentX = -12
pic(0).CurrentY = 650
pic(0).Print "Speed (rpm)"
```

```
pic(0).CurrentX = 190
pic(0).CurrentY = -20
pic(0).Print "time (sec)"
```

```
pic(0).DrawWidth = 2
speed(Index + 1) = counter
pic(0).Line (Index, speed(Index))-(Index + 1, speed(Index + 1)), vbBlue
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Index = Index + 1
End Sub
Sub scale_1()

    pic(1).DrawWidth = 1
    pic(1).Scale (-15, 1260)-(205, 560)
    pic(1).Line (0, 600)-(200, 600)
    pic(1).Line (0, 600)-(0, 1200)

    For i = 600 To 1200 Step 50
        pic(1).Line (-2, i)-(200, i)
        pic(1).CurrentX = -13
        pic(1).CurrentY = i + 9
        pic(1).Print i
    Next i

    For j = 10 To 200 Step 20
        pic(1).Line (j, 1200)-(j, 600)
    Next j

    pic(1).CurrentX = -12
    pic(1).CurrentY = 1250
    pic(1).Print "Speed (rpm)"

    pic(1).CurrentX = 195
    pic(1).CurrentY = 595
    pic(1).Print "time"

    pic(1).DrawWidth = 2
    speed(Index + 1) = counter
    pic(1).Line (Index, speed(Index))-(Index + 1, speed(Index + 1)), vbBlue
    Index = Index + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Sub scale\_2()

pic(2).DrawWidth = 1

pic(2).Scale (-15, 1860)-(205, 1160)

pic(2).Line (0, 1200)-(200, 1200)

pic(2).Line (0, 1200)-(0, 1800)

For i = 1200 To 1800 Step 50

pic(2).Line (-2, i)-(200, i)

pic(2).CurrentX = -13

pic(2).CurrentY = i + 9

pic(2).Print i

Next i

For j = 10 To 200 Step 20

pic(2).Line (j, 1800)-(j, 1200)

Next j

pic(2).CurrentX = -12

pic(2).CurrentY = 1850

pic(2).Print "Speed (rpm)"

pic(2).CurrentX = 195

pic(2).CurrentY = 1195

pic(2).Print "time"

pic(2).DrawWidth = 2

speed(Index + 1) = counter

pic(2).Line (Index, speed(Index))-(Index + 1, speed(Index + 1)), vbBlue

Index = Index + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Sub scale\_3()

pic(3).DrawWidth = 1

pic(3).Scale (-15, 2460)-(205, 1760)

pic(3).Line (0, 2400)-(200, 2400)

pic(3).Line (0, 1800)-(0, 2400)

For i = 1800 To 2400 Step 50

pic(3).Line (-2, i)-(200, i)

pic(3).CurrentX = -13

pic(3).CurrentY = i + 9

pic(3).Print i

Next i

For j = 10 To 200 Step 20

pic(3).Line (j, 2400)-(j, 1800)

Next j

pic(3).CurrentX = -12

pic(3).CurrentY = 2450

pic(3).Print "Speed (rpm)"

pic(3).CurrentX = 195

pic(3).CurrentY = 1795

pic(3).Print "time"

pic(3).DrawWidth = 2

speed(Index + 1) = counter

pic(3).Line (Index, speed(Index))-(Index + 1, speed(Index + 1)), vbBlue

Index = Index + 1

End Sub

Sub scale\_4()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pic(4).DrawWidth = 1
pic(4).Scale (-15, 3060)-(205, 2360)
pic(4).Line (0, 2400)-(200, 2400)
pic(4).Line (0, 2400)-(0, 3000)

For i = 2400 To 3000 Step 50
pic(4).Line (-2, i)-(200, i)
pic(4).CurrentX = -13
pic(4).CurrentY = i + 9
pic(4).Print i
Next i

For j = 10 To 200 Step 20
pic(4).Line (j, 3000)-(j, 2400)
Next j

pic(4).CurrentX = -12
pic(4).CurrentY = 3050
pic(4).Print "Speed (rpm)"

pic(4).CurrentX = 195
pic(4).CurrentY = 2395
pic(4).Print "time"

pic(4).DrawWidth = 2
speed(Index + 1) = counter
pic(4).Line (Index, speed(Index))-(Index + 1, speed(Index + 1)), vbBlue
Index = Index + 1

End Sub

Private Sub optFor_Click()
    lblStatus_F.Visible = True
    lblStatus_R.Visible = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

status_For = True
status_Rev = False
End Sub
Private Sub optRev_Click()
    lblStatus_F.Visible = False
    lblStatus_R.Visible = True
    status_For = False
    status_Rev = True

```

```

End Sub
Private Sub trmComp_Timer()
If status_For = True Then
    error = txtSpeeds - counter
    sum = (error + error_Old)
    error_Old = sum
    Ui = sum * 0.05
    Up = error * 0.05
    U = Ui + Up
    U_PCA = Round(U / 41 + 128)
    lblStatus_F.Visible = True
    lblStatus_R.Visible = False

    msc.Output = Chr(U_PCA)

```

\*\*\*\*\* PCA upper bound Protection \*\*\*\*\*

```

If U_PCA > 190 Then
    trmComp.Enabled = False
End If

```

\*\*\*\*\*

```

lblhsc.Caption = U_PCA
lblError.Caption = error
lblPCA.Caption = U

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    error = txtSpeeds - counter
    sum = (error + error_Old)
    error_Old = sum
    Ui = sum * 0.05
    Up = error * 0.05
    U = Ui + Up
    U_PCA = Round(128 - U / 41)
    lblStatus_F.Visible = False
    lblStatus_R.Visible = True

    msc.Output = Chr(U_PCA)

***** PCA upper bound Protection *****
    If U_PCA < 66 Then
        trmComp.Enabled = False
    End If
*****

    lblhsc.Caption = U_PCA
    lblError.Caption = error
    lblPCA.Caption = U
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ไมโครคอนโทรลเลอร์

```
CKCON EQU 08FH ; Clock Control Register
CMOD EQU 0D9H ; PCA Timer/Counter Mode
CCAP0H EQU 0FCH ; PCA Compare/Capture Module 0 High
CCAP0L EQU 0ECH ; PCA Compare/Capture Module 0 Low
CCAPM0 EQU 0DCH ; PAC Timer/Counter Mode0
CCON EQU 0D8H ; PCA Timer/Counter Control
```

```
ORG 0000H
```

```
-----BAUD RETE SETTING
MOV IE,#00000000B ; ensure that IE register never set before
MOV TMOD,#00100101B ; setting T1 auto reload T0 COUNTR 16 BIT
MOV TL1,#251D ; setting
MOV TH1,#251D ; setting
MOV SCON,#01010000B ; setting serial communication mode 1 (popular
mode)
CLR RI ; ensure that RI can't be "1" before
CLR TI
SETB TR1 ; begin communication
mov p0,#00000000B
-----*/

MOV CKCON,#11011110B
MOV CMOD,#11111001B
MOV CCAP0H,#128D
MOV CCAP0L,CCAP0H
MOV CCAPM0,#01000010B
MOV CCON,#01000000B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MAIN: JNB RI,$ ;
      CLR RI ; waite for recive data
      MOV A,SBUF ;
LOOP: NOP
      MOV CKCON,#11011110B
      MOV CMOD,#11111001B
      MOV CCAP0H,A
      MOV CCAP0L,CCAP0H
      MOV CCAPM0,#01000010B
      MOV CCON,#01000000B ; begin PCA

      MOV TL0,#00D ; setting
      MOV TH0,#00D ; setting
      SETB P3.4
      SETB TR0
      ACALL DELAY
      CLR TR0

      MOV SBUF,TL0
      JNB TI,$
      CLR TI
      MOV SBUF,TH0
      JNB TI,$
      CLR TI

      MOV P0,A
      JNB RI,LOOP
      SJMP MAIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DELAY: MOV R7,#60D ; 240 equal delay 100 mS

LOOP1: MOV R6,#159D

LOOP2: NOP

NOP

DJNZ R6,LOOP2

DJNZ R7,LOOP1

RET



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 1 KB of On-chip XRAM
- 32 KB of On-chip Flash Memory
  - Data Retention: 10 Years at 85°C
  - Read/Write Cycle: 10K
- 2 KB of On-chip Flash for Bootloader
- 2 KB of On-chip EEPROM
- Read/Write Cycle: 100K
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- Maximum Crystal Frequency 40 MHz, in X2 Mode, 20 MHz (CPU Core, 20 MHz)
- Five Ports: 32 + 2 Digital I/O Lines
- Five-channel 16-bit PCA with:
  - PWM (8-bit)
  - High-speed Output
  - Timer and Edge Capture
- Double Data Pointer
- 21-bit Watchdog Timer (7 Programmable Bits)
- 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- On-chip Emulation Logic (Enhanced Hook System)
- Power Saving Modes:
  - Idle Mode
  - Power-down Mode
- Power Supply: 3V to 5.5V
- Temperature Range: Industrial (-40° to +85°C)
- Packages: VQFP44, PLCC44

## Description

The AT89C51AC2 is a high performance Flash version of the 80C51 single chip 8-bit microcontrollers. It contains a 32 KB Flash memory block for program and data.

The 32 KB Flash memory can be programmed either in parallel mode or in serial mode with the ISP capability or with software. The programming voltage is internally generated from the standard VCC pin.

The AT89C51AC2 retains all features of the 80C51 with 256 bytes of internal RAM, a 7-source 4-level interrupt controller and three timer/counters. In addition, the AT89C51AC2 has a 10-bit A/D converter, a 2 KB Boot Flash memory, 2 KB EEPROM for data, a Programmable Counter Array, an XRAM of 1024 bytes, a Hardware Watch-Dog Timer, and a more versatile serial channel that facilitates multiprocessor communication (EUART). The fully static design of the AT89C51AC2 reduces system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The AT89C51AC2 has two software-selectable modes of reduced activity and an 8-bit clock prescaler for further reduction in power consumption. In the idle mode the CPU is frozen while the peripherals and the interrupt system are still operating. In the Power-down mode the RAM is saved and all other functions are inoperative.

The added features of the AT89C51AC2 make it more powerful for applications that need A/D conversion, pulse width modulation, high speed I/O and counting capabilities such as industrial control, consumer goods, alarms, motor control, among others. While remaining fully compatible with the 80C51, the T89C51AC2 offers a superset of this standard microcontroller. In X2 mode, a maximum external clock rate of 20 MHz reaches a 300 ns cycle time.



Enhanced 8-bit  
Microcontroller  
with 32 KB Flash  
Memory

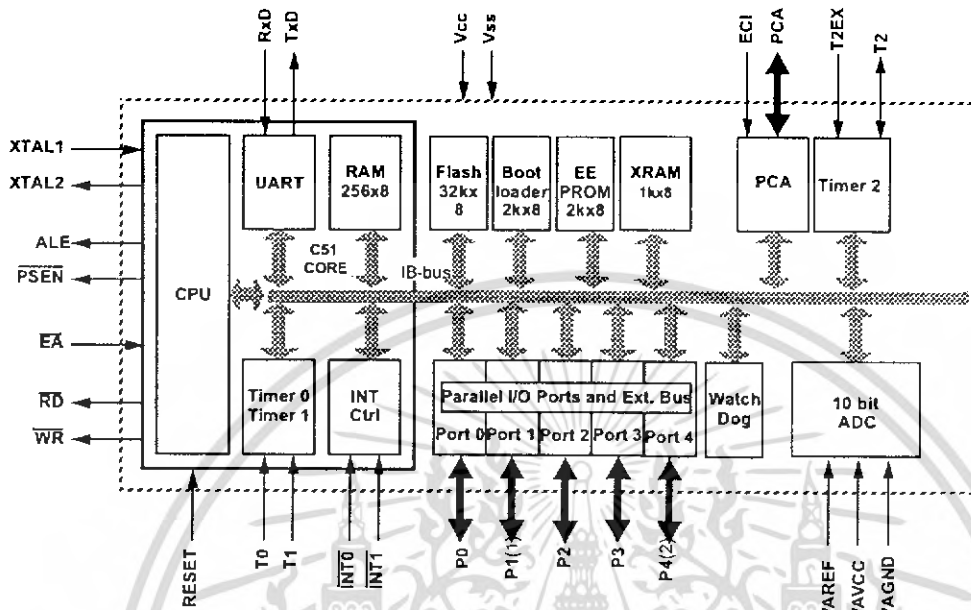
AT89C51AC2  
T89C51AC2

Rev. 4127F-8051-03/05



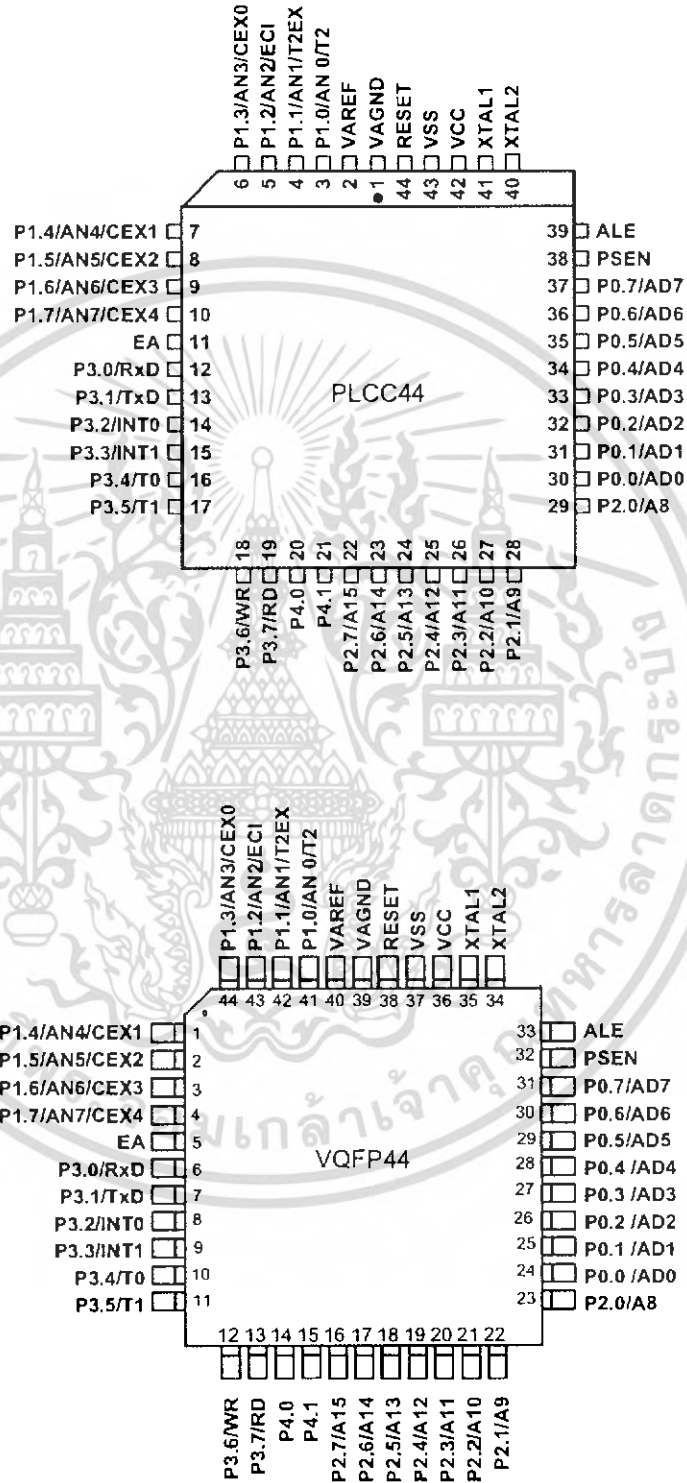
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Block Diagram



- Notes:
1. 8 analog Inputs/8 Digital I/O
  2. 2-Bit I/O Port

Pin Configuration



# TLP250

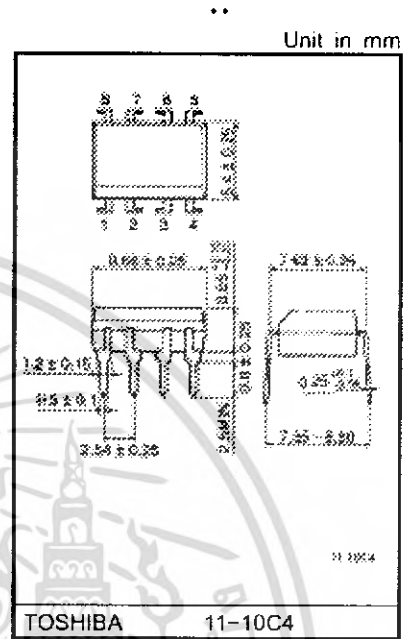
Transistor Inverter  
 Inverter For Air Conditionor  
 IGBT Gate Drive  
 Power MOS FET Gate Drive

The TOSHIBA TLP250 consists of a GaAlAs light emitting diode and a integrated photodetector.  
 This unit is 8-lead DIP package.  
 TLP250 is suitable for gate driving circuit of IGBT or power MOS FET.

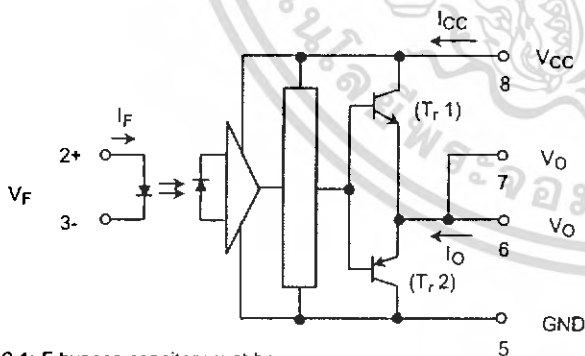
- Input threshold current:  $I_F=5\text{mA}(\text{max.})$
- Supply current ( $I_{CC}$ ):  $11\text{mA}(\text{max.})$
- Supply voltage ( $V_{CC}$ ):  $10\text{--}35\text{V}$
- Output current ( $I_O$ ):  $\pm 1.5\text{A}(\text{max.})$
- Switching time ( $t_{pLH}/t_{pHL}$ ):  $1.5\mu\text{s}(\text{max.})$
- Isolation voltage:  $2500V_{\text{rms}}(\text{min.})$
- UL recognized: UL1577, file No.E67349
- Option (D4) type  
 VDE approved: DIN VDE0884/06.92,certificate No.76823  
 Maximum operating insulation voltage:  $630\text{V}_{\text{PK}}$   
 Highest permissible over voltage:  $4000\text{V}_{\text{PK}}$

(Note) When a VDE0884 approved type is needed, please designate the "option (D4)"

- Creepage distance:  $6.4\text{mm}(\text{min.})$   
 Clearance:  $6.4\text{mm}(\text{min.})$

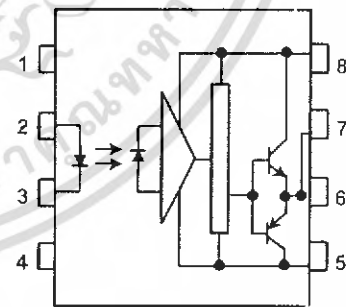


### Schmatic



A 0.1 $\mu\text{F}$  bypass capacitor must be connected between pin 8 and 5 (See Note 5).

### Pin Configuration (tcp view)



- 1: N.C.
- 2: Anode
- 3: Cathode
- 4: N.C.
- 5: GND
- 6:  $V_O$  (Output)
- 7:  $V_O$
- 8:  $V_{CC}$

### Truth Table

		Tr1	Tr2
Input LED	On	On	Off
	Off	Off	On

เอกสารนี้เป็นเอกสารของบริษัทที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

**Absolute Maximum Ratings (Ta = 25°C)**

Characteristic		Symbol	Rating	Unit	
LED	Forward current	$I_F$	20	mA	
	Forward current derating (Ta ≥ 70°C)	$\Delta I_F / \Delta T_a$	-0.36	mA / °C	
	Peak transient forward current (Note 1)	$I_{FPT}$	1	A	
	Reverse voltage	$V_R$	5	V	
	Junction temperature	$T_J$	125	°C	
Detector	"H" peak output current ( $P_W \leq 2.5\mu s, f \leq 15kHz$ ) (Note 2)	$I_{OPH}$	-1.5	A	
	"L" peak output current ( $P_W \leq 2.5\mu s, f \leq 15kHz$ ) (Note 2)	$I_{OPL}$	+1.5	A	
	Output voltage	(Ta ≤ 70°C)	$V_O$	35	V
		(Ta = 85°C)		24	
	Supply voltage	(Ta ≤ 70°C)	$V_{CC}$	35	V
		(Ta = 85°C)		24	
	Output voltage derating (Ta ≥ 70°C)	$\Delta V_O / \Delta T_a$	-0.73	V / °C	
	Supply voltage derating (Ta ≥ 70°C)	$\Delta V_{CC} / \Delta T_a$	-0.73	V / °C	
Junction temperature	$T_J$	125	°C		
Operating frequency (Note 3)	$f$	25	kHz		
Operating temperature range	$T_{opr}$	-20-85	°C		
Storage temperature range	$T_{stg}$	-55-125	°C		
Lead soldering temperature (10 s) (Note 4)	$T_{sol}$	260	°C		
Isolation voltage (AC, 1 min., R.H. ≤ 60%) (Note 5)	$BV_S$	2500	Vrms		

Note 1: Pulse width  $P_W \leq 1\mu s, 300pps$

Note 2: Exponential waveform

Note 3: Exponential waveform,  $I_{OPH} \leq -1.0A (\leq 2.5\mu s), I_{OPL} \leq +1.0A (\leq 2.5\mu s)$

Note 4: It is 2 mm or more from a lead root.

Note 5: Device considered a two terminal device: Pins 1, 2, 3 and 4 shorted together, and pins 5, 6, 7 and 8 shorted together.

Note 6: A ceramic capacitor(0.1μF) should be connected from pin 8 to pin 5 to stabilize the operation of the high gain linear amplifier. Failure to provide the bypassing may impair the switching property. The total lead length between capacitor and coupler should not exceed 1cm.

**Recommended Operating Conditions**

Characteristic	Symbol	Min.	Typ.	Max.	Unit
Input current, on (Note 7)	$I_{F(ON)}$	7	8	10	mA
Input voltage, off	$V_{F(OFF)}$	0	—	0.8	V
Supply voltage	$V_{CC}$	15	—	30   20	V
Peak output current	$I_{OPH}/I_{OPL}$	—	—	±0.5	A
Operating temperature	$T_{opr}$	-20	25	70   85	°C

Note 7: Input signal rise time (fall time) < 0.5 μs.

**Electrical Characteristics (Ta = -20~70°C, unless otherwise specified)**

Characteristic		Symbol	Test Circuit	Test Condition	Min.	Typ.*	Max.	Unit
Input forward voltage		V <sub>F</sub>	—	I <sub>F</sub> = 10 mA, Ta = 25°C		1.6	1.8	V
Temperature coefficient of forward voltage		ΔV <sub>F</sub> / ΔTa	—	I <sub>F</sub> = 10 mA	—	-2.0	—	mV / °C
Input reverse current		I <sub>R</sub>	—	V <sub>R</sub> = 5V, Ta = 25°C		—	10	μA
Input capacitance		C <sub>T</sub>	—	V = 0, f = 1MHz, Ta = 25°C	—	45	250	pF
Output current	"H" level	I <sub>OPH</sub>	3	V <sub>CC</sub> = 30V (*1) I <sub>F</sub> = 10 mA V <sub>B-6</sub> = 4V	-0.5	-1.5	—	A
	"L" level	I <sub>OPL</sub>	2		I <sub>F</sub> = 0 V <sub>B-5</sub> = 2.5V	0.5	2	
Output voltage	"H" level	V <sub>OH</sub>	4	V <sub>CC1</sub> = +15V, V <sub>EE1</sub> = -15V R <sub>L</sub> = 200Ω, I <sub>F</sub> = 5mA	11	12.8	—	V
	"L" level	V <sub>OL</sub>	5	V <sub>CC1</sub> = +15V, V <sub>EE1</sub> = -15V R <sub>L</sub> = 200Ω, V <sub>F</sub> = 0.8V	—	-14.2	-12.5	
Supply current	"H" level	I <sub>CCH</sub>	—	V <sub>CC</sub> = 30V, I <sub>F</sub> = 10mA Ta = 25°C	—	7	—	mA
				V <sub>CC</sub> = 30V, I <sub>F</sub> = 10mA	—	—	11	
	"L" level	I <sub>CCL</sub>	—	V <sub>CC</sub> = 30V, I <sub>F</sub> = 0mA Ta = 25°C	—	7.5	—	
				V <sub>CC</sub> = 30V, I <sub>F</sub> = 0mA	—	—	11	
Threshold input current	"Output L→H"	I <sub>FLH</sub>	—	V <sub>CC1</sub> = +15V, V <sub>EE1</sub> = -15V R <sub>L</sub> = 200Ω, V <sub>O</sub> > 0V	—	1.2	5	mA
Threshold input voltage	"Output H→L"	I <sub>FHL</sub>	—	V <sub>CC1</sub> = +15V, V <sub>EE1</sub> = -15V R <sub>L</sub> = 200Ω, V <sub>O</sub> < 0V	0.8	—	—	V
Supply voltage		V <sub>CC</sub>	—		10	—	35	V
Capacitance (input-output)		C <sub>S</sub>	—	V <sub>S</sub> = 0, f = 1MHz Ta = 25°C	—	1.0	2.0	pF
Resistance(input-output)		R <sub>S</sub>	—	V <sub>S</sub> = 500V, Ta = 25°C R.H. ≤ 60%	1 × 10 <sup>-12</sup>	10 <sup>14</sup>	—	Ω

\* All typical values are at Ta = 25°C (\*1): Duration of I<sub>O</sub> time ≤ 50μs

**Switching Characteristics (Ta = -20~70°C , unless otherwise specified)**

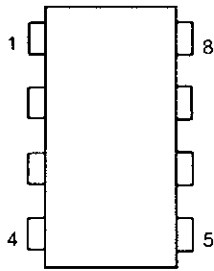
Characteristic	Symbol	Test Circuit	Test Condition	Min.	Typ.*	Max.	Unit
Propagation delay time	L→H	6	I <sub>F</sub> = 8mA (Note 7) V <sub>CC1</sub> = +15V, V <sub>EE1</sub> = -15V R <sub>L</sub> = 200Ω	—	0.15	0.5	μs
	H→L			—	0.15	0.5	
Output rise time	t <sub>r</sub>			—	—	—	
Output fall time	t <sub>f</sub>			—	—	—	
Common mode transient immunity at high level output	C <sub>MH</sub>	7	V <sub>CM</sub> = 600V, I <sub>F</sub> = 8mA V <sub>CC</sub> = 30V, Ta = 25°C	-5000	—	—	V / μs
Common mode transient immunity at low level output	C <sub>ML</sub>	7	V <sub>CM</sub> = 600V, I <sub>F</sub> = 0mA V <sub>CC</sub> = 30V, Ta = 25°C	5000	—	—	V / μs

\* All typical values are at Ta = 25°C

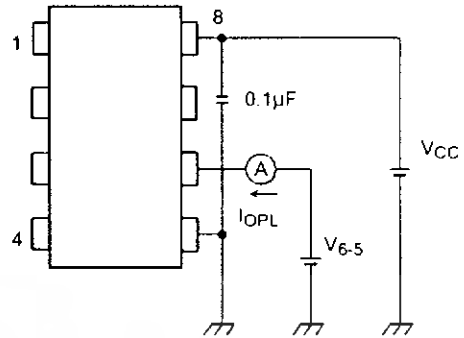
Note 7: Input signal rise time (fall time) < 0.5 μs.



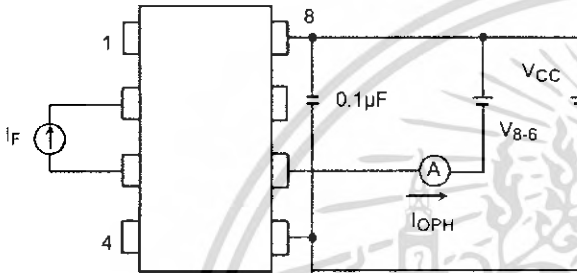
Test Circuit 1 :



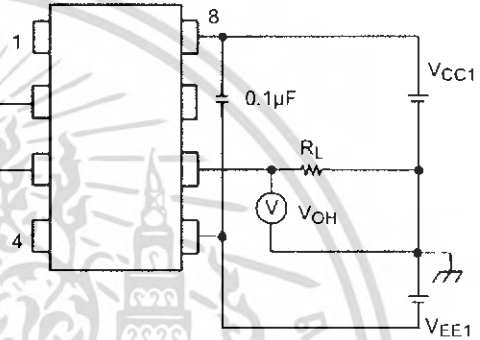
Test Circuit 2 : IOPL



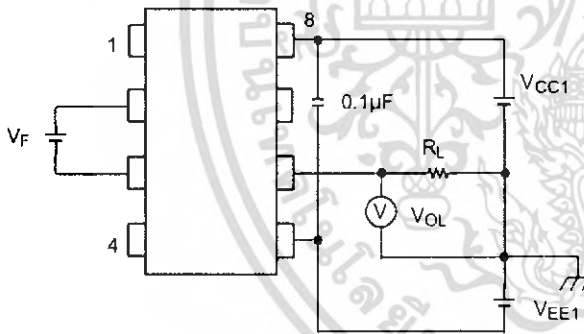
Test Circuit 3 : IOPH



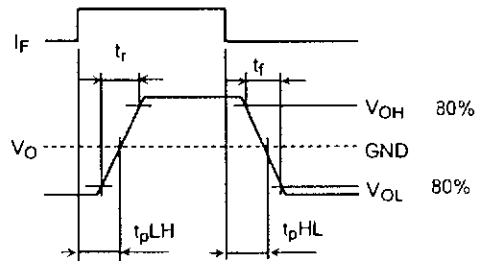
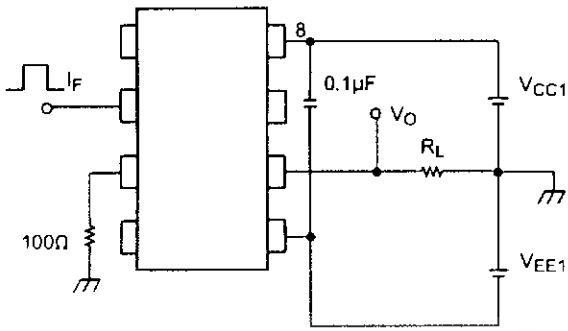
Test Circuit 4 : VOH



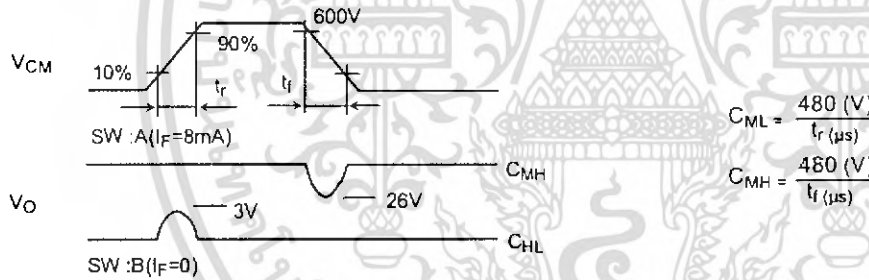
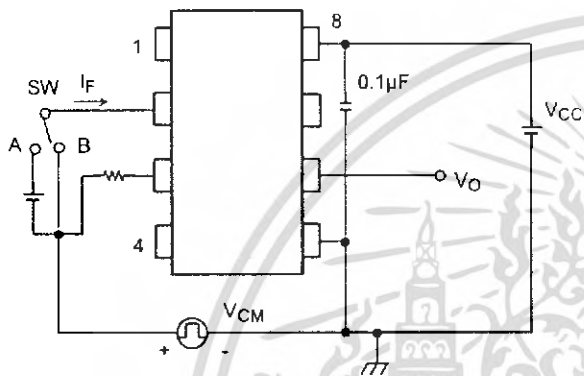
Test Circuit 5 : VOL



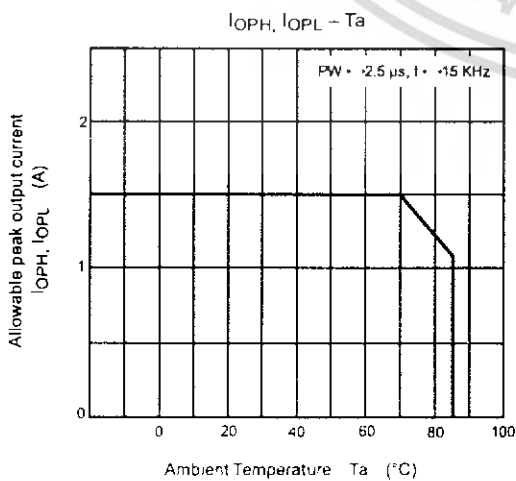
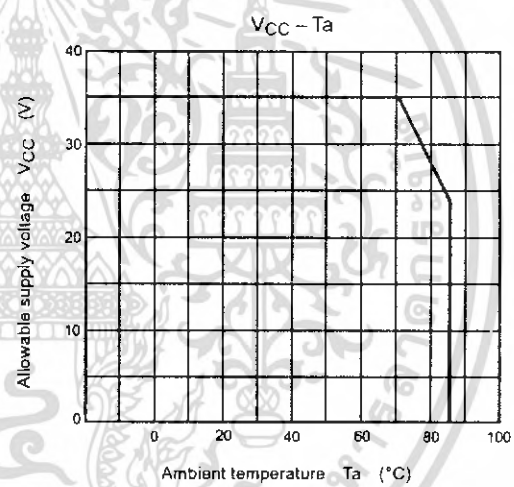
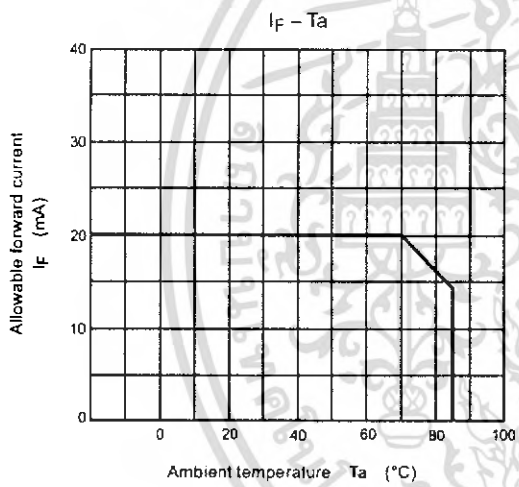
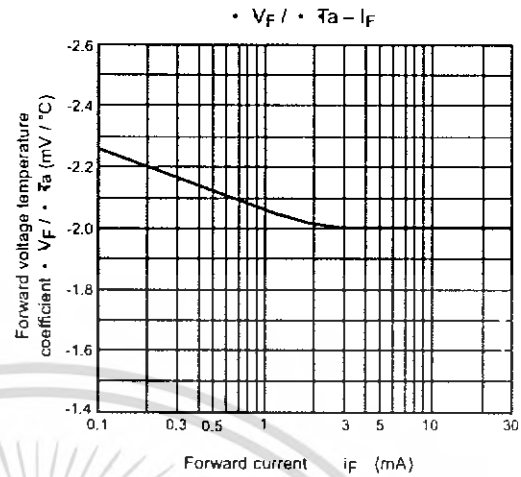
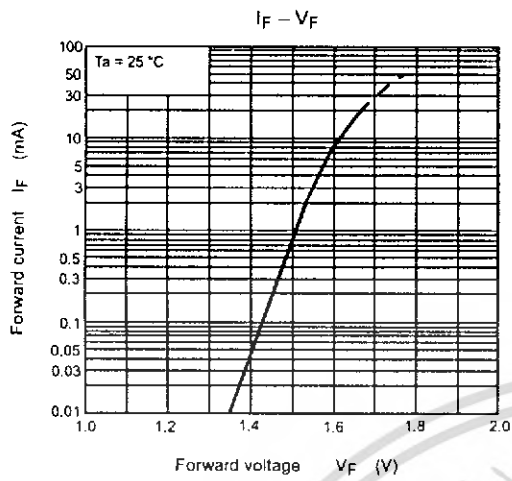
Test Circuit 6:  $t_{pLH}$ ,  $t_{pHL}$ ,  $t_r$ ,  $t_f$



Test Circuit 7:  $C_{MH}$ ,  $C_{ML}$



$C_{ML}(C_{MH})$  is the maximum rate of rise (fall) of the common mode voltage that can be sustained with the output voltage in the low (high) state.



# GT60M303

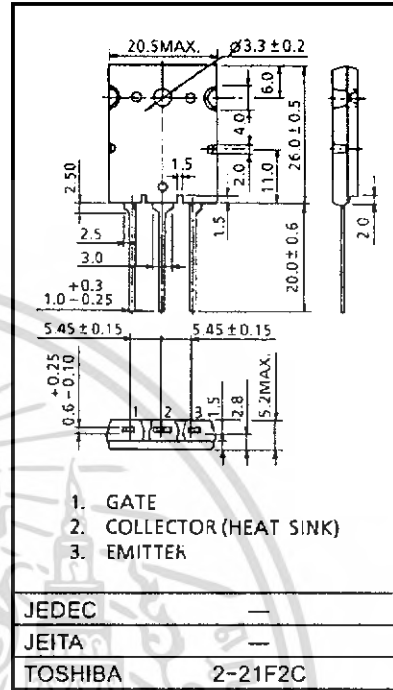
## HIGH POWER SWITCHING APPLICATIONS

- The 4th Generation
- FRD Included Between Emitter and Collector
- Enhancement-Mode
- High Speed IGBT :  $t_f = 0.25\mu s$  (TYP.)  
FRD :  $t_{rr} = 0.7\mu s$  (TYP.)
- Low Saturation Voltage :  $V_{CE(sat)} = 2.1V$  (TYP.)

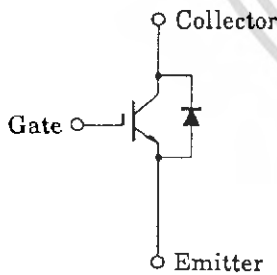
## MAXIMUM RATINGS (Ta = 25°C)

CHARACTERISTIC		SYMBOL	RATING	UNIT
Collector-Emitter Voltage		$V_{CES}$	900	V
Gate-Emitter Voltage		$V_{GES}$	$\pm 25$	V
Collector Current	DC	$I_C$	60	A
	1ms	$I_{CP}$	120	
Emitter-Collector Forward Current	DC	$I_{ECF}$	15	A
	1ms	$I_{ECFP}$	120	
Collector Power Dissipation (Tc = 25°C)		$P_C$	170	W
Junction Temperature		$T_j$	150	°C
Storage Temperature Range		$T_{stg}$	-55~150	°C
Screw Torque		—	0.8	N·m

Unit: mm

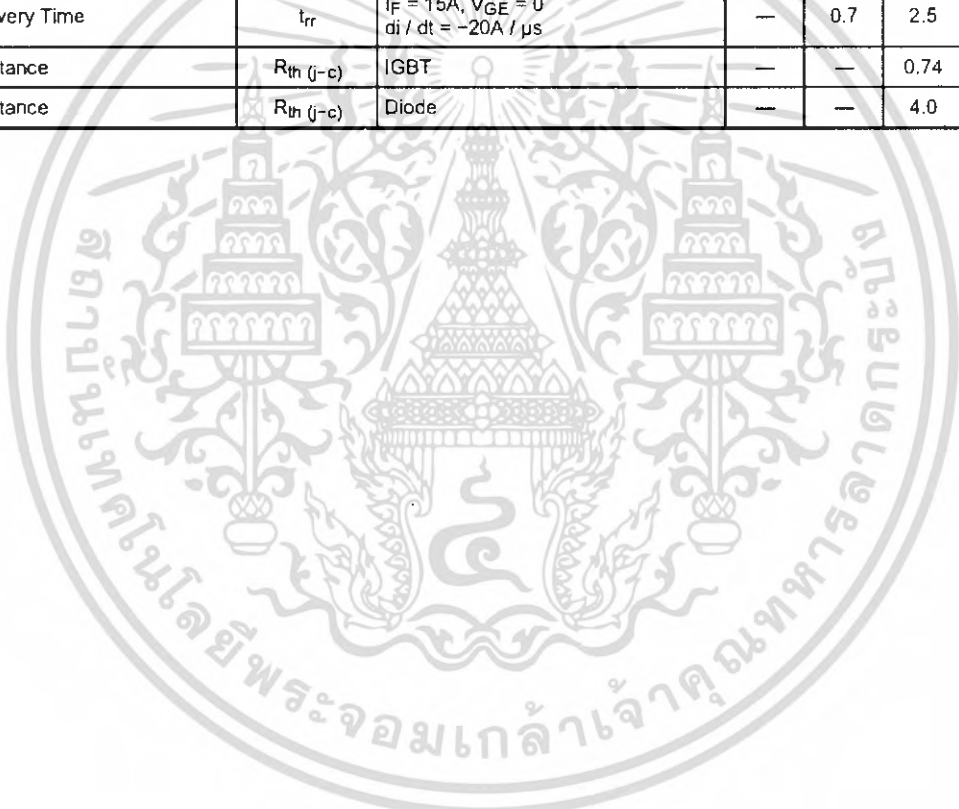


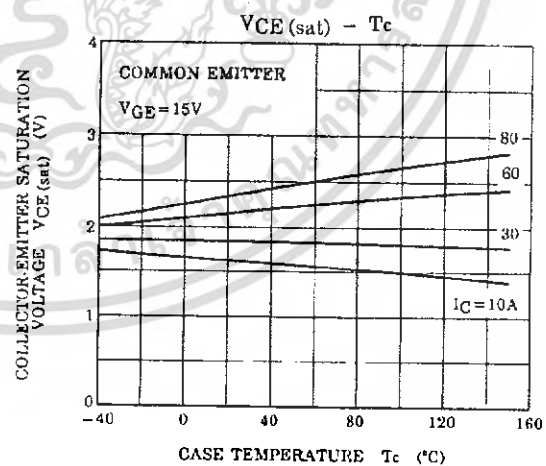
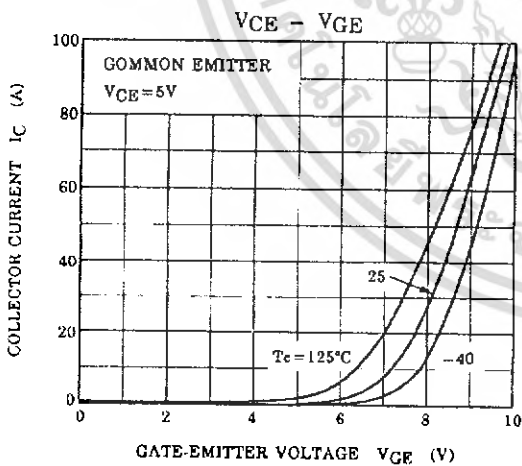
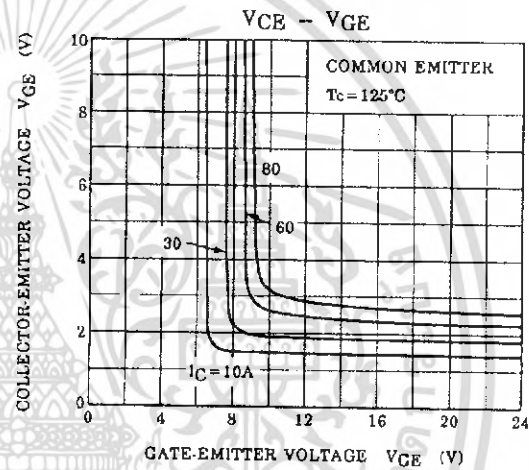
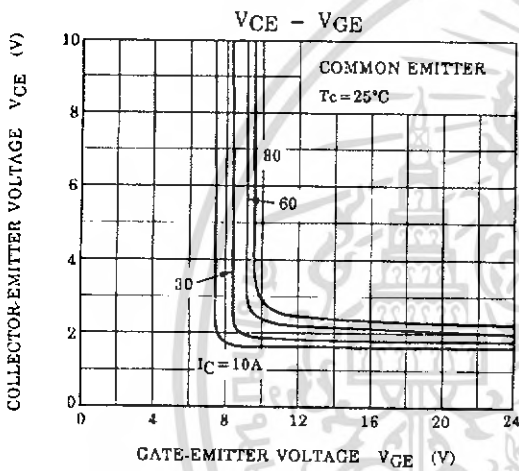
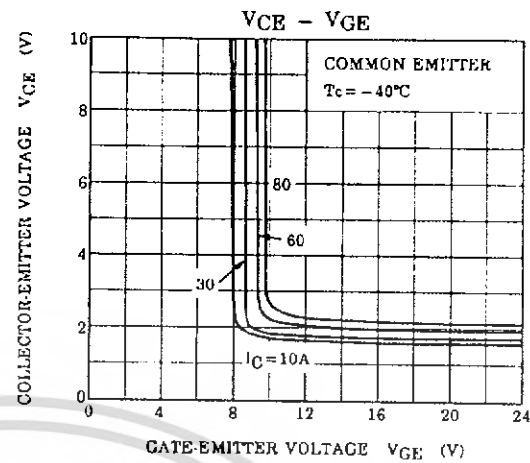
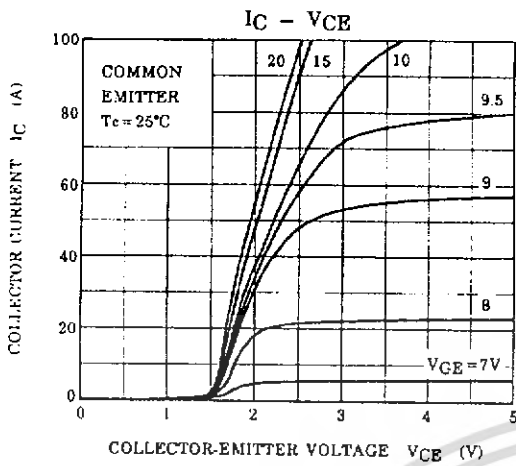
## EQUIVALENT CIRCUIT

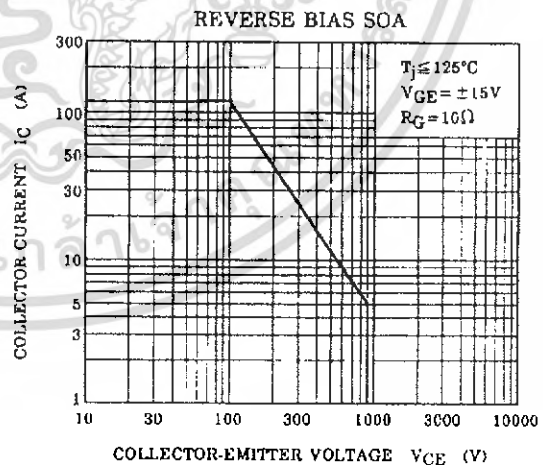
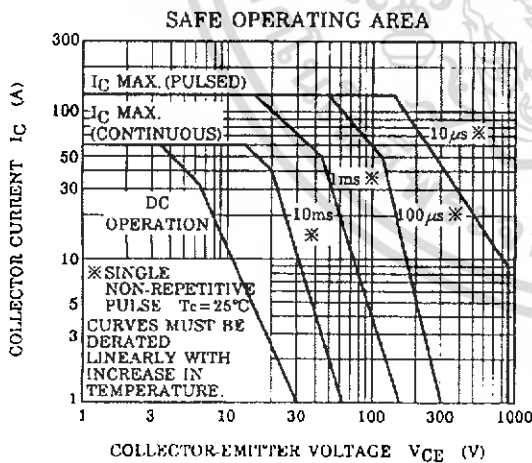
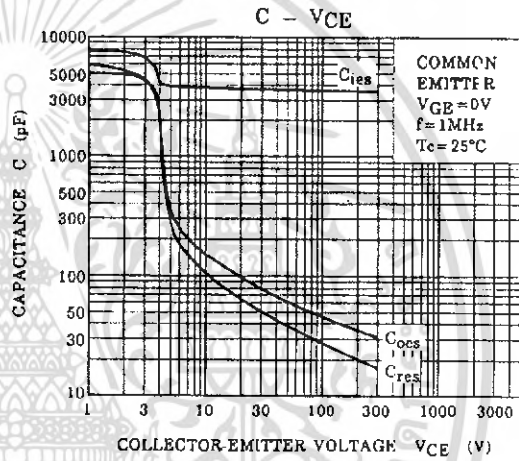
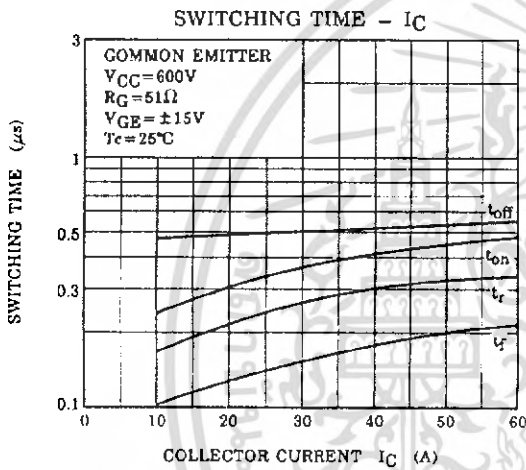
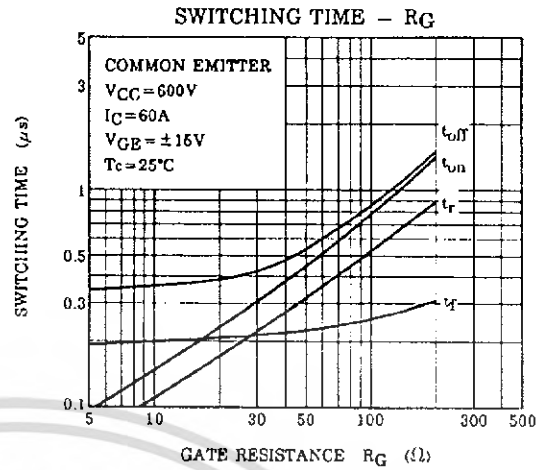
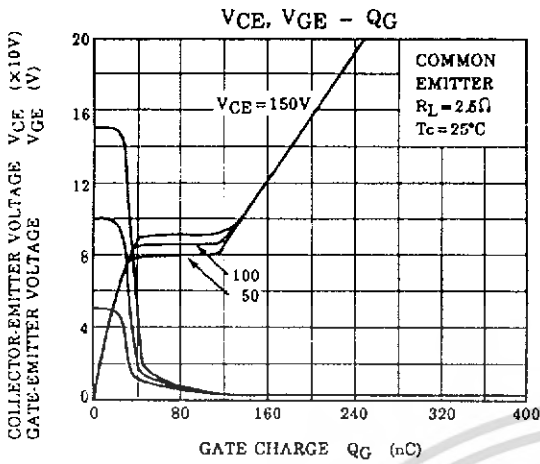


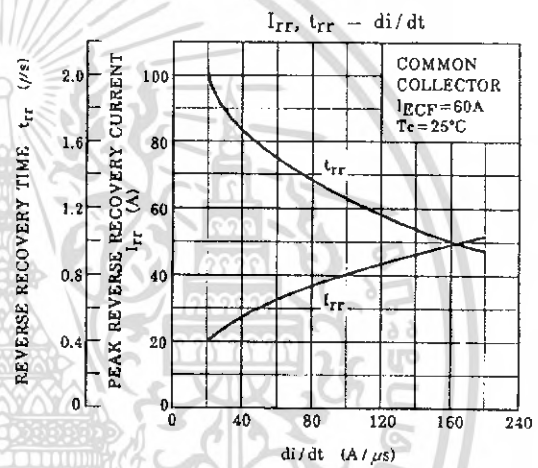
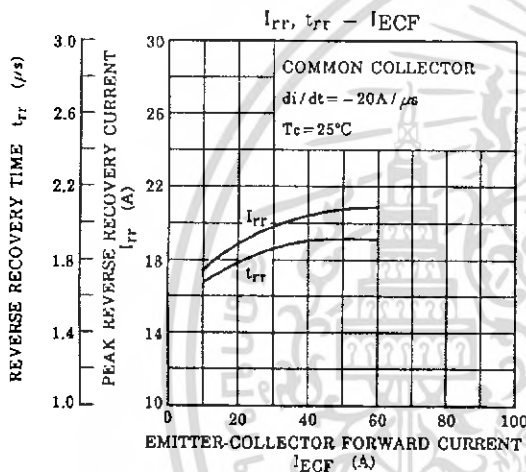
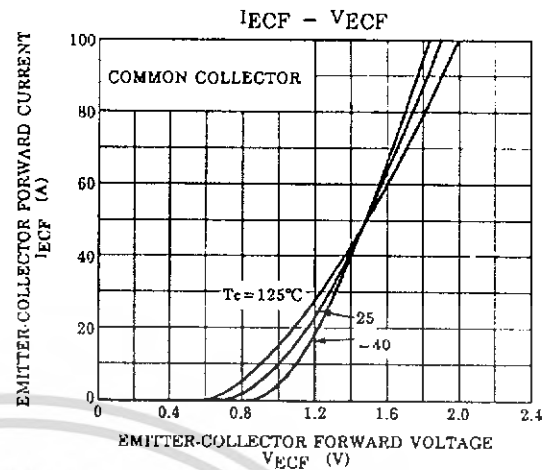
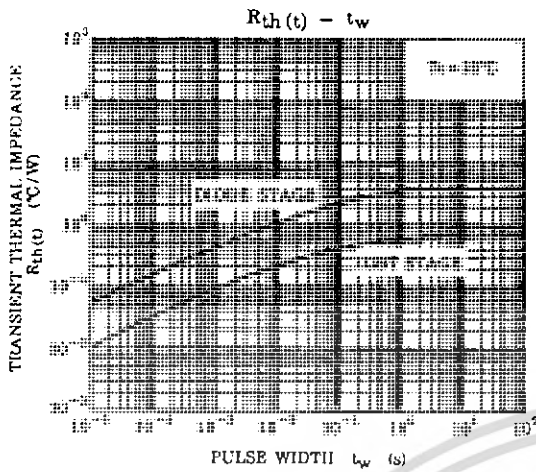
**ELECTRICAL CHARACTERISTICS (Ta = 25°C)**

CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN	TYP.	MAX	UNIT	
Gate Leakage Current	$I_{GES}$	$V_{GE} = \pm 25V, V_{CE} = 0$	—	—	$\pm 500$	nA	
Collector Cut-off Current	$I_{CES}$	$V_{CE} = 900V, V_{GE} = 0$	—	—	1.0	mA	
Gate-Emitter Cut-off Voltage	$V_{GE(OFF)}$	$I_C = 60mA, V_{CE} = 5V$	3.0	—	6.0	V	
Collector-Emitter Saturation Voltage	$V_{CE(sat)}(1)$	$I_C = 10A, V_{GE} = 15V$	—	1.6	2.2	V	
Collector-Emitter Saturation Voltage	$V_{CE(sat)}(2)$	$I_C = 60A, V_{GE} = 15V$	—	2.1	2.7	V	
Input Capacitance	$C_{ies}$	$V_{CE} = 10V, V_{GE} = 0, f = 1MHz$	—	3800	—	pF	
Switching Time	Rise Time		—	0.35	0.60	μs	
	Turn-On Time		$t_{on}$	—	0.46		0.75
	Fall Time		$t_f$	—	0.25		0.40
	Turn-Off Time		$t_{off}$	—	0.60		0.70
Emitter-Collector Forward Voltage	$V_{ECF}$	$I_{EC} = 15A, V_{GE} = 0$	—	1.5	2.0	V	
Reverse Recovery Time	$t_{rr}$	$I_F = 15A, V_{GE} = 0$ $di/dt = -20A/\mu s$	—	0.7	2.5	μs	
Thermal Resistance	$R_{th(j-c)}$	IGBT	—	—	0.74	°C/W	
Thermal Resistance	$R_{th(j-c)}$	Diode	—	—	4.0	°C/W	









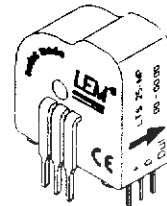
## Current Transducer LTS 25-NP

$I_{PN} = 8 - 12 - 25 \text{ A}$

For the electronic measurement of currents : DC, AC, pulsed, mixed, with a galvanic isolation between the primary circuit (high power) and the secondary circuit (electronic circuit).



Preliminary



### Electrical data

$I_{PN}$	Primary nominal r.m.s. current	25	At
$I_p$	Primary current, measuring range	0 .. $\pm 80$	At
$V_{OUT}$	Analog output voltage	$I_p = 0$	2.5 <sup>1)</sup>
		$\pm I_{PN}$	$2.5 \pm 0.625$
$N_s$	Number of secondary turns ( $\pm 0.1 \%$ )	2000	
$R_L$	Load resistance	$\geq 2$	k $\Omega$
$R_{IM}$	Internal measuring resistance ( $\pm 0.5 \%$ )	50	$\Omega$
$TCR_{IM}$	Thermal drift of $R_{IM}$	$< 50$	ppm/K
$V_C$	Supply voltage ( $\pm 5 \%$ )	5	V
$I_C$	Current consumption @ $V_C = 5 \text{ V}$	Typ $20 + I_s$	mA
$V_d$	R.m.s. voltage for AC isolation test, 50/60 Hz, 1 mn	2.5	kV
$V_b$	R.m.s. rated voltage	525 <sup>2)</sup>	V

### Features

- Closed loop (compensated) multi-range current transducer using the Hall effect
- Unipolar voltage supply
- Insulated plastic case recognized according to UL 94-V0
- Compact design for PCB mounting
- Incorporated measuring resistance
- Extended measuring range.

### Advantages

- Excellent accuracy
- Very good linearity
- Very low temperature drift
- Optimized response time
- Wide frequency bandwidth
- No insertion losses
- High immunity to external interference
- Current overload capability.

### Accuracy - Dynamic performance data

X	Accuracy @ $I_{PN}, T_A = 25^\circ\text{C}$	$\pm 0.2$	%
	Accuracy with $R_{IM}$ @ $I_{PN}, T_A = 25^\circ\text{C}$	$\pm 0.7$	%
$\epsilon_L$	Linearity	$< 0.1$	%
$TCV_{OUT}$	Thermal drift of $V_{OUT}$ @ $I_p = 0$	-10°C .. +85°C	Typ 50
		-10°C .. +85°C	Max 100
$TCE_G$	Thermal drift of the gain	-10°C .. +85°C	50 <sup>3)</sup>
$V_{OM}$	Residual voltage @ $I_p = 0$ , after an overload of $3 \times I_{PN}$	$5 \times I_{PN}$	$\pm 0.5$
		$5 \times I_{PN}$	$\pm 2.0$
		$10 \times I_{PN}$	$\pm 2.0$
$t_r$	Reaction time @ 10 % of $I_{p,max}$	$< 50$	ns
$t_f$	Response time @ 90 % of $I_{p,max}$	$< 200$	ns
di/dt	di/dt accurately followed	$> 100$	A/ $\mu\text{s}$
f	Frequency bandwidth (0 .. -0.5 dB)	DC .. 100	kHz
		DC .. 200	kHz

### Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
- Uninterruptible Power Supplies (UPS)
- Switched Mode Power Supplies (SMPS)
- Power supplies for welding applications.

### General data

$T_A$	Ambient operating temperature	-10 .. +85	$^\circ\text{C}$
$T_S$	Ambient storage temperature	-25 .. +100	$^\circ\text{C}$
m	Mass	10	g
	Standards	EN 50178	

Notes : <sup>1)</sup> Absolute value @  $T_A = 25^\circ\text{C}$ ,  $2.4875 < V_{OUT} < 2.5125$

<sup>2)</sup> Pollution class 2, category III

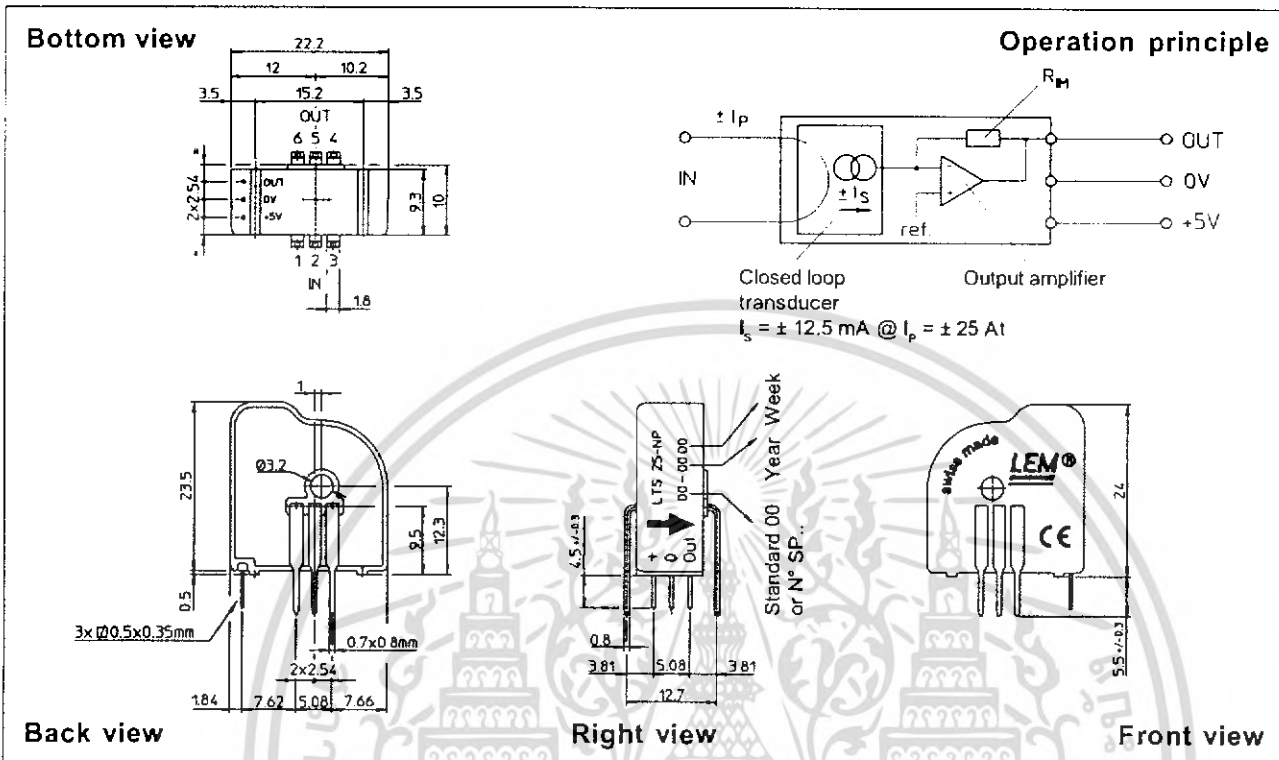
<sup>3)</sup> Only due to  $TCR_{IM}$

Copyright protected.

981222/4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## Dimensions LTS 25-NP (in mm. 1 mm = 0.0394 inch)

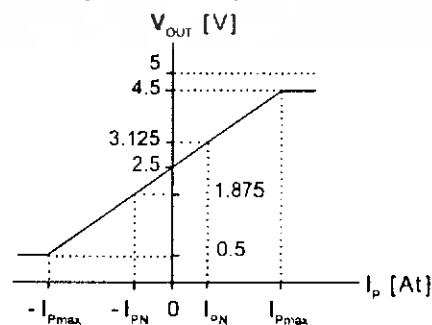


Number of primary turns	Primary nominal r.m.s. current $I_{PN}$ [A]	Nominal output voltage $V_{OUT}$ [V]	Primary resistance $R_p$ [mΩ]	Primary insertion inductance $L_p$ [μH]	Recommended connections
1	± 25	2.5 ± 0.625	0.18	0.013	
2	± 12	2.5 ± 0.600	0.81	0.05	
3	± 8	2.5 ± 0.600	1.62	0.12	

### Mechanical characteristics

- General tolerance ± 0.2 mm
- Fastening & connection of primary 6 pins 0.7 x 0.8 mm  
Recommended PCB hole 1.3 mm
- Fastening & connection of secondary 3 pins 0.5 x 0.35 mm  
Recommended PCB hole 0.8 mm
- Additional primary through-hole Ø 3.2 mm

### Output Voltage - Primary Current



### Remark

- $V_{OUT}$  is positive when  $I_p$  flows from terminals 1, 2, 3 to terminals 6, 5, 4

LEM reserves the right to carry out modifications on its transducers, in order to improve them, without previous notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้