

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แขนกล

ROBOT ARM



โดย
นายวรุฒม์ ชันศิริ
นายวัชรพล สุตตเขตต์
นายอนุวัฒน์ คณิตตานนท์

๘๖๓
๖ ๓๓๗๘
๘๕๕๐

เลขหมู่.....
เลขทะเบียน.....**82450**
วัน,เดือน,ปี.....**11 ก.ค. 2551**

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

b.....**11947184**
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนกล

ROBOT ARM

โดย

นายวรุฒม์	ชั้นศิริ	รหัส	47010672
นายวัชรพล	สุตตเขตต์	รหัส	47010680
นายอนุวัฒน์	กณทัตตานนท์	รหัส	47010929

อาจารย์ที่ปรึกษา

รศ.ดร. ชูชาติ ปิณฑวิรุจน์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2550

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แขนกล

ผู้จัดทำ

1. นายวรุฒม์ ชันศิริ
2. นายวัชรพล สุตตเขตต์
3. นายอนุวัฒน์ กยทัตตานนท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนบกล

นายวรุฒม์ ชั้นศิริ รหัส 47010672
 นายวัชรพล สุตตเขตต์ รหัส 47010680
 นายอนุวัฒน์ วัฒนรัตนันท์ รหัส 47010929
 รศ.ดร. ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา
 ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อศึกษาและพัฒนาการทำงานของแขนกลร่วมกับกล้องวิดีโอ โดยแขนกลสามารถแยกแยะรูปแบบของวัตถุตลอดจนจัดวัตถุให้อยู่ในตำแหน่งที่กำหนดไว้ได้ ซึ่งจะมีกล้องวิดีโอ เป็นตัวจับภาพของวัตถุ ส่งภาพนั้นไปประมวลผลผ่านเครื่องคอมพิวเตอร์ เพื่อแยกแยะรูปแบบและกำหนดพิกัดของวัตถุ จากนั้นคอมพิวเตอร์จะส่งคำสั่งผ่านพอร์ตอนุกรมไปยัง microcontroller เพื่อควบคุมการทำงานของ servo motor ที่ติดตั้งในแขนกลอีกที

ROBOT ARM

Mr.Varuth Chansiri ID.47010672

Mr.Vatcharapol Suttakhate ID.47010680

Mr. Anuwat Khanathattanon ID.47010929

Assoc. Prof. Dr. Chuchart Pinthaviruj Advisor
Education Year 2007

Abstract

This thesis concerns design and development of robot arm controlled by video cameras. The robot arm is able to classify the matching objects. The video cameras will capture the scene and transfer the information to be operated by the computer. The computer will sort out the matching objects and calculate the destination. Then it will send the command code via serial port to microcontroller for controlling the operation of servo motor to pick up the objects.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติประกาศ

ขอขอบคุณ รศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษาโครงการงานแขนงกล ที่ให้คำแนะนำ และให้ความสะดวกในการ จัดหาอุปกรณ์เพื่อทำโครงการ และขอขอบคุณภาควิชาอิเล็กทรอนิกส์ที่ เอื้อเพื่อสถานที่และเครื่องมือในการจัดทำโครงการครั้งนี้

ขอขอบคุณพี่ๆ นักศึกษาปริญญาโท และเพื่อนๆ ที่คอยให้คำปรึกษาแนะนำ รวมทั้งเป็น กำลังใจช่วยเหลือในด้านต่างๆ



นายวุฒม์ ชันศิริ

นายวัชรพล สุตตเขตต์

นายอนุวัฒน์ คณจัตตานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	V
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 คีชีเซอร์ไวโมเตอร์	3
2.1.1 หลักการทำงานของเซอร์ไวโมเตอร์	4
2.1.2 การกำหนดความเร็วของเซอร์ไวโมเตอร์	5
2.1.3 ข้อดีของเซอร์ไวโมเตอร์	6
2.2 การติดต่อสื่อสาร	6
2.2.1 ความรู้เบื้องต้นเกี่ยวกับพอร์ต RS-232(Serial Port)	7
2.2.2 การจัดขา ของคอนเน็กเตอร์ อนุกรมแบบ DB9	8
2.2.3 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9	8
2.2.4 การทำงานของขาสัญญาณ DB9	9
2.2.5 ระดับสัญญาณของ RS232	9
2.2.6 อัตราการส่งข้อมูล (Baud rate)	10
2.2.7 รูปแบบการสื่อสารแบบอนุกรม	10
2.3 ไมโครคอนโทรเลอร์ MCS-51	12
2.3.1 การทำงานของวงจร Programmable Counter Array (PCA)	13
2.3.2 การทำงานของ PCA Timer	16
2.3.3 คุณสมบัติของรีจิสเตอร์ CMOD	17
2.3.4 คุณสมบัติของรีจิสเตอร์ CCON	18
2.3.5 การทำงานของ PCA Module	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.6 คุณสมบัติของรีจิสเตอร์ CCAPMx	20
2.3.7 การทำงานของ PCA ในโหมด Pulse Width Modulation	23
2.4 การเปลี่ยนค่าพิกัดในระนาบ 3 มิติ	28
2.4.1 ทฤษฎีบทสามเหลี่ยม	28
2.4.2 การประยุกต์ทฤษฎีมาใช้อย่างจริงจังกับแกนกล	29
บทที่ 3 การปรับเทียบกล้อง (Camera Calibration)	33
3.1 บทนำ	33
3.2 เรขาคณิตสำหรับการเห็นใน 3 มิติ (Geometry for 3D Vision)	33
3.2.1 การเกิดภาพในกรณีที่พิกัดโลกซ้อนทับพิกัดกล้อง	33
3.2.2 การแปลงจากหน่วยความยาวเป็นพิกเซล	34
3.2.3 กรณีที่พิกัดของโลกไม่ซ้อนทับกับพิกัดของกล้อง	36
3.3 การปรับเทียบกล้อง (Camera Calibration)	37
3.4 การหาพิกัด 3 มิติของจุดในฉากโดยใช้กล้อง 2 ตัว	39
บทที่ 4 การแปลงเชิงเรขาคณิต	40
4.1 บทนำ	40
4.2 การแปลงเชิงเรขาคณิตแบบเชิงเส้น	40
4.2.1 การเลื่อนพิกัด (Translation)	42
4.2.2 การสเกล (Scaling)	44
4.2.3 การหมุนวัตถุ (Rotation)	45
4.2.4 การสะท้อน (Reflection)	46
4.2.5 การเฉือน (Shearing)	48
4.2.6 การฉายภาพ (Projection)	49
4.2.6.1 การฉายภาพแบบตั้งฉาก (Orthographic Projections)	49
4.2.6.2 การฉายภาพแบบเพอร์สเปกทีฟ (Perspective Projections)	50
4.2.7 การแปลงแบบผสม (Multiple Transformations)	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 การออกแบบ	54
5.1 โครงสร้างแกนกล	54
5.1.1 รายละเอียดของแกนกลจำลอง	55
5.2 การออกแบบวงจรควบคุม	57
5.2.1 การติดต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม	58
5.2.2 การสร้างพัลส์สัญญาณควบคุมเซอร์โวมอเตอร์	59
5.2.3 Flow chat การทำงานของไมโครคอนโทรลเลอร์ MCS-51	60
5.2.4 การทดสอบสัญญาณรบกวน	61
5.2.5 วงจรของบอร์ดที่ใช้ควบคุมเซอร์โวมอเตอร์	61
5.3 การออกแบบโปรแกรม	63
5.3.1 เป้าหมายของการออกแบบโปรแกรม	63
5.3.2 การออกแบบโปรแกรมในส่วน User Interface	63
5.3.3 Flow chat การทำงานแบบคลิกที่วัตถุ	65
5.3.4 Flow Chat การทำงานแบบปรับเมตซ์ิ่ง	66
5.3.4 Flow Chat การทำงานแบบปรับค่าเอง	67
บทที่ 6 การทดลองและผลการทดลอง	68
การทดลองที่ 1 วัดความกว้างของพัลส์สัญญาณในซีบววกของสัญญาณควบคุม เปรียบเทียบกับตำแหน่งของเซอร์โวมอเตอร์	68
การทดลองที่ 2 ทดสอบการใช้งาน โปรแกรมควบคุมแกนกล	68
การทดลองที่ 2.1 การทำงานของระบบปรับด้วยมือ	68
การทดลองที่ 2.2 การทำงานของระบบอัตโนมัติ	70
การทดลองที่ 3 ทดสอบเพื่อหาขอบเขตการทำงานของแกนกล	71
การทดลองที่ 4 ทดสอบความแม่นยำของการจับวัตถุของแกนกลโดยการกำหนด ให้แกนกลจับวัตถุที่ตำแหน่งต่างๆที่ระบุพิกัดใน โปรแกรม	72

สารบัญ (ต่อ)

	หน้า
การทดลองที่ 5 ทดสอบความแม่นยำของการระบุตำแหน่งวัตถุของโปรแกรม	73
การทดลอง 5.1 ระบุตำแหน่งวัตถุผ่านหน้าจอคอมพิวเตอร์โดยใช้ การคลิกเมาส์ที่วัตถุ	
การทดลอง 5.2 ระบุตำแหน่งวัตถุผ่านหน้าจอคอมพิวเตอร์โดยใช้ วิธีการเมคซิ่งวัตถุ	74
การทดลองที่ 6 ทดสอบความแม่นยำของการจับวัตถุของแขนกล โดยสั่งงาน ผ่านโปรแกรมด้วยวิธีการคลิกเมาส์ที่วัตถุ	75
การทดลองที่ 7 ทดสอบความแม่นยำของตำแหน่งและการจับวัตถุของแขนกล โดยสั่งงานผ่าน โปรแกรมด้วยวิธีการเมคซิ่งวัตถุ	76
การทดลองที่ 8 ทำการใช้โปรแกรมเมทเล็บเพื่อเปรียบเทียบค่าตำแหน่งต่างๆ ของวัตถุ	77
บทที่ 7 บทสรุปและข้อคิดเห็น	78
7.1 หลักการทำงาน	78
7.2 ปัญหาและอุปสรรคที่พบระหว่างดำเนินงาน	79
7.3 แนวทางในการแก้ไขปัญหา	80
7.4 ประโยชน์ที่ได้รับในการพัฒนาโครงการ	80
บรรณานุกรม	81
ภาคผนวก	82

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 ภาพตัวอย่าง เซอร์ไวโมเตอร์	3
รูปที่ 2.2 ภาพแสดงมุมของเซอร์ไวโมเตอร์ที่ความกว้างพัลส์ต่างกัน	4
รูปที่ 2.3 ภาพแสดงการรับส่งข้อมูล	6
รูปที่ 2.4 ภาพแสดงลักษณะของพอร์ตอนุกรมของ PC DB9 ตัวผู้(Male) และ พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย(Female)	7
รูปที่ 2.5 ภาพของDB9 ตัวผู้ เมื่อมองจากด้านหลัง	8
รูปที่ 2.6 ภาพแสดงระดับสัญญาณแบบ RS-232 และแบบ TTL	9
รูปที่ 2.7 ภาพแสดงรูปแบบข้อมูลแบบซิงโครนัส	10
รูปที่ 2.8 ภาพแสดงรูปแบบข้อมูลแบบอะซิงโครนัส	11
รูปที่ 2.9 แสดงการจัดขาสัญญาณของ P89C51RD2/T89C51RD2	12
รูปที่ 2.10 แสดงลักษณะของระบบ PCA Timer	14
รูปที่ 2.11 แสดงแผนผังการสร้างสัญญาณ Interrupt ของ PCA Timer	15
รูปที่ 2.12 แสดงลักษณะ โครงสร้างของวงจร PCA Timer	16
รูปที่ 2.13 แสดง โครงสร้างของ CMOD Register	17
รูปที่ 2.14 แสดง โครงสร้างของ CCON Register	18
รูปที่ 2.15 แสดง โครงสร้างของ CCAPMx Register	20
รูปที่ 2.16 แสดง โครงสร้างการทำงานของ PCA ในโหมด PWM	23
รูปที่ 2.17 รูปสามเหลี่ยมมุมฉาก	28
รูปที่ 2.18 รูปสามเหลี่ยมใด ๆ	28
รูปที่ 2.19 ระบบพิกัดเมื่อจุดกำเนิดอยู่ที่จุดกำเนิด	28
รูปที่ 2.20 ระบบพิกัดเมื่อจุดกำเนิดไม่เท่ากับจุดกำเนิด	29
รูปที่ 2.21 แสดงแกนกลจำลองเทียบกับพิกัด 3 มิติ	29
รูปที่ 2.22 แสดงแกนกลเทียบกับพิกัด 2 มิติ	29
รูปที่ 2.23 พิกัดของแกนกลจำลอง	30
รูปที่ 2.24 แสดงความสัมพันธ์ของพิกัดเก่าและพิกัดใหม่	30

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.25 แสดงความสัมพันธ์ระหว่างค่าพิกัดใหม่และความสัมพันธ์ของสามเหลี่ยม ที่ได้จากแกนกล	30
รูปที่ 2.26 แสดงแกนกลกับระบบพิกัด	31
รูปที่ 2.27 แสดงแกนกลกับสามเหลี่ยมใด ๆ	31
รูปที่ 3.1 แบบจำลองของระบบสร้างภาพ	34
รูปที่ 3.2 พิกัดของภาพ	35
รูปที่ 3.3 แบบจำลองการเกิดภาพ	36
รูปที่ 3.4 ภาพของกล่องพลาสติกใสที่นำมาใช้ในการเปรียบเทียบกล้อง	38
รูปที่ 4.1 ส่วนประกอบของระบบพิกัดโฮโมจีนียส	42
รูปที่ 4.2 การเลื่อนจุดกึ่งกลางของปริมาตรต้นฉบับมายังจุดกำเนิด	43
รูปที่ 4.3 การสเกลวัตถุ	45
รูปที่ 4.4 การแปลงแบบหมุนรอบแกนทั้งสาม	46
รูปที่ 4.5 การสะท้อนวัตถุกับระนาบ XY	47
รูปที่ 4.6 การเฉือนโดยสัมพันธ์กับค่า z เมื่อ $Sh_{zx} = 0$ และ $Sh_{zy} = 1$	48
รูปที่ 4.7 แสดงการฉายภาพลงบนระนาบ $x = 0$, $y = 0$ และ $z = 0$	50
รูปที่ 4.8 การฉายภาพแบบเพอร์สเปกทีฟ	50
รูปที่ 5.1 แสดงโครงสร้างแกนกล	54
รูปที่ 5.2 แสดงภาพแกนกลที่ออกแบบ	56
รูปที่ 5.3 แสดงการจัดขาและการนำไปใช้งานของ MAX232	58
รูปที่ 5.4 แสดงรูปการใช้ PC817 แยกกราวด์	61
รูปที่ 5.5 แสดงวงจรของบอร์ดควบคุม	62
รูปที่ 5.6 แสดงโปรแกรมที่ใช้ควบคุมแกนกล	64
รูปที่ 6.1 ภาพแสดงตำแหน่งและค่ามุมต่างๆของแกนกล	68
รูปที่ 6.2 ภาพแสดงตำแหน่งและค่ามุมต่างๆของแกนกล	69
รูปที่ 6.3 ภาพแสดงตำแหน่งและค่ามุมต่างๆของแกนกล	69
รูปที่ 6.4 แสดงพิกัดของวัตถุและค่ามุมต่างๆของเซอร์โวมอเตอร์	70
รูปที่ 6.5 แสดงภาพแกนกลเมื่อหีบวัตถุและภาพแกนกลที่นำวัตถุไปยังตำแหน่งปลายทาง	70

สารบัญรูปภาพ (ต่อ)

รูปที่ 6.6 กราฟแสดงผลการทดลองที่ 3 ใน 2 มิติ

หน้า

71



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงการจัดสรรขาสัญญาณพอร์ต P1 สำหรับใช้งานร่วมกับ PCA	15
ตารางที่ 2.2 แสดงคุณสมบัติของรีจิสเตอร์ CPS0:CPS1	18
ตารางที่ 2.3 แสดงการกำหนดค่าสำหรับเลือกโหมดการทำงานของ โมดูล PCA	21
ตารางที่ 5.1 แสดงค่ามุมของเซอร์โวมอเตอร์ที่ความกว้างพัลส์ค่าต่างๆ	59
ตารางที่ 6.1 แสดงผลการทดลองที่ 1	68
ตารางที่ 6.2 แสดงผลการทดลองที่ 4	72
ตารางที่ 6.3 แสดงผลการทดลองที่ 5.1	73
ตารางที่ 6.4 แสดงผลการทดลองที่ 5.2	74
ตารางที่ 6.5 แสดงผลการทดลองที่ 6	75
ตารางที่ 6.6 แสดงผลการทดลองที่ 7	76
ตารางที่ 6.8 แสดงผลการทดลองที่ 8	77

บทที่ 1

บทนำ

เนื่องจากในปัจจุบันนี้วิวัฒนาการความก้าวหน้าทางเทคโนโลยีได้ก้าวไกลไปอย่างมาก กระบวนการต่างๆ ได้ใช้เทคโนโลยีเข้าไปช่วยอย่างกว้างขวาง โดยเฉพาะในระบบอุตสาหกรรมนั้นได้มีการพัฒนาระบบต่างๆ ไม่ว่าจะเป็นการจัดการ โรงงานที่คือองอาศัยคอมพิวเตอร์เป็นตัวเก็บและวิเคราะห์ข้อมูลต่างๆ การประหยัดพลังงาน การรักษาสิ่งแวดล้อม รวมทั้งกระบวนการผลิตที่มักจะเป็นการผลิตแบบอัตโนมัติโดยใช้หุ่นยนต์เข้าช่วย ซึ่งต้องอาศัยการพัฒนาความรู้ทางด้านต่างๆ เพื่อให้ได้ประสิทธิภาพที่ดีต้นทุนที่ถูกลงซึ่งจะเห็นว่าเทคโนโลยีต่างๆ ส่วนใหญ่จะนำเข้าจากต่างประเทศมีราคาแพง ผู้จัดทำโครงการจึงมีแนวความคิดที่จะศึกษาและพัฒนาระบบการทำงานที่เป็นอัตโนมัติซึ่งเรียกว่า หุ่นยนต์ (Robot)

การทำงานของหุ่นยนต์(Robot) ต้องอาศัยส่วนต่างๆ แต่ที่สำคัญและขาดไม่ได้คือระบบขับเคลื่อนและการควบคุมและที่นิยมใช้ก็มักจะเป็นระบบขับเคลื่อนด้วยมอเตอร์ไฟฟ้าและควบคุมด้วยคอมพิวเตอร์ซึ่งระบบควบคุมนั้นจะต้องสามารถควบคุมตำแหน่งได้ส่วนของชุดขับเคลื่อนนั้นก็ขึ้นอยู่กับขนาดของมอเตอร์ที่ใช้

ในส่วนของการควบคุมแบบดิจิทัลนั้นจะใช้ไมโครคอนโทรลเลอร์เป็นตัวที่รับคำสั่งจากคอมพิวเตอร์และประมวลผลคำสั่งนั้นเพื่อให้มอเตอร์หมุนไปในทิศทางต่างๆ ในส่วนของการส่งคำสั่งจากคอมพิวเตอร์นั้นก็จะเป็นการเขียนโปรแกรมคอมพิวเตอร์เพื่อที่จะติดต่อระหว่างผู้ใช้กับระบบนั่นเอง

1.1 วัตถุประสงค์

1. เขียนโปรแกรมของไมโครคอนโทรลเลอร์เพื่อไปควบคุม Servo Motor ได้
2. เขียนโปรแกรมส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรมเพื่อควบคุมการทำงานของแขนกล
3. สามารถออกแบบและสร้างวงจรเพื่อใช้ติดต่อควบคุมระหว่างเครื่องคอมพิวเตอร์กับแขนกล
4. สร้างแขนกลและทำการพัฒนาโปรแกรมที่ใช้ในการควบคุมได้

1.2 ขอบเขตของโครงการ

1. ออกแบบวงจรและเขียนโปรแกรมสำหรับควบคุมการทำงานของแขน
2. ออกแบบแขนกลสำหรับทดสอบการเคลื่อนไหว
3. สร้างแขนกลสำหรับทดสอบการเคลื่อนไหว
4. แขนกลสามารถควบคุมได้จากการระบุพิกัดที่ต้องการ



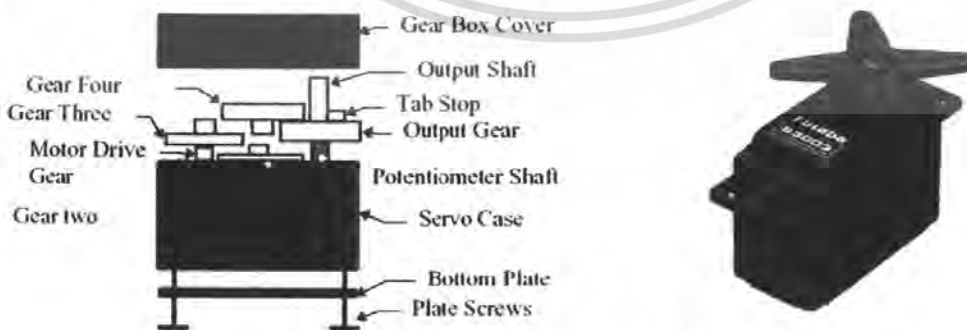
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ดีซีเซอร์โวมอเตอร์(DC SERVO MOTOR)

Servo motor คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์ และ ส่วนควบคุม ต่างๆ ไว้ใน โมดูลเดียวกัน หรือ ภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสาย ต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC,GNDและสายสัญญาณควบคุม(Control Line) ซึ่งสามารถควบคุม ให้มอเตอร์หมุนซ้าย หรือ ขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็น สัญญาณ พัลส์วามอด (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลท์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือ จะมีขนาดเล็ก น้ำหนักเบา, ให้แรงบิดสูง ,กินพลังงานน้อย และ สามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้ โดยตรงไม่จำเป็นต้องต่อวงจรขับ(Driver) อื่นๆ เพราะ มอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายใน อยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุน ไปในตำแหน่ง หรือ ทิศทางองศาที่ต้องการได้ โดย อาศัยสัญญาณความกว้างพัลส์ ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้แค่เพียงในช่วง ประมาณ 180° หรือ ครึ่งรอบเท่านั้น หรือ บางรุ่นอาจหมุนได้ถึง 210° แต่จะ ไม่สามารถหมุนเป็นวงรอบ ได้เนื่องจาก โครงสร้างภายในจะประกอบด้วย ตัวด้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบ ตำแหน่งการหมุนของมอเตอร์ และ ตัวด้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัว ด้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้น เซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียง แค่ประมาณ 180 องศา หรือ ครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดกับตัวด้านทานปรับค่าได้ แต่ถ้าหากเราต้องการ ให้มอเตอร์หมุนเป็นวงรอบ (360°) นั้นก็สามารถทำได้ โดยจะต้องทำการปรับแต่ง (Modify) คัดแปลงชิ้นส่วนบางอย่างของมอเตอร์ ซึ่งวิธีการต่างๆ จะได้กล่าวไว้ในภายหลัง

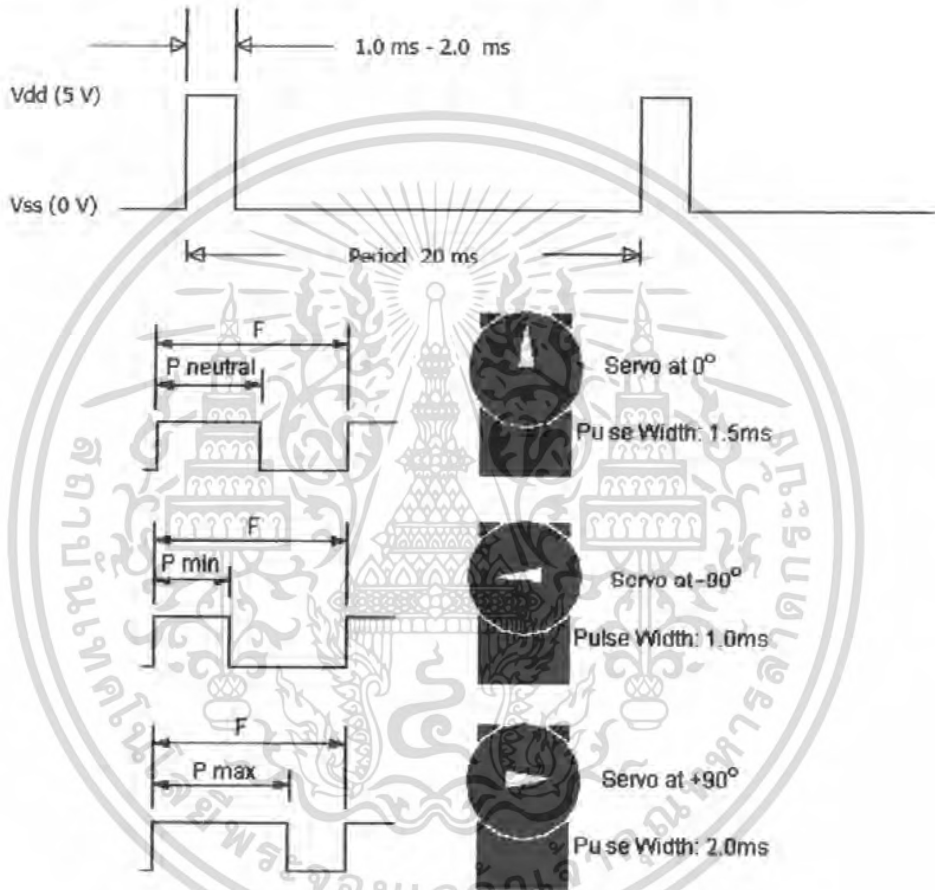


รูปที่ 2.1 ภาพตัวอย่าง เซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 หลักการทำงานของเซอร์โวมอเตอร์

การควบคุมการทำงานของเซอร์โวมอเตอร์ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุดดังรูปคือ



รูปที่ 2.2 ภาพแสดงมุมของเซอร์โวมอเตอร์ที่ความกว้างพัลส์ต่างกัน

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือจุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกา

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่นๆนั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆโดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม - 45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 ms เป็นต้น และสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 ms(Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้โดยหลักการก็คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ที่จ่ายให้กับมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาของวงจร RC ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากันมอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของ วงจร RCเปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control line) มอเตอร์จึงจะหยุดหมุน

2.1.2 การกำหนดความเร็วของเซอร์โวมอเตอร์

ในการกำหนดความเร็วของ Servo motor นั้นเราสามารถกำหนดได้โดยการป้อนค่าสัญญาณความกว้างให้กับ Servo motor เป็นระยะทุก 20 ms เช่นถ้าหากเราต้องการให้ Servo motor หมุนจากตำแหน่งตรงกลางไปยังตำแหน่งซ้ายสุดซึ่งค่าสัญญาณความกว้าง pulse ของตำแหน่งตรงกลางคือ 1.5 ms และค่าความกว้างของสัญญาณความกว้าง pulse ที่ตำแหน่งซ้ายสุดคือ 0.5 ms เราสามารถที่จะกำหนดความเร็วของ Servo ได้ดังนี้

ป้อน 1.5 ms (20 ms ผ่านไป) ป้อน 0.5 ms (1)

ป้อน 1.5 ms (20 ms ผ่านไป) ป้อน 1.0 ms (20 ms ผ่านไป) ป้อน 0.5 ms (2)

จะได้ว่าแบบที่สองจะเคลื่อนที่ได้ช้ากว่าแบบที่ 1 ถ้าต้องการจะให้ Servo เคลื่อนที่ได้ช้าลงหรือเร็วขึ้นก็สามารถทำได้โดยการแก้ไขจำนวนครั้งในการป้อนนั่นเอง

2.1.3 ข้อดีของเซอร์โวมอเตอร์

1. มีน้ำหนักเบา ซึ่ง Servo ขนาดมาตรฐานมีน้ำหนักเพียง 49 กรัม
2. มีแรงบิดสูง เมื่อเปรียบเทียบกับ Stepper motor
3. ใช้สายสัญญาณในการส่งเพียง 1 เส้น อีก 2 เส้นเป็น Vcc กับ ground
4. กินพลังงานน้อย
5. สามารถควบคุมด้วย logic ที่เป็น TTL ได้โดยตรง

2.2 การติดต่อสื่อสาร

การอินเทอร์เฟสกับคอมพิวเตอร์หรือไมโครโปรเซสเซอร์ คือ การทำงานติดต่อกันระหว่างซีพียูกับอุปกรณ์อื่นๆกับการโอนถ่ายข้อมูลระหว่างอุปกรณ์ต่างๆ นอกเหนือจากจะต้องทำงานติดต่อกับ RAM,ROM แล้วยังต้องมีการติดต่อกับอุปกรณ์ภายนอกที่มีการส่งข้อมูลอินพุต,เอาต์พุตอีกทางหนึ่ง ซึ่งเป็นการเพิ่มประสิทธิภาพให้ระบบสมบูรณ์ ในระบบค่างของอุปกรณ์อิเล็กทรอนิกส์ จะทำงานต่อเนื่องเป็นลูกโซ่ ดังเช่น การส่งรับข้อมูลจากซีพียูไปยังส่วนอื่นๆ เป็นต้น



รูปที่ 2.3 ภาพแสดงการรับส่งข้อมูล

การที่จะโอนย้ายข้อมูลทุกตัวนั้นจะต้องมีแหล่งที่ส่งข้อมูล และแหล่งที่รับข้อมูลสำหรับขบวนการเหล่านั้น จะมีส่วนที่สำคัญว่า ข้อมูลนั้นเป็น แอดเดรสหรือว่าเป็นค่าค่า จะส่งไปยังจุดไหน ตัวอย่างเช่น ส่ง ไปยังหน่วยความจำ หรืออุปกรณ์อินพุต/เอาต์พุต และจะส่งเมื่อไร การทำงานเหล่านี้โดยทั่วไป จะต้องมีสัญญาณ ในการตรวจสอบอุปกรณ์ว่าพร้อมที่จะส่ง/รับข้อมูล หรือยังก่อนเสมอ เนื่องจากจุดที่ส่งและรับ ข้อมูล จะต้องมีสัญญาณตรวจสอบความพร้อมเสมอเพื่อที่จะให้ข้อมูลที่เรารู้

งานนั้นๆ เป็นระเบียบ ตัวอย่างเช่น ส่งข้อมูลจากซีพียูไปที่อุปกรณ์รอบข้าง เป็นต้น ซึ่งจุดรับส่งคู่หนึ่งๆ อาจจะเป็นระหว่างซีพียูด้วยกัน หรือ ซีพียูกับหน่วยความจำ หรือ ซีพียูกับอุปกรณ์รอบข้าง หรือ ระหว่างอุปกรณ์รอบข้างด้วยกัน หรือ ระหว่างหน่วยความจำกับอุปกรณ์รอบข้าง ก็ได้ สำหรับข้อมูลที่โอนย้ายไปมานั้นจะอยู่ในลักษณะของเลขฐานสอง ตัวอย่างเช่น $\rightarrow 01101100_2$ ซึ่งเลขแต่ละตัวจะแทนด้วย 1 bit อาจเป็น 8 bit หรือ 16 bit ก็ขึ้นอยู่กับของระบบนั้นๆ ถ้าหากเป็นการต่อจากพอร์ตพีซีไม่ว่าจะเป็น Serial หรือ Parallel ในสัญญาณที่ส่งมาจะมีระบบแรงดันไฟฟ้า

2.2.1 ความรู้เบื้องต้นเกี่ยวกับพอร์ต RS-232(Serial Port)

การสื่อสารแบบอนุกรม นับว่ามีความสำคัญ ต่อการใช้งาน ไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้เป็นพิมพ์ และจอภาพของ PC เป็น อินพุต และ เอาต์พุต ในการติดต่อ หรือ ควบคุม ไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อย เพียง 3 เส้นเท่านั้น คือ

- สายส่งสัญญาณ TX
- สายรับสัญญาณ RX
- และสาย GND

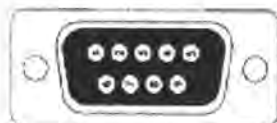
โดยปกติพอร์ตอนุกรม RS-232C จะสามารถต่อสายได้ยาว 50 ฟุต โดยประมาณ ขึ้นอยู่กับ ชนิดของสายสัญญาณ, ระยะทาง, และ ปริมาณ สัญญาณ ครอบคลุม



รูปที่ 2.4 ภาพแสดงลักษณะของ

พอร์ตอนุกรมของ PC DB9 ตัวผู้(Male) และ พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย(Female)

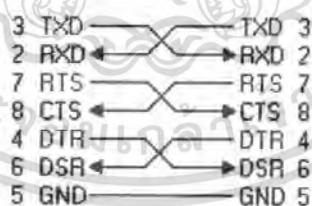
2.2.2 การจัดขา ของคอนเน็กเตอร์ อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ



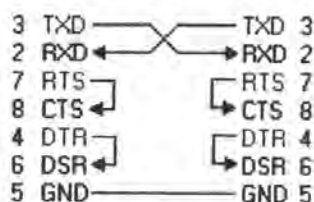
รูปที่ 2.5 ภาพของDB9 ตัวผู้ เมื่อมองจากด้านหลัง

Pin	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

2.2.3 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9



การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem



การต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การทำงานของขาสัญญาณ DB9

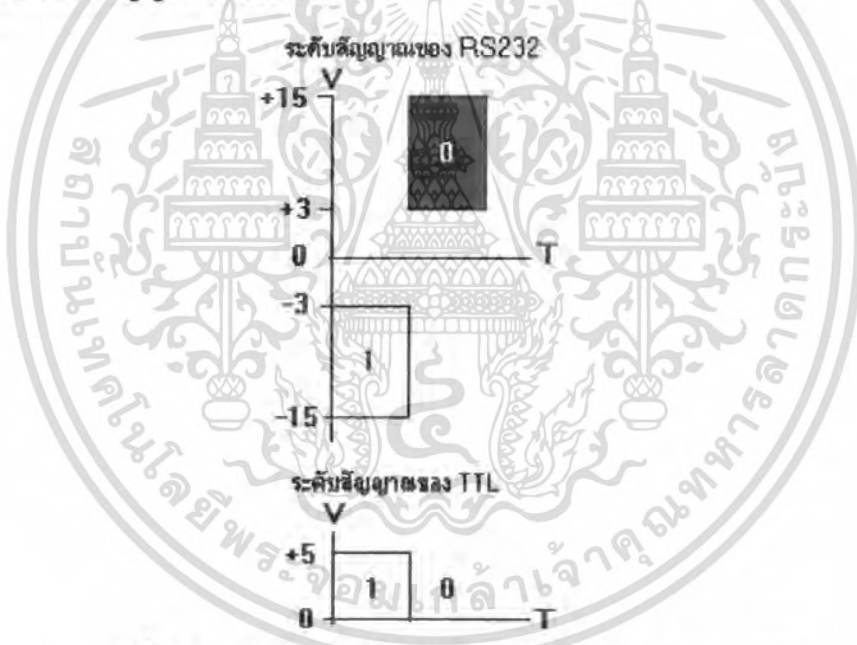
TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

DTR แสดงสถานะพอร์ตว่าเปิดใช้งาน ,DSR ตรวจสอบว่าพอร์ต ที่ติดต่อด้วย เปิดอยู่หรือไม่

- เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าการติดต่อด้วย
- ในขณะที่เดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่ RTS แสดงสถานะพอร์ตว่าการส่งข้อมูล ,CTS ตรวจสอบว่าพอร์ตที่ติดต่อด้วย ต้องการส่งข้อมูลหรือไม่
- เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF
- ในขณะที่เดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่ GND ขา ground

2.2.5 ระดับสัญญาณของ RS232



รูปที่ 2.6 ภาพแสดงระดับสัญญาณแบบ RS-232 และแบบ TTL

สัญญาณรบกวนที่เกิดขึ้น ในสายนำสัญญาณ มักจะมีแรงดันเป็นบวก เมื่อเทียบกับกราวด์ - เพื่อป้องกันสัญญาณรบกวนนี้ จึงออกแบบแรงดัน ของลอจิก “1” เป็นลบ คืออยู่ในช่วง -3V ถึง -15V ส่วนแรงดันของลอจิก “0” อยู่ในช่วง +3V ถึง +15V และเหตุที่ ระดับสัญญาณ ของ RS232 อยู่ในช่วง +15V ถึง -15V ก็เพื่อให้ต่อสายสัญญาณ ไปได้ไกลขึ้นดังนั้นจึงจำเป็นต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 อัตราการส่งข้อมูล (Baud rate)

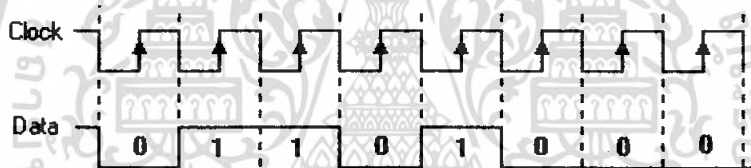
คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิตต่อวินาทีเช่น 300, 1,200, 2,400, 4,800 , 9,600,14,400 ,19,200, 38,400 ,56,000 เป็นต้น
การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับ ชนิดของสายสัญญาณ, ระยะทาง,และปริมาณสัญญาณรบกวน

2.2.7 รูปแบบการสื่อสารแบบอนุกรม

มีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส(Asynchronous)

การสื่อสารแบบซิงโครนัส (Synchronous)

-การรับส่งข้อมูล จะมีสัญญาณนาฬิกา ซึ่งเป็นตัวกำหนด จังหวะเวลา การส่งข้อมูลร่วมอยู่ด้วยอีกเส้นหนึ่งใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด



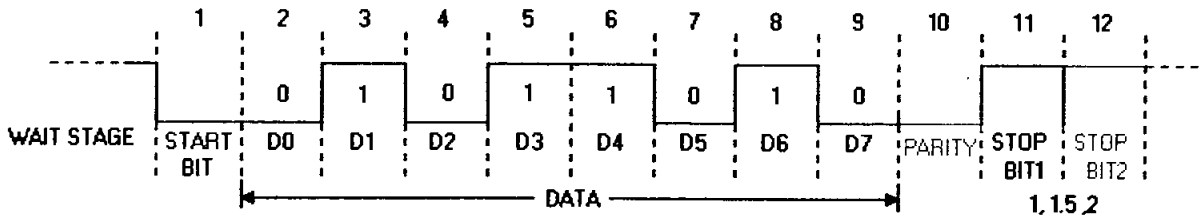
รูปที่ 2.7 ภาพแสดงรูปแบบข้อมูลแบบซิงโครนัส

การสื่อสารแบบอะซิงโครนัส (Asynchronous)

- การรับส่งข้อมูล โดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วยแต่จะใช้ให้ตัวส่งและตัวรับ มีอัตราการส่งข้อมูลที่เท่ากัน

รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
2. บิตข้อมูล (Data) มีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity bit) มีขนาด 1 บิตหรือไม่มี
4. บิตหยุด (Stop bit) มีขนาด 1, 1.5, 2 บิต

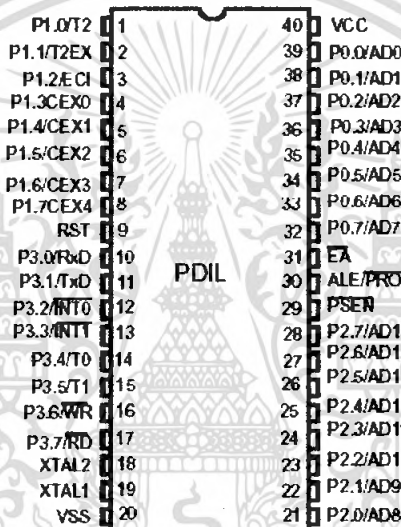


รูปที่ 2.8 ภาพแสดงรูปแบบข้อมูลแบบอะซิงโครนัส

- เมื่อไม่มีการส่งข้อมูล ขา data จะมีสถานะเป็น โลจิก "1" หรือ สถานะหยุดรอ (Waiting stage)
 - เมื่อเริ่มต้นส่งข้อมูลจะให้ขา data เป็น โลจิก "0" เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น (Start bit)
 - จากนั้นก็จะเริ่มต้นส่งข้อมูล โดยส่งบิตต่ำ ไปก่อน (LSB)
 - แล้วตามด้วยพาริตีบิต (จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับการคิดตั้งค่า ของทั้งสองฝ่าย)
 - สุดท้ายตามด้วยโลจิก "1" อย่างน้อย 1 บิต (มีขนาด 1, 1.5, หรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล
- การรับและส่งข้อมูลแบบอนุกรมยังแบ่งออกเป็นลักษณะการใช้งานได้ 3 แบบคือ
- 1). แบบซิมเพลกซ์ (Simplex) เป็นการส่ง หรือรับข้อมูล แบบทิศทางเดียว เท่านั้น
 - 2). แบบฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการส่งและรับข้อมูลแบบสลับกัน คือเมื่อด้านหนึ่งส่งอีกด้านหนึ่งเป็นฝ่ายรับ สลับกัน ไม่สามารถรับ-ส่งในเวลาเดียวกันได้
 - 3). แบบฟูลดูเพลกซ์ (Full Duplex) สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

2.3 ไมโครคอนโทรเลอร์ MCS-51

สำหรับวิธีการสร้างสัญญาณ Pulse สำหรับควบคุมการหมุนของ DC SERVO MOTOR นั้นสามารถทำได้หลายวิธี ขึ้นอยู่กับเทคนิคการเขียนโปรแกรมของแต่ละคน เช่น การใช้วิธีการหน่วงเวลาในการ SET และ RESET สถานะของสัญญาณที่ Port Pin ของ CPU เพื่อให้ได้สัญญาณ Pulse ตามค่าเวลาที่กำหนดไว้จากโปรแกรม หรืออาจใช้ Timer สำหรับทำการนับเวลาเพื่อสร้างเป็นความถี่ของสัญญาณ Pulse ตามต้องการแต่สำหรับ CPU เบอร์ P89C51RD2 ของ Philips หรือ T89C51RD2 ของ ATMEL นั้น จะมีวงจรสำหรับสร้างสัญญาณ PWM บรรจุไว้ภายในตัว CPU ด้วยอยู่แล้ว



รูปที่ 2.9 แสดงการจัดขาสัญญาณของ P89C51RD2/T89C51RD2

ซึ่งการใช้วงจรถ่าย Pulse Width Modulation หรือ PWM ในการสร้างสัญญาณ เพื่อควบคุมการหมุนของ DC SERVO MOTOR นั้น นับว่าเป็นวิธีการและแนวทางที่ง่ายและมีประสิทธิภาพมากที่สุด เนื่องจากสามารถลดความยุ่งยากในการเขียนโปรแกรมเพื่อสร้างเป็นสัญญาณ Pulse ไปได้เป็นอย่างมาก และ CPU เองก็ไม่ต้องสูญเสียเวลาในการไปประมวลผลเพื่อคำนวณค่าเวลาในการควบคุม Port Pin เพื่อสร้างความถี่ให้ยุ่งยาก เนื่องจากระบบฮาร์ดแวร์ PWM นั้น เมื่อเราทำการกำหนดค่าต่างๆให้กับรีจิสเตอร์ ไปได้แล้ว วงจร PWM ก็จะสามารถสร้างสัญญาณ PWM ออกมาได้อย่างต่อเนื่อง โดยไม่รบกวนการทำงานของ CPU อีกเลย แต่เมื่อต้องการเปลี่ยนแปลงค่าความถี่หรือคาบเวลาของ PWM ก็เพียงแค่ทำการกำหนดค่าให้กับรีจิสเตอร์ใหม่เท่านั้น ซึ่งจะเห็นได้ว่าการใช้วงจรถ่าย PWM ในการสร้างสัญญาณ Pulse เพื่อควบคุมการทำงานของมอเตอร์นั้น สามารถลดความยุ่งยากในการเขียนโปรแกรมไปได้มากเลยทีเดียว ซึ่งในที่นี้จะขอแนะนำถึงวิธีการใช้งานระบบ Timer ของ PCA ในการสร้างสัญญาณ PWM แบบพอสังเขปเพื่อเป็นแนวทางประกอบการใช้งานดังนี้

2.3.1 การทำงานของวงจรถ่าย Programmable Counter Array (PCA)

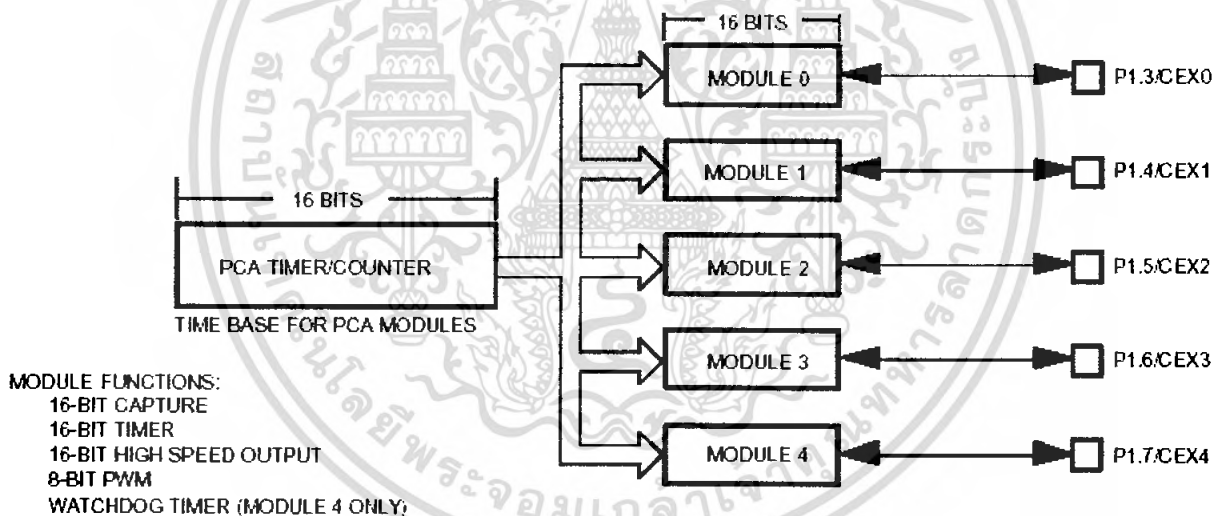
วงจรถ่าย Programmable Counter Array หรือ PCA เป็นส่วนเพิ่มเติมของ ระบบ Timer ใน MCS51 ซึ่งวงจรถ่ายส่วนนี้ คำนวณปรกติแล้วจะ ไม่มีอยู่ใน โครงสร้างมาตรฐานของ CPU ในตระกูล MCS51 ทั่วๆ ไป ซึ่งระบบการทำงานของวงจรถ่าย Timer/Counter แบบ PCA นี้จะมีขีดความสามารถที่สูงกว่าระบบการทำงานของ Timer/Counter ปรกติของ MCS51 ซึ่งจะช่วยให้ลดความยุ่งยากของโปรแกรมในการสั่งงาน CPU และช่วยลดภาระการทำงานของ CPU ได้เป็นอย่างมาก โดยลักษณะโครงสร้างของวงจรถ่าย PCA จะประกอบไปด้วย ส่วนของวงจรถ่าย Timer/Counter หลัก (PCA Timer) และเชื่อมต่อกับวงจรถ่ายของ Capture/Compare (PCA Module) จำนวน 5 ชุด โดยเชื่อมต่อกันแบบเรียงลำดับ (Array) ซึ่งการทำงานของ PCA Module แต่ละชุดจะอ้างอิงความถี่จากวงจรถ่าย PCA Counter ชุดเดียวกัน หรือใช้ฐานเวลาร่วมกันซึ่งระบบฐานเวลาหลัก (PCA Timer) ของวงจรถ่าย PCA นั้น จะสามารถทำการ โปรแกรมหรือกำหนดเงื่อนไขในการนับ ได้ว่าจะให้นับจากสัญญาณนาฬิกา Input แบบใด โดยสามารถกำหนดได้ 4 แบบ โดยการเลือกกำหนดจากบิต CPS1:CPS0 ของรีจิสเตอร์ CMOD ดังนี้คือ

- กำหนดให้นับจากความถี่สัญญาณนาฬิกา Oscillatorหาร 12 (หาร 6 ในโหมด X2)
- กำหนดให้นับจากความถี่สัญญาณนาฬิกา Oscillatorหาร 4 (หาร 2 ในโหมด X2)
- กำหนดให้นับจากการ Overflow ของวงจรถ่าย Timer0
- กำหนดให้นับจากสัญญาณนาฬิกา Input ภายนอกที่ป้อนให้กับขาสัญญาณ ECI หรือ P1.2

ส่วนการทำงานของวงจร PCA Module ทั้ง 5 ชุด นั้น แต่ละ โมดูลสามารถกำหนดหน้าที่การทำงานของวงจรได้อย่างอิสระ โดยแต่ละ โมดูล เองก็ยังสามารถ โปรแกรมการทำงานได้หลายๆหน้าที่ ดังนี้

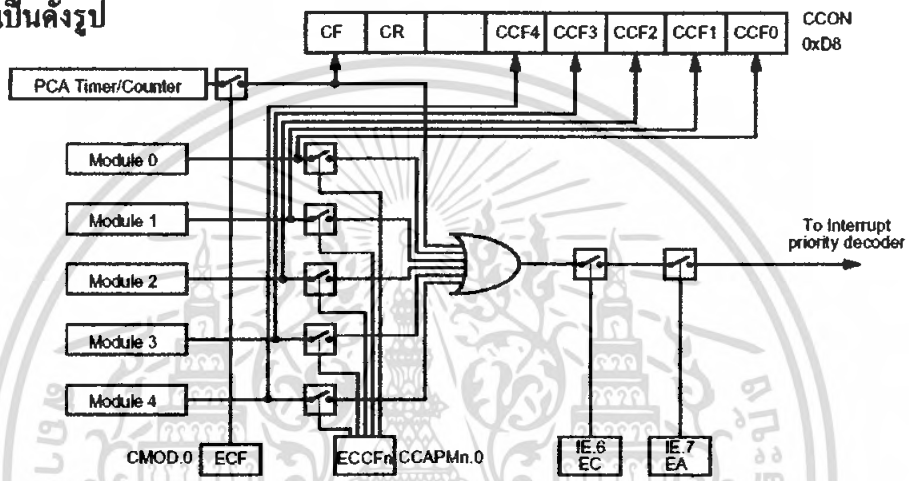
- กำหนดให้ทำการตรวจจับสัญญาณ Input (Capture) โดยสามารถเลือกตรวจจับได้ทั้งสัญญาณที่เป็นแบบขอบขาขึ้น (Rising Edge) และ ขอบขาลง(Falling Edge)
- กำหนดให้ทำการสร้างสัญญาณ Output ความเร็วสูง แบบ High Speed Output
- กำหนดให้ทำการสร้างสัญญาณ Pulse Width Moduration หรือ PWM
- กำหนดให้ทำหน้าที่เป็น Timer สำหรับนับสัญญาณนาฬิกา (Software Timer)

นอกจากนี้แล้ว ในส่วนของโมดูล PCA ชุดที่4 นั้น ยังมีความสามารถพิเศษกว่าโมดูลอื่นๆ กล่าวคือ สามารถทำการ โปรแกรมหน้าที่ให้เป็น Watch-Dog Timer ได้อีกด้วย



รูปที่ 2.10 แสดงลักษณะของระบบ PCA Timer

โดยเมื่อวงจร PCA แต่ละ โมดูล ถูก โปรแกรมให้ทำหน้าที่แบบ Input Capture หรือ SoftwareTimer หรือ High Speed Output นั้น การทำงานของวงจร PCA แต่ละชุด สามารถส่งสัญญาณ ไปร้องขอการ Interrupt จาก CPU ได้ด้วย โดยที่โมดูลของ PCA ทั้ง 5 ชุดนี้จะใช้สัญญาณ Interrupt ร่วมกัน ซึ่งไม่ว่าจะเกิดการร้องขอการ Interrupt จาก โมดูลชุดใด ก็จะมีตำแหน่ง Vector ในการบริการ Interrupt ที่เหมือนกัน แต่ผู้ใช้งานสามารถทำการตรวจสอบแหล่งที่มาของการร้องขอการ Interrupt ได้ ว่ามาจากโมดูลชุดใดจากบิต CCFx ในรีจิสเตอร์ CCON โดยลักษณะการสร้างสัญญาณ Interrupt ของ PCA Timer เป็นดังรูป



รูปที่ 2.11 แสดงแผนผังการสร้างสัญญาณ Interrupt ของ PCA Timer

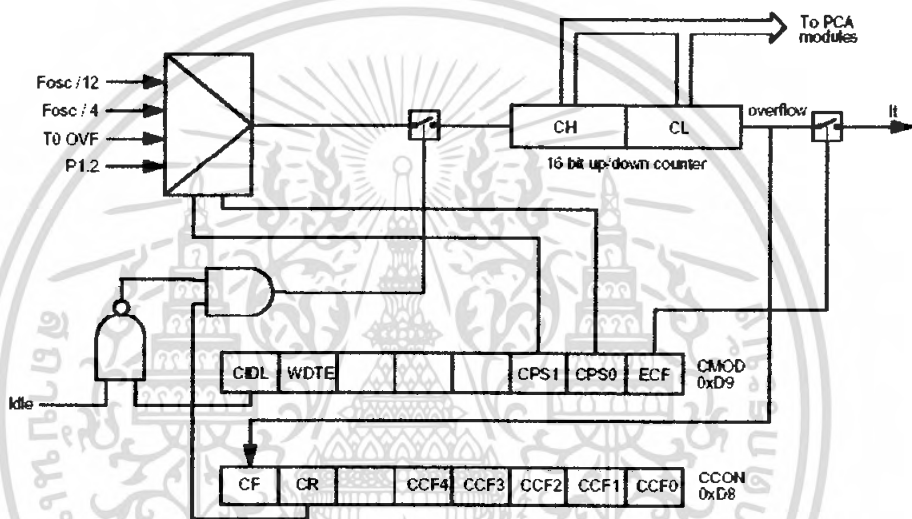
สำหรับการทำงานของ PCA ทั้ง 5 ชุดนั้น จะใช้สัญญาณจาก พอร์ต P1 (P1.2-P1.7) เป็นจุดผ่านของ สัญญาณในการทำงาน ซึ่งเมื่อไม่มีการ โปรแกรมการทำงานให้กับวงจร PCA แล้ว ขาสัญญาณของ พอร์ต P1 ก็ยังสามารถนำไปใช้งานเป็น Input หรือ Output ทั่วๆ ไปได้ตามต้องการ แต่ในกรณีที่มีการ โปรแกรมการทำงานของวงจร PCA ไว้ ขาสัญญาณของพอร์ต P1 ก็จะถูกควบคุมการทำงานโดยวงจร PCAแต่ละชุด โดยสามารถแสดงให้เห็นได้ดังนี้คือ

การทำงานของวงจร PCA	ขาสัญญาณ พอร์ต P1
16 บิต Counter	P1.2 / ECI
16 บิต Module0	P1.3 / CEX0
16 บิต Module1	P1.4 / CEX1
16 บิต Module2	P1.5 / CEX2
16 บิต Module3	P1.6 / CEX3
16 บิต Module4	P1.7 / CEX4

ตารางที่ 2.1 แสดงการจัดสรรขาสัญญาณพอร์ต P1 สำหรับใช้งานร่วมกับ PCA

2.3.2 การทำงานของ PCA Timer

วงจรวางจร PCA Timer จัดเป็นวงจรมับ ขนาด 16 บิต ทำหน้าที่สำหรับส่งสัญญาณ Output ที่ได้จากการนับไปยังโมดูลของ PCA ทั้ง 5 ชุด โดยทำงานของวงจรวางจร PCA Timer นั้น จะถูกควบคุมการทำงานจากรีจิสเตอร์ CMOD และ CCON ซึ่งวงจรวางจร PCA โมดูลทั้ง 5 ชุด นั้น จะใช้ฐานเวลาจากวงจรวางจร PCA Timer ชุดเดียวกันทั้งหมด โดยการนับของ PCA Timer นั้น จะกำหนดลักษณะของสัญญาณ Input ได้ 4 แบบ โดยกำหนดผ่านทางบิต CPS0:CPS1 ของรีจิสเตอร์ CMOD ซึ่งลักษณะโครงสร้างของวงจรวางจร PCA Timer นั้นสามารถแสดงให้เห็นได้ดังรูป



รูปที่ 2.12 แสดง ลักษณะ โครงสร้างของวงจรวางจร PCA Timer

จากรูปจะเห็นได้ว่า บิต CPS1:CPS0 ของรีจิสเตอร์ CMOD จะถูกใช้งานร่วมกันสำหรับทำหน้าที่เลือกแหล่งกำเนิดสัญญาณนาฬิกาสำหรับป้อนเป็น Input การนับของวงจรวางจร PCA Timer (CH:CL) โดยการทำงานของวงจรวางจร PCA Timer นั้นจะเริ่มต้นทำงานได้ก็ต่อเมื่อทำการเซตบิต CR ในรีจิสเตอร์ CCON ให้มีค่าเป็น “1” โดยผลการนับของวงจรวางจร PCA Timer นั้น เมื่อเกิดการ Overflow ขึ้น สามารถที่จะควบคุมให้การ Overflow ของวงจรมับเพื่อส่งสัญญาณร้องขอการ Interrupt ไปยัง CPU ได้จากควบคุมของบิต ECF ในรีจิสเตอร์ CMOD โดยลักษณะการนับของวงจรวางจร PCA Timer นั้น จะเป็นวงจรมับแบบอิสระไม่สามารถจะทำการโปรแกรมค่าการนับให้กับวงจรวางจรได้โดยตรง กล่าวคือ เมื่อทำการอนุญาตให้วงจรวางจร PCA Timer เริ่มต้นทำงาน (เซตบิต CR ใน CCON เป็น “1”) การนับของ PCA Timer ก็จะเริ่มต้นนับต่อเนื่องกันไปไม่รู้จักจบ ตามค่า Input ที่ได้รับ จนกว่าจะมีการสั่งหยุดการทำงาน โดยผลการนับของวงจรวางจร PCA Timer นั้น จะอยู่ในรีจิสเตอร์ CH:CL แต่อย่างไรก็ตามจำนวนค่าการนับของวงจรวางจร PCA Timer นั้น

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เราสามารถทำการโปรแกรมจำนวนครั้งในการนับให้กับวงจร PCA Time ได้ด้วยวิธีการทางฮาร์ดแวร์ โดยใช้เงื่อนไขของการ Overflow เป็นตัวกำหนด ตัวอย่างเช่น ถ้าต้องการให้วงจร PCA Timer ทำการนับจำนวน 500 ครั้ง ก็สามารถทำได้โดยการกำหนดค่าเริ่มต้นให้กับ CH:CL ด้วยค่าการนับสูงสุดหักลบออกจากจำนวนครั้งที่ต้องการคือ 500 (65536-500) และจะต้องการกำหนดค่านี้ให้กับ CH:CL ทุกๆครั้งที่เกิดการ Overflow ขึ้น ไม่เช่นนั้นแล้วการนับของ PCA Timer จะกลับไปเริ่มต้นใหม่จากศูนย์แทน

2.3.3 คุณสมบัติของรีจิสเตอร์ CMOD

สำหรับรีจิสเตอร์ CMOD เป็นรีจิสเตอร์ขนาด 8 บิต ใช้สำหรับกำหนดโหมดการทำงานของ PCA Timer โดยรีจิสเตอร์ CMOD นี้จะจัดไว้ในส่วนของ SFR (Special Function Register) ตำแหน่งแอดเดรส D9H ซึ่งในการเข้าถึงรีจิสเตอร์ตัวนี้สามารถเข้าถึงได้แบบ Byte ด้วยวิธีการแบบ Direct หรือการอ้างตำแหน่งแอดเดรสของรีจิสเตอร์โดยตรงเพียงอย่างเดียวเท่านั้น ไม่สามารถใช้วิธีการเข้าถึงแบบระดับบิตได้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
CIDL	WDTE	-	-	-	CPS1	CPS0	ECF

รูปที่ 2.13 แสดงโครงสร้างของ CMOD Register

- **CIDL** เป็นบิต Counter Idle Control ใช้สำหรับควบคุมการทำงานของ PCA Counter เมื่อ CPU เข้าทำงานในโหมดประหยัดพลังงาน (Idle Mode) โดยถ้ากำหนดให้บิต CIDL นี้มีค่าเป็น "0" จะเป็นการปล่อยให้วงจร PCA Counter ทำงานต่อเนื่องไปตลอด ถึงแม้ว่า CPU จะหยุดทำงานแล้วก็ตาม แต่ถ้ากำหนดให้บิต CIDL นี้มีค่าเป็น "1" จะหมายถึง การบังคับให้วงจร PCA หยุดการทำงานทันที เมื่อ CPU เริ่มต้นเข้าทำงานใน Idle โหมด
- **WDTE** เป็นบิต Watchdog Timer Enable ใช้สำหรับควบคุมการทำงานของวงจร Watchdog ที่สร้างจากวงจร PCA โมดูลที่ 4 โดยถ้ากำหนดให้บิตนี้มีค่าเป็น "0" จะเป็นการปิดการทำงานของ PCA Watchdog แต่ถ้ากำหนดให้บิต WDTE นี้มีค่าเป็น "1" จะเป็นการเปิดการทำงานของวงจร PCA Watchdog

- CPS0:CPS1 เป็นบิต PCA Count Pulse Select ใช้ร่วมกันสำหรับเลือกแหล่งของสัญญาณ Input ที่จะส่งให้กับวงจร PCA Timer ทำการนับ โดยมีคุณสมบัติดังนี้คือ

CPS	CPS	สัญญาณที่ป้อนให้กับ PCA Counter
1	0	
0	0	Internal Clock หรือ $F_{osc}/12$ ($F_{osc}/6$ ใน X2 Mode)
0	1	Internal Clock หรือ $F_{osc}/4$ ($F_{osc}/2$ ใน X2 Mode)
1	0	การ Overflow ของ Timer-0
1	1	External Clock ที่ป้อนให้กับขาสัญญาณ ECI หรือ P1.2 (ความถี่สูงสุด $F_{osc}/8$)

ตารางที่ 2.2 แสดงคุณสมบัติของรีจิสเตอร์ CPS0:CPS1

- ECF เป็นบิต PCA Enable Counter Overflow Interrupt ใช้สำหรับควบคุมการร้องขอการ Interrupt ของวงจร PCA โดยถ้ากำหนดให้บิต ECF นี้มีค่าเป็น "0" จะหมายถึง เป็นการปิดการร้องขอการ Interrupt จากวงจร PCA Timer แต่ถ้ากำหนดให้บิต ECF นี้มีค่าเป็น "1" จะเป็นการอนุญาตให้บิต CF ในรีจิสเตอร์ CCON ทำการร้องขอการ Interrupt ไปยัง CPU เมื่อการทำงานของโมดูล PCA Timer เกิดการ Overflow ขึ้น

2.3.4 คุณสมบัติของรีจิสเตอร์ CCON

สำหรับรีจิสเตอร์ CCON เป็นรีจิสเตอร์ขนาด 8 บิต ใช้สำหรับแสดงสถานะและควบคุมการทำงานของวงจร PCA Timer โดยรีจิสเตอร์ CCON นี้จะจัดไว้ในส่วนของ SFR (Special Function Register) ตำแหน่งแอดเดรส D8H ซึ่งในการเข้าถึงรีจิสเตอร์ตัวนี้สามารถเข้าถึงได้แบบ Byte ด้วยวิธีการแบบ Direct หรือการอ้างตำแหน่งแอดเดรสของรีจิสเตอร์โดยตรงเพียงอย่างเดียวเท่านั้น ไม่สามารถใช้วิธีการเข้าถึงแบบระดับบิตได้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0

รูปที่ 2.14 แสดงโครงสร้างของ CCON Register

- **CF** เป็นบิต PCA Counter Overflow Flag ใช้แสดงสถานะการทำงานของ PCA Counter โดยบิตนี้จะมีค่าเป็น "1" เมื่อการทำงานของ PCA Counter เกิดการ Overflow ขึ้น โดยถ้ามีการกำหนดให้บิต ECF ในรีจิสเตอร์ CMOD มีค่าเป็น "1" ไว้ด้วย เมื่อบิต CF นี้ถูกเซ็ตเป็น "1" จะมีการร้องขอการ Interrupt ไปยัง CPU ด้วยเสมอ โดยเมื่อบิต CF นี้ถูกเซ็ตเป็น "1" แล้วต้องทำการรีเซ็ตหรือสั่งทำการเคลียร์ให้กลับเป็น "0" ด้วยโปรแกรมเพียงอย่างเดียวเท่านั้น

- **CR** เป็นบิต PCA Counter Run Control ใช้สำหรับควบคุมการทำงานของ PCA Timer โดยเมื่อกำหนดให้บิตนี้มีค่าเป็น "1" จะเป็นการสั่งให้ PCA Timer เริ่มต้นทำงาน และเมื่อกำหนดให้บิตนี้มีค่าเป็น "0" จะเป็นการสั่งหยุดการทำงานของ PCA Timer

- **CCFx** เป็นบิต PCA Modulex Interrupt Flag ใช้สำหรับแสดงสถานะการทำงานของ PCAModule แต่ละชุด ตามค่าบิต x เช่น CCF4 ใช้แสดงสถานะของ PCA โมดูลที่ 4 ส่วน CCF1 ก็ใช้แสดงสถานะการทำงานของ PCA โมดูลที่ 1 เป็นต้น ซึ่งบิต CCFx นี้จะถูกเซ็ตให้มีค่าเป็น "1" เมื่อการทำงานของโมดูลนั้นๆตรงตามเงื่อนไขที่กำหนดไว้ และเมื่อบิต CCFx ถูกเซ็ตเป็น "1" แล้วจะต้องสั่งรีเซ็ตหรือเคลียร์ให้กลับเป็น "0" ด้วยโปรแกรมเพียงอย่างเดียวเท่านั้น

2.3.5 การทำงานของ PCA Module

วงจร PCA Module ใน CPU เมอร์ P89C51RD2/T89C51RD2 จะมีอยู่ด้วยกันทั้งหมด 5 ชุดด้วยกัน โดยแต่ละชุด สามารถที่จะโปรแกรมหน้าที่การทำงานของแต่ละโมดูล แยกออกจากกัน ได้อย่างอิสระซึ่งแต่ละโมดูลของ PCA นั้น สามารถโปรแกรมหน้าที่การทำงานได้หลายหน้าที่ดังนี้คือ- วงจรตรวจจับ Input ขนาด 16 บิตแบบขอบขาขึ้น (Capture แบบ Positive-Edge Trigger)

- วงจรตรวจจับ Input ขนาด 16 บิตแบบขอบขาลง (Capture แบบ Negative-Edge Trigger)

- วงจรตรวจจับ Input ขนาด 16 บิตแบบทั้ง 2 ขอบ (Capture แบบ Positive และ Negative Edge Trigger)

- วงจรนับขนาด 16 บิต (16 Bit Software Timer)

- วงจรสร้างสัญญาณ Output ความเร็วสูงขนาด 16 บิต (16 Bit High Speed Output)

- วงจรสร้างสัญญาณ Pulse Width Modulation ขนาด 8 บิต (8 Bit PWM)

โดยการกำหนดโหมดการทำงานของวงจร PCA แต่ละโมดูลนั้นจะใช้รีจิสเตอร์ CCAPMx ในการควบคุมตั้งงาน แต่ละโมดูล โดยที่รีจิสเตอร์ CCAPM จะมีทั้งหมด 5 ชุด คือ CCAPM0-CCAPM4 โดยที่รีจิสเตอร์ CCAPM ทุกๆตัว จะมีคุณสมบัติที่เหมือนกันทุกประการ เพียงแค่ว่าจะทำหน้าที่ในการควบคุมการทำงานของวงจร PCA แต่ละโมดูลตามลำดับ โดย รีจิสเตอร์ CCAPM0 ก็จะใช้ควบคุมวงจร PCA โมดูลที่0 ส่วนรีจิสเตอร์ CCAPM1 ก็จะใช้สำหรับควบคุมการทำงานของวงจร PCA โมดูลที่1 และในทำนองเดียวกันรีจิสเตอร์ CCAPM4 ก็จะใช้สำหรับควบคุมการทำงานของวงจร PCA โมดูลที่4 เป็นต้น

2.3.6 คุณสมบัติของรีจิสเตอร์ CCAPMx

สำหรับรีจิสเตอร์ CCAPMx นั้นจัดเป็นรีจิสเตอร์ขนาด 8 บิต ใช้สำหรับเลือกหน้าที่การทำงานให้กับโมดูล CPA โดยที่รีจิสเตอร์นี้ จะมีอยู่ด้วยกันจำนวน 5 ชุด คือ CCAPM0, CCAPM1, CCAPM2, CCAPM3 และ CCAPM4 โดยคุณสมบัติของรีจิสเตอร์แต่ละตัว จะเหมือนกันทุกประการ ต่างกันเพียงแต่ละตัวจะใช้สำหรับควบคุมการทำงานของ โมดูล PCA ต่างชุดกัน กล่าวคือ รีจิสเตอร์ CCAPM0 ก็จะใช้สำหรับควบคุมการทำงานของ วงจร PCA โมดูล0 เป็นต้นแต่เนื่องจากหน้าที่การทำงานของรีจิสเตอร์ทั้ง 5 ชุดนี้ มีลักษณะเหมือนกัน ดังนั้นในที่นี้จึงขอกล่าวอธิบายรวมกัน โดยอ้างชื่อเป็น CCAPMx แทน ซึ่งค่า x ก็จะหมายถึง โมดูลชุดที่ x ซึ่งมีค่า 0-4 นั่นเอง โดยรีจิสเตอร์ CCAPMx นี้จะจัดไว้ในส่วนของ SFR (Special Function Register) และมีตำแหน่งแอดเดรส อยู่ระหว่าง DAH-DEH ตามลำดับ ซึ่งในการเข้าถึงรีจิสเตอร์ตัวนี้สามารถเข้าถึงได้แบบ Byte ด้วยวิธีการแบบ Direct หรือการอ้างตำแหน่งแอดเดรสของรีจิสเตอร์โดยตรงเพียงอย่างเดียวเท่านั้น ไม่สามารถใช่วิธีการเข้าถึงแบบระดับบิตได้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
-	ECOMx	CAPPx	CAPNx	MATx	TOGx	PWMx	ECCFx

รูปที่ 2.15 แสดงโครงสร้างของ CCAPMx Register

- **ECOMx** เป็นบิต Enable Comparator ใช้สำหรับเลือกการทำงานของโมดูล PCA ให้ทำหน้าที่เป็น วงจร Comparator โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการเปิดการทำงานฟังก์ชัน Comparator ของโมดูล PCA ที่ถูกควบคุมโดยรีจิสเตอร์ชุดนั้นๆ เช่น ถ้าทำการกำหนดให้บิต ECOM4 ของรีจิสเตอร์ CCAPM4 ก็จะเป็นการเปิดการทำงานฟังก์ชัน Comparator ของ PCA โมดูลที่ 4 เป็นต้น
- **CAPPx** เป็นบิต Capture Positive โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการกำหนดให้วงจร PCA ทำการตรวจจับสัญญาณ Input (Capture) ในขณะที่สัญญาณเป็นขอบขาขึ้น (Rising Edge)
- **CAPNx** เป็นบิต Capture Negative โดยถ้ากำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการกำหนดให้วงจร PCA ทำการตรวจจับสัญญาณ Input (Capture) ในขณะที่สัญญาณเป็นขอบขาลง (Falling Edge)
- **MATn** เป็นบิต Match ใช้สำหรับแสดงสถานะการทำงานของวงจร PCA โดยบิตนี้จะถูกเซ็ตให้มีค่าเป็น “1” เมื่อการทำงานของ โมดูล PCA นับแล้วมีค่าตรงกับค่าของ PCA Counter
- **TOGx** เป็นบิต Toggle ใช้สำหรับแสดงสถานะการทำงานของ โมดูล PCA
- **PWMx** เป็นบิต Pulse Width Modulation Mode ใช้สำหรับ เลือกการทำงานของ โมดูล PCA ให้ทำหน้าที่เป็นวงจรสร้างสัญญาณ PWM โดยเมื่อกำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการเลือกกำหนดให้ โมดูล PCA ที่ถูกเลือกทำหน้าที่สร้างสัญญาณ PWM ออกที่ขาสัญญาณ CEXn
- **ECCFx** เป็นบิต Enable CCF Interrupt ใช้สำหรับควบคุมการร้องขอ Interrupt ของโมดูล PCA โดยถ้า กำหนดให้บิตนี้มีค่าเป็น “1” จะเป็นการส่งอนุญาตให้โมดูล PCA ทำการร้องขอการ Interrupt ถ้าบิต CCFx ในรีจิสเตอร์ CCON ของโมดูล PCA นั้นๆ ถูกกำหนดให้มีค่าเป็น “1” ไว้ก่อนแล้ว

ECOMx	CAPPx	CAPNx	MATx	TOGx	PWMx	ECCFx	โหมดการทำงาน
0	0	0	0	0	0	0	ไม่ทำงาน
X	1	0	0	0	0	X	16 Bit Capture ขอบขาขึ้น
X	0	1	0	0	0	X	16 Bit Capture ขอบขาลง
X	1	1	0	0	0	X	16 Bit Capture ทั้ง 2 ขอบ
1	0	0	1	0	0	X	16 Bit Timer/Compare
1	0	0	1	1	0	X	16 Bit High Speed Output
1	0	0	1	0	1	0	8 Bit PWM
1	0	0	1	X	0	X	Watchdog Timer (PCA-4)

ตารางที่ 2.3 แสดงการกำหนดค่าสำหรับเลือกโหมดการทำงานของ โมดูล PCA

จากตารางจะเห็นได้ว่าโมดูล PCA นั้น สามารถจะโปรแกรมหน้าที่การทำงานในรูปแบบต่างๆ ได้ถึง 5 โหมดการทำงานด้วยกัน คือ

- 16 บิต Capture ในโหมดนี้ การทำงานของโมดูล PCA จะทำหน้าที่คอยตรวจจับการเปลี่ยนแปลงของสัญญาณ Input โดยสามารถกำหนดลักษณะการตรวจจับให้กับโมดูล PCA ได้ 3 ลักษณะคือ กำหนดให้คอยตรวจจับเมื่อสัญญาณเปลี่ยนแปลงสถานะเป็นช่วงขอบขาขึ้น (Rising Edge หรือ Positive Edge) ขอบขาลง (Falling Edge หรือ Negative) หรืออาจกำหนดให้คอยทำการตรวจจับการเปลี่ยนแปลงทั้งช่วงที่เป็นขอบขาขึ้นและขอบขาลงเลขก็ได้

- 16 บิต Software Timer/Compare ในโหมดนี้ โมดูล PCA จะทำหน้าที่คอยเปรียบเทียบการนับของโมดูล PCA และ PCA Counter โดยเมื่อการนับของวงจรทั้ง 2 มีค่าเท่ากันจะเกิดการ Interrupt ขึ้น

- 16 บิต High Speed Output ในโหมดนี้ โมดูล PCA จะทำการเปรียบเทียบการนับของ PCA และ โมดูล PAC Counter แต่จะมีการกลับสถานะของ Pin Port ของแต่ละ โมดูลด้วย เมื่อผลการเปรียบเทียบตรงตามที่กำหนดไว้

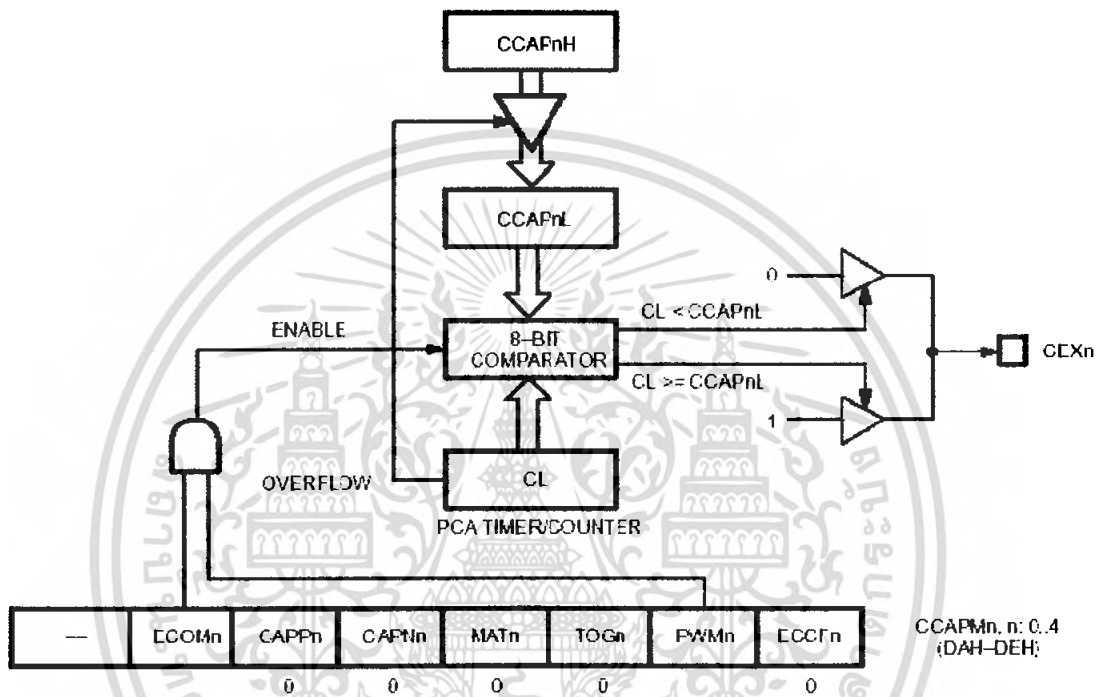
- 8 บิต Pulse Width Modulation ในโหมดนี้ โมดูล PCA จะทำหน้าที่สร้างสัญญาณ PWM ออกไปยังขา CEX ของแต่ละ โมดูล PCA โดยการทำงานของโมดูล PCA ในโหมดนี้จะสามารถควบคุมให้สร้างสัญญาณ PWM ขนาด 8 บิต โดยสามารถ ทำการกำหนดหรือเปลี่ยนแปลงค่า Duty Cycle ของสัญญาณ PWM ได้ตามต้องการ

- Watchdog Timer ใน โหมดนี้ จะใช้ได้กับโมดูล PCA4 เพียงชุดเดียวเท่านั้น ส่วน โมดูล PCA อื่นๆ จะไม่สามารถกำหนดการทำงานให้ทำหน้าที่ใน โหมดนี้ได้เนื่องจากหน้าที่การทำงาน ของโมดูล PCA นั้น มีอยู่ด้วยกันหลายโหมดคงได้กล่าวไปแล้วในข้างต้น แต่สำหรับในที่นี้จะขอกล่าวรายละเอียดการใช้งานของ PCA โมดูล เฉพาะส่วนการทำงานของ PCA โมดูล ในโหมดของการสร้างสัญญาณ PWM เท่านั้น เนื่องจากเป็นส่วนที่เกี่ยวข้องกับการสร้างสัญญาณ Pulse สำหรับใช้ควบคุมการทำงานของ DC SERVO MOTOR ส่วนการทำงานของ PCA ในโหมดอื่นๆ นั้น

ขอให้ผู้ใช้ศึกษาเพิ่มเติมได้จากส่วนของ Data Sheet ของ CPU เอง

2.3.7 การทำงานของ PCA ในโหมด Pulse Width Modulation

ในโหมดนี้ โมดูล PCA จะทำหน้าที่สร้างสัญญาณ PWM ออกไปยังขา CEX ของโมดูล PCA โดยการทำงานของ โมดูล PCA ในโหมดนี้จะสามารถควบคุมให้สร้างสัญญาณ PWM ขนาด 8 บิต โดยสามารถทำการกำหนดหรือเปลี่ยนแปลงค่า Duty Cycle ของสัญญาณ PWM ได้ตามต้องการ



รูปที่ 2.16 แสดงโครงสร้างการทำงานของ PCA ในโหมด PWM

สำหรับหลักการการทำงานของ PCA Timer ในการสร้างสัญญาณ PWM นั้นจะอาศัยผลการเปรียบเทียบค่าการนับแบบ 8 บิต ระหว่าง PCA Timer (ซึ่งตามปกติมีขนาด 16 บิต แต่ในโหมดนี้จะใช้การนับแบบ 8 บิต) โดยค่าการนับของ PCA Timer นั้นจะใช้รีจิสเตอร์ CL เป็นตัวเก็บค่าการนับ โดยผลการนับของ CL จะถูกนำไปเปรียบเทียบกับค่าที่กำหนดไว้ในรีจิสเตอร์ CCAPnL โดยถ้าค่าการนับของ PCA Timer ในรีจิสเตอร์ CL มีค่าน้อยกว่าค่าที่กำหนดไว้ใน รีจิสเตอร์ CCAPnL แล้วจะได้ผลลัพธ์ของสัญญาณมีค่าเป็น "0" ที่ Pin Port ของ CPU ที่ทำหน้าที่เป็น Output Pin ของ PCA ช่องนั้นๆ (CEXn = "0") แต่ถ้าค่าการนับของรีจิสเตอร์ CL มีค่าเท่ากับหรือมากกว่าค่าที่กำหนดไว้ในรีจิสเตอร์ CCAPnL แล้วจะได้ผลลัพธ์เป็น "1" ที่ Pin Port ของ CPU ที่ทำหน้าที่เป็น Output Pin ของ PCA ช่องนั้นๆ (CEXn = "1") โดยจะเห็น

ว่ารีจิสเตอร์ CL ซึ่งถูกใช้สำหรับทำการนับนั้น มีขนาดเป็น 8 บิต ดังนั้นค่าผลการนับของรีจิสเตอร์ CL จึงมีค่าการนับแบบวนรอบอยู่ระหว่าง 00H-FFH (0-255) หรือ 256 ค่าพอดี โดยทุกๆครั้งที่ค่าการนับของรีจิสเตอร์ CL เกิดค่าเกินหรือ Overflow หรือค่าการนับเกินกว่า FFH แล้ว ค่าการนับของรีจิสเตอร์ CL จะวนกลับไปเริ่มต้นที่ 00H ใหม่เสมอ ดังนั้นค่า Period หรือ ค่าคาบเวลาของสัญญาณ PWM นั้นจะขึ้นอยู่กับผลคูณของ 256 กับ ค่าความถี่ของสัญญาณนาฬิกาที่ป้อนให้กับวงจร PCA เสมอ ตัวอย่างเช่น สัญญาณนาฬิกาที่ป้อนให้กับ PCA เพื่อทำการนับมีค่าความถี่ 1Hz ค่าคาบเวลาทั้งหมดของสัญญาณ PWM ก็จะมีค่าเป็น 256 Hz โดยปริยาย ส่วนค่า Duty Cycle หรือค่าคาบเวลาที่เป็นซิกบวกลบของสัญญาณ PWM นั้น ได้จากการกำหนดค่าอ้างอิงการเปรียบเทียบที่กำหนดให้กับรีจิสเตอร์ CCAPnL แทน โดยค่า Duty Cycle ของสัญญาณ PWM จะกำหนดโดยค่าใน CCAPnL โดยถ้ากำหนดให้ค่าของ CCAPnL มีค่าน้อยจะได้ลักษณะของสัญญาณ PWM ที่มีคาบเวลาด้านบวกลบมาก แต่ถ้ากำหนดให้ค่าของ CCAPnL มีค่ามากจะทำให้ได้ลักษณะของสัญญาณ Pulse ที่มีคาบเวลาซิกบวกลบน้อย โดยค่าที่จะกำหนดให้กับรีจิสเตอร์ CCAPnL นั้น จะไม่ได้กำหนดให้กับรีจิสเตอร์นี้โดยตรง แต่จะกำหนดไว้ในรีจิสเตอร์ CCAPnH แทน โดยจะเห็นได้ว่าในทุกๆครั้งที่ค่าการนับของรีจิสเตอร์ CL นั้นเกิด Overflow หรือเกินกว่า FFH จะมีการส่งสัญญาณ ไปที่ การ Reload ค่าในรีจิสเตอร์ CCAPnH มายังรีจิสเตอร์ CCAPnL ทุกครั้งด้วยเสมอ ตัวอย่างเช่น ถ้ามีการกำหนดค่าให้กับรีจิสเตอร์ CCAPnL (กำหนดผ่าน CCAPnH) ด้วยค่า 10 จะได้ว่า ลักษณะของสัญญาณ PWM ที่ได้จะมีคาบเวลาของสัญญาณซิกบวกลบ ("0") เป็นเวลา 10/256 หน่วยของคาบเวลาทั้งหมด และก็จะมีความถี่ในซิกบวกลบ("1") เป็นเวลา 246/256 หน่วยของคาบเวลาทั้งหมดเป็นต้น

แนวคิดและวิธีการในการคำนวณค่าพารามิเตอร์ต่างๆสำหรับสั่งงานให้ระบบ PCATimer สร้างสัญญาณ PWM ที่มีคาบเวลา 20mS และมีค่า Duty Cycle ของสัญญาณซีกบวกขนาด 1mS และ 2mS สำหรับควบคุมการทำงานของ DC SERVO MOTOR สามารถแสดงให้เห็นได้ดังขั้นตอนต่อไปนี้

ต้องการคาบเวลาของ PWM(Period)	=	20mS	
วงรอบการนับของ PCA Timer (8 Bit)	=	256 Cycle	
ต้องป้อนสัญญาณนาฬิกาให้ PCA Timer	=	20mS / 256 Cycle	= 78.125 uS
ค่าความถี่ XTAL ที่ป้อนให้กับ CPU	=	18.432 MHz	
คาบเวลาของสัญญาณนาฬิกา 1 Cycle	=	1/18.432 MHz	
	=	54.253 nS	
1 Machine Cycle ของ CPU (X2 Mode)	=	54.253 nS x 6 Clock	
	=	325.520 nS	
1 Machine Cycle ของ Timer0 (X2 Mode)	=	54.253 nS x 6 Clock	= 325.520 nS
ต้องการให้ Timer Overflow ที่ 78.125uS	=	78.125uS / 325.520nS	
	=	240 Cycle	
ต้องกำหนดค่า Auto Reload ให้ Timer0	=	256-240 Cycle	
	=	16 Cycle	
ต้องการคาบเวลาของ PWM ช่วงบวก 1mS	=	1mS / 78.125uS	
	=	12.8 Cycle	= 13 Cycle
	=	256-13 Cycle	= 243 Cycle
ต้องกำหนดค่าเปรียบเทียบให้ CCAPnH	=	243	
ต้องการคาบเวลาของ PWM ช่วงบวก 2mS	=	2mS / 78.125uS	
	=	25.6 Cycle	= 26 Cycle
	=	256-26 Cycle	= 230 Cycle
ต้องกำหนดค่าเปรียบเทียบให้ CCAPnH	=	230	

แสดงวิธีการคำนวณค่าพารามิเตอร์ต่างๆสำหรับใช้ในการสร้างสัญญาณ PWM

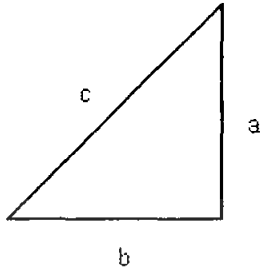
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างโปรแกรมข้างต้นจะเห็นว่า จะมีการแบ่งแสดงส่วนของโปรแกรมต่างๆ ออกเป็น 3 ส่วนด้วยกัน โดยโปรแกรมในส่วนแรกจะเป็นส่วนของการ Initial ค่ารีจิสเตอร์ต่างๆของ PCA Timer และส่วนที่เกี่ยวข้องเพื่อกำหนดหน้าที่การทำงานของ PCA โมดูลที่ 1 และ 2 (CEX1 หรือ P1.4 และ CEX2 หรือ P1.5) สำหรับทำหน้าที่ในโหมด PWM โดยในตัวอย่างโปรแกรมจะเลือกใช้ Timer0 และ PCA Timer ร่วมกันในการสร้างสัญญาณ PWM ซึ่งระบบ Timer0 จะถูกกำหนดให้ทำหน้าที่สำหรับสร้างฐานเวลาการนับ(Input Clock) เพื่อป้อนให้กับวงจรมับของ PCA Timer ส่วน PCA Timer นั้นจะส่งผลค่าการนับให้กับโมดูลPCA ทั้ง 5 ชุด แต่ในตัวอย่างนั้น จะมีการโปรแกรมการใช้งานโมดูล PCA เพียง 2 ชุด คือ PCA โมดูล1 และPCA โมดูล 2 สำหรับวิธีการคำนวณค่าที่ต้องกำหนดให้กับรีจิสเตอร์สามารถอธิบายให้เข้าใจพอเป็นแนวทางในการใช้งานได้ดังนี้จากวิธีการคำนวณหาค่าพารามิเตอร์ต่างๆข้างต้น จะเห็นว่า ระบบการทำงานของ PCA Timer ในโหมดของการสร้างสัญญาณ PWM นั้น การทำงานของวงจรมับจะเป็นแบบ 8 บิต โดยให้รีจิสเตอร์ CL ทำการนับค่า ซึ่งค่าการนับของ CL จะวนรอบการนับอยู่ระหว่าง 00H-FFH หรือ 256 ค่า และเมื่อการนับเกิดการ Overflow หรือเกินจาก FFH ก็จะไปเริ่มต้นนับใหม่จาก 00H อีก โดยก่อนที่จะเริ่มต้นนับใหม่จาก FFH เป็น 00H นั้น จะมีการ Reload ค่าจาก CCAPnH มาให้กับ CCAPnL ก่อนเสมอ โดยค่าของ CCAPnH จะเป็นค่าสำหรับกำหนดคาบเวลาของสัญญาณซีกกลับและซีกบวก (Duty Cycle ของ PWM) ส่วนค่าของ $20\text{ms} / (\text{CL} \times 256)$ จะเป็นค่า Period ของสัญญาณ PWM ที่จะทำการ Reload ซ้ำใหม่ซึ่งสัญญาณ PWM ที่เราต้องการในการควบคุมการหมุนของ DC SERVO MOTOR นั้น เราต้องการสัญญาณ PWM ที่มีขนาดของคาบเวลา (PERIOD) เท่ากับ 20ms ดังนั้นเพื่อให้การทำงานของโปรแกรมไม่เกิดความซ้ำซ้อน เราจึงไม่ต้องการเข้าไปเปลี่ยนแปลงค่าการนับของรีจิสเตอร์ CL กล่าวคือ จะปล่อยให้รีจิสเตอร์ CL ทำการนับแบบอิสระแบบวนรอบจาก 00H-FFH ต่อเนื่องกันไปไม่รู้จบ จนกว่าจะมีการสั่งหยุดการนับ ดังนั้นในอันดับแรกเราจึงต้องทำการคำนวณหาค่าของสัญญาณนาฬิกาที่จะป้อนเป็น Input ของการนับของ PCA Timer หรือ CL เพื่อจะได้ทราบว่า จะต้องป้อนค่าสัญญาณนาฬิกาด้วยค่าความถี่เท่าใดให้กับ PCA Timer ทำการนับจำนวน 256 ครั้ง (00H-FFH) จึงจะได้คาบเวลา 20ms ซึ่งจะได้เท่ากับ 78.125 ไมโครวินาที ($78.125\mu\text{s}$)ซึ่งระบบการนับของ PCA Timer นั้นสามารถเลือกแหล่งกำเนิดสัญญาณนาฬิกาได้ 4 แหล่ง แต่แหล่งที่สามารถจะนำมาใช้ได้กับกรณีนี้ คือ จากการ Overflow ของ Timer0 ดังนั้นจะต้องทำการกำหนดโหมดการทำงานของระบบ Timer0 เพื่อให้สร้างสัญญาณนาฬิกาที่มีคาบเวลา 78.125 ไมโครวินาที($78.125\mu\text{s}$) ซึ่งค่าความถี่ของ XTAL ที่ป้อนให้กับ CPU และ Timer0 นั้นจะเป็น 18.432MHz ซึ่งจะได้คาบเวลาของสัญญาณนาฬิกา 18.432MHz เป็น $1/18.432\text{MHz}$ หรือ

54.253นาโนวินาที(54.253nS) โดยการทำงานของ CPU และ Timer0 จะทำงานแบบ X2 โหมด หรือ 6 Clock / 1Machine Cycle ดังนั้น ซึ่งจะมีค่าเป็น 54.253nS x6 หรือ 325.520 นาโนวินาที (325.520nS) โดยเราต้องการให้ระบบ Timer0 ทำการนับและเกิดการ Overflow ด้วยคาบเวลา 78.125uS ซึ่ง 1 Cycle ของ Timer0 ที่ทำการนับจะมีค่าเป็น 325.520nS ดังนั้นจะต้องให้ Timer0 ทำการนับจำนวน 240 ครั้ง (78.125uS / 325.520nS) ซึ่งจะเห็นว่า การนับของ Timer มีค่าไม่เกิน 256 ครั้ง ดังนั้นเราจึงทำการกำหนด โหมดการทำงานของ Timer0 ให้ทำงานแบบ 8 Bit Auto-Reload เพื่อโปรแกรมจะได้ไม่เสียเวลาไปทำการ Reload ค่าการนับให้กับ Timer0 ใหม่ เมื่อเกิดการ Overflow ในแต่ละครั้ง แต่เนื่องจากระบบการนับของ Timer0 เป็นแบบนับขึ้น ดังนั้นในการกำหนดค่าการนับให้กับ Timer0 จะต้องนำค่าสูงสุดของการนับ (FFH หรือ 256) มาทำการหักลบออกจากค่าที่ต้องการให้นับ ซึ่งในที่นี้เราต้องการให้ Timer0 ทำการนับ 240 ครั้ง ดังนั้นจึงต้องกำหนดค่าการ Reload ของ Timer0 เป็น 16 (256-240) สำหรับการคำนวณหาค่า Duty Cycle ของสัญญาณ PWM จะคำนวณเพียง 2 ค่า คือ 1mS และ 2mS ซึ่งการนับของ PCA Timer นั้นจะนับจากสัญญาณนาฬิกา 78.125uS ดังนั้นจึงได้ค่าสำหรับกำหนดค่า Duty Cycle ของ PWM เป็น $243(256-(1\text{mS}/78.125\text{uS}))$ และ $230(256-(2\text{mS}/78.125\text{uS}))$ ตามลำดับซึ่งจากวิธีการกำหนดค่าพารามิเตอร์ต่างๆข้างต้น จะเป็นการควบคุมให้วงจร PWM1(P1.4) และ PWM2(P1.5) ทำการสร้างสัญญาณ PWM ซึ่งมีคาบเวลา หรือ Period ขนาด 20mS คงที่เท่ากันทั้ง 2 ช่อง ส่วนค่า Duty Cycle นั้น PWM แต่ละช่อง สามารถกำหนดได้อย่างอิสระตามต้องการระหว่าง 78.125uS ไปจนถึง 20mS โดยค่า Duty Cycle จะกำหนดให้กับรีจิสเตอร์ CCAPnH (PWM1 = CCAP1H ส่วน PWM2 = CCAP2H) โดยค่าที่กำหนดให้กับรีจิสเตอร์ CCAPnH นั้นจะเป็นจุดแบ่งของสัญญาณ PWM ระหว่าง "0" และ "1" เช่น ถ้ากำหนดค่าของ CCAPnH เป็น 243 จะหมายถึงว่า สัญญาณ PWM จะมีสถานะเป็น "0" จำนวน 243 Cycle และจะมีสถานะเป็น "1" จำนวน 13 Cycle ดังนั้นการที่กำหนดให้ค่า CCAPnH มีค่า 243 จึงเป็นการกำหนดให้สัญญาณ PWM มีสถานะเป็น "0" เป็นเวลา 19mS และมีสถานะเป็น "1" เป็นเวลา 1mS โดยประมาณนั่นเอง

2.4 การเปลี่ยนค่าพิกัดในระบบ 3 มิติ (XYZ) เป็นพิกัดที่ใช้ในการควบคุม Servo motor ในแต่ละจุด

2.4.1 ทฤษฎีบทสามเหลี่ยม

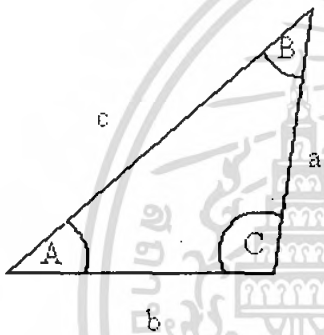


รูปที่ 2.17 รูปสามเหลี่ยมมุมฉาก

24.1.1 ทฤษฎีสามเหลี่ยมมุมฉาก

ความสัมพันธ์ของสามเหลี่ยมมุมฉาก

$$c^2 = a^2 + b^2$$



รูปที่ 2.18 รูปสามเหลี่ยมใด ๆ

24.1.2 ทฤษฎีสามเหลี่ยมใด ๆ

ความสัมพันธ์ของสามเหลี่ยมใด ๆ

$$c^2 = a^2 + b^2 - 2ab \cos C$$

$$b^2 = a^2 + c^2 - 2ac \cos B$$

$$a^2 = b^2 + c^2 - 2bc \cos A$$

24.1.3 ระบบพิกัด

- เมื่อจุดกำเนิดอยู่ที่จุด Original

ความสัมพันธ์จากรูป

$$r^2 = x^2 + y^2 + z^2$$

$$z = r \sin \alpha$$

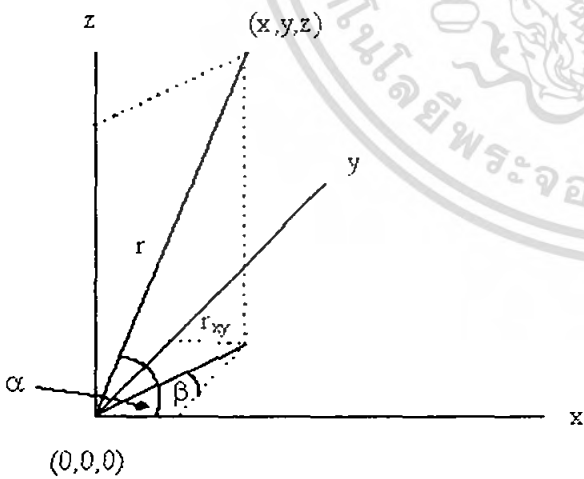
$$r_{xy} = r \cos \alpha$$

$$y = r_{xy} \sin \beta$$

$$= r \cos \alpha \sin \beta$$

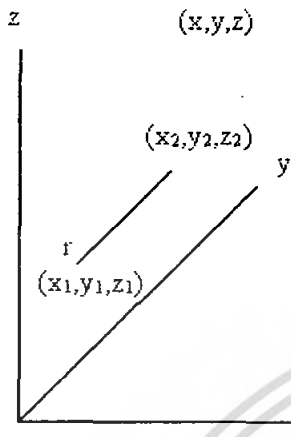
$$x = r_{xy} \cos \beta$$

$$= r \cos \alpha \cos \beta$$



รูปที่ 2.19 ระบบพิกัดเมื่อจุดกำเนิดอยู่ที่จุดกำเนิด

- เมื่อจุดกำเนิดไม่ได้อยู่ ณ จุดกำเนิด

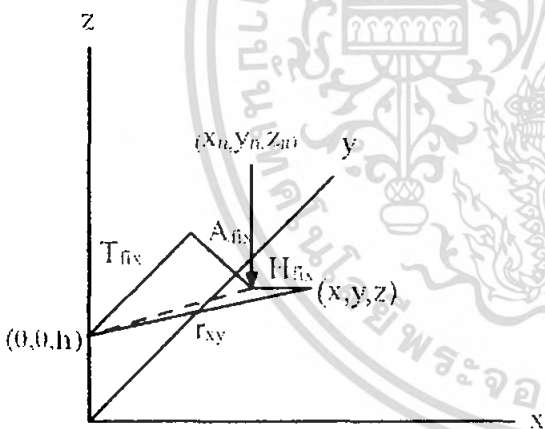


$$r^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2$$

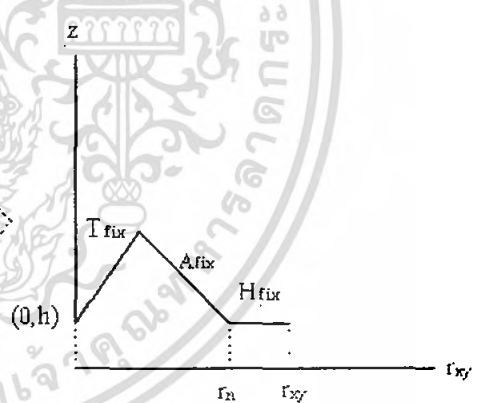
รูปที่ 2.20 ระบบพิกัดเมื่อจุดกำเนิดไม่เท่ากับจุดกำเนิด

2.4.2 การประยุกต์ทฤษฎีมาใช้งานจริงกับแขนกล

- ภาพสเก็ตของแขนกลในรูปแบบ 3 มิติ และ 2 มิติ



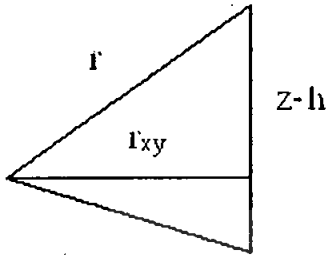
รูปที่ 2.21 แสดงแขนกลจำลองเทียบกับพิกัด 3 มิติ



รูปที่ 2.22 แสดงแขนกลเทียบกับพิกัด 2 มิติ

สำหรับการนำสมการมาประยุกต์ใช้จริงกับการคำนวณหาค่าตำแหน่งที่แท้จริงของแขนกลนั้นจะคิดจากระนาบทั้ง 3 มิติ คือ คิดตามแกน XYZ ซึ่งจะเป็นการคำนวณจากระนาบ 3 มิติ มาเทียบกับ แกน XY ที่เป็น 2 มิติ โดยในรูปทางด้านขวามือ นั้นเราจะมิต่างๆ ที่ถูกระบุไว้แล้วคือ T_{fix} , A_{fix} , H_{fix} ซึ่งเป็นค่าของ

Twin, Arm และ Hand ตามลำดับจากนั้นเราจะสามารถคำนวณหาค่าของ r_{xy} เพื่อจะใช้ในการคำนวณหาตำแหน่งที่ถูกดึงต่อไปได้



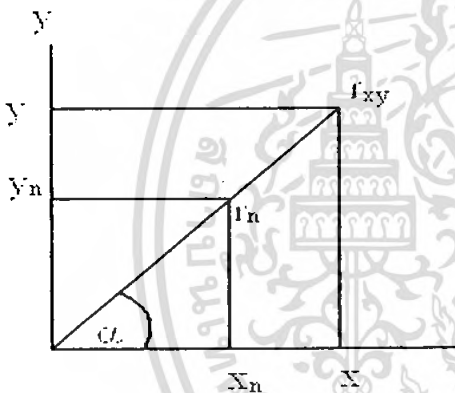
ความสัมพันธ์

$$r_{xy} = r^2 - (z-h)^2$$

$$r_n = r_{xy} - H_{fix}$$

รูปที่ 2.23 พิกัดของแขนกลจำลอง

พิจารณาตามรูปข้างล่างนี้จะได้สมการว่า



ความสัมพันธ์

$$\alpha = \tan^{-1}(y/x)$$

$$x_n = r_n \cos \alpha$$

$$y_n = r_n \sin \alpha$$

รูปที่ 2.24 แสดงความสัมพันธ์ของพิกัดเก่าและพิกัดใหม่

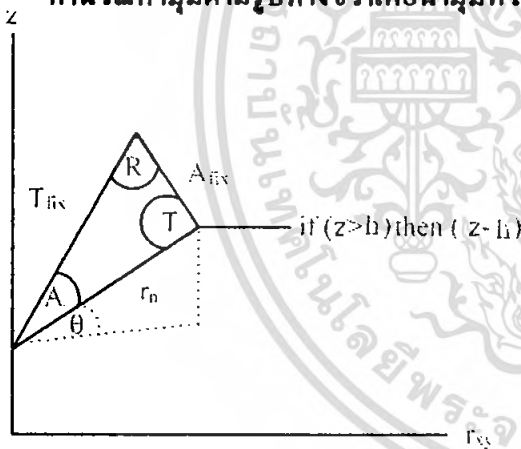


รูปที่ 2.25 แสดงความสัมพันธ์ระหว่างค่าพิกัดใหม่และความสัมพันธ์ของสามเหลี่ยมที่ได้จากแขนกล

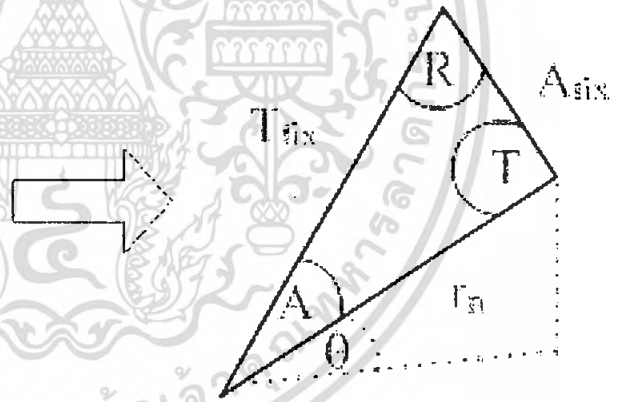
จากสมการในหัวข้อที่ 2. จะหาค่าของ T_{fix} , A_{fix} และ r_n ได้ดังนี้

- $T = \cos^{-1} \left(\frac{A_{fix}^2 + r_n^2 - T_{fix}^2}{2 A_{fix} r_n} \right)$
- $A = \cos^{-1} \left(\frac{T_{fix}^2 + r_n^2 - A_{fix}^2}{2 T_{fix} r_n} \right)$
- $R = \cos^{-1} \left(\frac{A_{fix}^2 + T_{fix}^2 - r_n^2}{2 A_{fix} T_{fix}} \right)$

จากรูปข้างบนนี้ ได้แสดงถึงขั้นตอนสุดท้ายของการแปลงค่าต่างๆ ที่ได้มา มาเปลี่ยนเป็นค่ามุม เพื่อป้อนเป็นค่าแก่ servo motor โดยจากรูปทางด้านซ้ายนั้น เป็นการแสดงถึงส่วนของ Twin หรือแขนคู่ ที่ใช้ในการขับเคลื่อนที่มีการควบคุม servo motor พร้อมกันถึง 2 ตัว ซึ่งจากการใช้สูตรในการคำนวณ จากสมการที่กล่าวมาแล้วก็จะสามารถหาค่า ต่างๆ ได้ครบถ้วนทุกค่า จากนั้นเราก็จะนำค่านั้นๆ มาคำนวณหามุมตามรูปทางขวา เพื่อนำมุมที่ได้จากการคำนวณนั้นไปใช้ในการควบคุมแขนกลต่อไป



รูปที่ 2.26 แสดงแขนกลกับระบบพิกัด



รูปที่ 2.27 แสดงแขนกลกับสามเหลี่ยมใด ๆ

จากรูปข้างบนนี้เป็นการแสดงถึงการคำนวณหาค่ามุมที่จะนำไปใช้จริงในการควบคุมแขนกล ซึ่งในกรณีนี้เป็นกรณีที่ ค่า z มีค่ามากกว่า h โดยเราจะสามารถหาค่า θ ได้ดังนี้

$$\theta = \sin^{-1} \left(\frac{z - h}{r_n} \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปเราจะคำนวณหามุมที่แท้จริงที่ใช้ในการควบคุมแขนกลทั้งหมด คือนุมของ Shoulder, Elbow, Wrist และ Base โดยค่ามุมของ Elbow นั้นเป็นมุมคงที่ที่ได้จากรูปข้างต้นคือ R ส่วนมุมของ Shoulder นั้นก็มีอยู่ทั้งหมด 2 กรณีคือกรณีที่ $z > h$ และกรณีที่ $z < h$ ดังนี้

$$\begin{aligned} \text{Shoulder} &= \arccos\left(\frac{z}{h}\right) && \text{เมื่อ } z > h \\ &= \arccos\left(\frac{z}{h}\right) && \text{เมื่อ } z < h \\ \text{Elbow} &= R \\ \text{Wrist} &= T \\ \text{Base} &= \alpha \end{aligned}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การปรับเทียบกล้อง(Camera Calibration)

3.1 บทนำ

บทนี้เรากล่าวถึงการเห็นใน 3 มิติซึ่งเป็นการดึงข้อมูล 3 มิติออกจากภาพ 2 มิติ คณิตศาสตร์ที่เกี่ยวข้องกับการมองเห็นใน 3 มิติและ คณิตศาสตร์ที่เกี่ยวข้องกับการปรับเทียบกล้อง (Camera Calibration) ซึ่งจะแบ่งออกเป็น การประมาณค่าเมทริกซ์ M การแก้สมการเพื่อหาค่าเมทริกซ์ M

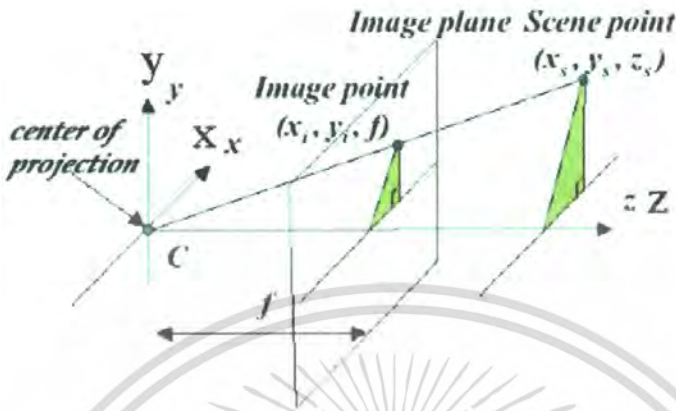
3.2 เรขาคณิตสำหรับการเห็นใน 3 มิติ (Geometry for 3D Vision)

3.2.1 การเกิดภาพในกรณีที่พิกัดโลกซ้อนทับพิกัดกล้อง

ในระบบภาพ 2 มิติส่วนใหญ่ได้มาจากการมองวัตถุ 3 มิติผ่านเลนส์ของตาหรือเลนส์ของกล้อง ขบวนการเกิดภาพสามารถถูกจำลองได้โดยใช้ Perspective Transformation รูปที่ 3.1 แสดงแบบจำลองง่ายๆของระบบสร้างภาพ เราแทนแกนระบบพิกัดโลก (World Coordinate System) ด้วยอักษรตัวใหญ่ (X, Y, Z) และแทนแกนของระบบพิกัดของกล้องด้วยอักษรตัวเล็ก (x, y, z) เราสมมติว่าพิกัดของโลกซ้อนทับกับระบบพิกัดของกล้อง แสงจากวัตถุใน 3 มิติส่องผ่านศูนย์กลางของการโปรเจกชันแล้วโปรเจกต์ลงบนภาพ ให้จุด (x_s, y_s, z_s) แทนจุดพิกัดของวัตถุ 3 มิติให้ (x_i, y_i) แทนจุดพิกัดของวัตถุในระนาบของภาพ เราสมมติว่าระนาบของภาพอยู่ f ห่างไปทางขวาของจุดกำเนิดของพิกัดกล้องโดยที่ f ความยาวโฟกัสของเลนส์ โดยการใช้สามเหลี่ยมคล้ายได้ว่า

$$x_i = f \frac{x_s}{z_s} \quad (3.1)$$

$$y_i = f \frac{y_s}{z_s} \tag{3.2}$$



รูปที่ 3.1 แบบจำลองของระบบสร้างภาพ

จากแบบจำลองการสร้างภาพเราสามารถหาความสัมพันธ์ระหว่างระบบพิกัดโลก (World Coordinate System) กับระบบพิกัดของกล้องด้วยสมการที่ (3.3)

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \tag{3.3}$$

โดยที่

$$x_i = \frac{u}{w} \tag{3.4}$$

$$y_i = \frac{v}{w} \tag{3.5}$$

3.2.2 การแปลงจากหน่วยความยาวเป็นพิกเซล

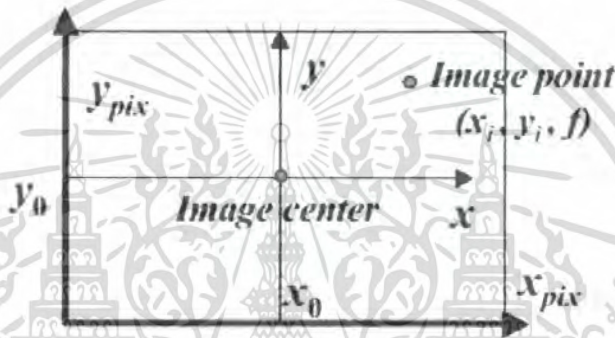
การแปลงหน่วยความยาวไปเป็นพิกเซลเราจะเริ่มจากกำหนดให้พิกัดของภาพมีจุดกำเนิดอยู่ที่มุมล่างซ้ายของภาพดังแสดงในรูป 3.2 การแปลงจากความยาวเป็นพิกเซลเราจำเป็นต้องรู้ค่าการปรับสเกล (Aspect Ratio) ของแต่ละแกนของระบบพิกัดภาพค่าการปรับสเกลเป็นค่าที่ระบุจำนวนพิกเซลต่อหนึ่งหน่วยความยาว ในระบบภาพบางระบบพิกเซลในแถวถัดไปอาจมีการเอียงกัน จำเป็นต้องรู้ค่าที่เรียกว่า พารามิเตอร์ของการเอียง (Skew parameter) นอกจากนี้ยังจำเป็นต้องรู้จุดหลัก (Principal Point) $U_0 =$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(x_0, y_0) หรือบางครั้งถูกเรียกว่าจุดกึ่งกลางของภาพของขบวนการปรับ โดยที่ k_x คือค่าการปรับสเกลในทิศทาง x , k_y คือค่าการปรับสเกลในทิศทาง y และ s แทนพารามิเตอร์ของการเอียง เราเขียนในรูปเทียบกล้อง (Camera Calibration Procedure) เป็นจุดตัดระหว่างแกนอ็อบติคกับระนาบภาพพิกัดของจุด (x_i, y_i, f) หาได้จากความสัมพันธ์

$$x_{pix} = k_x x_i + s y_i + x_0 = k_x f \frac{x_s}{z_s} + s f \frac{y_s}{z_s} + x_0 \quad (3.6)$$

$$y_{pix} = k_y y_i + y_0 = k_y f \frac{y_s}{z_s} + y_0 \quad (3.7)$$



รูปที่ 3.2 พิกัดของภาพ

เราสามารถอยู่ในรูปของของเมทริกซ์ได้เป็น

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \alpha_x & s & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad (3.7)$$

โดยที่ $\alpha_x = k_x x_s$ ซึ่งเป็นค่าความยาวโฟกัสในทิศ x ในหน่วยของพิกเซลและ $\alpha_y = k_y y_s$ ซึ่งเป็นค่าความยาวโฟกัสในทิศ y ในหน่วยของพิกเซล เราสามารถเขียน

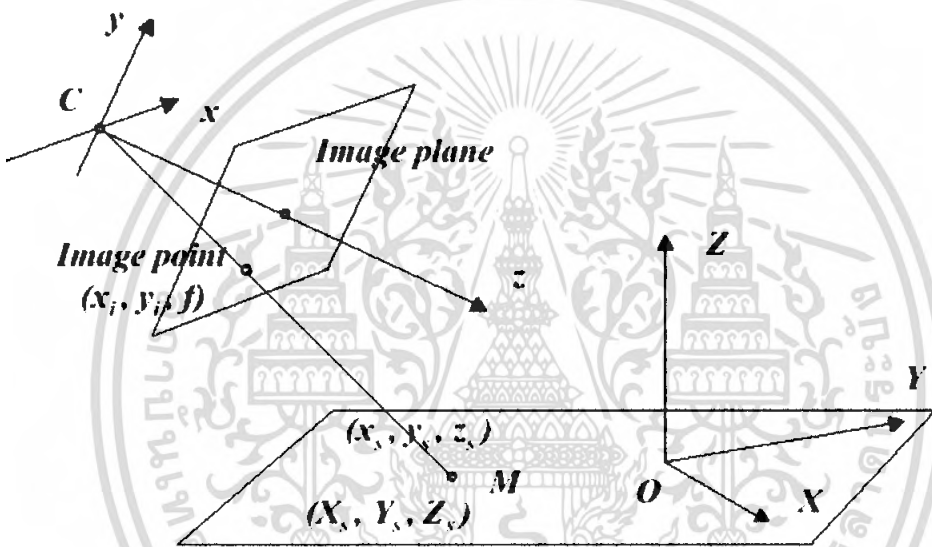
$$\begin{bmatrix} \alpha_x & s & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \kappa [I_3 | \mathbf{o}_3] \quad (3.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมทริกซ์ K มีลักษณะเป็นสามเหลี่ยมข้างบน (Upper Triangle) ขนาด 3×3 มีชื่อว่า Calibration Matrix เป็นเมทริกซ์ที่ประกอบพารามิเตอร์ภายใน (Intrinsic Parameter) ที่สำคัญของกล้อง

3.2.3 กรณีที่พิกัดของโลกไม่ซ้อนทับกับพิกัดของกล้อง

ในกรณีที่พิกัดโลกไม่ซ้อนทับกับพิกัดกล้องนั้นเราจะพิจารณาจากรูปที่ 3.3 โดยจะเห็นว่าจุดกึ่งกลางของการโปรเจกชันของกล้องถูกวางไว้ที่ตำแหน่งอื่นที่ไม่ใช่ตำแหน่งเดียวกับตำแหน่งพิกัดโลก ทำให้จุดกึ่งกลางของกล้องมีการส่ายและเอียงทำมุมกับแกนอ้างอิง



รูปที่ 3.3 แบบจำลองการเกิดภาพ

ในกรณีนี้เราสามารถหาความสัมพันธ์ของการเกิดภาพจากกล้องสามารถหาได้ง่ายโดยการเปลี่ยนแปลง (Modify) สมการ (3.7) โดยการ คำนวณการคูณด้วยเมทริกซ์การเคลื่อนย้าย (Translation) และการหมุน (Rotation) ใน 3 มิติ เข้าไปยังสมการที่ (3.7) ซึ่งเราจะได้สมการความสัมพันธ์ของพิกัดโลกกับพิกัดกล้องในกรณีที่พิกัดของโลกไม่ซ้อนทับกับพิกัดของกล้องเป็นไปตามสมการที่ (3.9)

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = K [I_3 | \mathbf{0}_3] \begin{bmatrix} \mathbf{R} & -\mathbf{T} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad (3.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ R แทนเมทริกซ์ที่เป็นผลรวมของการหมุนระบบพิกัด และ T แทนเวกเตอร์ที่เป็นผลรวมของการย้ายระบบพิกัด เรากล่าวว่าเมทริกซ์

$$\begin{bmatrix} R & -T \\ \mathbf{o}_3^T & 1 \end{bmatrix} \quad (3.10)$$

เราสามารถเขียนสมการ (3.9) ในรูปแบบที่ง่ายได้เป็นดังสมการที่ (3.11) และ M คือเมทริกซ์ของ Perspective Transformation

$$\mathbf{x} = M \mathbf{X} \quad (3.11)$$

3.3 การปรับเทียบกล้อง (Camera Calibration)

การปรับเทียบกล้องเป็นการหาประมาณเมทริกซ์ M จากจุดในฉาก 3 มิติที่รู้ตำแหน่งและภาพที่เกิดขึ้นของจุด จากนั้นทำการหาพารามิเตอร์ภายในและพารามิเตอร์ภายนอก พิจารณาการปรับเทียบโดยใช้กล้องตัวเดียว

ในการหาเมทริกซ์ M เราทำการหาจุดในฉากที่เราทราบพิกัด $\mathbf{X} = [x, y, z]^T$ และจุดในภาพ 2 มิติที่สอดคล้องกัน $[u, v]^T$ ซึ่งจะได้สมการที่อยู่ในรูปสมการ (3.11) ดังนี้

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.12)$$

$$\begin{bmatrix} \alpha u \\ \alpha v \\ \alpha \end{bmatrix} = \begin{bmatrix} m_{11}x + m_{12}y + m_{13}z + m_{14} \\ m_{21}x + m_{22}y + m_{23}z + m_{24} \\ m_{31}x + m_{32}y + m_{33}z + m_{34} \end{bmatrix} \quad (3.13)$$

ทำการนอร์มอลไลซ์ให้สมาชิกที่ 3 เป็น 1 เราจะได้สมการ

$$u(m_{31}x + m_{32}y + m_{33}z + m_{34}) = m_{11}x + m_{12}y + m_{13}z + m_{14} \quad (3.14)$$

$$v(m_{31}x + m_{32}y + m_{33}z + m_{34}) = m_{21}x + m_{22}y + m_{23}z + m_{24} \quad (3.15)$$

ดังนั้นเราจะได้สมการเชิงเส้น 2 สมการสำหรับจุดในฉาก 3 มิติหนึ่งจุดและจุดในภาพ 2 มิติที่สอดคล้องกัน ทำการเขียนสมการ (3.15) ใหม่ในรูปของเมทริกซ์ได้เป็น

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & -u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & -v \\ & & & & & & & & \vdots & & & \\ & & & & & & & & & & & m_{34} \end{bmatrix} = 0 \quad (3.16)$$

$AM=0$



รูปที่ 3.4 ภาพของกล่องพลาสติกใสที่นำมาใช้ในการปรับเทียบกล้อง

จะเห็นว่าเรามีตัวแปรที่ไม่ทราบค่า 11 ตัวแปร แทนที่จะเป็น 12 ตัวแปรเนื่องจากเฟคเตอร์การย่อ/ขยายที่ไม่สามารถทราบค่าได้ ในการแก้สมการโฮโมจีเนียสเราต้องใช้อย่างน้อย 6 จุด ถ้ามีมากกว่า 6 จุดเราจะได้สมการ Over-determined ซึ่งสามารถแก้ได้โดยใช้วิธี Least Square หรือใช้การหา Singular Value Decomposition (SVD) ของ A เมทริกซ์ M คือแถวสุดท้ายของ V เมื่อได้เมทริกซ์ M แล้วขั้นตอนต่อไปทำการแยกพารามิเตอร์ภายใน (Intrinsic Parameter) เนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mathbf{M} = [\mathbf{KR} \mid -\mathbf{KRT}] = [\mathbf{A} \mid \mathbf{B}] \quad (3.17)$$

ดังนั้นเมทริกซ์ย่อย 3×3 แทนด้วย \mathbf{A} เวกเตอร์คอลัมน์ขวามือสุดแทนด้วย \mathbf{b} เวกเตอร์ที่มีพารามิเตอร์ระยะเคลื่อนที่สามารถแยกออกมาได้ง่ายโดย $\mathbf{t} = -\mathbf{A}^{-1} \mathbf{b}$ เวกเตอร์ \mathbf{t} ให้ข้อมูลที่บอกถึงตำแหน่งกึ่งกลางของ Image Plane

จากนั้นพิจารณา $\mathbf{A} = \mathbf{KR}$ โดยที่ \mathbf{K} เป็นเมทริกซ์สามเหลี่ยมบน(Upper Triangle) และเมทริกซ์ \mathbf{R} เป็นเมทริกซ์ออร์โธกอนอล การแยกเมทริกซ์ \mathbf{K} และ \mathbf{R} สามารถทำได้โดยใช้เทคนิค QR Decomposition สำหรับเมทริกซ์ \mathbf{A}

3.4 การหาพิกัด 3 มิติของจุดในฉากโดยใช้กล้อง 2 ตัว

เมื่อพิจารณาการหาพิกัด 3 มิติของจุดในฉากโดยใช้กล้อง 2 ตัว โดยที่กล้องทั้ง 2 ต้องผ่านขบวนการปรับเทียบกล้องดังที่กล่าวมาแล้ว ในขบวนการปรับเทียบเราให้กล้องทั้งสองถ่ายภาพวัตถุในฉาก 3 มิติที่เราทราบพิกัด ซึ่งอาจเป็นตารางหมากรุกที่เราทราบขนาดของแต่ละตาราง จากนั้นทำการหาพิกัดของจุดตัดของตารางในกล้องทั้งสองใช้สมการ 5.16 ทำการคำนวณเมทริกซ์ของ Perspective Transformation \mathbf{M} สำหรับแต่ละกล้อง ต่อไปทำการวางวัตถุที่เราต้องการพิกัด 3 มิติในฉาก ทำการหาจุดสอดคล้องของจุดในฉาก 3 มิติของกล้องทั้งสองตัว กำหนดให้เมทริกซ์ของ Perspective Transformation \mathbf{M} ของกล้องตัวที่ 1 และ 2 เป็น \mathbf{M} และ \mathbf{M}' ตามลำดับ เราแทนแต่ละแถวของเมทริกซ์ \mathbf{M} ด้วย m_1^T, m_2^T, m_3^T ทำนองเดียวกันสำหรับกล้องที่สอง เราแทนแต่ละแถวของเมทริกซ์ \mathbf{M}' ด้วย $m_1'^T, m_2'^T, m_3'^T$

สำหรับกล้องที่ 1 เราได้

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{M}\mathbf{X} = \begin{bmatrix} m_1^T \\ m_2^T \\ m_3^T \end{bmatrix} \mathbf{X}$$

ทำนองเดียวกันสำหรับกล้องที่สองเราได้

$$\mathbf{u}' = \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{M}'\mathbf{X} = \begin{bmatrix} m_1'^T \\ m_2'^T \\ m_3'^T \end{bmatrix} \mathbf{X}$$

โดยที่ u และ u' แทนพิกัดในภาพของวัตถุในกล้อง 1 และ 2 ตามลำดับ ในการกำจัดค่าสเกลที่ไม่ทราบค่า เราทำการหาสัดส่วนระหว่างสามแถวในเมทริกซ์ Perspective Transformation M

$$u : v : w = m_1^T X : m_2^T X : m_3^T X$$

$$u' : v' : w' = m_1'^T X : m_2'^T X : m_3'^T X$$

ดังนั้นเราได้สมการสำหรับกล้อง 1 และ 2

$$u m_2^T X = v m_1^T X$$

$$u' m_2'^T X = v' m_1'^T X$$

$$u m_3^T X = w m_1^T X$$

$$u' m_3'^T X = w' m_1'^T X$$

$$v m_3^T X = w m_2^T X$$

$$v' m_3'^T X = w' m_2'^T X$$

ทำการจัดรูปสมการข้างต้นใหม่จะได้สมการใหม่ดังนี้

$$\begin{bmatrix} u m_3^T - w m_1^T \\ v m_3^T - w m_2^T \\ u' m_3'^T - w' m_1'^T \\ v' m_3'^T - w' m_2'^T \end{bmatrix} X = AX = 0$$

ทำการแก้สมการโฮโมจีเนียสด้านบนสำหรับพิกัด 3 มิติ หา X โดยเราสนใจเฉพาะคำตอบประเภท Non-Trivial หรือกรณี $\det(A) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การแปลงเชิงเรขาคณิต

4.1 บทนำ

การแปลงเชิงเรขาคณิตเป็นทฤษฎีพื้นฐานทางคอมพิวเตอร์กราฟิกกว่าด้วยการแปลงพิกัดภาพหรือวัตถุหนึ่งไปยังพิกัดใหม่โดยใช้พารามิเตอร์ของการแปลงจำนวนหนึ่ง

4.2 การแปลงเชิงเรขาคณิตแบบเชิงเส้น

ในงานทางด้านคอมพิวเตอร์กราฟิกนั้น บางครั้งจำเป็นต้องมีการเคลื่อนย้ายภาพไปยังตำแหน่งใหม่ ย่อ-ขยายภาพ หรือมีการหมุนภาพให้อยู่ในมุมมองที่เหมาะสม ซึ่งสามารถทำได้โดยใช้การแปลงเชิงเรขาคณิต(Geometric transformation)

ในการพิจารณาการแปลงนั้นจะพิจารณาได้ว่าเป็นการแปลงจุดในปริภูมิสามมิติ และเพื่อความเข้าใจง่ายจึงมักจะเขียนอยู่ในรูปของเมทริกซ์ โดยกำหนดให้จุดที่ต้องการแปลงเป็นจุด P มีพิกัดเป็น (x, y, z) ใดๆ เมื่อต้องการย้ายตำแหน่งของจุด P ไปยังพิกัดใหม่คือ P' ซึ่งมีพิกัด (x', y', z') ก็ทำได้โดยการนำพิกัดของจุด P ไปบวกกับเมทริกซ์การแปลงซึ่งก็คือ T สามารถแสดงเมทริกซ์ของการแปลงได้ดังนี้

$$P' = P + T = [x \ y \ z] + [T_x \ T_y \ T_z] \quad (4.1)$$

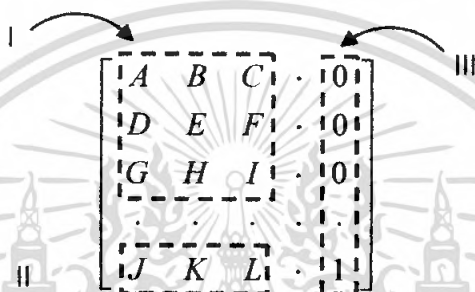
นอกจากนี้เมื่อนำพิกัดของจุด P มาคูณกับเมทริกซ์การแปลง T ให้ได้พิกัดใหม่คือ P' นั้นจะเป็นการสเกล การหมุน การสะท้อน หรืออื่นๆ ขึ้นอยู่กับค่าพารามิเตอร์ภายในเมทริกซ์การแปลงนั้นและสามารถเขียนในรูปของเมทริกซ์ได้ดังนี้

$$P' = P \cdot T = [x \ y \ z] \cdot \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (4.2)$$

สำหรับการแปลงเชิงเรขาคณิตในทางคอมพิวเตอร์กราฟิกโดยทั่วไปแล้วจะประกอบด้วย

แปลงหลายๆชนิดประกอบกัน และเนื่องจากเป็นการคำนวณในรูปของเมทริกซ์จึงทำให้ยากในการคำนวณเมทริกซ์ที่มีรูปแบบแตกต่างกัน จึงได้มีการนำระบบพิกัดโฮโมจีเนียส (Homogeneous Coordinate System) เข้ามาใช้เพื่อแก้ปัญหานี้

ในระบบพิกัดโฮโมจีเนียส จุดในระบบสามมิติ (x, y, z) จะถูกแทนด้วยระบบสี่มิตินั้นคือ (x, y, z, H) โดยจะมี H เป็นค่าแพกเตอร์ที่มีหน้าที่สเกลซึ่งจะมีค่าไม่เท่ากับศูนย์ จุด (x, y, z, H) จะถูกนอร์มอลไลซ์ (Normalize) เป็น $(x/H, y/H, z/H, 1)$ โดยทั่วไปแล้วค่า H จะมีค่าเท่ากับ 1 ดังนั้นจุด (x, y, z) ใดๆจะมีค่าในระบบพิกัดโฮโมจีเนียสเป็น $(x, y, z, 1)$



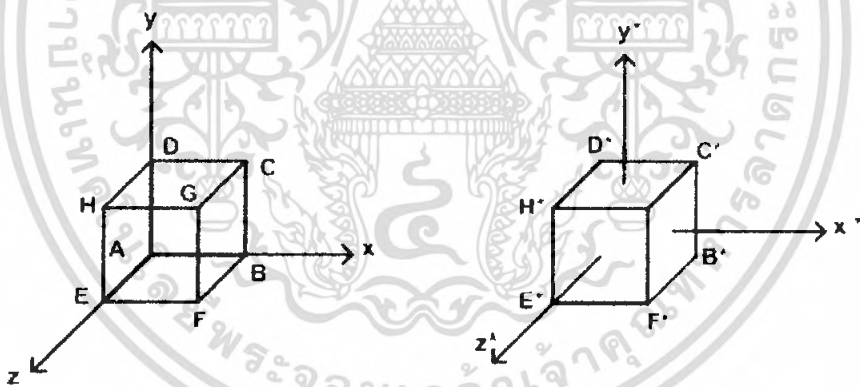
โดยที่ค่า T_x, T_y และ T_z แสดงระยะทางที่ต้องการเคลื่อนย้ายตำแหน่งไปในแนวแกน X, Y และ Z ตามลำดับ และพิกัดของจุดในระบบโฮโมจีเนียส (x', y', z', h) เป็น

$$[x' \ y' \ z' \ h] = [x \ y \ z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \quad (4.4)$$

หรือ

$$[x' \ y' \ z' \ h] = [x + T_x \ y + T_y \ z + T_z \ 1] \quad (4.5)$$

การเลื่อนเมทริกซ์เป็นการแปลงที่มีประโยชน์มากในทางคอมพิวเตอร์กราฟิก เนื่องจากข้อมูลภาพต้นฉบับที่นำมาใช้จะมีตำแหน่งอยู่ในด้านบวกเท่านั้น จึงจำเป็นต้องการอาศัยการแปลงแบบนี้เพื่อเลื่อนพิกัดกึ่งกลางของข้อมูลต้นฉบับให้มาอยู่ที่จุดกำเนิด $(0,0,0)$ เสียก่อน แสดงในรูปที่ 4.2 จึงถือได้ว่าการแปลงแบบเลื่อนพิกัดเป็นพื้นฐานที่สำคัญของการสร้างภาพทางคอมพิวเตอร์กราฟิก



รูปที่ 4.2 การเลื่อนจุดกึ่งกลางของปริมาตรต้นฉบับมายังจุดกำเนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การสเกล (Scaling)

การสเกลเป็นการแปลงเพื่อปรับเปลี่ยนขนาดของวัตถุที่ต้องการ มีเมทริกซ์การแปลง (S) ดังนี้

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

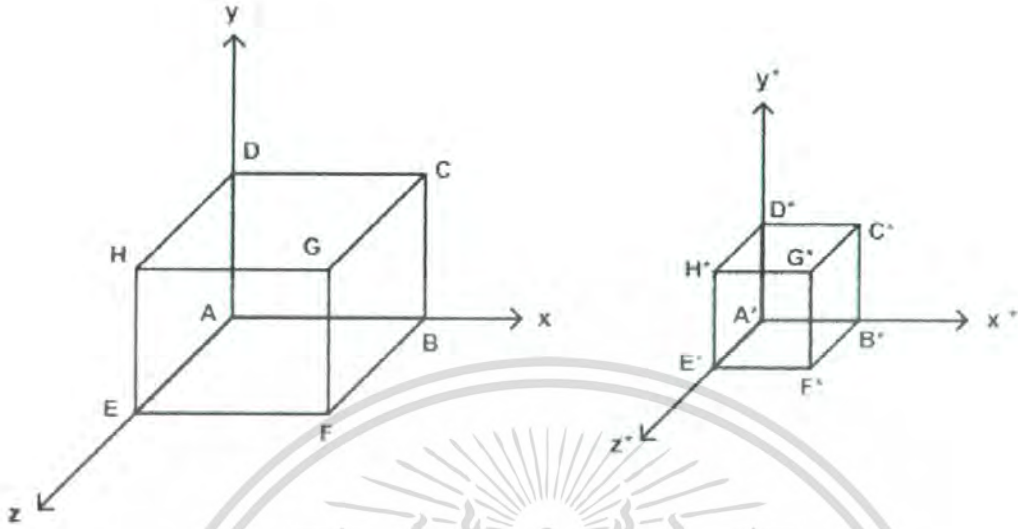
โดยที่ S_x, S_y และ S_z เป็นสัมประสิทธิ์การสเกลสำหรับพิกัด X, Y และ Z ตามลำดับ และพิกัดของจุดในระบบโฮโมจีเนียส (x', y', z', h) เป็น

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

หรือ

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} xS_x & yS_y & zS_z & 1 \end{bmatrix} \quad (4.8)$$

ค่า S_x, S_y และ S_z จะเป็นตัวกำหนดว่าเป็นการย่อหรือขยายรูป กล่าวคือ ถ้า S มีค่ามากกว่า 1 จะเป็นการขยายภาพและถ้า S มีค่าน้อยกว่า 1 จะเป็นการย่อขนาดภาพ



รูปที่ 4.3 การสเกลวัตถุ

4.2.3 การหมุนวัตถุ (Rotation)

การหมุนวัตถุเป็นการหมุนวัตถุรอบแกนต่างๆทั้งสามแกน ซึ่งได้แก่ แกน X, Y หรือ Z โดยการอ้างอิงการมองในลักษณะมองออกจากจุดกำเนิดไปตามแนวแกน เมทริกซ์การแปลงของการหมุนรอบแกนต่างๆในทิศทางตามเข็มนาฬิกา มีดังนี้

เมทริกซ์การแปลงของการหมุนรอบแกน X ไปเป็นมุม θ คือ

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

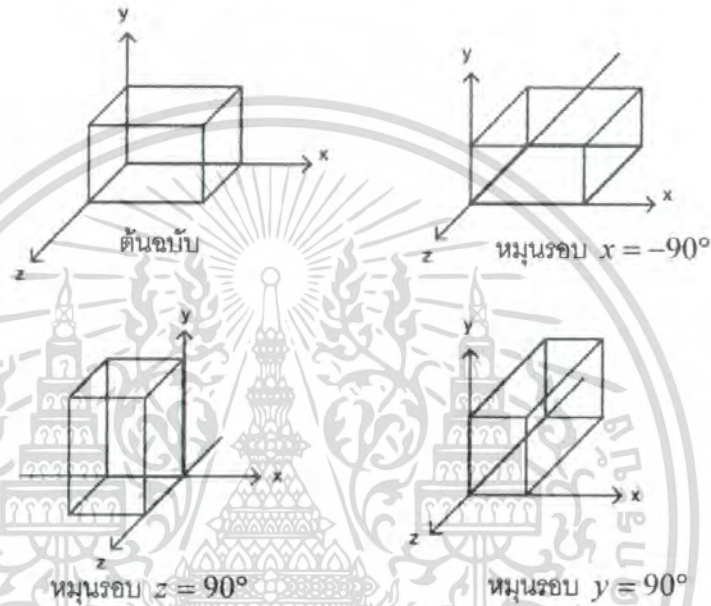
เมทริกซ์การแปลงของการหมุนรอบแกน Y ไปเป็นมุม ϕ คือ

$$R_y = \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมทริกซ์การแปลงของการหมุนรอบแกน Z ไปเป็นมุม φ คือ

$$R_z = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$



รูปที่ 4.4 การแปลงแบบหมุนรอบแกนทั้งสาม

4.2.4 การสะท้อน (Reflection)

การสะท้อนวัตถุเป็นการสะท้อนพิกัดของวัตถุไปยังพิกัดที่อยู่อีกด้านหนึ่งของระนาบการสะท้อน XY, YZ หรือ XZ เมทริกซ์ของการสะท้อนมีดังต่อไปนี้

เมทริกซ์ของการสะท้อนกับระนาบ XY คือ

$$Rf_{XY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

จะเปลี่ยนเฉพาะพิกัด z โดย $z' = -z$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมทริกซ์ของการสะท้อนกับระนาบ YZ คือ

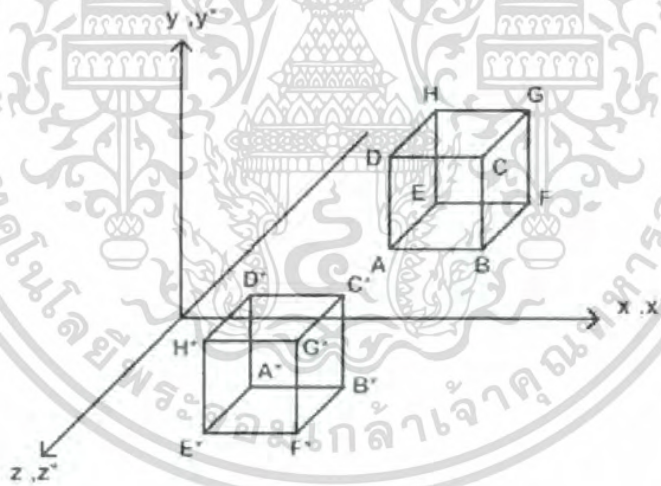
$$Rf_{yz} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.13)$$

จะเปลี่ยนเฉพาะพิกัด x โดย $x' = -x$

เมทริกซ์ของการสะท้อนกับระนาบ XZ คือ

$$Rf_{xz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

จะเปลี่ยนเฉพาะพิกัด y โดย $y' = -y$



รูปที่ 4.5 การสะท้อนวัตถุกับระนาบ XY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 การเฉือน (Shearing)

การเฉือนในปริภูมิสามมิติเป็นการแปลงพิกัดตามแกนของสองแกนใดๆ (สมมติว่าเป็น x และ y) โดยสัมพันธ์กับค่าของพิกัดตามแนวแกนที่สาม (z) เป็นผลให้ค่าพิกัดตามแนวแกนของสองแกนแรกเปลี่ยนไป ในขณะที่ค่าพิกัดตามแนวแกนที่สามนั้นมีค่าคงเดิม

เมทริกซ์ของการเฉือนโดยสัมพันธ์กับแกน X

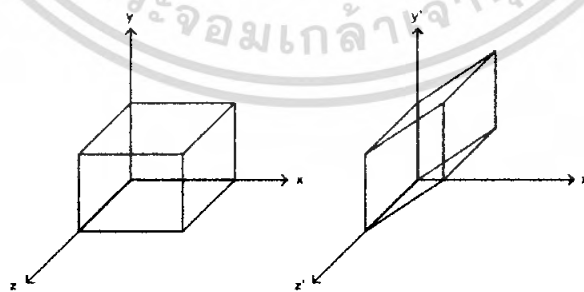
$$Sh_x = \begin{bmatrix} 1 & Sh_{xy} & Sh_{xz} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

เมทริกซ์ของการเฉือนโดยสัมพันธ์กับแกน Y

$$Sh_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sh_{yx} & 1 & Sh_{yz} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

เมทริกซ์ของการเฉือนโดยสัมพันธ์กับแกน Z

$$Sh_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ Sh_{zx} & Sh_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.17)$$



รูปที่ 4.6 การเฉือนโดยสัมพันธ์กับค่า z เมื่อ $Sh_{zx} = 0$ และ $Sh_{zy} = 1$

4.2.6 การฉายภาพ (Projection)

การฉายภาพเป็นการแปลงจาก n มิติไปเป็น $n-1$ มิติ การฉายภาพมี 2 ประเภทคือแบบเชิงเส้นและไม่เป็นเชิงเส้น ในวิชานี้สนใจเป็นพิเศษในเฉพาะการฉายภาพแบบเชิงเส้นเท่านั้น สำหรับการฉายภาพแบบเชิงเส้นยังสามารถแบ่งได้เป็น 2 ประเภทคือ

4.2.6.1 การฉายภาพแบบตั้งฉาก (Orthographic Projections)

การฉายภาพแบบตั้งฉากเป็นวิธีฉายภาพแบบขนานอย่างหนึ่งที่มีรูปแบบอย่างง่ายและถูกใช้ในการวาดภาพทางวิศวกรรม โดยจะแสดงขนาดจริงของวัตถุในแต่ละด้าน เมทริกซ์ของการฉายภาพลงบนระนาบต่างๆเป็นดังนี้

เมทริกซ์ของการฉายภาพลงบนระนาบ $x = 0$

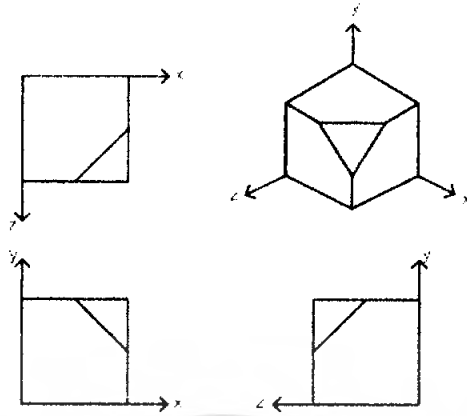
$$P_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

เมทริกซ์ของการฉายภาพลงบนระนาบ $y = 0$

$$P_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.19)$$

เมทริกซ์ของการฉายภาพลงบนระนาบ $z = 0$

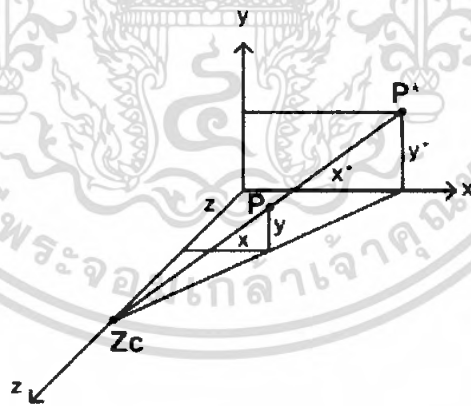
$$P_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$



รูปที่ 4.7 แสดงการฉายภาพลงบนระนาบ $x = 0$, $y = 0$ และ $z = 0$

4.2.6.2 การฉายภาพแบบเพอร์สเปกทีฟ (Perspective Projections)

เป็นรูปแบบการแปลงเชิงเรขาคณิตอย่างหนึ่งซึ่งแตกต่างจากการแปลงแบบอื่นๆที่ได้กล่าวมาแล้ว คือการแปลงแบบเลื่อนพิกัด การสเกล การหมุน การสะท้อน และการเฉือนนั้นเป็นการแปลงซึ่งคงคุณสมบัติความขนานกันของเส้น แต่การแปลงแบบเพอร์สเปกทีฟนี้แตกต่างออกไป และการแปลงแบบนี้ยังมีผลให้ขนาดของวัตถุเปลี่ยนแปลงไปโดยมีความสัมพันธ์แบบผกผันกับระยะห่างระหว่างวัตถุกับจุดศูนย์กลางของการฉายภาพ



รูปที่ 4.8 การฉายภาพแบบเพอร์สเปกทีฟ

พิจารณารูปที่ 3.8 เป็นการแปลงแบบเพอร์สเปกทีฟและทำการฉายภาพจากจุด P ไปยังจุด P^* บนระนาบ $z = 0$ โดยมี z_c เป็นจุดศูนย์กลางของการฉายภาพ (Center of projection) จากคุณสมบัติของสามเหลี่ยมคล้าย ได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำมาใช้

$$\frac{x^*}{z_c} = \frac{x}{z_c - z}$$

หรือ

$$x^* = \frac{x}{1 - \frac{z}{z_c}}$$

และ

$$\frac{y^*}{\sqrt{(x^*)^2 + z_c^2}} = \frac{y}{\sqrt{x^2 + (z_c - z)^2}}$$

หรือ

$$y^* = \frac{y}{1 - \frac{z}{z_c}}$$

จะได้ x^*, y^* เป็นพิกัดของ P^*

$$\text{กำหนดให้ } r = \frac{-1}{z_c} \text{ จะได้ } x^* = \frac{x}{rz+1} \text{ และ } y^* = \frac{y}{rz+1} \quad (4.21)$$

หรือเขียนในรูปของเมทริกซ์ได้ดังนี้

$$[x \ y \ z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ 0 \ rz+1] \quad (4.22)$$

เนื่องจาก $h = rz + 1 \neq 1$ ดังนั้นจึงคูณสมการที่ 3.22 ด้วย $\frac{1}{rz+1}$

$$\text{จะได้ } [x^* \ y^* \ z^* \ 1] = \begin{bmatrix} \frac{x}{rz+1} & \frac{y}{rz+1} & 0 & 1 \end{bmatrix} \quad (4.23)$$

จะเห็นว่าสมการที่ (4.21) มีค่าเท่ากับสมการที่ (4.23)

จะได้เมทริกซ์ของการแปลงแบบเพอร์สเปกทีฟเมื่อจุดศูนย์กลางของการฉายภาพอยู่บนแกน X, Y และ Z ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่มีจุดศูนย์กลางของการฉายภาพอยู่บนแกน X หรือที่พิกัด $[-1/p \ 0 \ 0 \ 1]$

$$[x \ y \ z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ px+1] \quad (4.24)$$

และ

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{px+1} \ \frac{y}{px+1} \ \frac{z}{px+1} \ 1 \right] \quad (4.25)$$

กรณีที่มีจุดศูนย์กลางของการฉายภาพอยู่บนแกน Y หรือที่พิกัด $[0 \ -1/q \ 0 \ 1]$

$$[x \ y \ z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ qy+1] \quad (4.26)$$

และ

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{qy+1} \ \frac{y}{qy+1} \ \frac{z}{qy+1} \ 1 \right] \quad (4.27)$$

กรณีที่มีจุดศูนย์กลางของการฉายภาพอยู่บนแกน Z หรือที่พิกัด $[0 \ 0 \ -1/r \ 1]$

$$[x \ y \ z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ rz+1] \quad (4.28)$$

และ

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{rz+1} \ \frac{y}{rz+1} \ \frac{z}{rz+1} \ 1 \right] \quad (4.29)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 การแปลงแบบผสม (Multiple Transformations)

โดยทั่วไปในการใช้งานทางด้านคอมพิวเตอร์กราฟิกจำเป็นต้องใช้การแปลงหลายแบบร่วมกัน เช่นในการหมุนวัตถุรอบเส้นตรงใดๆ ซึ่งขนานกับแกน Z จำเป็นต้องทำการเลื่อนให้จุดศูนย์กลางของวัตถุไปยังจุดกำเนิดก่อน จึงจะทำการหมุนวัตถุรอบแกน Z จากนั้นจึงทำการเลื่อนวัตถุกลับไปยังตำแหน่งเดิม

ตัวอย่าง หมุนวัตถุเป็นมุม φ รอบเส้นตรงที่ขนานกับแกน Z ซึ่งผ่านจุด $(-T_x, -T_y, -T_z)$ กำหนดให้เมทริกซ์ M เป็นเมทริกซ์การแปลงแบบผสม จะได้

$$M = T_T \cdot R_z \cdot T_T^{-1} \quad (4.30)$$

เมื่อ

$$T_T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \quad T_T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -T_x & -T_y & -T_z & 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 & 0 \\ -\sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

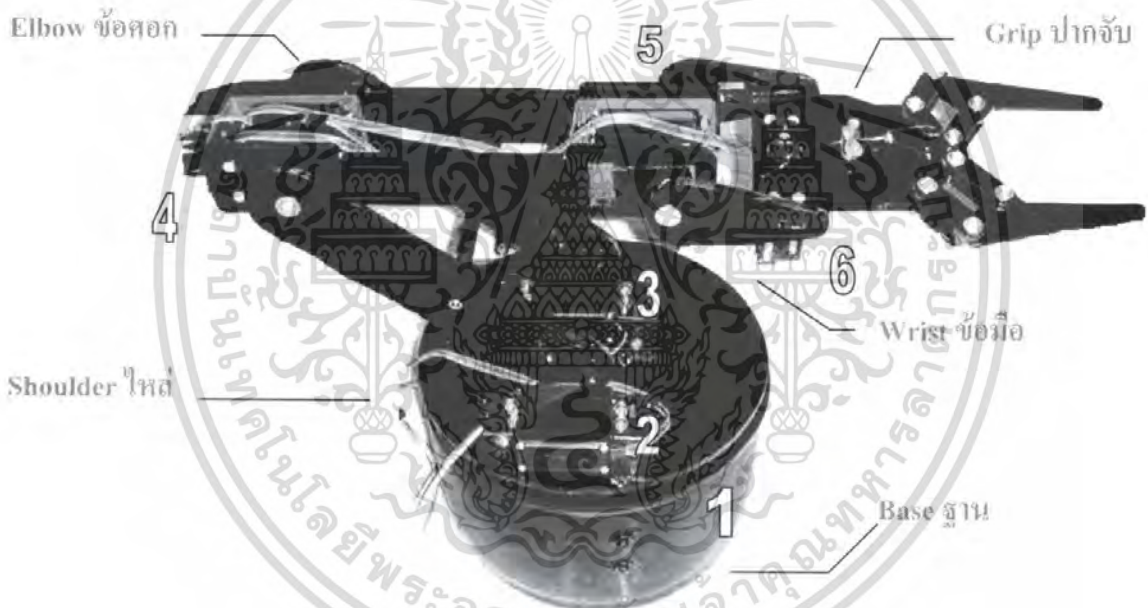
เนื่องจากเมทริกซ์ไม่มีคุณสมบัติการสลับที่ ดังนั้นลำดับของเมทริกซ์ของการแปลงจึงมีความสำคัญอย่างยิ่ง

บทที่ 5

การออกแบบ

ในบทที่แล้วได้อธิบายถึงทฤษฎีพื้นฐานและหลักการทำงานของส่วนประกอบที่สำคัญของแขนกลซึ่งจากบทที่แล้วจะเป็นพื้นฐานสำคัญในการทำความเข้าใจกับระบบแขนกลที่ใช้ในโรงงานอุตสาหกรรม และจากข้อดีหลาย ๆ อย่างของระบบแขนกล จึงมีการนำแขนกลเข้ามาใช้อย่างแพร่หลายในโรงงานอุตสาหกรรมใหญ่ ซึ่งต่อไปก็จะได้กล่าวถึงขั้นตอนการสร้างแขนกลเพื่อศึกษาการทำงานจริงรวมทั้งจะได้อธิบายถึงขั้นตอนการออกแบบวงจรควบคุมและ โปรแกรมที่จะทำการพัฒนารวมทั้งหน้าจอ Interface ด้วย

5.1 โครงสร้างแขนกล



รูปที่ 5.1 แสดงโครงสร้างแขนกล

ได้ทำการออกแบบและสร้างแขนกล โดยได้พิจารณาเลือกวัสดุที่ราคาถูก หาง่ายมีน้ำหนักเบา ในส่วนที่เป็นฐานจะมีการถ่วงน้ำหนักเพื่อให้แขนกลอยู่กับที่เนื่องจากต้องรับน้ำหนักและเมื่อมีการหมุนเหวี่ยง ระบบกลไกของแขนกลจะมีแกนหมุน (Degree of freedom) ทั้งหมดอยู่ 5 แกนหมุน คือ Base, Shoulder, Elbow, Wrist และ Gripper ซึ่งทำหน้าที่จับวัตถุ ในบริเวณส่วนของ Shoulder ต้องใช้มอเตอร์ 2 ตัว เนื่องจากต้องรับน้ำหนักมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแขนกลจำลองที่ได้พัฒนาขึ้นนั้นจะใช้ Servo Motor ทั้งหมด 6 ตัว และ Servo 6 ตัวจะประกอบอยู่ในตำแหน่ง 6 ตำแหน่ง ซึ่งแต่ละตำแหน่งทำหน้าที่ดังต่อไปนี้

ตำแหน่งที่ 1 เป็น Servo Motor ที่ติดอยู่กับฐานด้านในของแขนกลทำหน้าที่หมุนแขนกลให้เคลื่อนที่ได้ครึ่งวงกลม(-90° ถึง 90°)

ตำแหน่งที่ 2 และ 3 ใช้ Servo Motor 2 ตัว ทำหน้าที่เสมือนหัวไหล่เพื่อยกต้นแขนของแขนกล จะช่วยให้แขนกลนั้นเคลื่อนที่ไปด้านหน้าหรือด้านหลังก็ได้ทำหน้าที่ในการเคลื่อนที่ร่วมกับ Servo Motor ตำแหน่งที่ 3 Servo Motor ในตำแหน่งที่ 2 นี้จะต้องเคลื่อนที่ไปพร้อมกันในทิศทางเดียวกัน

ตำแหน่งที่ 4 ใช้ Servo Motor 1 ตัวทำหน้าที่คล้ายกับศอกจะทำหน้าที่ร่วมกับ Servo Motor ตำแหน่งที่ 2 ในการเคลื่อนที่ไปในจุดต่าง ๆ สามารถพับเข้าและกางออกได้

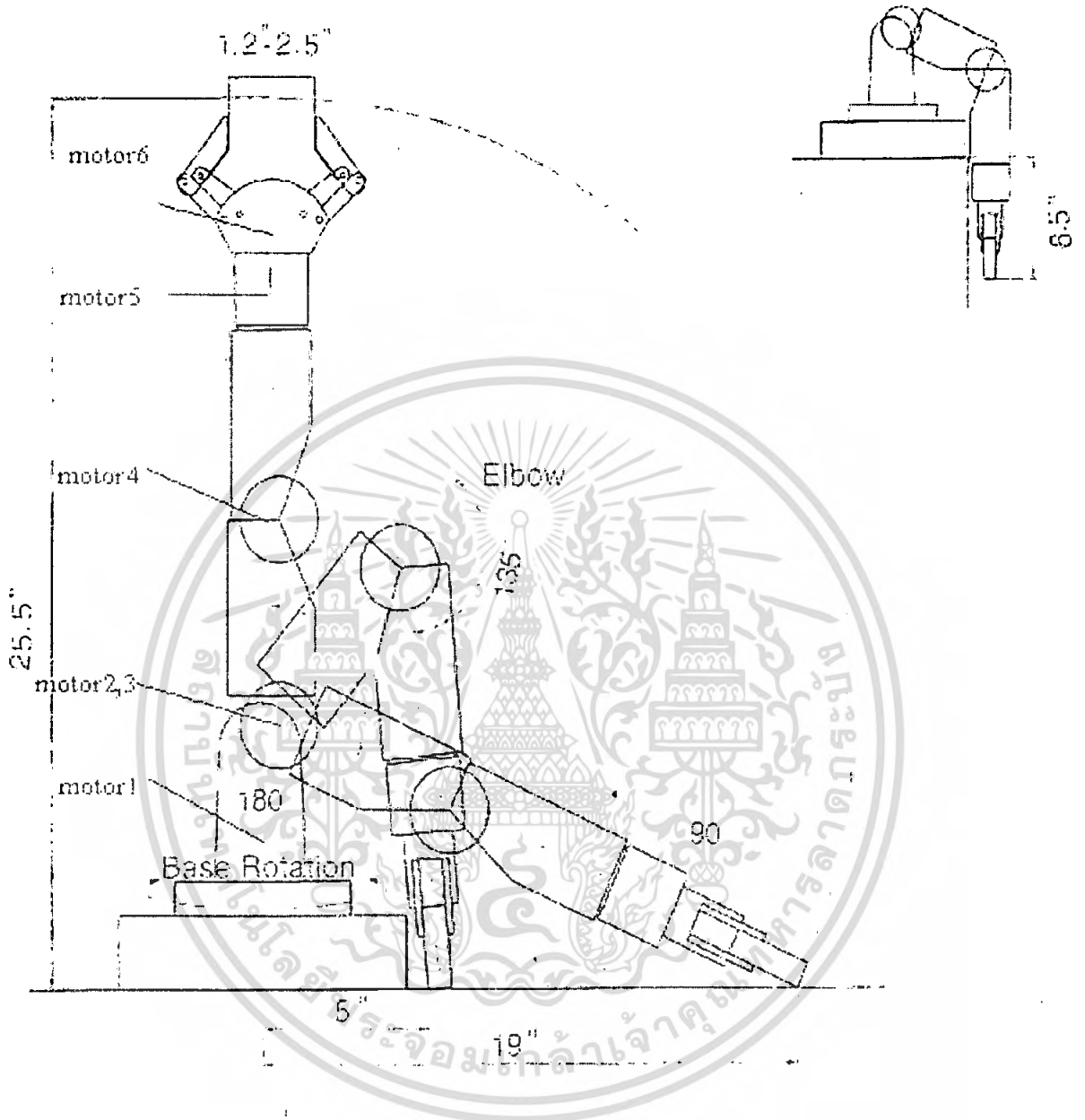
ตำแหน่งที่ 5 ทำหน้าที่คล้ายกับข้อมือ ใช้เพื่อให้แขนกลจำลองสามารถจับได้มากขึ้นแต่ในโครงการนี้จะกำหนดให้ Servo motor ตำแหน่งนี้ ขนานกับพื้นตลอดเวลาเพื่อให้เวลาจับนั้นกระชับขึ้น

ตำแหน่งที่ 6 เป็น Servo Motor ที่ทำหน้าที่ในการจับและปล่อยวัตถุที่ต้องการ

5.1.1 รายละเอียดของแขนกลจำลอง

1. ฐานกว้าง 6.5 นิ้ว สูง 4.5 นิ้ว
2. ระยะสูงสุดตามแนวแกน Z 25.5 นิ้ว
3. ระยะสูงสุดตามแนวแกน X 19 นิ้ว
4. ระยะสั้นสุดตามแนวแกน X 5 นิ้ว
5. ปากจับกว้างสุด 25 นิ้ว แคบสุด 1.25 นิ้ว
6. มุมในการหมุนฐาน 180 องศา

ซึ่งแขนกลที่พัฒนาขึ้นมาสามารถที่จะหยิบยกวัตถุขนาด 1.25 นิ้ว ถึง 2.5 นิ้วได้โดยสามารถที่จะหมุนได้ 180 องศาและแขนกลนี้สามารถจะเขียนโปรแกรมเพื่อควบคุมการทำงานได้



รูปที่ 5.2 แสดงภาพแขนกลที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การออกแบบวงจรควบคุม

บล็อกไดอะแกรมการทำงานของแขนกล

ป้อนค่าพิกัดลงในเครื่อง
คอมพิวเตอร์



เครื่องคอมพิวเตอร์ประมวลผลและส่ง
คำสั่งผ่านพอร์ตอนุกรม RS232



MAX 232 แปลงระบบแรงดัน ไฟฟ้าและส่ง
สัญญาณไปยัง MCS-51



MCS-51 รับคำสั่งและ
สร้างพัลส์สัญญาณ

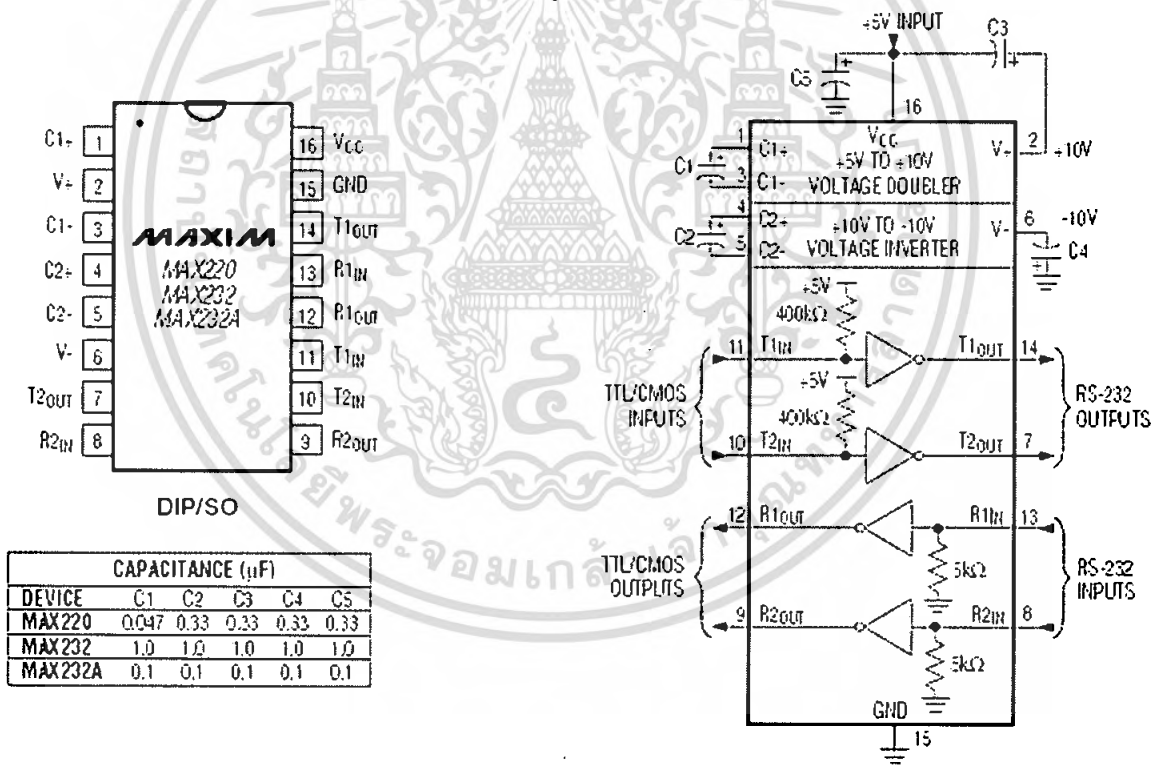


สัญญาณควบคุมถูกส่งไปยัง
SERVO MOTOR แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.1 การติดต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม

การสื่อสารผ่านพอร์ตอนุกรมนั้นจะใช้การสื่อสารในรูปแบบที่เรียกว่า RS - 232 ซึ่งสามารถที่จะใช้เป็นอุปกรณ์ติดต่อสื่อสารกับอุปกรณ์ภายนอกเครื่องคอมพิวเตอร์ได้ แต่ระบบ RS - 232 นั้นจะมีความแตกต่างจากระบบของ TTL (Transistor Transistor Logic) ที่เป็นระบบที่ใช้ในไมโครคอนโทรลเลอร์ ซึ่งลอจิก 1 ในระบบ TTL จะเป็นแรงดันไฟฟ้าระดับ 5 โวลต์ ส่วนในระบบ RS - 232 จะเป็นแรงดันขนาด -15 โวลต์ และระดับลอจิก 0 สำหรับ TTL จะเป็น 0 โวลต์ แต่ระบบ RS 232 จะเป็น 15 โวลต์ ดังนั้นในการติดต่อสื่อสารจึงต้องการใช้ตัวแปลงระดับเพื่อให้ระบบทั้งสองติดต่อกันได้ ดังนั้นในการนำข้อมูลจากคอมพิวเตอร์ไปประมวลผลในไมโครคอนโทรลเลอร์ จะต้องมีการแปลงระดับสัญญาณซึ่งในการแปลงระดับสัญญาณได้ใช้ไอซี MAX 232 ซึ่งไอซีจะรับสัญญาณ RS - 232 และแปลงเป็นTTL และในขณะเดียวกันก็สามารถแปลงระบบ TTL เป็น RS - 232 ได้เช่นกัน ซึ่งการจัดการขาต่างๆของไอซีและการใช้งานแสดงได้ดังรูปที่ 3.



รูปที่ 5.3 แสดงการจัดการขาและการนำไปใช้งานของ MAX232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

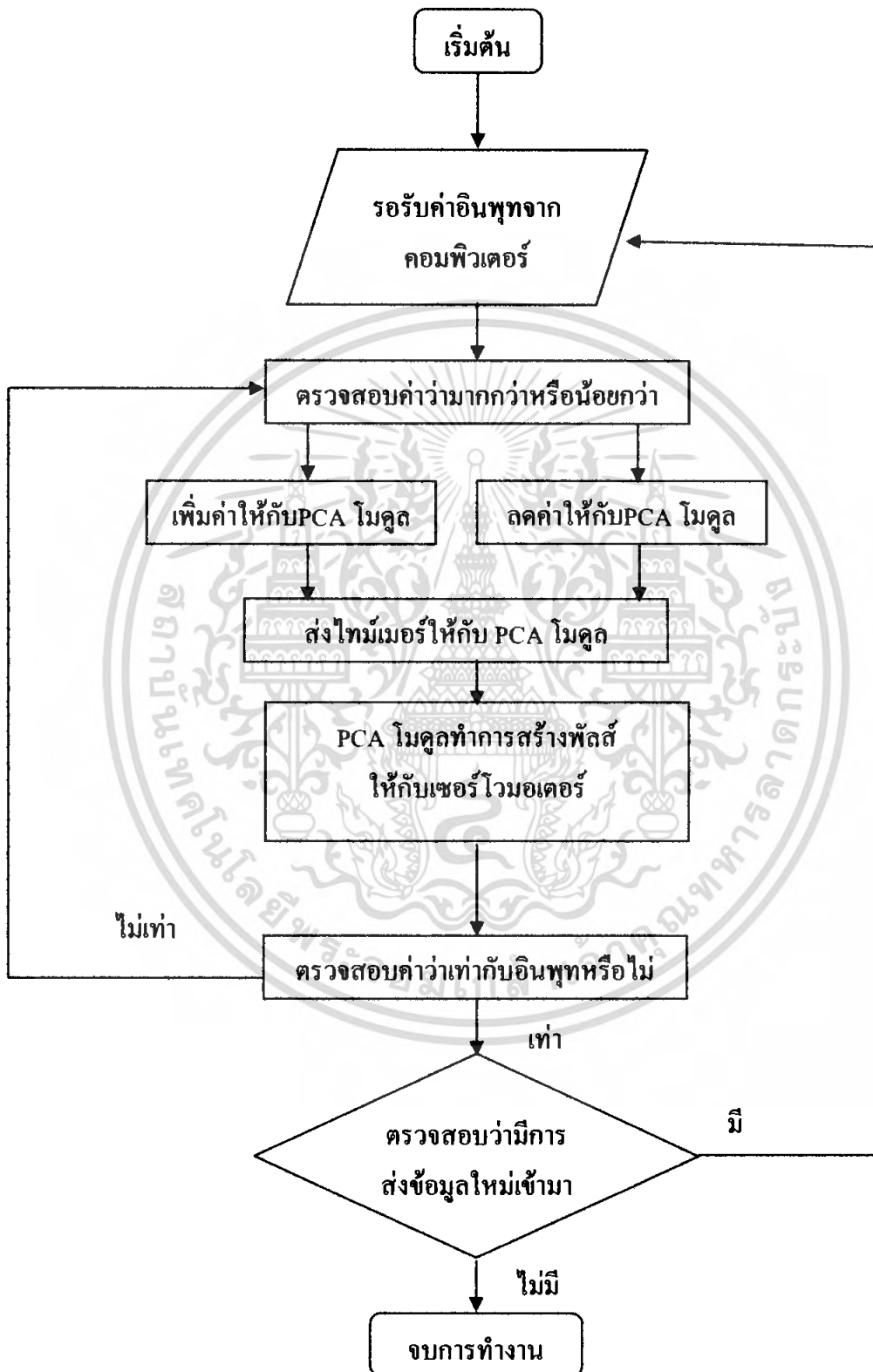
5.2.2 การสร้างพัลส์สัญญาณควบคุมเซอร์โวมอเตอร์

ในส่วนของการเขียน โปรแกรมภายใน MCS-51 เพื่อทำการควบคุมเซอร์โวมอเตอร์นั้นอาศัยหลักการ ควบคุมการทำงานของเซอร์โวมอเตอร์ซึ่งทำได้โดยการป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยอาศัย โมดูล PCA ใน MCS-51 เบอร์ P89V51RD2 ซึ่งสามารถสร้างพัลส์สัญญาณ (pulse Width Modulation หรือ PWM) ได้แม่นยำและมีประสิทธิภาพกว่าการใช้ timer ภายใน MCS-51 สร้างพัลส์สัญญาณ โดยใช้สัญญาณที่มีคาบอยู่ที่ 8 ms และมีความกว้างของพัลส์ในซีกบวกอยู่ในช่วง 0.5 ms – 2.5 ms ซึ่งค่าความกว้างของพัลส์ค่าต่างๆจะแสดงได้ดังตารางที่ 5.1

ความกว้างของพัลส์สัญญาณในซีกบวก	มุมของเซอร์โวมอเตอร์
0.5 ms	-90°
1.0 ms	-45°
1.5 ms	0°
2.0 ms	45°
2.5 ms	90°

ตารางที่ 5.1 แสดงค่ามุมของเซอร์โวมอเตอร์ที่ความกว้างพัลส์ค่าต่างๆ

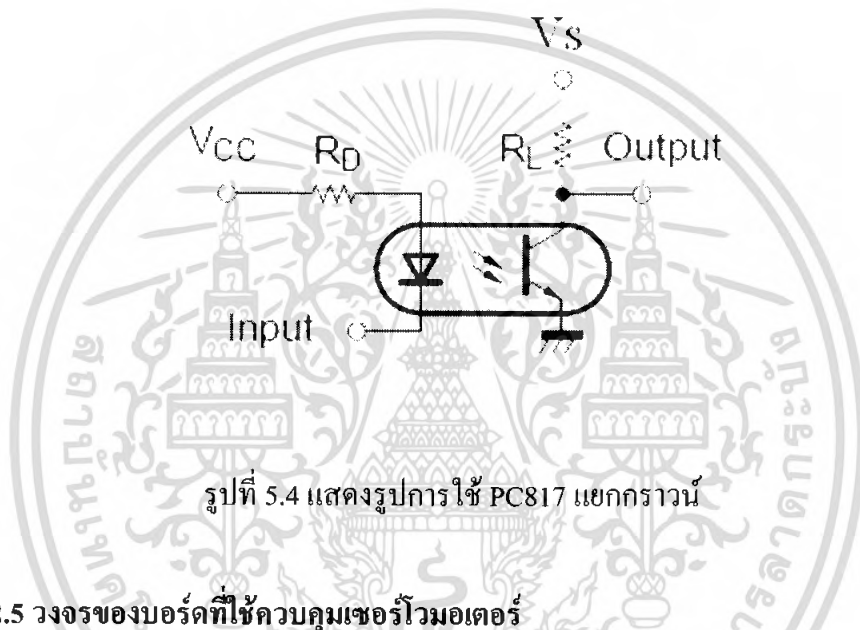
5.2.3 Flow chat การทำงานของไมโครคอนโทรลเลอร์ MCS-51



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4 การลดสัญญาณรบกวน

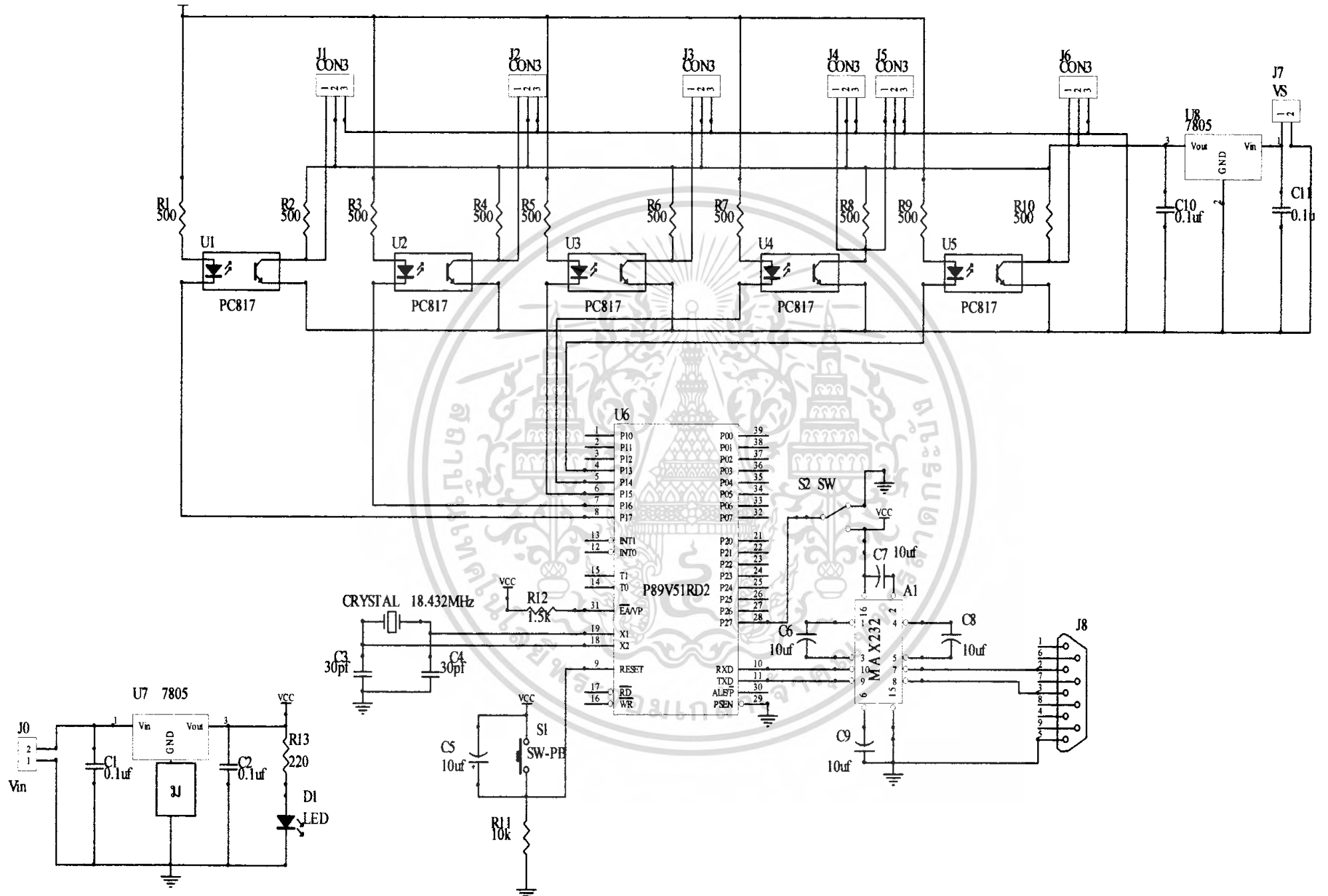
เมื่อได้สัญญาณที่ใช้ควบคุมมอเตอร์จาก MCS-51 มาแล้วนั้นเราไม่สามารถที่จะนำสัญญาณดังกล่าวป้อนเข้าที่สายสัญญาณของเซอร์โวมอเตอร์ได้โดยตรงเนื่องจากระดับแรงดันของสัญญาณควบคุมนั้นมีความแตกต่างกับระดับแรงดันของไฟเลี้ยงมอเตอร์จะทำให้เกิดสัญญาณรบกวนขึ้นจึงจำเป็นต้องทำการแยกกราวด์ (isolated) ในโครงการได้ใช้ ไอซีเบอร์ PC 817 ซึ่งอาศัยการนำสัญญาณด้วยแสง ทำการแยกกราวด์ของสัญญาณและมอเตอร์ออกจากกัน



รูปที่ 5.4 แสดงรูปการใช้ PC817 แยกกราวด์

5.2.5 วงจรของบอร์ดที่ใช้ควบคุมเซอร์โวมอเตอร์

บอร์ดควบคุมที่สร้างขึ้นจะประกอบด้วยส่วนที่เชื่อมต่อกับคอมพิวเตอร์และส่วนที่ใช้ควบคุมเซอร์โวมอเตอร์ โดยที่บอร์ดควบคุมจะทำหน้าที่รับคำสั่งจากคอมพิวเตอร์ส่งมายังไมโครคอนโทรเลอร์ จากนั้นไมโครคอนโทรเลอร์จะทำการประมวลผลแล้วส่งค่าออกไปเป็นความกว้างของพัลส์สัญญาณค่าต่างๆเพื่อไปควบคุมเซอร์โวมอเตอร์อีกทีหนึ่ง ซึ่งวงจรของบอร์ดควบคุมดังกล่าวได้แสดงไว้ดังรูปที่ 5.5



รูปที่ 5.5 แสดงวงจรของบอร์ดควบคุม

5.3 การออกแบบโปรแกรม

5.3.1 เป้าหมายของการออกแบบโปรแกรม

1. โปรแกรมจะมีหน้าจอ Interface คล้ายกับโปรแกรมทั่วไปเพื่อที่จะให้ user คุ้นเคยและสามารถทำความเข้าใจโปรแกรมได้อย่างรวดเร็ว
2. โปรแกรมจะสามารถรับค่าผ่านทาง Keyboard และการคลิกเมาส์ที่ภาพ
3. โปรแกรมจะต้องแจ้งข้อความต่าง ๆ ที่จำเป็นให้ User ทราบ เช่น error ต่าง ๆ หรือค่า output จากการทำงานของโปรแกรม

5.3.2 การออกแบบโปรแกรมในส่วน User Interface

ภาพรวมของโปรแกรมนั้นจะแบ่งออกเป็นสามส่วนหลักๆ ส่วนแรกจะทำงานผ่านการคลิกเมาส์ที่ภาพ ส่วนถัดมาเป็นส่วนของการเมตซิง และส่วนสุดท้ายเป็นการทำงานแบบปรับค่าเอง (Manual)

ในส่วนการทำงานผ่านการคลิกเมาส์ที่ภาพนั้นจะเป็นการกำหนดตำแหน่งของวัตถุผ่านการคลิกที่รูป คือเมื่อเราคลิกที่รูปโปรแกรมจะทำการแปลงพิกัดภาพไปเป็นพิกัดโลก ซึ่งค่าที่ได้ออกมาเป็น รัศมี ความสูง และองศาที่แกนกลต้องกวาดไป จากนั้นโปรแกรมจะทำการคำนวณมุมองศาของเซอร์ไวมอเตอร์แต่ละตัว และนำค่ามุมองศาไปเปรียบเทียบกับตารางไทเมอร์ เพื่อส่งค่าไทเมอร์ให้กับไมโครคอนโทรเลอร์อีกทีหนึ่ง

ในส่วนของการทำงานโดยการเมตซิง โปรแกรมจะทำการหาส่วนประกอบของภาพที่เหมือนกับภาพที่เราเลือกไว้มากที่สุด จากนั้นเมื่อหาพบแล้วโปรแกรมก็จะทำการแปลงพิกัดภาพไปเป็นพิกัดโลก ซึ่งจะได้ค่าออกมาเป็น รัศมี ความสูง และองศาที่แกนกลต้องกวาดไป จากนั้นโปรแกรมจะทำการคำนวณมุมองศาของเซอร์ไวมอเตอร์แต่ละตัว และนำค่ามุมองศาไปเปรียบเทียบกับตารางไทเมอร์ เพื่อส่งค่าไทเมอร์ให้กับไมโครคอนโทรเลอร์อีกทีหนึ่ง

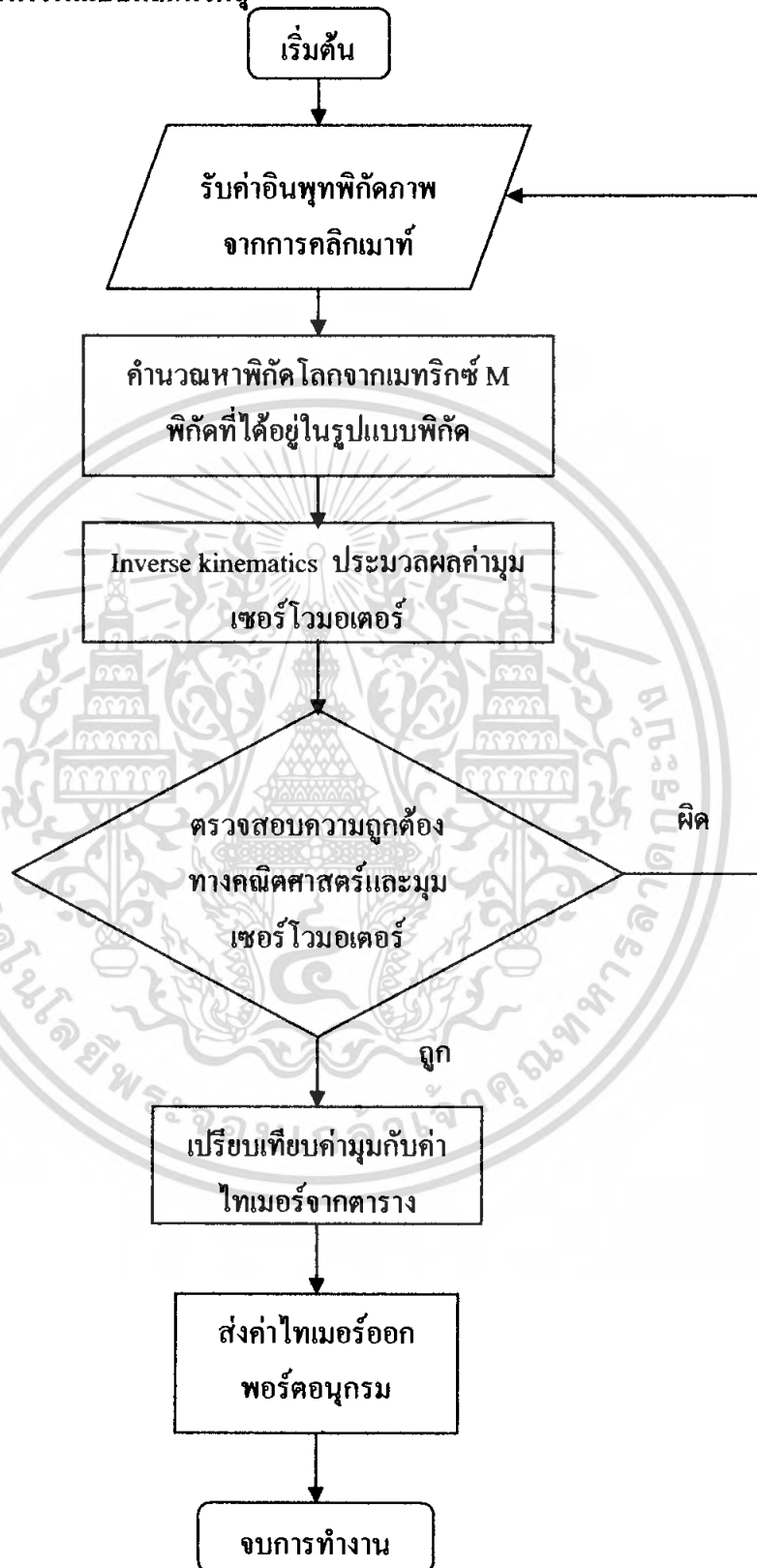
ในส่วนของการทำงานแบบปรับค่าเองนั้น ผู้ใช้สามารถที่จะเลื่อนสกรอร์บาร์ เพื่อเลือกมุมที่เซอร์ไวมอเตอร์แต่ละตัวต้องทำการหมุน พร้อมทั้งแสดงค่าไทเมอร์ที่ส่งให้กับไมโครคอนโทรเลอร์



รูปที่ 5.6 แสดงโปรแกรมที่ใช้ควบคุมแขนกล

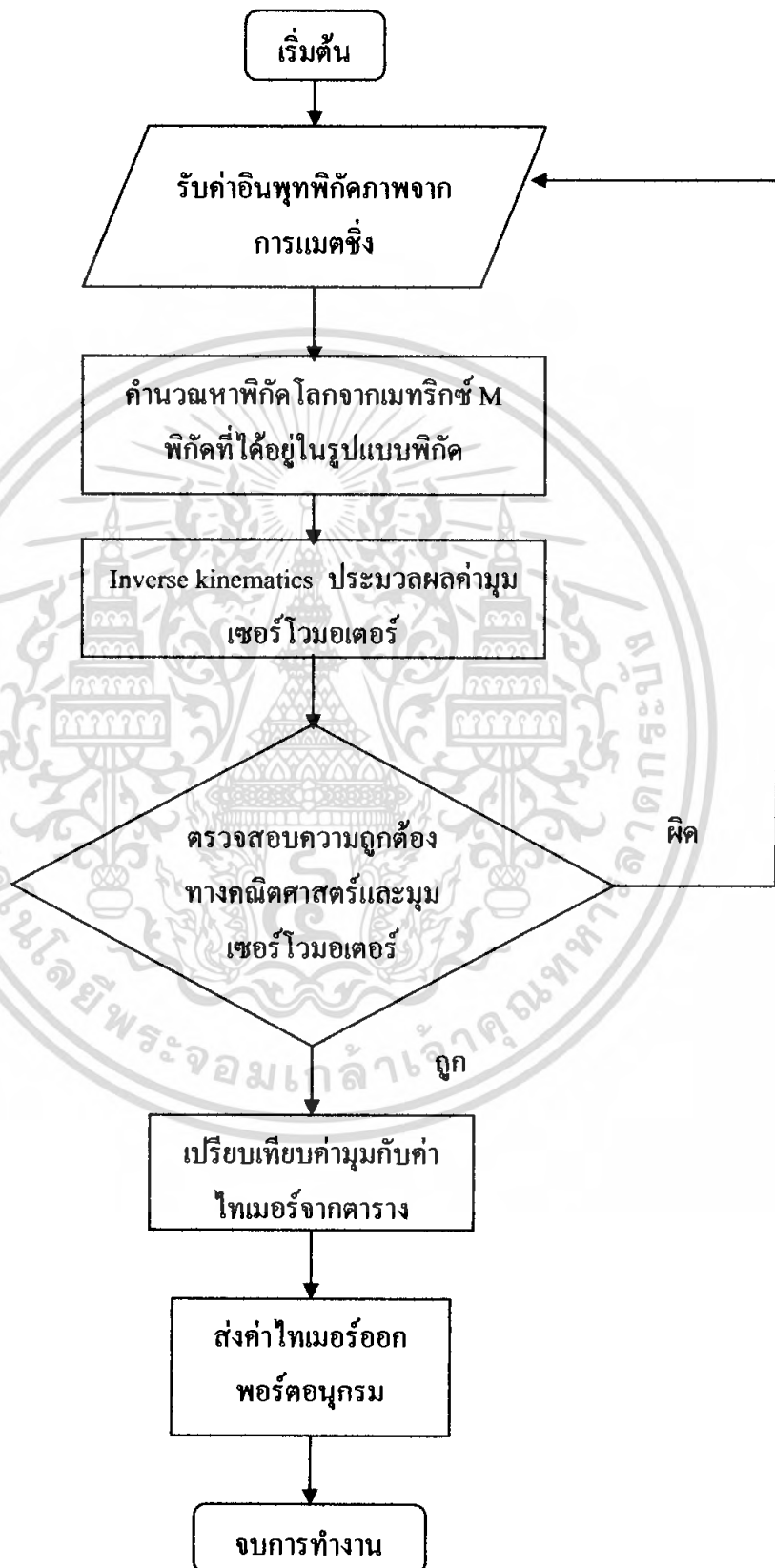
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 Flow chat การทำงานแบบคลิกที่วัตถุ



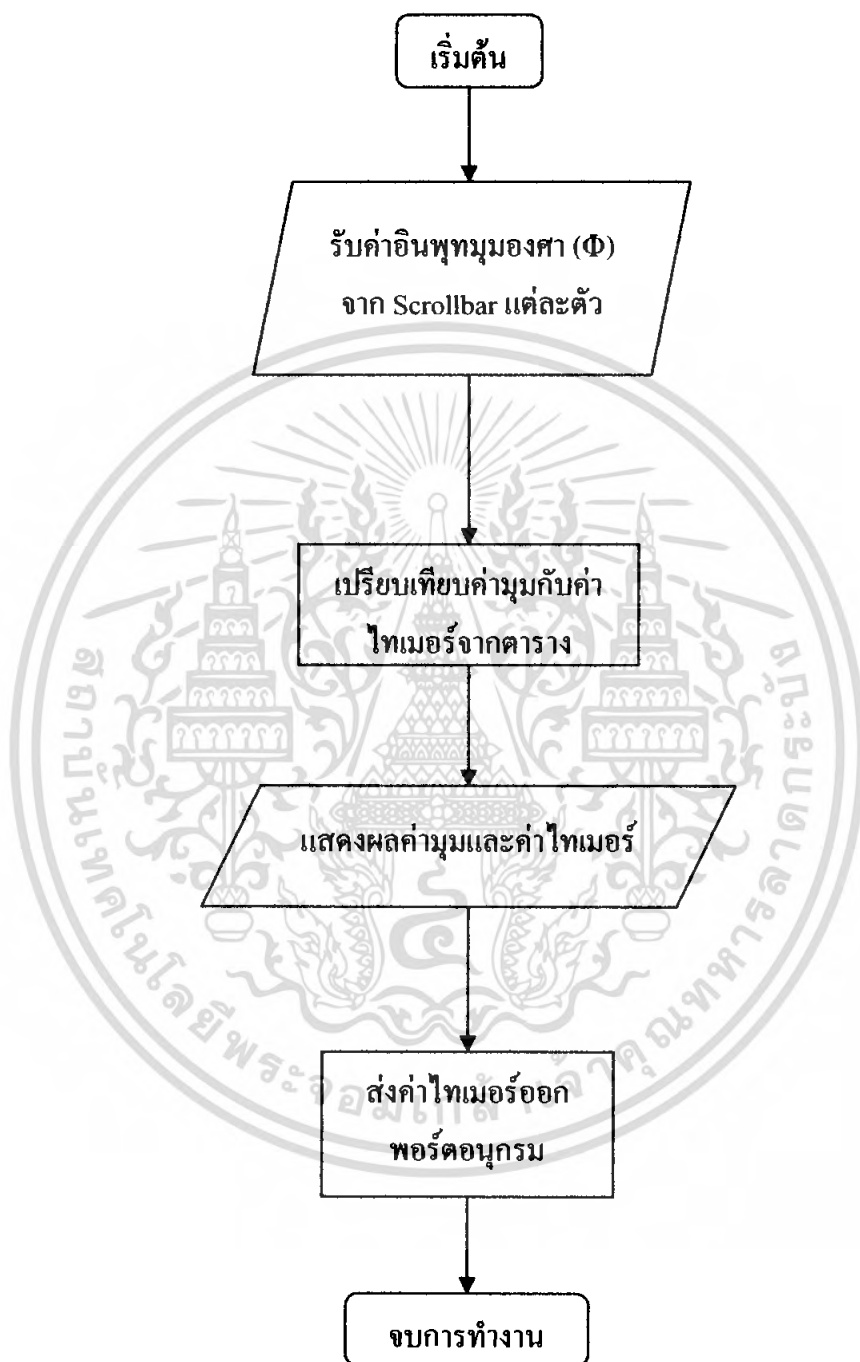
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 Flow chat การทำงานแบบแมตซ์ซิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 Flow Chat การทำงานแบบปรับค่าเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

การทดลองที่ 1

ทำการวัดความกว้างของพัลส์สัญญาณ ในซีกบวคของสัญญาณควบคุมเปรียบเทียบกับตำแหน่งของเซอร์โวมอเตอร์ซึ่งแสดงผลได้ดังตาราง

ความกว้าง ของพัลส์ สัญญาณใน ซีกบวค	มุมของ มอเตอร์ (ทฤษฎี)	มุมของ มอเตอร์ ตัวที่1 (ทดลอง)	มุมของ มอเตอร์ ตัวที่2 (ทดลอง)	มุมของ มอเตอร์ ตัวที่3 (ทดลอง)	มุมของ มอเตอร์ ตัวที่4 (ทดลอง)	มุมของ มอเตอร์ ตัวที่5 (ทดลอง)	มุมของ มอเตอร์ ตัวที่6 (ทดลอง)
0.5 ms	-90°	-90°	-90°	-90°	-90°	-90°	-90°
1.0 ms	-45°	-45°	-45°	-45°	-45°	-45°	-45°
1.5 ms	0°	0°	0°	0°	0°	0°	0°
2.0 ms	45°	45°	45°	45°	45°	45°	45°
2.5 ms	90°	90°	90°	90°	90°	90°	90°

ตารางที่ 6.1 แสดงผลการทดลองที่ 1

การทดลองที่ 2

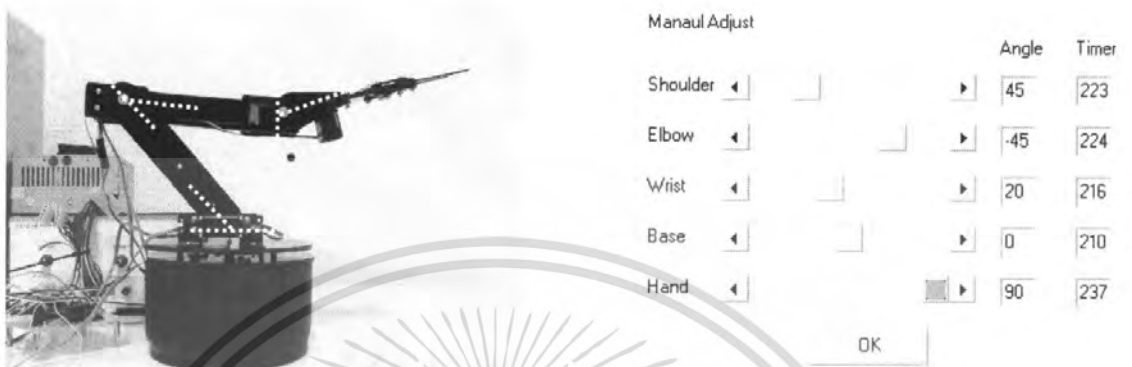
ทำการทดสอบการใช้งาน โปรแกรมควบคุมแขนกล โดยแบ่งเป็น 2 ส่วน คือ การทำงานของระบบปรับด้วยมือและการทำงานของระบบอัตโนมัติ

การทดลองที่ 2.1 การทำงานของระบบปรับด้วยมือ



รูปที่ 6.1 ภาพแสดงตำแหน่งและค่ามุมต่างๆของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 ภาพแสดงตำแหน่งและค่ามุมต่างๆของแขนกล



รูปที่ 6.3 ภาพแสดงตำแหน่งและค่ามุมต่างๆของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2.2 การทำงานของระบบอัตโนมัติ

Auto				Angle	Timer
R	15	Inch	Shoulder	-57	192
Z	3	Inch	Elbow	-9	213
ψ	0	degree	Wrist	66	229
OK			Base	0	210

รูปที่ 6.4 แสดงพิกัดของวัตถุและค่ามุมต่างๆของเซอร์โวมอเตอร์



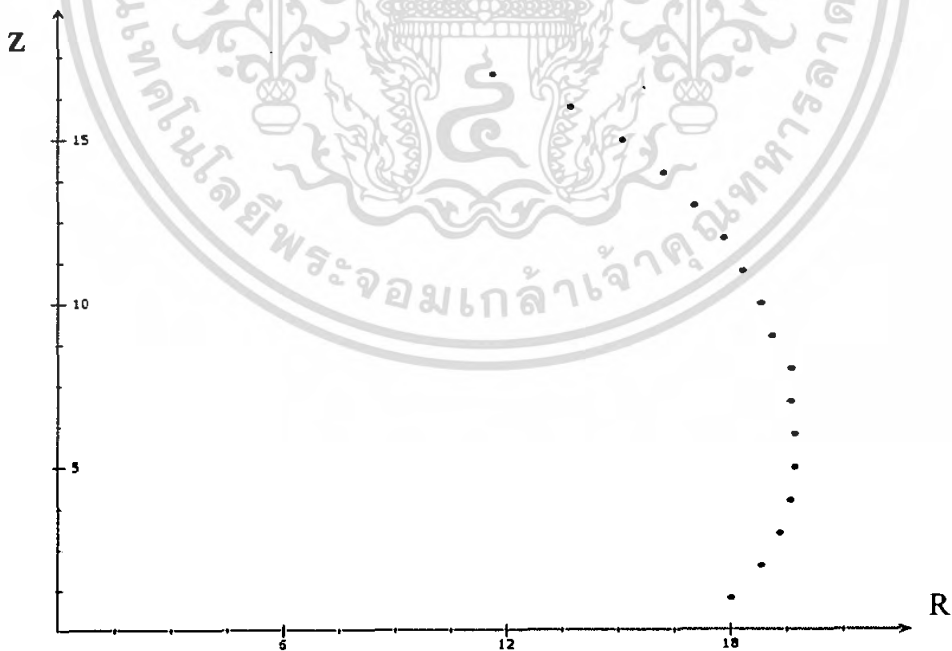
รูปที่ 6.5 แสดงภาพแขนกลเมื่อหยิบวัตถุที่มีน้ำหนักประมาณ 300 กรัมและภาพแขนกลที่นำวัตถุไปยังตำแหน่งปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3

ทำการทดสอบเพื่อหาขอบเขตการทำงานของแขนกลโดยทำการทดสอบทุกช่วงการใช้งาน ตั้งแต่ $0 - 180^\circ$

ความสูง(นิ้ว)	ระยะที่สามารถไปได้ (นิ้ว)	ความสูง(นิ้ว)	ระยะที่สามารถไปได้ (นิ้ว)
1	18	10	18.8
2	18.8	11	18.3
3	19.3	12	17.8
4	19.6	13	17
5	19.7	14	16.2
6	19.7	15	15.1
7	19.6	16	13.7
8	19.6	17	11.6
9	19.1		



รูปที่ 6.6 กราฟแสดงผลการทดลองที่ 3 ใน 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4

ทำการทดสอบความแม่นยำของตำแหน่งและการจับวัตถุของแขนกลโดยที่วัตถุมีน้ำหนักประมาณ 300 กรัม และมีการกำหนดให้แขนกลจับวัตถุที่ตำแหน่งต่างๆที่ระบุพิกัดใน โปรแกรม ซึ่งได้ผลดังนี้

พิกัดของวัตถุ			จำนวนครั้งที่ทดลอง	จำนวนครั้งที่จับได้	เปอร์เซ็นต์ความผิดพลาด
r(นิ้ว)	θ(องศา)	Z(นิ้ว)			
12	0	3	20	19	5%
12	20	3	20	18	10%
12	60	3	20	20	0%
12	90	3	20	20	0%
14	0	3	20	20	0%
14	20	3	20	19	5%
14	60	3	20	18	10%
14	90	3	20	19	5%
16	0	3	20	19	5%
16	20	3	20	19	5%
16	60	3	20	18	10%
16	90	3	20	18	10%

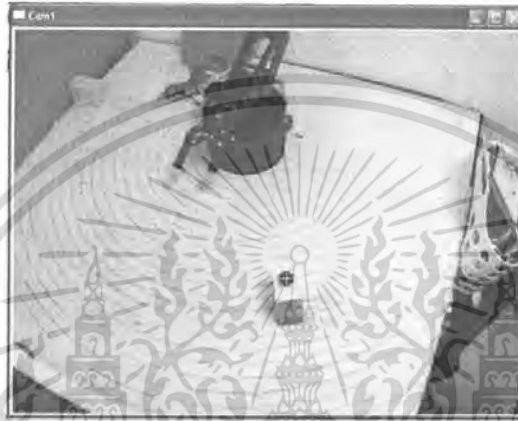
ตารางที่ 6.2 แสดงผลการทดลองที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

ทำการทดลองเพื่อทดสอบความแม่นยำของการระบุตำแหน่งวัตถุของโปรแกรมที่ใช้ควบคุมแขนกล(วัตถุมีน้ำหนักประมาณ 300 กรัม)

การทดลอง 5.1 ระบุตำแหน่งวัตถุผ่านหน้าจอกอมพิวเตอร์โดยใช้การคลิกเมาส์ที่วัตถุ



r(นิ้ว)			Ø(องศา)			Z(นิ้ว)		
ค่าจริง	ค่าจากการทดลอง	% ความผิดพลาด	ค่าจริง	ค่าจากการทดลอง	% ความผิดพลาด	ค่าจริง	ค่าจากการทดลอง	% ความผิดพลาด
16	16.29	1.81	35	33.90	3.14	3.25	3.23	0.61
12	12.39	3.25	35	33.86	3.26	3.25	3.18	2.15
15	15.17	1.13	5	5.84	16.80	3.25	3.35	3.07
13	13.25	1.92	20	18.87	5.65	3.25	3.22	0.92
17	16.91	0.53	30	28.11	6.30	3.25	3.33	2.46
6	6.10	1.67	5	5.50	10.00	3.25	3.19	1.85
9	9.10	1.11	25	24.77	0.92	3.25	3.27	0.62
12	12.12	1.00	65	63.33	2.57	3.25	3.28	0.92
23	22.83	0.74	20	19.45	2.75	3.25	3.26	0.31
16	15.96	0.25	20	18.99	5.05	3.25	3.23	0.62

ตารางที่ 6.3 แสดงผลการทดลองที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลอง 5.2 ระบุตำแหน่งวัตถุผ่านหน้าจอคอมพิวเตอร์โดยใช้วิธีการแมตซ์วงวัตถุ



r(นิ้ว)			Ø(องศา)			Z(นิ้ว)		
ค่าจริง	ค่าจากการทดลอง	% ความผิดพลาด	ค่าจริง	ค่าจากการทดลอง	% ความผิดพลาด	ค่าจริง	ค่าจากการทดลอง	% ความผิดพลาด
13.5	13.56	0.44	20	19.12	4.40	3.25	3.23	0.61
12	12.06	0.50	10	10.02	0.20	3.25	3.22	0.92
12	11.87	1.08	30	28.61	4.63	3.25	3.25	0
16	15.80	1.25	45	42.62	5.29	3.25	3.27	0.61
18	17.70	1.67	5	5.50	10.00	3.25	3.24	0.30
16	16.18	1.13	35	33.43	4.49	3.25	3.23	0.61
5	4.96	0.80	35	32.21	7.97	3.25	3.26	0.30
15	14.86	0.93	15	15.10	0.67	3.25	3.28	0.92
20	19.00	5.00	25	24.38	2.48	3.25	3.21	1.23
16	16.41	2.56	40	38.36	4.10	3.25	3.22	0.92

ตารางที่ 6.4 แสดงผลการทดลองที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

ทำการทดสอบความแม่นยำของตำแหน่งและการจับวัตถุของแขนกลโดยสั่งงานผ่านโปรแกรม ด้วยวิธีการคลิกเมาส์ที่วัตถุ (วัตถุน้ำหนักประมาณ 300 กรัม)

พิกัดของวัตถุ			จำนวนครั้งที่ทดลอง	จำนวนครั้งที่จับได้	เปอร์เซ็นต์ความผิดพลาด
r(นิ้ว)	θ(องศา)	Z(นิ้ว)			
12	0	3	20	18	10%
12	20	3	20	17	15%
12	60	3	20	18	10%
12	90	3	20	19	5%
14	0	3	20	19	5%
14	20	3	20	17	15%
14	60	3	20	18	10%
14	90	3	20	17	15%
16	0	3	20	19	5%
16	20	3	20	18	10%
16	60	3	20	17	15%
16	90	3	20	16	20%

ตารางที่ 6.5 แสดงผลการทดลองที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7

ทำการทดสอบความแม่นยำของตำแหน่งและการจับวัตถุของแขนกลโดยสั่งงานผ่านโปรแกรมด้วยวิธีการแมตซ์วัตถุ (วัตถุมีน้ำหนักประมาณ 300 กรัม)

พิกัดของวัตถุ			จำนวนครั้งที่ทดลอง	จำนวนครั้งที่จับได้	เปอร์เซ็นต์ความผิดพลาด
r(นิ้ว)	Ø(องศา)	Z(นิ้ว)			
12	0	3	20	19	5%
12	20	3	20	17	15%
12	60	3	20	18	10%
12	90	3	20	18	10%
14	0	3	20	19	5%
14	20	3	20	18	10%
14	60	3	20	18	10%
14	90	3	20	17	15%
16	0	3	20	18	10%
16	20	3	20	16	20%
16	60	3	20	18	10%
16	90	3	20	17	15%

ตารางที่ 6.6 แสดงผลการทดลองที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 8

ทำการใช้โปรแกรมแมทแล็บเพื่อเปรียบเทียบค่าตำแหน่งต่างๆของวัตถุ



	x		y		z	
ค่าจริง	ค่าจากการรีโปรเจก	ค่าจริง	ค่าจากการรีโปรเจก	ค่าจริง	ค่าจากการรีโปรเจก	
0	-0.83	0	-0.18	0	-0.43	
100	100.16	0	0.00	0	0.62	
100	100.54	100	100.10	0	0.64	
0	-0.05	100	100.06	0	-0.91	
0	0.45	0	0.43	82	82.68	
100	100.10	0	-0.26	82	81.0	
100	99.21	100	100.17	82	81.83	
0	0.29	100	99.67	82	82.46	

ตารางที่ 6.7 แสดงผลการทดลองที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทสรุปและข้อคิดเห็น

ในบทนี้จะได้ทำการสรุปโครงการทั้งหมดที่ได้ดำเนินการมา โดยเริ่มต้นจะอธิบายถึงหลักการ ทำงานคร่าวๆของระบบ จากนั้นจะได้อธิบายถึง ปัญหาและอุปสรรคที่พบ วิธีการแก้ปัญหา รวมทั้ง ข้อเสนอแนะเกี่ยวกับโครงการ

7.1 หลักการทำงาน

โครงการนี้แบ่งการทำงานออกเป็นสองส่วนใหญ่คือ การพัฒนาในส่วนของฮาร์ดแวร์ ซึ่งก็คือ การพัฒนาแขนกลและการพัฒนาในส่วนซอฟต์แวร์ ซึ่งก็คือ การพัฒนาโปรแกรมเพื่อควบคุมแขนกล

ในส่วนแรกนั้นคือการพัฒนาแขนกล ก่อนการพัฒนาแขนกลนั้นจำเป็นต้องมีความรู้เกี่ยวกับ เซอร์โวมอเตอร์และการออกแบบวงจรที่ใช้ในการควบคุมแขนกล ซึ่งสรุปพอสังเขปได้ดังนี้

- เซอร์โวมอเตอร์ เป็นอุปกรณ์หลักในการพัฒนาแขนกลเพราะจะทำหน้าที่ส่งแรงบิดและหมุน ไปยังตำแหน่งต่าง ๆ ซึ่งการหมุนของเซอร์โวมอเตอร์นั้นมาจากค่าที่รับเข้าไปผ่านสายสัญญาณแล้วไป เปรียบเทียบกับค่าเวลาที่อยู่ภายในถ้าหากไม่เท่ากันก็จะทำให้เกิดการหมุนเพื่อให้ค่าทั้งสองเท่ากันและ ทุก ๆ 20 ms จะมีการตรวจสอบค่าที่ส่งเข้ามาอีก ถ้าหากไม่มีค่าที่ส่งเข้าไปเซอร์โวมอเตอร์จะถือว่าค่าที่ ส่งเข้าไปเป็น 0 ทำให้ เซอร์โวมอเตอร์หมุนมาอยู่ที่ตำแหน่งเริ่มต้น

- วงจรควบคุม เป็นบอร์ดที่ใช้ในการควบคุม เซอร์โวมอเตอร์เป็นวงจรที่รับสัญญาณจากพอร์ต อนุกรม ซึ่งจะมี ไมโครคอนโทรเลอร์MCS-51 ทำหน้าที่สร้างพัลส์สัญญาณและส่งให้เซอร์โวมอเตอร์ แต่ละตัวซึ่งถ้าไม่มีสัญญาณเข้ามาก็จะทำการส่งค่าเดิม ไปให้กับเซอร์โวมอเตอร์ต่อไปเรื่อยๆ

หลังจากได้ทำการศึกษาความสัพันธ์ต่าง ๆ แล้วก็ได้ทำการออกแบบแขนกลขึ้น โดยเริ่มต้น จากการร่างรูปแบบของแขนกลและกำหนดจุดติดตั้งเซอร์โวมอเตอร์จากนั้นจึงขึ้นแบบโดยใช้กระดาษ แข็งแล้วจึงวาดแบบของจริงออกมาตัดแบบ ซึ่งงานส่วนใหญ่จะอยู่ที่งานฝีมือในการเจาะและประกอบ

ในส่วนที่สองก็คือการพัฒนาโปรแกรมเพื่อควบคุมแขนกลจำลอง ในการพัฒนาโปรแกรมจะใช้ BorlandC++ Builder ในการพัฒนา ซึ่งการพัฒนาโปรแกรมก็จะมีอยู่สองระบบก็คือ ระบบอัตโนมัติ และระบบปรับค่าเอง โดยที่ระบบอัตโนมัติที่อาศัยค่าพิกัดที่ได้จากการศึกษาเรื่อง จลนศาสตร์การเคลื่อนที่ มาแปลงเป็นค่าตำแหน่ง ให้กับ เซอร์โวมอเตอร์ในการเปลี่ยนค่าพิกัดไปเป็นค่าตำแหน่งของ เซอร์โวมอเตอร์ จากนั้นก็กำหนด ตำแหน่งให้กับ แขนกลเพื่อให้ แขนกลเคลื่อนที่ไปตามตำแหน่งที่กำหนด

7.2 ปัญหาและอุปสรรคที่พบระหว่างดำเนินงาน

- ความชำนาญในงานประดิษฐ์ไม่พอ ในการพัฒนาแขนกลนั้นจำเป็นต้องอาศัยความชำนาญและความประณีต ในการเจาะ แม้กระทั่งการประกอบชิ้นส่วนต่าง ๆ เข้าด้วยกัน ซึ่งทางผู้พัฒนาขาดทักษะในจุดนี้ทำให้แขนกลที่ได้ไม่สวยงามเท่าที่ควร

- ความจำกัดของอุปกรณ์ ในการพัฒนาแขนกลนั้น เซอร์โวมอเตอร์เป็นอุปกรณ์ที่ขาดไม่ได้เลย และเนื่องด้วยชิ้นส่วนซึ่ง หาได้ยากทำให้เวลาที่ซื้ของเซอร์โวมอเตอร์เสียหาย จำเป็นต้องใช้เวลาค่อนข้างมากในการซื้อมาเปลี่ยน ทำให้การดำเนินงานในส่วนอื่นล่าช้าไปด้วย

- อุปกรณ์เสียหายอยู่เนืองนิตย์ เวลาที่ทำการทดสอบโปรแกรมกับแขนกลจำลองบางครั้งทำให้ชุดเกียร์ของ เซอร์โวมอเตอร์เกิดความเสียหายและต้องเปลี่ยนแต่ด้วยอุปสรรคในปัญหาข้างต้น จึงเกิดความล่าช้า และสาเหตุที่ อุปกรณ์เสียหายอาจมีหลายสาเหตุ เช่น แขนกลเคลื่อนที่เร็วเกินไป เกิดติดขัดระหว่างการเคลื่อนที่

- การทำงานระหว่างซอฟต์แวร์กับฮาร์ดแวร์มีปัญหา ในการส่งค่าไปควบคุมการทำงานของแขนกลจำลอง แขนกลจำลอง ไม่สามารถทำงานตามที่ซอฟต์แวร์ ต้องการ ได้ เช่นเคลื่อนที่ไปคนละจุดกับที่โปรแกรมคำนวณได้

- ปัญหาในการควบคุมแขนกลจำลอง เมื่อทำการส่งค่าไปควบคุมแขนกลจำลอง ปรากฏว่าแขนกลเคลื่อนที่ไม่ถูกต้องตามที่กำหนดเท่าใดนักและระหว่างการเคลื่อนที่นั้น แขนกลจำลองจะสั้นอยู่ตลอดเวลา ซึ่งอาจจะเกิดจากกระแสไฟฟ้าที่ส่งไปไม่เพียงพอ

7.3 แนวทางในการแก้ไข้ปัญหา

ในการทดลองต่าง ๆ ระหว่างการทำโครงการนั้นมีปัญหาหลายอย่างเกิดขึ้น ปัญหาส่วนใหญ่นั้นเป็นปัญหาทางด้านเทคนิคมีแนวทางการแก้ไข้ปัญหาดังนี้

- เปลี่ยนรูปแบบของโปรแกรม ในบางครั้งผู้พัฒนาไม่สามารถที่จะพัฒนาโปรแกรมให้มีความสามารถได้ตามที่กำหนด จึงจำเป็นต้องทำการยกเลิก การทำงานบางอย่าง เพื่อให้โปรแกรมสามารถที่จะใช้งานได้และไม่มีข้อผิดพลาด
- สอบถามผู้เชี่ยวชาญในภาษาที่ใช้ในการพัฒนาโปรแกรม ซึ่งในบางกรณีอาจจะมีวิธีการที่คิดว่าการเขียน โปรแกรมเพื่อที่จะให้ผลลัพธ์ออกมาเหมือนกัน หรือผู้พัฒนาเองไม่เข้าใจในการใช้ภาษาที่ใช้พัฒนาดังนั้นจำเป็นอย่างยิ่งที่จะต้องสอบถามจากผู้เชี่ยวชาญ
- การฉ่นของเซอร์ไวโมเตอร์ ซึ่งเกิดจากกระแสไฟฟ้าที่ส่งไปไม่เพียงพอจึงจัดหาแหล่งจ่ายไฟที่สามารถจ่ายกระแสได้สูงขึ้น
- หาข้อมูลต่าง ๆ ผ่านอินเทอร์เน็ต การทำงานบางอย่างมีบุคคลอื่นทำการพัฒนาไว้แล้ว ดังนั้นจึงเป็นการดีที่จะนำข้อมูลนั้น ๆ มาประยุกต์ในโครงการ เพื่อลดเวลาการทำงาน
- หาซื้ออุปกรณ์เพิ่ม ในการพัฒนาโครงการหากอุปกรณ์ในการพัฒนาเสียหาย และได้ทำการตรวจสอบแล้วว่าไม่สามารถแก้ไขได้จำเป็นอย่างยิ่งที่ต้องจัดซื้ออุปกรณ์เพิ่ม

7.4 ประโยชน์ที่ได้รับในการพัฒนาโครงการ

- มีความรู้ความเข้าใจเกี่ยวกับการทำงานของระบบแขนกล
- มีความรู้เกี่ยวกับการพัฒนาโปรแกรมด้วย Borland C++ Builder รวมถึงการรับและส่งค่าผ่านพอร์ตคอนุกรมด้วย
- มีความรู้ความเข้าใจเกี่ยวกับการทำงานของเซอร์ไวโมเตอร์และ ออกแบบวงจรที่ใช้ในการควบคุมแขนกล
- มีความเข้าใจในการเขียนโปรแกรมให้กับ ไมโครคอนโทรเลอร์ MCS-51 ในการรับค่าจากคอมพิวเตอร์และสร้างสัญญาณให้กับเซอร์ไวโมเตอร์

บรรณานุกรม

1. นกุล กระจาย , “ การเขียนโปรแกรมแบบวิซวลด้วย C++Builder 5 ”, สายส่งศึกษิต
เก็ตไทย , กรุงเทพฯ , 2544
2. เฉษฐธิ์ มณีธรรม , สำเร็จ เต็มราม , “ คัมภีร์ไมโครคอนโทรลเลอร์ MCS-51 (Microcontroller
MCS-51)” , บริษัท เกทีพี คอมพ์ แอนด์ คอนซัลท์ , กรุงเทพฯ , 2548
3. ยุทธนา ลีลาศวัฒนกุล , “ เริ่มต้นการเขียนโปรแกรมด้วยภาษา C++ ”, ดวงกลมสมัย , กรุงเทพฯ ,
2547
4. วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, “ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ฉบับ
P89C51RD2 = Philips's P89C51RD2 Microcontroller experiment book” กรุงเทพฯ : อิน โนว
ตีฟ เอ็กเพอริเมนต์
5. รศ.ดร.ชูชาติ ปิณฑวิรุจน์ “เอกสารประกอบการสอนวิชา Digital Image Processing”
6. <http://www lynxmotion.com/>
7. <http://www.thaizone99.com>
8. <http://www.ett.co.th>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P89V51RD2

8-bit 80C51 5 V low power 64 kB Flash microcontroller
with 1 kB RAM

Rev. 01 — 01 March 2004

Product data

1. General description

The P89V51RD2 is an 80C51 microcontroller with 64 kB Flash and 1024 bytes of data RAM.

A key feature of the P89V51RD2 is its X2 mode option. The design engineer can choose to run the application with the conventional 80C51 clock rate (12 clocks per machine cycle) or select the X2 mode (6 clocks per machine cycle) to achieve twice the throughput at the same clock frequency. Another way to benefit from this feature is to keep the same performance by reducing the clock frequency by half, thus dramatically reducing the EMI.

The Flash program memory supports both parallel programming and in serial In-System Programming (ISP). Parallel programming mode offers gang-programming at high speed, reducing programming costs and time to market. ISP allows a device to be reprogrammed in the end product under software control. The capability to field/update the application firmware makes a wide range of applications possible.

The P89V51RD2 is also In-Application Programmable (IAP), allowing the Flash program memory to be reconfigured even while the application is running.

2. Features

- 80C51 Central Processing Unit
- 5 V Operating voltage from 0 to 40 MHz
- 64 kB of on-chip Flash program memory with ISP (In-System Programming) and IAP (In-Application Programming)
- Supports 12-clock (default) or 6-clock mode selection via software or ISP
- SPI (Serial Peripheral Interface) and enhanced UART
- PCA (Programmable Counter Array) with PWM and Capture/Compare functions
- Four 8-bit I/O ports with three high-current Port 1 pins (16 mA each)
- Three 16-bit timers/counters
- Programmable Watchdog timer (WDT)
- Eight interrupt sources with four priority levels
- Second DPTR register
- Low EMI mode (ALE inhibit)
- TTL- and CMOS-compatible logic levels

- Brown-out detection
- Low power modes
 - ◆ Power-down mode with external interrupt wake-up
 - ◆ Idle mode
- PDIP40, PLCC44 and TQFP44 packages

3. Ordering information

Table 1: Ordering information

P89V51RD2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V51RD2FBC	TQFP44	plastic thin quad flat package; 44 leads	SOT376-1
P89V51RD2BN	PDIP40	plastic dual in-line package; 40 leads	SOT129-1

3.1 Ordering options

Table 2: Ordering options

P89V51RD2FA	-40 °C to +85 °C	0 to 40 MHz
P89V51RD2FBC	-40 °C to +85 °C	
P89V51RD2BN	0 °C to +70 °C	

4. Block diagram

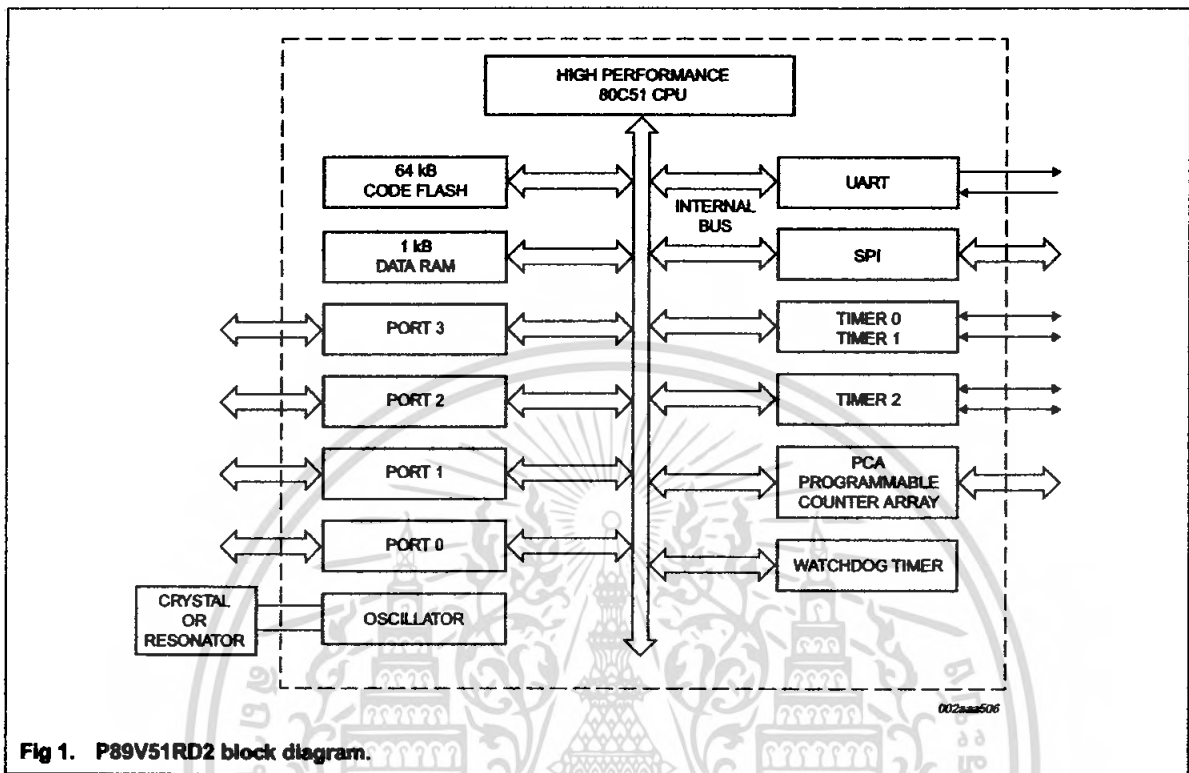
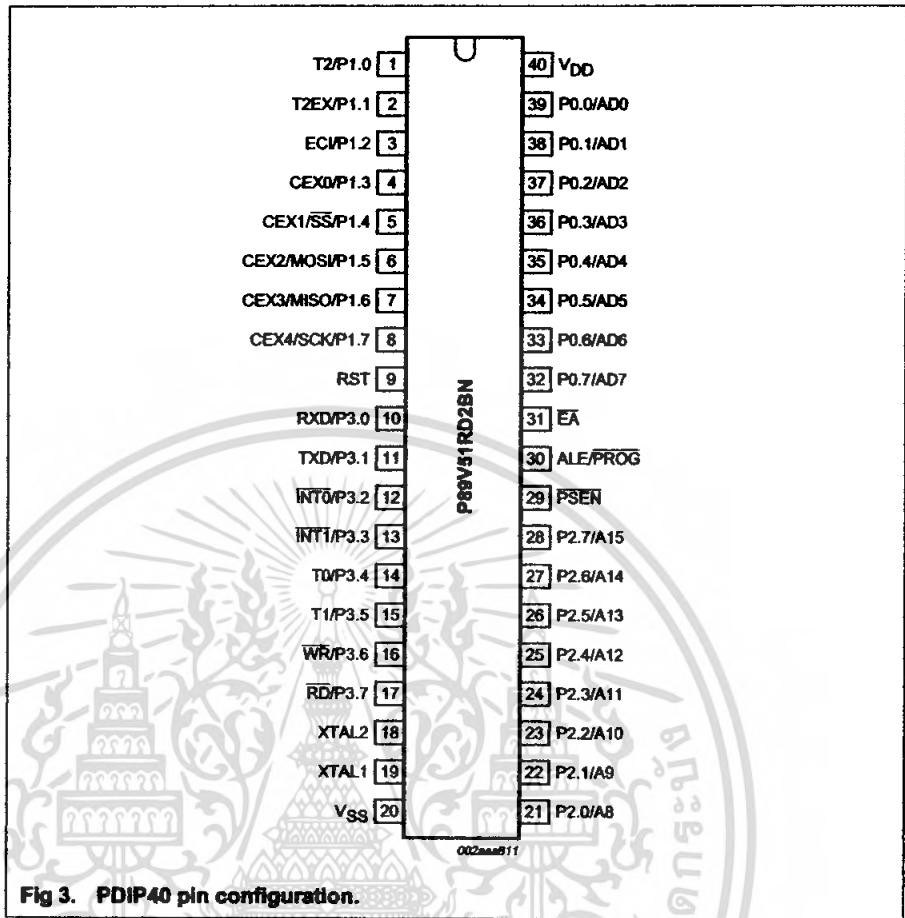


Fig 1. P89V51RD2 block diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 Pin description

Table 3: P89V51RD2 pin description

P0.0 to P0.7	39-32	37-30	43-36	I/O	
					Port 0: Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have '1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external code and data memory. In this application, it uses strong internal pull-ups when transitioning to '1's. Port 0 also receives the code bytes during the external host mode programming, and outputs the code bytes during the external host mode verification. External pull-ups are required during program verification or as a general purpose I/O port.
P1.0 to P1.7	1-8	40-44, 1-3	2-9	I/O with internal pull-up	Port 1: Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 pins are pulled high by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 1 pins that are externally pulled LOW will source current (I _{IL}) because of the internal pull-ups. P1.5, P1.6, P1.7 have high current drive of 16 mA. Port 1 also receives the low-order address bytes during the external host mode programming and verification.
P1.0	1	40	2	I/O	T2: External count input to Timer/Counter 2 or Clock-out from Timer/Counter 2
P1.1	2	41	3	I	T2EX: Timer/Counter 2 capture/reload trigger and direction control
P1.2	3	42	4	I	ECI: External clock input. This signal is the external clock input for the PCA.
P1.3	4	43	5	I/O	CEX0: Capture/compare external I/O for PCA Module 0. Each capture/compare module connects to a Port 1 pin for external I/O. When not used by the PCA, this pin can handle standard I/O.
P1.4	5	44	6	I/O	SS: Slave port select input for SPI CEX1: Capture/compare external I/O for PCA Module 1
P1.5	6	1	7	I/O	MOSI: Master Output Slave Input for SPI CEX2: Capture/compare external I/O for PCA Module 2
P1.6	7	2	8	I/O	MISO: Master Input Slave Output for SPI CEX3: Capture/compare external I/O for PCA Module 3
P1.7	8	3	9	I/O	SCK: Master Output Slave Input for SPI CEX4: Capture/compare external I/O for PCA Module 4

Table 3: P89V51RD2 pin description...continued

P2.0 to P2.7	21-28	18-25	24-31	I/O with internal pull-up	Port 2: Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 2 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. Port 2 sends the high-order address byte during fetches from external program memory and during accesses to external Data Memory that use 16-bit address (MOVX@DPTR). In this application, it uses strong internal pull-ups when transitioning to '1's. Port 2 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0 to P3.7	10-17	5, 7-13	11, 13-19	I/O with internal pull-up	Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 3 pins that are externally pulled LOW will source current (I_{IL}) because of the internal pull-ups. Port 3 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0	10	5	11	I	RXD: serial input port
P3.1	11	7	13	O	TXD: serial output port
P3.2	12	8	14	I	INT0: external interrupt 0 input
P3.3	13	9	15	I	INT1: external interrupt 1 input
P3.4	14	10	16	I	T0: external count input to Timer/Counter 0
P3.5	15	11	17	I	T1: external count input to Timer/Counter 1
P3.6	16	12	18	O	WR: external data memory write strobe
P3.7	17	13	19	O	RD: external data memory read strobe
PSEN	29	26	32	I/O	Program Store Enable: PSEN is the read strobe for external program memory. When the device is executing from internal program memory, PSEN is inactive (HIGH). When the device is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. A forced HIGH-to-LOW input transition on the PSEN pin while the RST input is continually held HIGH for more than 10 machine cycles will cause the device to enter external host mode programming.
RST	9	4	10	I	Reset: While the oscillator is running, a HIGH logic state on this pin for two machine cycles will reset the device. If the PSEN pin is driven by a HIGH-to-LOW input transition while the RST input pin is held HIGH, the device will enter the external host mode, otherwise the device will enter the normal operation mode.

Table 3: P89V51RD2 pin description...continued

EA	31	29	35	I	External Access Enable: EA must be connected to V _{SS} in order to enable the device to fetch code from the external program memory. EA must be strapped to V _{DD} for internal program execution. However, Security lock level 4 will disable EA, and program execution is only possible from internal program memory. The EA pin can tolerate a high voltage of 12 V.
ALE/ PROG	30	27	33	I/O	Address Latch Enable: ALE is the output signal for latching the low byte of the address during an access to external memory. This pin is also the programming pulse input (PROG) for flash programming. Normally the ALE ^[1] is emitted at a constant rate of 1/6 the crystal frequency ^[2] and can be used for external timing and clocking. One ALE pulse is skipped during each access to external data memory. However, if AO is set to '1', ALE is disabled.
NC	-	6, 17, 28, 39	1, 12, 23, 34	I/O	No Connect
XTAL1	19	15	21	I	Crystal 1: Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	14	20	O	Crystal 2: Output from the inverting oscillator amplifier.
V _{DD}	40	38	44	I	Power supply
V _{SS}	20	16	22	I	Ground

- [1] ALE loading issue: When ALE pin experiences higher loading (>30 pF) during the reset, the microcontroller may accidentally enter into modes other than normal working mode. The solution is to add a pull-up resistor of 3 kΩ to 50 kΩ to V_{DD}, e.g., for ALE pin.
- [2] For 6-clock mode, ALE is emitted at 1/3 of crystal frequency.

6. Special function registers

Remark: Special Function Registers (SFRs) accesses are restricted in the following ways:

- User must not attempt to access any SFR locations not defined.
- Accesses to any defined SFR locations must be strictly for the functions for the SFRs.
- SFR bits labeled '-', '0' or '1' can only be written and read as follows:
 - '-' Unless otherwise specified, must be written with '0', but can return any value when read (even if it was written with '0'). It is a reserved bit and may be used in future derivatives.
 - '0' must be written with '0', and will return a '0' when read.
 - '1' must be written with '1', and will return a '1' when read.



Table 4: Special function registers...continued
** indicates SFRs that are bit addressable.*

FST	Flash Status Register	B6	-	SB	-	-	-	EDC	-	-	-
		Bit address	AF	AE	AD	AC	AB	AA	A9	A8	
IEN0*	Interrupt Enable 0	A8H	EA	EC	ET2	ES0	ET1	EX1	ET0	EX0	
		Bit address	EF	EE	ED	EC	EB	EA	E9	E8	
IEN1*	Interrupt Enable 1	E8H	-	-	-	-	EBO				
		Bit address	BF	BE	BD	BC	BB	BA	B9	B8	
IP0*	Interrupt Priority	B8H	-	PPC	PT2	PS	PT1	PX1	PT0	PX0	
IP0H	Interrupt Priority 0 HIGH	B7H	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	
		Bit address	FF	FE	FD	FC	FB	FA	F9	F8	
IP1*	Interrupt Priority 1	F8H	-	-	-	-	PBO				
IP1H	Interrupt Priority 1 HIGH	F7H	-	-	-	-	PBOH				
FCF		B1H	-	-	-	-	-	-	-	-	BSEL
		Bit address	87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	
		Bit address	97	96	95	94	93	92	91	90	
P1*	Port 1	90H	CEX4/ SPICLK	CEX3/ MISO	CEX2/ MOSI	CEX1/SS	CEX0	ECI	T2EX	T2	
		Bit address	A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	
		Bit address	B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD	
PCON	Power Control Register	87H	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL	
		Bit address	D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	
RCAP2H	Timer2 Capture HIGH	CBH									
RCAP2L	Timer2 Capture LOW	CAH									
		Bit address	9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial Port Control	98H	SM0/FE_	SM1	SM2	REN	TB8	RB8	TI	RI	
SBUF	Serial Port Data Buffer Register	99H									

Table 4: Special function registers...continued
 * indicates SFRs that are bit addressable.

			[REDACTED]							
SADDR	Serial Port Address Register	A9H								
SADEN	Serial Port Address Enable	B9H								
		Bit address	87⁽¹⁾	86⁽¹⁾	85⁽¹⁾	84⁽¹⁾	83⁽¹⁾	82⁽¹⁾	81⁽¹⁾	80⁽¹⁾
SPCTL	SPI Control Register	D5H	SPIE	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
SPCFG	SPI Configuration Register	AAH	SPIF	SPWCOL	-	-	-	-	-	-
SPDAT	SPI Data	86H								
SP	Stack Pointer	81H								
		Bit address	8F	8E	8D	8C	8B	8A	89	88
TCON*	Timer Control Register	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
		Bit address	CF	CE	CD	CC	CB	CA	C9	C8
T2CON*	Timer2 Control Register	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
T2MOD	Timer2 Mode Control	C9H	-	-	ENT2				T2OE	DCEN
TH0	Timer 0 HIGH	8CH								
TH1	Timer 1 HIGH	8DH								
TH2	Timer 2 HIGH	CDH								
TL0	Timer 0 LOW	8AH								
TL1	Timer 1 LOW	8BH								
TL2	Timer 2 LOW	CCH								
TMOD	Timer 0 and 1 Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0
WDTC	Watchdog Timer Control	C0H	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT
WDTD	Watchdog Timer Data/Reload	85H								

[1] Unimplemented bits in SFRs (labeled '-') are 'X's (unknown) at all times. Unless otherwise specified, '1's should not be written to these bits since they may be used for other purposes in future derivatives. The reset values shown for these bits are '0's although they are unknown when read.

$$\text{Period} = (255 - \text{WDTD}) \cdot 344064 \cdot 1/f_{\text{CLK}} (\text{XTAL1})$$

where WDTD is the value loaded into the WDTD register and f_{osc} is the oscillator frequency.

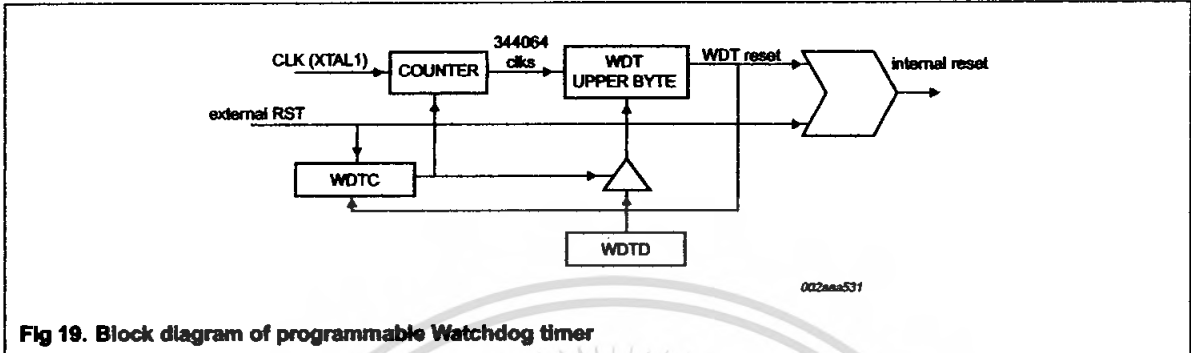


Fig 19. Block diagram of programmable Watchdog timer

Table 33: WDT C - Watchdog control register (address COH) bit allocation
Bit addressable; Reset value: 00H

Symbol	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT
--------	---	---	---	-------	------	------	-----	------

Table 34: WDT C - Watchdog control register (address COH) bit description

7 to 5	-	Reserved for future use. Should be set to '0' by user programs.
4	WDOUT	Watchdog output enable. When this bit and WDRE are both set, a Watchdog reset will drive the reset pin active for 32 clocks.
3	WDRE	Watchdog timer reset enable. When set enables a Watchdog timer reset.
2	WDTS	Watchdog timer reset flag, when set indicates that a WDT reset occurred. Reset in software.
1	WDT	Watchdog timer refresh. Set by software to force a WDT reset.
0	SWDT	Start Watchdog timer, when set starts the WDT. When cleared, stops the WDT.

7.8 Programmable Counter Array (PCA)

The PCA includes a special 16-bit Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P1.3 (CEX0), module 1 to P1.4 (CEX1), etc. Registers CH and CL contain current value of the free running up counting 16-bit PCA timer. The PCA timer is a common time base for all five modules and can be programmed to run at: $\frac{1}{8}$ the oscillator frequency, $\frac{1}{2}$ the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR (see Table 35 and Table 36).

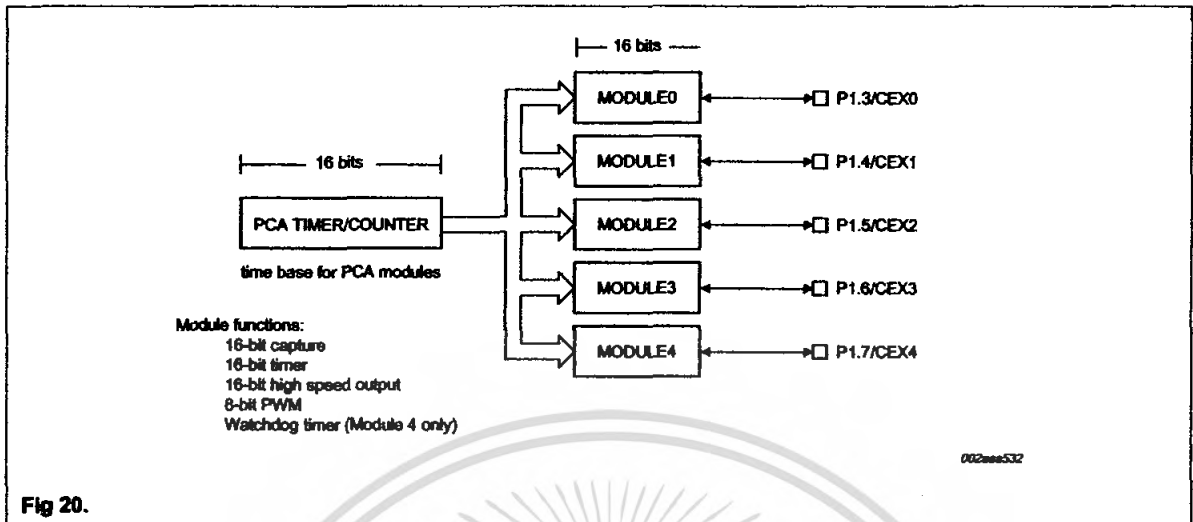


Fig 20.

In the CMOD SFR there are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the Watchdog function on module 4, and ECF which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows.

The Watchdog timer function is implemented in module 4 of PCA.

The CCON SFR contains the run control bit for the PCA (CR) and the flags for the PCA timer (CF) and each module (CCF4:0). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set. The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags can only be cleared by software. All the modules share one interrupt vector. The PCA interrupt system is shown in Figure 21.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. The registers contain the bits that control the mode that each module will operate in.

The ECCF bit (from CCAPMn.0 where n = 0, 1, 2, 3, or 4 depending on the module) enables the CCFn flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module (see Figure 21).

PWM (CCAPMn.1) enables the pulse width modulation mode.

The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register.

The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition.

The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

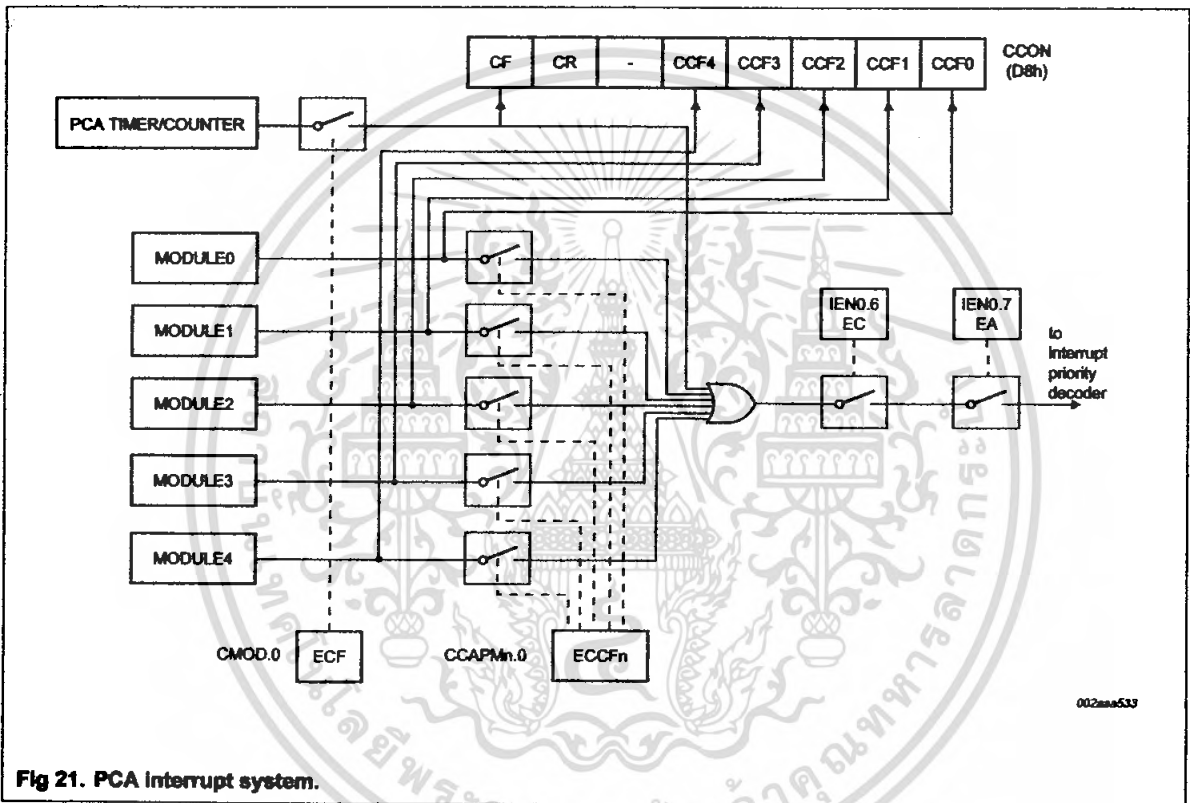


Fig 21. PCA interrupt system.

Table 35: CMOD - PCA counter mode register (address D9H) bit allocation
Not bit addressable; Reset value: 00H

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

Table 36: CMOD - PCA counter mode register (address D9H) bit description

7	CIDL	Counter Idle Control: CIDL = 0 programs the PCA Counter to continue functioning during idle Mode. CIDL = 1 programs it to be gated off during idle.
6	WDTE	Watchdog Timer Enable: WDTE = 0 disables Watchdog timer function on module 4. WDTE = 1 enables it.
5 to 3	-	Reserved for future use. Should be set to '0' by user programs.
2 to 1	CPS1, CPS0	PCA Count Pulse Select (see Table 37 below).
0	ECF	PCA Enable Counter Overflow Interrupt: ECF = 1 enables CF bit in CCON to generate an interrupt. ECF = 0 disables that function.

Table 37: CMOD - PCA counter mode register (address D9H) count pulse select

0	0	0 Internal clock, $f_{osc} / 6$
0	1	1 Internal clock, $f_{osc} / 6$
1	0	2 Timer 0 overflow
1	1	3 External clock at ECI/P1.2 pin (max rate = $f_{osc} / 4$)

Table 38: CCON - PCA counter control register (address 0D8H) bit allocation
Bit addressable; Reset value: 00H

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

Table 39: CCON - PCA counter control register (address 0D8H) bit description

7	CF	PCA Counter Overflow Flag. Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.
6	CR	PCA Counter Run Control Bit. Set by software to turn the PCA counter on. Must be cleared by software to turn the PCA counter off.
5	-	Reserved for future use. Should be set to '0' by user programs.
4	CCF4	PCA Module 4 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
3	CCF3	PCA Module 3 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
2	CCF2	PCA Module 2 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
1	CCF1	PCA Module 1 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
0	CCF0	PCA Module 0 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.

Table 40: CCAPMn - PCA modules compare/capture register (address CCAPM0 0DAH, CCAPM1 0DBH, CCAPM2 0DCH, CCAPM3 0DDH, CCAPM4 0DEH) bit alloc.

Not bit addressable; Reset value: 00H

Symbol	-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
--------	---	-------	-------	-------	------	------	------	-------

Table 41: CCAPMn - PCA modules compare/capture register (address CCAPM0 0DAH, CCAPM1 0DBH, CCAPM2 0DCH, CCAPM3 0DDH, CCAPM4 0DEH) bit desc.

7	-	Reserved for future use. Should be set to '0' by user programs.						
6	ECOMn	Enable Comparator. ECOMn = 1 enables the comparator function.						
5	CAPPn	Capture Positive, CAPPn = 1 enables positive edge capture.						
4	CAPNn	Capture Negative, CAPNn = 1 enables negative edge capture.						
3	MATn	Match. When MATn = 1 a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.						
2	TOGn	Toggle. When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.						
1	PWMn	Pulse Width Modulation Mode. PWMn = 1 enables the CEXn pin to be used as a pulse width modulated output.						
0	ECCFn	Enable CCF Interrupt. Enables compare/capture flag CCFn in the CCON register to generate an interrupt.						

Table 42: PCA module modes (CCAPMn register)

0	0	0	0	0	0	0	no operation
x	1	0	0	0	0	x	16-bit capture by a positive-edge trigger on CEXn
x	0	1	0	0	0	x	16-bit capture by a negative-edge trigger on CEXn
x	1	1	0	0	0	x	16-bit capture by any transition on CEXn
1	0	0	1	0	0	x	16-bit software timer
1	0	0	1	1	0	x	18-bit high speed output
1	0	0	0	0	1	0	8-bit PWM
1	0	0	1	x	0	x	Watchdog timer

7.8.1 PCA capture mode

To use one of the PCA modules in the capture mode (Figure 22) either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH).

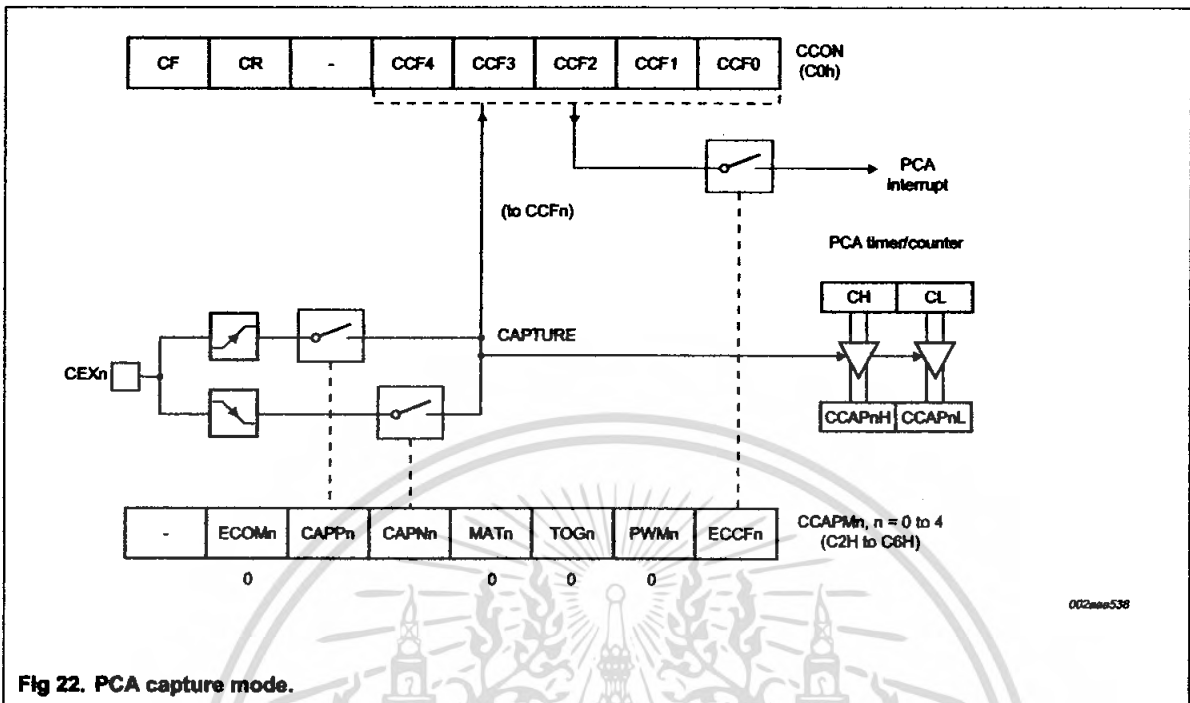
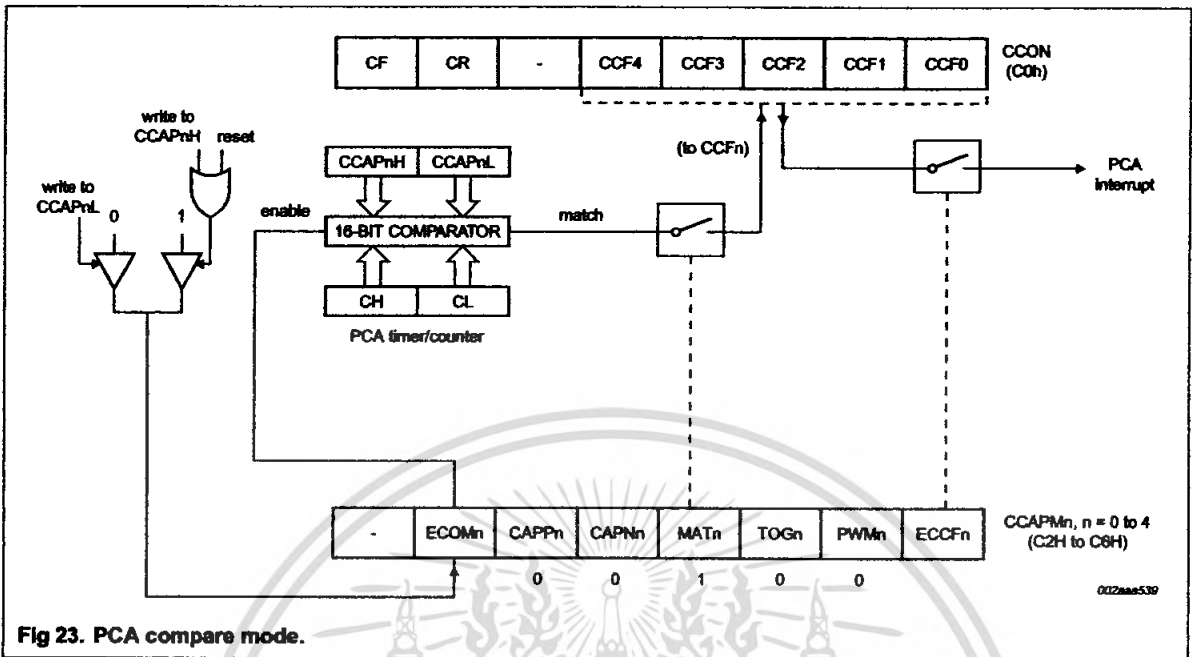


Fig 22. PCA capture mode.

If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated.

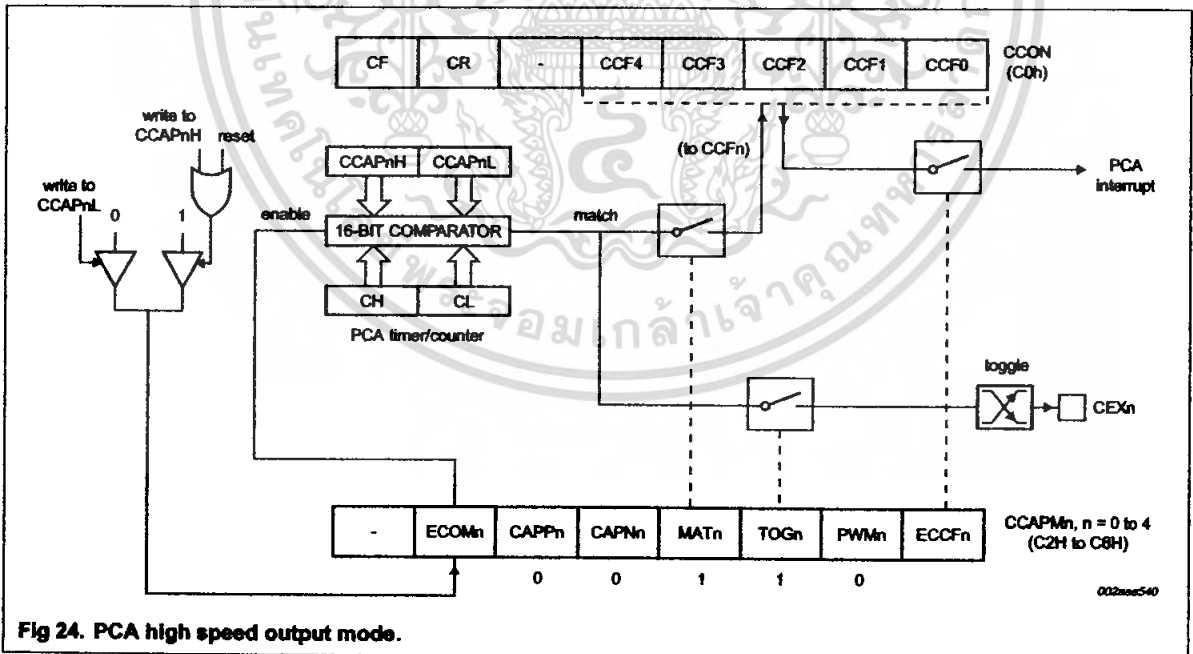
7.8.2 16-bit software timer mode

The PCA modules can be used as software timers (Figure 23) by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set.



7.8.3 High speed output mode

In this mode (Figure 24) the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set.



7.8.4 Pulse width modulator mode

All of the PCA modules can be used as PWM outputs (Figure 25). Output frequency depends on the source for the PCA timer.

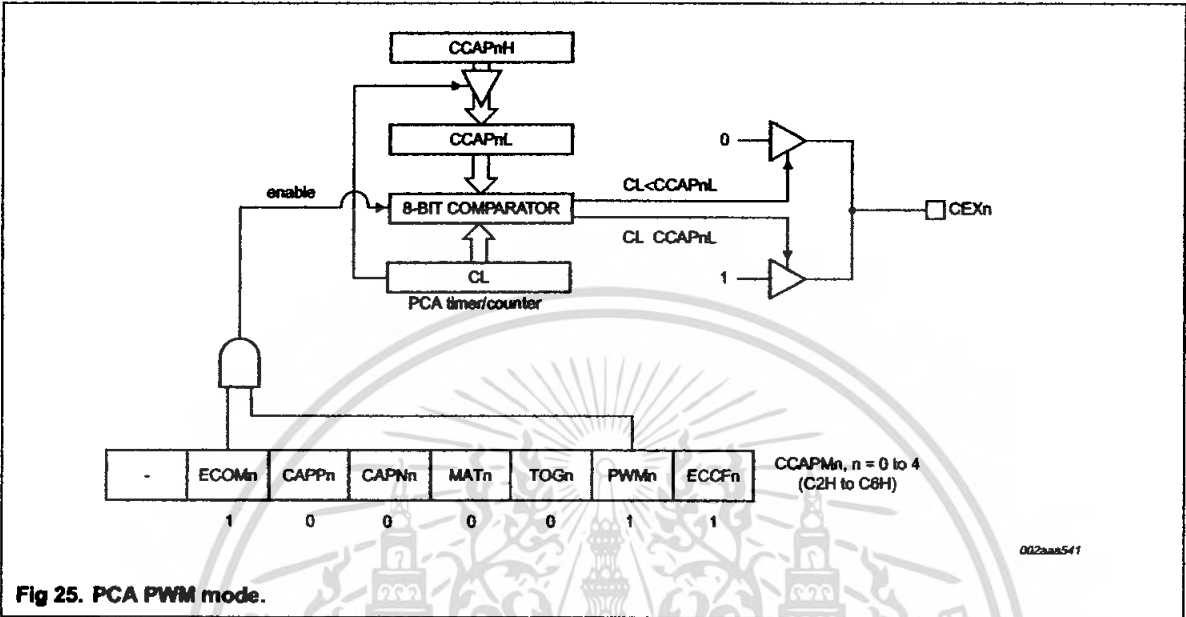


Fig 25. PCA PWM mode.

All of the modules will have the same frequency of output because they all share one and only PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPnL. When the value of the PCA CL SFR is less than the value in the module's CCAPnL SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPnL is reloaded with the value in CCAPnH, this allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

7.8.5 PCA Watchdog timer

An on-board Watchdog timer is available with the PCA to improve the reliability of the system without increasing chip count. Watchdog timers are useful for systems that are susceptible to noise, power glitches, or electrostatic discharge. Module 4 is the only PCA module that can be programmed as a Watchdog. However, this module can still be used for other modes if the Watchdog is not needed. Figure 25 shows a diagram of how the Watchdog works. The user pre-loads a 16-bit value in the compare registers. Just like the other compare modes, this 16-bit value is compared to the PCA timer value. If a match is allowed to occur, an internal reset will be generated. This will not cause the RST pin to be driven high.

user's software then must periodically change (CCAP4H,CCAP4L) to keep a match from occurring with the PCA timer (CH,CL). This code is given in the WATCHDOG routine shown above.

In order to hold off the reset, the user has three options:

8. Limiting values

Table 61: Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

$T_{amb(bias)}$	operating bias ambient temperature	-55	+125	°C
T_{stg}	storage temperature range	-65	+150	°C
V_{EA}	voltage on EA pin to V_{SS}	-0.5	14	V
V_n	DC voltage on any pin to ground potential	-0.5	$V_{DD} + 0.5$	V
V_{It}	transient voltage (<20 ns) on any other pin to V_{SS}	-1.0	$V_{DD} + 1.0$	V
$I_{OL(I/O)}$	maximum I_{OL} per I/O pins P1.5, P1.6, P1.7	-	20	mA
$I_{OL(I/O)}$	maximum I_{OL} per I/O for all other pins	-	15	mA
$P_{tot(pack)}$	total power dissipation per package	$T_{amb} = 25\text{ °C}$		-
	through hole lead soldering temperature	10 seconds	-	300 °C
	surface mount lead soldering temperature	3 seconds	-	240 °C
	output short circuit current	[1]	-	50 mA

[1] Outputs shorted for no more than one second. No more than one output shorted at a time. (Based on package heat transfer limitations, not device power consumption.)

9. Recommended operating conditions

Table 62: Operating range

T_{amb}	ambient temperature under bias			
	commercial	0	+70	°C
	Industrial	-40	+85	°C
V_{DD}	supply voltage	4.5	5.5	V
f_{osc}	oscillator frequency	0	40	MHz
	oscillator frequency for in-application programming	0.25	40	MHz

Table 63: Reliability characteristics

$N_{END}^{[1]}$	endurance	10,000	cycles	JEDEC Standard A117
$T_{DR}^{[1]}$	data retention	100	years	JEDEC Standard A103
$I_{LTH}^{[1]}$	latch up	$100 + I_{DD}$	mA	JEDEC Standard 78

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

Table 64: AC conditions of test^[1]

Input rise/fall time	10 ns
Output load	$C_L = 100$ pf

[1] See Figure 35 and Figure 37.

Table 65: Recommended system power-up timings

$T_{PU-READ}^{[1]}$	Power-up to read operation	100	μ s
$T_{PU-WRITE}^{[1]}$	Power-up to write operation	100	μ s

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

Table 66: Pin impedance
($V_{DD} = 3.3$ V, $T_{amb} = 25^\circ$ C, $f = 1$ MHz, other pins open)

$C_{I/O}^{[1]}$	I/O pin capacitance	$V_{I/O} = 0$ V	15	pF
$C_{IN}^{[1]}$	input capacitance	$V_{IN} = 0$ V	12	pF
L_{PIN}	pin inductance		20	nH

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

10. Static characteristics

Table 67: DC electrical characteristics

$T_{amb} = 0\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C or }-40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}; V_{DD} = 4.5\text{ V to }5.5\text{ V}; V_{SS} = 0\text{ V}$

V_{IL}	LOW-level input voltage	$4.5\text{ V} < V_{DD} < 5.5\text{ V}$	-0.5	$0.2V_{DD} - 0.1$	V	
V_{IH}	HIGH-level input voltage	$4.5\text{ V} < V_{DD} < 5.5\text{ V}$	$0.2V_{DD} + 0.9$	$V_{DD} + 0.5$	V	
V_{IH1}	HIGH-level input voltage (XTAL1, RST)	$4.5\text{ V} < V_{DD} < 5.5\text{ V}$	$0.7V_{DD}$	$V_{DD} + 0.5$	V	
V_{OL}	LOW-level output voltage (ports 1.5, 1.6, 1.7)	$V_{DD} = 4.5\text{ V}; I_{OL} = 16\text{ mA}$	-	1.0	V	
V_{OL}	LOW-level output voltage (ports 1, 2, 3) ^[1]	$V_{DD} = 4.5\text{ V}$				
		$I_{OL} = 100\text{ }\mu\text{A}$	-	0.3	V	
		$I_{OL} = 1.6\text{ mA}$	-	0.45	V	
		$I_{OL} = 3.5\text{ mA}$	-	1.0	V	
V_{OL1}	LOW-level output voltage (Port 0, ALE, PSEN) ^{[1][3]}	$V_{DD} = 4.5\text{ V}$				
		$I_{OL} = 200\text{ }\mu\text{A}$	-	0.3	V	
		$I_{OL} = 3.2\text{ mA}$	-	0.45	V	
V_{OH}	HIGH-level output voltage (ports 1, 2, 3, ALE, PSEN) ^[4]	$V_{DD} = 4.5\text{ V}$				
		$I_{OH} = -10\text{ }\mu\text{A}$	$V_{DD} - 0.3$	-	V	
		$I_{OH} = -30\text{ }\mu\text{A}$	$V_{DD} - 0.7$	-	V	
V_{OH1}	HIGH-level output voltage (Port 0 in External Bus Mode) ^[4]	$V_{DD} = 4.5\text{ V}$				
		$I_{OH} = -200\text{ }\mu\text{A}$	$V_{DD} - 0.3$	-	V	
V_{OH1}		$V_{DD} = 4.5\text{ V}$				
		$I_{OH} = -3.2\text{ mA}$	$V_{DD} - 0.7$	-	V	
V_{BOD}	brown-out detection voltage		3.65	4.15	V	
I_{IL}	logic 0 input current (ports 1, 2, 3)	$V_{IN} = 0.4\text{ V}$	-	-75	μA	
I_{TL}	logic 1-to-0 transition current (ports 1, 2, 3) ^[5]	$V_{IN} = 2\text{ V}$	-	-650	μA	
I_{LI}	input leakage current (Port 0)	$0.45\text{ V} < V_{IN} < V_{DD} - 0.3\text{ V}$	-	± 10	μA	
R_{RST}	RST pull-down resistor		40	225	$\text{k}\Omega$	
C_{IO}	pin capacitance ^[6]	@ 1 MHz, $T_{amb} = 25\text{ }^{\circ}\text{C}$	-	15	pF	
I_{DD}	power supply current	active mode	@ 12 MHz	-	11.5	mA
			@ 40 MHz	-	50	mA
	idle mode	@ 12 MHz	-	8.5	mA	
		@ 40 MHz	-	42	mA	
	Power-down mode (min. $V_{DD} = 2\text{ V}$)	$T_{amb} = 0\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C}$	-	80	μA	
		$T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$	-	90	μA	

[1] Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

- a) Maximum I_{OL} per 8-bit port: 28 mA
- b) Maximum I_{OL} total for all outputs: 71 mA
- c) If I_{OL} exceeds the test condition, V_{OH} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than $5\mu W$. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Next-Generation Device Features

- ◆ For Low-Voltage, Integrated ESD Applications
MAX3222E/MAX3232E/MAX3237E/MAX3241E/
MAX3246E: +3.0V to +5.5V, Low-Power, Up to
1Mbps, True RS-232 Transceivers Using Four
0.1 μF External Capacitors (MAX3246E Available
in a UCSP™ Package)
- ◆ For Low-Cost Applications
MAX221E: $\pm 15kV$ ESD-Protected, +5V, 1 μA ,
Single RS-232 Transceiver with AutoShutdown™

Applications

Portable Computers
Low-Power Modems
Interface Translation
Battery-Powered RS-232 Systems
Multidrop RS-232 Networks

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Ordering Information continued at end of data sheet.
*Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (μF)	SHDN & Three-State	Rx Active In SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	18-Pin Plastic DIP (derate 11.1mW/°C above +70°C)	889mW
V ₊ (Note 1)	(V _{CC} - 0.3V) to +14V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW
V ₋ (Note 1)	+0.3V to +14V	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
T _{IN}	-0.3V to (V _{CC} - 0.3V)	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN} (Except MAX220)	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
R _{IN} (MAX220)	±25V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
T _{OUT} (Except MAX220) (Note 2)	±15V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
T _{OUT} (MAX220)	±13.2V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
Output Voltages		Operating Temperature Ranges	
T _{OUT}	±15V	MAX2_AC, MAX2_C	0°C to +70°C
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	MAX2_AE, MAX2_E	-40°C to +85°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2_AM, MAX2_M	-55°C to +125°C
Continuous Power Dissipation (T _A = +70°C)		Storage Temperature Range	-65°C to +160°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	Lead Temperature (soldering, 10s) (Note 3)	+300°C

Note 1: For the MAX220, V₊ and V₋ can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Note 2: Input voltage measured with T_{OUT} in high-impedance state, SHDN or V_{CC} = 0V.

Note 3: Maximum reflow temperature for the MAX233A is +225°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C₁-C₄ = 0.1μF, MAX220, C₁ = 0.047μF, C₂-C₄ = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND	±5	±8		V	
Input Logic Threshold Low			1.4	0.8	V	
Input Logic Threshold High	All devices except MAX220	2	1.4		V	
	MAX220: V _{CC} = 5.0V	2.4				
Logic Pullup/Input Current	All except MAX220, normal operation		5	40	μA	
	SHDN = 0V, MAX222/MAX242, shutdown, MAX220		±0.01	±1		
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V, MAX222/MAX242		±0.01	±10	μA	
	V _{CC} = SHDN = 0V	V _{OUT} = ±15V	±0.01	±10		
	MAX220, V _{OUT} = ±12V			±25		
Data Rate			200	116	kbps	
Transmitter Output Resistance	V _{CC} = V ₊ = V ₋ = 0V, V _{OUT} = ±2V	300	10M		Ω	
Output Short-Circuit Current	V _{OUT} = 0V	V _{OUT} = 0V	±7	±22	mA	
		MAX220		±60		
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range				±30	V	
	MAX220			±25		
RS-232 Input Threshold Low	V _{CC} = 5V	All except MAX243 R _{2IN}	0.8	1.3	V	
		MAX243 R _{2 IN} (Note 4)	-3			
RS-232 Input Threshold High	V _{CC} = 5V	All except MAX243 R _{2IN}		1.8	2.4	V
		MAX243 R _{2 IN} (Note 4)		-0.5	-0.1	

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1μF, MAX220, C1 = 0.047μF, C2–C4 = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, V _{CC} = 5V, no hysteresis in SHDN		0.2	0.5	1.0	V
	MAX220		0.3			
	MAX243		1			
RS-232 Input Resistance	T _A = +25°C (MAX220)		3	5	7	kΩ
			3	5	7	
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA		0.2		0.4	V
	I _{OUT} = 1.6mA (MAX220)		0.4			
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Sinking V _{OUT} = V _{CC}		10	30		
TTL/CMOS Output Leakage Current	SHDN = V _{CC} or EN = V _{CC} (SHDN = 0V for MAX222), 0V ≤ V _{OUT} ≤ V _{CC}		±0.05		±10	μA
EN Input Threshold Low	MAX242		1.4		0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V _{CC} Supply Current (SHDN = V _{CC}), figures 5, 6, 11, 19	No load	MAX220	0.5		2	μA
		MAX222/MAX232A/MAX233A/MAX242/MAX243	4		10	
	3kΩ load both inputs	MAX220	12			
		MAX222/MAX232A/MAX233A/MAX242/MAX243	15			
Shutdown Supply Current	MAX222/MAX242	T _A = +25°C	0.1		10	μA
		T _A = 0°C to +70°C	2		50	
		T _A = -40°C to +85°C	2		50	
		T _A = -55°C to +125°C	35		100	
SHDN Input Leakage Current	MAX222/MAX242				±1	μA
SHDN Threshold Low	MAX222/MAX242		1.4		0.8	V
SHDN Threshold High	MAX222/MAX242		2.0	1.4		V
Transition Slew Rate	C _L = 50pF to 2500pF, R _L = 3kΩ to 7kΩ, V _{CC} = 5V, T _A = +25°C, measured from +3V to -3V or -3V	MAX222/MAX232A/MAX233/MAX242/MAX243	6	12	30	V/μs
		MAX220	1.5	3	30.0	
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	t _{PHLT}	MAX222/MAX232A/MAX233/MAX242/MAX243	1.3		3.5	μs
		MAX220	4		10	
	t _{PLHT}	MAX222/MAX232A/MAX233/MAX242/MAX243	1.5		3.5	
		MAX220	5		10	

Note 4: MAX243 R_{2OUT} is guaranteed to be low when R_{2IN} is ≥ 0V or is floating.

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

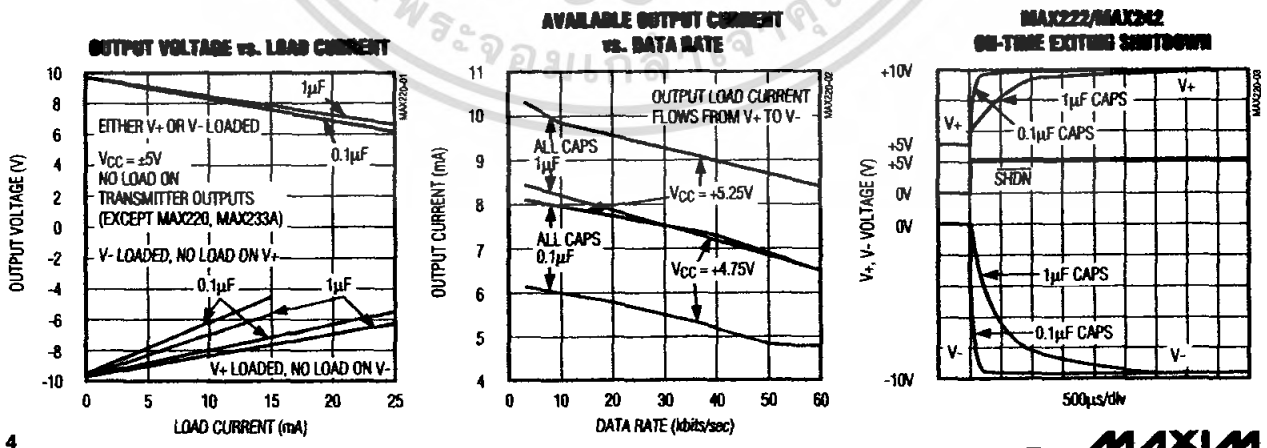
ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1-C4 = 0.1μF, MAX220, C1 = 0.047μF, C2-C4 = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Receiver Propagation Delay RS-232 to TLL (Normal Operation), Figure 2	t _{PHLR}	MAX222/MAX232A/MAX233/ MAX242/MAX243	0.5	1		μs
		MAX220	0.6	3		
	t _{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243	0.6	1		
		MAX220	0.8	3		
Receiver Propagation Delay RS-232 to TLL (Shutdown), Figure 2	t _{PHLS}	MAX242	0.5	10		μs
	t _{PHLS}	MAX242	2.5	10		
Receiver-Output Enable Time, Figure 3	t _{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t _{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time (SHDN Goes High), Figure 4	t _{ET}	MAX222/MAX242, 0.1μF caps (includes charge-pump start-up)		250		μs
Transmitter-Output Disable Time (SHDN Goes Low), Figure 4	t _{DT}	MAX222/MAX242, 0.1μF caps		600		ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t _{PHLT} - t _{PLHT}	MAX222/MAX232A/MAX233/ MAX242/MAX243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (Normal Operation)	t _{PHLR} - t _{PLHR}	MAX222/MAX232A/MAX233/ MAX242/MAX243		100		ns
		MAX220		225		

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

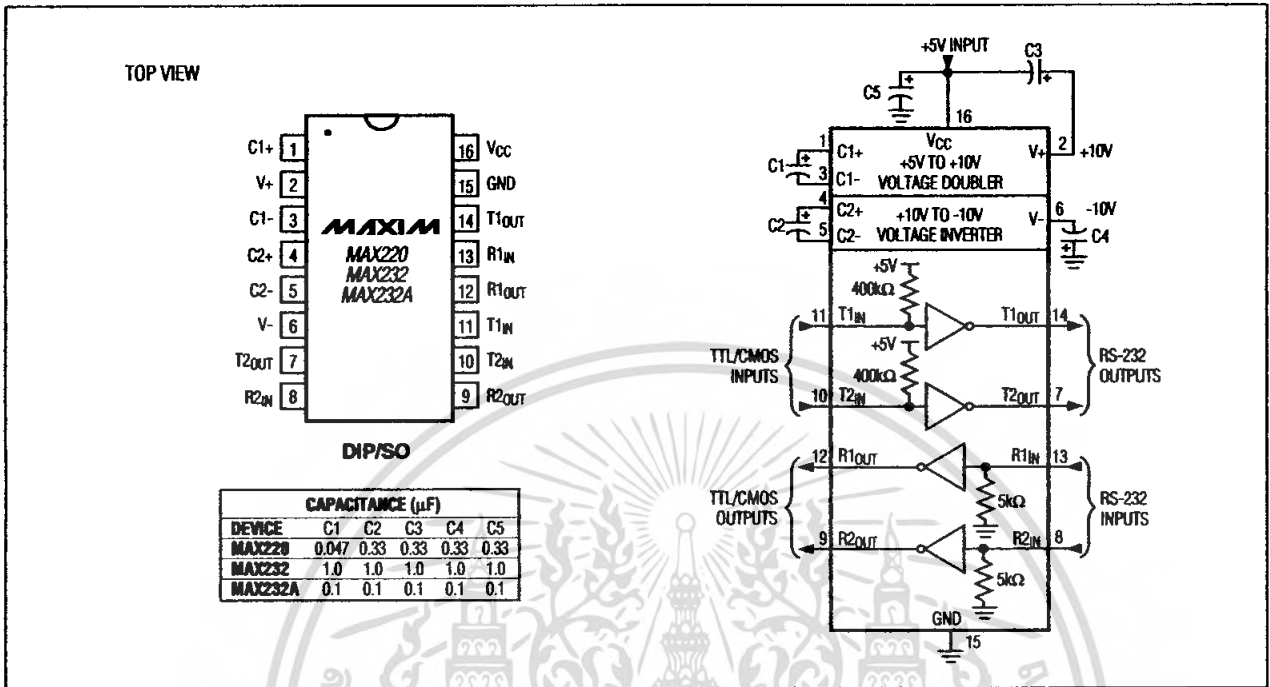


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

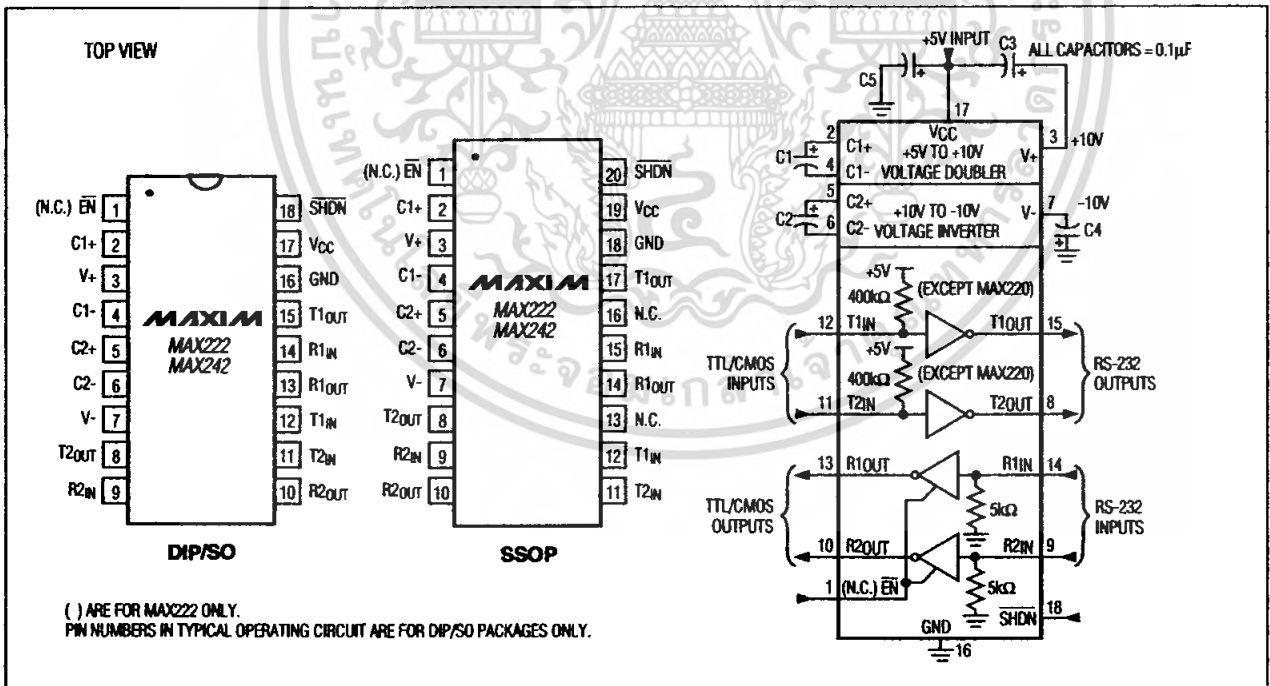


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC817 Series

High Density Mounting Type Photocoupler

- Lead forming type (I type) and taping reel type (P type) are also available. (PC817/PC817P)
- TÜV (VDE0884) approved type is also available as an option.

■ Features

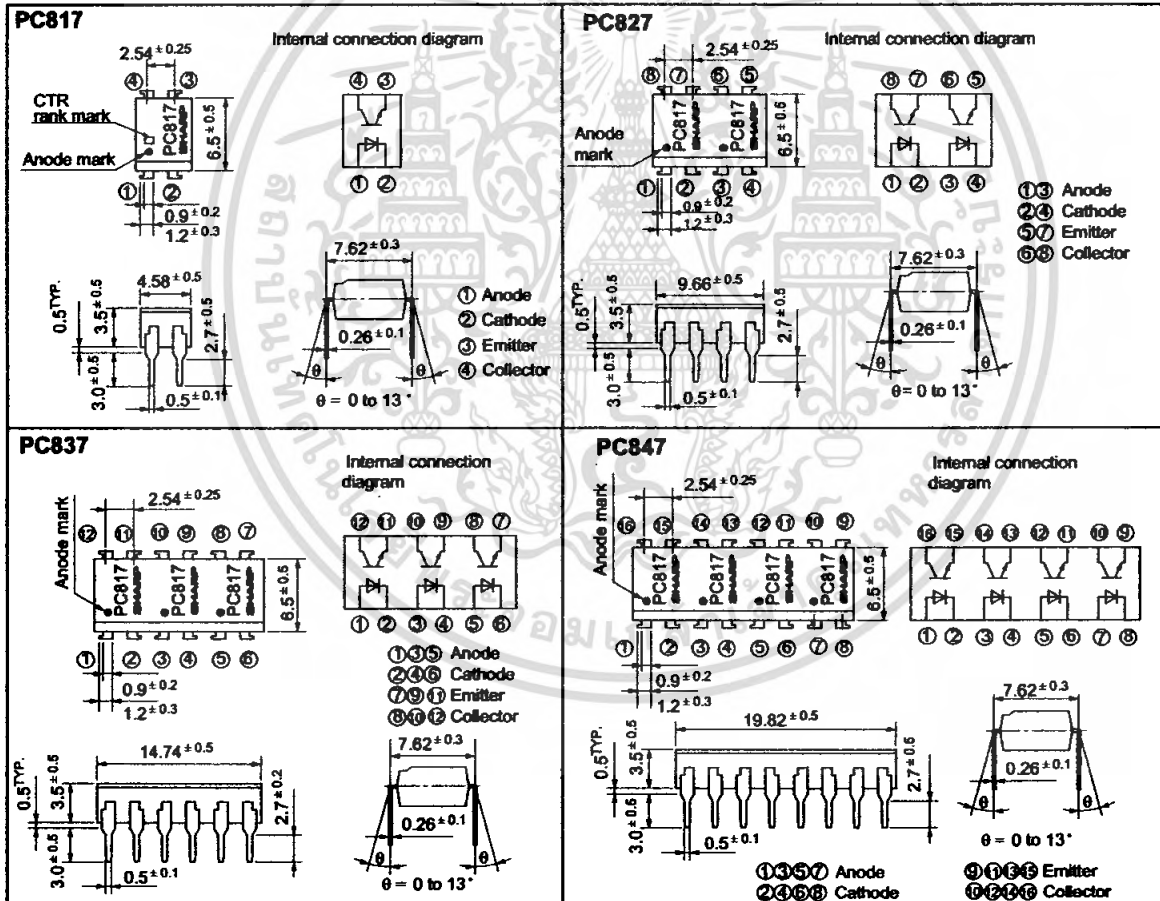
1. Current transfer ratio
(CTR: MIN. 50% at $I_F = 5\text{mA}$, $V_{CE} = 5\text{V}$)
2. High isolation voltage between input and output ($V_{iso} : 5000\text{V}_{rms}$)
3. Compact dual-in-line package
PC817 : 1-channel type
PC827 : 2-channel type
PC837 : 3-channel type
PC847 : 4-channel type
4. Recognized by UL, file No. E64380

■ Applications

1. Computer terminals
2. System appliances, measuring instruments
3. Registers, copiers, automatic vending machines
4. Electric home appliances, such as fan heaters, etc.
5. Signal transmission between circuits of different potentials and impedances

■ Outline Dimensions

(Unit : mm)



* In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that occur in equipment using any of SHARP's devices, shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest version of the device specification sheets before using any SHARP's device.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ Absolute Maximum Ratings

($T_a = 25^\circ\text{C}$)

	Parameter	Symbol	Rating	Unit
Input	Forward current	I_F	50	mA
	*1Peak forward current	I_{FM}	1	A
	Reverse voltage	V_R	6	V
	Power dissipation	P	70	mW
Output	Collector-emitter voltage	V_{CE0}	35	V
	Emitter-collector voltage	V_{ECO}	6	V
	Collector current	I_C	50	mA
	Collector power dissipation	P_C	150	mW
	Total power dissipation	P_{tot}	200	mW
	*2Isolation voltage	V_{iso}	5 000	V _{rms}
	Operating temperature	T_{opr}	- 30 to + 100	$^\circ\text{C}$
	Storage temperature	T_{stg}	- 55 to + 125	$^\circ\text{C}$
	*3Soldering temperature	T_{sol}	260	$^\circ\text{C}$

*1 Pulse width $\leq 100\mu\text{s}$, Duty ratio : 0.001

*2 40 to 60% RH, AC for 1 minute

*3 For 10 seconds

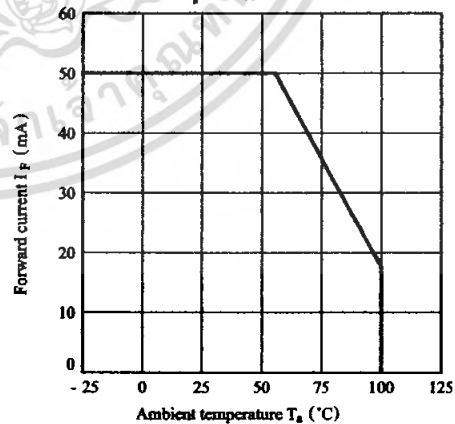
■ Electro-optical Characteristics

($T_a = 25^\circ\text{C}$)

	Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input	Forward voltage	V_F	$I_F = 20\text{mA}$	-	1.2	1.4	V
	Peak forward voltage	V_{FM}	$I_{FM} = 0.5\text{A}$	-	-	3.0	V
	Reverse current	I_R	$V_R = 4\text{V}$	-	-	10	μA
	Terminal capacitance	C_t	$V = 0, f = 1\text{kHz}$	-	30	250	pF
Output	Collector dark current	I_{CBO}	$V_{CE} = 20\text{V}$	-	-	10^{-7}	A
Transfer characteristics	*4Current transfer ratio	CTR	$I_F = 5\text{mA}, V_{CE} = 5\text{V}$	50	-	600	%
	Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_F = 20\text{mA}, I_C = 1\text{mA}$	-	0.1	0.2	V
	Isolation resistance	R_{iso}	DC500V, 40 to 60% RH	5×10^{10}	10^{11}	-	Ω
	Floating capacitance	C_f	$V = 0, f = 1\text{MHz}$	-	0.6	1.0	pF
	Cut-off frequency	f_c	$V_{CE} = 5\text{V}, I_C = 2\text{mA}, R_L = 100\Omega, -3\text{dB}$	-	80	-	kHz
	Response time	Rise time	t_r	$V_{CE} = 2\text{V}, I_C = 2\text{mA}, R_L = 100\Omega$	-	4	18
Fall time		t_f	-		3	18	μs

*4 Classification table of current transfer ratio is shown below.

Fig. 1 Forward Current vs. Ambient Temperature



Model No.	Rank mark	CTR (%)
PC817A	A	80 to 160
PC817B	B	130 to 260
PC817C	C	200 to 400
PC817D	D	300 to 600
PC8●7AB	A or B	80 to 260
PC8●7BC	B or C	130 to 400
PC8●7CD	C or D	200 to 600
PC8●7AC	A, B or C	80 to 400
PC8●7BD	B, C or D	130 to 600
PC8●7AD	A, B, C or D	80 to 600
PC8●7	A, B, C, D or No mark	50 to 600

● : 1 or 2 or 3 or 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fig. 2 Collector Power Dissipation vs. Ambient Temperature

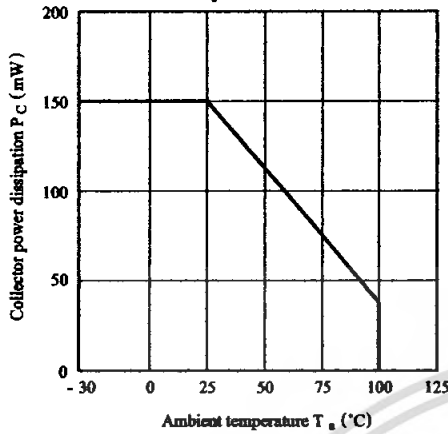


Fig. 3 Peak Forward Current vs. Duty Ratio

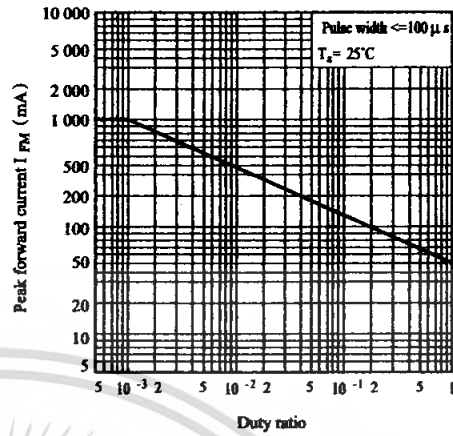


Fig. 4 Current Transfer Ratio vs. Forward Current

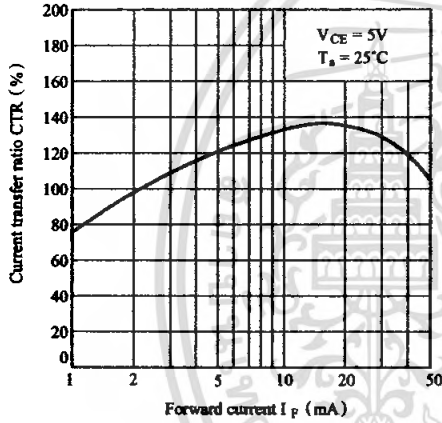


Fig. 5 Forward Current vs. Forward Voltage

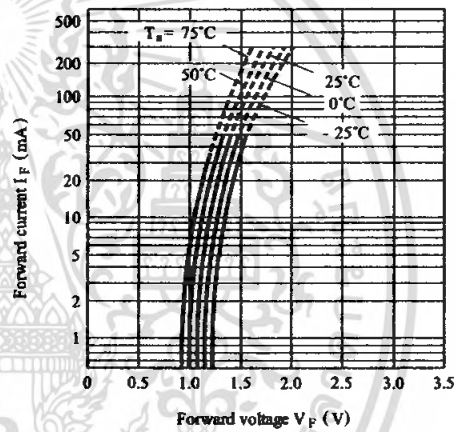


Fig. 6 Collector Current vs. Collector-emitter Voltage

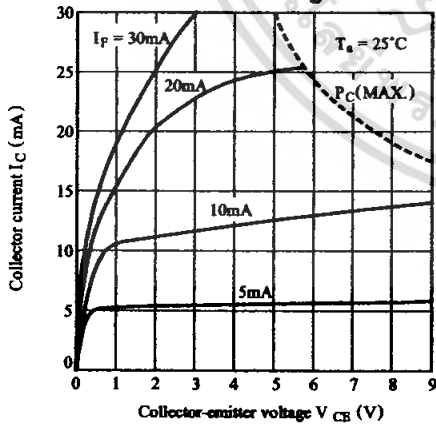


Fig. 7 Relative Current Transfer Ratio vs. Ambient Temperature

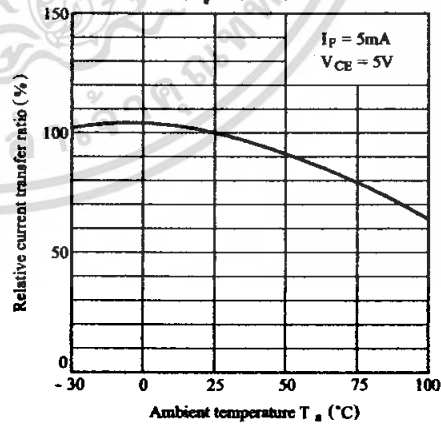


Fig. 8 Collector-emitter Saturation Voltage vs. Ambient Temperature

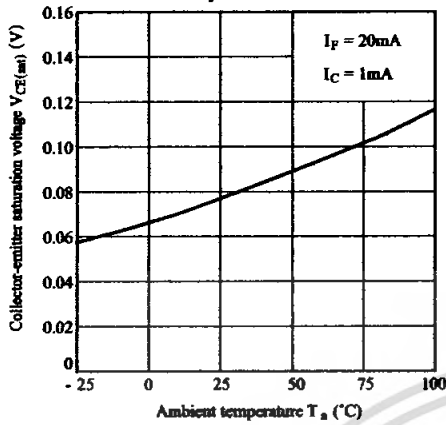


Fig. 9 Collector Dark Current vs. Ambient Temperature

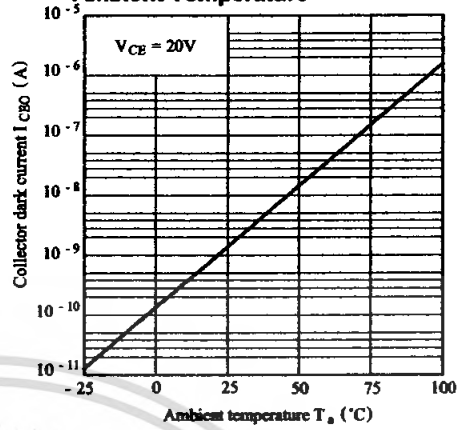


Fig.10 Response Time vs. Load Resistance

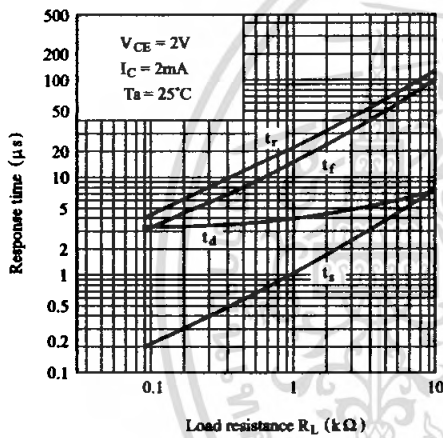
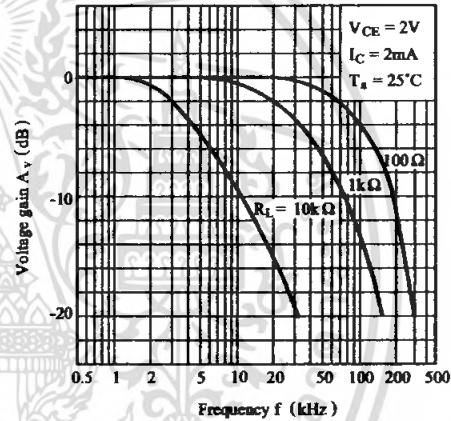
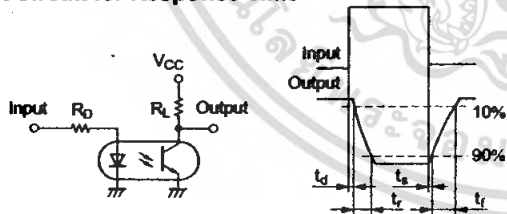


Fig.11 Frequency Response



Test Circuit for Response Time



Test Circuit for Frequency Response

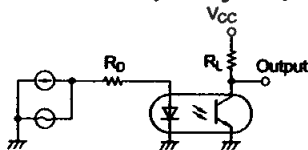
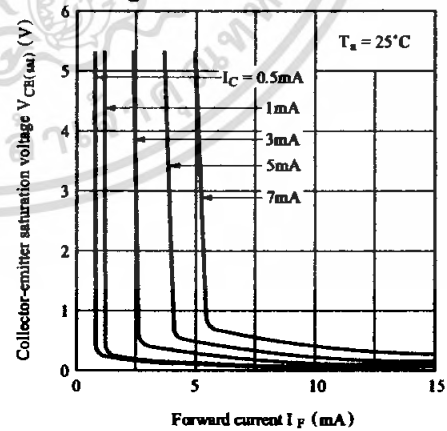


Fig.12 Collector-emitter Saturation Voltage vs. Forward Current



● Please refer to the chapter "Precautions for Use"

NOTICE

- The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.
- Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.
- Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:
 - (i) The devices in this publication are designed for use in general electronic equipment designs such as:
 - Personal computers
 - Office automation equipment
 - Telecommunication equipment [terminal]
 - Test and measurement equipment
 - Industrial control
 - Audio visual equipment
 - Consumer electronics
 - (ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection with equipment that requires higher reliability such as:
 - Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
 - Traffic signals
 - Gas leakage sensor breakers
 - Alarm equipment
 - Various safety devices, etc.
 - (iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:
 - Space applications
 - Telecommunication equipment [trunk lines]
 - Nuclear power control equipment
 - Medical and other life support equipment (e.g., scuba).
- Contact a SHARP representative in advance when intending to use SHARP devices for any "specific" applications other than those recommended by SHARP or when it is unclear which category mentioned above controls the intended use.
- If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Control Law of Japan, it is necessary to obtain approval to export such SHARP devices.
- This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.
- Contact and consult with a SHARP representative if there are any questions about the contents of this publication.