

หุ่นยนต์ควบคุมด้วยระบบฟัซซี่ลอจิก

Robot Controlled by fuzzy logic system



นาย นิพัฒน์ ชัยวรมุขกุล รหัส 41014224 ชั้นปีที่ 4C

นาย รัชฎ อรุณพันธุ์ รหัส 41014346 ชั้นปีที่ 4C

โดย
อาจารย์ที่ปรึกษา
รศ.ดร. กิตติ ไพฑูรย์วัฒนกิจ

เลขหมู่.....
เลขทะเบียน... 46288
วัน, เดือน, ปี 2 1 ส.ค. 2546

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

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประจำภาคเรียนที่ 2 ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ปีการศึกษา 2544

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ควบคุมด้วยระบบพีซีอัตโนมัติ

ผู้จัดทำ

- | | | | |
|----------------|-------------|------|----------|
| 1) นาย นิพัฒน์ | ชัยวรมุขกุล | รหัส | 41014224 |
| 2) นาย รัชฎ | อรุณพันธุ์ | รหัส | 41014346 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการเรื่อง (ภาษาไทย) หุ่นยนต์ควบคุมด้วยระบบฟัซซี่ลอจิก
(ภาษาอังกฤษ) Robot Controlled by fuzzy logic system

จัดทำโดย

1) นาย นิพัทธ์ ชัยวรมุขกุล รหัส 41014224

2) นาย รัชฎ อรุณพันธุ์ รหัส 41014346

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ควบคุมด้วยระบบฟัซซี่ลอจิก

นาย นิพนธ์ ชัยวรมงคล 41014224
นาย รัชฎ อรุณพันธุ์ 41014346
รศ. ดร. กิตติ ไพฑูรย์วัฒนกิจ
อาจารย์ที่ปรึกษา
ปีการศึกษา 2544 เทอม 2

บทคัดย่อ

ความพยายามที่จะสร้างหุ่นยนต์ให้มีลักษณะใกล้เคียงมนุษย์จริง นับเป็นสิ่งที่นักประดิษฐ์หลายๆคนต่างให้ความสนใจและพยายามค้นคว้ากันมาโดยตลอด โครงการนี้เป็นอีกโครงการหนึ่ง ซึ่งได้จำลองลักษณะการตอบสนองต่อสิ่งเร้าของมนุษย์ ลงในการควบคุมทิศทางกล้องด้วยระบบFuzzy Logic เริ่มต้นด้วยการใช้ไมโครโฟนรับค่า input ซึ่งเป็นเสียงที่เกิดขึ้นรอบๆตัวกล้องก่อน หลังจากนั้นจะใช้การประมาณค่าแบบ Fuzzy Logic ในการคำนวณหาตำแหน่งกำเนิดของเสียงที่เกิดขึ้น ได้ค่า Output เป็นมุมของต้นกำเนิดเสียงเทียบกับตำแหน่งปัจจุบันของกล้อง แล้วจึงทำการควบคุมมอเตอร์ ให้หมุนกล้องเพื่อจับภาพของต้นกำเนิดเสียงนั้นเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Robot Controlled by fuzzy logic system

Nipat Chaiworamukkul 41014224

Rachot Arunphan 41014346

Assoc. Prof. Dr. Kitti Pitoonwattanakij

Advisor 2001/2

Abstract

An attempt to invent the robot with human-like functions has attracted scientists' interest for a long time as we can see from the number of researches on this topic from worldwide. This project is one of those which aims to simulate the response of human to environmental excitement into the control of camera direction by using Fuzzy Logic system. The simulation starts from acquiring the input from near area via microphone and then the input will be estimated by using Fuzzy Logic to calculate the origin of the signal source. The output of the process will be shown as the value of the angle compared with current position of the camera. Then the motor will finally be controlled to pan the camera to the signal source.

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
2.1 เสียง (Sound)	3
2.1.1 แหล่งกำเนิดของการสั่นสะเทือน	3
2.1.2 ลักษณะสมบัติของคลื่นเสียง	4
2.1.3 เสียงเดินทางในอากาศได้อย่างไร	5
2.1.4 เฟส (phase)	5
2.1.5 เดซิเบล (Decibel)	7
2.2 ไมโครโฟน (Microphone)	9
ไมโครโฟนแบบตัวเก็บประจุ	9
2.3 ไมโครคอนโทรลเลอร์ (Microcontroller) 8051	10
2.3.1 รายละเอียดของขาสัญญาณของ 8051	10
2.3.2 ชุดคำสั่งของ 8051	11
2.4 พอร์ตอนุกรม (serial port)	14
2.4.1 การสื่อสารข้อมูลแบบอซิงโครนัส	14
2.4.2 คอนเน็กเตอร์ พอร์ต RS-232 และการเชื่อมต่อ	14
2.4.3 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232	15
2.4.4 การเชื่อมต่อ computer กับ อุปกรณ์ภายนอก	16
2.4.5 การเขียนโปรแกรม Visual Basic เพื่อใช้งานพอร์ตอนุกรม	17
บทที่ 3 การออกแบบและการคำนวณ	22
3.1 ลักษณะเฉพาะทางโครงสร้าง	24
3.2 อุปกรณ์ที่ใช้	25
3.3 กฎการออกแบบ	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	3.3.1	การหาระยะห่างของไมโครโฟนและค่า Delay ที่วัดได้	26
	3.3.2	ค่า Delay ที่สามารถวัดได้	27
	3.3.3	การตรวจวัดมุมด้วย ADC และ การหมุนของ Motor	28
บทที่	4	วงจรควบคุมการทำงาน	29
	4.1	วงจรแหล่งจ่ายไฟตรง (DC Power Supply Circuit)	29
	4.2	วงจรรับเสียง (Sound Detect Circuit)	30
	4.3	วงจรเชื่อมต่อคอมพิวเตอร์ผ่านพอร์ตอนุกรม (Serial Port Interface Circuit)	31
	4.4	วงจรขับเคลื่อนมอเตอร์กระแสตรง (DC Motor Drive Circuits)	33
	4.5	วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (Analog to Digital Converter Circuit (ADC))	35
	4.6	วงจรไมโครคอนโทรลเลอร์และวงจรรวม (Microcontroller and All Circuit)	37
บทที่	5	โปรแกรม MCS-51	39
	5.1	หลักการงานพื้นฐาน (Basic Concept)	39
	5.2	โปรแกรมหลัก (Main Program)	40
	5.3	Manual Mode	41
	5.3.1	โปรแกรมย่อยคำนวณทิศทางที่ใกล้สุดพร้อมสั่งหยุดเมื่อเกิดข้อผิดพลาด	42
	5.3.2	โปรแกรมย่อยเรียกค่าจาก ADC (ADC Loading Program)	44
	5.4	Auto Mode	46
		โปรแกรมย่อยวัดค่า Delay (Delay Counter Program)	47
	5.5	โปรแกรมของ MCS-51 (Source Code MCS-51)	49
บทที่	6	โปรแกรมควบคุมการทำงาน	55
	6.1	ส่วนโปรแกรมติดต่อผู้ใช้	55
	6.1.1	หน้าจอโปรแกรมการใช้งาน Auto Mode	55
	6.1.2	หน้าจอโปรแกรมการใช้งาน Manual Mode	57
	6.1.3	วิธีการใช้งานโปรแกรมติดต่อกับผู้ใช้ Auto Mode	57
	6.1.4	วิธีการใช้งานโปรแกรมติดต่อกับผู้ใช้ Manual Mode	59
	6.1.5	การเริ่มโปรแกรมใหม่, การปิดโปรแกรม และ การดูข้อมูลของโปรแกรม	60
	6.2	ส่วนของการติดต่อกับหุ่นยนต์ผ่าน พอร์ตอนุกรม	60
	6.3	Flowchart การทำงาน	61
บทที่	7	ฟัซซี่ลอจิก	65
	7.1	ฟัซซี่ลอจิก (Fuzzy logic)	65
	7.2	การควบคุมแบบฟัซซี่ลอจิก	66
	7.3	ประวัติและพัฒนาการของฟัซซี่	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 การออกแบบเพื่อใช้ตรรกแบบฟัซซี่	67
7.5 รูปแบบของฟัซซี่ลอจิกในทางคณิตศาสตร์	68
7.5.1 Fuzzy Set and Membership Functions	68
7.5.2 Universe of discourse (U)	68
7.5.3 Linguistic variables and Linguistic values	70
7.5.4 Definition	71
7.5.5 Membership Function Formulation	73
7.6 การประยุกต์ใช้งานฟัซซี่ลอจิก (Fuzzy Logic Application)	76
7.6.1 วิเคราะห์ระบบ เพื่อกำหนด Input และ Output ของระบบ	76
7.6.2 Fuzzification	77
7.6.3 Fuzzy Rules	78
7.6.4 Defuzzification	78
บทที่ 8 ผลการทดลอง	80
ตอนที่ 1 การทดสอบการหมุนกลิ้ง	80
ตอนที่ 2 การหาค่าความรู้พื้นฐาน	83
บทที่ 9 สรุปและวิจารณ์	85
บรรณานุกรม	87
กิตติกรรมประกาศ	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
รูปที่ 2.1 แสดงลักษณะคลื่นตามยาวและคลื่นตามขวาง	3
รูปที่ 2.2 แสดงรูปคลื่นเสียงมีลักษณะเป็นรูปคลื่น sine คาบแทนด้วย t ความถี่ $1/t$	4
รูปที่ 2.3 แสดงรูปผลของเฟส	6
รูปที่ 2.4 แสดงความสูงของจุดที่เปลี่ยนตามค่ามุม \sin มุมของเฟสคือจำนวนองศาของการหมุน	7
รูปที่ 2.5 แสดง โครงสร้างของไมโครโฟนแบบตัวเก็บประจุ	9
รูปที่ 2.6 แสดงขาของ 89C51	12
รูปที่ 2.7 คอนเน็คเตอร์อนุกรม 9 เส้น แบบ DB-9	14
รูปที่ 2.8 การต่ออุปกรณ์ภายนอก กับ พอร์ตอนุกรมภายนอกในลักษณะต่างๆ	16
รูปที่ 2.9 แสดงรูปขั้นตอนแรกการเลือกใช้งาน MSComm Control	20
รูปที่ 2.10 แสดงรูปขั้นตอนที่2การเลือกใช้งาน MSComm Control	21
รูปที่ 2.11 แสดงรูปขั้นตอนที่3การเลือกใช้งาน MSComm Control	21
บทที่ 3 การออกแบบและการคำนวณ	22
รูปที่ 3.1 แสดงลักษณะตัวหูนุ่ ภายนอก	22
รูปที่ 3.2 แสดงลักษณะตัวหูนุ่ ภายนอก (ด้านหน้า)	23
รูปที่ 3.3 แสดงลักษณะตัวหูนุ่ ภายนอก (ด้านบน)	23
รูปที่ 3.4 แสดงลักษณะตัวหูนุ่ ภายใน	24
รูปที่ 3.5 แสดงรูปร่างของไมโครโฟนที่ใช้	24
รูปที่ 3.6 แสดงลักษณะ โครงสร้างภายนอกของกล่อง LG LPC-UC35	25
บทที่ 4 วงจรควบคุมการทำงาน	29
รูปที่ 4.1 แสดงวงจรแหล่งจ่ายไฟฟ้ากระแสตรง	29
รูปที่ 4.2 แสดงวงจรรับเสียงจากภายนอก	30
รูปที่ 4.3 แสดงขาและองค์ประกอบของ ICL232	31
รูปที่ 4.4 แสดงวงจรเชื่อมต่อกอมพิวเตอร์ผ่านพอร์ตอนุกรม	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	รูปที่ 4.5 การจัดเรียงขาของ L293	33
	รูปที่ 4.6 วงจร Drive Motor โดยใช้ IC L293	34
	รูปที่ 4.7 แสดงขาของ ADC0804	35
	รูปที่ 4.8 วงจรของ ADC0804	36
	รูปที่ 4.9 วงจร Microcontroller MCS-51	37
บทที่	5 โปรแกรม MCS-51	39
	รูปที่ 5.1 โหมคพื้นฐานในการทำงาน	39
	รูปที่ 5.2 Flow Chart ของ Main Program	41
	รูปที่ 5.3 การตัดสินใจเลือกทิศทางการหมุน	42
	รูปที่ 5.4 Flow Chart ของ โปรแกรมย่อยคำนวณทิศทางที่ใกล้สุดพร้อมสั่งหยุดเมื่อเกิดข้อผิดพลาด	43
	รูปที่ 5.5 Timing Diagrams ของ ADC0804	44
	รูปที่ 5.6 Flow Chart ของ โปรแกรมย่อยเรียกค่าจาก ADC	45
	รูปที่ 5.7 Auto Mode Block Diagram	47
	รูปที่ 5.8 Flow Chart ของ โปรแกรมย่อยวัดค่า Delay	48
บทที่	6 โปรแกรมควบคุมการทำงาน	55
	รูปที่ 6.1 แสดงส่วนประกอบของ โปรแกรม Auto Mode	55
	รูปที่ 6.2 แสดงส่วนประกอบของ โปรแกรม Fuzzy Setup	56
	รูปที่ 6.3 แสดงส่วนประกอบของ โปรแกรม Fuzzy Graph	56
	รูปที่ 6.4 แสดงส่วนประกอบของ โปรแกรม Manual Mode	57
	รูปที่ 6.5 Flow Chart ของ โปรแกรม Auto Mode	61
	รูปที่ 6.6 Flow Chart ของการคำนวณค่า Input	61
	รูปที่ 6.7 Flow Chart ของ โปรแกรม Manual Mode	63
บทที่	7 ฟัซซี่ลอจิก	65
	รูปที่ 7.1 Fuzzy Sets ของค่าไม้ค้อนเนื่องที่มีการเรียงลำดับ	69
	รูปที่ 7.2 Fuzzy Sets ของค่าที่ต่อเนื่อง	69
	รูปที่ 7.3 Fuzzy Sets ต่างๆ แบ่งตาม Linguistic Values ของอายุคน	70
	รูปที่ 7.4 แสดง Core, Support และ Cross Overpoints ของ Fuzzy Set หนึ่ง	71
	รูปที่ 7.5 รูป Fuzzy Set ($A \subseteq B$)	72
	รูปที่ 7.6 รูป Fuzzy Set ($A \text{ OR } B$)	72
	รูปที่ 7.7 รูป Fuzzy Set ($A \text{ AND } B$)	73
	รูปที่ 7.8 รูป Fuzzy Set (Not A)	73
	รูปที่ 7.9 Triangular Membership Function	74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.10 Trapezoidal Membership Function	74
รูปที่ 7.11 รูปแบบการทำงานพื้นฐานในการควบคุมแบบ Fuzzy	76
รูปที่ 7.12 แสดงลักษณะการแบ่งค่า Delay เอามาใช้งาน	77
รูปที่ 7.13 กราฟอัตราความเป็นสมาชิกของ Input	77
รูปที่ 7.14 กราฟอัตราความเป็นสมาชิกของ Output	78
บทที่ 8 การทดลอง	80
บทที่ 9 สรุปและวิจารณ์	85
บรรณานุกรม	87
กิตติกรรมประกาศ	88



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
ตารางที่ 2.1 แสดงระดับความดังของเสียงจากแหล่งกำเนิดเสียงต่างๆ	8
ตารางที่ 2.2 แสดงหน้าที่ของขา port 3 ของ 89C51	11
ตารางที่ 2.3 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรม RS-232 แบบ DB – 9	15
บทที่ 3 การออกแบบและการคำนวณ	22
ตารางที่ 3.1 แสดงคุณสมบัติที่สำคัญของกล้อง	25
บทที่ 4 วงจรควบคุมการทำงาน	29
บทที่ 5 โปรแกรม MCS-51	39
บทที่ 6 โปรแกรมควบคุมการทำงาน	55
บทที่ 7 พีซีสตอจิก	65
บทที่ 8 การทดลอง	80
ตารางที่ 8.1 แสดงผลการทดสอบการหมุนของกล้อง	80
ตารางที่ 8.2 แสดงผลการทดสอบการหาค่าความรู้พื้นฐาน	83
บทที่ 9 สรุปและวิจารณ์	85
บรรณานุกรม	87
กิตติกรรมประกาศ	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

มนุษย์ นับเป็นสิ่งประเสริฐจากธรรมชาติที่ยอดเยี่ยมที่สุด นับตั้งแต่โลกใบนี้ได้ถือกำเนิดขึ้น มนุษย์สามารถ คิด ตัดสินใจ เพื่อเอาชีวิตรอดจากอันตรายต่างๆ ได้ มนุษย์ มีความรัก มีความเกลียด ความกลัว ความเขลา และความโลภ มนุษย์ เป็นทั้งผู้สร้าง และเป็นทั้งผู้ทำลายล้าง ไม่มีสิ่งใดจะเทียบเคียงมนุษย์ได้ นอกจากมนุษย์ด้วยกันเอง

มนุษย์ได้พยายามที่จะสร้างสิ่งทีเลียนแบบตนเองมาโดยตลอด โดยให้คำนิยามถึงนั้นว่า “หุ่นยนต์” หุ่นยนต์ตัวแรก ใครเป็นผู้คิด ใครเป็นผู้สร้าง คงไม่อาจที่จะระบุชัดเจนได้ แต่ในปัจจุบัน หุ่นยนต์กำลังจะกลายเป็น “สัญลักษณ์” ของโลกอนาคต

นับตั้งแต่บริษัท Sony ได้ให้กำเนิด “Aibo” หุ่นยนต์สุนัขตัวแรกขึ้นมา ถือเป็นต้นกำเนิดของการสร้าง หุ่นยนต์ที่มีความใกล้เคียงกับสิ่งมีชีวิตมากที่สุดตัวหนึ่ง พร้อมด้วยยอดขายที่เหลือเชื่อ ต่างจุดประกายให้บริษัทต่างๆ หันมาสนใจที่จะสร้างหุ่นยนต์ให้มีลักษณะใกล้เคียงสิ่งมีชีวิตกันมากขึ้น โดยมีเป้าหมายที่จะกระจายตลาดหุ่นยนต์ ซึ่งโดยทั่วไปจะเน้นหนักในงานทางด้านอุตสาหกรรม ให้กลายเป็น “เครื่องใช้ในครัวเรือน” มากยิ่งขึ้น

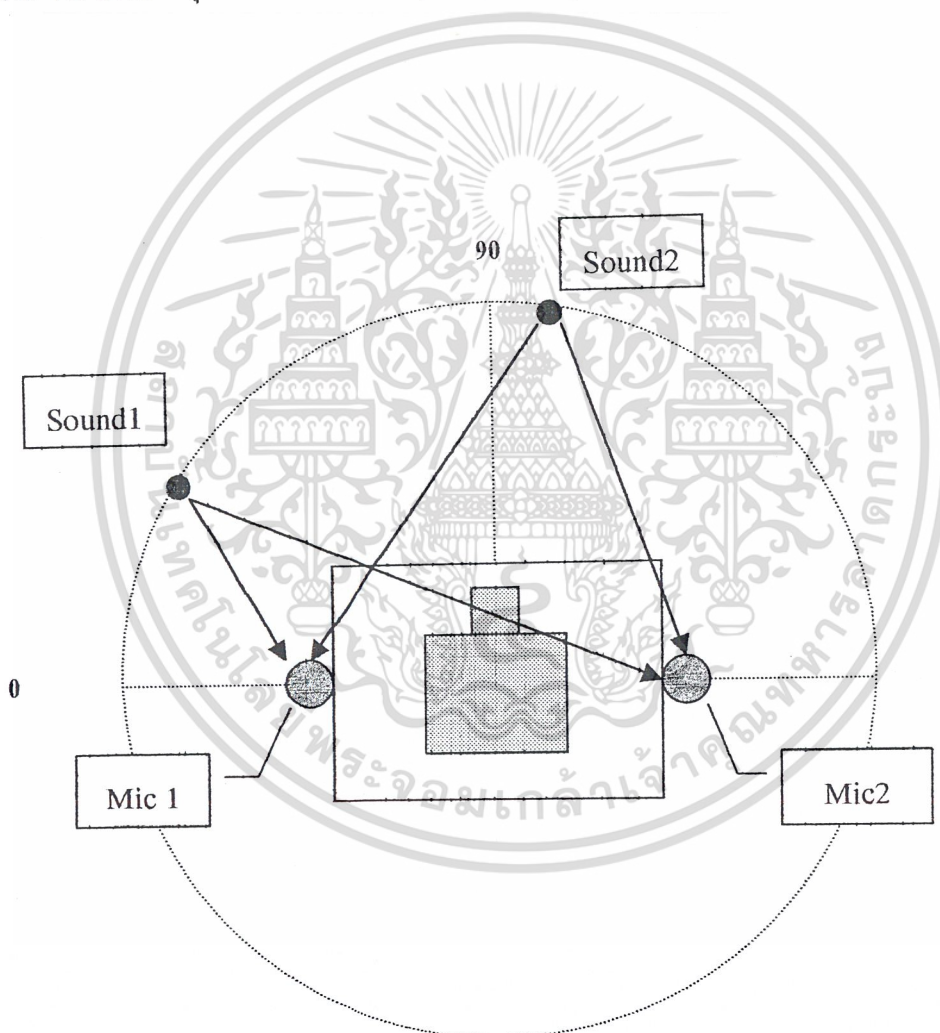
บริษัท Honda เป็นอีกหนึ่งบริษัทที่ประสบความสำเร็จในการสร้าง “มนุษย์จำลอง” หรือ “P3” หุ่นยนต์เดินได้ด้วยสองขาเยี่ยงมนุษย์ตัวแรกของโลก ซึ่งสามารถทรงตัวได้ในทุกอิริยาบถ ไม่ว่าจะเดินก้าวขึ้นบันได เตะฟุตบอล หรือแม้กระทั่งถูกผลัก ก็ยังสามารถทรงตัวได้ด้วยสองขาเช่นเดียวกับมนุษย์ นี่จึงเป็นอีกหนึ่งสิ่งที่ยืนยันได้ว่า หุ่นยนต์กำลังจะกลายมาเป็นสิ่งที่ใกล้ตัวเรามากยิ่งขึ้น ไม่เช่นนั้นบริษัทเหล่านี้คงไม่ลงทุนทำงานวิจัยมูลค่ามหาศาลเช่นนี้ออกมาอย่างแน่นอน

แต่การสร้างหุ่นยนต์ในแง่ของ “หุ่นยนต์ที่มีความรู้สึก” ยังไม่มีให้เห็นมากนัก โครงการนี้จึงขอนำเสนออีกหนึ่งแนวคิด ที่จะสร้างหุ่นยนต์ให้มีความใกล้เคียงมนุษย์มากขึ้น โดยได้จำลองลักษณะหนึ่งของมนุษย์ ซึ่งในทางจิตวิทยาเรียกว่า “การตอบสนองต่อสิ่งเร้า” ลงในการควบคุมหุ่นยนต์ด้วยระบบ Fuzzy Logic

ลองนึกภาพว่าเรากำลังนั่งเล่นอยู่ในห้องที่เงียบสงบห้องหนึ่ง หันหน้าออกไปดูทีวีที่สนัที่อยู่นอกหน้าต่าง ส่วนข้างหลังเป็นประตูห้อง แต่ทันใดนั้นเอง ก็มีเสียงเคาะประตูดังขึ้นจากทางด้านหลัง ถามว่าเราจะทำอย่างไร?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิบัติการแรกที่ทุกคนคงจะต้องทำก็คือ “หันหน้าไปที่ประตู เพื่อดูว่ามีอะไรเกิดขึ้น” นี้เองจึงเป็นจุดเริ่มต้นแนวคิดของโครงการนี้ โดยในโครงการนี้จะแทน “ดวงตา” ของมนุษย์ด้วยกล้องดิจิทัล ที่ต่ออยู่กับคอมพิวเตอร์ และแทนไมโครโฟนที่อยู่รอบๆตัวกล้องอย่างสมมาตรด้วย “หู” ของมนุษย์ เมื่อมีเสียงเกิดขึ้น เสียงซึ่งมีลักษณะการกระจายอกรอบข้าง แบบทรงกลม จะทำให้ไมโครโฟนที่วางอยู่ที่จุดต่างๆเกิด ค่า Delay Time ในการรับเสียงเริ่มต้นแตกต่างกัน ซึ่งเราจะนำเอาค่านี้ มาประมวลผลโดยใช้การประมาณค่าแบบ Fuzzy Logic Control ในการตัดสินใจว่า ต้นกำเนิดเสียงมาจากทิศทางใด แล้วจึงทำการหมุนกล้อง เพื่อจับภาพต้นกำเนิดเสียงนั้นไว้ ซึ่งตัวกล้องเองจะต่ออยู่กับมอเตอร์ และทำการป้อนกลับด้วย VR เพื่อควบคุมตำแหน่งของการหมุนให้ถูกต้อง ตามที่ต้องการ



รูปที่ 1.1 Delay Time ที่เกิดขึ้นจากการรับเสียงของไมโครโฟน 2 ตัว ซึ่งไม่พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 เสียง (Sound)

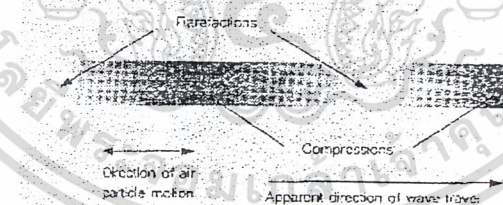
2.1.1 แหล่งกำเนิดของการสั่นสะเทือน

คลื่นเป็นลักษณะอย่างหนึ่งของการถ่ายทอดพลังงานจากตำแหน่งหนึ่ง ไปยังอีกตำแหน่งหนึ่ง อาจอยู่ในรูปพลังงานกล, ความร้อน, แสง, แม่เหล็กไฟฟ้า หรือ เสียง

เสียงเกิดเมื่อวัตถุ (แหล่งกำเนิด) สั่นสะเทือน ทำให้อากาศโดยรอบวัตถุเคลื่อนที่ พิจารณาเป็นรูปทรงกลม เห็นการเคลื่อนที่เข้าออกรอบๆ ทรงกลม มองดูเหมือนการตี squash ลูกบอลจะกระดอนกลับไปกลับเป็นระยะ จนขนาดของทรงกลมเกิดการแกว่ง หรือ oscillate ในลักษณะที่ใหญ่กว่าระดับปกติเล็กน้อย และเล็กกว่าระดับปกติเล็กน้อยสลับกัน ในขณะที่เกิดการเปลี่ยนแปลงจะเกิดการอัดขยายของอากาศรอบๆ ทำให้ลักษณะการอัดและขยายเป็นไปอย่างต่อเนื่องเรื่อยๆ จากทรงกลม ลักษณะเดียวกับการพลิว (ripple) ทำให้เกิดคลื่นขึ้นมานั่นเอง

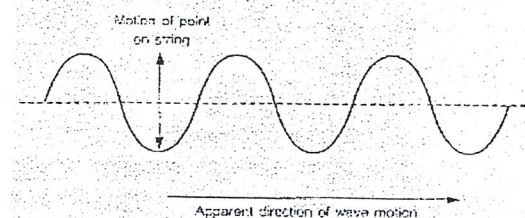
ถ้าหากโยนก้อนหินลงไปในน้ำนิ่งกลางสระน้ำ คลื่นน้ำจะเกิดขึ้น อาศัยน้ำเป็นตัวกลาง คลื่นเคลื่อนที่ไปข้างๆ แผ่กระจายออกไป โดยน้ำกลางสระไม่ได้เคลื่อนที่ไปตามคลื่นเพียงกระเพื่อมขึ้นลง

คลื่นที่มีการสั่นสะเทือนลักษณะทุกๆอนุภาคของตัวกลางไม่วิ่งตามคลื่น เรียกว่า คลื่นตามขวาง (Transverse wave) ส่วนคลื่นที่สลับกับคลื่นตามขวางคือ คลื่นตามยาว อนุภาคตัวกลางเคลื่อนที่ไปมาในทิศทางขนานกับทิศทางคลื่น



๗) คลื่นตามขวางที่เกิดขึ้นจากคลื่นอัดและขยายของอากาศ

รูปที่ 1.1 (๗)

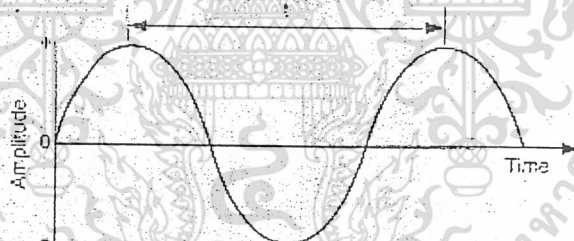


รูปที่ 2.1 แสดงลักษณะคลื่นตามยาวและคลื่นตามขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ลักษณะสมบัติของคลื่นเสียง

ลักษณะการแกว่งของแหล่งกำเนิดทำให้เกิดความถี่ของคลื่นเสียง มีหน่วยเป็นเฮิรตซ์ (Hz) หรือ รอบต่อวินาที (cps) จำนวนการอัดและขยายของอากาศที่มีผลมาจากการเคลื่อนที่ของทรงกลมคือ ส่วนสูง หรือ amplitude ของคลื่นเสียงมีส่วนเกี่ยวข้องกับความดัง (Loudness) ของเสียงที่บ่งบอกถึงความเข้มเสียง ที่มาถึงหูมนุษย์ ระยะทางระหว่างยอดคลื่นที่อยู่ใกล้ชิดกัน หรือระยะทางของ 1 ลูกคลื่น เรียกว่า ความยาวคลื่น (Wavelength) หรือ แลมด้า (λ) อย่างไรก็ตาม ความยาวคลื่นจะขึ้นอยู่กับความเร็วของคลื่น เมื่อคลื่นเดินทางด้วยความเร็ว ระยะทางระหว่างยอดคลื่นจะมีค่ามากกว่าคลื่นที่เดินทางช้าโดยให้เวลาคงที่ระหว่างช่วงอัดของยอดคลื่น ดังรูปที่ 2.8 ลักษณะสมบัติของคลื่นเสียงแทนได้ด้วยกราฟ แกนทางแนวดิ่งเป็นค่า amplitude แกนทางแนวนอนเป็นเวลา แทนช่วงการอัดของอากาศเป็นบวก(+) และช่วงขยายเป็นลบ(-) อนุภาคของตัวกลางจะสั่นสะเทือนด้วยอัตราเดียวกันกับการสั่นสะเทือนของวัตถุต้นกำเนิด เป็นไปอย่างมีจังหวะ ในรูปแบบ simple harmonic หากความถี่ของการสั่นสะเทือนอยู่ในช่วง 20 Hz - 20 kHz โดยประมาณ เป็นเสียงที่มนุษย์สามารถได้ยิน และหากไม่อยู่ในค่านี้อาจฟังไม่ได้ยิน



รูปที่ 2.2 แสดงรูปคลื่นเสียงมีลักษณะเป็นรูปคลื่น sine คาบแทนด้วย t ความถี่ $1/t$.

2.1.3 เสียงเดินทางในอากาศได้อย่างไร

คลื่นเสียงในอากาศเป็นชนิดของคลื่นตามยาวที่โมเลกุลของอากาศเคลื่อนที่รอบตำแหน่งสมดุลกับจุดกึ่งกลางในแนวขนานกับการเคลื่อนที่ของคลื่นเสียง ดังนั้น เมื่อมีเสียงเดินทางไปในตัวกลางที่เป็นอากาศ จะเกิดการอัดตัวและขยายตัวสลับกันไปดังที่กล่าวมาแล้ว ลักษณะคลื่นที่มีการสั่นสะเทือนของตัวกลางเป็นการสั่นสะเทือนที่มีความยืดหยุ่น เรียก Elastic property ทั้งนี้หากตัวกลางไม่มีความยืดหยุ่นแล้ว การสั่นสะเทือนของส่วนใดส่วนหนึ่งของตัวกลางย่อมไม่สามารถทำให้ส่วนที่อยู่ถัดไปของตัวกลางสั่นสะเทือนได้

ในกรณีของคลื่นเสียงในอากาศ หากอากาศมีความหนาแน่นสม่ำเสมอ คลื่นเสียงจะมีอัตราความเร็วคงที่ประมาณ 340 m/sec แต่ความเร็วของเสียงในอากาศที่จุดน้ำเป็นน้ำแข็งความเร็วลดลงเหลือประมาณ 330 m/sec ในเหล็กความเร็วประมาณ 5100 m/sec

ค่าความถี่และความยาวคลื่นของคลื่นเสียง เกี่ยวข้องกัน โดยตรงกับค่าความเร็วเสียง (c) ได้ว่า

$$c = f\lambda \quad \text{หรือ} \quad \lambda = c / f$$

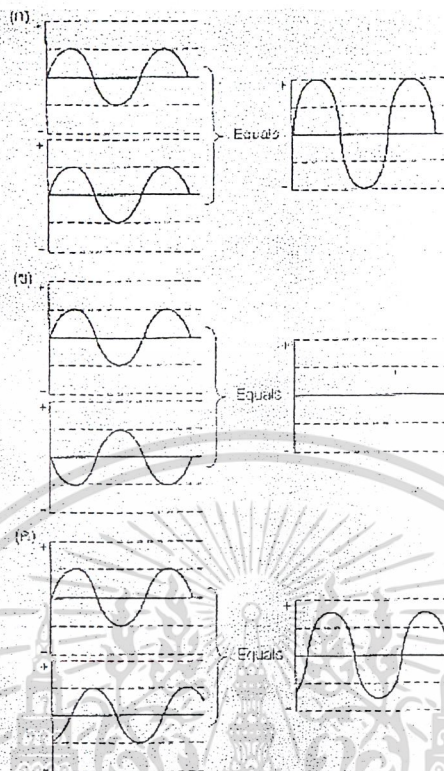
ตัวอย่างเช่น ความยาวคลื่นของเสียงในอากาศที่ 20 Hz อุณหภูมิห้องปกติควรมีค่าเท่าใด?

$$\lambda = 340 / 20 = 17m$$

สำหรับความยาวคลื่นของเสียงในอากาศที่ 20 kHz มีค่า 1.7 cm

2.1.4 เฟส (phase)

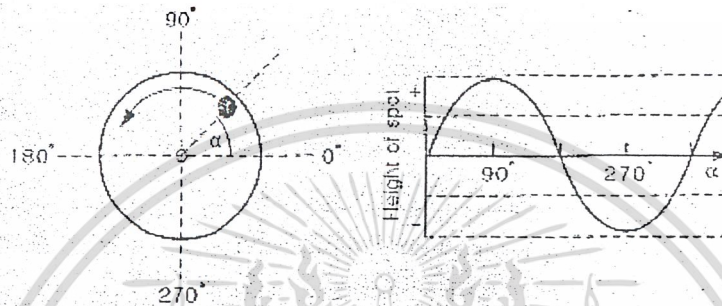
หากคลื่น 2 คลื่นมีความถี่และ amplitude เดียวกัน หรือเกิดขึ้นพร้อมกัน คือเริ่มสั่นจุดเดียวกันและจบลงที่จุดเดียวกันและจบที่จุดเดียวกัน เรียกว่า คลื่นที่มีเฟสเดียวกัน (Inphase) ดังรูปที่ ๑ ถ้าสัญญาณ 2 สัญญาณมีเฟสเดียวกัน ค่า amplitude จะเป็นผลบวก หมายถึงความสูงของ amplitude จะมีค่าเป็น 2 เท่า สำหรับคลื่นที่มีเฟสต่างกัน (Out of phase) เกิดจากคลื่น 2 คลื่นที่มีความถี่และ amplitude เดียวกัน แต่เกิดขึ้นไม่พร้อมกัน กล่าวคือ คลื่นแรกจะเริ่มต้นทางครึ่งรอบบวก (Positive half-cycle) คลื่นหลังจะเริ่มต้นทางครึ่งรอบลบ (Negative half-cycle) ในลักษณะที่ตรงกันข้ามกัน ผลรวมที่ได้จะเป็น 0 คือไม่มีสัญญาณนั่นเอง อีกกรณีหากสัญญาณ 2 สัญญาณมีขนาดเท่ากัน แต่เฟสไม่ตรงกันแต่นำมารวมกัน ผลที่ได้คือคลื่นบางส่วนเป็นบวก บางส่วนหักล้างกัน โดยผลรวมคลื่นแสดงในรูปที่



รูปที่ 2.3 แสดงรูปผลของเฟส

ความแตกต่างระหว่างสัญญาณ เป็นผลจากการประวิงเวลาระหว่างคลื่น 2 คลื่น เมื่อมี 2 สัญญาณ ออกจากแหล่งกำเนิดที่มีระยะห่างจากผู้ฟังเท่ากัน ใช้เวลาและเฟสเดียวกันจากเวลาที่กำหนด สัญญาณเสียง ที่กล่าวมานี้ก็จะไปถึงผู้ฟัง แต่ถ้าหากแหล่งกำเนิดเสียงหนึ่งมีระยะทางมากกว่าแหล่งอื่น การประวิงก็จะเกิดขึ้น ความสัมพันธ์ของเฟสจะขึ้นอยู่กับการประวิงเวลา ใช้กฎนิ้วหัวแม่มือ เสียงจะเดินทางประมาณ 30 cm (1 ฟุต) ต่อ 1 ใน 1,000 วินาที ดังนั้นถ้าแหล่งกำเนิดที่ 2 อยู่สูงกว่าแหล่งกำเนิดที่หนึ่ง 1 m (3 ฟุตกว่า) เวลาการประวิงจะมีมากกว่า 3 ms ความสัมพันธ์ทางเฟสระหว่าง 2 สัญญาณควรขึ้นอยู่กับความถี่ของเสียง ที่ 330 Hz เวลาประวิง 3 ms สอดคล้องกับ 1 ความยาวคลื่น ทำให้การประวิงของสัญญาณเป็นไปตาม ลักษณะที่มีเฟสเดียวกัน และไม่มี การประวิงสัญญาณถ้าใช้เวลาในการประวิงเพียงครึ่งเดียว (1.5 ms) สัญญาณทั้ง 2 ควรจะมีเฟสที่ต่างกันที่ 330 Hz

เมื่อก้าวถึงเรื่องเฟสมักจะเกี่ยวข้องกับค่าองศา โดยพิจารณาได้จากรูปที่ 2.10 แสดงคลื่นรูป sine ที่ได้จากการเคลื่อนที่ของจุดบนขอบนอกของแผ่นกลม เริ่มต้นจาก 0-360 องศา ต่อ 1 รอบ แต่ละองศาของมุมการเคลื่อนที่ของบนจุด จะเป็นสัดส่วนโดยตรงกับค่า \sin ในตรีโกณมิติ เมื่อการเคลื่อนที่ของจุดมีมุม 90 องศา กราฟที่ได้จะมีค่าสูงสุดทางบวก ($\sin 90 = 1$) และที่มุม 180 องศา กราฟจะมีค่าเป็น 0 ($\sin 180 = 0$) ที่ 270 องศา มีค่าสูงสุดทางลบ ($\sin 270 = -1$) และที่ 360 องศา มีค่า 0 ต่อไป



รูปที่ 2.4 แสดงความสูงของจุดที่เปลี่ยนตามค่ามุม \sin มุมของเฟสคือจำนวนองศาของการหมุน

ถ้าคลื่น 2 คลื่นมีขนาดเท่ากันแต่เฟสไม่ตรงกัน คลื่นที่ 1 ต่างเฟสกับคลื่นที่ 2 คือเฟสทั้ง 2 จะต้องสัมพันธ์กันอย่างแท้จริง โดยเฉพาะรูปคลื่นที่เกิดขึ้นซ้ำๆ กันอย่างต่อเนื่อง เช่น ผลบางประการที่เกิดขึ้นต่อสัญญาณเสียงในภาวะชั่วคราวในช่วงเวลาสั้นๆ คือ การเปิด-ปิดเครื่องขยายมีเสียงดังออกจากลำโพงทันทีทันใด แม้ว่าจะไม่มีการเปิดเสียงก็ตามและส่งผลให้ลำโพงเสียหายได้หากไม่แก้ไข เพราะสัญญาณกระแสไฟสลับกระชอกเข้าวงจรอย่างรวดเร็ว เรียกว่า เสียงในภาวะชั่วคราว (Transient sounds)

ดังนั้นเรื่องของเฟสจึงมีส่วนเกี่ยวข้องกับวงจรขยายเสียง กล่าวคือ เมื่อมีสัญญาณผ่านเข้าวงจร มีการเปลี่ยนแปลงทาง amplitude และเฟสเกิดขึ้น จะเปลี่ยนไปตามความถี่ต่างๆ โดยการเปลี่ยนแปลง amplitude ต่อความถี่เรียกว่า การตอบสนองความถี่ (Frequency response) ส่วนเฟสของสัญญาณที่เลื่อนไปตามค่าความถี่ต่างๆ เรียกว่า การตอบสนองเฟส (Phase response)

2.1.5 เดซิเบล (Decibel)

เดซิเบล (Decibel) หรือ dB เป็นหน่วยพื้นฐานที่ใช้แพร่หลายในวิศวกรรมระบบเสียง พบเห็นอยู่บ่อยๆ ในการวัดระดับสัญญาณเสียง มีความสัมพันธ์กับ logarithm เรียกว่า logarithm ของอัตราส่วนระหว่างกำลังเสียง 2 สัญญาณ ตั้งหน่วยของระบบเป็น เบล (Bel) ให้เกียรติแก่ อเล็กซานเดอร์ เกรแฮม เบลล์ ผู้ประดิษฐ์โทรศัพท์ขึ้นเป็นคนแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned} \text{Bel} &= \log(P_1/P_2) \\ \text{dB} &= 10\log(P_1/P_2) \end{aligned}$$

ถ้าเดซิเบลเป็นอัตราส่วนระหว่างกำลังเสียง 2 สัญญาณ การเปรียบเทียบกับค่าอื่นๆที่นอกเหนือกำลังของเสียง หมายถึงกำลังเสียงที่อยู่ในรูปของแรงดันไฟฟ้า ได้ว่า

$$\begin{aligned} \text{dB} &= 10\log(V_1^2/V_2^2) & \text{เมื่อ } P &= V \cdot R \\ &= 10\log(V_1/V_2)^2 \\ &= 20\log(V_1/V_2) \end{aligned}$$

จากหน่วยวัด dB กำหนดเสียงที่หูมนุษย์ฟังได้ยินสูงสุดโดยไม่เป็นอันตรายต่อหู คือ ช่วง 120 dB ส่วนเสียงที่หูมนุษย์ฟังได้ยินเบาสุดคือ 0 dB

แหล่งกำเนิดเสียง	ระดับความดัง (dB)
เสียงเบาที่สุดที่หูมนุษย์เริ่มได้ยิน	0
เสียงลมหายใจ	10
เสียงใบไม้กระทบกันเมื่อลมพัด	20
เสียงในห้องเงียบๆ	30
เสียงนกร้องจากระยะไกลๆ	40
เสียงในห้องที่เปิดประตูทิ้งไว้	50
เสียงสนทนาห่างกัน 3 ฟุต	60
เสียงสนทนาห่างกัน 1 ฟุต	70
เสียงดนตรีออร์เคสตราในระดับเฉลี่ย	80
เสียงพัดลมระบายอากาศขนาดใหญ่	90
เสียงเลื่อยยนต์	100
เสียงวงดนตรี concert แนว rock	110
เสียงเครื่องบินไอพ่นขึ้นใน ระยะ 1500 ฟุต	120
หูมนุษย์เริ่มได้รับอันตรายจากการได้ยิน	130-140
จุดที่ทำให้หูหนวกเฉียบพลัน	150-160

ตารางที่ 2.1 แสดงระดับความดังของเสียงจากแหล่งกำเนิดเสียงต่างๆ

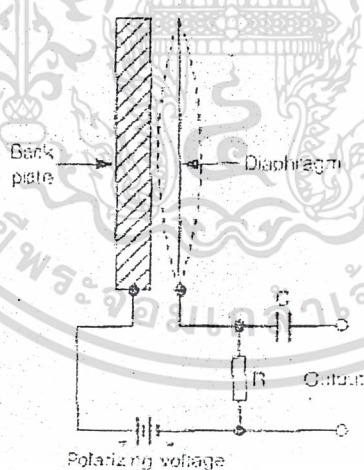
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไมโครโฟน (Microphone)

ไมโครโฟนเป็นอุปกรณ์พื้นฐานชนิดหนึ่งที่ทำหน้าที่เปลี่ยนพลังงานเสียงให้เป็นพลังงานไฟฟ้า พลังงานเสียงจะเคลื่อนที่ในลักษณะคลื่น มีขนาดและความถี่ปนกันออกไป ไมโครโฟนจะใช้คุณลักษณะการเคลื่อนที่แบบคลื่นของพลังงานเสียงมาแปลงเป็นพลังงานไฟฟ้า ไมโครโฟนที่นิยมใช้กันทั่วไปมี 3 แบบ คือ ชนิดขดลวดที่เคลื่อนที่ได้ในสนามแม่เหล็ก เรียกว่า ไดนามิก ชนิดรีบอน และชนิดล้วยเก็บประจุ หรือ คอนเด็นเซอร์

2.2.1 ไมโครโฟนแบบตัวเก็บประจุ

ไมโครโฟนแบบตัวเก็บประจุ จะมี ไดอะแฟรม แตกต่างจากไมโครโฟนแบบอื่นๆ กล่าวคือ ประกอบด้วยแผ่นโลหะ 2 แผ่น แผ่นหนึ่งจะอยู่กับที่ อีกแผ่นเคลื่อนที่ได้อย่างอิสระ ทำหน้าที่เป็นแผ่นไดอะแฟรม เป็นผลทำให้ค่าความจุเปลี่ยนแปลง โดยแผ่นโลหะทั้ง 2 จะถูกกั้นด้วยฉนวน เมื่อมีคลื่นเสียงมากระทบที่แผ่นไดอะแฟรม ทำให้ค่าความจุเปลี่ยนแปลง เกิดการเปลี่ยนแปลงของกระแสที่ไหลผ่านตัวต้านทาน และค่าแรงดันตกคร่อมเป็นสัญญาณ output ที่นำไปใช้งาน



รูปที่ 2.5 แสดง โครงสร้างของไมโครโฟนแบบตัวเก็บประจุ

โดยทั่วไปไมโครโฟนแบบนี้จะมีแหล่งจ่ายกระแสไฟฟ้ามาเลี้ยงตัวไมโครโฟน แสดงข้อความโครงสร้างคร่อม 2 แผ่นโลหะ ผ่านตัวต้านทานที่มีความต้านทานสูงมาก หากพิจารณาแล้ว ไมโครโฟนชนิดนี้ มีคุณภาพสูง ความไวเสียงในการรับเสียงสูง ตอบสนองความถี่ได้ดี แต่ราคาแพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ไมโครคอนโทรลเลอร์ (Microcontroller) 89C51

ไมโครคอนโทรลเลอร์ 8051 เป็นไมโครคอนโทรลเลอร์ในตระกูล mcs-51 ที่บริษัท Intel ผลิตขึ้น มีคุณสมบัติดังนี้

- เป็นไมโครโปรเซสเซอร์ขนาด 8 bit สำหรับงานควบคุมต่างๆ
- มีความสามารถประมวลผลของ logic ได้ในระดับ bit
- มีขนาดของหน่วยความจำสำหรับโปรแกรมทำงานได้ถึง 64 kbytes (Program Memory)
- มีขนาดของหน่วยความจำสำหรับเก็บข้อมูลได้ถึง 64 kbytes (Data Memory)
- มีหน่วยความจำสำหรับโปรแกรมภายในขนาด 4 kbytes
- มีหน่วยความจำข้อมูลภายในขนาด 128 bytes
- มีพอร์ตควบคุม 4 พอร์ตและสามารถอ้างอิงได้ระดับ bit ต่อ bit
- มีชุด Timer / Counter ขนาด 16 bit 2 ชุด
- มี Full duplex UART
- มีโครงสร้างรับบริการ Interrupt 6 แหล่งกำเนิดสัญญาณและ 5 ตำแหน่งโปรแกรมทำงานตอบรับการ Interrupt โดยสามารถจัดระดับความสำคัญได้ 2 ระดับ
- มีแหล่งกำเนิดความถี่อ้างอิงการทำงานในตัว

2.3.1 รายละเอียดของขาสัญญาณของ 89C51

Vcc	:	ขาแหล่งจ่ายไฟฟ้า (+5 V)
Vss	:	ขา Ground
P0	:	ขา port 0 ของ 8051 ขนาด 8 bit ชนิด 2 ทิศทาง แต่ละสัญญาณต่อพ่วง TTL แบบ LS ได้ 8 ตัว เป็นขาให้ Multiplex ระหว่างสัญญาณข้อมูล กับ 8 bit ล่างของ Address กรณีใช้หน่วยความจำภายนอก
P1	:	ขา port 1 ของ 8051 ขนาด 8 bit ชนิด 2 ทิศทางแบบ Quasi-bidirectional ถ้าต้องการให้เส้นใดเป็น input ให้ค่า 1 ที่ bit นั้น ต่อพ่วง LS TTL 4 ตัว
P2	:	ขา port 2 ของ 8051 ขนาด 8 bit ชนิด 2 ทิศทางแบบ Quasi-bidirectional ยังให้สัญญาณ Address 8 bit บนกรณีใช้หน่วยความจำภายนอก กรณีอ้างอิง Address หน่วยความจำขนาด 16 bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

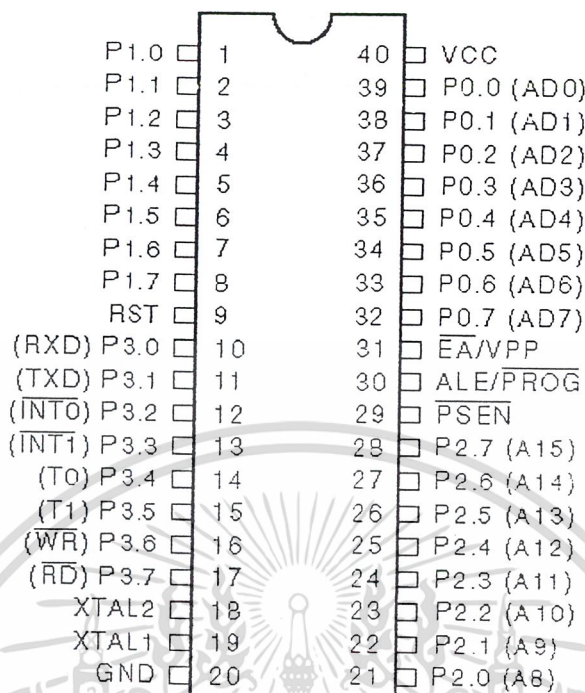
P3 : ขา port 3 ของ 89C51 ขนาด 8 bit ชนิด 2 ทิศทางแบบ Quasi-bidirectional มีหน้าที่พิเศษดังตารางข้างล่าง

ขา Port	หน้าที่พิเศษ	อธิบาย
P3.0	RxD	รับข้อมูลแบบอนุกรม
P3.1	TxD	ส่งข้อมูลแบบอนุกรม
P3.2	$\overline{INT0}$	ขา Interrupt ภายนอก 0
P3.3	$\overline{INT1}$	ขา Interrupt ภายนอก 1
P3.4	T0	ขา input ของ Timer 0
P3.5	T1	ขา input ของ Timer 1
P3.6	\overline{WR}	เขียนหน่วยความจำข้อมูลภายนอก
P3.7	\overline{RD}	อ่านหน่วยความจำข้อมูลภายนอก

ตารางที่ 2.2 แสดงหน้าที่ของขา port 3 ของ 89C51

- RST : ขาสำหรับ Reset การทำงานของ 8051 โดยให้ logic 1 อย่างน้อย 2 ช่วง Machine Cycle
- ALE : ขาสัญญาณออกของ Address Latch Enable สำหรับ Latch ค่า Address 8 bit ที่ได้จาก port 0 มีความถี่ออกมาที่ 1/6 ของความถี่อ้างอิงของ 89C51
- PSEN : ขาสัญญาณ Program Store Enable สำหรับหน่วยความจำโปรแกรมภายนอก
- \overline{EA} : ขาสัญญาณ External Access Enable ให้อ่านหน่วยความจำภายในหรือภายนอก ให้ logic 0 89C51 จะอ่านหน่วยความจำโปรแกรมภายนอก และ logic 1 8051 จะอ่านหน่วยความจำโปรแกรมภายใน
- XTAL 1 : ขาเข้าของวงจรกำเนิดความถี่อ้างอิงภายในของ 89C51
- XTAL 2 : ขาออกของวงจรกำเนิดความถี่อ้างอิงภายในของ 89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงขาของ 89C51

2.3.2 ชุดคำสั่งของ 89C51

การใช้งานชุดคำสั่งของ 89C51 มีการอ้างอิงตำแหน่งหลายแบบ ดังนี้

- **Direct Addressing** เป็นการอ้างอิงที่ต้องถูกกระทำ มีขนาด 8 bit ใช้ได้เฉพาะกับหน่วยความจำข้อมูลและ SFR เท่านั้น
- **Indect Addressing** เป็นการอ้างตำแหน่ง โดยการใช้ register เป็นตัวชี้ไปยังข้อมูลที่สั่งการจะกระทำ เป็นการอ้างตำแหน่งของหน่วยความจำข้อมูลทั้งภายในและภายนอก อ้างได้โดยใช้ register ขนาด 8 และ 16 bit โดยอ้างอิงขนาด 8 bit ใช้ R0 และ R1 หรือใช้ SP เป็นตัวชี้ แต่กรณีขนาด 16 bit ใช้ DTPR
- **Register Addressing** เป็นคำสั่งเฉพาะเพื่อการอ้าง register R0 – R7 ตัวชี้เป็นส่วนหนึ่งของคำสั่ง 3 bit
- **Immediate Constants** เป็นการกำหนดค่าคงที่โดยตรง รูปแบบ MOV A, #100
- **Indexed Addressing** เป็นลักษณะสร้างตาราง โดยมีตัวชี้เป็นตัวอ้างอิงและไม่เปลี่ยนแปลง ใช้ได้เฉพาะส่วนหน่วยความจำโปรแกรมเท่านั้น ใช้ register DTPR และ PC เป็นตัวชี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

89C51 มีชุดคำสั่งที่สามารถแบ่งออกเป็นกลุ่มๆ ได้ ดังนี้

1. กลุ่มคำสั่งการคำนวณ (Arithmetic Instructions) :ทำหน้าที่ในการคำนวณทางคณิตศาสตร์ เช่น การบวก ,ลบ , คูณ ,หาร ข้อมูลภายใน register ต่างๆ
2. กลุ่มคำสั่งลอจิก (Logic Instructions) :ทำหน้าที่เกี่ยวกับการประมวลผลแบบ logic ต่างๆ เช่น AND , OR , XOR หรือ NOT ระหว่างข้อมูลใน register A กับ register ทั่วไป และการทำคำสั่งเกี่ยวกับการเลื่อน หรือหมุน bit ข้อมูลใน register A เป็นต้น
3. กลุ่มคำสั่งโอนย้ายข้อมูล (Data Transfers Instructions) :ทำหน้าที่ในการโอนย้ายข้อมูล ระหว่าง register หรือ หน่วยความจำ
4. กลุ่มคำสั่งข้อมูลตาราง (Lookup Table Instructions) :ทำหน้าที่อ่านข้อมูลที่เก็บในลักษณะของตารางใช้ได้เฉพาะในหน่วยความจำโปรแกรมเท่านั้น
5. กลุ่มคำสั่งกระทำทาง Boolean (Boolean Instructions) :ทำหน้าที่ดำเนินการประมวลผลแบบ Boolean จะกระทำการประมวลผลเฉพาะระดับ bit เท่านั้น
6. กลุ่มคำสั่งกระโดด (Jump Instructions) :ทำหน้าที่ในการกระโดดไปยังตำแหน่งต่างๆ ภายใน program แทนที่จะดำเนินการเป็นลำดับต่อเนื่อง สามารถกระทำการกระโดดได้ทั้งแบบมี และ ไม่มีเงื่อนไข รวมไปถึงการเรียกใช้งานโปรแกรมย่อย และการกลับมาทำโปรแกรมหลักด้วย

โดยสามารถหารรายละเอียดของคำสั่งต่างๆได้จาก ภาคผนวก ด้านหลัง

2.4 พอร์ตอนุกรม (serial port)

การสื่อสารแบบอนุกรมแบ่งออกเป็น 2 แบบ คือ การสื่อสารแบบซิงโครนัส และ การสื่อสารแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาที่ร่วมอยู่กับการรับและส่งสัญญาณด้วย สายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา สายอีกเส้นจะเป็นสายของข้อมูล และจะมีสายของ กราวด์อีกเส้น อย่างน้อยต้องมี 3 เส้น

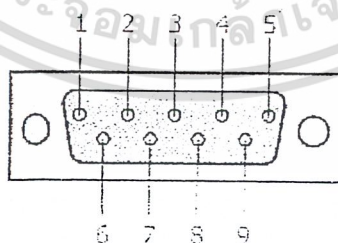
2.4.1 การสื่อสารข้อมูลแบบซิงโครนัส

การส่งและรับข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาที่ร่วมด้วยแต่ใช้การกำหนดค่าสัญญาณนาฬิกาให้มีค่าเท่ากัน เรียกว่า อัตราการถ่ายเทข้อมูล หรือ บอครเรล (baudrate) หน่วยเป็น บิตต่อวินาทีรูปแบบข้อมูลประกอบด้วย 4 ส่วน คือ

1. บิตเริ่มต้น (Start Bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) มีขนาด 1 บิต หรือ ไม่มี
4. บิตปิดท้าย (Stop Bit) มีขนาด 1,1.5 หรือ 2 บิต

2.4.2 คอนเน็กเตอร์ พอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 ใช้คอนเน็กเตอร์แบบ DB-9 ตัวผู้มีอยู่ 9 เส้น ดังรูป



รูปที่ 2.7 คอนเน็กเตอร์อนุกรม 9 เส้น แบบ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา DB – 9	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	Data Carrier Detect : DCD	Input
2	Received Data : RxD	Input
3	Transmitted Data : TxD	Output
4	Data Terminal Ready : DTR	Output
5	Ground : GND	-
6	Data Set Ready : DSR	Input
7	Request To Send : RTS	Output
8	Clear To Send : CTS	Input
9	Ring Indicator : RI	Input

ตารางที่ 2.3 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรม RS-232 แบบ DB – 9

ขาสัญญาณ input ของพอร์ตอนุกรม

ขาที่ทำหน้าที่เป็นขา input ได้แก่ ขา DCD,CTS,DSR,RI และ RxD

ขาสัญญาณ output ของพอร์ตอนุกรม

ขาที่ทำหน้าที่เป็นขา output ได้แก่ ขา DTR,RTS และ TxD

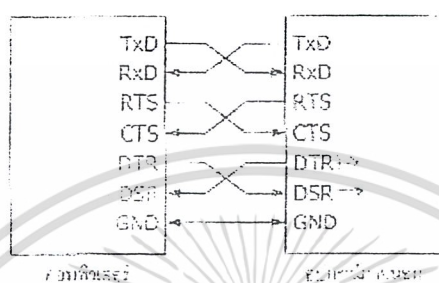
2.4.3 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม RS-232

มาตรฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันการทำงานของพอร์ตอนุกรมไว้ว่า ที่ ลอจิก “0” มีระดับแรงดัน +3 ถึง +15 V ส่วน ลอจิก “1” จะมีระดับสัญญาณ -3 ถึง -15 V ระดับสัญญาณนี้ไม่สามารถที่จะนำขา output ใดๆ ค่อยเข้ากับลอจิกเกต เพื่อใช้งานได้โดยตรง ต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน ปกติใช้ IC พวก RS-232 transceiver ที่นิยมคือ MAX232 และ ICL232 จะทำหน้าที่แปลงแรงดันของ RS-232 ให้อยู่ในระดับ TTL โดยลอจิก “0” เดิมมีระดับสัญญาณ +3 ถึง +15 V จะถูกแปลงเป็น 0V ส่วน ลอจิก “1” จาก -3 ถึง -15 V แปลงเป็น +5 V เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ digital อื่นๆ ได้

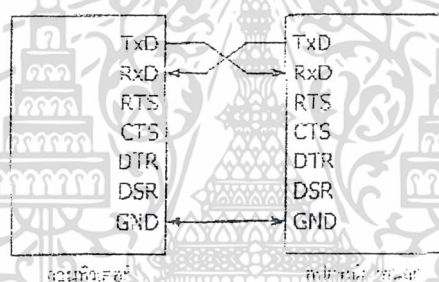
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 การเชื่อมต่อ computer กับ อุปกรณ์ภายนอก

จากรูปเป็นการเชื่อมต่อแบบ Null Modem ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์



(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null modem



(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

รูปที่ 2.8 การต่ออุปกรณ์ภายนอก กับ พอร์ตอนุกรมภายนอกในลักษณะต่างๆ

Receive Data : RD , RxD : ใช้เพื่อรับสัญญาณอนุกรมเข้ามายัง computer โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์

Transmitted Data : TD , TxD : ใช้เพื่อส่งข้อมูลออกจาก computer โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

Signal Ground : GND : ขากราวด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 การเขียนโปรแกรม Visual Basic เพื่อใช้งานพอร์ตอนุกรม

สำหรับโครงการนี้ได้เลือกใช้งาน โปรแกรม Visual Basic 6.0 ในการเขียนโปรแกรม เพื่อใช้งาน พอร์ตอนุกรม ซึ่งจะมี custom control สำหรับการสื่อสารอนุกรมผ่านพอร์ตอนุกรมของ computer มาให้แล้ว คือ MSCOMM32.OCX

MSComm ได้จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวก ทางแรกคือ การสื่อสารข้อมูลที่ กระตุ้นด้วยเหตุการณ์ (Event - Driven Communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมาก สำหรับการตอบสนองแบบทันทีทันใด สามารถตรวจจับสัญญาณได้ทันที ทางเลือกที่ 2 เป็นการคอย ตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ

CommEvent

คอนโทรล MSComm 1 ตัว สามารถควบคุมการทำงานของ พอร์ตอนุกรมได้ 1 พอร์ต ถ้าใช้งาน มากกว่า 1 พอร์ต ต้องใช้คอนโทรล MSComm มากกว่า 1 ตัว คอนโทรล MSComm มีคุณสมบัติ (property) มากมาย ดังนี้

CommPort ใช้กำหนดและอ่านค่าพอร์ตอนุกรมที่ติดตั้งอยู่ (COM1 , COM2 , COM3 , COM4)

รูปแบบการใช้งาน `object . CommPort [= value]`

Setting ใช้ในการกำหนดและอ่านค่าอัตราบอด, พาริตี, จำนวนของบิตข้อมูล, จำนวนบิตปิดท้าย

รูปแบบการใช้งาน `object . Setting [= value]`

PortOpen ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดปิดพอร์ตอนุกรม

รูปแบบการใช้งาน `object . PortOpen [= value]`

Input อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน `object . Input`

InBufferCount ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน `object . InBufferCount [= value]`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- InBufferSize** กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยไบต์
รูปแบบการใช้งาน `object . InBufferSize [= value]`
- InputLen** กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ
รูปแบบการใช้งาน `object . InputLen [= value]`
- InputMode** กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input
รูปแบบการใช้งาน `object . InputMode [= value]`
- Output** ใช้ในการส่งขบวนของข้อมูล ไปยังบัฟเฟอร์ส่งข้อมูล
รูปแบบการใช้งาน `object . Output [= value]`
- OutBufferCount** คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และ เพื่อเคลียร์บัฟเฟอร์ภาคส่ง
รูปแบบการใช้งาน `object . OutBufferCount [= value]`
- OutBufferSize** กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดสลับแปรเป็นแบบ ไบต์
รูปแบบการใช้งาน `object . OutBufferSize [= value]`
- ParityReplace** กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี
รูปแบบการใช้งาน `object . ParityReplace [= value]`
- DTREnable** กำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) สัญญาณของขา DTR ส่งจาก computer ไปยัง modem เพื่อแสดงว่าพร้อมที่จะรับข้อมูลแล้วชนิดแบบบูลีน
รูปแบบการใช้งาน `object . DTREnable [= value]`
- RTSEnable** กำหนดสถานะลอจิกให้ขา Request To Send (RTS) เป็นสัญญาณ ส่งจาก computer ไปยัง modem เพื่อร้องขอส่งข้อมูล ชนิดข้อมูลแบบ Boolean
รูปแบบการใช้งาน `object . RTSEnable [= value]`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EOFEnable กำหนดให้ MSCComm รอสัญลักษณ์แสดงส่วนท้ายสุดของไฟล์ (End Of File : EOF) ระหว่างการรับ Input เข้ามา ถ้าพบ EOF ภาคอินพุทจะหยุดรับข้อมูล เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน

รูปแบบการใช้งาน `object . EOFEnable [= value]`

CTSHolding ตรวจสอบการทำงาน Clear To Send (CTS) ได้ว่ามีการสถานะ 0 หรือ 1 ค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่าเป็น True มีลอจิก 1 และ ถ้าค่าเป็น False มีลอจิก 0

รูปแบบการใช้งาน `object . CTSHolding`

CDHolding ตรวจสอบการทำงาน Data Carrier Detect (DCD) ได้ว่ามีการสถานะ 0 หรือ 1 ค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่าเป็น True มีลอจิก 1 และ ถ้าค่าเป็น False มีลอจิก 0

รูปแบบการใช้งาน `object . CDHolding`

DSRHolding ตรวจสอบการทำงาน DSR ได้ว่ามีการสถานะ 0 หรือ 1 ค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่าเป็น True มีลอจิก 1 และ ถ้าค่าเป็น False มีลอจิก 0

รูปแบบการใช้งาน `object . DSRHolding`

Handshaking กำหนดคุณสมบัติและค่านำรูปแบบ Handshaking ทาง Hardware

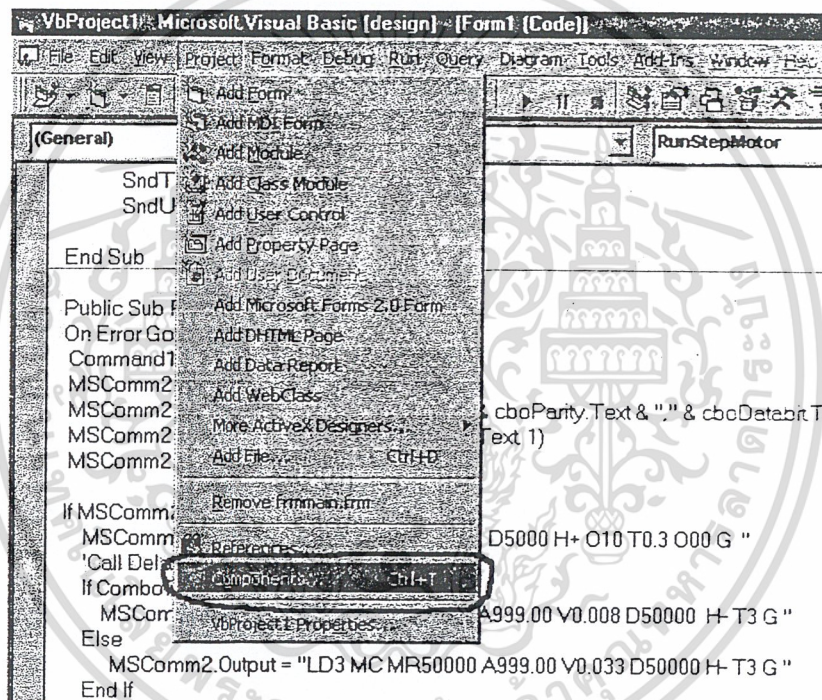
รูปแบบการใช้งาน `object . Handshaking [= value]`

Break ใช้เซตและเคดิตร์ค่าสัญญาณ Break ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน `object . Break [= value]`

ในทางปฏิบัติสามารถทำได้โดยใช้ VB Control ที่ชื่อว่า MSComm โดย กำหนด Custom Control เข้าไปที่ เมนู Project--> Components แล้วเลือกที่ช่อง MSComm ก็จะปรากฏ เป็นรูปไอคอนโทรศัพท์สีเหลือง ให้คลิกที่ไอคอนลากนำมาไว้บน Form ใน Project ของ โปรแกรมเรา โดยสามารถทำตามวิธีที่กล่าวมา ได้ดังรูป

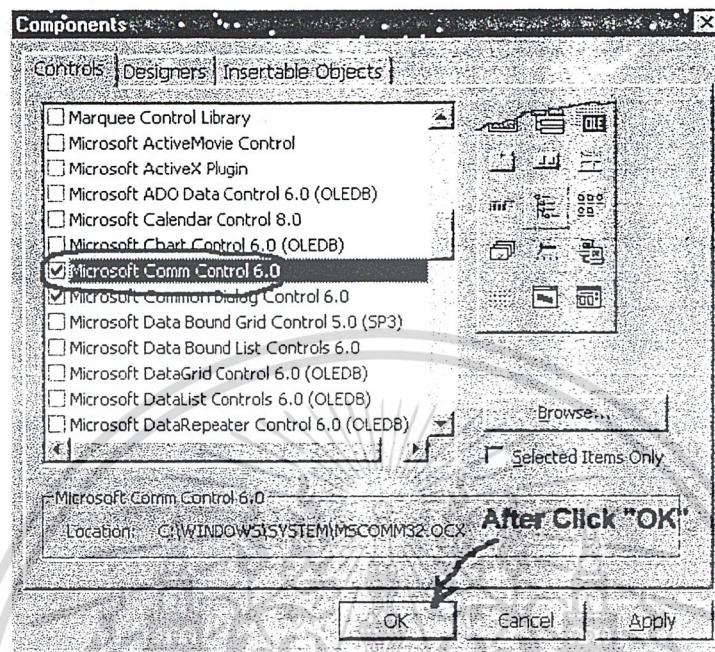
1. ขั้นที่ตอนแรก เลือกที่เมนูบาร์ด้านบนของ โปรแกรม Visual Basic ดังรูปด้านล่าง



รูปที่ 2.9 แสดงรูปขั้นตอนแรกการเลือกใช้งาน MSComm Control

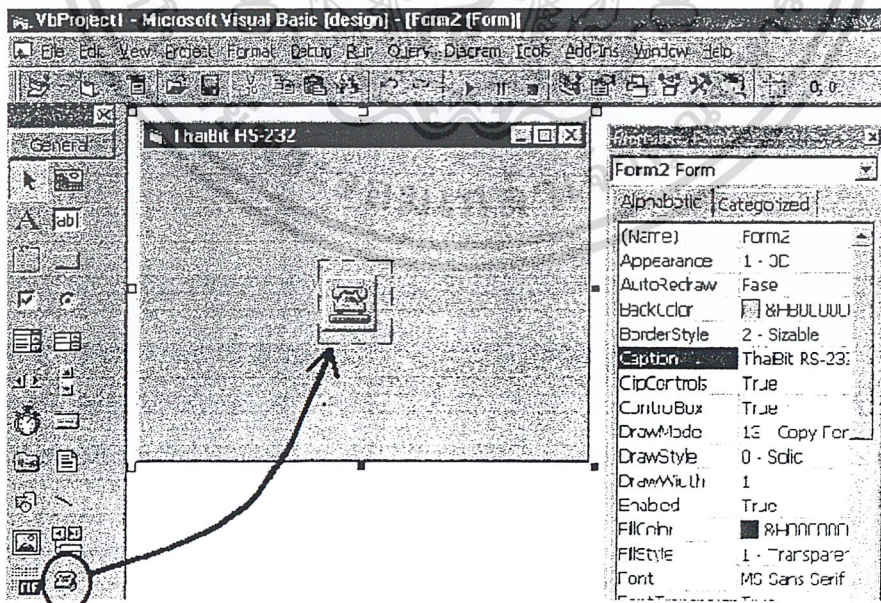
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือกชื่อ Control ชื่อ Microsoft Comm Control 6 ดังรูปด้านล่าง



รูปที่ 2.10 แสดงรูปขั้นตอนที่ 2 การเลือกใช้งาน MSComm Control

3. ลาก Control ชื่อ Microsoft Comm จาก Toolbox มาไว้บน Form ดังรูป



รูปที่ 2.11 แสดงรูปขั้นตอนที่ 3 การเลือกใช้งาน MSComm Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

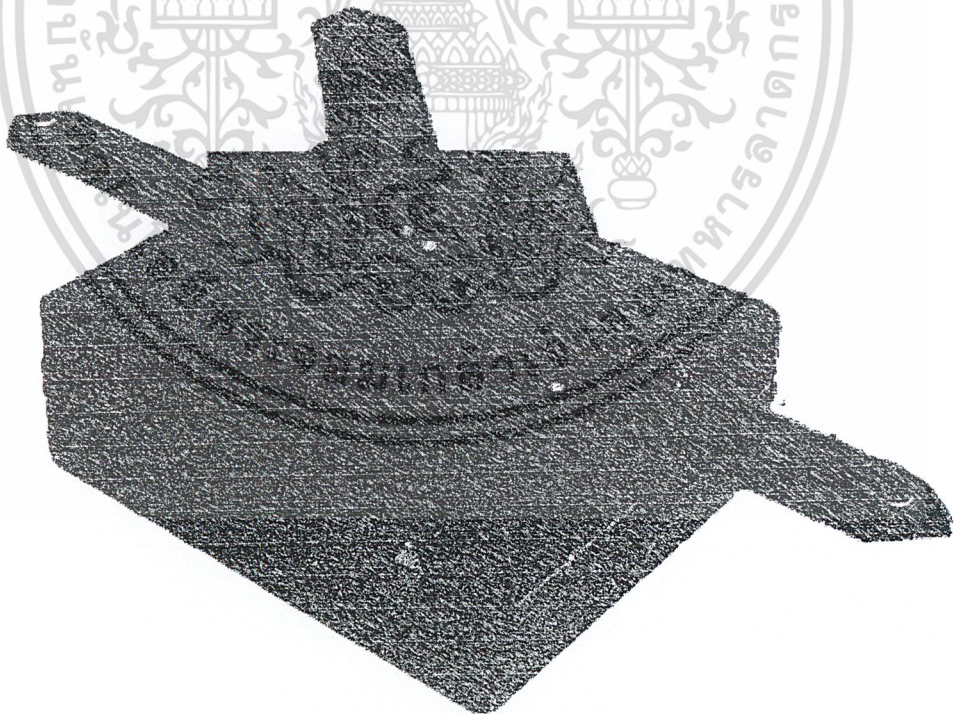
บทที่ 3

การออกแบบและการคำนวณ

3.1 ลักษณะเฉพาะทางโครงสร้าง (Mechanics Details)

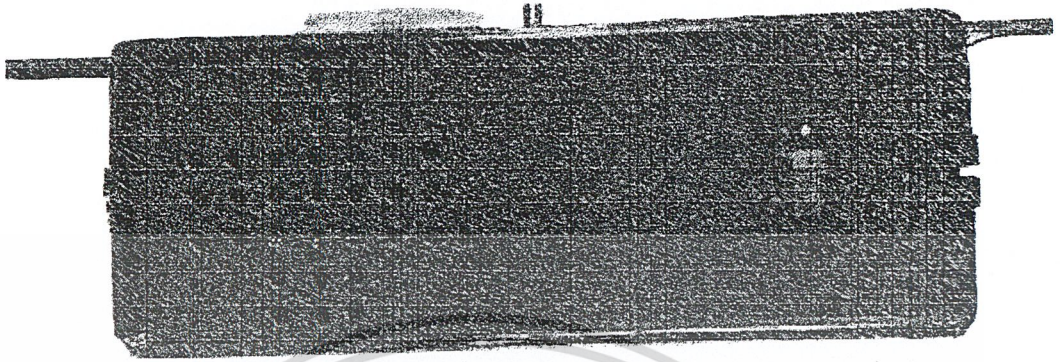
ลักษณะของหุ่นประกอบด้วย

1. ขนาดของตัวหุ่น $20 \times 22 \times 8$ cm (ถ้ารวมตัวกล้องด้วยจะมีความสูง 15 cm)
 2. โครงสร้างตัวหุ่นใช้กล่องพลาสติก
 3. ประกอบด้วยมอเตอร์ในการหมุนตัวกล้องเป็น DC Motor มีการทดเฟืองเพื่อไปหมุนความต้านทานปรับค่าได้ โดยมีอัตราทดของ VR ต่อ Motor เป็น 52:30 (เหตุผลที่ต้องใช้เฟืองทดที่ VR มากกว่าเพราะ VR ไม่สามารถหมุนได้รอบ แต่ต้องการสอบการหมุนให้ครบรอบ 0- 360 องศา นั่นเอง)
 4. ไมโคร โฟน 2 ตัววางห่างจากจุดหมุนกล้อง 19 ซม. วางที่จุด 0 องศา และ 180 องศา และวางห่างจากกัน 38 ซม.
 5. มีวงจรควบคุมการทำงานอยู่ภายในของตัวหุ่น
 6. ใช้หม้อแปลงไฟฟ้ากระแสสลับชนิด Primary 220 V และชนิด Secondary 9 และ 12 V
- ลักษณะตัวหุ่น ทั้งภายนอกและภายใน แสดงดังรูป

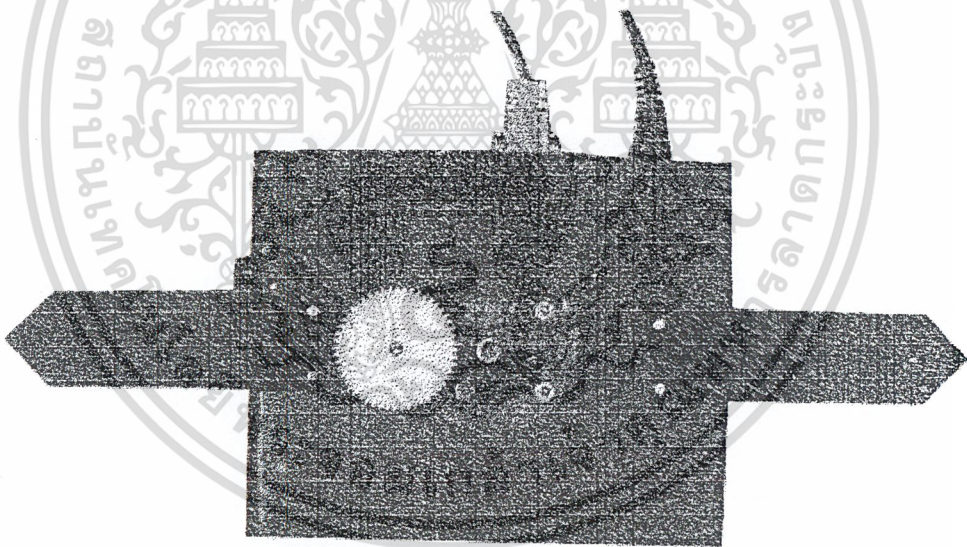


รูปที่ 3.1 แสดงลักษณะตัวหุ่น ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

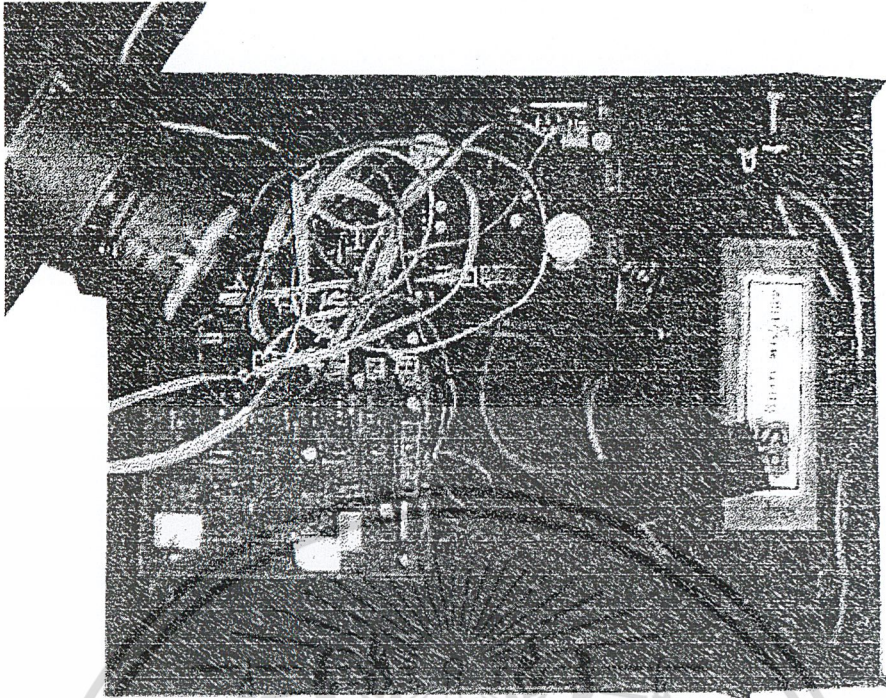


รูปที่ 3.2 แสดงลักษณะตัวหุ่น ภายนอก (ด้านหน้า)



รูปที่ 3.3 แสดงลักษณะตัวหุ่น ภายนอก (ด้านหลัง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงลักษณะตัวหุ่น ภายใน



รูปที่ 3.5 แสดงรูปร่างของไมโครโฟนที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อุปกรณ์ที่ใช้ (Hardware Tools)

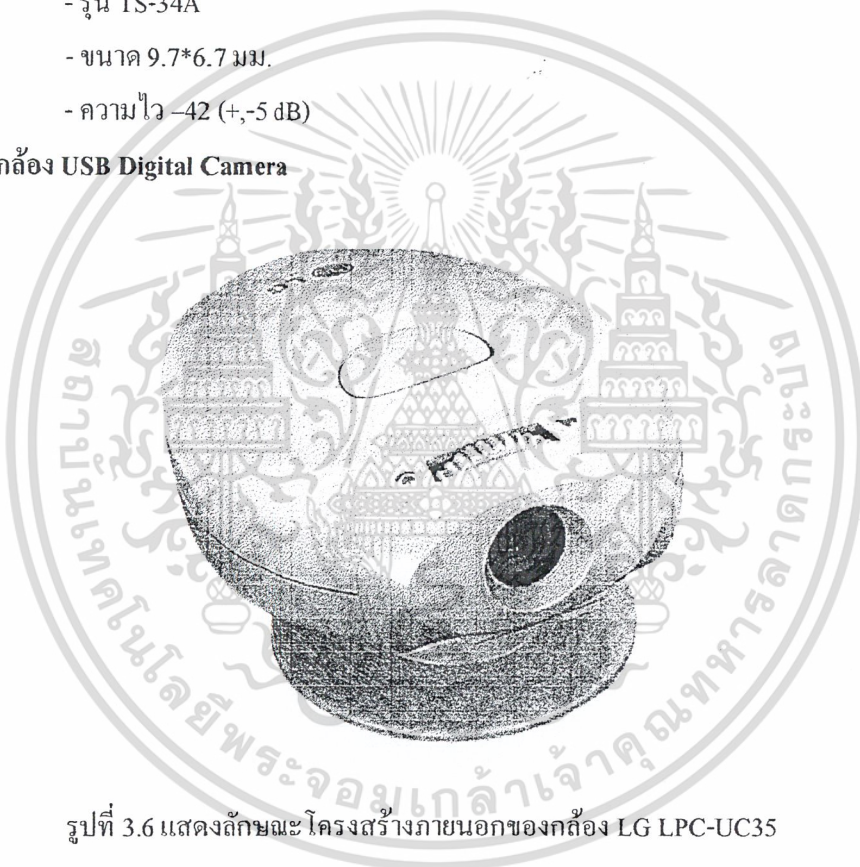
1. มอเตอร์ขับเคลื่อน (Motor Driver)

- เป็น Geared DC Motor 12 V
- ความเร็วรอบ 30 RPM
- ใช้กระแสที่ No-load 0.2 A

2. ไมโครโฟน (Microphone)

- เป็น ไมโครโฟนแบบคอนเด็นเซอร์ แบบ Omnidirectional Pattern
- รุ่น TS-34A
- ขนาด 9.7*6.7 มม.
- ความไว -42 (+,-5 dB)

3. กล้อง USB Digital Camera



รูปที่ 3.6 แสดงลักษณะ โครงสร้างภายนอกของกล้อง LG LPC-UC35

คุณสมบัติที่สำคัญของกล้อง

เลนส์	F24
ระยะ โฟกัส	$f = 6.0 \text{ mm}$
องศาของมุมมอง ***	แนวระดับ : 38° , แนวตั้ง : 33°
โฟกัส	15 mm - ∞
ชนิดของอุปกรณ์ตรวจจับภาพ	1/4 " CCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวน Pixel	350,000 pixels
ความเร็วในการประมวลผล Transmitting (max)	QSIF : 30 Frames/sec
	SIF : 30 Frames/sec
	VGA : 15 Frames/sec
ลักษณะของ Output	USB Format
แหล่งจ่ายไฟ	5.0V ผ่าน USB
ขนาด	70 x 44 x 95.5 (กว้าง x สูง x ยาว)
น้ำหนัก	90 กรัม

ตารางที่ 3.1 แสดงคุณสมบัติที่สำคัญของกล้อง

สิ่งที่จำเป็นต้องมีในการใช้งานกล้อง ได้แก่

- เครื่อง Computer ตั้งแต่รุ่น Pentium Processor MMX 233MHz ขึ้นไป
- Port USB
- RAM 32MB ขึ้นไป แนะนำ 64 MB
- Windows 98,2000 และ XP
- จอที่มี Resolution 640 x 480 ขึ้นไป
- เนื้อที่ harddisk 6MB เพื่อลง program Camera Viewer
- CD-ROM Drive

3.3 กฎการออกแบบ (Design Rules)

ในการออกแบบจำเป็นต้องมีการคำนวณค่าต่างๆเพื่อใช้เป็นพื้นฐานในการออกแบบชิ้นต้นซึ่งอาจเรียกได้ว่าเป็น "กฎการออกแบบ" ซึ่งมีดังต่อไปนี้

3.3.1 การหาระยะห่างของไมโครโฟนและค่า Delay ที่วัดได้

ดูจากโครงสร้างทาง Mechanics จะเห็นได้ว่าไมโครโฟนสองตัวห่างกัน 38 ซม. และถ้ากำหนดให้ความเร็วของเสียงที่ 25 องศาเซลเซียส มี ค่า 340 M/S ได้ ค่า Delay ที่เป็นไปได้ คือ

$$0.38/340 = 1.12 \text{ mS}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ค่า Delay ที่สามารถวัดได้

การวัดค่า Delay จะใช้ Timer0 ของ MCS-51 ซึ่งตั้งค่าแบบ 16 Bits และใช้

- คริสตอลขนาด 11.0592 MHz เป็นฐานเวลา สิ่งที่ต้องคำนึงถึง 3 อย่างในการวัด Delay คือ 1.ค่าเวลาที่วัดได้(ค่าล่าช้าในการเริ่มวัด) 2.ค่าล่าช้าจากการหยุดนับ 3.ค่า Delay มากที่สุดที่สามารถวัดได้

1. ค่าเวลาที่วัดได้ สามารถวัดได้ โดยดูจากจำนวน Machine Cycle ของคำสั่งที่ใช้ (ดูจาก Source Code MCS-51 บทที่ 5) ซึ่งคำสั่งที่ใช้ในการวัด Delay ดังนี้

- JMP ;2 Machine Cycle
- SETB TR0 ;1 Machine Cycle

การคำนวณเวลาล่าช้าในการเริ่มวัด สามารถคำนวณได้จากสมการ

$$T=C*(12/Fxtal)$$

ซึ่ง T = ค่าเวลาที่ใช้ในการประมวลผล

C = จำนวน Machine Cycle

Fxtal = ความถี่คริสตอลที่ใช้

จากกรณีข้างต้น ใช้ 3 Machine Cycle และคริสตอล 11.0592 ได้

$$3*(12/11.0592*10^6)=3.26 \text{ uS}$$

2.ค่าล่าช้าจากการหยุดนับ ก็เช่นเดียวกับกรณีแรก แต่ก่อนที่จะหยุดนับ จะเสียเวลาในการสั่งดังนี้

- JB..... ;2 Machine Cycle
- JC..... ;2 Machine Cycle
- CLR TR0 ;1 Machine Cycle

คำนวณค่าได้

$$5*(12/11.0592*10^6)=5.43 \text{ uS}$$

3.ค่า Delay มากที่สุดที่สามารถวัดได้ ดูได้จากลักษณะการทำงานของ Timer ซึ่งจะเพิ่มทุก Machine Cycle และเราใช้ 16 Bits Timer ซึ่งเท่ากับ 65536 Machine Cycle คำนวณเวลาได้

$$65536*(12/11.0592*10^6)=71\text{mS}$$

3.3.3 การตรวจวัดมุมด้วย ADC และ การหมุนของ Motor

การทำงานของ ADC0804 จะใช้การ Interrupt ของ MCS-51 ในการส่งค่า (รายละเอียด อยู่ในบทที่ 5) ซึ่งเราต้องตรวจวัดความเร็วของการหมุนของ Motor เทียบกับความเร็วในการแปลงค่าของ ADC และคำสั่งที่ MCS-51 ใช้

1. ความเร็วในการหมุนเทียบกับค่าโวลต์ที่เปลี่ยนแปลง จากความเร็วรอบของ Motor และการทดเฟือง ซึ่งมีอัตราทดของ VR ต่อ Motor เป็น 52:30 และความเร็วรอบ 30 RPM คิดเป็น 0.5 รอบ ต่อ วินาที นั่นคือใน 1 วินาที จะหมุนเฟืองของ Motor ไป 15 ซี่ และหมุนเฟือง VR ไป 15 ซี่ เช่นกัน คิดเป็นอัตราส่วน 15/52 ซึ่งนำมาเทียบกับการแปลงซึ่งมี 255 ค่า จะได้อัตราการเปลี่ยนแปลงค่าโวลต์(ดิจิตอล)ใน 1 วินาที เท่ากับ $(15/52)*255 = 73.557$ ประมาณได้ 74 ระดับต่อ 1 วินาที

2. ความเร็วในการแปลง จากคำสั่งของ MCS-51 ที่ใช้และความเร็วในการแปลงค่าของ ADC

สมมุติว่าที่เวลาขณะหนึ่ง MCS-51 มีค่าจาก ADC เรียบร้อยและกำลังจะหมุนไปอีกองศาหนึ่ง จะใช้คำสั่ง

MOV A.... ;1 Machine Cycle

CJNE A.... ;2 Machine Cycle

MOV A.... ;1 Machine Cycle

CJNE A... ;2 Machine Cycle

ADC ใช้เวลาแปลงค่ามากที่สุด 100 μ S

เมื่อเกิด Interrupt ใช้คำสั่ง คือ

CALL..... ; 2 Machine Cycle

MOV ; 2 Machine Cycle

CLR..... ; 1 Machine Cycle

SETB..... ; 1 Machine Cycle

RETI..... ; 2 Machine Cycle

ดังนั้นเวลาดานานสุดที่ใช้ในการแปลงค่าหนึ่งเท่ากับ

$$(100*10^{-6}) + (14*(12/11.0592*10^6))=0.115 \text{ mS}$$

ดังนั้น อัตราการแปลงค่าน้อยที่สุดใน 1 วินาที มีค่าเท่ากับ

$$1/115*10^{-3} = 8681 \text{ ระดับ ต่อ วินาที}$$

บทที่ 4

วงจรควบคุมการทำงาน

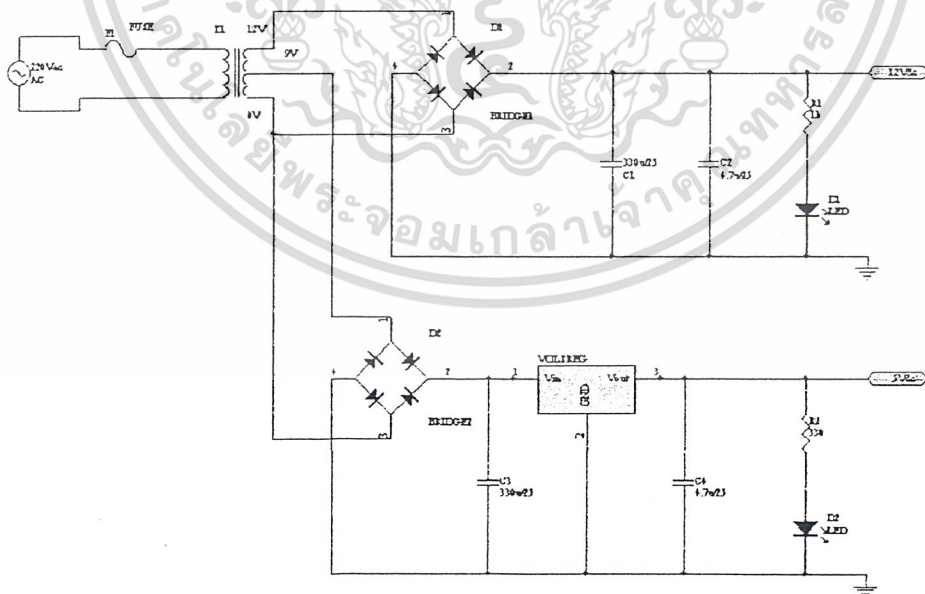
4.1 วงจรจ่ายไฟกระแสตรง (DC Power Supply Circuit)

จากรูปวงจรแหล่งจ่ายไฟตรงที่ 4.1 ไฟฟ้ากระแสสลับ 220 Volts 50 Hz ผ่านหม้อแปลงไฟฟ้า ซึ่งมี

- ขด Primary 0 V 220 V
- ขด Secondary 0 V 9V 12 V

ขด Secondary 12 V ผ่านวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์ (Full wave Bridge Rectifier) Bridge1 ผ่านตัวเก็บประจุ C1 3300uF/25V และ C2 4.7uF/25V เพื่อลดแรงดันกระเพื่อม ได้แรงดันไฟตรง 12 Volts DC เพื่อจ่ายให้แก่วงจรต่างๆ ส่วน LED D1 เป็นส่วนแสดงผล

ขด Secondary 9 V ผ่านวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์ (Full wave Bridge Rectifier) Bridge2 ผ่าน C3 3300uF/25V เพื่อลดแรงดันกระเพื่อม เข้าไปยัง 7805 regulator ที่ขา 1 และออกขา 3 ได้ Voltage ออกมา 5 Volts DC เพื่อจ่ายแก่วงจรต่างๆ LED D2 เป็นส่วนแสดงผลเช่นกัน



รูปที่ 4.1 แสดงวงจรแหล่งจ่ายไฟกระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 วงจรรับเสียง (Sound Detected Circuit)

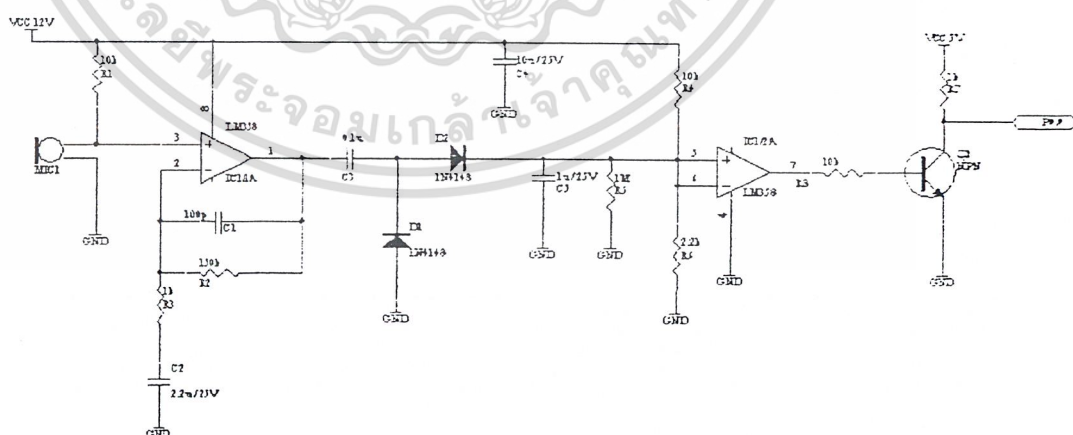
จากรูปวงจรรับเสียงที่ 4.2 เมื่อเสียงเดินทางเข้าสู่คอนเด็นเซอร์ ไมโครโฟน MIC1 รับสัญญาณเสียงได้ ก็ส่งสัญญาณเสียงที่รับได้เข้าขา 3 ของ LM358 IC1/1 เป็นวงจรขยายแบบ Non - inverting (วงจรขยายแบบไม่กลับเฟส) มี C1 เพื่อช่วยในการตอบสนองความถี่สูง

สัญญาณที่ถูกขยายจะถูกเรียงเป็นไฟตรงด้วยไดโอด D1 และ D2 (1N4148) สัญญาณไฟลบจะผ่าน D1 ลง GND ช่วงบวกผ่าน D2 เข้าขา 5 ของ IC1/2 คือเป็นวงจรเปรียบเทียบแรงดัน

ระหว่าง D2 ถึง ขา 5 ของ IC1/2 ผ่าน C5 กับ R5 ทำการหน่วงสัญญาณที่ออกทางขา 7 ของ IC จะหน่วงเวลาไว้ประมาณ 1 วินาทีเท่านั้น

Output ที่เข้าขา 5(+) ของ IC1/2 จะถูกเปรียบเทียบกับแรงดันที่เข้าขา 6(-) ซึ่งมีประมาณ 2 Volts ถ้าขา 5 มีแรงดันมากกว่าขา 6 แล้วขา 7 (output) จะมีแรงดันออกมา ซึ่งจะเป็น bias ให้แก่ transistor Q1 ทำงาน ส่งผลให้มีแรงดันออกไปที่ output ของวงจรซึ่งต่อกับ Port 0.0 ของ 89C51

แต่ถ้าขา 5 มีแรงดันน้อยกว่าขา 6 (ไม่มีเสียงมาหรือเสียงต่ำมาก) ขา 7 จะไม่มี output ออกมาเลย ทำให้ Q1 ทำงานและไม่มีแรงดันออกไปที่ output ด้วย



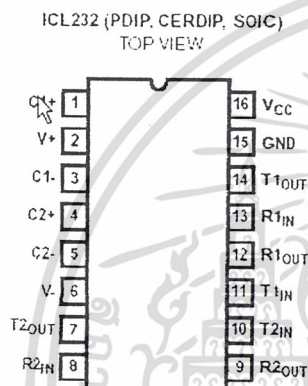
รูปที่ 4.2 แสดงวงจรรับเสียงจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

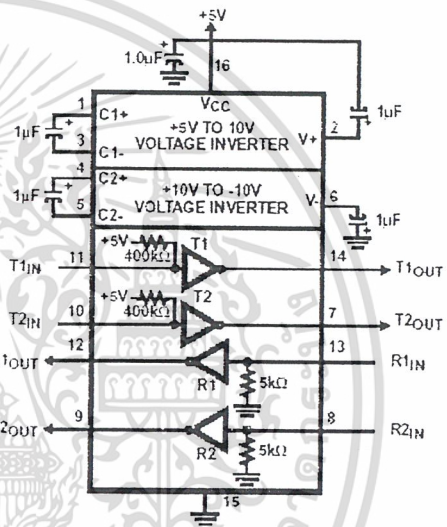
4.3 วงจรเชื่อมต่อคอมพิวเตอร์ผ่านพอร์ตอนุกรม (Serial Port Interface Circuit)

IC ที่นิยมใช้กันในการแปลงแรงดันจากมาตรฐาน RS-232 เป็น TTL,CMOS Logic คงจะหนีไม่พ้น 2 ตัวนี้ คือ MAX232 และ ICL232 สำหรับในที่นี้จะขออธิบาย ICL232 อย่างเดียว เพราะการใช้งานเหมือนกัน

Pinout



Functional Diagram

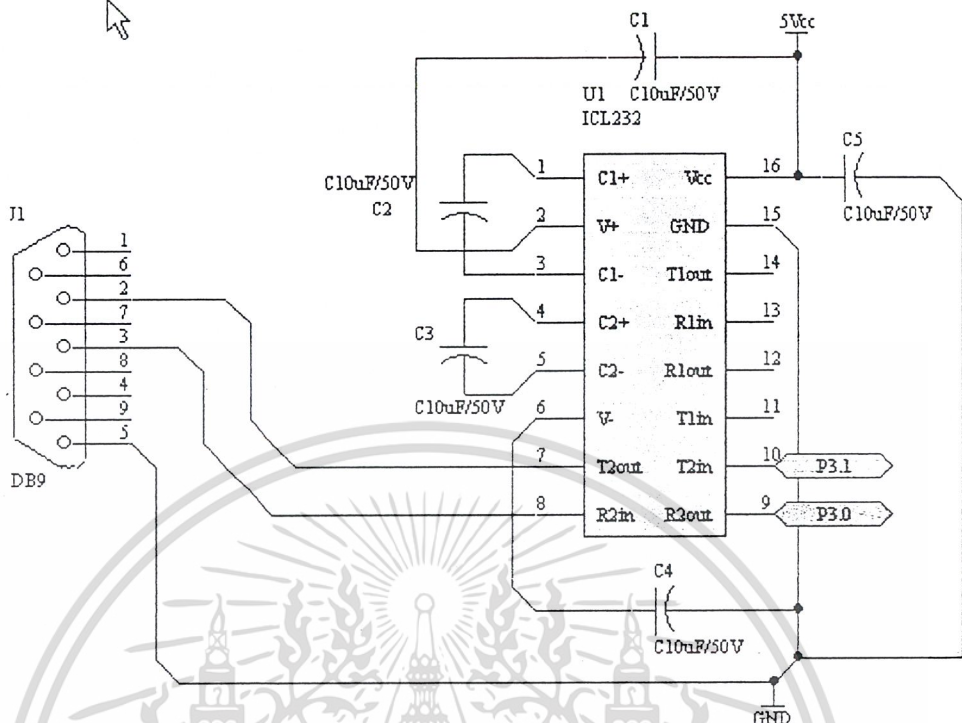


รูปที่ 4.3 แสดงขาและองค์ประกอบของ ICL232

คุณลักษณะเด่น

- ใช้แหล่งจ่ายไฟเดียว 5V
- มี Voltage Doubler, Inverter ในตัว
- Driver 2 ตัว, Receiver 2 ตัว
- Slew Rate 30V/uS
- ใช้ได้ทั้งระดับสัญญาณแบบ TTL, CMOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงวงจรเชื่อมต่อคอมพิวเตอร์ผ่านพอร์ตอนุกรม

จากทฤษฎีเรื่อง พอร์ตอนุกรม ในบทที่ก่อน เราใช้ ICL232 ทำหน้าที่ติดต่อกับ Computer ผ่าน พอร์ตอนุกรม RS-232 จากรูปวงจรที่ 4.4 จาก พอร์ตอนุกรม RS-232 แบบ DB-9 ใช้เพียง 3 ขา คือ

- ขา 2 (RxD) ต่อเข้ากับขา 7 (T2out) ของ ICL232 เพื่อรับค่าจาก ICL232
- ขา 3 (TxD) ต่อเข้ากับขา 8 (R2in) ของ ICL232 เพื่อส่งค่าออกไปยัง ICL232
- ขา 5 GND ต่อ GND ของวงจร

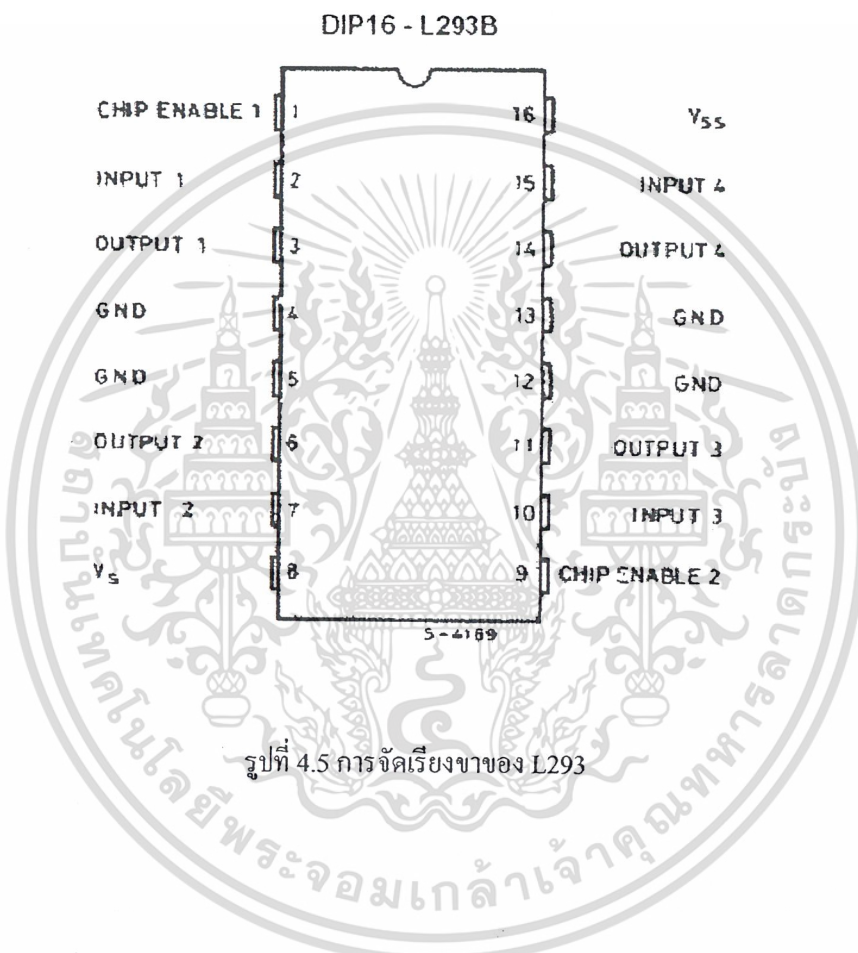
ส่วน ICL232

- ขา 9 (R2out) ไปต่อเข้ากับ ขา 10 ของ 89C51 (Port 3.0, RxD) เพื่อส่งค่าออกไปยัง 89C51
- ขา 10 (T2in) ไปต่อเข้ากับขา 11 ของ 89C51 (Port 3.1, TxD) เพื่อรับค่าจาก 89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

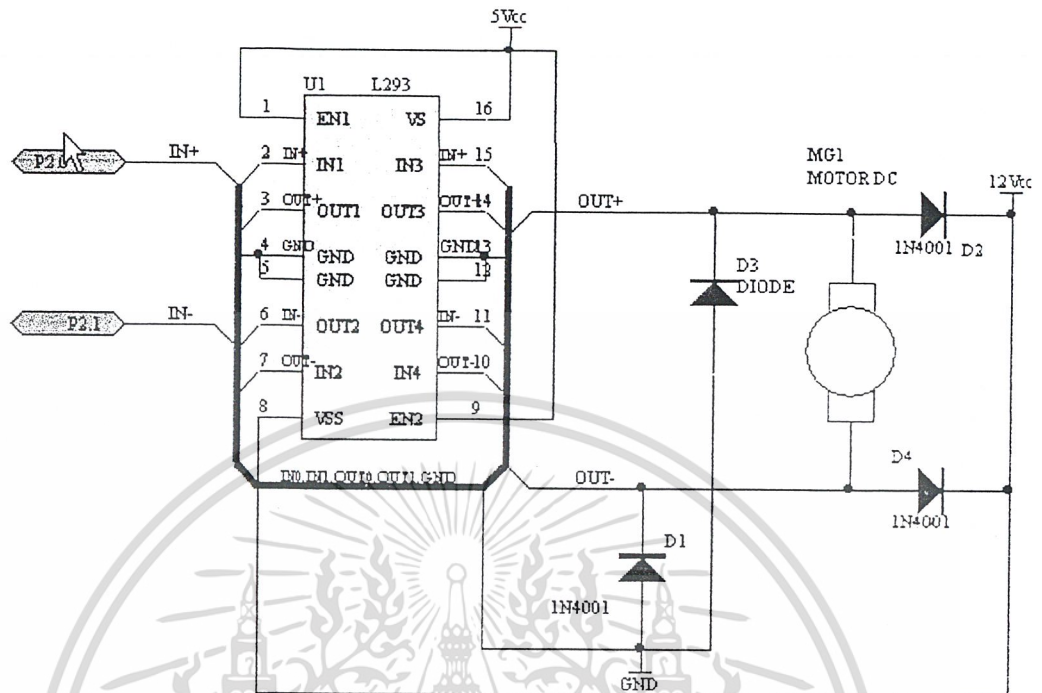
4.4 วงจรขับเคลื่อนมอเตอร์กระแสตรง (DC Motor Drive Circuits)

IC L293 เป็นอีกหนึ่ง IC ที่มีผู้นิยมใช้กันมาก ในการใช้ขับเคลื่อน DC Motor ซึ่งมีคุณลักษณะเด่น ดังนี้



- IC หนึ่งตัวสามารถขับเคลื่อน Motor แบบสองทางได้ถึง 2 ตัว
- ในแต่ละตัวสามารถขับกระแสได้สูงถึง 1 A
- กระแสสูงสุด (Peak Current) ขับได้ถึง 2 A ในแต่ละตัว
- มีระบบป้องกันอุณหภูมิที่สูงเกิน
- แยกระดับ Logic Supply ต่างหาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 วงจร Drive Motor โดยใช้ IC L293

จากรูปวงจร จะสังเกตเห็นว่าการต่อ INPUT1 กับ INPUT4 และ INPUT2 กับ INPUT3 พร้อมทั้งต่อ OUTPUT1 กับ OUTPUT4 และ OUTPUT2 กับ OUTPUT3 ซึ่งจะเป็นการต่อขานาน วงจร Drive Motor ทั้งสองของ L293 เข้าด้วยกัน เพื่อให้สามารถจ่ายกระแสได้ถึง 2 A และรับ Peak Current ได้ถึง 4 A.

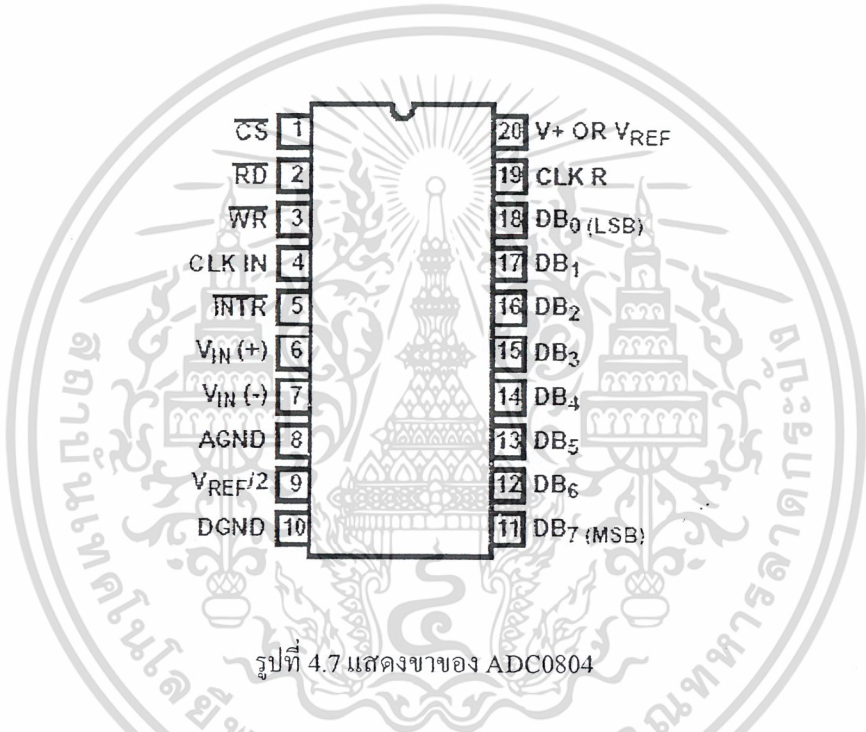
ในส่วนของ Chip Enable 1, Chip Enable 2 ต่อเข้ากับ 5 Vcc เพื่อให้ Enable การทำงานของวงจร Drive ทั้งสองให้ทำงานตลอดเวลา ส่วนการควบคุมการทำงานของ L293 ทำได้โดยการป้อน Logic ที่ต่างกันเข้าที่ IN+ (INPUT1 กับ INPUT4) และ IN- (INPUT2 กับ INPUT3) Motor ก็ทำงาน ยกตัวอย่างเช่น ให้ (IN+) = Logic 1 และ (IN-) = Logic 0, Motor จะหมุนไปทางซ้าย และถ้าให้ (IN+) = Logic 0 และ (IN-) = Logic 1, Motor จะหมุนไปทางขวา และถ้าให้ Logic (IN+) = Logic 0 และ (IN-) = Logic 0 หรือ (IN+) = Logic 1 และ (IN-) = Logic 1, Motor ก็หยุดทำงาน

ส่วน Diode ทั้งสี่ตัวมีไว้เพื่อป้องกัน Back EMF จาก Motor นั้นเอง

4.5 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (Analog to Digital Converter Circuit (ADC)

สิ่งสำคัญอีกสิ่งหนึ่งซึ่งจะทำให้เราสามารถทราบได้ว่า ในขณะที่กล่องหมูนอยู่ที่ตำแหน่งใดแล้ว ก็คือการ Feedback ด้วย VR ปรับค่าได้ แต่สัญญาณที่ได้จากการ Feedback นี้เป็นสัญญาณ Analog ซึ่ง Microcontroller ไม่สามารถนำไปประมวลผลได้ จึงต้องอาศัย IC ประเภท ADC (Analog to Digital Converter) เข้ามาช่วย

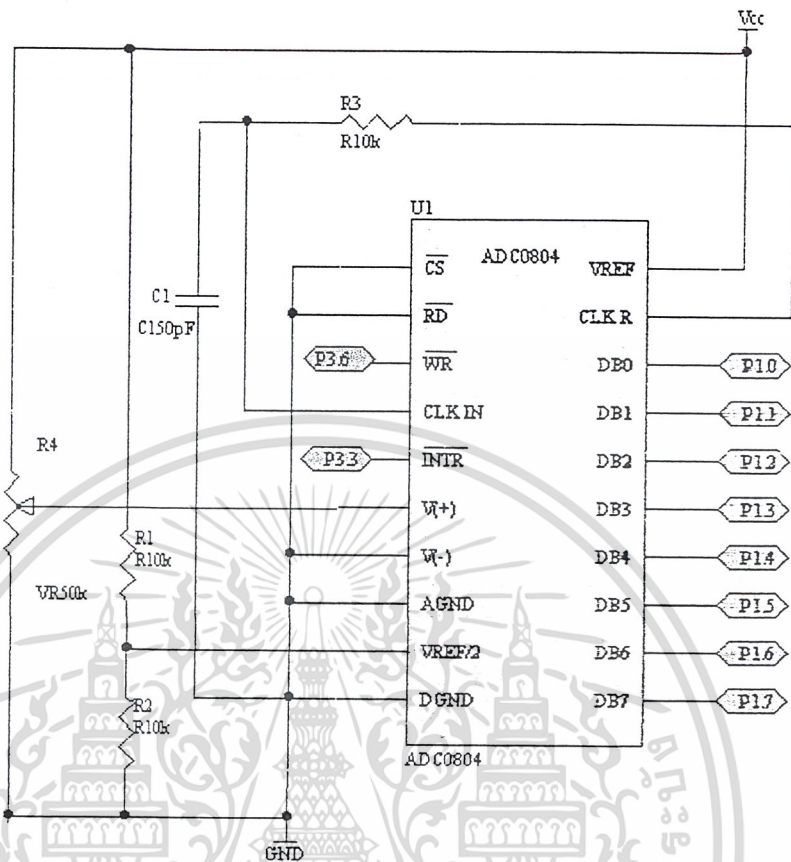
สำหรับในโครงการนี้ได้เลือก ADC0804 ซึ่งเป็น ADC ประเภท Successive-Approximation มาใช้โดยมีคุณสมบัติเด่นดังนี้



รูปที่ 4.7 แสดงขาของ ADC0804

- สามารถเชื่อมต่อกับ Microprocessor ชนิดต่างๆได้ง่าย
- ไม่ต้องเปลี่ยนแปลงระดับสัญญาณในการ Interface
- เหมาะกับระดับสัญญาณแบบ TTL
- ไม่ต้องปรับค่าศูนย์
- ใช้ไฟเลี้ยงเดี่ยวเพียง 5 V
- มีวงจรสร้าง Clock ในตัว
- ใช้เวลาในการแปลงข้อมูล (Conversion Time) < 100uS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 วงจรของ ADC0804

จากรูปวงจร อุปกรณ์ที่จำเป็นสำหรับ ADC0804 นี้ คือ C150 pF กับ R10 kOhm ต่อเข้ากับขา 4 (CLK IN) และ ขา19 (CLK R) เพื่อใช้สร้างฐานเวลา (Clock) เพื่อใช้ในการแปลงสัญญาณ และค่า R 10 kohm 2 ตัว ต่อแบ่งแรงดันครึ่งหนึ่ง ซึ่งจะได้ Voltage ที่ขา 9 (Vref/2) เท่ากับ 2.5 V ใช้เป็นค่ากลางในการเปรียบเทียบสำหรับการแปลงระดับสัญญาณ

ส่วนขา 2 (\overline{RD}) ต่อลง Ground เพื่อทำงานในโหมดต่อเนื่อง (Continuous Conversion) ที่ขา 6 (V+) ต่อแบ่งแรงดันจาก ไฟเลี้ยง 5 Vcc กับ VR 50 kOhm ซึ่งต่อทลเพื่ออยู่กับ Motor ทำการ Feedback ค่า Analog Voltage กลับมาให้ ADC ทำการแปลงค่า

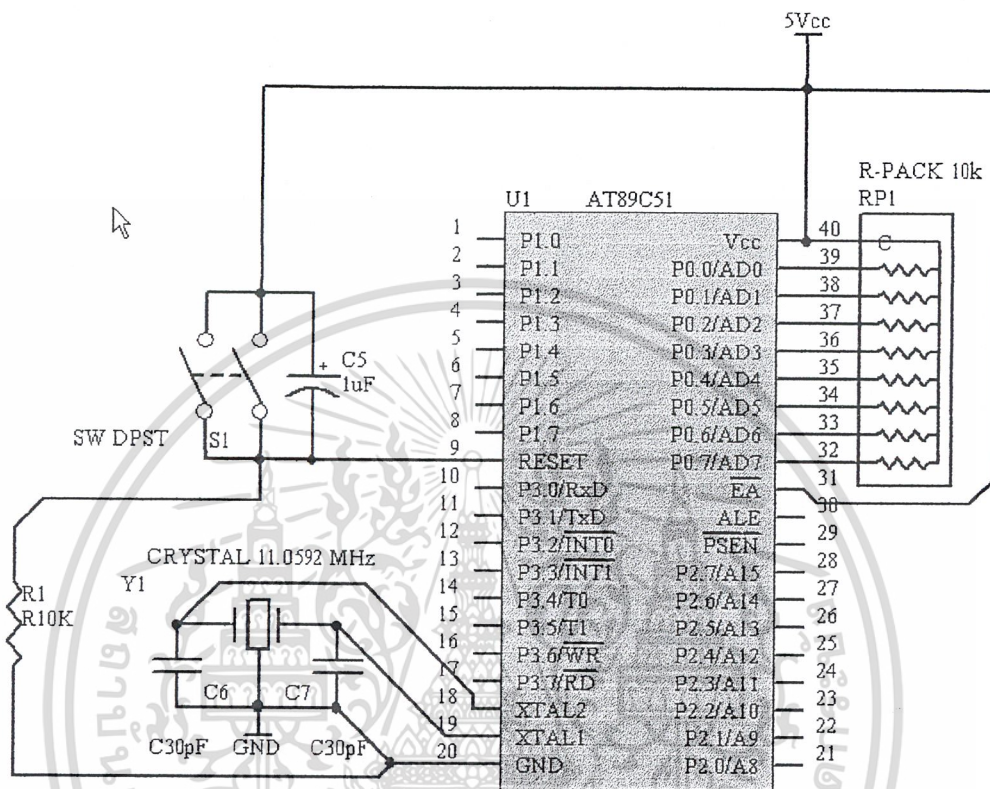
ขา 11-18 (DB0-DB7) เป็นค่า Digital Output ซึ่งต่อเข้ากับ Port 1 ของ Microcontroller พร้อมกับมี LED แสดงผล(ดูรูปวงจรรวม)

ขา 3 (\overline{WR}) ต่อกับ P3.6 ของ Microcontroller เพื่อใช้ป้อน Pulse ให้ ADC เริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 วงจรไมโครคอนโทรลเลอร์และวงจรรวม (Microcontroller and Mix Circuits)

สำหรับในรูปที่ 4.10 จะแสดงส่วนสำคัญและอุปกรณ์ที่จำเป็น สำหรับการใช้งาน Microcontroller MCS-51



รูปที่ 4.10 วงจร Microcontroller MCS-51

ที่ขา 9 (Reset) ของ MCS-51 จะต่ออยู่กับ R และ C เพื่อทำให้เกิดการ Power Up Reset นั่นคือ ทุกครั้งที่เริ่มจ่ายไฟ Microcontroller จะ Reset เองทุกครั้ง เพราะเมื่อขา 9 ได้รับ Logic 1 (5V) เป็นเวลาไม่น้อยกว่า 2 แมกซ์วินไซเคิล Microcontroller จะ Reset นี่จึงเป็นเหตุผลที่ต้องใช้ R,C เพื่อหน่วงเวลาไว้ และเมื่อต้องการ Reset เองก็สามารถทำได้โดยการกดปุ่ม Reset ซึ่งใช้ Switch แบบกดคิดปลอยคับนั่นเอง

ที่ขา 18 และ 19 (Xtal และ Xtal2) ต่อเข้า Crystal 11.0592 MHz และ C 30 pF สองตัว เพื่อสร้างสัญญาณนาฬิกา(Clock)ให้แก่ Microcontroller

ที่ขา 31 (\overline{EA}) ใช้เลือกจะใช้หน่วยความจำภายในหรือภายนอก ในที่นี้เลือกใช้หน่วยความจำภายใน จึงต่อขา \overline{EA} กับ 5 V

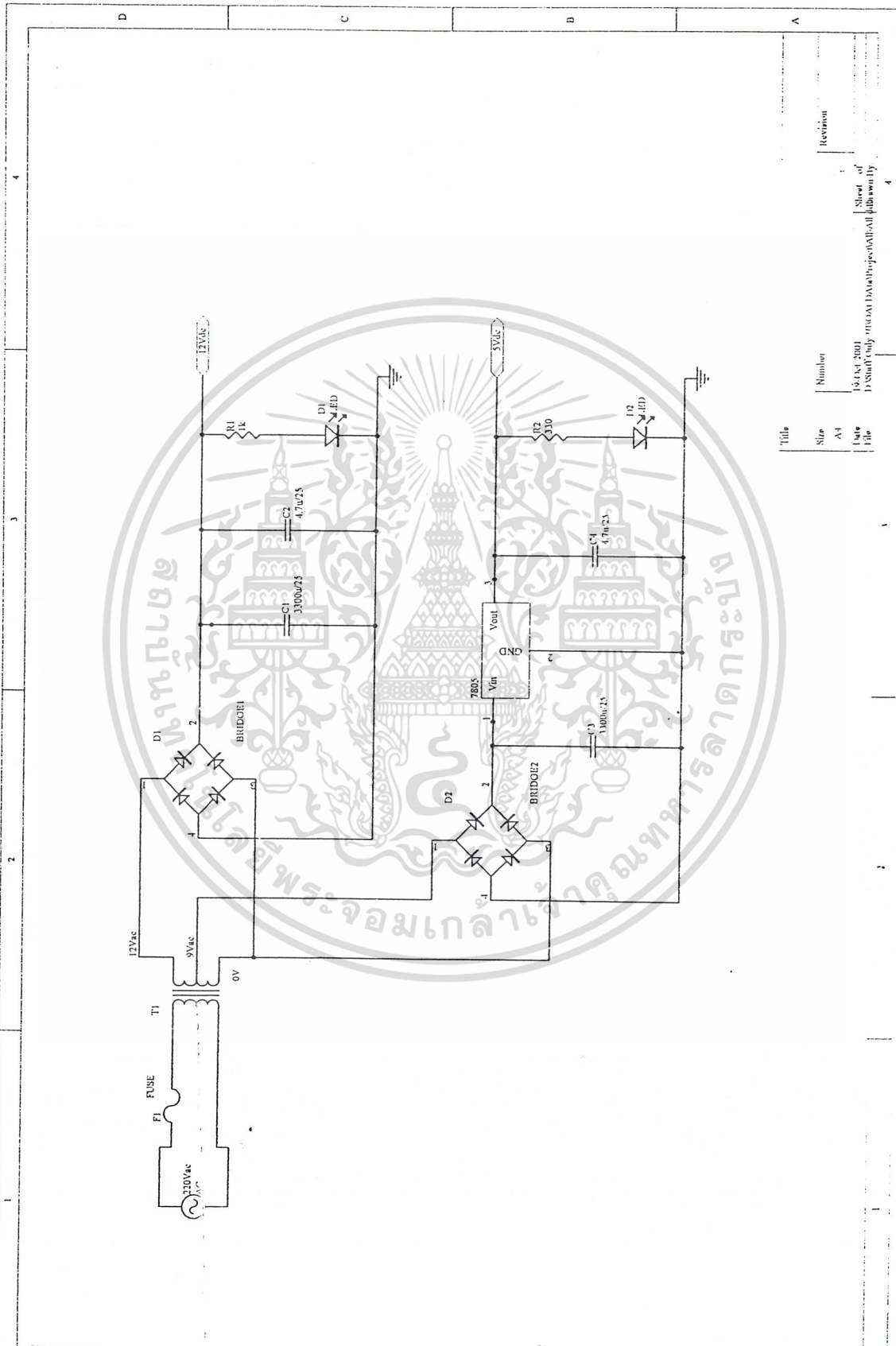
ที่ Port 0 (ขา 32-39) เป็น Port ที่แตกต่างจาก Port อื่น เพราะสามารถรับ Input ได้อย่างเดียว ถ้าต้องการให้เป็น Port Output ด้วย ต้องต่อ R-Pull Up เอาไว้

ขา 40 และ 20 เป็นขาไฟเลี้ยงกับ Ground

และในหน้าถัดไปจะแสดงรูปวงจรทั้งหมด

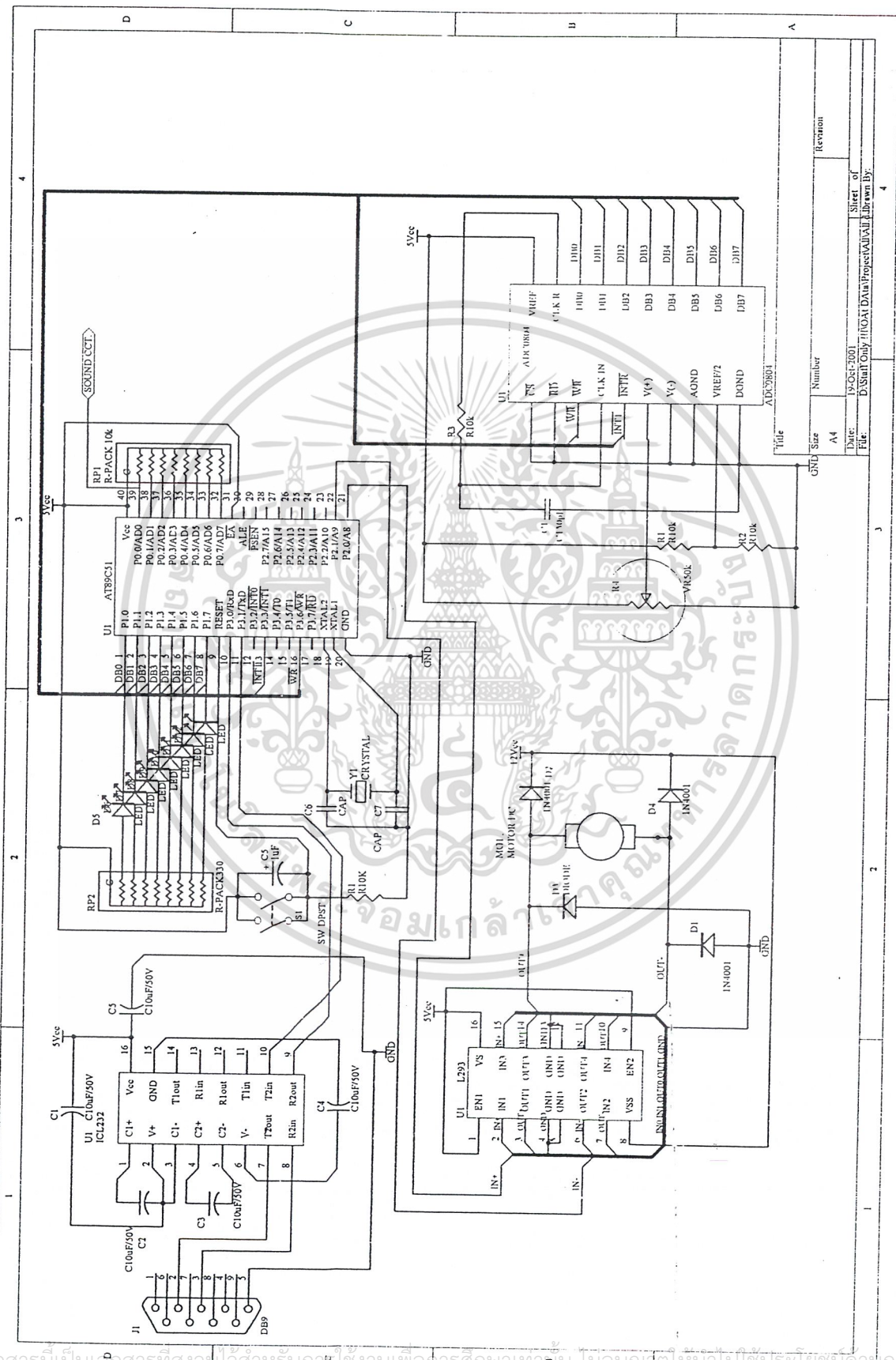


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title	Size	Number	Revision
	A4	1	
	Date	16-Oct-2001	
	File	D:\Staff Only\mcom\DA\mcom\poc\ball\lab\work\...	
		Sheet of	4
		D Sheet Only	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Sheet of	4
Drawn by	D'Silva Only
Checked by	
Date	12/01/2001
Rev	
Number	A4
Size	GND
Title	ADS-2001

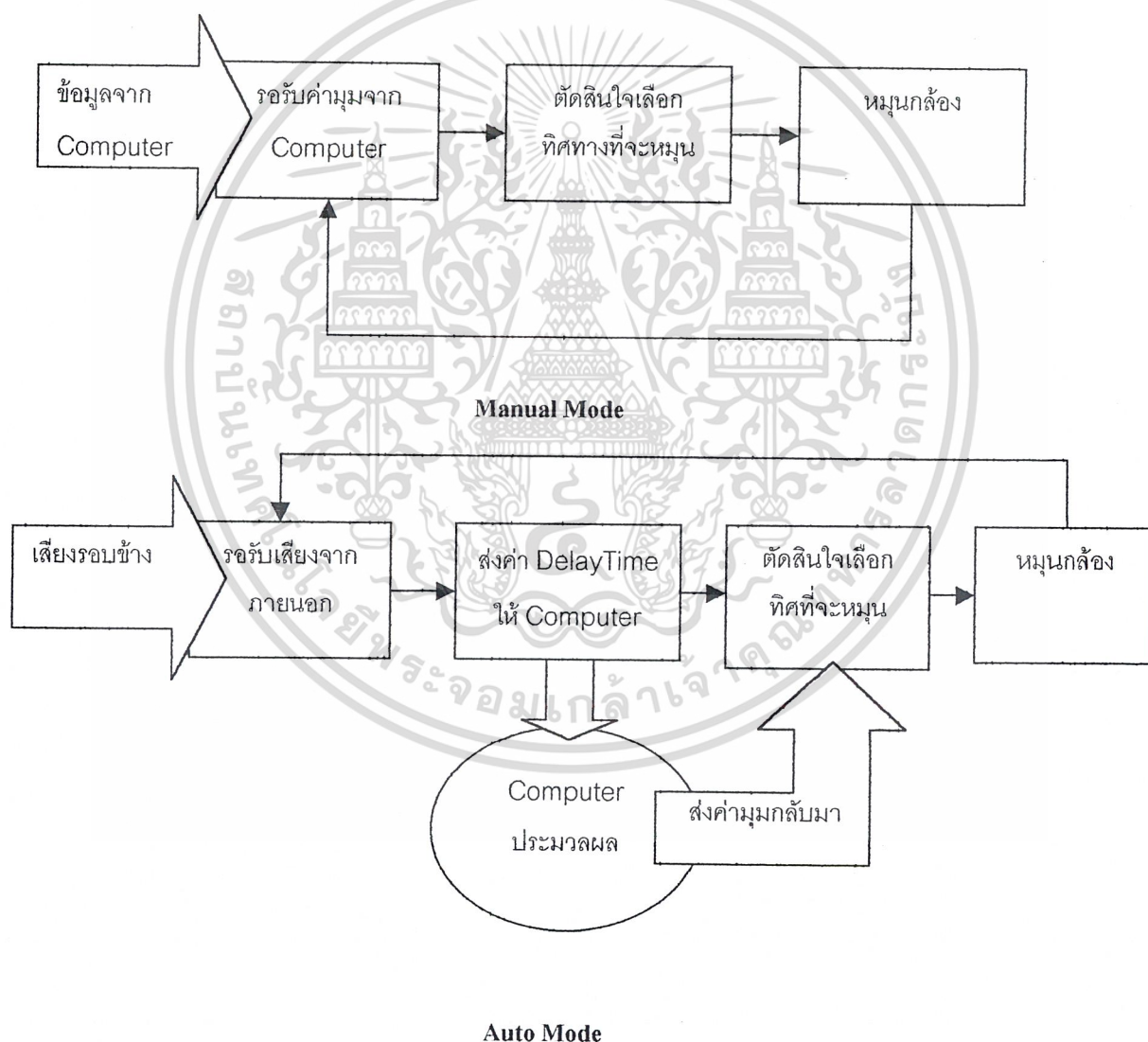
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคคลใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

โปรแกรม MCS-51

5.1 หลักการทำงานพื้นฐาน (Basic Concept)

การทำงานของหุ่นยนต์ จะแบ่งออกเป็นสองโหมด คือ Mode1 (Manual) และ Mode2 (Auto) โดยใน Manual Mode, Microcontroller จะรอรับค่ามุมจาก Computer ก่อน แล้วจึงนำค่าที่ได้ไปประมวลผล ส่วน Auto Mode จะรอรับสัญญาณเสียงเข้ามาก่อน Microcontroller จะทำการวัดค่า Delay Time ก่อน แล้วจึงส่งค่า Delay นี้ไปให้คอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ใช้ Fuzzy Logic ทำการประมวลผล แล้วจึงส่งค่ามุมที่คำนวณได้กลับมา เพื่อให้ Microcontroller ตัดสินใจหมุนกล่องต่อไป



รูปที่ 5.1 โหมดพื้นฐานในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 โปรแกรมหลัก (Main Program)

เริ่มต้นกันด้วยการประกาศค่า Register ที่ต้องการใช้งาน โดยให้ ADC เป็น Register ใช้เก็บค่า Digital Voltage จาก VR ที่หัดเพื่อต่อกับ Motor โดยรับมาจาก ADC ทาง Port1

ส่วน Register VBDATA ใช้เก็บค่ามุมที่ Computer ส่งมาทาง Serial Port เพื่อใช้เปรียบเทียบกับ มุมปัจจุบันของ Register ADC

Register DEY1 และ DEY11 เป็น Register ที่ใช้เก็บค่า Delay ที่วัดได้จาก Timer0 ซึ่งตั้งค่าเป็นแบบ 16 Bits ดังนั้น ค่าที่สามารถวัดได้จึงมีค่าตั้งแต่ 0-65536 ระดับ ซึ่งถ้าค่ามากกว่าจะถือว่า Delay Error ไม่สามารถวัดได้ ซึ่งจะอธิบายรายละเอียดในหัวข้อถัดไป

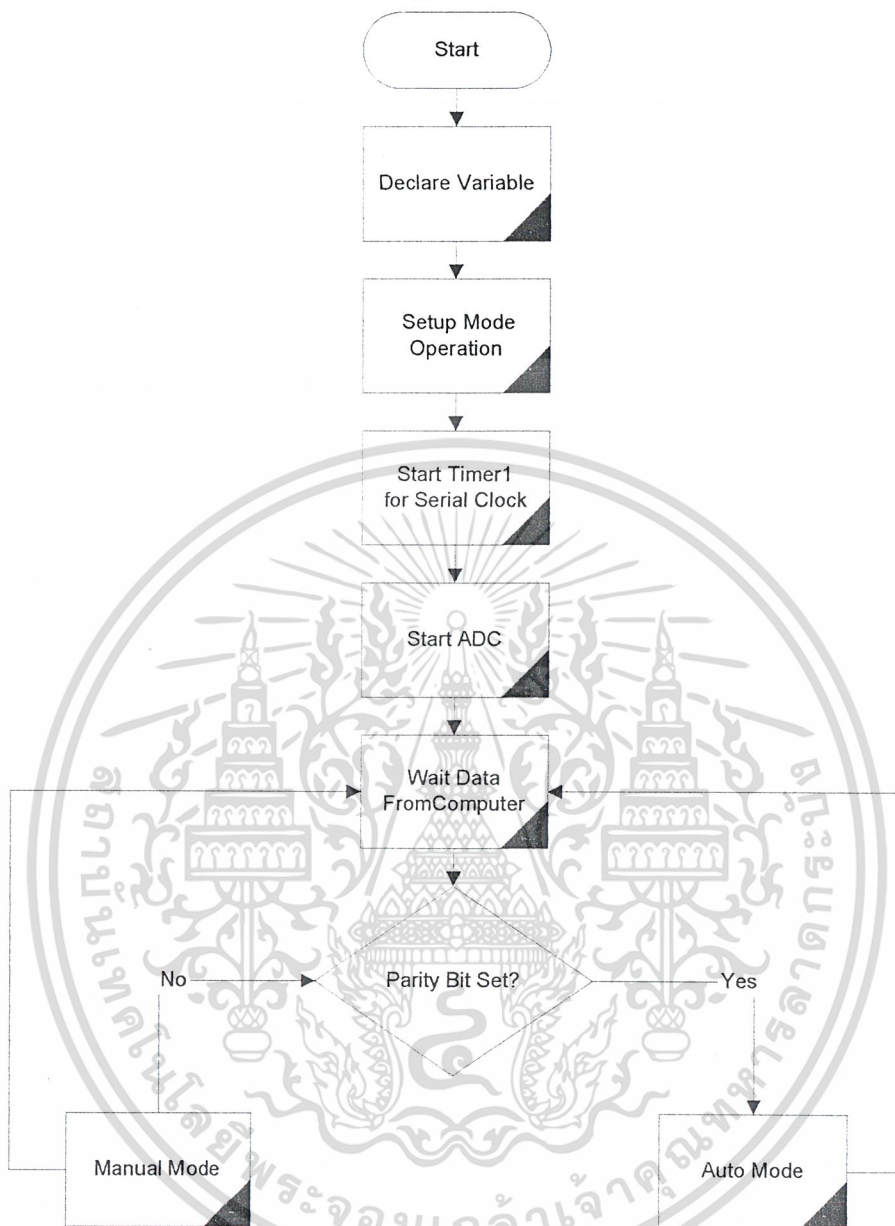
Enable Interrupt ต่างๆที่ต้องการใช้งาน ในที่นี้ใช้ $\overline{INT1}$ ควบคุมการทำงานของ ADC หลังจากนั้นตั้งค่า Timer โดยใช้ Timer1 ในโหมด 2 Auto Reload ใช้กำหนด Baud Rate ซึ่งตั้งไว้ที่ 9600 bit/sec ส่วน Timer0 เป็นแบบ 16 Bits ใช้นับ Delay

จากนั้นจึงเริ่มทำงาน โดยสั่งให้ Timer1 ของ Microcontroller เริ่มนับ เพื่อกำหนดฐานเวลาให้การติดต่อสื่อสารแบบอนุกรม และส่ง Start Pulse ให้ ADC, ADC จะส่งสัญญาณ Interrupt Microcontroller และเก็บค่าของมุมปัจจุบันเอาไว้ใน Register ADC ตลอดเวลา

รอรับค่าจาก Computer และดูว่า Parity Bit Set หรือไม่ (ใน VB จะสั่งให้ Parity Bit Set หรือไม่ Set อย่างอิสระ เพื่อใช้กำหนด Mode) ถ้า ไม่ Set จะเป็น Mode1 (Manual), ถ้า Set จะเป็น Mode2 (Auto) ส่วนข้อมูล 8 Bit จาก Computer จะเก็บไว้ใน Register VBDATA เมื่อทำงานในแต่ละ Mode เสร็จ MCS-51 จะกลับมารับค่า จากคอมพิวเตอร์อีกครึ่งหนึ่ง

สามารถเขียนเป็น Flow Chart ได้ในหน้าถัดไป

Main Program Flow Chart



รูปที่ 5.2 Flow Chart ของ Main Program

5.3 Manual Mode

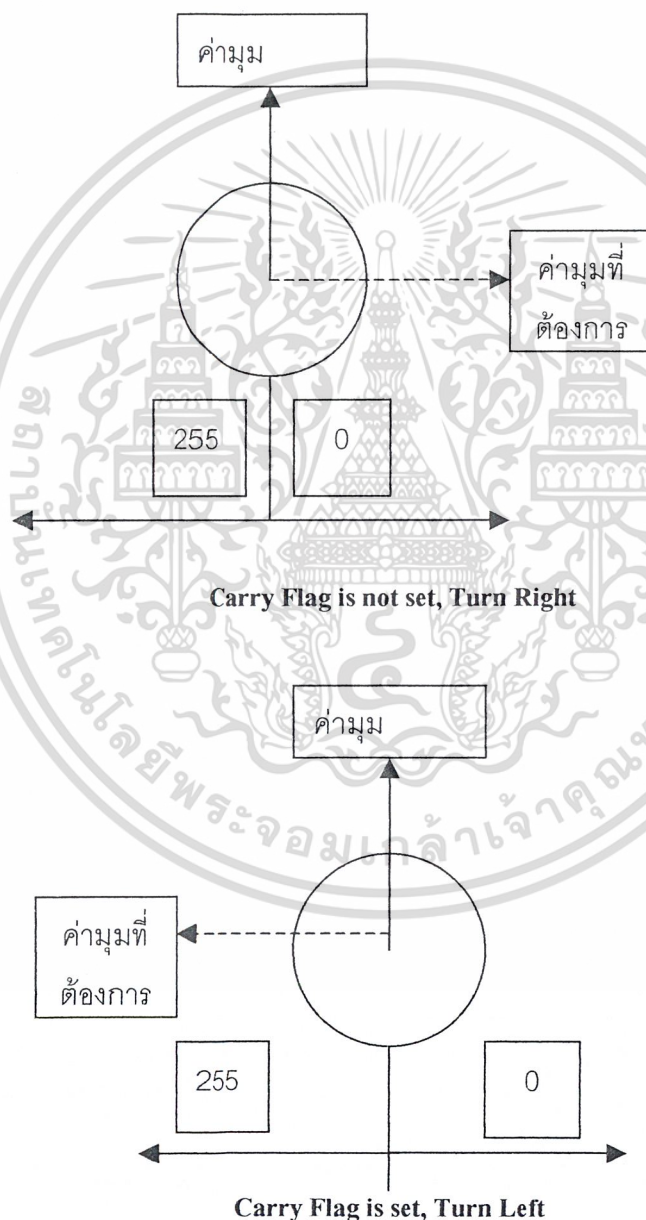
ใน Manual Mode เมื่อตรวจค่าที่รับมาจาก Computer ตรวจสอบว่าค่าที่รับมาว่า Parity Bit Set หรือเปล่า ถ้าไม่ Set จะเข้ามาทำงานใน Manual Mode ซึ่ง Computer จะส่งค่าองศาที่ต้องการกลับมา ซึ่ง MCS-51 จะทำการคำนวณทิศทางที่ใกล้ที่สุดก่อน(อธิบายในหัวข้อถัดไป)แล้วจึงสั่งให้ Motor หมุน พร้อมทั้งเซตค่าตำแหน่งปัจจุบันจาก ADC และตรวจ Emergency Stop ไปในตัว จนค่า ADC มีค่าเท่ากับ VBDATA จึงสั่ง L293 ทาง P2.0,P2.1 ให้ Motor หยุดทำงานแล้วไปรอรับค่าจาก Computer ใหม่อีกครั้ง ซึ่งจะมีโปรแกรมย่อยที่น่าสนใจ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.1 โปรแกรมย่อยคำนวณทิศทางที่ใกล้สุดพร้อมสั่งหยุดเมื่อเกิดข้อผิดพลาด

(Calculate the Nearest Direction and Emergency Stop Program)

การค้นหาทิศทางที่ใกล้ที่สุดนั้น อาศัยคำสั่ง SUBB A, Direct Address โดยเอาค่า Register ADC (ค่ามุมปัจจุบัน) ใส่เข้าไปใน Accumulator เป็นตัวตั้ง ลบด้วย Register VBDATA (มุมที่ Computer ต้องการ) ถ้ามีค่าเท่ากัน ให้หยุดทันที แต่ถ้าค่าเป็นบวก Carry Flag ไม่ Set ก็สั่งให้ Motor หมุนซ้าย และถ้าค่าเป็นลบ Carry Flag จะ Set ก็สั่งให้ Motor หมุนไปทางขวา (ทางขวาสุด ค่า Digital Voltage จาก ADC มีค่าน้อยที่สุดคือ 0 และทางซ้ายมีค่ามากที่สุด 255)



รูปที่ 5.3 การตัดสินใจเลือกทิศทางในการหมุน

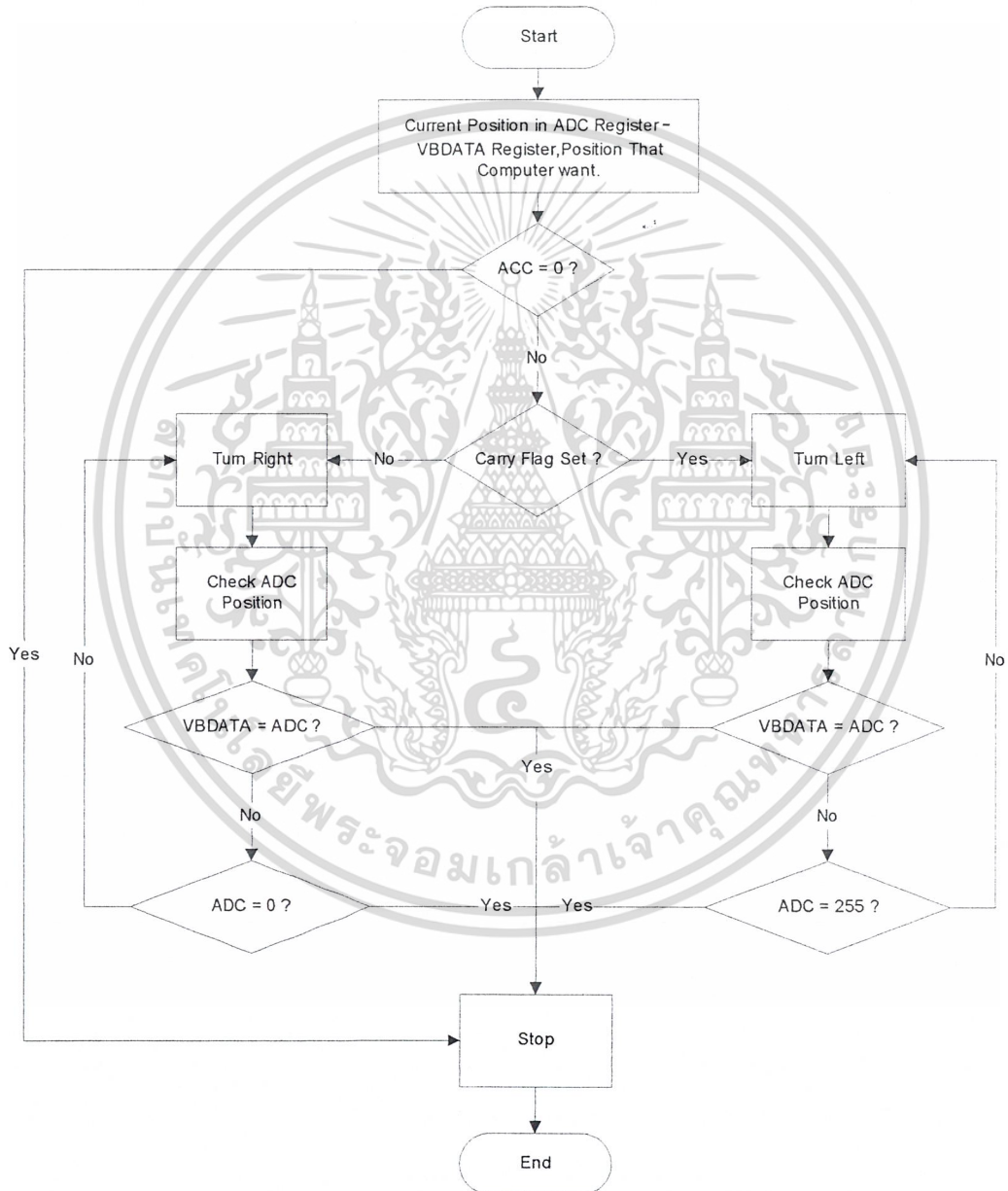
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตัดสินใจเลือกทิศทางในการหมุนได้แล้ว ก็สั่งให้ Motor หมุน โดยเช็คค่ามุมปัจจุบันจาก ADC เทียบกับมุมที่ต้องการใน VBDATA ถ้าเท่ากันเมื่อไหร่ก็หยุดหมุน

ในขณะที่หมุนและเช็ค ค่า ADC อยู่ นั้น จะเช็คด้วยว่าค่ามุมที่วัดได้นั้น ถึงพิกัดที่ ADC จะวัดได้ หรือยัง (0 กับ 255) ถ้าถึง นั้นหมายความว่า VR หมุนมาสุดแล้ว ต้องสั่งให้ Motor หยุดหมุนทันที ซึ่งถ้า หมุนขวาต้องตรวจสอบกับค่า 0 และหมุนซ้ายตรวจสอบกับค่า 255 ตามลำดับ

เขียนเป็น Flow Chart ได้ดังนี้

Calculate the Nearest Direction and Emergency Stop Program Flow Chart



รูปที่ 5.4 Flow Chart ของ โปรแกรมย่อยคำนวณทิศทางที่ใกล้สุดพร้อมสั่งหยุดเมื่อเกิดข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2 โปรแกรมย่อยเรียกค่าจาก ADC (ADC Loading Program)

โปรแกรมนี้จะใช้การ Interrupt เพื่อดึงค่าจาก ADC เข้ามาเก็บใน Register ADC โดยเขียนโดยเขียนโดยอ้างอิงจาก Timing Diagram ของ ADC0804

Timing Diagrams

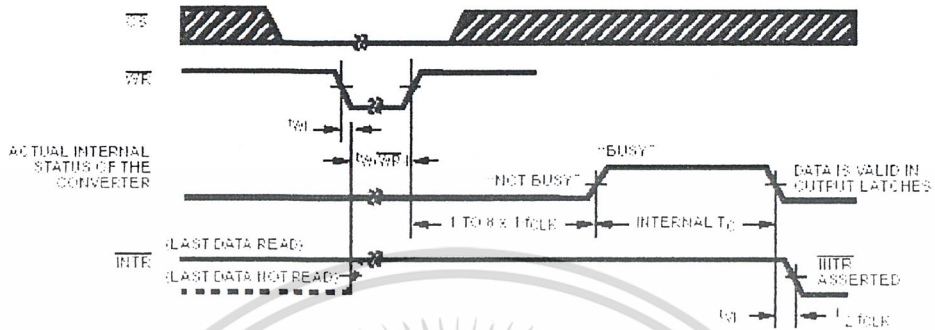


FIGURE 10A. START CONVERSION

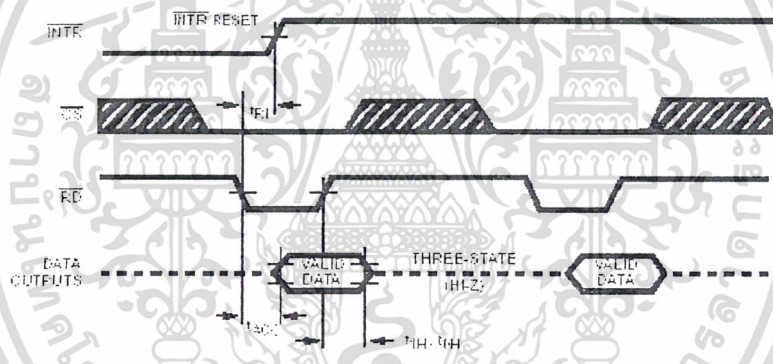


FIGURE 10B. OUTPUT ENABLE AND RESET INTR

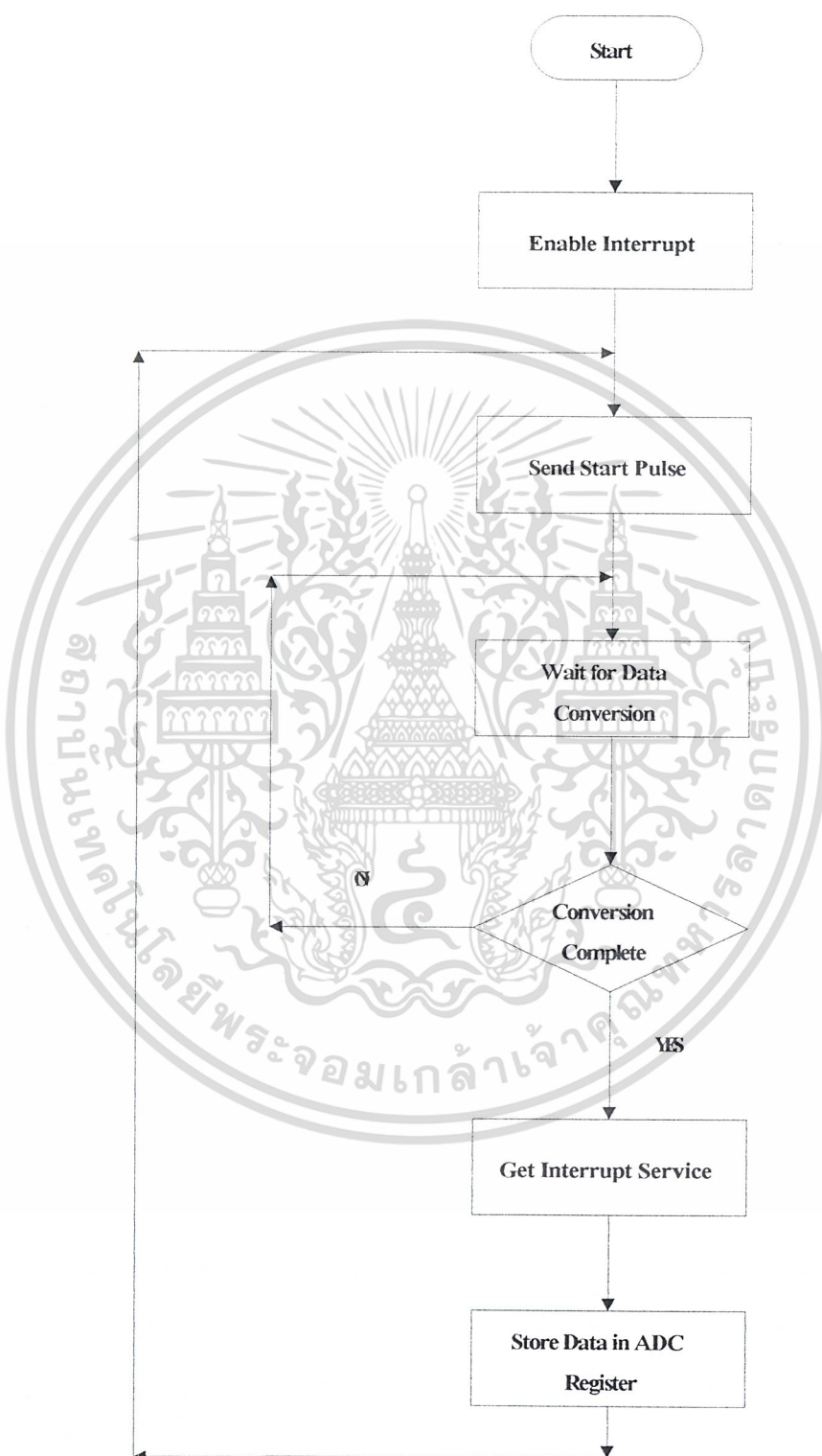
รูปที่ 5.5 Timing Diagrams ของ ADC0804

เมื่อดูจาก Timing Diagrams จะเห็นได้ว่า การที่จะสั่งให้ ADC ทำงานนั้น จะต้องทำให้ขา \overline{WR} มี Logic 0 เป็นระยะเวลาไม่น้อยกว่า $t_w(wr)$ ซึ่งมีค่าประมาณ 100 nS หลังจากนั้น ADC จะเริ่มทำการแปลงสัญญาณจาก Analog ให้เป็น Digital จนเสร็จสิ้น และส่งสัญญาณ End of conversion ออกมาที่ขา \overline{INTR} โดยมีค่าเป็น Logic 0 ในที่สุด

จาก Timing Diagrams นี้ ทำให้เราสามารถออกแบบการ Interface กับ MCS-51 ได้ โดยกำหนดขาใดขาหนึ่ง เป็นขาที่ใช้สั่งให้ ADC เริ่มทำงาน และต่อขา \overline{INTR} เข้ากับ $\overline{INT0}$ หรือ $\overline{INT1}$ ของ MCS-51 ก็ได้ สำหรับในโครงงานนี้เลือกใช้ P3.6 และ $\overline{INT1}$ ซึ่งเขียนเป็น Flow Chart ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart ADC Conversion



รูปที่ 5.6 Flow Chart ของโปรแกรมย่อยเรียกค่าจาก ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นด้วยการ Enable Interrupt $\overline{INT1}$ หลังจากนั้นที่ส่ง Pulse ซึ่ง Logic Low ไม่น้อยกว่า $t_w(wr)$ แล้วจึงทำงานต่อไปเรื่อยๆ โดยไม่ต้องรอ End of Conversion เพราะเมื่อ ADC Conversion เสร็จสิ้น จะส่ง Logic 0 มาที่ขา $\overline{INT1}$ ซึ่งเป็นการ Interrupt จากภายนอก ทำให้ Program กระโดดไปทำงานในส่วนของการบริการย่อย Interrupt จากนั้นเราจึงนำค่าที่ Port 1 เข้าไปเก็บที่ Register ADC ซึ่งได้ประกาศไว้ที่ตอนต้นของโปรแกรม (ดู Source Code MCS-51) แล้วจึงส่ง Start Pulse อีก เพื่อให้ ADC ทำงานอย่างต่อเนื่องก่อนออกจากส่วนของบริการย่อย Interrupt

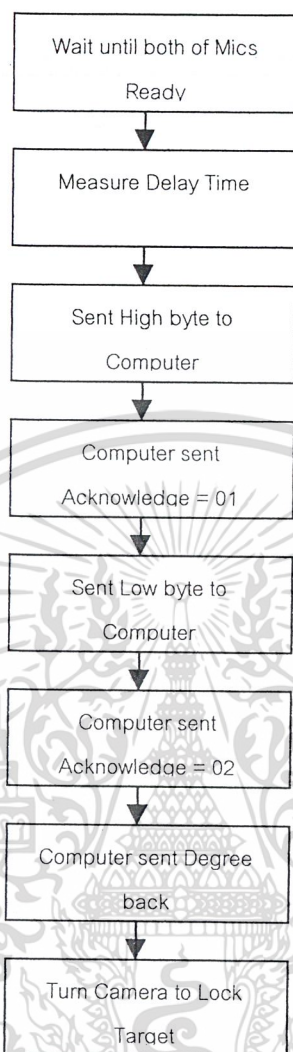
และเมื่อต้องการตรวจสอบค่าตำแหน่งของกล่องปัจจุบัน ก็สามารถทำได้ด้วยการตรวจสอบค่า ADC Register ซึ่งทุกครั้งที่ Interrupt ค่า ADC Register จะเปลี่ยนแปลงตามค่าตำแหน่งล่าสุดที่ VR หมุนไป ซึ่งการ Conversion จะเร็วมากเมื่อเทียบกับความเร็วในการหมุนกล่อง ทำให้เราสามารถทราบตำแหน่งปัจจุบันที่ถูกต้องและแม่นยำได้

5.4 Auto Mode

สำหรับ Auto Mode จะมีขั้นตอนที่ซับซ้อนกว่า Manual Mode เริ่มจากหลังจากรับค่าจาก Computer แล้วตรวจว่า Parity Bit Set รีเบิ้ล่า ถ้า Set จะเข้ามาทำงานใน Auto Mode ซึ่งใน Auto Mode จะทำการวัดค่า Delay จากไมโคร โฟนทั้งสองข้าง ซึ่งสิ่งที่ต้องคำนึงถึงก็คือไมโคร โฟนทั้งสองข้าง ต้องพร้อมที่จะรับเสียง กล่าวคือ ไมโคร โฟนทั้งสองข้างไม่ได้ยินเสียงอะไรเลย ในตอนแรก (Input จาก P2.6 และ P2.7 ต้องเท่ากับ 1 ทั้งคู่) ซึ่งถ้าไมโคร โฟนยังไม่พร้อม MCS-51 จะส่งค่า 0FFH ให้ Computer ตลอด จนเมื่อไมโคร โฟนพร้อมแล้ว จึงส่งค่า 00H กลับไป ซึ่ง Computer จะตอบรับ (Acknowledge) ด้วยการส่งค่า 00H กลับมา แล้วจึงเริ่มทำการวัด Delay ได้ โดยตั้งค่าเริ่มต้นของ TH0, TL0, DEY11, DEY1, Carry flag, Parity Bit (ที่จะส่ง) ให้เป็น 0 ให้หมด ทำการวัด Delay (โปรแกรมย่อยในการวัดจะอธิบายในหัวข้อย่อยต่อไป) เมื่อวัดเสร็จและไม่มี Error จะทำการส่งค่า DEY1 ซึ่งเป็นค่า Delay 8 Bits บน ไปก่อน จากนั้นรอรับสัญญาณ Acknowledge จาก Computer ซึ่งจะส่งค่า 01H กลับมาเพื่อบอกว่ารับข้อมูลที่ 1 ได้แล้ว จากนั้นก็ส่ง DEY11 ตามไป (ซึ่ง VB จะทำการคำนวณค่ารวมกันเอง) และ Computer จะตอบกลับ มาด้วยค่า 02H หมายความว่าได้รับข้อมูล 8 Bits ดังแล้ว จากนั้น VB จะทำการคำนวณค่ามุมที่ถูกต้อง แล้วส่งกลับมาให้ MCS-51 ซึ่งจะทำการเรียกโปรแกรมย่อยของ Manual Mode คือ ADC Loading Program และ Calculate the Nearest Direction and Emergency Stop Program เพื่อตรวจค่ามุมคำนวณทิศทางที่ใกล้ที่สุด และสั่ง Motor ให้หมุน ตรวจ Emergency Stop จนค่า ADC มีค่าเท่ากับ VB DATA ก็สั่ง Motor หยุดทำงานแล้วไปรอรับค่าจาก Computer ใหม่อีกครั้งหนึ่ง

Auto Mode เขียนเป็น Block Diagram ได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



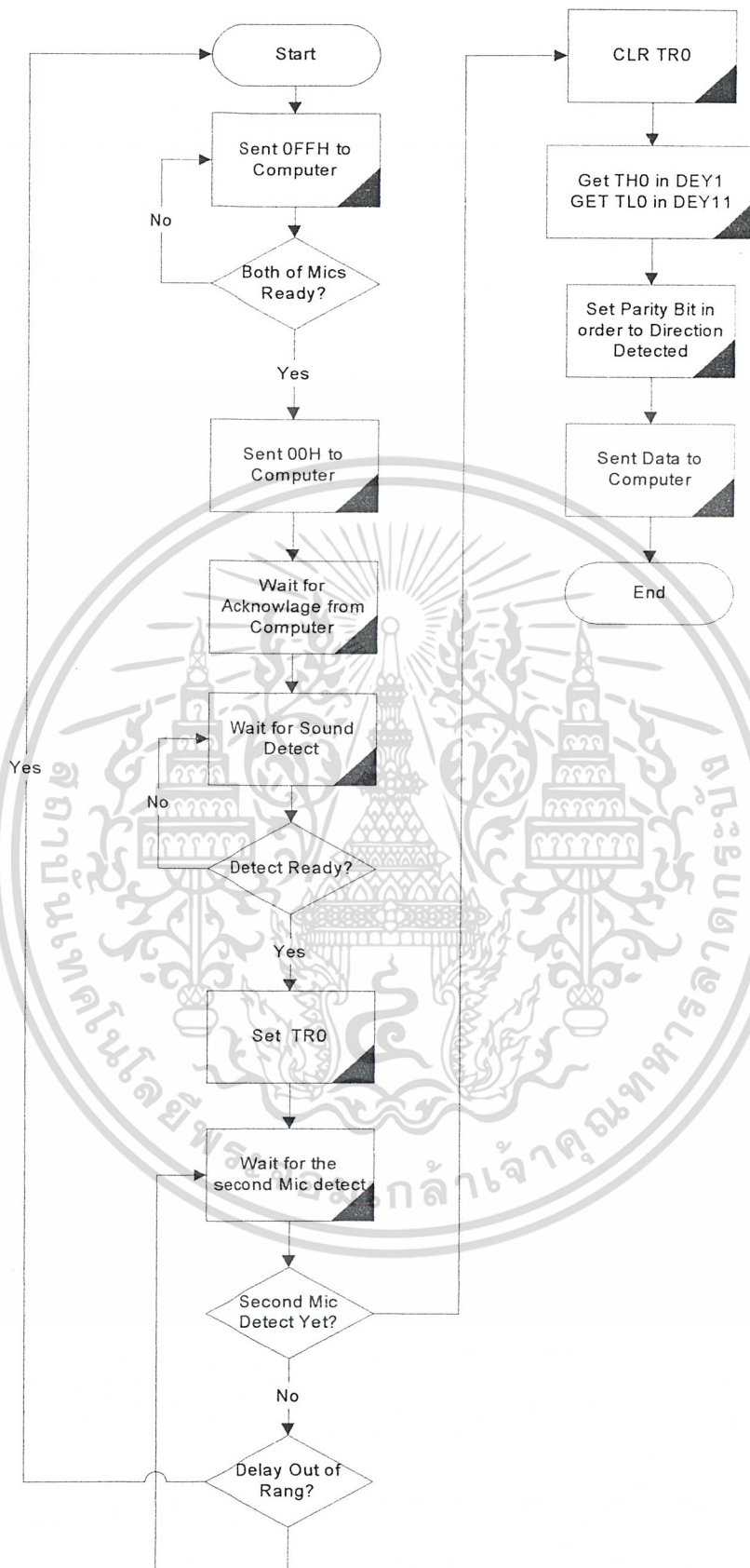
รูปที่ 5.7 Auto Mode Block Diagram

5.4.1 โปรแกรมย่อยวัดค่า Delay (Delay Counter Program)

หลังจากที่เข้าสู่ Auto Mode ,MCS-51 ต้องรอให้ไมโคร โฟนพร้อมรับเสียงก่อน ซึ่งหมายถึง ไมโคร โฟนต้องส่งค่า Input มาเป็น 1 ทั้งสองตัว จึงจะเริ่มนับ Delay ได้ ถ้ายังไม่พร้อม MCS-51 จะส่งค่า 0FFH ตลอดเวลา จนเมื่อพร้อมแล้ว จึงส่งค่า 00H ให้ Computer แล้ว Initial Register ที่เกี่ยวข้องทุกค่า จากนั้นจะรอรับค่าเสียง เมื่อรับไมโคร โฟนตัวใดตัวหนึ่งได้ โปรแกรมจะเริ่มนับ Delay ทันที และตรวจสอบว่าเมื่อไหร่ ไมโคร โฟนอีกตัวจะจับเสียงได้ ในช่วงนี้ หาก Timer0 นับเกินค่า 16 Bits จะเกิด Error ในการรับค่าขึ้น ก็จะส่งค่า 0FFH ไปให้ Computer และกลับไปรอรับเสียงใหม่ จนกระทั่ง นับได้ไม่เกิน 16 Bits จะตั้งหยุดนับ แล้วนำค่า TH0 ใส่ใน Register DEY1 และ TL0 ใส่ใน Register DEY11 และตรวจสอบว่าไมโคร โฟนด้านไหนตรวจได้ก่อนกัน ถ้าด้านซ้ายติดก่อน ให้ Set Parity Bit ก่อนส่ง แต่ถ้าด้านขวาติดก่อน ให้ Clear Parity Bit ก่อนแล้วจึงส่งค่าให้ Computer ต่อไป

เขียนเป็น Flow Chart ได้ดังหน้าถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 Flow Chart ของโปรแกรมย่อยวัดค่า Delay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 โปรแกรมของ MCS-51 (Source Code MCS-51)

```

DEY1 EQU 41H
DEY11 EQU 40H
VBDATA EQU 31H
ADC EQU 30H
;RB8 FLAG BIT
ORG 0000H
JMP MAIN
;-----
ORG 000BH
CLR C ;TIMER OUT OF RANGE
RETI
ORG 0013H
JMP INTER1
;-----
MAIN: MOV IE,#10000110B
MOV IP,#00010000B
MOV PCON,#00H
MOV TMOD,#00100001B ;USE TIMER1 MODE 2 AUTO
RELOAD,TIMER0 16 BIT
MOV SCON,#11010000B ;SERIAL MODE 3
MOV TH1,#0FDH ;BAUD RATE 9600
CLR RB8 ;CLEAR MODE
SETB TR1
;-----ADC GO!-----
CLR P3.6 ;LET'S CHECK DEGREE!
SETB P3.6 ;DATA IN ADC
;-----
START: JNB RI,START ;RECEIVE DATA FROM COM YET?
CLR RI
MOV VBDATA,SBUF
MOV A,#00 ;(00)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CJNE A,VBDATA,START ;COMPUTER READY
;-----
;-----START CONNECTION-----
JNB RBB, MAN
;-----AUTO MODE-----
AUTO: CALL DEAY0
SETB P2.0
CLR TI ;SENT NOT READY DATA (FF)
MOV SBUF,#0FFH
JNB TI,$
MOV A,#01100000B
ANL A,P2
CJNE A,#01100000B,AUTO
;-----
CALL DELAY0
CLR TI ;SENT READY DATA (00)
MOV SBUF,#00
JNB TI,$
SETB P2.5
SETB P2.6
;-----
AAA: JNB RI,$ ;RECEIVE ACKNOWLEDGE FROM COM YET?
CLR RI
MOV VBDATA,SBUF ;WAIT DATA = 1
MOV A,#0
CJNE A,VBDATA,AAA ;COMPUTER READY1
CLR TR1 ;STOP SERIAL CLOCK
;-----COUNT DELAY-----
OK: CLR TRO
MOV TH0,#00 ;SET INNITIAL ALL TIMER
MOV TL0,#00
MOV DEY1,#00 ;INNITIAL ALL DELAY
MOV DEY11,#00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR RB8                ;CLEAR FRAG BIT
SETB C                 ;SET INNITAL ERROR BIT CHECK
SETB P2.5
SETB P2.6

;-----MAIN DETECTED-----
CM1:  JB P2.6,CM2      ;P2.6 MIC1 (LEFT)
      JMP CM1D
CM2:  JB P2.5,CM1      ;P2.5 MIC2 (RIGHT)
      JMP CM2D

;*****
*****
;-----MIC 1 DETECT-----
CM1D: SETB TR0         ;COUNT DEY1
;-----
CK1:  JB P2.5,ERR1
      CLR TR0
      MOV DEY1,TH0
      MOV DEY11,TL0
      SETB RB8         ;LEFT MIC DETECTED AT THE FIRST
      JMP OUTPUT
ERR1: JC CK1
      JMP ERROR

;-----
;-----MIC 2 DETECTED-----
CM2D: SETB TR0
;-----
CK2:  JB P2.6,ERR2
      CLR TR0
      MOV DEY1,TH0
      MOV DEY11,TL0
      CLR RB8         ;RIGHT MIC DETECTED AT THE FIRST
      JMP OUTPUT
ERR2: JC CK2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JMP ERROR
;-----
;-----
OUTPUT: SETB TR1                ;START SERIAL CLOCK AGAIN
        CLR TR0
;-----

        CLR TI
        MOV SBUF,DEY1          ;SENT DELAY HIGHT BYTE
        JNB TI,$
;-----

        JNB RI,$              ;RECEIVE ACKNOWLGE FROM COM YET?
        CLR RI
        MOV VBDATA,SBUF       ;WAIT DATA = 1
        MOV A,#01H
        CJNE A,VBDATA,OUTPUT  ;COMPUTER READY1
;-----
OUTPUT2:CALL DELAY0
        CALL DELAY0
        CALL DELAY0
        CALL DELAY0
        CLR TI
        MOV SBUF,DEY1         ;SENT DELAY LOW BYTE
        JNB TI,$
;-----

        JNB RI,$              ;RECEIVE ACKNOWLGE FROM COM YET?
        CLR RI
        MOV VBDATA,SBUF       ;WAIT DATA = 2
        MOV A,#02H
        CJNE A,VBDATA,OUTPUT2 ;COMPUTER READY1
;-----WAIT FOR SERIEL INTERRUPT-----
FREE:  JNB RI,$              ;WAIT DEGREE FROM COM
        CLR RI
        MOV VBDATA,SBUF       ;DEGREE IN VB DATA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
CALL GO
;-----TURN OK-----
JMP START
;-----

ERROR:CLR TR0
SETB TR1 ;START SERIAL CLOCK
MOV DEY1,#00 ;INNITIAL ALL DELAY
MOV DEY11,#00
CLR P2.0
CLR TI ;SENT NOT READY DATA (FF)
MOV SBUF,#0FFH
JNB TI,$
JMP AUTO
;-----

MAN: JNB RI,$ ;RECEIVE DATA FROM COM YET?
CLR RI
MOV VBDATA,SBUF
CALL GO
JMP START
;-----DRIVE MOTOR MODULE-----
;-----CALCULATE DIRECTION-----

GO: MOV A,ADC
CLR C
SUBB A,VBDATA
JZ STOP
JC TURNL

;-----MAKE A DECISION-----

TURNR: MOV P2,#0000001B ;TURN RIGHT
CHKR: MOV A,VBDATA
CJNE A,ADC,CHKER
JMP STOP

CHKER: MOV A,#0000H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CJNE A,ADC,CHKR
JMP STOP          ;DEGREE R DETECTED

```

```

;-----

```

```

TURNL:MOV P2,#00000010B ;TURN LEFT

```

```

CHKL: MOV A,VBDATA
      CJNE A,ADC,CHKEL
      JMP STOP

```

```

CHKEL:MOV A,#0FFH
      CJNE A,ADC,CHKL

```

```

;-----

```

```

STOP: MOV P2,#00          ;DEGREE L,R DETECTED AND STOP
      RET

```

```

;-----

```

```

DELAY0: MOV R7,#050H
DELEY1: MOV R6,#0FFH
DELEY2:  DJNZ R6,DELEY2
        DJNZ R7,DELEY1
        RET

```

```

;-----

```

```

;-----ALL INTERRUPT-----

```

```

INTER1: MOV ADC,PI
        CLR P3.6
        SETB P3.6

```

```

RETI

```

```

END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

โปรแกรมควบคุมการทำงาน

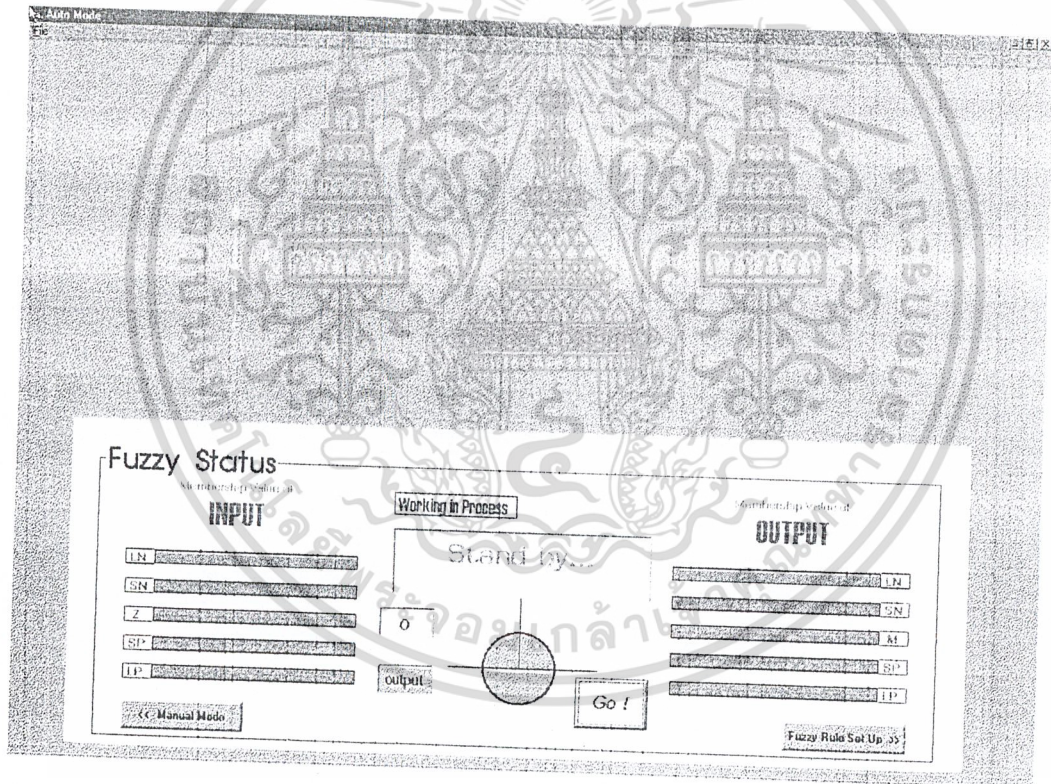
6.1 ส่วนโปรแกรมติดต่อผู้ใช้

6.1.1 หน้าจอโปรแกรมการใช้งาน Auto Mode

แบ่งออกเป็น 3 ส่วน ได้แก่

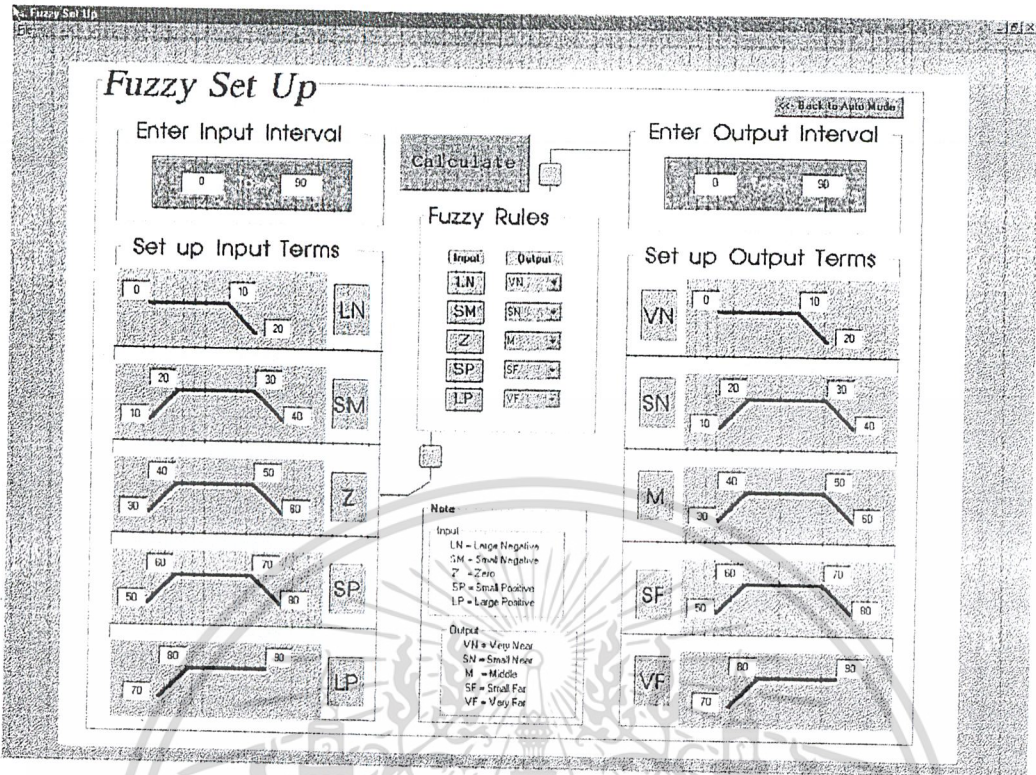
1. หน้าต่าง Auto Mode
2. หน้าต่าง Fuzzy Setup
3. หน้าต่าง Fuzzy Graph

ใช้โปรแกรม Microsoft Visual Basic 6.0 สร้างโปรแกรมในส่วนติดต่อผู้ใช้งาน ดังรูปที่ 6.1

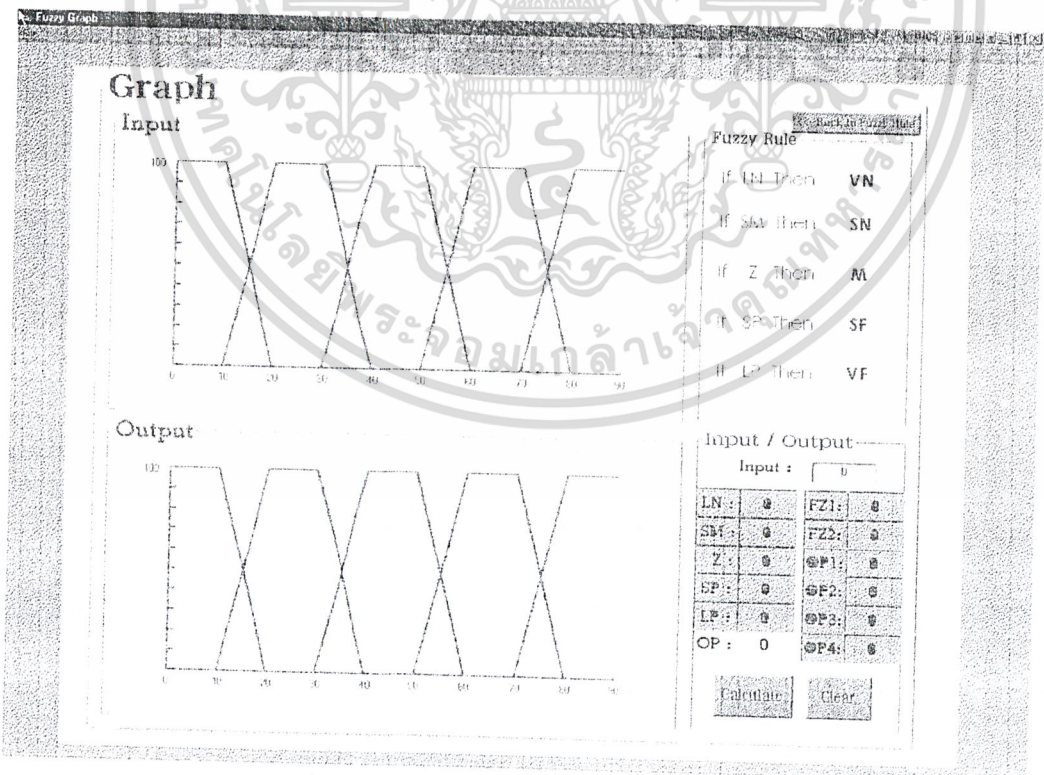


รูปที่ 6.1 แสดงส่วนประกอบของ โปรแกรม Auto Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 แสดงส่วนประกอบของโปรแกรม Fuzzy Setup

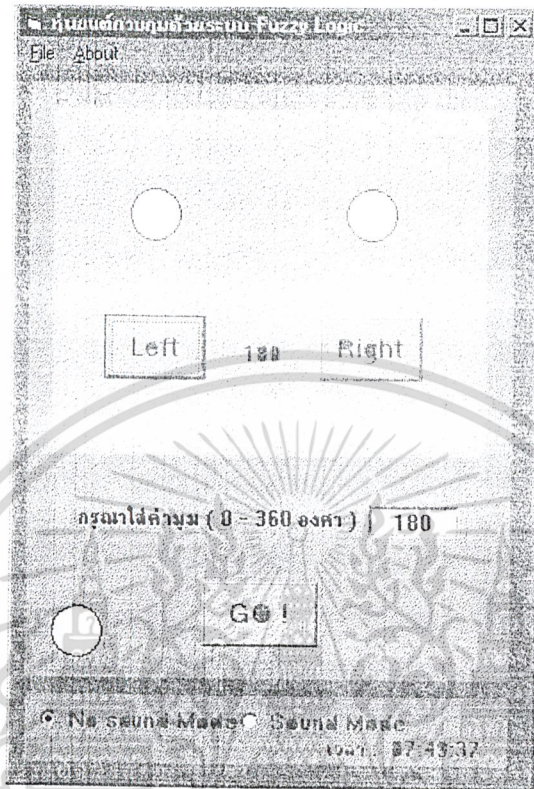


รูปที่ 6.3 แสดงส่วนประกอบของโปรแกรม Fuzzy Graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.2 หน้าจอโปรแกรมการใช้งาน Manual Mode

ใช้โปรแกรม Microsoft Visual Basic 6.0 สร้างโปรแกรมในส่วนติดต่อผู้ใช้ ดังรูปที่ 6.1



รูปที่ 6.4 แสดงส่วนประกอบของโปรแกรมที่เขียนด้วย Visual Basic 6.0

6.1.2 วิธีการใช้งานโปรแกรมติดต่อกับผู้ใช้

1. เปิดโปรแกรม Auto Mode ขึ้นมา จะทำการเปิดโปรแกรม Digital ขึ้นมาโดยอัตโนมัติ จากนั้นกดปุ่ม Power ของหุ่นยนต์
2. เมื่อกดปุ่ม GO โปรแกรมจะส่งอักขระ 00 ออกไปให้หุ่นยนต์ แล้วรอรับค่าจากหุ่นยนต์
3. หากหุ่นยนต์ส่งค่า 00 กลับมา หน้าจอจะขึ้นคำว่า "Ready" แล้ว Program จะส่งอักขระ 00 ออกไปอีกเพื่อสั่งให้หุ่นยนต์รับค่า Delay ของเสียง แล้วส่งค่า Low byte มาให้ Program ก่อนจากนั้น Program จะส่งค่า 01 ออกมา แล้วรอรับค่า High byte จากหุ่นยนต์ แต่ถ้าหากว่า หุ่นยนต์ส่งค่า OFF กลับมา หน้าจอจะขึ้นคำว่า "Not Ready" หรือ "Error" แสดงว่าหุ่นยนต์ยังไม่พร้อมที่จะใช้งาน กล่าวคือ ไมโครโฟนยัง ON อยู่ ต้องรอนกว่า ไมโครโฟนทั้งสองข้างจะ OFF ทั้งคู่ก่อน แล้วจะส่งค่า 00 มาให้ Program แล้ว Program จะส่งอักขระ 00 ออกไปอีกเพื่อสั่งให้หุ่นยนต์รับค่า Delay ของเสียง แล้วส่งค่า Low byte มาให้ Program ก่อนจากนั้น Program จะส่งค่า 01 ออกมา แล้วรอรับค่า High byte จากหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. จากนั้น Program จะทำการคำนวณค่า Low byte และ High Byte มาเป็นค่าในการคำนวณ Fuzzy แล้วนำค่า Output ที่ได้แสดงผลออกมาทางหน้าจอ แล้วส่งค่า Output นี้ส่งออกไปให้หุ่นยนต์ เพื่อให้หุ่นยนต์หมุนตัวกลิ้งไปตามทิศทางที่เสียงเข้ามา
5. ผู้ใช้สามารถปรับแต่งค่าในการคำนวณได้ด้วยตัวเองโดยคลิกปุ่ม Fuzzy Rule Setup แล้ว Program Fuzzy Rule Setup จะเปิดขึ้นมาผู้ใช้สามารถปรับแต่งค่าต่างๆดังนี้

ทางด้าน Input

Input Interval	:	ค่าสูงสุด - ต่ำสุดของ Input ที่ใช้งาน
Input Term	:	ค่าต่างๆ ใน Fuzzy ทางด้าน Input ได้แก่
LN	:	Large Negative
SM	:	Small Negative
Z	:	Zero
SP	:	Small Positive
Lp	:	Large Positive

ทางด้าน Output

Output Interval	:	ค่าสูงสุด - ต่ำสุดของ Output ที่ใช้งาน
Output Term	:	ค่าต่างๆ ใน Fuzzy ทางด้าน Output ได้แก่
LN	:	Large Negative
SM	:	Small Negative
Z	:	Zero
SP	:	Small Positive
Lp	:	Large Positive

Fuzzy Rule : กฎของ Fuzzy สามารถปรับแต่งกฎต่างๆได้ว่า Input อยู่ใน ช่วงนี้ จะให้ Output ออกมาทางช่วงไหน อาศัยกฎ “If Input = x then Output = y”

6. เมื่อผู้ใช้ต้องการจะทดสอบกฎของ Fuzzy ที่ปรับแต่งเองสามารถทำได้โดยคลิกปุ่ม Calculate เพื่อดูกราฟ Fuzzy และ กฎที่กำหนดไว้ได้ โดย Program Fuzzy Graph จะเปิดขึ้นมา
7. ผู้ใช้สามารถป้อนค่า Input ลงไปได้เองแล้วคลิกปุ่ม Calculate ใน Program Fuzzy Graph แล้ว Program จะทำการคำนวณค่าต่างๆออกมา แล้วเขียนกราฟ Fuzzy ออกมาให้ ได้ค่า Output ออกมาด้วย

6.1.2 วิธีการใช้งานโปรแกรมติดต่อกับผู้ใช้

เมื่อเปิดโปรแกรมการควบคุมหุ่นยนต์ขึ้นมา จากนั้นก็กดปุ่ม power ของหุ่นยนต์ เราสามารถแบ่งการทำงานของโปรแกรมออกเป็น 2 mode การทำงาน โดยการเลือกที่ option button 1 และ 2

1. ไม่ใช้เสียง (No Sound Mode)

โดยการเลือกที่ option button 1 : No Sound Mode

เมื่อเปิดโปรแกรมขึ้นมา ตัวกล้องจะหมุนไปยังมุม 180 องศา คือ หันไปทางด้านหน้าของกล้อง แล้ว Serial Port จะทำการตั้งค่าไปที่ port 1 มี บอกระต 9600 Hz , ไม่มีพาริตี bit , ข้อมูลทั้งหมด 8 bit และ บิตปิดท้าย 1 bit แล้วทำการเปิด port

ในโหมดนี้จะให้ผู้ใช้สามารถกำหนดค่าองศาในการหมุนของตัวกล้องเองโดยไม่อาศัยเสียงจากภายนอก แบ่งออกเป็น 2 ส่วน คือ

1.1 ผู้ใช้สามารถบังคับตัวกล้องเองได้อย่างอิสระ (Manual Mode)

ในส่วนนี้ผู้ใช้สามารถกดปุ่ม Left หรือ Right เพื่อบังคับกล้องให้สามารถหมุนไปทางซ้ายหรือขวาได้เอง และยังสามารถกดปุ่ม ซ้ายหรือขวา ของ keyboard เพื่อบังคับกล้องแทนปุ่มได้อีกด้วย

1.2 ผู้ใช้ป้อนค่ามุมเพื่อให้กล้องหมุนเองโดยอัตโนมัติ (Auto Mode)

ในส่วนนี้ให้ผู้ใช้ป้อนค่ามุมของกล้องใน text box แล้วกดปุ่ม OK หรือกด Enter ของ keyboard เพื่อให้กล้องหมุนไปยังค่ามุมนั้น โดยยึดเอา

0 องศา	:	ด้านหลังของหุ่นยนต์
90 องศา	:	ด้านขวาของหุ่นยนต์
180 องศา	:	ด้านหน้าของหุ่นยนต์
270 องศา	:	ด้านซ้ายของหุ่นยนต์
360 องศา	:	ด้านหลังของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.3 การเริ่มโปรแกรมใหม่ ,การปิดโปรแกรม

เมื่อผู้ใช้ต้องการที่จะเริ่มโปรแกรมการทำงานใหม่ สามารถทำได้โดยเปิด menu File แล้วเลือก New เพื่อทำการ reset โปรแกรมใหม่ กล้องจะหมุนไปอยู่ที่ตำแหน่ง 180 องศาอีกครั้ง

เมื่อผู้ใช้ต้องการที่จะปิดโปรแกรม สามารถเปิด menu File แล้วเลือก Exit เพื่อทำการปิดโปรแกรม และทำการปิด port ด้วย

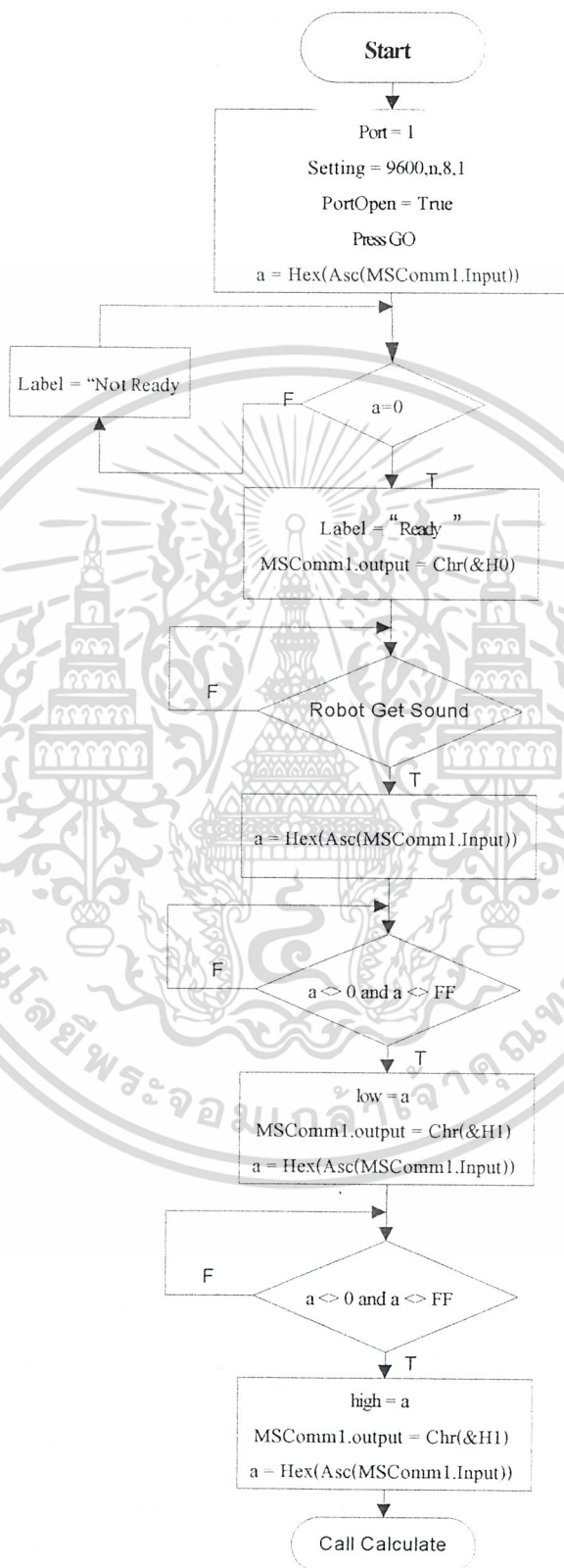
6.2 ส่วนของการติดต่อ กับหุ่นยนต์ผ่าน พอร์ตอนุกรม

จากทฤษฎีเรื่อง พอร์ตอนุกรม ในบทที่แล้ว เราสามารถนำ MSComm Control เข้ามาใช้งานใน program ของเราได้แล้วจากนั้นก็ทำการเขียน โปรแกรมควบคุมการติดต่ออุปกรณ์ภายนอกโดย

1. ทำการ ตั้งค่า port ของ computer ให้ใช้ comport 1 โดยใช้คำสั่ง `MSComm1.CommPort = 1`
2. ทำการ set ค่าของ MSComm Control ให้ใช้ บอดเรท 9600, ไม่มีพาริตีบิต, กำหนดจำนวน บิตมีค่า 8 บิต และ จำนวนบิตปิดท้าย มีค่า 1 โดยใช้คำสั่ง `MSComm1.Settings = "9600,n,8,1"`
3. ทำการ open port โดยใช้คำสั่ง `MSComm1.PortOpen = True`
4. ถ้าจะทำการส่งค่า output ไปที่ขา TxD ใช้คำสั่ง `MSComm1.Output = Chr(a)` โดยกำหนด ค่า a ให้เป็น integer แล้วส่งออกไป
5. ถ้าจะให้ program ทำการอ่านค่า input จากภายนอกเข้ามา ใช้คำสั่ง `obj = MSComm1.Input`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

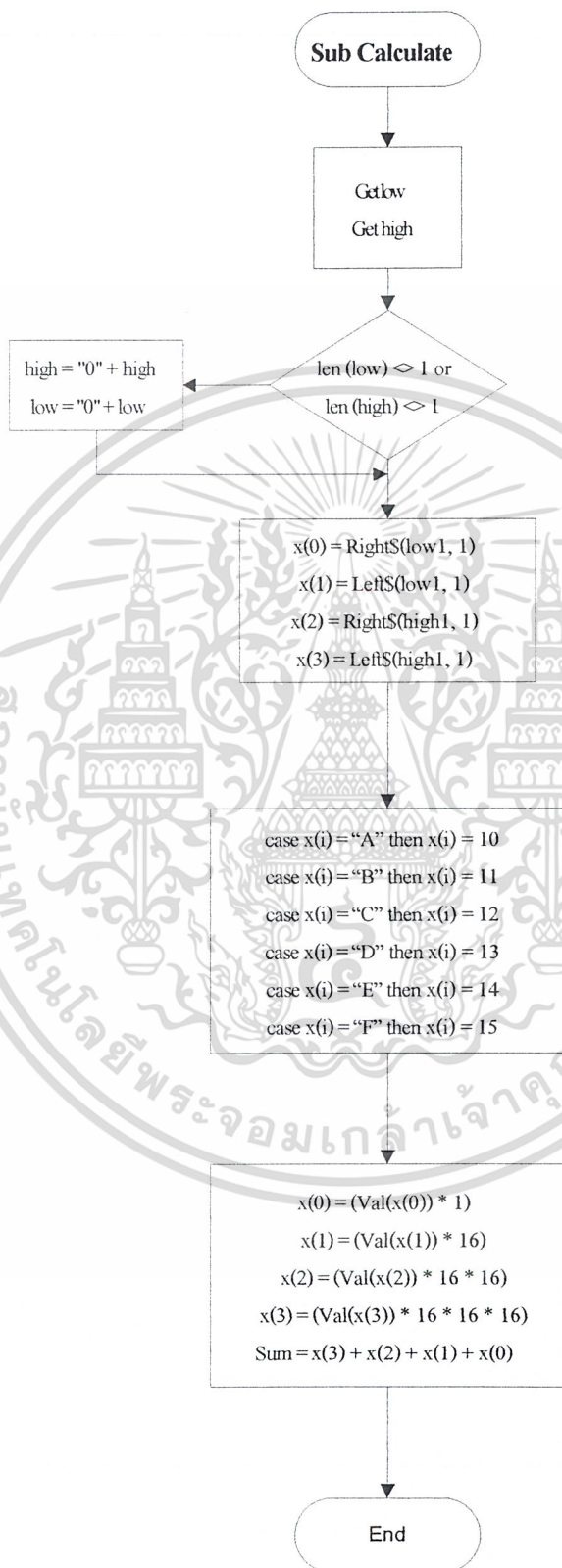
Flowchart แสดง Auto Mode



รูปที่ 6.5 Flow Chart ของโปรแกรม Auto Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

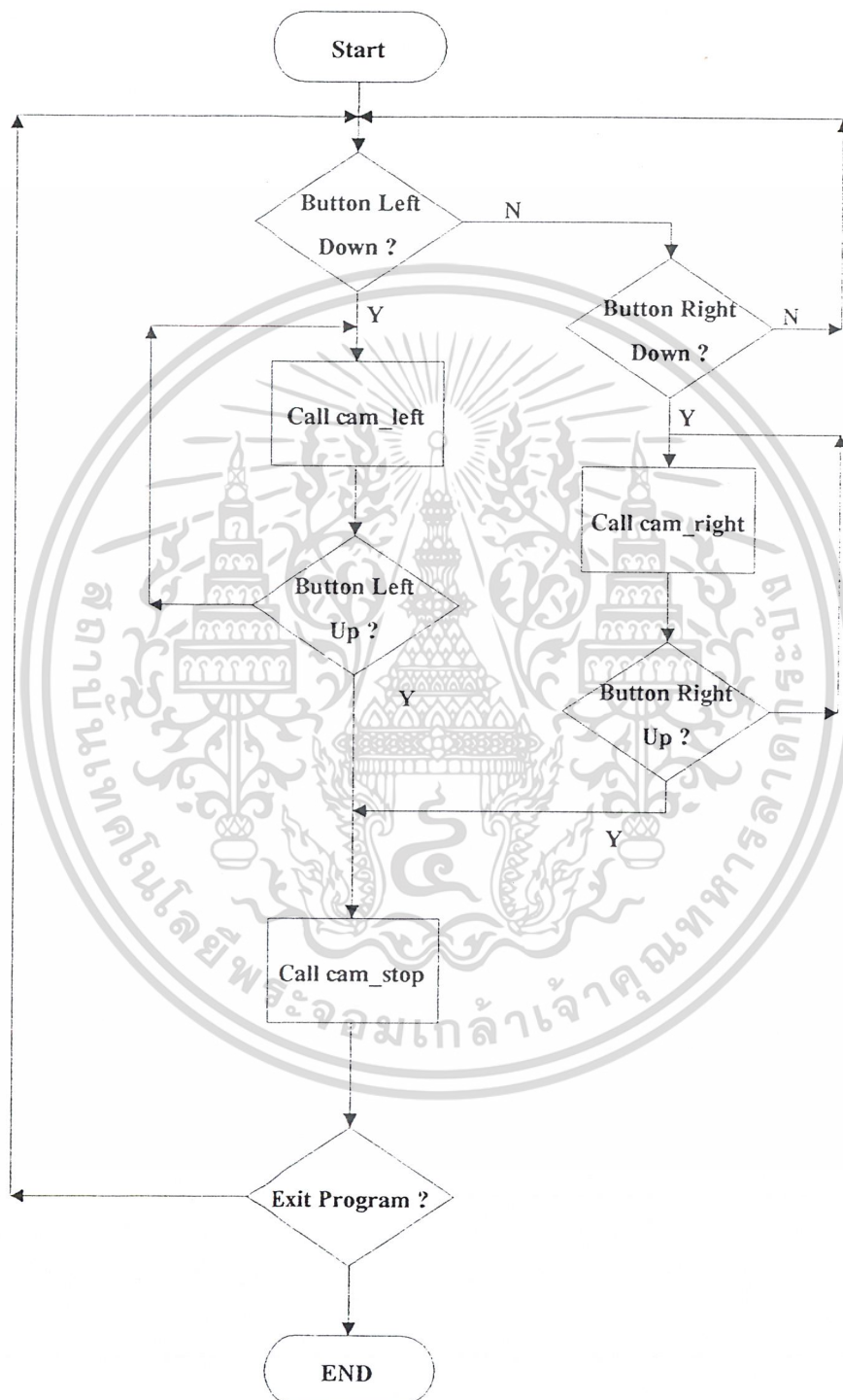
Flowchart แสดงการคำนวณ Input



รูปที่ 6.6 Flow Chart ของการคำนวณค่า Input

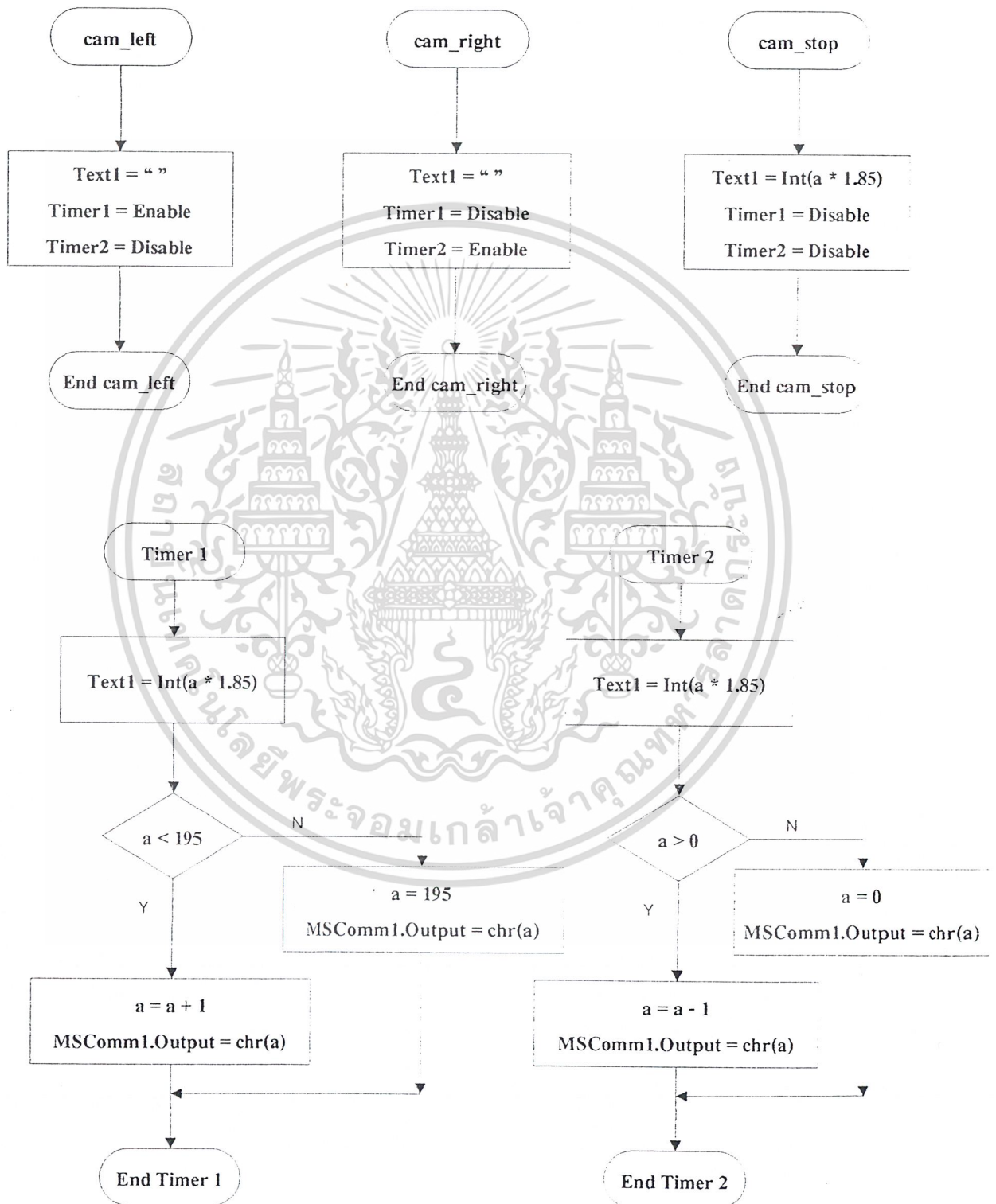
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FlowChart แสดง Manual Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart แสดง cam_left , cam_right , cam_stop , timer 1 , timer 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ฟัซซีลอจิก

7.1 ฟัซซีลอจิก (Fuzzy logic)

เป็นรูปแบบหนึ่งของตรรกศาสตร์ที่นำเอาบูลีนลอจิก(boolean logic) มาใช้ในการนิยามเหตุการณ์ หรือสถานการณ์ที่ไม่สามารถอธิบายได้อย่างชัดเจน ความสำคัญของการใช้ฟัซซีลอจิกอยู่ที่โดยธรรมชาติแล้วการให้เหตุผล หรือการระบุค่า โดยอาศัยเหตุผลตามสามัญสำนึกของคนเราส่วนใหญ่มักจะเป็นไปในลักษณะของการประมาณ

หน้าที่ของไมโครคอนโทรลเลอร์รับสัญญาณอินพุตและสร้างสัญญาณเอาต์พุตส่งออกไป ซึ่งการจะทำเช่นนี้ได้มันจะต้องอาศัยสมการต่าง ๆ หรืออาศัยกฎเกณฑ์ต่าง ๆ มากมาย เมื่อมีอินพุตเข้ามาเมื่อพิจารณาโดยใช้สมการหรือค้นหาข้อมูลจากกฎเกณฑ์ต่าง ๆ ที่ตั้งไว้ก็จะได้รับผลลัพธ์ออกมา

อย่างไรก็ตามเมื่อปัญหาที่มีความซับซ้อนมากขึ้นนั้น ก็จะทำให้กฎเกณฑ์และสมการต่าง ๆ มีเพิ่มมากขึ้นจนการวิเคราะห์ค่อนข้างยุ่งยากและมีความถูกต้องลดลง จากสาเหตุที่กล่าวมาข้างต้นทำให้ต้องเปลี่ยนแนวคิดจากเดิมเวลาตัดสินใจแบบตรรกศาสตร์ก็จะเลือกไม่อย่างใดก็อย่างหนึ่ง เช่น ไม่ขาวก็ดำมาเป็นในรูปผสมกัน ถ้าเป็นสีก็เหมือนกับเจดสีเทาที่ผสมกันระหว่างสีดำและสีขาว จึงเป็นที่มาของแนวคิดของฟัซซีลอจิก

ยกตัวอย่างเช่น เมื่อเวลาที่เราเปิดก๊อกน้ำแล้วทำการตรวจสอบว่าอุณหภูมิของน้ำเป็นอย่างไร ถ้า น้ำร้อนมากเราก็จะหมุนลูกบิดมากเพื่อทำให้น้ำเย็นขึ้น แต่ถ้าน้ำร้อนไม่มากนัก เราก็หมุนลูกบิดเล็กน้อย แต่น้ำร้อนกำลังพอดีเราก็ไม่จำเป็นต้องปรับลูกบิดแต่อย่างใด

แน่นอน เราไม่เคยนิยามว่า ร้อนมาก ร้อนน้อย หรือว่าพอดี ต้องมีค่าเป็นเท่าไร สิ่งเหล่านี้เรารู้ได้จากประสบการณ์และค่อนข้างจะแตกต่างกันไปขึ้นอยู่กับแต่ละบุคคล

นั่นคือไม่มีการระบุขอบเขตที่ชัดเจนระหว่างคำว่า ร้อนมาก หรือ ร้อนน้อย เช่นน้ำที่อุณหภูมิ 120 องศาอาจจะเรียกว่าร้อนน้อยคิดเป็น 40% และร้อนมากคิดเป็น 60% ส่วนน้ำที่อุณหภูมิ 200 องศาจะเรียกว่าร้อนมากคิดเป็น 100%

7.2 การควบคุมแบบฟัซซี่ลอจิก

ขั้นแรกของการควบคุมแบบฟัซซี่ก็คือขั้นตอนที่เรียกว่า fuzzification ซึ่งในขั้นตอนนี้ทำหน้าที่แปลงข้อมูลที่รับเข้ามาจากระบบควบคุมให้เป็นอินพุทของระบบควบคุมแบบฟัซซี่ คือมีความหมายเดียวกับระบบฟัซซี่ ซึ่งจะอยู่ในรูปแบบของค่าความเป็นสมาชิก(grade of membership) ในรูปของฟัซซี่เซต ขั้นตอนนี้ก็คล้าย ๆ กับตัวอย่างก่อนหน้า เช่น น้ำร้อนน้อย คิดเป็น 40% และน้ำร้อนมากคิดเป็น 60%

ขั้นตอนต่อไปก็คือการหาตีความและวิเคราะห์อินพุทตามเงื่อนไขที่ได้มีการกำหนดไว้หรือเข้ากฎพื้นฐานของฟัซซี่(knowledge base)แล้วจะได้ผลลัพธ์ออกมา เช่นการหาว่าถ้า น้ำร้อนน้อยคิดเป็น 100% ก็จะปรับอุณหภูมิต่ำเพื่อให้เย็นขึ้นเล็กน้อย(“a littele bit” cooler) แต่ถ้า น้ำร้อนมากคิดเป็น 100% ก็จะปรับอุณหภูมิต่ำเพื่อให้เย็นขึ้นมาก ๆ (“a lot” cooler) โดยที่คำว่า “a little bit” และ “a lot” เป็นส่วนหนึ่งที่มีการนิยามเอาไว้ในส่วนของกฎพื้นฐานของฟัซซี่

ขั้นตอนนี้ถัดไปเป็นขั้นตอนที่เรียกว่า “defuzzify” โดยผลของการเข้ากฎพื้นฐานฟัซซี่จะถูกแปลงกลับไปเป็นค่าเอาต์พุทจริง ผลลัพธ์ทั้งหมดที่ได้จะถูกประมวลเข้าด้วยกัน ยกตัวอย่างเช่นเราจะคำนวณว่าค่าที่บอกว่าเล็กน้อย 40% (40% of a little bit) หรือจำนวนมาก 60% (60% of a lot) หมายความว่าอย่างไร ผลที่ได้จะเป็นเอาต์พุท

ในการอธิบายให้เห็นภาพในเชิงปริมาณมากกว่านี้จำเป็นต้องอธิบายไปถึงเรื่องของฟังก์ชันความเป็นสมาชิก(membership function) การนอร์มัลไลซ์อินพุท และการทำดีฟัซซิฟิเคชัน(defuzzification) ของสัญญาณ

7.3 ประวัติและพัฒนาการของฟัซซี่

ในปี ค.ศ.1920 Jan Lukasiewicz นักตรรกศาสตร์ชาว โปแลนด์ ได้ตั้งหลักเกณฑ์ของตรรกแบบหลายค่าขึ้น โดยที่ค่าของตรรกมีค่าระหว่าง 0 ถึง 1 ในปี ค.ศ.1937 Max Black นักปรัชญาได้เสนอบทความเกี่ยวกับเซตแบบกำกวม ในหนังสือ Philosophy of Science เป็นบทความที่เกี่ยวกับการประยุกต์ในเซตแบบหลายค่า และได้เขียนกราฟของเซต หลายค่าขึ้นเป็นครั้งแรก ค.ศ. 1966 Loffi A. Zadeh ทำงานที่ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยแคลิฟอร์เนียเบิร์กเลย์ ได้ ตีพิมพ์ Fuzzy Sets ขึ้น โดยได้ประยุกต์ตรรกของ Lukasiewicz ให้ใช้ได้กับเซตทุก ๆ อย่าง และได้สร้างพีชคณิตของเซต แบบฟัซซี่ขึ้นได้อย่างสมบูรณ์แบบ

กลางปี ค.ศ.1970 Ebrahim H.Mamdani วิทยาลัยคินส์เมรี กรุงลอนดอน ได้ออกแบบเครื่องจักรไอน้ำที่ควบคุม โดยใช้ระบบฟัซซี่ ปี ค.ศ.1985 Masalo Togai และ Hiroyuki Watanabe ซึ่งทำงานอยู่ที่ AT & T Bell lab. ได้ร่วมกันสร้าง ดิจิตอลฟัซซี่ชิปขึ้นเป็นครั้งแรกซึ่งสามารถประมวลผลได้ 16 กฎเกณฑ์ในเวลา 12.5 ไมโคร วินาที นับเป็นจุดเริ่มต้น ทาง การพัฒนา ด้านระบบฟัซซี่ที่เป็นวงจรรสำเร็จรูปขณะนี้ บริษัท Togai Infralogic ได้ใช้ชิปที่สามารถประมาณ 2 ล้าน กฎเกณฑ์ต่อวินาที

7.4 การออกแบบเพื่อใช้ตรรกแบบฟัซซี่

การออกแบบระบบฟัซซี่เป็นขั้นตอนที่ยู่ยากยิ่งระบบนั้นมีความซับซ้อนมีตัวแปรมากการออกแบบก็จะยุ่งยากเป็นเงาตามตัวต้องอาศัยคอมพิวเตอร์ช่วยในการออกแบบแต่ในบทความนี้จะเสนอตัวอย่างง่าย ๆ เพื่อให้ผู้อ่านเห็น แนวทางการทำงานในระบบฟัซซี่เท่านั้น โดยจะใช้อุณหภูมิจากไปควบคุมการหมุนมอเตอร์ของเครื่องปรับอากาศ ซึ่งสามารถจำแนกขั้นตอนได้ 4 ขั้นตอนดังนี้

- 1) กำหนดอินพุตและช่วงฟัซซี่ของอินพุตรวมทั้งค่าดีกรีของอินพุตตามรูปที่ 2 เขียนอินพุตไว้บนแกน X
- 2) กำหนดเอาต์พุตและช่วงฟัซซี่ของเอาต์พุตรวมทั้งค่าดีกรีของเอาต์พุตตามรูปที่ 2 เขียนเอาต์พุตไว้บน แกน Y
- 3) สร้างกฎเกณฑ์และเงื่อนไขที่ใช้ค่าจากเอาต์พุต โดยใช้เงื่อนไข...แล้ว (if...then) ในที่นี้ได้เงื่อนไขคือ
 - กฎที่ 1 ถ้าหนาวแล้วมอเตอร์หยุดหมุน
 - กฎที่ 2 ถ้าเย็นแล้วมอเตอร์หมุนช้า
 - กฎที่ 3 ถ้าพอดีแล้วมอเตอร์หมุนปานกลาง
 - กฎที่ 4 ถ้าอบอุ่นแล้วมอเตอร์หมุนเร็ว
 - กฎที่ 5 ถ้าร้อนแล้วมอเตอร์หมุนเร็วมาก
- 4) หาค่าเอาต์พุตโดยวิธีทำดีฟัซซี่ฟิเคชัน (Defuzzification) วิธีที่นิยมใช้กันมากและแม่นยำสูงคือ วิธีการหา จูกรมมวล (Centroid) ของกราฟเอาต์พุต

จากรูปที่ 2 ถ้าพิจารณาอินพุตที่อุณหภูมิ 68° F มีค่าตรรกเท่ากับ 0.2 (เย็น) และมีค่าตรรกเท่ากับ 0.7 (พอดี) เมื่อใช้เงื่อนไขของ if...then จะทำให้ได้ความเร็วของมอเตอร์มีตรรก 0.2 (หมุนช้า) และ 0.7 (หมุนปานกลาง) ซึ่งเป็นไป ไม่ได้ที่มอเตอร์จะหมุนได้ 2 ระดับในเวลาเดียวกันเราจึงหาค่า 1 ค่าของความเร็วของมอเตอร์ โดยวิธีหาจูกรมมวล ระหว่างพื้นที่การหมุนช้ากับการหมุนปานกลางในระดับ 0.2 และ 0.7 ตามลำดับ จะได้ค่าการหมุนของมอเตอร์ดีฟัซซี่เท่ากับ 47 รอบ/นาที จึงสรุปได้ว่าถ้าอุณหภูมิ 68° F มอเตอร์จะหมุนด้วยความเร็ว 47 รอบ/นาที ส่วนที่อุณหภูมิอื่น ๆ ก็หาค่าได้ในทำนองเดียวกัน ระบบการทำงานนี้เป็นการออกแบบเพื่อใช้ควบคุมอุณหภูมิของห้องนั่นเอง ถ้าระบบ มีความซับซ้อนมากบางครั้งจำเป็นต้องอาศัยตัวดำเนินการ AND, OR, NOT เข้าช่วยเพื่อให้ได้คำตอบเดียวอีกทีหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 รูปแบบของพีชคณิตในทางคณิตศาสตร์

ก่อนอื่นขออธิบายลักษณะของ Fuzzy Sets ในรูปแบบสมการทางคณิตศาสตร์ เพื่อนำไปใช้อ้างอิงกับการประยุกต์ใช้งานจริงดังต่อไปนี้

1. Fuzzy Set and Membership Functions

Fuzzy Set A ใน U สามารถเขียนในรูปคู่อันดับของ

$$A = \{ (x, \mu_A(x)) \mid x \in U \}$$

โดยที่

$\mu_A(x)$ เรียกว่า Membership Function (MF) สำหรับ Fuzzy Set A

U เรียกว่า Universe of discourse

ข้อสังเกต

- MF เป็นการเทียบค่า x เข้ากับระดับการเป็นสมาชิก โดยมีค่าตั้งแต่ 0 แต่ไม่เกิน 1
- ถ้าค่า MF ของ $\mu_A(x)$ มีเพียง 0 หรือ 1 Set A จะกลายเป็น Classical Set ไป

2. Universe of discourse (U)

แบ่งได้เป็น

2.1 ค่าที่ไม่ต่อเนื่อง (Discrete Object)

- มีการเรียงลำดับ (Ordered)
- ไม่มีการเรียงลำดับ (Nonordered)

2.2 ค่าที่ต่อเนื่อง

ตัวอย่างที่ 1 Fuzzy Sets ของค่าไม่ต่อเนื่องที่ไม่เรียงลำดับ

กำหนด $X = \{ \text{San Francisco, Boston, Los Angeles} \}$ แทนเซตของเมืองที่เลือก

Fuzzy Set C = “ระดับการตัดสินใจในการเลือกอยู่อาศัย”

ได้ผลเป็น

$$C = \{ (\text{San Francisco}, 0.9), (\text{Boston}, 0.8), (\text{Los Angeles}, 0.6) \}$$

หมายเหตุ – ค่า X ในที่นี้แสดงชื่อเมือง ที่มีค่าไม่ต่อเนื่อง และไม่มีการเรียงลำดับ

- ผลของการตัดสินใจขึ้นอยู่กับแต่ละบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

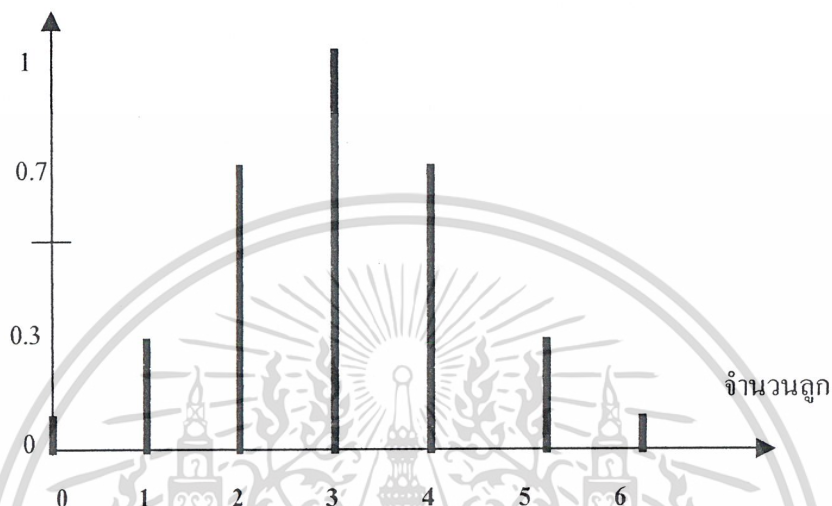
ตัวอย่างที่ 2 Fuzzy Sets ของค่าไม่ต่อเนื่องที่มีการเรียงลำดับ

กำหนด $X = \{0,1,2,3,4,5,6\}$ แทนเซตของจำนวนบุตรในแต่ละครอบครัว

Fuzzy Set A = “จำนวนบุตรที่อยากมีในแต่ละครอบครัว”

ได้ผลเป็น

$$A = \{(0,0.1), (1,0.3), (2,0.7), (3,1), (4,0.7), (5,0.3), (6,0.1)\}$$



รูปที่ 7.1 Fuzzy Sets ของค่าไม่ต่อเนื่องที่มีการเรียงลำดับ

ตัวอย่างที่ 3 Fuzzy Sets ของค่าที่ต่อเนื่อง

กำหนด $X = \{R^+\}$ แทนเซตความเป็นไปได้ของอายุคน

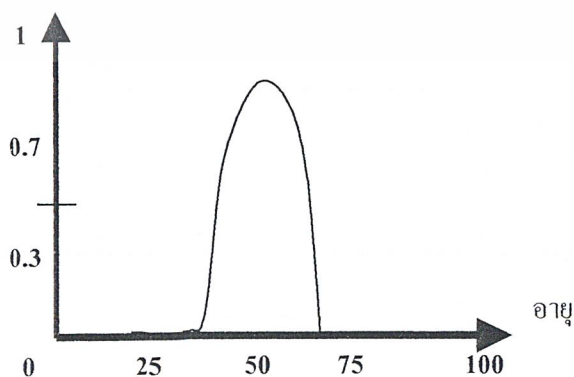
Fuzzy Set B = “คนที่มีอายุประมาณ 50 ปี”

ได้ผลเป็น

$$B = \{(x, U_B(x)) \mid x \in X\}$$

โดยที่

$$U_B(x) = \frac{1}{1 + [(x-50)/10]^4}$$



รูปที่ 7.2 Fuzzy Sets ของค่าที่ต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Linguistic variables and Linguistic values

X (Universe of discourse) สามารถประกอบด้วยหลาย Fuzzy Sets ซึ่งมี Membership Function อยู่บนค่า X

โดยที่ชื่อของ Fuzzy Set แต่ละตัวเป็นชื่อในเชิงภาษาที่เราเรียกกันอย่างคุ้นเคย เช่น

“มาก”, “ปานกลาง” หรือ “น้อย”

“สูง”, “กลาง” หรือ “ต่ำ”

เราจะเรียกค่าเชิงภาษานี้ว่า **Linguistic Values** หรือ **Linguistic Labels**

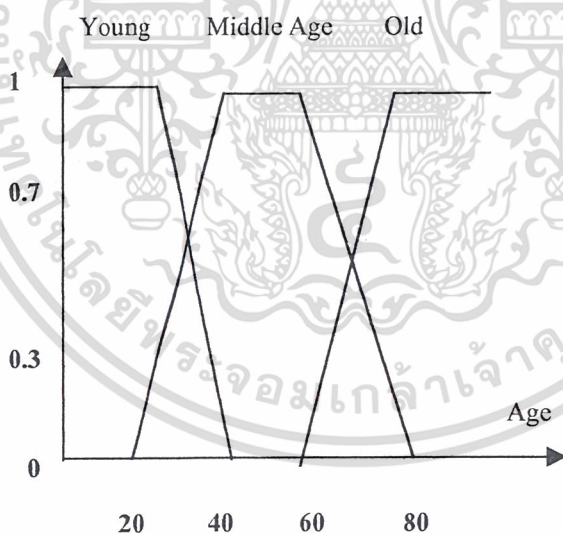
ตัวอย่างที่ 4 Linguistic Values และ Linguistic Labels

กำหนด $X = \text{“Age”}$ แทนเซตความเป็นไปได้ของอายุคน

ประกอบด้วย Fuzzy Sets ของ “Young”, “Middle Age”, “Old”

Fuzzy แต่ละตัวมีคุณสมบัติตาม Membership Function เป็น

$$U_{\text{Old}}(x), U_{\text{Middle Age}}(x), U_{\text{Young}}(x)$$



รูปที่ 7.3 Fuzzy Sets ต่างๆ แบ่งตาม Linguistic Values ของอายุคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Definition

ในหัวข้อนี้ จะอธิบายนิยามต่างๆที่ใช้ในระบบ Fuzzy logic

4.1 Support

ค่า Support ของ Fuzzy Set A แทนเซตของทุกค่า x ใน X ที่มีค่า $U_A(x) > 0$

$$\text{Support}(A) = \{x \mid U_A(x) > 0\}$$

4.2 Core

ค่า Core ของ Fuzzy Set A แทนเซตของทุกค่า x ใน X ที่มีค่า $U_A(x) = 1$

$$\text{Core}(A) = \{x \mid U_A(x) = 1\}$$

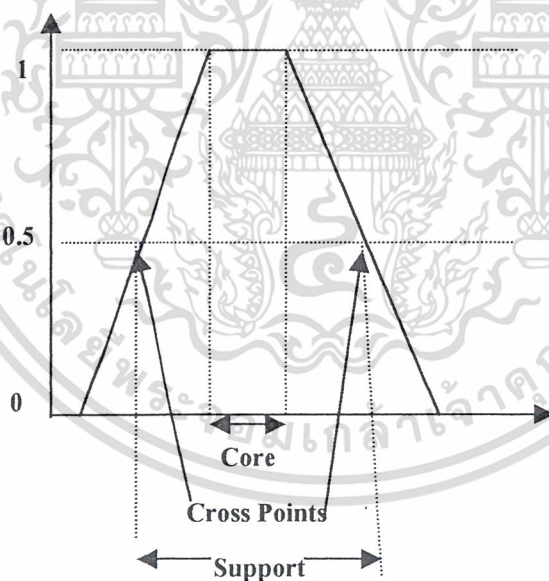
4.3 Crossover points

ค่า Crossover points ของ Fuzzy Set A เป็นจุดที่ $x \in X$ โดยที่ $U_A(x) = 0.5$

$$\text{Crossover}(A) = \{x \mid U_A(x) = 0.5\}$$

4.4 Fuzzy Singleton

Fuzzy singleton เป็น Fuzzy Set ที่มีค่าเดียวใน X ที่ $U_A(x) = 1$



รูปที่ 7.4 แสดง Core, Support และ Cross Overpoints ของ Fuzzy Set หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 Containment or Subset

Fuzzy Set A เป็น Containment ใน Fuzzy Set B ก็ต่อเมื่อ $U_A(x) \leq U_B(x)$ ในทุก x

$$A \subseteq B \iff U_A(x) \leq U_B(x)$$

4.5 Union

Union ของ Fuzzy Set A และ Fuzzy Set B และ Fuzzy Set C

เขียนในรูป $C = A \cup B$ หรือ $C = A \text{ OR } B$ เขียนในรูป Membership Function ได้

$$U_C(x) = \max (U_A(x) , U_B(x))$$

4.6 Intersection

Intersection ของ Fuzzy Set A และ Fuzzy Set B เป็น Fuzzy Set C

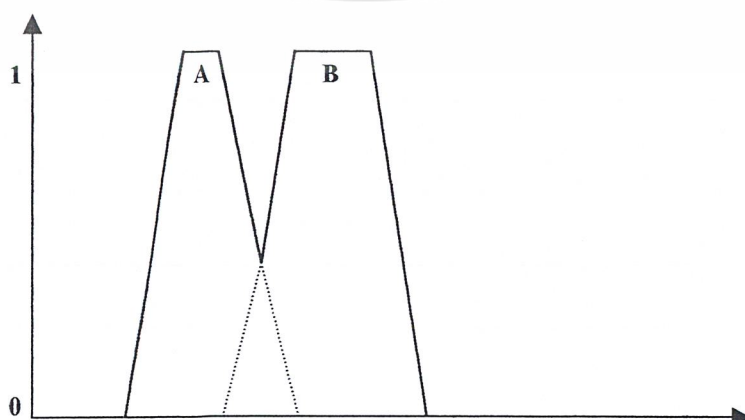
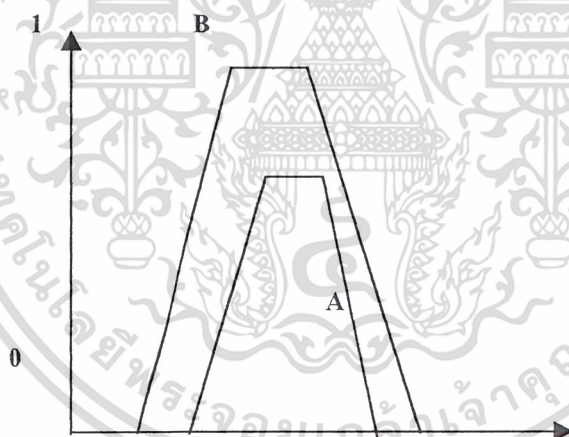
เขียนในรูป $C = A \cap B$ หรือ $C = A \text{ AND } B$ เขียนในรูป Membership Function ได้

$$U_C(x) = \min (U_A(x) , U_B(x))$$

4.7 Complement

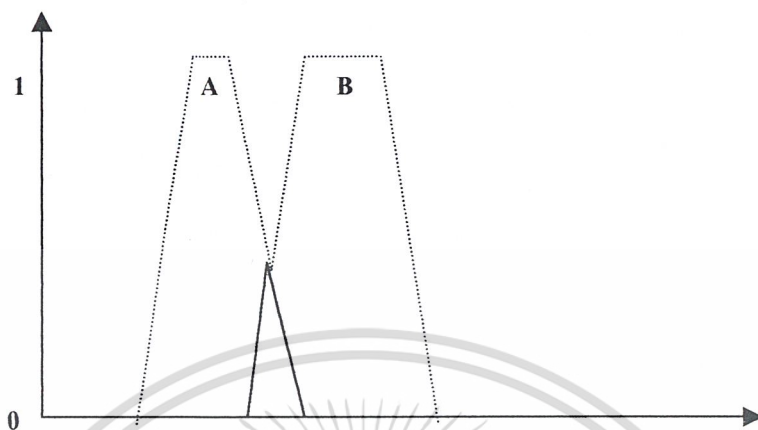
Complement ของ Fuzzy Set A เขียนแทนด้วย Not A

$$U_{\text{Not } A}(x) = 1 - U_A(x)$$

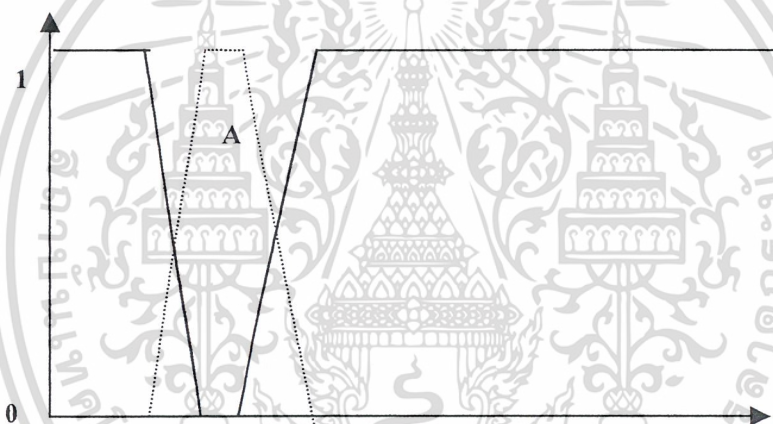


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.6 รูป Fuzzy Set (A OR B)



รูปที่ 7.7 รูป Fuzzy Set (A AND B)



รูปที่ 7.8 รูป Fuzzy Set (Not A)

5. Membership Function Formulation

เราสามารถเลือก Membership Function Formulation ได้อย่างอิสระตามระบบที่เราออกแบบ โดยคำนวณค่า Membership Value ได้จากแต่ละสมการดังนี้

5.1 Triangular MF

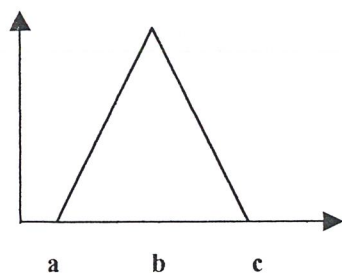
ประกอบด้วยพารามิเตอร์ 3 ตัว คือ $\{a, b, c\}$ โดยที่ $a < b < c$

และ x แทนพิกัดร่วมอยู่ระหว่างจุดทั้งสาม

$$\text{Triangle } (x:a,b,c) \begin{cases} 0 & , x \leq a \\ x - a & , a \leq x \leq b \\ b - a & \\ c - x & , b \leq x \leq c \\ c - b & \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$0, c \leq x$$

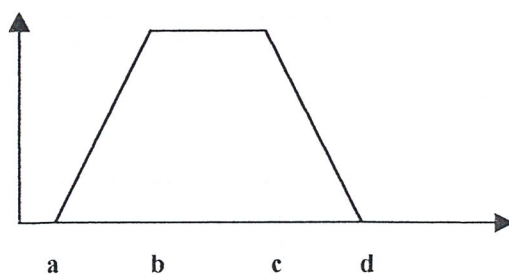


รูปที่ 7.9 Triangular Membership Function

5.2 Trapezoidal MF

ประกอบด้วยพารามิเตอร์ 4 ตัว คือ $\{a, b, c, d\}$ โดยที่ $a < b < c < d$ และ x แทนพิกัดร่วมอยู่ระหว่างจุดทั้งสาม

$$\text{Trapezoidal}(x:a,b,c,d) = \begin{cases} 0 & , x \leq a \\ \frac{x-a}{b-a} & , a \leq x \leq b \\ 1 & , a \leq x \leq b \\ \frac{d-x}{d-c} & , c \leq x \leq d \\ 0 & , d \leq x \end{cases}$$



รูปที่ 7.10 Trapezoidal Membership Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกต ทั้ง Triangle , Trapezoidal มีคุณสมบัติเชิงเส้น(Linear)และไม่ราบเรียบ (Not Smooth) ซึ่ง MF ต่อจากนี้จะตรงข้าม คือ ไม่เชิงเส้นและมีความราบเรียบ ได้แก่ Gaussian, Generalized bell และ Sigmoidal MF

5.3 Gaussian MF

ประกอบด้วยพารามิเตอร์ 2 ตัว คือ $\{c, \sigma\}$

โดยที่ c = จุดศูนย์กลางของ MF

σ = แทนความกว้างของ MF

$$\text{Gaussian}(x; c, \sigma) = e^{-0.5 \cdot (x-c)^2 / \sigma^2}$$

5.4 Generalized bell MF

ประกอบด้วยพารามิเตอร์ 3 ตัว คือ $\{a, b, c\}$

โดยที่ a = แทนตัวกำหนดความกว้างของ MF

b = ตัวกำหนดความชันของ MF

c = จุดศูนย์กลางของ MF

$$\text{bell}(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

ข้อสังเกต ทั้ง Gaussian , Generalized bell MF มีคุณสมบัติสมมาตร ส่วน Sigmoidal MF จะใช้ในการออกแบบที่ไม่สมมาตร

5.5 Sigmoidal MF

ประกอบด้วยพารามิเตอร์ 2 ตัว คือ $\{a, c\}$

โดยที่ a = แทนตัวกำหนดความชันของ MF ที่จุด Cross points

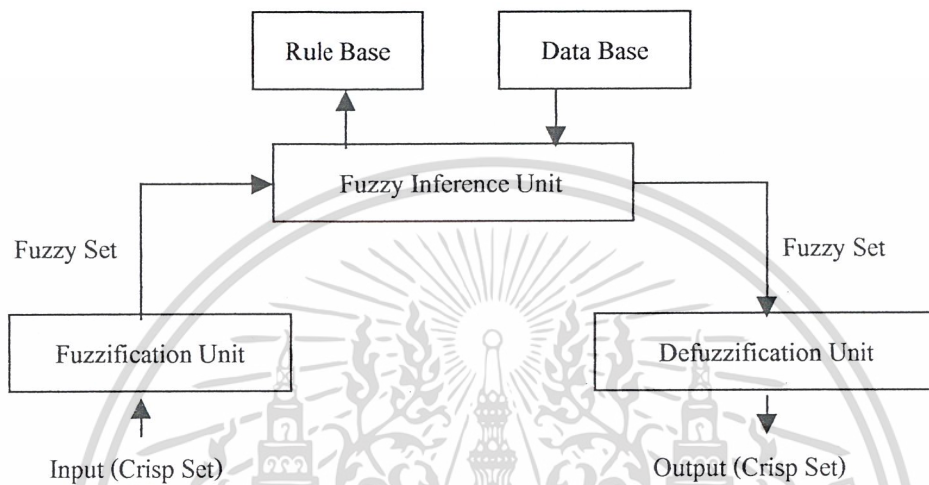
c = แทน Cross points

$$\text{sig}(x; a, c) = \frac{1}{1 + \exp[-a(x-c)]}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6 การประยุกต์ใช้งานฟัซซีลอจิก (Fuzzy Logic Application)

รูปแบบพื้นฐานในการประยุกต์ใช้งาน Fuzzy Logic ในการควบคุมระบบ สามารถสรุปได้ดังรูป



รูปที่ 7.11 รูปแบบการทำงานพื้นฐานในการควบคุมแบบ Fuzzy

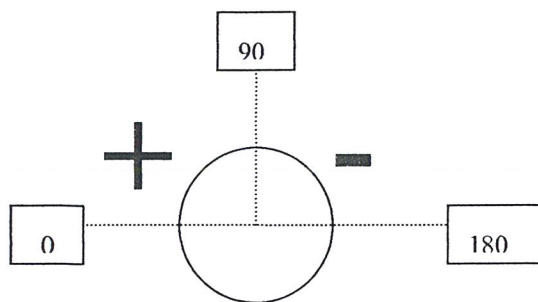
ในที่นี้จะอธิบายถึงการนำเอาทฤษฎีฟัซซีลอจิกไปควบคุม Input และ Output ของระบบ โดยดำเนินการตามขั้นตอนดังต่อไปนี้

1. วิเคราะห์ระบบ เพื่อกำหนด Input และ Output ของระบบ

ถือเป็น กระบวนการแรกของการออกแบบ โดยทั่วไป Input และ Output ของระบบจะถูกกำหนดจาก ลักษณะของ Input และ Output ที่เป็นไปได้ ซึ่งถ้ามองจากสมการ $A = \{ (x, \mu_A(x)) \mid x \in U \}$ ค่าดังกล่าวก็คือ Universe of discourse (U) ของ Input และ Output นั้นเอง

- 1.1 การหาค่า Input ที่เป็นไปได้ ในที่นี้ Input ของระบบก็คือค่า Delay ของ Mic สองตัวแรกที่สามารถรับเสียงได้ก่อน ซึ่งได้มาจากการทดลอง มี 1 Input โดยมีค่าอยู่ระหว่าง -65536 ถึง $+65536$ (-2^{16} , $+2^{16}$) ซึ่งได้จากค่า Delay 16 บิต ของ Timer 0 นั้นเอง โดย Delay บวกเป็น Delay ของฝั่งซ้ายมือของกล้อง และ Delay ลบ เป็น Delay ของฝั่งขวามือของกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



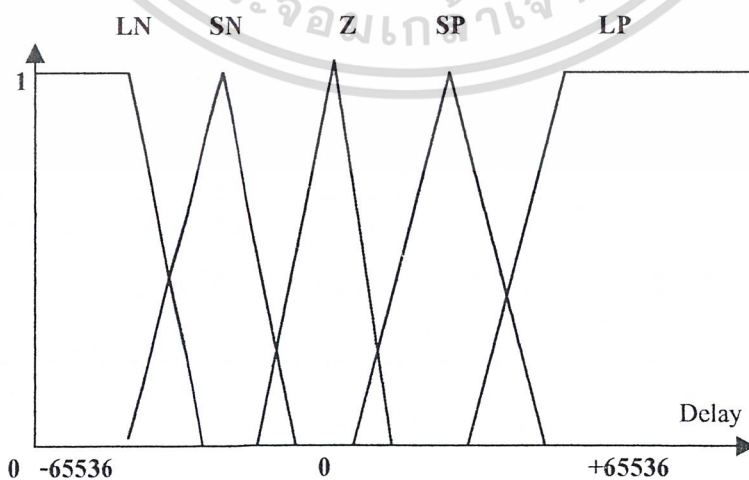
รูปที่ 7.12 แสดงลักษณะการแบ่งค่า Delay เอามาใช้งาน

(ดูหัวข้อ การทดลอง ตอนที่ 2 ซึ่งจะระบุถึงวิธีการ ผล สรุป และวิเคราะห์อย่างชัดเจน)

1.2 การหาค่า Output ที่เป็นไปได้ กำหนดให้หุ่นสามารถตรวจจับทิศทางของเสียงได้ ตั้งแต่ 0-180 องศา

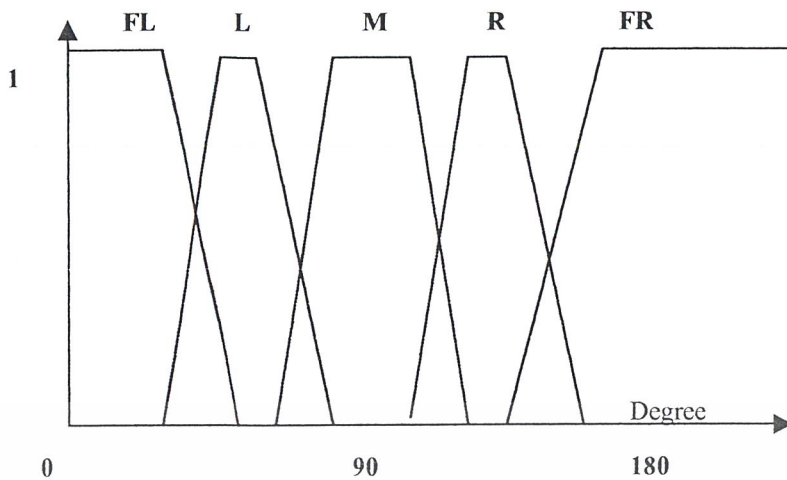
2. Fuzzification

คือการนำค่า Input ที่กำหนดไว้ ซึ่งเป็น Crisp Set มาแปลงเป็น Fuzzy Set โดยกำหนด จำนวน Fuzzy Set หรือ Fuzzy Term ตามลักษณะของ Linguistic Labels ต่างๆ ดังนี้ Large Negative (LN), Small Negative (SN), Zero (Z), Small Positive (SP), Large Positive (LP) มีจำนวน 5 เทอมสำหรับ Input และ 5 เทอมคือ Further Left (FL), Left (L), Middle (M), Right (R), Further Right (FR) สำหรับ Output และ เลือกใช้ กราฟรูป สี่เหลี่ยมคางหมูหรือสามเหลี่ยม (Trapezoidal ,Triangle Membership Function) ซึ่งสามารถปรับ ได้อิสระจาก โปรแกรมในคอมพิวเตอร์ ในการกำหนดค่าระดับความเป็นสมาชิกของ Fuzzy Set ($\mu_x(x)$) ซึ่งในขั้นตอนนี้เป็นการเลือกได้ตามอิสระ ขึ้นอยู่กับระบบและประสบการณ์ของผู้ออกแบบเอง



รูปที่ 7.13 กราฟอัตราความเป็นสมาชิกของ Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.14 กราฟอัตราความเป็นสมาชิกของ Output

3. Fuzzy Rules

ขั้นตอนนี้จะเป็นการกำหนดกฎพื้นฐาน (Base Rules) ซึ่งอยู่ในรูปแบบของ

IF (Antecedent) THEN (Consequent)

ดังนี้

กฎที่	IF	Input	THEN	Output
1	IF	Large Negative (LN)	THEN	Further Left (FL)
2	IF	Small Negative (SN)	THEN	Left (L)
3	IF	Zero (Z)	THEN	Middle (M)
4	IF	Small Positive (SP)	THEN	Right (R)
5	IF	Large Positive (LP)	THEN	Further Right (FR)

4. Defuzzification

คือขั้นตอนท้ายสุดในการควบคุมระบบ โดยทำการแปลงค่าจาก Fuzzy Set ให้กลับไปเป็นค่า Crisp Set ค่าเดียว เพื่อตัดสินใจนำไปควบคุมระบบต่อไป

โดยทั่วไปการ Defuzzification มีหลายวิธี แต่ที่นิยมกัน มีอยู่ 4 วิธี คือ

- 4.1 **Min** เป็นการกำหนดค่า Output จากค่าความเป็นสมาชิกของกฎที่มีค่าต่ำสุด
- 4.2 **Max** เป็นการกำหนดค่า Output จากค่าความเป็นสมาชิกของกฎที่มีค่าสูงสุด
- 4.3 **MoM (Mean of Maximum)** เป็นการกำหนดค่า Output จากค่าเฉลี่ยของค่าความเป็นสมาชิกของกฎที่มีค่าสูงสุด
- 4.4 **CoA (Centroid of Area)** เป็นการกำหนดค่า Output จากทุกกฎ โดยการหาจุดศูนย์กลางถ่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในที่นี้จะใช้วิธี CoA (Centroid of Area) เนื่องจากจะนำเอาทุกๆกฎมาใช้งาน ซึ่งจะให้ค่า Output ที่สอดคล้องกับความเป็นจริงมากที่สุด มีสมการดังนี้

$$I = \frac{\sum (un * Un)}{\sum (un)}$$

โดยที่

I = Result of Centroid Method

un = Degree of Membership

Un = Status from IF/Then Rules

ซึ่งจะได้ Output เป็น Crisp Set ค่าเดียว นำไปควบคุมระบบต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

ผลการทดลอง

ตอนที่ 1 การทดสอบการหมุนกลิ้ง

วิธีการทดลอง

1. กำหนดจุดใดจุดหนึ่งบนตัวกลิ้ง เป็นจุดวัด
2. ตั้งค่ากลิ้งไว้ที่ 0 องศา (มุมเปรียบเทียบ) แล้วจึงสั่งให้กลิ้งหมุน
3. นำอุปกรณ์ที่ใช้วัดองศาตั้งให้ตรงกับตำแหน่งจุดที่สั่งไว้
4. เพิ่มองศาของกลิ้งเป็น 10 องศา (มุมที่ต้องการวัด) แล้วจึงสั่งให้กลิ้งหมุน
5. วัดองศาเทียบกับตำแหน่งแรก
6. กลับไปทำข้อ 2 ใหม่ จน มุมที่ต้องการวัดเท่ากับ 90,180,270 แล้วจึงตั้งค่าเหล่านี้เป็นมุมเปรียบเทียบ แล้วทำข้อ 2 ถึง ข้อ 5 จะได้ผลการทดลองเป็น 4 กลุ่ม ดังนี้

ผลการทดลอง

ค่ามุมเปรียบเทียบ (องศา)	ค่ามุมที่ต้องการวัด (องศา)	ค่ามุมที่วัดได้ (องศา)	ค่าผิดพลาด (องศา)	เปอร์เซ็นต์ ผิดพลาด (%)
0	10	30	+20	200.00
0	20	50	+30	150.00
0	30	60	+30	100.00
0	40	69	+29	72.50
0	50	77	+22	44.00
0	60	84	+24	40.00
0	70	90	+20	28.57
0	80	99	+19	23.75
0	90	105	+15	16.66
90	100	115	+15	15.00
90	110	120	+10	9.09
90	120	127	+7	5.83
90	130	140	+10	7.69
90	140	145	+5	3.57
90	150	155	+5	3.33
90	160	162	+2	1.25
90	170	175	+5	2.94
90	180	183	+3	1.66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่ามุมเปรียบเทียบ (องศา)	ค่ามุมที่ต้องการวัด (องศา)	ค่ามุมที่วัดได้ (องศา)	ค่าผิดพลาด (องศา)	เปอร์เซ็นต์ ผิดพลาด (%)
180	190	200	+10	5.26
180	200	212	+12	6.00
180	210	224	+14	6.66
180	220	238	+18	8.18
180	230	246	+16	6.95
180	240	260	+20	8.33
180	250	270	+20	8.00
180	260	275	+15	5.76
180	270	285	+15	5.55
270	280	292	+12	4.28
270	290	305	+15	5.17
270	300	315	+15	5.00
270	310	321	+11	3.54
270	320	333	+13	4.06
270	330	341	+11	3.33
270	340	350	+10	2.94
270	350	358	+8	2.29
270	360	368	+8	2.22
		\bar{X}	14.278	
		σ	7.097	

$$\text{เปอร์เซ็นต์ผิดพลาด (\%)} = \frac{(\text{ค่าที่วัดได้} - \text{ค่าที่ต้องการ})}{\text{ค่าที่ต้องการ}} * 100$$

ตารางที่ 8.1 แสดงผลการทดสอบการหมุนของกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

จากผลการทดลองแบ่งการผลทดลองเป็น 4 ช่วง ตามค่ามุมที่ใช้เปรียบเทียบได้ว่า

- ที่ 0 องศา มีค่าผิดพลาดมากที่สุด 30 องศา และผิดพลาดน้อยสุด 15 องศา
- ที่ 90 องศา มีค่าผิดพลาดมากที่สุด 15 องศา และผิดพลาดน้อยสุด 2 องศา
- ที่ 180 องศา มีค่าผิดพลาดมากที่สุด 20 องศา และผิดพลาดน้อยสุด 10 องศา
- ที่ 270 องศา มีค่าผิดพลาดมากที่สุด 15 องศา และผิดพลาดน้อยสุด 8 องศา

วิเคราะห์ผลการทดลอง

- จากการทดลองพบว่า ที่ 0-90 องศา มีค่าผิดพลาดมากที่สุด เนื่องจากในช่วงนี้ เป็นค่าช่วงคั่นๆ ของ VR ค่าจึงผิดพลาดสูงมาก
- ช่วง 90-180 องศา ค่าผิดพลาดน้อยมาก ดูแล้วเหมือนว่าน่าจะเหมาะสมดี แต่เมื่อลองวัดใหม่พบว่า ค่าที่ได้ใกล้เคียงกับ ช่วง 180-270 และ 270-360 องศา จึงถือเป็นความคลาดเคลื่อนจากการวัดและตัวผู้วัดเอง
- ช่วง 180-270 และ 270-360 องศา เป็นช่วงที่วัดได้ค่าผิดพลาดใกล้เคียงกัน ถือว่าเป็นช่วงที่ผลการทดลองน่าเชื่อถือที่สุด
- ถ้าสังเกตในแต่ละช่วงจะพบว่า เมื่อหมุนใกล้ๆ ค่าผิดพลาดจะสูงมาก เพราะ Motor พังเริ่มหมุนและหยุดหมุนไม่ทัน แต่เมื่อหมุนไปไกลๆ Motor หยุดทัน ค่าผิดพลาดจะลดลง จนเหลือค่าผิดพลาดน้อยสุดที่ 1.25 % (จากผลการทดลอง)
- สรุปแล้ว ค่าผิดพลาดที่เกิดขึ้นเกิดจากปัญหาทางด้าน Mechanics มากกว่า Electronics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนที่ 2 การหาค่าความรู้พื้นฐาน

การทดลองในขั้นนี้จะเป็นการทดลองเพื่อหาค่า Delay ของเวลาที่เกิดขึ้นจากไมโครโฟน 2 ตัว เพื่อเป็นความรู้พื้นฐานในการกำหนด Universe of discourse ของ Fuzzy Terms หรือ Fuzzy Sets นั้นเอง

อุปกรณ์

1. แหล่งกำเนิดเสียงที่มีความถี่และแอมพลิจูดคงที่ (ในที่นี้ใช้ Buzzer 28 Vdc)
2. ค่ายาว 90 ซม. (เพื่อให้ระยะห่างจากจุดศูนย์กลางคงที่)

วิธีการทดลอง

1. มัด Buzzer กับแกนกลางของ Motor ให้แน่น โดยให้ระยะห่างคงที่
2. กำหนดองศาต่างๆตั้งแต่ 0-180 องศา บนตัวกล่องเพื่อเป็นพิกัดในการเทียบตำแหน่ง
3. เริ่มที่ 0 องศา ค้างค้ายให้ติง กดปุ่มให้ Buzzer ทำงาน,บันทึกค่า Delay
4. เปลี่ยนองศาถัดไป 10 องศาแล้วทำซ้ำข้อ 3 จนครบ 180 องศา

ผลการทดลอง

ตารางที่ 8.2 แสดงผลการทดสอบการหาค่าความรู้พื้นฐาน

มุม (องศา)	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5	Min	Mean	Max
180	16705	2595	9383	13139	14803	2595	8364.4	16705
170	1847	1887	1847	1647	1819	1647	1809.4	1887
160	1103	991	2255	1871	1199	991	1483.8	2255
150	967	1911	2211	1951	1715	967	1751	2211
140	1587	1615	2111	1583	1583	1583	1695.8	2111
130	635	951	1299	1243	951	635	1015.8	1299
120	1043	483	763	563	979	483	766.2	1043
110	155	879	891	855	315	155	619	891
100	599	233	451	175	431	175	377.8	599
90	619	275	839	11	19	11	352.6	839
80*	143	183	31	59	52	31	93.6	183
70	383	919	439	663	1855	383	851.8	1855
60	1347	1115	567	635	535	535	839.8	1347
50	691	955	963	967	943	691	903.8	967
40	1103	1359	1327	1389	1331	1103	1301.8	1389
30	1451	1459	1427	1495	1207	1207	1407.8	1495
20	1531	1759	1539	1531	1722	1531	1616.4	1759
10	1255	1839	1807	1543	1775	1255	1643.8	1839
0	2027	1839	1779	1881	9383	1839	3381.8	2027

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

- จากการทดลอง ค่า Delay จะแบ่งเป็น 2 ฟังก์ชัน ฟังก์ชันแรก 180-90 องศา, ที่ 180 องศา ค่า Delay จะมีค่ามากที่สุด และจะลดลงเมื่อองศาเข้าใกล้ 90 องศา ซึ่งค่า Delay ที่จุดนี้จะมีค่าน้อยที่สุด
- ฟังก์ชัน 90-0 องศา Delay มีอัตราเพิ่มขึ้น โดยที่ 80 องศา จะเริ่มมีการแปรปรวนของการตรวจจับ กล่าวคือ บางครั้ง ไมโครโฟนที่ 180 องศาจะตรวจจับเสียงได้ก่อน บางครั้ง ไมโครโฟน ที่ 0 องศาจะตรวจจับเสียงได้ก่อน และที่ 0 องศา ค่า Delay มีค่าสูงสุด

วิเคราะห์ผลการทดลอง

- ผลที่ได้เป็นไปตามที่คาดการณ์ไว้ กล่าวคือ ที่ 0 และ 180 องศา จะเป็นจุดที่เสียงระยะเดินทางของเสียงต่างกันมากที่สุด และที่ 80 -100 องศา (บริเวณด้านหน้ากล้อง) จะเป็นจุดที่ ค่า Delay มีค่าน้อยที่สุด เพราะเป็นบริเวณที่ ไมโครโฟนทั้งสองมีระยะห่างจากต้นกำเนิดเสียงใกล้เคียงกัน
- ปัญหาที่ผลคือ ระยะห่าง, ความถี่, ความดังของต้นกำเนิดเสียง ล้วนแล้วแต่มีผลต่อความเร็วของเสียง และการตอบสนองของ ไมโครโฟน ดังนั้นจึงต้อง Fix ระยะห่าง และลักษณะของต้นกำเนิดเสียง ถึงอย่างไรก็ตามความแปรปรวนก็ยังคงสูงอยู่ดี เพราะควบคุมสภาพแวดล้อมไม่ได้
- ความชื้น อุณหภูมิ ความหนาแน่นของอากาศ และเสียงรบกวน ล้วนแล้วแต่มีผลต่อความเร็วของเสียง ซึ่งควบคุมได้ยาก
- ที่ 80 องศา มีการแปรปรวนของทิศทางการตรวจจับเสียงของ ไมโครโฟนมากที่สุด คั้งที่ได้กล่าวไว้ ทั้งนี้เป็นผลจากการ ปรับค่า Voltage Reference ของ Op-Amp ที่ไม่เท่ากัน ทำให้จุดที่แปรปรวนมากที่สุด ไม่อยู่ที่ 90 องศา
- ปัญหาทั้งหมดแก้ไขได้โดยการปรับค่า Membership Function ในการควบคุมของ Fuzzy Logic

บทที่ 9

สรุปและวิจารณ์

จากการพัฒนาชิ้นงานทั้งหมดถึงแม้ว่าจะสามารถทำงานได้ตามที่ต้องการ แต่ยังคงประสบกับปัญหาอีกหลายปัญหาที่ยังไม่สามารถแก้ไขได้สมบูรณ์ ดังนี้

8.1 ไม่สามารถใช้กับเสียงพูดโดยทั่วไปของมนุษย์ได้ เพราะเสียงพูดของมนุษย์ประกอบด้วยความถี่หลาย Harmonics ซึ่งส่งผลต่อความถี่และความเร็วทำให้การวัดค่า Delay เพื่อค้นหาทิศทางนั้นไม่เพียงพอ

8.2 ค่า Delay ที่วัดได้มีความแปรปรวนสูงมาก เนื่องจากสภาพแวดล้อมที่ไม่คงที่ ปัจจัยต่างๆไม่ว่าจะเป็น ความชื้น อุณหภูมิ ความหนาแน่นของอากาศ และเสียงรบกวน ล้วนแล้วแต่มีผลต่อความเร็วของเสียง ซึ่งควบคุมได้ยาก

8.3 ที่มุม 150-170 องศา ค่า Delay มีความแปรปรวนสูงมากและไม่เป็นไปตามทฤษฎี กล่าวคือแทนที่ค่า Delay จะมีค่าเพิ่มขึ้น แต่กลับกลายเป็นลดลงก่อน แล้วค่อยเพิ่มขึ้น ถ้ามาดูที่ Mechanics จะพบว่าที่จุดนี้ จะมีเฟืองขวางไมโครโฟนที่ 0 องศาอยู่ ทำให้เสียงเกิดการสะท้อน การวัดค่า Delay จึงผิดพลาด

8.4 การเพิ่มไมโครโฟนมากกว่า 2 ตัว จากการทดลองขั้นต้นพบว่า สามารถระบุทิศทางคร่าวๆของต้นกำเนิดเสียงได้ดี แต่ค่า Delay ที่วัดได้แปรปรวนมากจนไม่สามารถนำมาใช้ได้ อาจเกิดเนื่องจากเหตุผลจากข้อ 8.2 จากการใช้ Input ซึ่งมีความแปรปรวนมากอยู่แล้ว เมื่อเพิ่มจำนวน Input ความแปรปรวนก็ยิ่งเพิ่มมากขึ้นเป็นเงาตามตัว

8.5 กฎที่ใช้ 5 กฎได้มาจากการประมาณมุมมองของกล้องที่สามารถจับได้คือ 38 องศา เทียบกับองศาที่ตรวจจับทั้งหมด คือ 180 องศา ซึ่งยังไม่ครอบคลุมค่าที่เป็นไปได้ดีเพียงพอ

8.6 การหมุนของกล้องยังไม่ดีพอ ซึ่งการใช้ VR Feedback เมื่อใช้ไปนานๆอาจเสื่อมสภาพได้ ในแง่ของความแม่นยำและความคงทนยังยอมรับไม่ได้

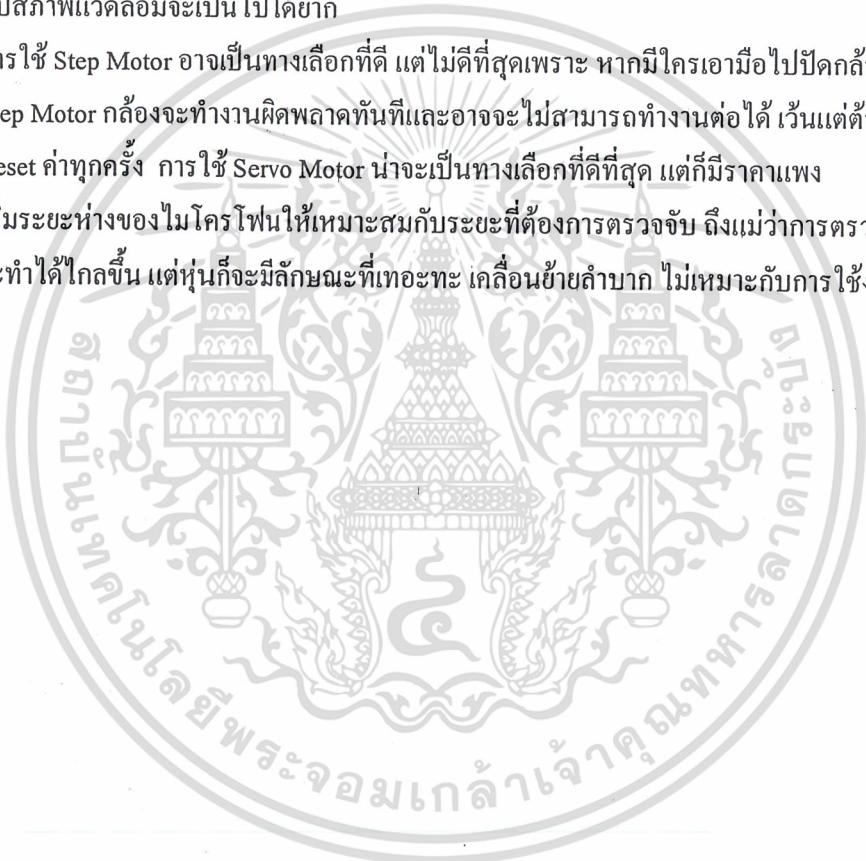
8.7 การตรวจจับสามารถทำได้แค่ระยะไม่เกิน 1 เมตร ไม่สามารถตรวจจับได้ไกลกว่านั้น

ซึ่งปัญหาในแต่ละข้อจะขอเสนอแนวทางแก้ไขดังนี้

- เพิ่ม Input ที่เหมาะสม เช่น ค่าความดัง เป็น Input ที่ดีอีกหนึ่ง เพราะนอกจากจะสามารถมาเทียบกับ ค่า Delay แล้ว ยังอาจื่อนำมาตรวจระยะห่างของแหล่งกำเนิดเสียงได้ด้วย เพราะความดังของเสียงจะถูกลดทอนลงเมื่อระยะไกลขึ้น
- ต้องมีการทดสอบและปรับค่าที่ดี อย่างในชิ้นงานนี้ วงจรเสียง Op-Amp ตัวที่สองจะใช้ VR 5 KOhm แบบ 25 รอบ ถึงจะปรับค่าให้ไมโครโฟนสองตัวมีความไวในการรับเสียงใกล้เคียงกัน และถ้าปรับค่าการรับเสียงต่ำเกินไป วงจรจะตรวจจับเสียงได้ทุกเสียงแม้แต่เสียงค้อยมากๆ ก็ตาม ซึ่งจะไม่มีการใช้ในที่ที่มีเสียงดังได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การวางไมโครโฟน ควรวางในจุดที่โล่ง ปราศจากสิ่งกีดขวาง และมีความสมมาตร จึงจะสามารถวัดค่าที่เกิดขึ้นได้อย่างสมบูรณ์
- การเพิ่มจำนวนไมโครโฟนจะดีในแง่ของการตรวจจับคร่าวๆ แต่การตรวจจับแบบละเอียดจะลำบาก เพราะต้องปรับค่าให้ ไมโครโฟนให้มีความไวเท่ากันหมด สมมุติว่ามีไมโครโฟน 3 ตัว ไมโครโฟน 1 ต้องไวเท่ากับไมโครโฟน 2 , ไมโครโฟน 2 ต้องไวเท่ากับไมโครโฟน 3 และ ไมโครโฟน 3 ต้องไวเท่ากับไมโครโฟน 1 ซึ่งในทางปฏิบัติไม่ใช่เรื่องง่ายเลย เพราะนอกจากจะต้องปรับค่าแล้ว การควบคุมสภาพแวดล้อมในการทดลองให้คงที่ก็ยิ่งลำบาก
- เพิ่มกฎ เป็น 7 กฎ จะเหมาะสมและครอบคลุมลักษณะที่เป็นไปได้มากที่สุด ตลอดจนโปรแกรมที่ใช้ต้องปรับค่า Membership Function ได้อย่างอิสระ มิเช่นนั้นการปรับค่าให้เข้ากับสภาพแวดล้อมจะเป็นไปได้ยาก
- การใช้ Step Motor อาจเป็นทางเลือกที่ดี แต่ไม่ดีที่สุดเพราะ หากมีใครเอามือไปปิดกั้น Step Motor ก็นឹងทำงานผิดพลาดทันทีและอาจจะไม่สามารถทำงานต่อได้ เว้นแต่ต้อง Reset ค่าทุกครั้ง การใช้ Servo Motor น่าจะเป็นทางเลือกที่ดีที่สุด แต่ก็มีราคาแพง
- เพิ่มระยะห่างของไมโครโฟนให้เหมาะสมกับระยะที่ต้องการตรวจจับ ถึงแม้ว่าการตรวจจับจะทำได้ไกลขึ้น แต่หุ่นก็จะมีลักษณะที่เทอะทะ เคลื่อนย้ายลำบาก ไม่เหมาะกับการใช้งาน



บรรณานุกรม

1. ชัยวัฒน์ ลิ่มพรจิตรวิไล,วรพจน์ กรแก้ววัฒนกุล “ เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ MCS-51 Flash Microcontroller “ บ. Inex innovative Experiment
2. ชัยวัฒน์ ลิ่มพรจิตรวิไล,วรพจน์ กรแก้ววัฒนกุล “ เรียนรู้และปฏิบัติการเชื่อมต่อ คอมพิวเตอร์กับอุปกรณ์ภายนอก ผ่านพอร์ตอนุกรม “ บ. Inex innovative Experiment
3. นาย พรชัยศ ศรีปัญญาพงศ์ “ เอกสารประกอบการอบรมเรื่อง ไมโครคอนโทรลเลอร์ 8051 “ ศูนย์บริการและพัฒนาวิศวกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร , 2542
4. กิตติ ภัคดีวัฒนกุล “ Visual Basic 6 ฉบับโปรแกรมเมอร์ “ บ. เคทีพี คอมพ์ แอนด์ คอนซัลท์,พิมพ์ครั้งที่ 3 , กรกฎาคม 2542
5. ธนกร ศิริพิทักษ์,“ทฤษฎีเครื่องเสียง audio amplifier”ฝ่ายวิชาการ บ.สกายมิกส์ จำกัด, พิมพ์ครั้งที่ 2 ตุลาคม 2542



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการหุ่นยนต์ควบคุมด้วยระบบพีซีที่ล่อจิก และรายงานประกอบฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี ก็เพราะได้รับการแนะนำและสั่งสอนมาจาก รศ.ดร. กิตติ ไพบูลย์วัฒนกิจ อาจารย์ที่ปรึกษาของโครงการ รวมทั้งคณาจารย์ภาควิชา อิเล็กทรอนิกส์ทุกท่าน ที่ได้ถ่ายทอดวิชาความรู้ต่างๆ ให้คำปรึกษา คำแนะนำ และความช่วยเหลืออยู่ตลอดเวลา

ขอขอบคุณพี่ๆ เพื่อนๆ น้องๆ ทุกคนในชุมชน Robot ที่ได้ช่วยเหลือ ให้ความสนับสนุน ทั้งด้านอุปกรณ์ต่างๆ และสถานที่ในการทำโครงการ ให้คำแนะนำเวลามีปัญหา รวมทั้งกำลังใจที่อยู่ตลอดเวลา อันได้แก่ พี่เต้ย , ชิว (อดีตประธานชุมชน) , วิม (อดีตเลขาฯ) , โย , น้องเท็น (ประธานคนปัจจุบัน) , น้องเบ็กซ์ , น้องอ้ม , น้องก้อง , น้องมุข , น้องคิด

และสุดท้ายนี้ขอขอบคุณ อาจารย์ทุกท่านที่ได้ช่วยกันประสิทธิ์ประสาทวิชาความรู้ให้แก่พวกเรา นับตั้งแต่เรายังเป็นเพียงเด็กตัวเล็กๆ ซึ่งไม่มีความรู้อะไรมากไปกว่า แค่ว่า “เราต้องการอะไร” จวบจนถึงวันนี้ วันที่เราต้องถามตัวเองแล้วว่า ด้วยความรู้ที่เรามีอยู่นี้ “เราจะต้องทำอะไร ? ”

ขอบคุณคุณพ่อ คุณแม่ ที่ให้กำเนิดเรามา ขอขอบคุณน้อง ที่คอยเป็นกำลังใจให้กับเรา ขอขอบคุณเพื่อนๆ ที่คอยให้คำปรึกษา ขอขอบคุณ 7Eleven ที่ช่วยให้เราคลายหิว ขอขอบคุณเพลงดีๆ ที่ช่วยคลายความเบื่อหน่าย ขอขอบคุณห้องฟ้า ที่ยังสวยงาม รอให้เราเฝ้ามองอยู่เสมอ และที่สำคัญ ขอขอบคุณอะไรก็ตามที่ทำให้ผม “ มีความคิด ”.

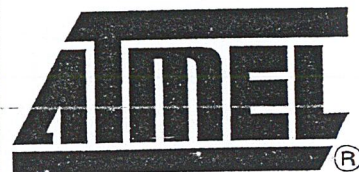
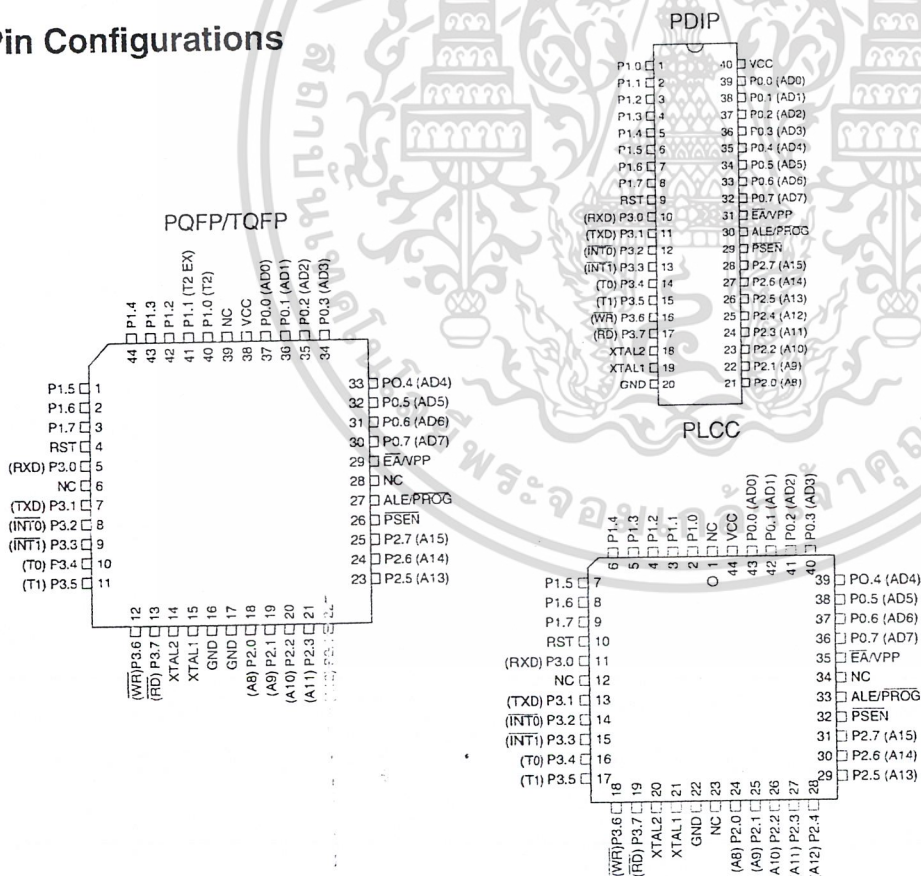
Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations



8-bit Microcontroller with 4K Bytes Flash

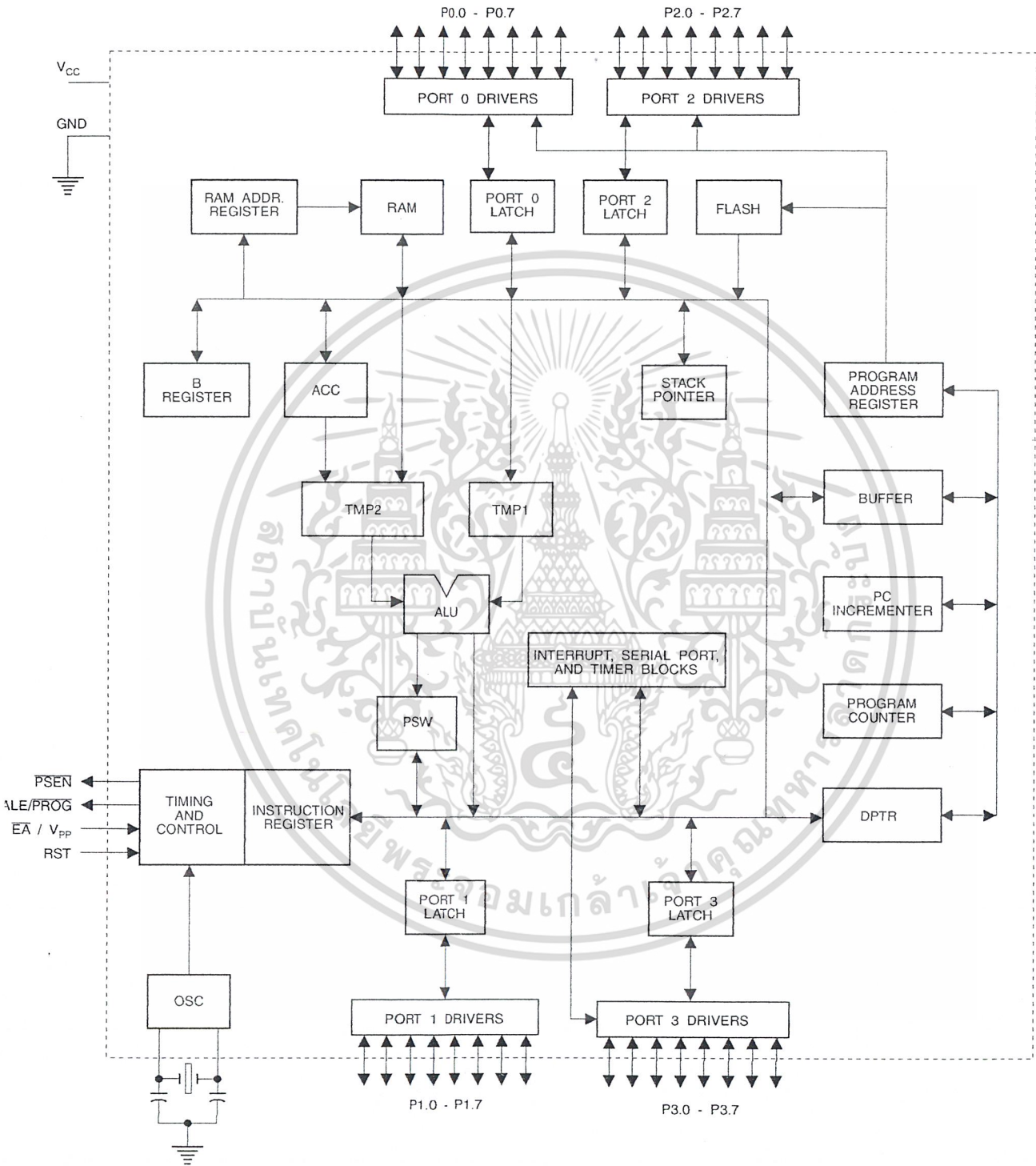
AT89C51

Rev. 0265G-02/00



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs,

Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE



pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}/\text{VPP}$

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP} .

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left

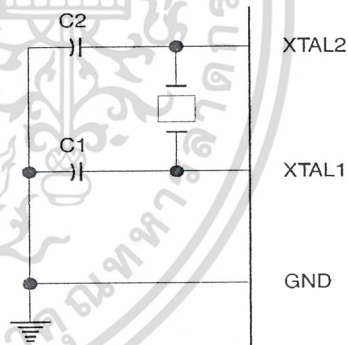
unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections

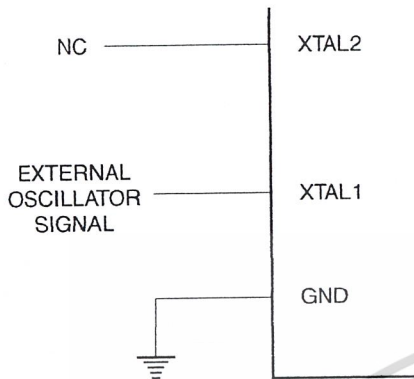


Note: C1, C2 = 30 pF \pm 10 pF for Crystals
= 40 pF \pm 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Regis-

ters retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of \overline{EA} be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, \overline{EA} is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) = FFH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse $\overline{ALE}/\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address

and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features \overline{Data} Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding $\overline{ALE}/\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 51H indicates 89C51
- (032H) = FFH indicates 12V programming
- (032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

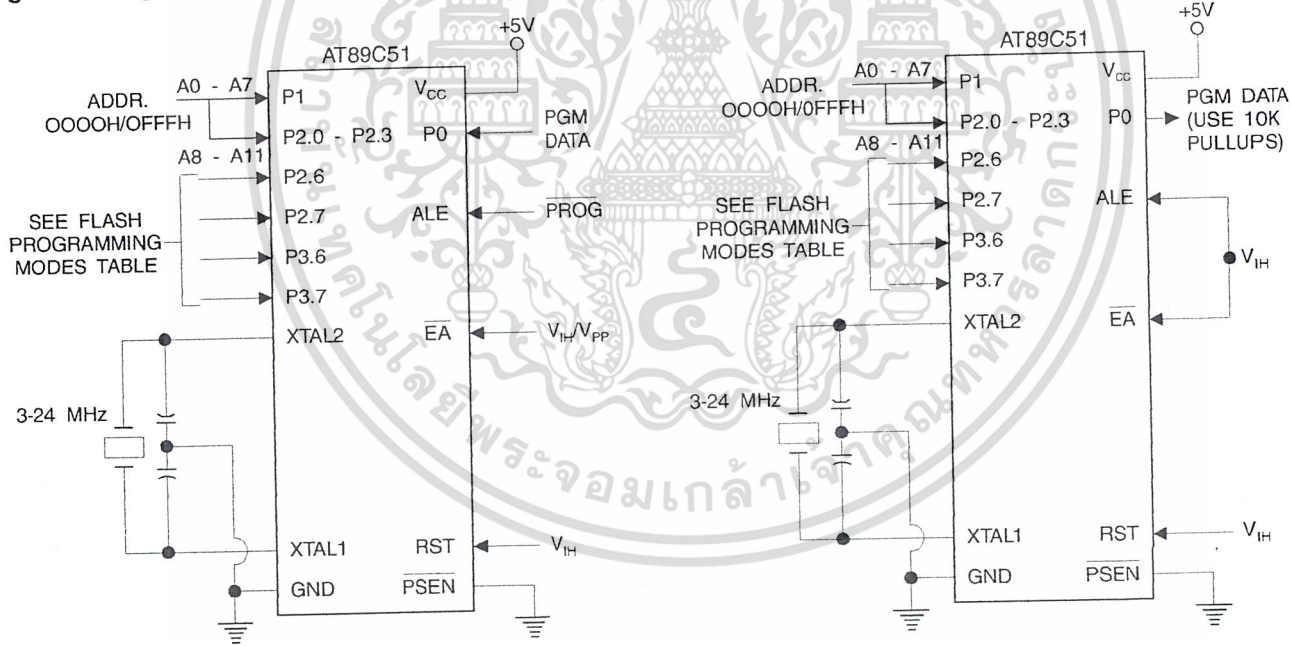
Flash Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	H	L		H/12V	H	H	H	H
				H/12V	H	H	L	L
				H/12V	H	L	H	L
Chip Erase	H	L	(1)	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

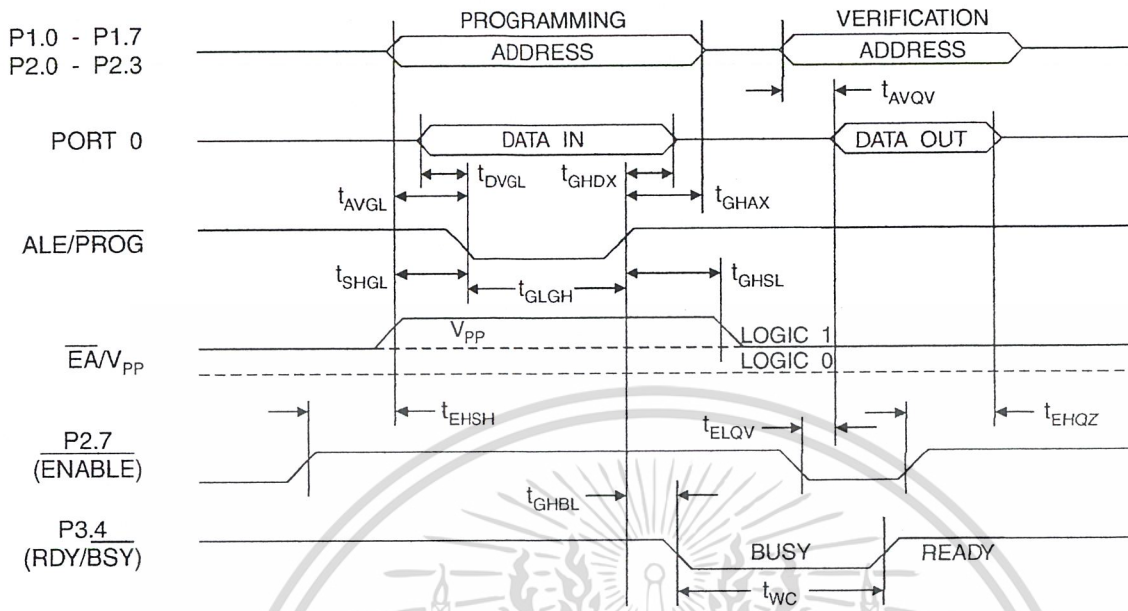
Figure 3. Programming the Flash

Figure 4. Verifying the Flash

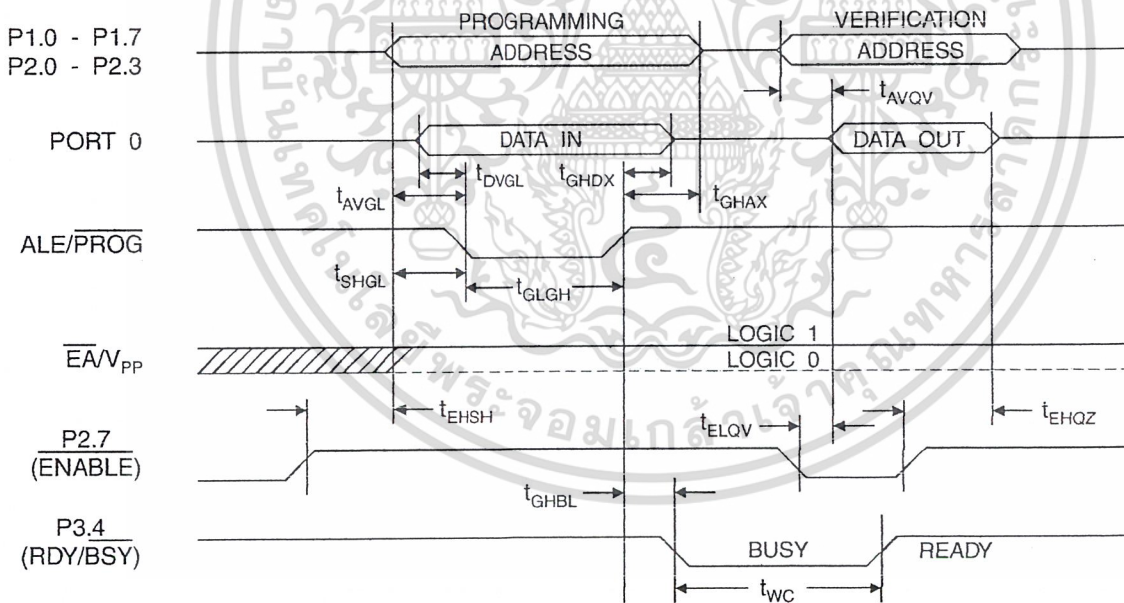


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High-voltage Mode ($V_{PP} = 12V$)



Flash Programming and Verification Waveforms - Low-voltage Mode ($V_{PP} = 5V$)



Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V _{PP} ⁽¹⁾	Programming Enable Voltage	11.5	12.5	V
I _{PP} ⁽¹⁾	Programming Enable Current		1.0	mA
1/t _{CLCL}	Oscillator Frequency	3	24	MHz
t _{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
t _{GHAX}	Address Hold After $\overline{\text{PROG}}$	48t _{CLCL}		
t _{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
t _{GHDX}	Data Hold After $\overline{\text{PROG}}$	48t _{CLCL}		
t _{EHS}	P2.7 (ENABLE) High to V _{PP}	48t _{CLCL}		
t _{SHGL}	V _{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t _{GHSL} ⁽¹⁾	V _{PP} Hold After $\overline{\text{PROG}}$	10		μs
t _{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t _{AVQV}	Address to Data Valid		48t _{CLCL}	
t _{ELQV}	ENABLE Low to Data Valid		48t _{CLCL}	
t _{EHQZ}	Data Float After ENABLE	0	48t _{CLCL}	
t _{GHBL}	$\overline{\text{PROG}}$ High to BUSY Low		1.0	μs
t _{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
DC Output Current.....	15.0 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V _{IL}	Input Low-voltage	(Except EA)	-0.5	0.2 V _{CC} - 0.1	V
V _{IL1}	Input Low-voltage (EA)		-0.5	0.2 V _{CC} - 0.3	V
V _{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V _{CC} + 0.9	V _{CC} + 0.5	V
V _{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V _{CC}	V _{CC} + 0.5	V
V _{OL}	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	I _{OL} = 1.6 mA		0.45	V
V _{OL1}	Output Low-voltage ⁽¹⁾ (Port 0, ALE, PSEN)	I _{OL} = 3.2 mA		0.45	V
V _{OH}	Output High-voltage (Ports 1,2,3, ALE, PSEN)	I _{OH} = -60 μA, V _{CC} = 5V ± 10%	2.4		V
		I _{OH} = -25 μA	0.75 V _{CC}		V
		I _{OH} = -10 μA	0.9 V _{CC}		V
V _{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I _{OH} = -800 μA, V _{CC} = 5V ± 10%	2.4		V
		I _{OH} = -300 μA	0.75 V _{CC}		V
		I _{OH} = -80 μA	0.9 V _{CC}		V
I _{IL}	Logical 0 Input Current (Ports 1,2,3)	V _{IN} = 0.45V		-50	μA
I _{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V _{IN} = 2V, V _{CC} = 5V ± 10%		-650	μA
I _{LI}	Input Leakage Current (Port 0, EA)	0.45 < V _{IN} < V _{CC}		±10	μA
RRST	Reset Pull-down Resistor		50	300	KΩ
C _{IO}	Pin Capacitance	Test Freq. = 1 MHz, T _A = 25°C		10	pF
I _{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode ⁽²⁾	V _{CC} = 6V		100	μA
		V _{CC} = 3V		40	μA

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

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port: Port 0: 26 mA

Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.

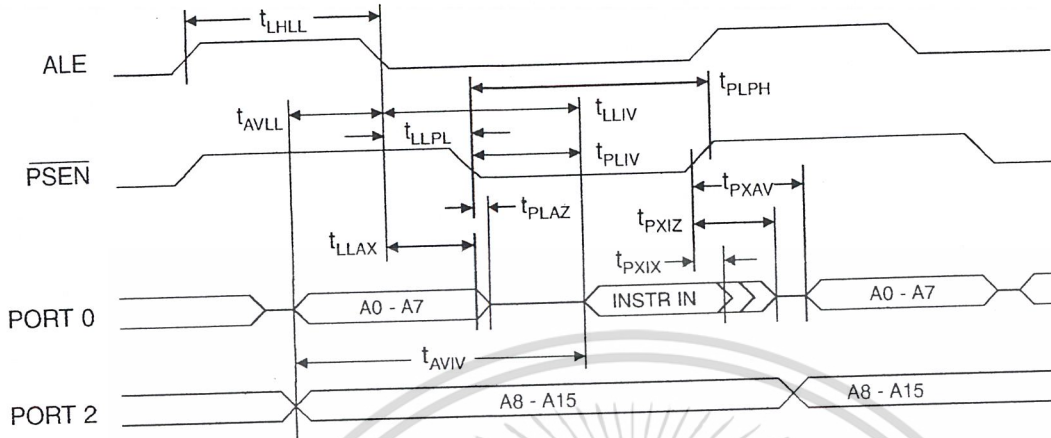
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

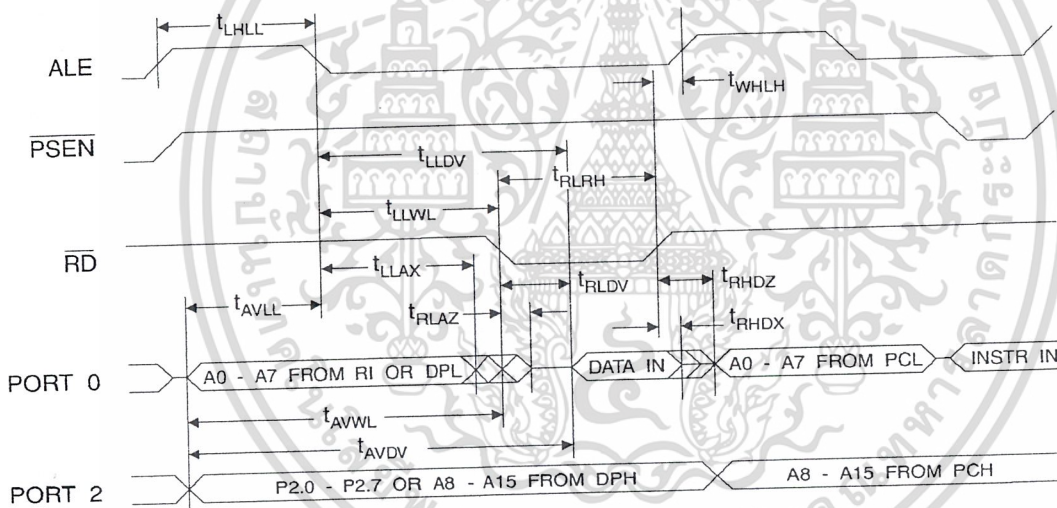
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{\text{CLCL}}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{\text{CLCL}}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
t_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-13$		ns
t_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-20$		ns
t_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-10$	ns
t_{PXAV}	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-55$	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{WLWH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
t_{RHDX}	Data Hold After $\overline{\text{RD}}$	0		0		ns
t_{RHDX}	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
t_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-20$		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-120$		ns
t_{WHQX}	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-20$		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-20$	$t_{\text{CLCL}}+25$	ns

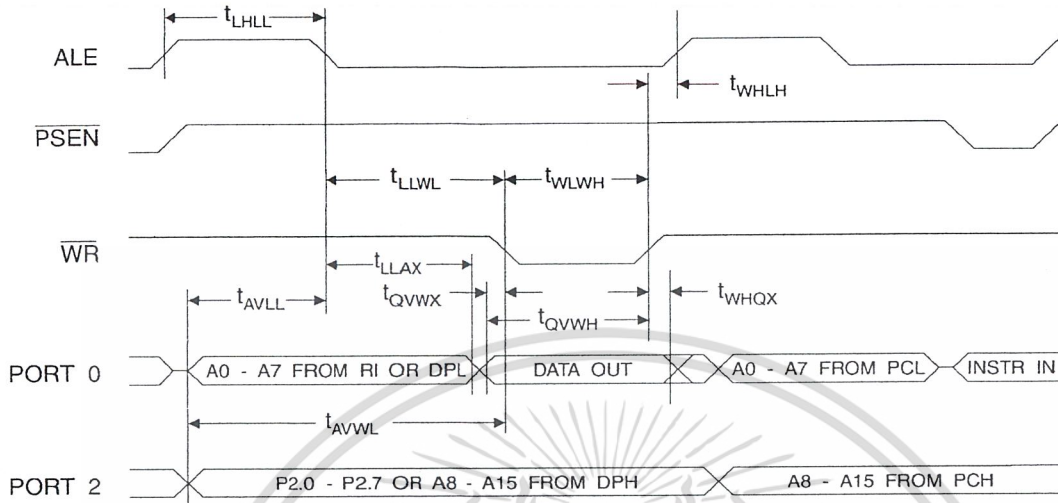
External Program Memory Read Cycle



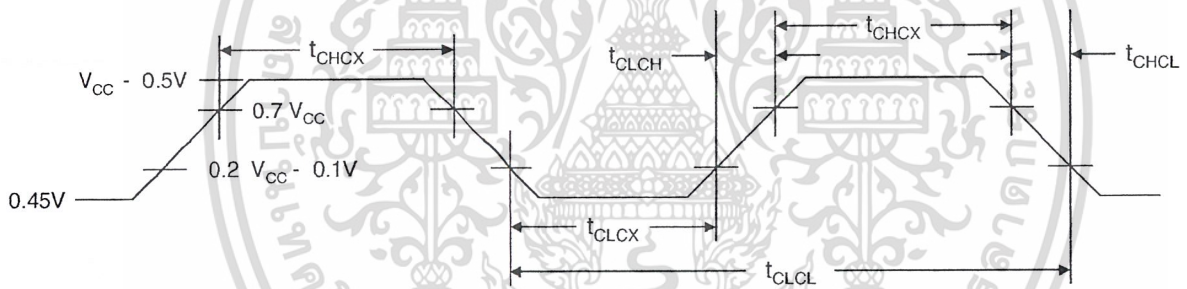
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

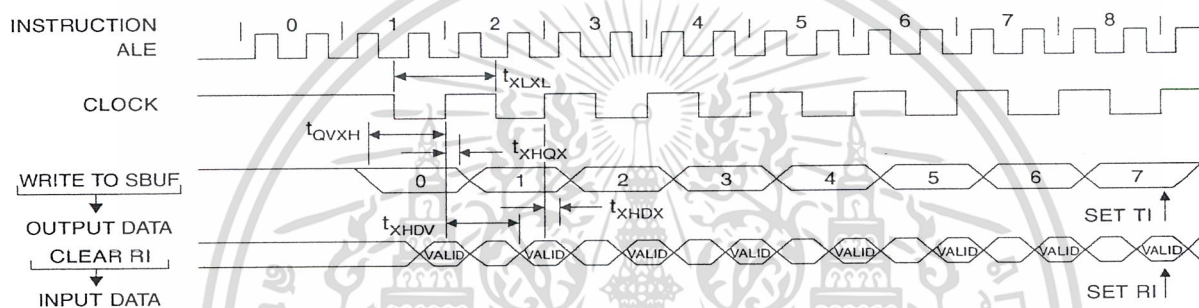


Serial Port Timing: Shift Register Mode Test Conditions

($V_{CC} = 5.0\text{ V} \pm 20\%$; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾

Float Waveforms⁽¹⁾



Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5\text{V}$ for a logic 1 and 0.45V for a logic 0. Timing measurements are made at $V_{IH\text{ min}}$ for a logic 1 and $V_{IL\text{ max}}$ for a logic 0.

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

ADC0802, ADC0803 ADC0804

8-Bit, Microprocessor-
Compatible, A/D Converters

August 1997

Features

- 80C48 and 80C80/85 Bus Compatible - No Interfacing Logic Required
- Conversion Time < 100µs
- Easy Interface to Most Microprocessors
- Will Operate in a "Stand Alone" Mode
- Differential Analog Voltage Inputs
- Works with Bandgap Voltage References
- TTL Compatible Inputs and Outputs
- On-Chip Clock Generator
- 0V to 5V Analog Voltage Input Range (Single + 5V Supply)
- No Zero-Adjust Required

Description

The ADC0802 family are CMOS 8-Bit, successive-approximation A/D converters which use a modified potentiometric ladder and are designed to operate with the 8080A control bus via three-state outputs. These converters appear to the processor as memory locations or I/O ports, and hence no interfacing logic is required.

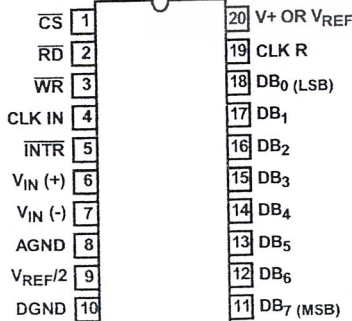
The differential analog voltage input has good common-mode-rejection and permits offsetting the analog zero-input-voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Ordering Information

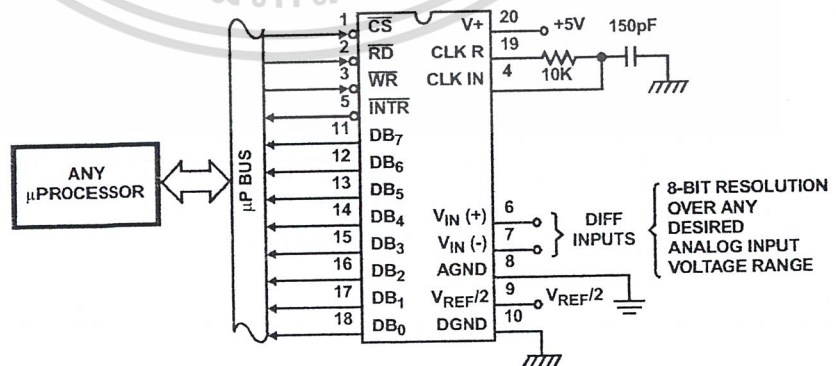
PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (°C)	PACKAGE	PKG. NO
ADC0802LCN	±1/2 LSB	V _{REF/2} = 2.500V _{DC} (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0802LCD	±3/4 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0802LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0803LCN	±1/2 LSB	V _{REF/2} Adjusted for Correct Full Scale Reading	0 to 70	20 Ld PDIP	E20.3
ADC0803LCD	±3/4 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0803LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3
ADC0803LD	±1 LSB	-55 to 125	20 Ld CERDIP	F20.3	
ADC0804LCN	±1 LSB	V _{REF/2} = 2.500V _{DC} (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0804LCD	±1 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0804LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3

Pinout

ADC0802, ADC0803, ADC0804
(PDIP, CERDIP)
TOP VIEW



Typical Application Schematic



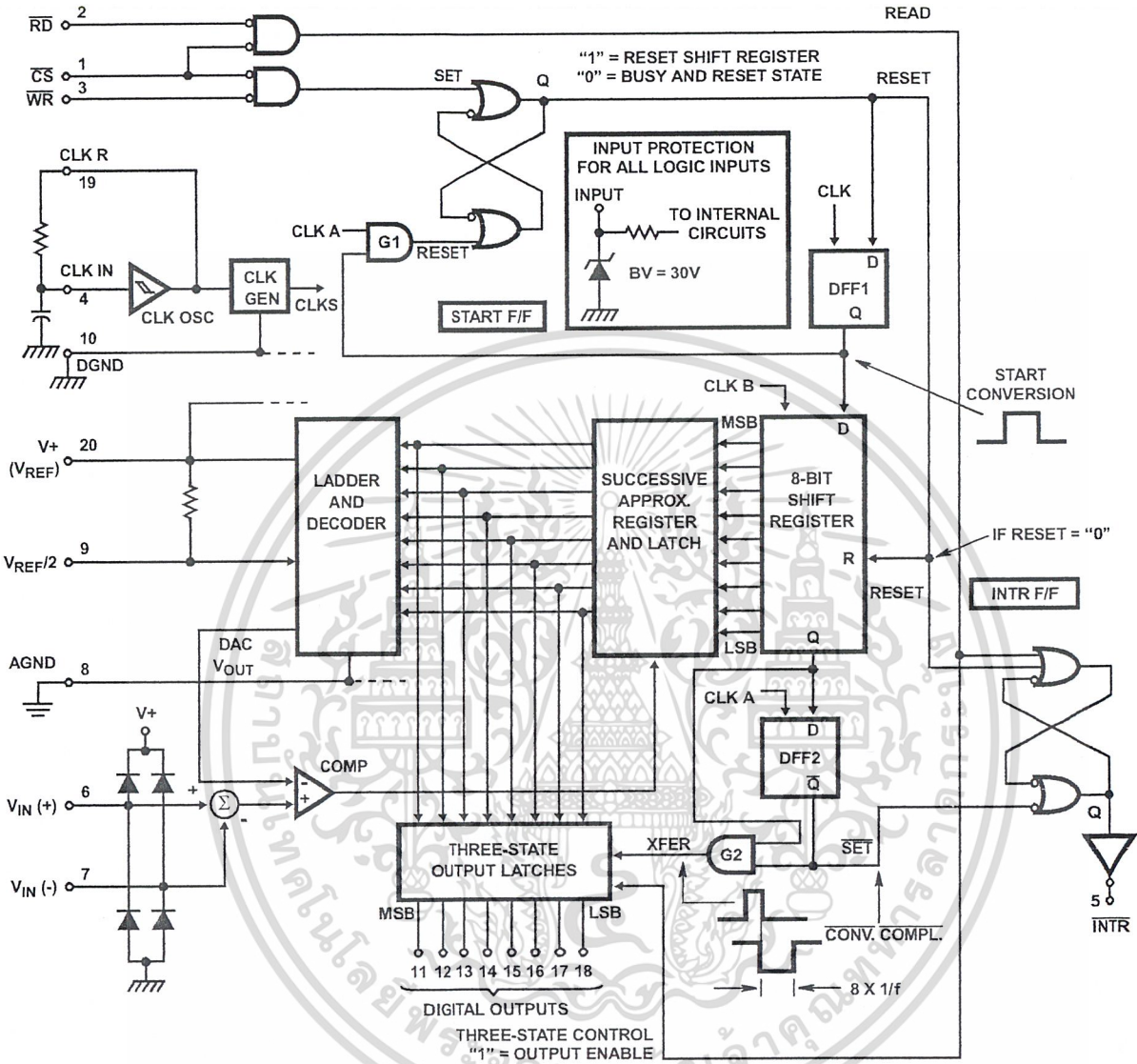
CAUTION: These devices are sensitive to electrostatic discharge; follow proper IC Handling Procedures.

File Number **3094.1**

1-888-INTERSIL or 321-724-7143 | Copyright © Intersil Corporation 1999

ADC0802, ADC0803, ADC0804

Functional Diagram



ADC0802, ADC0803, ADC0804

Absolute Maximum Ratings

Supply Voltage 6.5V
Voltage at Any Input -0.3V to (V⁺ + 0.3V)

Operating Conditions

Temperature Range
ADC0802/03LD -55°C to 125°C
ADC0802/03/04LCD -40°C to 85°C
ADC0802/03/04LCN 0°C to 70°C
ADC0803/04LCWM -40°C to 85°C

Thermal Information

Thermal Resistance (Typical, Note 1)	θ_{JA} (°C/W)	θ_{JC} (°C/W)
PDIP Package	125	N/A
CERDIP Package	80	20
SOIC Package	120	N/A
Maximum Junction Temperature		
Hermetic Package	175°C	
Plastic Package	150°C	
Maximum Storage Temperature Range	-65°C to 150°C	
Maximum Lead Temperature (Soldering, 10s)	300°C (SOIC - Lead Tips Only)	

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

- θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

Electrical Specifications (Notes 1, 7)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS V ⁺ = 5V, T _A = 25°C and f _{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V _{REF/2} = 2.500V	-	-	±1/2	LSB
ADC0803	V _{REF/2} Adjusted for Correct Full Scale Reading	-	-	±1/2	LSB
ADC0804	V _{REF/2} = 2.500V	-	-	±1	LSB
V _{REF/2} Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V ⁺) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	±1/16	±1/8	LSB
Power Supply Sensitivity	V ⁺ = 5V ±10% Over Allowed Input Voltage Range	-	±1/16	±1/8	LSB
CONVERTER SPECIFICATIONS V ⁺ = 5V, 0°C to 70°C and f _{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V _{REF/2} = 2.500V	-	-	±1/2	LSB
ADC0803	V _{REF/2} Adjusted for Correct Full Scale Reading	-	-	±1/2	LSB
ADC0804	V _{REF/2} = 2.500V	-	-	±1	LSB
V _{REF/2} Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V ⁺) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	±1/8	±1/4	LSB
Power Supply Sensitivity	V ⁺ = 5V ±10% Over Allowed Input Voltage Range	-	±1/16	±1/8	LSB
CONVERTER SPECIFICATIONS V ⁺ = 5V, -25°C to 85°C and f _{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	V _{REF/2} = 2.500V	-	-	±3/4	LSB
ADC0803	V _{REF/2} Adjusted for Correct Full Scale Reading	-	-	±3/4	LSB
ADC0804	V _{REF/2} = 2.500V	-	-	±1	LSB
V _{REF/2} Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V ⁺) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	±1/8	±1/4	LSB
Power Supply Sensitivity	V ⁺ = 5V ±10% Over Allowed Input Voltage Range	-	±1/16	±1/8	LSB

Electrical Specifications (Notes 1, 7) (Continued)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS $V_+ = 5V$, $-55^{\circ}C$ to $125^{\circ}C$ and $f_{CLK} = 640kHz$, Unless Otherwise Specified					
Total Unadjusted Error					
ADC0802	$V_{REF}/2 = 2.500V$	-	-	± 1	LSB
ADC0803	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	-	-	± 1	LSB
$V_{REF}/2$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	k Ω
Analog Input Voltage Range	(Note 2)	GND-0.05	-	$(V_+) + 0.05$	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm 1/8$	$\pm 1/4$	LSB
Power Supply Sensitivity	$V_+ = 5V \pm 10\%$ Over Allowed Input Voltage Range	-	$\pm 1/8$	$\pm 1/4$	LSB
AC TIMING SPECIFICATIONS $V_+ = 5V$, and $T_A = 25^{\circ}C$, Unless Otherwise Specified					
Clock Frequency, f_{CLK}	$V_+ = 6V$ (Note 3)	100	640	1280	kHz
	$V_+ = 5V$	100	640	800	kHz
Clock Periods per Conversion (Note 4), t_{CONV}		62	-	73	Clocks/Conv
Conversion Rate In Free-Running Mode, CR	\overline{INTR} tied to \overline{WR} with $\overline{CS} = 0V$, $f_{CLK} = 640kHz$	-	-	8888	Conv/s
Width of \overline{WR} Input (Start Pulse Width), $t_{W(WR)}$	$\overline{CS} = 0V$ (Note 5)	100	-	-	ns
Access Time (Delay from Falling Edge of \overline{RD} to Output Data Valid), t_{ACC}	$C_L = 100pF$ (Use Bus Driver IC for Larger C_L)	-	135	200	ns
Three-State Control (Delay from Rising Edge of \overline{RD} to HI-Z State), t_{1H} , t_{0H}	$C_L = 10pF$, $R_L = 10K$ (See Three-State Test Circuits)	-	125	250	ns
Delay from Falling Edge of \overline{WR} to Reset of \overline{INTR} , t_{WI} , t_{RI}		-	300	450	ns
Input Capacitance of Logic Control Inputs, C_{IN}		-	5	-	pF
Three-State Output Capacitance (Data Buffers), C_{OUT}		-	5	-	pF
DC DIGITAL LEVELS AND DC SPECIFICATIONS $V_+ = 5V$, and T_{MIN} to T_{MAX} , Unless Otherwise Specified					
CONTROL INPUTS (Note 6)					
Logic "1" Input Voltage (Except Pin 4 CLK IN), V_{INH}	$V_+ = 5.25V$	2.0	-	V_+	V
Logic "0" Input Voltage (Except Pin 4 CLK IN), V_{INL}	$V_+ = 4.75V$	-	-	0.8	V
CLK IN (Pin 4) Positive Going Threshold Voltage, V_{+CLK}		2.7	3.1	3.5	V
CLK IN (Pin 4) Negative Going Threshold Voltage, V_{-CLK}		1.5	1.8	2.1	V
CLK IN (Pin 4) Hysteresis, V_H		0.6	1.3	2.0	V
Logic "1" Input Current (All Inputs), I_{INH}	$V_{IN} = 5V$	-	0.005	1	μA
Logic "0" Input Current (All Inputs), I_{INLO}	$V_{IN} = 0V$	-1	-0.005	-	μA
Supply Current (Includes Ladder Current), I_+	$f_{CLK} = 640kHz$, $T_A = 25^{\circ}C$ and $\overline{CS} = HI$	-	1.3	2.5	mA
DATA OUTPUTS AND \overline{INTR}					
Logic "0" Output Voltage, V_{OL}	$I_O = 1.6mA$, $V_+ = 4.75V$	-	-	0.4	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADC0802, ADC0803, ADC0804

Electrical Specifications (Notes 1, 7) (Continued)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Logic "1" Output Voltage, V_{OH}	$I_O = -360\mu A$, $V_+ = 4.75V$	2.4	-	-	V
Three-State Disabled Output Leakage (All Data Buffers), I_{LO}	$V_{OUT} = 0V$	-3	-	-	μA
	$V_{OUT} = 5V$	-	-	3	μA
Output Short Circuit Current, I_{SOURCE}	V_{OUT} Short to Gnd $T_A = 25^\circ C$	4.5	6	-	mA
Output Short Circuit Current, I_{SINK}	V_{OUT} Short to V_+ $T_A = 25^\circ C$	9.0	16	-	mA

NOTES:

- All voltages are measured with respect to GND, unless otherwise specified. The separate AGND point should always be wired to the DGND, being careful to avoid ground loops.
- For $V_{IN(-)} \geq V_{IN(+)}$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see Block Diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_+ supply. Be careful, during testing at low V_+ levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct - especially at elevated temperatures, and cause errors for analog inputs near full scale. As long as the analog V_{IN} does not exceed the supply voltage by more than 50mV, the output code will be correct. To achieve an absolute 0V to 5V input voltage range will therefore require a minimum supply voltage of 4.950V over temperature variations, initial tolerance and loading.
- With $V_+ = 6V$, the digital logic interfaces are no longer TTL compatible.
- With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process.
- The \overline{CS} input is assumed to bracket the \overline{WR} strobe input so that timing is dependent on the \overline{WR} pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the \overline{WR} pulse (see Timing Diagrams).
- CLK IN (pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately.
- None of these A/Ds requires a zero-adjust. However, if an all zero code is desired for an analog input other than 0V, or if a narrow full scale span exists (for example: 0.5V to 4V full scale) the $V_{IN(-)}$ input can be adjusted to achieve this. See the Zero Error description in this data sheet.

Timing Waveforms

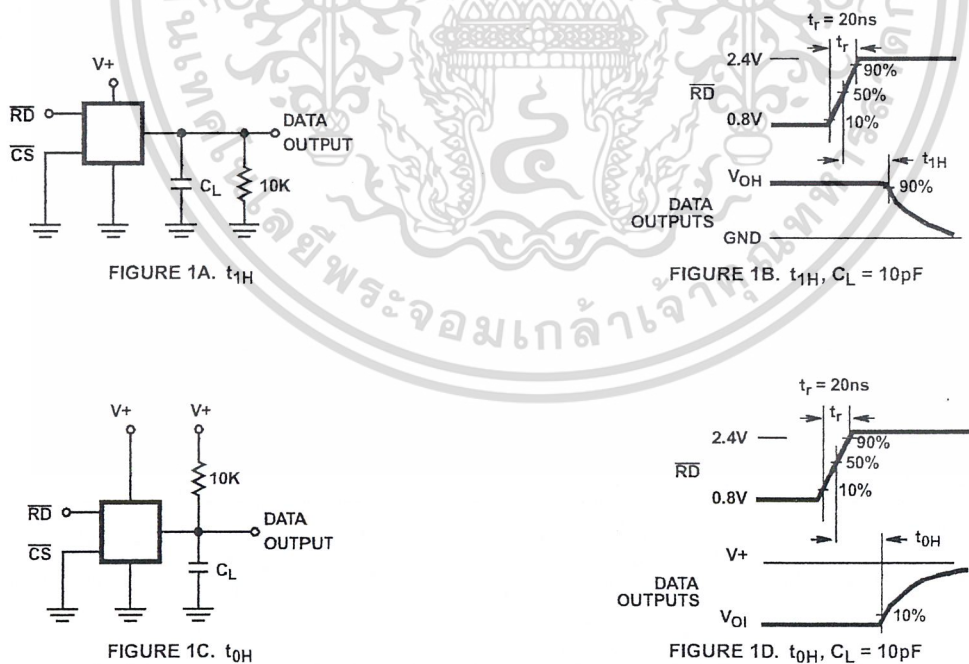


FIGURE 1. THREE-STATE CIRCUITS AND WAVEFORMS

Typical Performance Curves

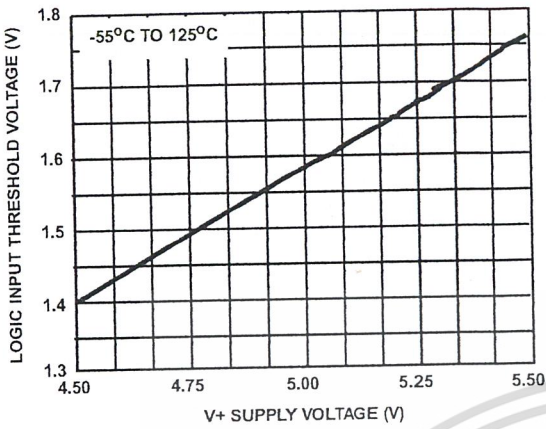


FIGURE 2. LOGIC INPUT THRESHOLD VOLTAGE vs SUPPLY VOLTAGE

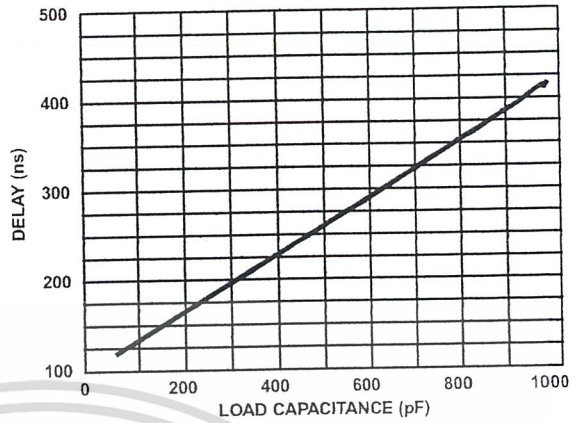


FIGURE 3. DELAY FROM FALLING EDGE OF RD TO OUTPUT DATA VALID vs LOAD CAPACITANCE

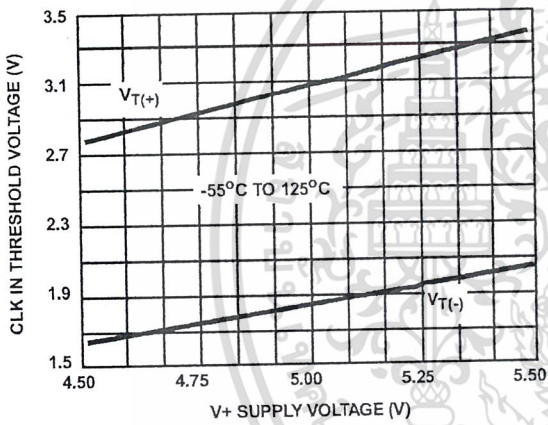


FIGURE 4. CLK IN SCHMITT TRIP LEVELS vs SUPPLY VOLTAGE

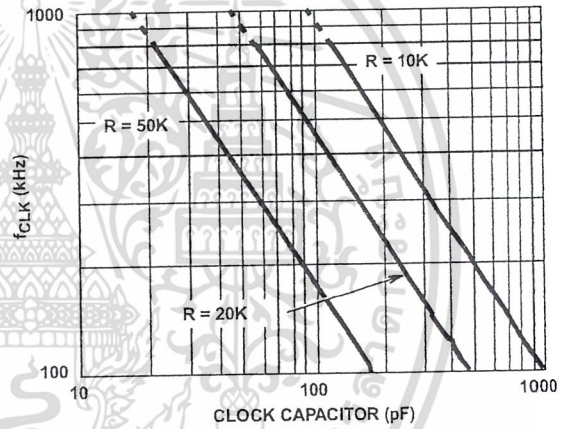


FIGURE 5. f_{CLK} vs CLOCK CAPACITOR

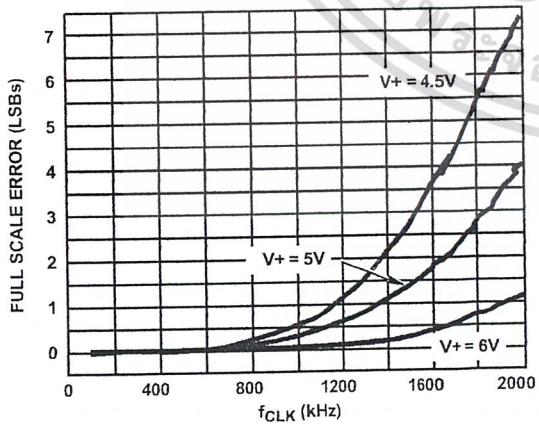


FIGURE 6. FULL SCALE ERROR vs f_{CLK}

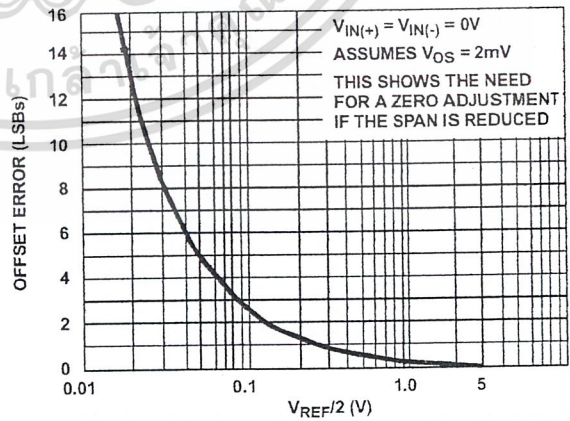


FIGURE 7. EFFECT OF UNADJUSTED OFFSET ERROR

ADC0802, ADC0803, ADC0804

Typical Performance Curves (Continued)

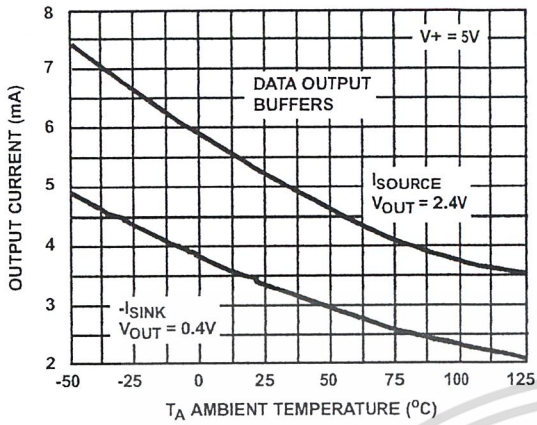


FIGURE 8. OUTPUT CURRENT vs TEMPERATURE

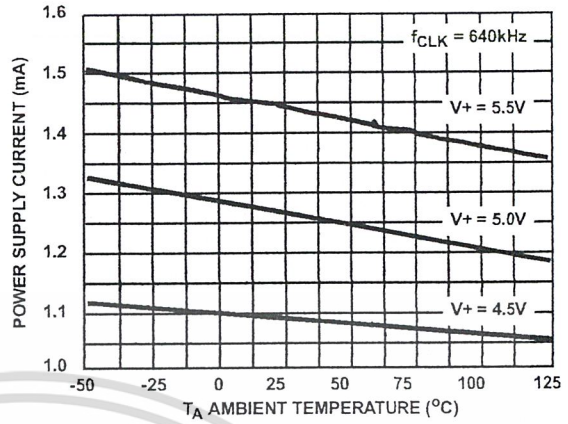


FIGURE 9. POWER SUPPLY CURRENT vs TEMPERATURE

Timing Diagrams

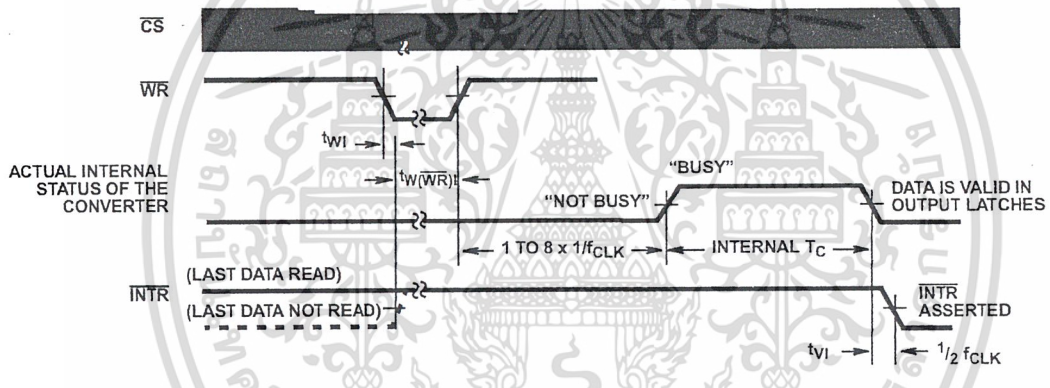


FIGURE 10A. START CONVERSION

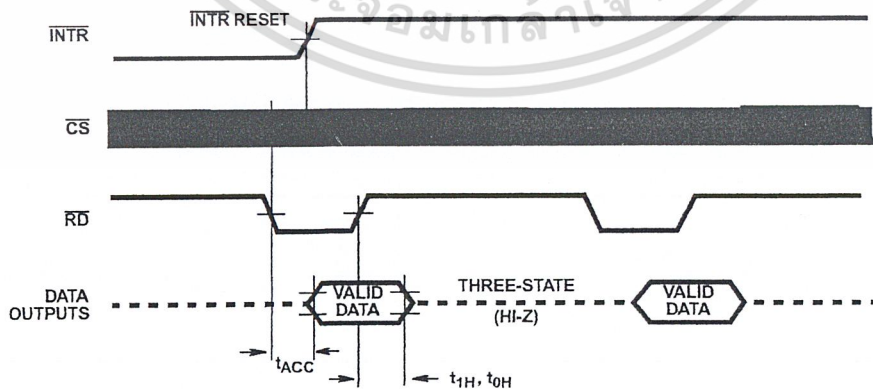


FIGURE 10B. OUTPUT ENABLE AND RESET INTR

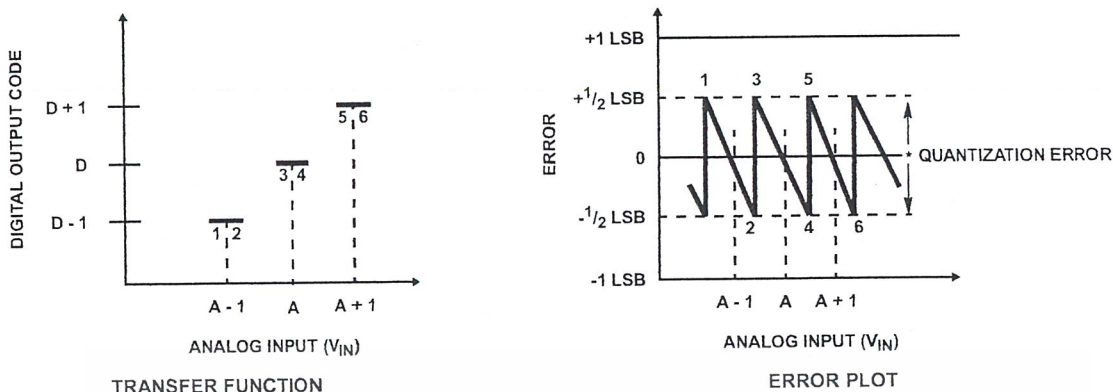


FIGURE 11A. ACCURACY = ± 0 LSB; PERFECT A/D

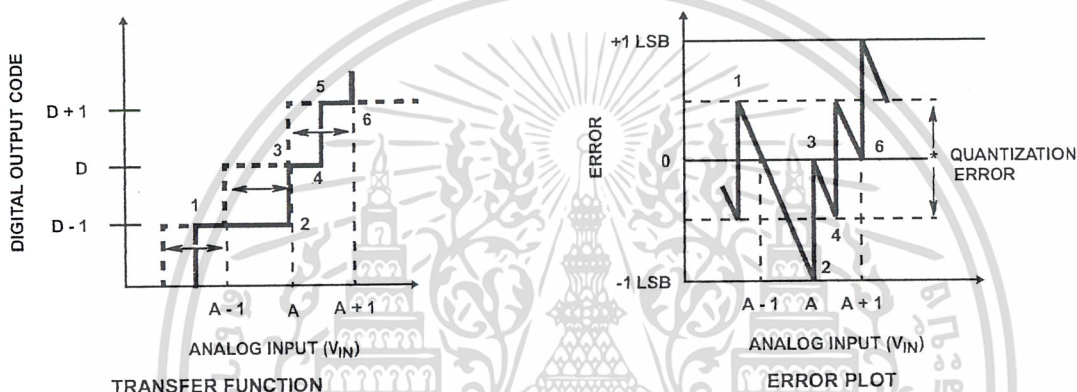


FIGURE 11B. ACCURACY = $\pm 1/2$ LSB

FIGURE 11. CLARIFYING THE ERROR SPECS OF AN A/D CONVERTER

Understanding A/D Error Specs

A perfect A/D transfer characteristic (staircase wave-form) is shown in Figure 11A. The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53mV with 2.5V tied to the $V_{REF}/2$ pin). The digital output codes which correspond to these inputs are shown as D-1, D, and D+1. For the perfect A/D, not only will center-value (A - 1, A, A + 1, . . .) analog inputs produce the correct output digital codes, but also each riser (the transitions between adjacent output codes) will be located $\pm 1/2$ LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages which extend $\pm 1/2$ LSB from the ideal center-values. Each tread (the range of analog input voltage which provides the same digital output code) is therefore 1 LSB wide.

The error curve of Figure 11B shows the worst case transfer function for the ADC0802. Here the specification guarantees that if we apply an analog input equal to the LSB analog voltage center-value, the A/D will produce the correct digital code.

Next to each transfer function is shown the corresponding error plot. Notice that the error includes the quantization uncertainty of the A/D. For example, the error at point 1 of Figure 11A is $+1/2$ LSB because the digital code appeared $1/2$ LSB in advance of the center-value of the tread. The error plots always have a

constant negative slope and the abrupt upside steps are always 1 LSB in magnitude, unless the device has missing codes.

Detailed Description

The functional diagram of the ADC0802 series of A/D converters operates on the successive approximation principle (see Application Notes AN016 and AN020 for a more detailed description of this principle). Analog switches are closed sequentially by successive-approximation logic until the analog differential input voltage [$V_{IN(+)} - V_{IN(-)}$] matches a voltage derived from a tapped resistor string across the reference voltage. The most significant bit is tested first and after 8 comparisons (64 clock cycles), an 8-bit binary code (1111 1111 = full scale) is transferred to an output latch.

The normal operation proceeds as follows. On the high-to-low transition of the WR input, the internal SAR latches and the shift-register stages are reset, and the INTR output will be set high. As long as the CS input and WR input remain low, the A/D will remain in a reset state. Conversion will start from 1 to 8 clock periods after at least one of these inputs makes a low-to-high transition. After the requisite number of clock pulses to complete the conversion, the INTR pin will make a high-to-low transition. This can be used to interrupt a processor, or otherwise signal the availability of a new conversion. A RD operation (with CS low) will clear the INTR line high again.

ADC0802, ADC0803, ADC0804

The device may be operated in the free-running mode by connecting INTR to the WR input with CS = 0. To ensure start-up under all possible conditions, an external WR pulse is required during the first power-up cycle. A conversion-in-process can be interrupted by issuing a second start command.

Digital Operation

The converter is started by having CS and WR simultaneously low. This sets the start flip-flop (F/F) and the resulting "1" level resets the 8-bit shift register, resets the Interrupt (INTR) F/F and inputs a "1" to the D flip-flop, DFF1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this "1" to the Q output of DFF1. The AND gate, G1, combines this "1" output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either WR or CS is a "1"), the start F/F is reset and the 8-bit shift register then can have the "1" clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would be at a "1" level) and the 8-bit shift register would continue to be held in the reset mode. This allows for asynchronous or wide CS and WR signals.

After the "1" is clocked through the 8-bit shift register (which completes the SAR operation) it appears as the input to DFF2. As soon as this "1" is output from the shift register, the AND gate, G2, causes the new digital word to transfer to the Three-State output latches. When DFF2 is subsequently clocked, the \bar{Q} output makes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the $\overline{\text{INTR}}$ output signal.

When data is to be read, the combination of both $\overline{\text{CS}}$ and $\overline{\text{RD}}$ being low will cause the INTR F/F to be reset and the three-state output latches will be enabled to provide the 8-bit digital outputs.

Digital Control Inputs

The digital control inputs (CS, RD, and WR) meet standard TTL logic voltage levels. These signals are essentially equivalent to the standard A/D Start and Output Enable control signals, and are active low to allow an easy interface to microprocessor control busses. For non-microprocessor based applications, the CS input (pin 1) can be grounded and the standard A/D Start function obtained by an active low pulse at the WR input (pin 3). The Output Enable function is achieved by an active low pulse at the RD input (pin 2).

Analog Operation

The analog comparisons are performed by a capacitive charge summing circuit. Three capacitors (with precise ratioed values) share a common node with the input to an auto-zeroed comparator. The input capacitor is switched between $V_{\text{IN}(+)}$ and $V_{\text{IN}(-)}$, while two ratioed reference capacitors are switched between taps on the reference voltage divider string. The net charge corresponds to the weighted difference between the input and the current total value set by the successive approximation register. A correction is made to offset the comparison by $1/2$ LSB (see Figure 11A).

Analog Differential Voltage Inputs and Common-Mode Rejection

This A/D gains considerable applications flexibility from the analog differential voltage input. The $V_{\text{IN}(-)}$ input (pin 7) can be used

to automatically subtract a fixed voltage value from the input reading (tare correction). This is also useful in 4mA - 20mA current loop conversion. In addition, common-mode noise can be reduced by use of the differential input.

The time interval between sampling $V_{\text{IN}(+)}$ and $V_{\text{IN}(-)}$ is $4^{1/2}$ clock periods. The maximum error voltage due to this slight time difference between the input voltage samples is given by:

$$\Delta V_E(\text{error}) = (V_{\text{PEAK}})(\text{common}) \left[\frac{4.5}{f_{\text{CLK}}} \right]$$

where:

ΔV_E is the error voltage due to sampling delay,

V_{PEAK} is the peak value of the common-mode voltage,

f_{CM} is the common-mode frequency.

For example, with a 60Hz common-mode frequency, f_{CM} , and a 640kHz A/D clock, f_{CLK} , keeping this error to $1/4$ LSB ($\sim 5\text{mV}$) would allow a common-mode voltage, V_{PEAK} , given by:

$$V_{\text{PEAK}} = \frac{[\Delta V_E(\text{error})](f_{\text{CLK}})}{(f_{\text{CM}})(\text{common})}$$

or

$$V_{\text{PEAK}} = \frac{(0.005)(640,000)}{(60)(4)} \cong 1.9\text{V}$$

The allowed range of analog input voltage usually places more severe restrictions on input common-mode voltage levels than this.

An analog input voltage with a reduced span and a relatively large zero offset can be easily handled by making use of the differential input (see Reference Voltage Span Adjust).

Analog Input Current

The internal switching action causes displacement currents to flow at the analog inputs. The voltage on the on-chip capacitance to ground is switched through the analog differential input voltage, resulting in proportional currents entering the $V_{\text{IN}(+)}$ input and leaving the $V_{\text{IN}(-)}$ input. These current transients occur at the leading edge of the internal clocks. They rapidly decay and do not inherently cause errors as the on-chip comparator is strobed at the end of the clock period.

Input Bypass Capacitors

Bypass capacitors at the inputs will average these charges and cause a DC current to flow through the output resistances of the analog signal sources. This charge pumping action is worse for continuous conversions with the $V_{\text{IN}(+)}$ input voltage at full scale. For a 640kHz clock frequency with the $V_{\text{IN}(+)}$ input at 5V, this DC current is at a maximum of approximately $5\mu\text{A}$. Therefore, **bypass capacitors should not be used at the analog inputs or the $V_{\text{REF}/2}$ pin** for high resistance sources ($>1\text{k}\Omega$). If input bypass capacitors are necessary for noise filtering and high source resistance is desirable to minimize capacitor size, the effects of the voltage drop across this input resistance, due to the average value of the input current, can be compensated by a full scale adjustment while the given source resistor and input bypass capacitor are both in place. This is possible because the average value of the input current is a precise linear function of the differential input voltage at a constant conversion rate.

Input Source Resistance

Large values of source resistance where an input bypass capacitor is not used will not cause errors since the input currents settle out prior to the comparison time. If a low-pass filter is required in the system, use a low-value series resistor ($\leq 1k\Omega$) for a passive RC section or add an op amp RC active low-pass filter. For low-source-resistance applications ($\leq 1k\Omega$), a $0.1\mu F$ bypass capacitor at the inputs will minimize EMI due to the series lead inductance of a long wire. A 100Ω series resistor can be used to isolate this capacitor (both the R and C are placed outside the feedback loop) from the output of an op amp, if used.

Stray Pickup

The leads to the analog inputs (pins 6 and 7) should be kept as short as possible to minimize stray signal pickup (EMI). Both EMI and undesired digital-clock coupling to these inputs can cause system errors. The source resistance for these inputs should, in general, be kept below $5k\Omega$. Larger values of source resistance can cause undesired signal pickup. Input bypass capacitors, placed from the analog inputs to ground, will eliminate this pickup but can create analog scale errors as these capacitors will average the transient input switching currents of the A/D (see Analog Input Current). This scale error depends on both a large source resistance and the use of an input bypass capacitor. This error can be compensated by a full scale adjustment of the A/D (see Full Scale Adjustment) with the source resistance and input bypass capacitor in place, and the desired conversion rate.

Reference Voltage Span Adjust

For maximum application flexibility, these A/Ds have been designed to accommodate a 5V, 2.5V or an adjusted voltage reference. This has been achieved in the design of the IC as shown in Figure 12.

Notice that the reference voltage for the IC is either $1/2$ of the voltage which is applied to the $V+$ supply pin, or is equal to the voltage which is externally forced at the $V_{REF}/2$ pin. This allows for a pseudo-ratiometric voltage reference using, for the $V+$ supply, a 5V reference voltage. Alternatively, a voltage less than 2.5V can be applied to the $V_{REF}/2$ input. The internal gain to the $V_{REF}/2$ input is 2 to allow this factor of 2 reduction in the reference voltage.

Such an adjusted reference voltage can accommodate a reduced span or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5V to 3.5V, instead of 0V to 5V, the span would be 3V. With 0.5V applied to the $V_{IN(-)}$ pin to absorb the offset, the reference voltage can be made equal to $1/2$ of the 3V span or 1.5V. The A/D now will encode the $V_{IN(+)}$ signal from 0.5V to 3.5V with the 0.5V input corresponding to zero and the 3.5V input corresponding to full scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range. The requisite connections are shown in Figure 13. For expanded scale inputs, the circuits of Figures 14 and 15 can be used.

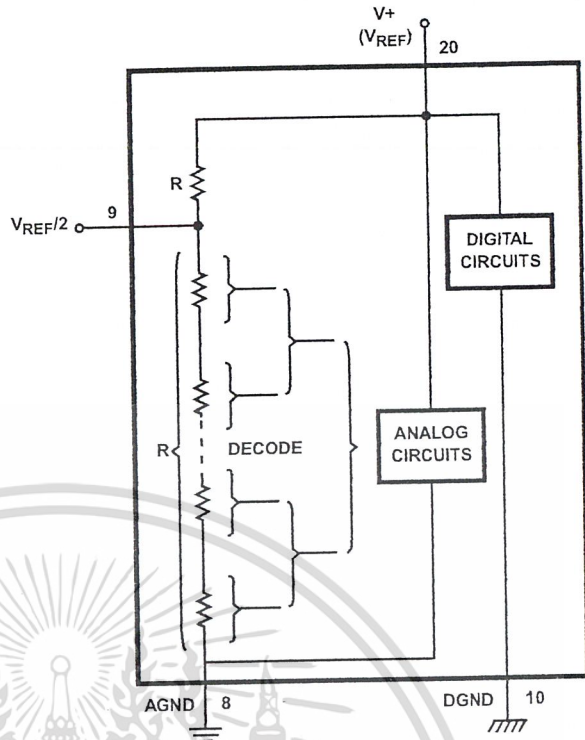


FIGURE 12. THE $V_{REFERENCE}$ DESIGN ON THE IC

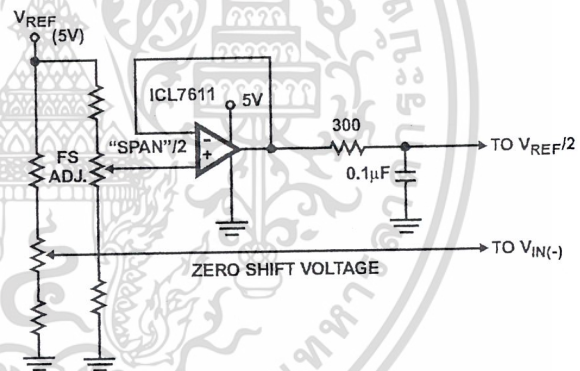


FIGURE 13. OFFSETTING THE ZERO OF THE ADC0802 AND PERFORMING AN INPUT RANGE (SPAN) ADJUSTMENT

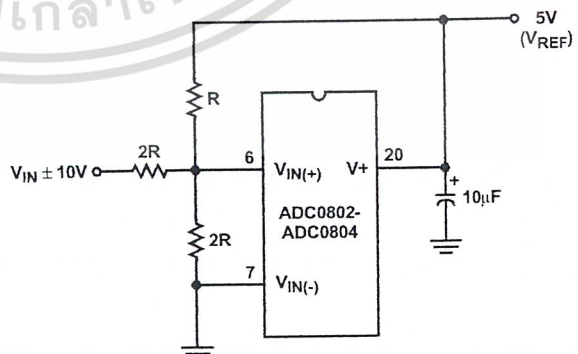


FIGURE 14. HANDLING $\pm 10V$ ANALOG INPUT RANGE

ADC0802, ADC0803, ADC0804

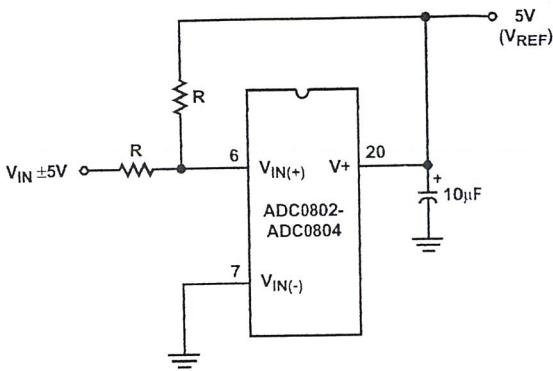


FIGURE 15. HANDLING ±5V ANALOG INPUT RANGE

Reference Accuracy Requirements

The converter can be operated in a pseudo-ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. In absolute conversion applications, both the initial value and the temperature stability of the reference voltage are important accuracy factors in the operation of the A/D converter. For $V_{REF}/2$ voltages of 2.5V nominal value, initial errors of $\pm 10\text{mV}$ will cause conversion errors of ± 1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20mV (5V span) to 10mV and 1 LSB at the $V_{REF}/2$ input becomes 5mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full scale errors in the A/D transfer function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive.

Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing the A/D $V_{IN(-)}$ input at this $V_{IN(MIN)}$ value (see Applications section). This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the $V_{IN(-)}$ input and applying a small magnitude positive voltage to the $V_{IN(+)}$ input. Zero error is the difference between the actual DC input voltage which is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal $1/2$ LSB value ($1/2$ LSB = 9.8mV for $V_{REF}/2 = 2.500\text{V}$).

Full Scale Adjust

The full scale adjustment can be made by applying a differential input voltage which is $1\frac{1}{2}$ LSB down from the desired analog full scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9) for a digital output code which is just changing from 1111 1110 to 1111 1111. When offsetting the zero and using a span-adjusted $V_{REF}/2$ voltage, the full scale adjustment is made by inputting V_{MIN} to the $V_{IN(-)}$ input of the A/D and applying a voltage to the $V_{IN(+)}$ input which is given by:

$$V_{IN(+)}^{fSADJ} = V_{MAX} - 1.5 \left[\frac{\dots\dots\dots}{256} \right],$$

where:

V_{MAX} = the high end of the analog input range,

and

V_{MIN} = the low end (the offset zero) of the analog range. (Both are ground referenced.)

Clocking Option

The clock for the A/D can be derived from an external source such as the CPU clock or an external RC network can be added to provide self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in Figure 16.

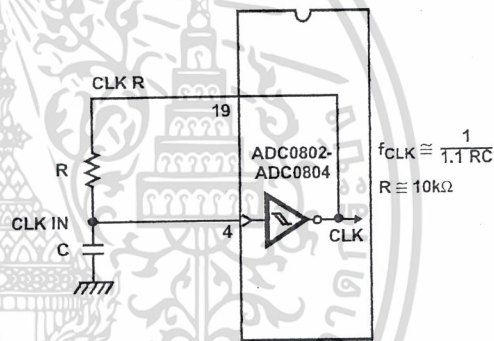


FIGURE 16. SELF-CLOCKING THE A/D

Heavy capacitive or DC loading of the CLK R pin should be avoided as this will disturb normal converter operation. Loads less than 50pF, such as driving up to 7 A/D converter clock inputs from a single CLK R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power TTL buffer or PNP input logic should be used to minimize the loading on the CLK R pin (do not use a standard TTL buffer).

Restart During a Conversion

If the A/D is restarted (\overline{CS} and \overline{WR} go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in progress is not completed. The data from the previous conversion remain in this latch.

Continuous Conversions

In this application, the \overline{CS} input is grounded and the \overline{WR} input is tied to the \overline{INTR} output. This \overline{WR} and \overline{INTR} node should be momentarily forced to logic low following a power-up cycle to insure circuit operation. See Figure 17 for details.

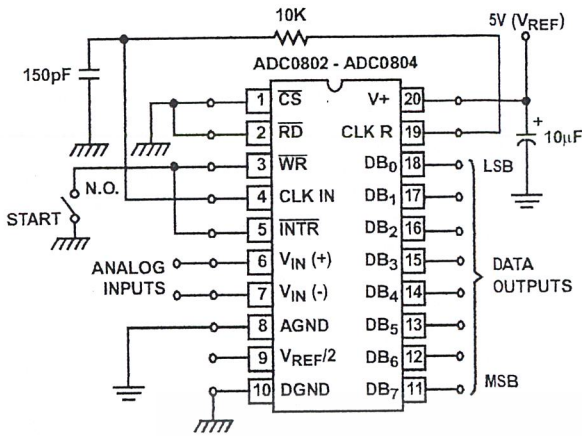


FIGURE 17. FREE-RUNNING CONNECTION

Driving the Data Bus

This CMOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuitry, which is tied to the data bus, will add to the total capacitive loading, even in three-state (high-impedance mode). Back plane busing also greatly adds to the stray capacitance of the data bus.

There are some alternatives available to the designer to handle this problem. Basically, the capacitive loading of the data bus slows down the response time, even though DC specifications are still met. For systems operating with a relatively slow CPU clock frequency, more time is available in which to establish proper logic levels on the bus and therefore higher capacitive loads can be driven (see Typical Performance Curves).

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock-extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be three-state buffers (low power Schottky is recommended, such as the 74LS240 series) or special higher-drive-current products which are designed as bus drivers. High-current bipolar bus drivers with PNP inputs are recommended.

Power Supplies

Noise spikes on the V+ supply line can cause conversion errors as the comparator will respond to this noise. A low-inductance tantalum filter capacitor should be used close to the converter V+ pin, and values of 1µF or greater are recommended. If an unregulated voltage is available in the system, a separate 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V+ supply. An ICL7663 can be used to regulate such a supply from an input as low as 5.2V.

Wiring and Hook-Up Precautions

Standard digital wire-wrap sockets are not satisfactory for breadboarding with this A/D converter. Sockets on PC boards can be used. All logic signal wires and leads should be grouped and kept as far away as possible from the analog

signal leads. Exposed leads to the analog inputs can cause undesired digital noise and hum pickup; therefore, shielded leads may be necessary in many applications.

A single-point analog ground should be used which is separate from the logic ground points. The power supply bypass capacitor and the self-clocking capacitor (if used) should both be returned to digital ground. Any VREF/2 bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of 1/4 LSB can usually be traced to improper board layout and wiring (see Zero Error for measurement). Further information can be found in Application Note AN018.

Testing the A/D Converter

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in Figure 18.

For ease of testing, the VREF/2 (pin 9) should be supplied with 2.560V and a V+ supply voltage of 5.12V should be used. This provides an LSB value of 20mV.

If a full scale adjustment is to be made, an analog input voltage of 5.090V (5.120 - 1 1/2 LSB) should be applied to the VIN(+) pin with the VIN(-) pin grounded. The value of the VREF/2 input voltage should be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of VREF/2 should then be used for all the tests.

The digital-output LED display can be decoded by dividing the 8 bits into 2 hex characters, one with the 4 most-significant bits (MS) and one with the 4 least-significant bits (LS). The output is then interpreted as a sum of fractions times the full scale voltage:

$$V_{OUT} = \left(\frac{\text{MS}}{16} + \frac{\text{LS}}{16} \right) (V_{FS})$$

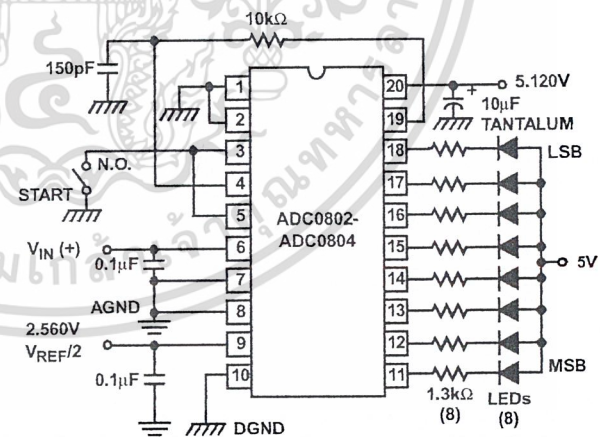


FIGURE 18. BASIC TESTER FOR THE A/D

For example, for an output LED display of 1011 0110, the MS character is hex B (decimal 11) and the LS character is hex (and decimal) 6, so:

$$V_{OUT} = \left(\frac{11}{16} + \frac{6}{16} \right) (5.12V) = 3.64V$$

ADC0802, ADC0803, ADC0804

Figures 19 and 20 show more sophisticated test circuits.

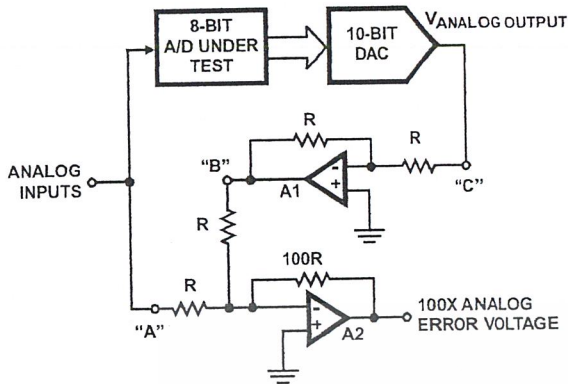


FIGURE 19. A/D TESTER WITH ANALOG ERROR OUTPUT. THIS CIRCUIT CAN BE USED TO GENERATE "ERROR PLOTS" OF FIGURE 11.

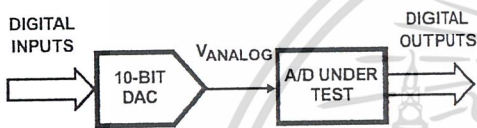


FIGURE 20. BASIC "DIGITAL" A/D TESTER

Typical Applications

Interfacing 8080/85 or Z-80 Microprocessors

This converter has been designed to directly interface with 8080/85 or Z-80 Microprocessors. The three-state output capability of the A/D eliminates the need for a peripheral interface device, although address decoding is still required to generate the appropriate \overline{CS} for the converter. The A/D can be mapped into memory space (using standard memory-address decoding for \overline{CS} and the \overline{MEMR} and \overline{MEMW} strobes) or it can be controlled as an I/O device by using the \overline{IOR} and \overline{IOW} strobes and decoding the address bits A0 → A7 (or address bits A8 → A15, since they will contain the same 8-bit address information) to obtain the \overline{CS} input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder, but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. See AN020 for more discussion of memory-mapped vs I/O-mapped interfaces. An example of an A/D in I/O space is shown in Figure 21.

The standard control-bus signals of the 8080 (\overline{CS} , \overline{RD} and \overline{WR}) can be directly wired to the digital control inputs of the A/D, since the bus timing requirements, to allow both starting the converter, and outputting the data onto the data bus, are met. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100pF.

It is useful to note that in systems where the A/D converter is 1 of 8 or fewer I/O-mapped devices, no address-decoding circuitry is necessary. Each of the 8 address bits (A0 to A7) can be directly used as \overline{CS} inputs, one for each I/O device.

Interfacing the Z-80 and 8085

The Z-80 and 8085 control buses are slightly different from that of the 8080. General \overline{RD} and \overline{WR} strobes are provided and separate memory request, \overline{MREQ} , and I/O request, \overline{IORQ} , signals have to be combined with the generalized strobes to provide the appropriate signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the \overline{RD} and \overline{WR} strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in Figure 22. By using \overline{MREQ} in place of \overline{IORQ} , a memory-mapped configuration results.

Additional I/O advantages exist as software DMA routines are available and use can be made of the output data transfer which exists on the upper 8 address lines (A8 to A15) during I/O input instructions. For example, MUX channel selection for the A/D can be accomplished with this operating mode.

The 8085 also provides a generalized \overline{RD} and \overline{WR} strobe, with an $\overline{IO/M}$ line to distinguish I/O and memory requests. The circuit of Figure 22 can again be used, with $\overline{IO/M}$ in place of \overline{IORQ} for a memory-mapped interface, and an extra inverter (or the logic equivalent) to provide $\overline{IO/M}$ for an I/O-mapped connection.

Interfacing 6800 Microprocessor Derivatives (6502, etc.)

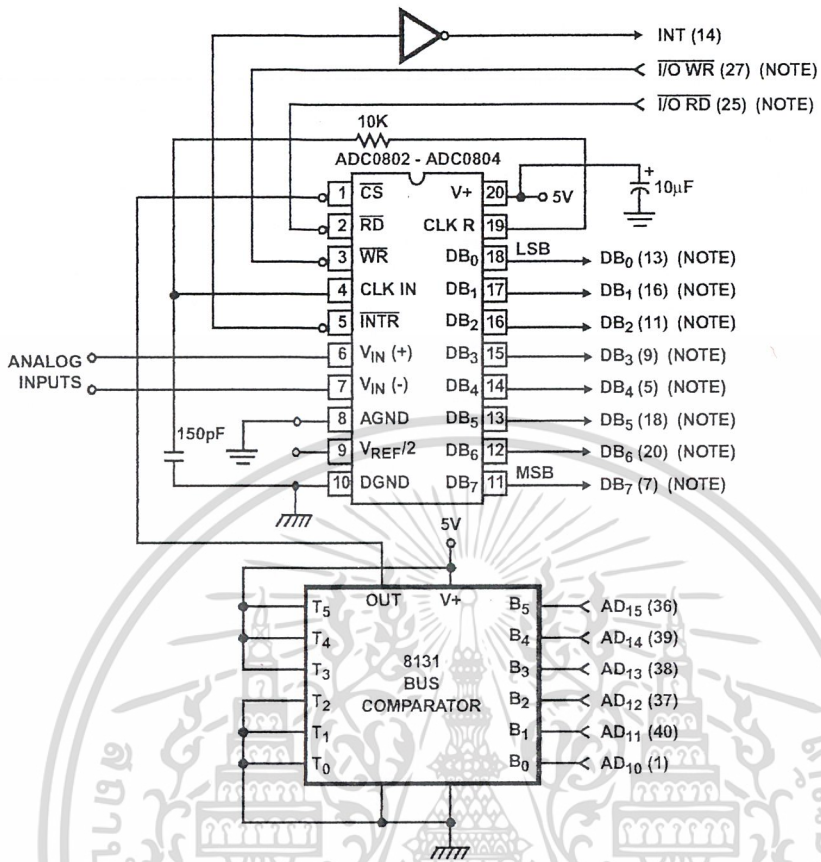
The control bus for the 6800 microprocessor derivatives does not use the \overline{RD} and \overline{WR} strobe signals. Instead it employs a single R/W line and additional timing, if needed, can be derived from the ϕ_2 clock. All I/O devices are memory-mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. Figure 23 shows an interface schematic where the A/D is memory-mapped in the 6800 system. For simplicity, the CS decoding is shown using $1/2$ DM8092. Note that in many 6800 systems, an already decoded $4/5$ line is brought out to the common bus at pin 21. This can be tied directly to the CS pin of the A/D, provided that no other devices are addressed at HEX ADDR: 4XXX or 5XXX.

In Figure 24 the ADC0802 series is interfaced to the MC6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter (PIA). Here the CS pin of the A/D is grounded since the PIA is already memory-mapped in the MC6800 system and no CS decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D RD pin can be grounded.

Application Notes

NOTE #	DESCRIPTION	AnswerFAX DOC. #
AN016	"Selecting A/D Converters"	9016
AN018	"Do's and Don'ts of Applying A/D Converters"	9018
AN020	"A Cookbook Approach to High Speed Data Acquisition and Microprocessor Interfacing"	9020
AN030	"The ICL7104 - A Binary Output A/D Converter for Microprocessors"	9030

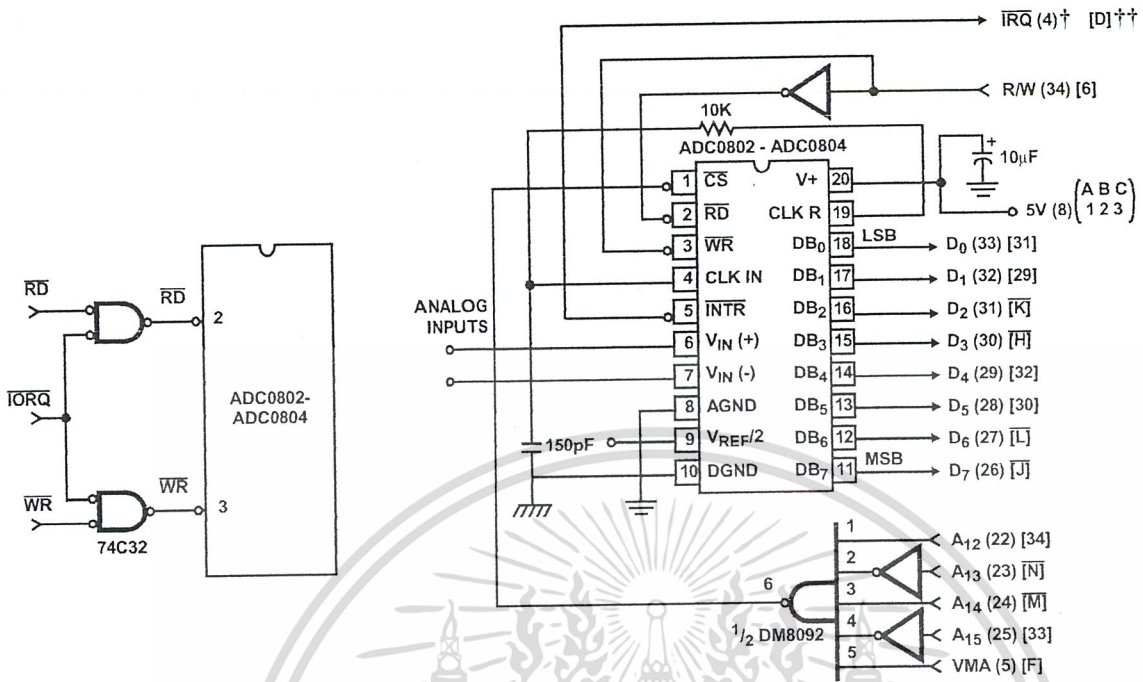
ADC0802, ADC0803, ADC0804



NOTE: Pin numbers for 8228 System Controller: Others are 8080A.

FIGURE 21. ADC0802 TO 8080A CPU INTERFACE

ADC0802, ADC0803, ADC0804



† Numbers in parentheses refer to MC6800 CPU Pinout.
 †† Numbers or letters in brackets refer to standard MC6800 System Common Bus Code.

FIGURE 22. MAPPING THE A/D AS AN I/O DEVICE FOR USE WITH THE Z-80 CPU

FIGURE 23. ADC0802 TO MC6800 CPU INTERFACE

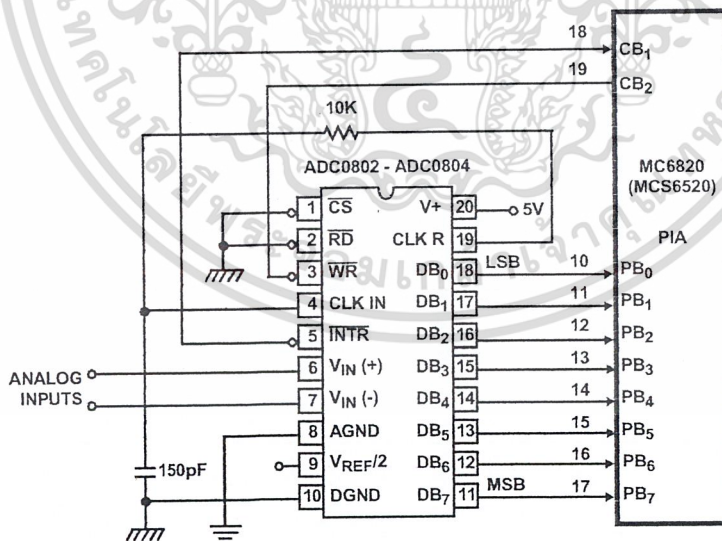


FIGURE 24. ADC0802 TO MC6820 PIA INTERFACE

+5V Powered, Dual RS-232 Transmitter/Receiver

Features

- Meets All RS-232C and V.28 Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- 2 Drivers
 - $\pm 9V$ Output Swing for +5V Input
 - 300Ω Power-off Source Impedance
 - Output Current Limiting
 - TTL/CMOS Compatible
 - $30V/\mu s$ Maximum Slew Rate
- 2 Receivers
 - $\pm 30V$ Input Voltage Range
 - $3k\Omega$ to $7k\Omega$ Input Impedance
 - 0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

Description

The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C and V.28 specifications. It requires a single +5V power supply, and features two onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply.

The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300Ω power-off source impedance. The receivers can handle up to +30V, and have a $3k\Omega$ to $7k\Omega$ input impedance. The receivers also have hysteresis to improve noise rejection.

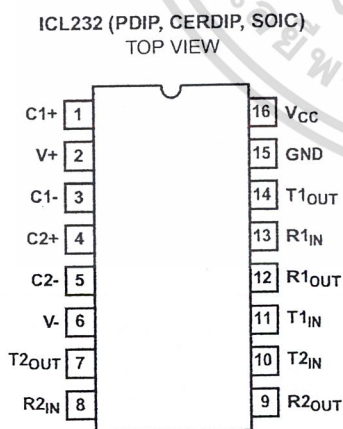
Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
ICL232CPE	0 to 70	16 Ld PDIP	E16.3
ICL232CBE	0 to 70	16 Ld SOIC	M16.3
ICL232IPE	-40 to 85	16 Ld PDIP	E16.3
ICL232IJE	-40 to 85	16 Ld CERDIP	F16.3
ICL232IBE	-40 to 85	16 Ld SOIC	M16.3
ICL232MJE	-55 to 125	16 Ld CERDIP	F16.3

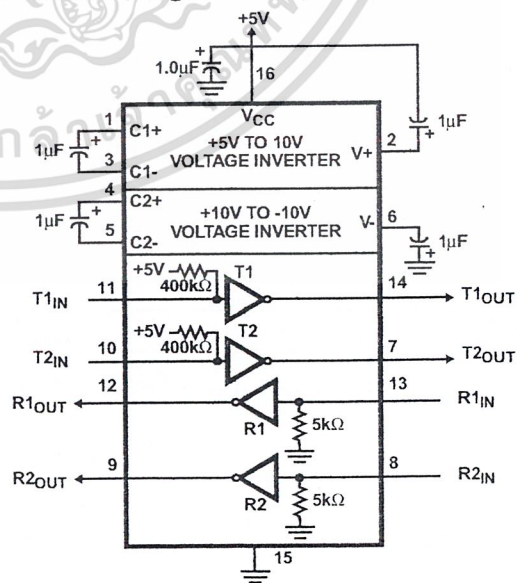
Applications

- Any System Requiring RS-232 Communications Port
 - Computer - Portable and Mainframe
 - Peripheral - Printers and Terminals
 - Portable Instrumentation
 - Modems
- Dataloggers

Pinout



Functional Diagram



ICL232

Absolute Maximum Ratings

V_{CC} to Ground	$(GND - 0.3V) < V_{CC} < 6V$
$V+$ to Ground	$(V_{CC} - 0.3V) < V+ < 12V$
$V-$ to Ground	$-12V < V- < (GND + 0.3V)$
Input Voltages	
$T1_{IN}, T2_{IN}$	$(V- - 0.3V) < V_{IN} < (V+ + 0.3V)$
$R1_{IN}, R2_{IN}$	$\pm 30V$
Output Voltages	
$T1_{OUT}, T2_{OUT}$	$(V- - 0.3V) < V_{TXOUT} < (V+ + 0.3V)$
$R1_{OUT}, R2_{OUT}$	$(GND - 0.3V) < V_{RXOUT} < (V_{CC} + 0.3V)$
Short Circuit Duration	
$T1_{OUT}, T2_{OUT}$	Continuous
$R1_{OUT}, R2_{OUT}$	Continuous

Thermal Information

Thermal Resistance (Typical, Note 1)	θ_{JA} ($^{\circ}C/W$)	θ_{JC} ($^{\circ}C/W$)
CERDIP Package	80	18
PDIP Package	100	N/A
SOIC Package	100	N/A
Maximum Junction Temperature		
Plastic Packages	150 $^{\circ}C$	
Ceramic Package	175 $^{\circ}C$	
Maximum Storage Temperature Range		
-65 $^{\circ}C$ to 150 $^{\circ}C$		
Maximum Lead Temperature (Soldering 10s)		
300 $^{\circ}C$		

Operating Conditions

Temperature Ranges	
ICL232C	0 $^{\circ}C$ to 70 $^{\circ}C$
ICL232I	-40 $^{\circ}C$ to 85 $^{\circ}C$
ICL232M	-55 $^{\circ}C$ to 125 $^{\circ}C$

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

- θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

Electrical Specifications

Test Conditions: $V_{CC} = +5V \pm 10\%$, $T_A =$ Operating Temperature Range. Test Circuit as in Figure 8 Unless Otherwise Specified

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Transmitter Output Voltage Swing, T_{OUT}	$T1_{OUT}$ and $T2_{OUT}$ Loaded with 3k Ω to Ground	± 5	± 9	± 10	V
Power Supply Current, I_{CC}	Outputs Unloaded, $T_A = 25^{\circ}C$	-	5	10	mA
$T1_{IN}$, Input Logic Low, V_{IL}		-	-	0.8	V
$T1_{IN}$, Input Logic High, V_{IH}		2.0	-	-	V
Logic Pullup Current, I_p	$T1_{IN}, T2_{IN} = 0V$	-	15	200	μA
RS-232 Input Voltage Range, V_{IN}		-30	-	+30	V
Receiver Input Impedance, R_{IN}	$V_{IN} = \pm 3V$	3.0	5.0	7.0	k Ω
Receiver Input Low Threshold, V_{IN} (H-L)	$V_{CC} = 5V, T_A = 25^{\circ}C$	0.8	1.2	-	V
Receiver Input High Threshold, V_{IN} (L-H)	$V_{CC} = 5V, T_A = 25^{\circ}C$	-	1.7	2.4	V
Receiver Input Hysteresis, V_{HYST}		0.2	0.5	1.0	V
TTL/CMOS Receiver Output Voltage Low, V_{OL}	$I_{OUT} = 3.2mA$	-	0.1	0.4	V
TTL/CMOS Receiver Output Voltage High, V_{OH}	$I_{OUT} = -1.0mA$	3.5	4.6	-	V
Propagation Delay, t_{PD}	RS-232 to TTL	-	0.5	-	μs
Instantaneous Slew Rate, SR	$C_L = 10pF, R_L = 3k\Omega, T_A = 25^{\circ}C$ (Notes 2, 3)	-	-	30	V/ μs
Transition Region Slew Rate, SR_T	$R_L = 3k\Omega, C_L = 2500pF$ Measured from +3V to -3V or -3V to +3V	-	3	-	V/ μs
Output Resistance, R_{OUT}	$V_{CC} = V+ = V- = 0V, V_{OUT} = \pm 2V$	300	-	-	Ω
RS-232 Output Short Circuit Current, I_{SC}	$T1_{OUT}$ or $T2_{OUT}$ Shorted to GND	-	± 10	-	mA

NOTES:

- Guaranteed by design.
- See Figure 4 for definition.

Test Circuits

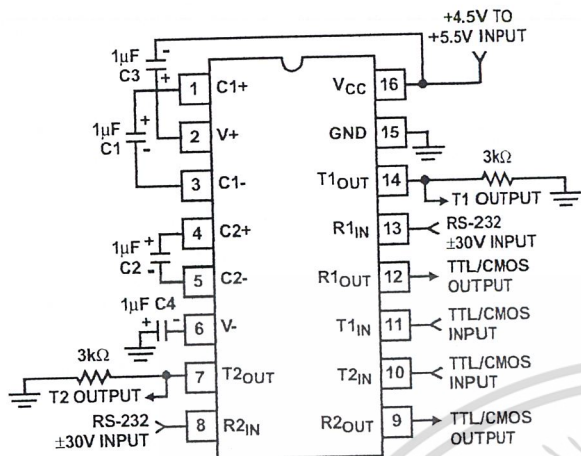


FIGURE 1. GENERAL TEST CIRCUIT

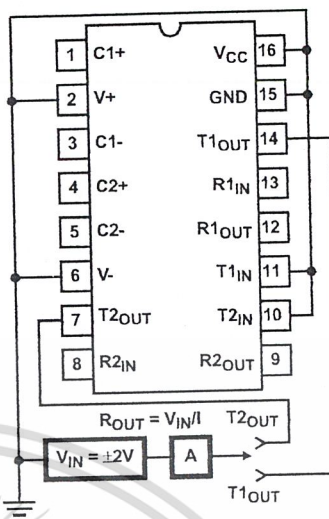


FIGURE 2. POWER-OFF SOURCE RESISTANCE CONFIGURATION

Typical Performance Curves

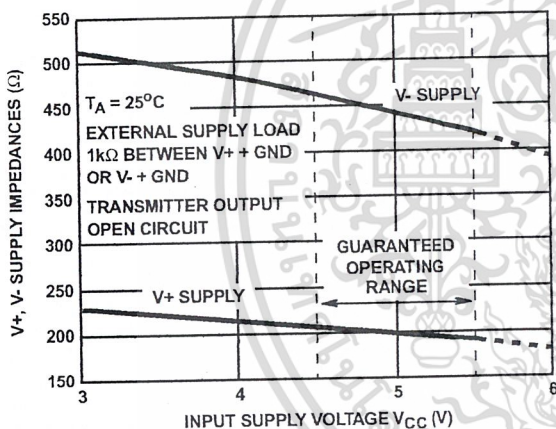


FIGURE 3. V+, V- OUTPUT IMPEDANCES vs VCC

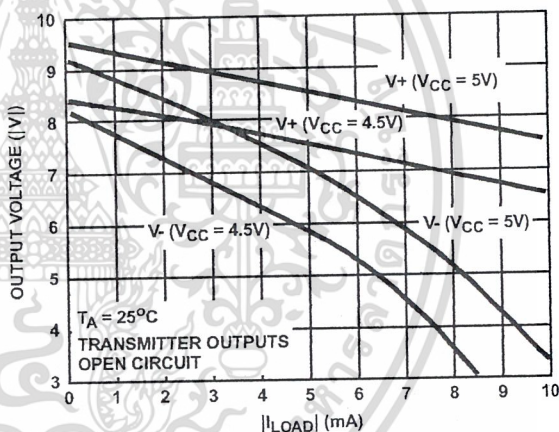


FIGURE 4. V+, V- OUTPUT VOLTAGES vs LOAD CURRENT

Pin Descriptions

PDIP, CERDIP	SOIC	PIN NAME	DESCRIPTION
1	1	C1+	External capacitor "+" for internal voltage doubler.
2	2	V+	Internally generated +10V (typical) supply.
3	3	C1-	External capacitor "-" for internal voltage doubler.
4	4	C2+	External capacitor "+" internal voltage inverter.
5	5	C2-	External capacitor "-" internal voltage inverter.
6	6	V-	Internally generated -10V (typical) supply.
7	7	T2OUT	RS-232 Transmitter 2 output ±10V (typical).
8	8	R2IN	RS-232 Receiver 2 input, with internal 5K pull-down resistor to GND.
9	9	R2out	Receiver 2 TTL/CMOS output.
10	10	T2IN	Transmitter 2 TTL/CMOS input, with internal 400K pull-up resistor to VCC.
11	11	T1IN	Transmitter 1 TTL/CMOS input, with internal 400K pull-up resistor to VCC.

Pin Descriptions (Continued)

PDIP, CERDIP	SOIC	PIN NAME	DESCRIPTION
12	12	R1 _{OUT}	Receiver 1 TTL/CMOS output.
13	13	R1 _{IN}	RS-232 Receiver 1 input, with internal 5K pulldown resistor to GND.
14	14	T1 _{OUT}	RS-232 Transmitter 1 output ±10V (typical).
15	15	GND	Supply Ground.
16	16	V _{CC}	Positive Power Supply +5V ±10%

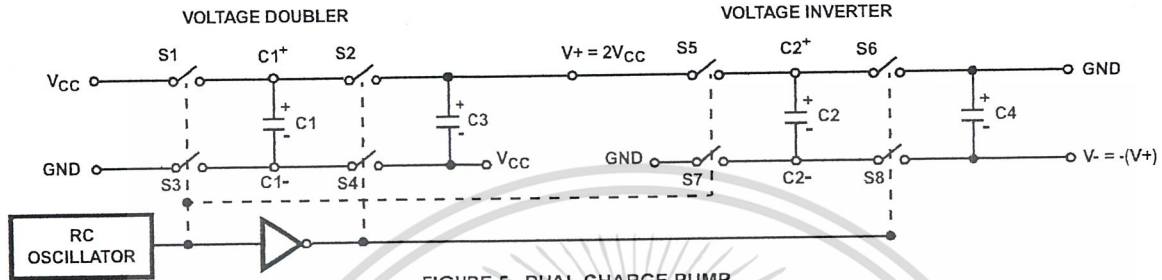


FIGURE 5. DUAL CHARGE PUMP

Detailed Description

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS232C specifications and features low power consumption. The functional diagram illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage doubler/inverter, dual transmitters, and dual receivers Voltage Converter.

An equivalent circuit of the dual charge pump is illustrated in Figure 5.

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and -10V. The nominal clock frequency is 16kHz. During phase one of the clock, capacitor C1 is charged to V_{CC}. During phase two, the voltage on C1 is added to V_{CC}, producing a signal across C2 equal to twice V_{CC}. At the same time, C3 is also charged to 2V_{CC}, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to -2V_{CC}. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler (V₊) is approximately 200Ω, and the output impedance of the inverter (V₋) is approximately 450Ω. Typical graphs are presented which show the voltage converters output vs input voltage and output voltages vs load characteristics. The test circuit (Figure 3) uses 1μF capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the V₊ and V₋ supplies.

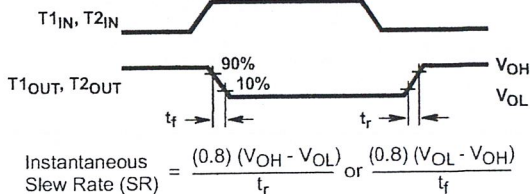


FIGURE 6. SLEW RATE DEFINITION

$$\text{Instantaneous Slew Rate (SR)} = \frac{(0.8)(V_{OH} - V_{OL})}{t_r} \text{ or } \frac{(0.8)(V_{OL} - V_{OH})}{t_f}$$

Transmitters

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of V_{CC}, or 1.3V for V_{CC} = 5V. A logic 1 at the input results in a voltage of between -5V and V₋ at the output, and a logic 0 results in a voltage between +5V and (V₊ - 0.6V). Each transmitter input has an internal 400kΩ pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232C specification of ±5V minimum with the worst case conditions of: both transmitters driving 3kΩ minimum load impedance, V_{CC} = 4.5V, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than 30V/μs. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300Ω with ±2V applied to the outputs and V_{CC} = 0V.

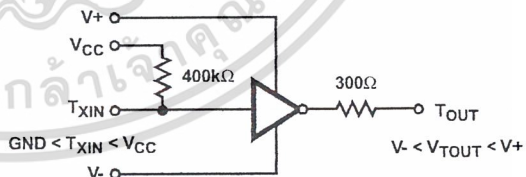


FIGURE 7. TRANSMITTER

Receivers

The receiver inputs accept up to ±30V while presenting the required 3kΩ to 7kΩ input impedance even if the power is off (V_{CC} = 0V). The receivers have a typical input threshold of 1.3V which is within the ±3V limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to V_{CC}. The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and -30V. The receivers feature 0.5V hysteresis to improve noise rejection.

ICL232

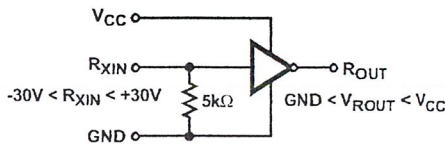
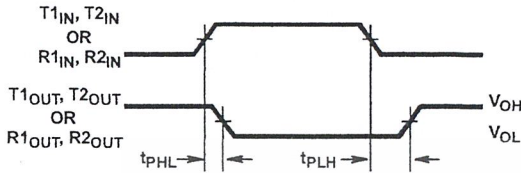


FIGURE 8. RECEIVER



$$\text{Average Propagation Delay} = \frac{t_{PHL} + t_{PLH}}{2}$$

FIGURE 9. PROPAGATION DELAY DEFINITION

is generated by driving them through a 5kΩ resistor connected to V+.

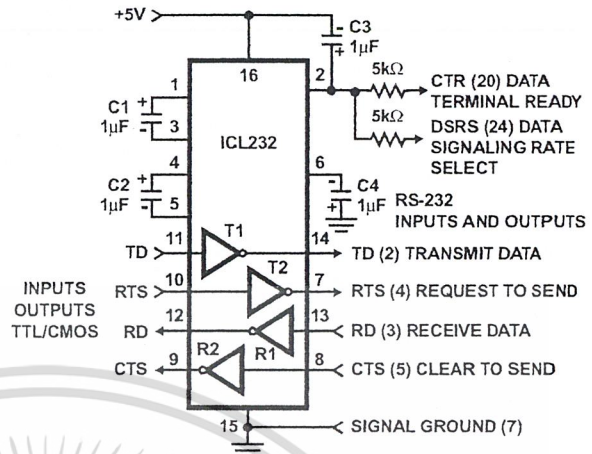


FIGURE 10. SIMPLE DUPLEX RS-232 PORT WITH CTS/RTS HANDSHAKING

Applications

The ICL232 may be used for all RS-232 data terminal and communication links. It is particularly useful in applications where ±12V power supplies are not available for conventional RS-232 interface circuits. The applications presented represent typical interface configurations.

A simple duplex RS-232 port with CTS/RTS handshaking is illustrated in Figure 10. Fixed output signals such as DTR (data terminal ready) and DSRS (data signaling rate select)

In applications requiring four RS-232 inputs and outputs (Figure 11), note that each circuit requires two charge pump capacitors (C1 and C2) but can share common reservoir capacitors (C3 and C4). The benefit of sharing common reservoir capacitors is the elimination of two capacitors and the reduction of the charge pump source impedance which effectively increases the output swing of the transmitters.

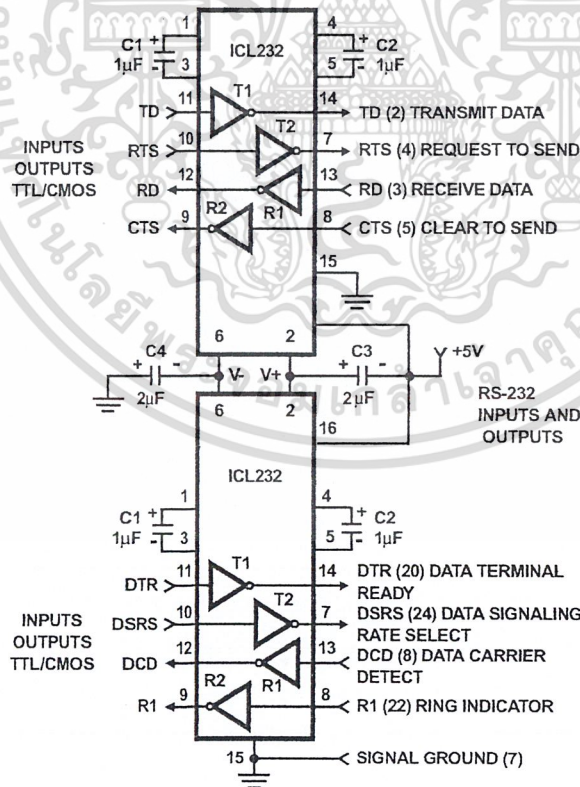
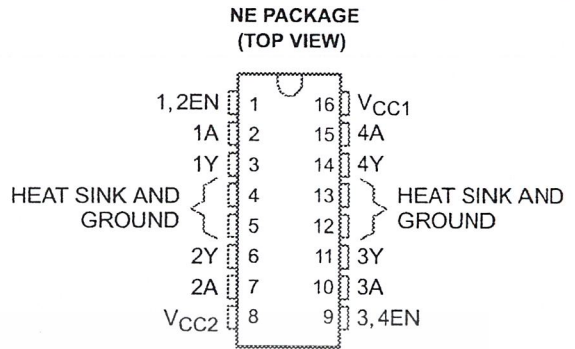


FIGURE 11. COMBINING TWO ICL232s FOR 4 PAIRS OF RS-232 INPUTS AND OUTPUTS

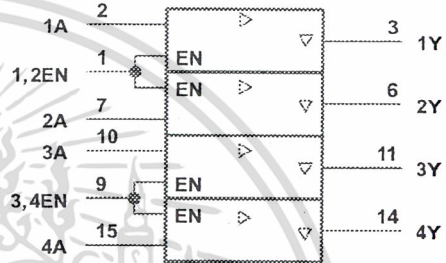
L293D QUADRUPLE HALF-H DRIVER

SLRS008A – SEPTEMBER 1986 – REVISED MAY 1990

- 600-mA Output Current Capability Per Driver
- Pulsed Current 1.2-A Per Driver
- Output Clamp Diodes for Inductive Transient Suppression
- Wide Supply Voltage Range 4.5 V to 36 V
- Separate Input-Logic Supply
- Thermal Shutdown
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Functional Replacement for SGS L293D

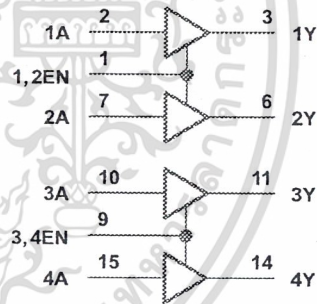


logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram



FUNCTION TABLE
(each driver)

INPUTS‡		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low level,
X = irrelevant, Z = high-impedance (off)
‡ In the thermal shutdown mode, the output is in the high-impedance state regardless of the input levels.

description

The L293D is a quadruple high-current half-H driver designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. It is designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL-compatible. Each output is a complete totem-pole drive circuit with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. External high-speed output clamp diodes should be used for inductive transient suppression. When the enable input is low, those drivers are disabled, and their outputs are off and in a high-impedance state. With the proper data inputs, each pair of drivers form a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

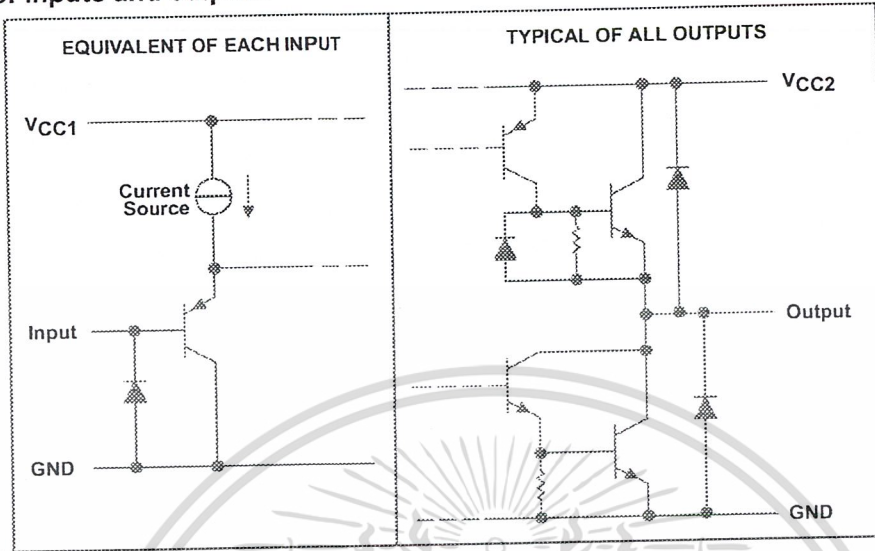
A V_{CC1} terminal, separate from V_{CC2} , is provided for the logic inputs to minimize device power dissipation.

The L293D is designed for operation from 0°C to 70°C.

L293D QUADRUPLE HALF-H DRIVER

SLRS008A – SEPTEMBER 1986 – REVISED MAY 1990

schematics of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Logic supply voltage range, V_{CC1} (see Note 1)	36 V
Output supply voltage range, V_{CC2}	36 V
Input voltage range, V_I	7 V
Output voltage range, V_O	-3 V to $V_{CC2} + 3$ V
Peak output current (nonrepetitive, $t \leq 100 \mu\text{s}$)	± 1.2 A
Continuous output current, I_O	± 600 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Notes 2 and 3)	2075 mW
Continuous total dissipation at 80°C case temperature (see Note 3)	5000 mW
Operating case or virtual junction temperature range, T_J	-40°C to 150°C
Storage temperature range, T_{stg}	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

- NOTES: 1. All voltage values are with respect to the network ground terminal.
 2. For operation above 25°C free-air temperature, derate linearly at the rate of 16.6 mW/°C.
 3. For operation above 25°C case temperature, derate linearly at the rate of 71.4 mW/°C. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection may be activated at power levels slightly above or below the rated dissipation.

recommended operating conditions

	MIN	MAX	UNIT
Logic supply voltage, V_{CC1}	4.5	7	V
Output supply voltage, V_{CC2}	V_{CC1}	36	V
High-level input voltage, V_{IH}	$V_{CC1} \leq 7$ V	2.3	V_{CC1}
	$V_{CC1} \geq 7$ V	7	V
Low-level input voltage, V_{IL}	-0.3†	1.5	V
Operating free-air temperature, T_A	0	70	°C

† The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

3-2 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L293D QUADRUPLE HALF-H DRIVER

SLRS008A – SEPTEMBER 1986 – REVISED MAY 1990

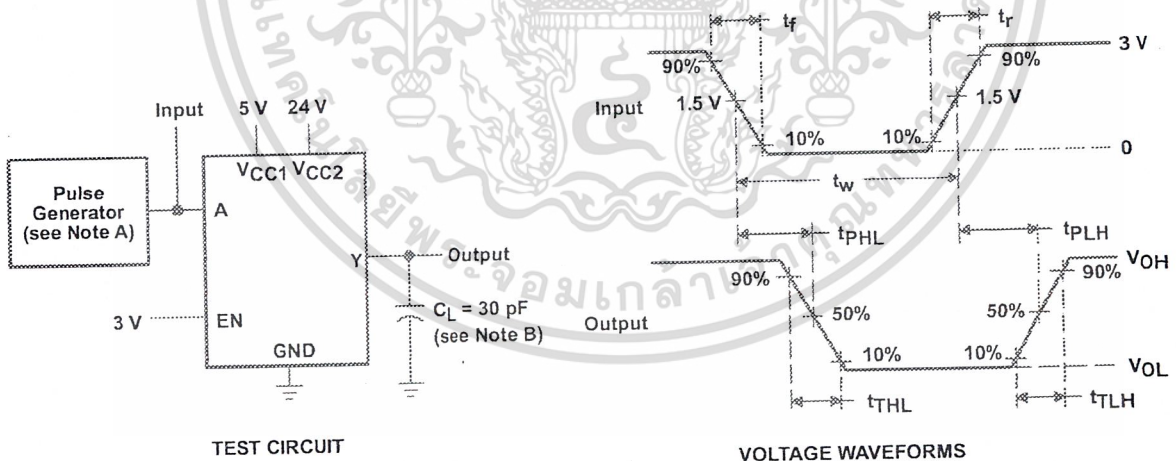
electrical characteristics, $V_{CC1} = 5\text{ V}$, $V_{CC2} = 24\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	$I_{OH} = -0.6\text{ A}$		$V_{CC2} - 1.8$	$V_{CC2} - 1.4$		V
V_{OL}	Low-level output voltage	$I_{OL} = 0.6\text{ A}$			1.2	1.8	V
V_{OKH}	High-level output clamp voltage	$I_{OK} = -0.6\text{ A}$			$V_{CC2} + 1.3$		V
V_{OKL}	Low-level output clamp voltage	$I_{OK} = -0.6\text{ A}$			1.3		V
I_{IH}	High-level input current	A	$V_I = 7\text{ V}$		0.2	100	μA
		EN			0.2	± 10	
I_{IL}	Low-level input current	A	$V_I = 0$		-3	-10	μA
		EN			-2	-100	
I_{CC1}	Logic supply current	$I_O = 0$		All outputs at high level	13	22	mA
				All outputs at low level	35	60	
				All outputs at high impedance	8	24	
					14	24	
I_{CC2}	Output supply current	$I_O = 0$		All outputs at high level	2	6	mA
				All outputs at low level	2	4	
				All outputs at high impedance	2	4	
					2	4	

switching characteristics, $V_{CC1} = 5\text{ V}$, $V_{CC2} = 24\text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output from A input	$C_L = 30\text{ pF}$, See Figure 1		800		ns
t_{PHL}	Propagation delay time, high-to-low-level output from A input			400		ns
t_{TLH}	Transition time, low-to-high-level output			300		ns
t_{THL}	Transition time, high-to-low-level output			300		ns

PARAMETER MEASUREMENT INFORMATION



- NOTES: A. The pulse generator has the following characteristics: $t_r \leq 10\text{ ns}$, $t_f \leq 10\text{ ns}$, $t_w = 10\text{ }\mu\text{s}$, $\text{PRR} = 5\text{ kHz}$, $Z_O = 50\text{ }\Omega$
 B. C_L includes probe and jig capacitance.

Figure 1. Test Circuit and Voltage Waveforms



POST OFFICE BOX 655303 * DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L293D QUADRUPLE HALF-H DRIVER

SLRS008A – SEPTEMBER 1986 – REVISED MAY 1990

APPLICATION INFORMATION

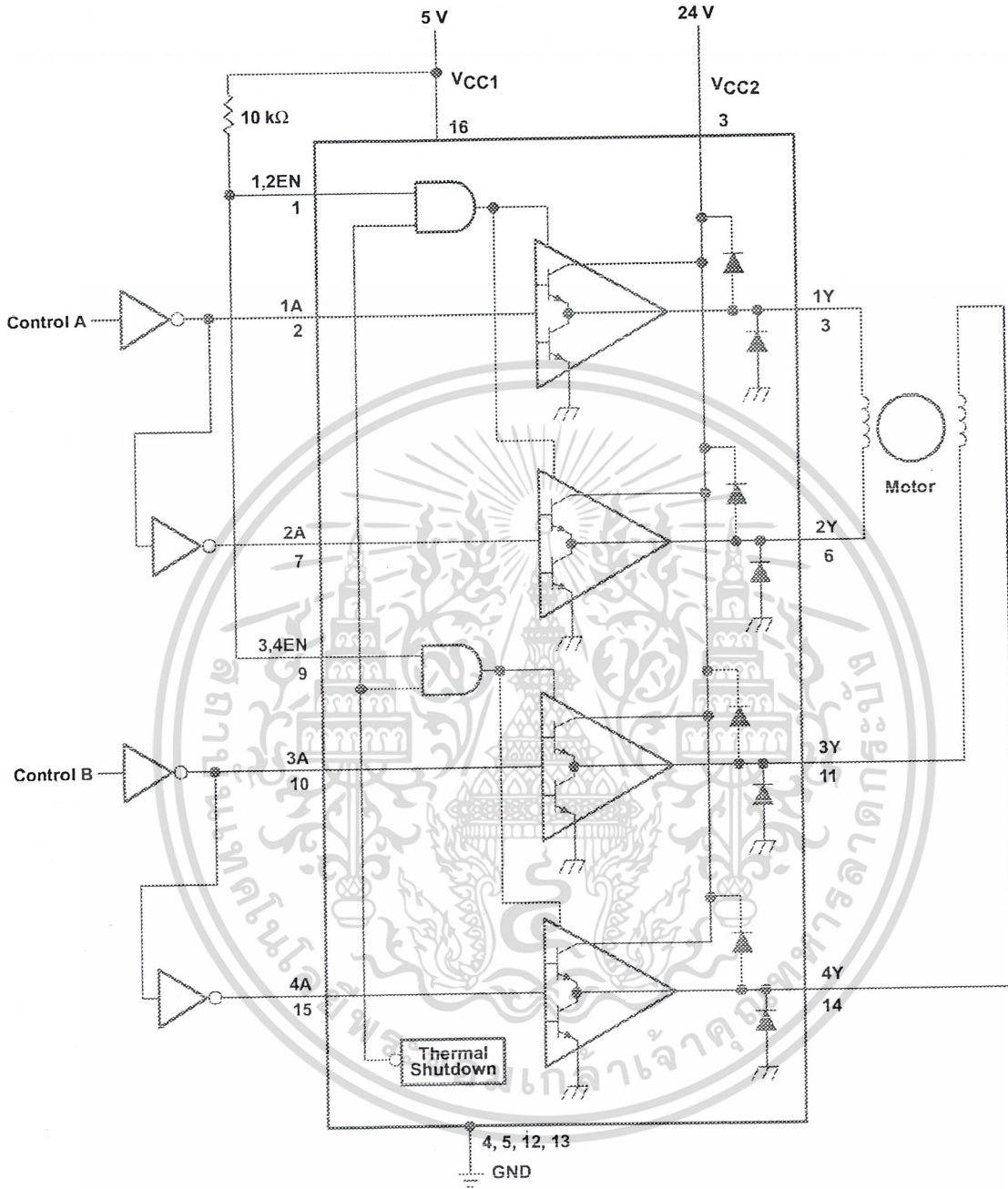


Figure 2. Two-Phase Motor Driver



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้