

การควบคุมแผงเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์โดยใช้ไมโครคอนโทรลเลอร์
Solar Tracking Controlled Panel Using Microcontroller



โดย
นายกฤษณะ คงวัลย์
นายอิสรา บุญญานวัตร

เลขหม.....
เลขทะเบียน 45713
วัน, เดือน, ปี 1 3 ก.พ. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2544

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมแผงเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์โดยใช้ไมโครคอนโทรลเลอร์
(Solar Tracking Controlled Panel Using Microcontroller)

ผู้จัดทำ

1. นายกฤษณะ คงวัลย์ รหัส 42015252
2. นายอิสรา บุญญานันต์ รหัส 42015292

..... อาจารย์ที่ปรึกษา
(อาจารย์วรพงษ์ ตั้งศรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การควบคุมแผงเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์ โดยใช้ไมโครคอนโทรลเลอร์
นักศึกษา	นายกฤษณะ คงวัลย์ นายอิสรา บุญญาอนุวัตร
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์วรงค์ ตั้งศรีรัตน์
ระดับการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม
ภาควิชา	วิศวกรรมระบบควบคุม
ปีการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2544

บทคัดย่อ

โครงการนี้นำเสนอการควบคุมแผงเซลล์แสงอาทิตย์ซึ่งเคลื่อนที่ตามแสงอาทิตย์ โดยใช้ไมโครคอนโทรลเลอร์ ระบบที่นำเสนอประกอบด้วยองค์ประกอบหลักที่สำคัญดังนี้คือไมโครคอนโทรลเลอร์ MCS51 บอร์ดรีเลย์อนเนกประสงค์ และดีซีมอเตอร์ นอกจากนี้หลักการดังกล่าวยังสามารถนำไปประยุกต์ใช้งานด้านการควบคุมการทำงาน เครื่องใช้ไฟฟ้า หรือเครื่องจักรในโรงงานอุตสาหกรรมอื่นๆ ได้อีกมากมาย

ABSTRACT

The design technique for controlling the solar tracking solar cell panel using microcontroller is presented in this project.the proposed system mainly consists of a microcontroller MCS51, mutipurpose relay board and DC motors with gearbox. In addition,this technique is also applied and is expected tobe useful in other application, such as, for some appliance,for industrial machine etc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญ	II
สารบัญภาพ	IV
บทที่ 1 บทนำ	1
1.1 ความสำคัญและความเป็นมาของโครงการ	1
1.2 จุดประสงค์ของปริญญาานิพนธ์	1
1.3 ขอบเขตของโครงการ	2
บทที่ 2 ทฤษฎีของ I²C	3
2.1 ความรู้เบื้องต้นเกี่ยวกับ I ² C	3
2.2 คุณสมบัติทั่วไปของบัส I ² C	4
2.3 หลักการของบัส I ² C	5
2.4 การทำงานของบัส I ² C	7
2.5 อุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I ² C	8
2.6 การเขียนโปรแกรมเชื่อมต่อบัส I ² C	9
บทที่ 3 ไมโครคอนโทรลเลอร์ MCS-51	11
3.1 คุณสมบัติของไมโครคอนโทรลเลอร์	11
3.2 สถาปัตยกรรมภายในของ MCS-513.3 พอร์ทของ 8051	11
3.4 วงจรสร้างสัญญาณนาฬิกาของ 8051	18
3.5 ฟังก์ชันสัญญาณเวลาซีพียู	20
3.6 การแบ่งประเภทของหน่วยความจำ	22
บทที่ 4 ทฤษฎี ADC และ DAC	24
4.1 ทฤษฎีของ Data Acquisition and Conversion	24
4.2 ทฤษฎีการ Sampling	24
4.3 การสุ่มและคงค่าสัญญาณ (Sample and Hold) และ Aperture error	26
4.4 Quantizing Theory	27
4.5 Quantizer Resolution and Error	30
4.6 รหัสตัวเลขสำหรับการเปลี่ยนข้อมูล	30
4.7 วงจรสุ่มและคงค่าสัญญาณ (Sample and Hold,S&H)	31
4.8 วงจรสุ่มแบบ Inverting Close Loop	33
4.9 วงจรแรงดันอ้างอิง (Voltage Reference Circuit)	34

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10 วงจร Digital to Analogue Converter (DAC)	38
4.11 Analog to Digital Converter	41
บทที่ 5 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลแบบซีกเซลซีฟแอฟหรือกซิมเข้	45
5.1 ความเที่ยงตรงของวงจร ADC	46
5.2 ค่าเวลาในการแปลงสัญญาณ	47
5.3 ข้อมูลเบื้องต้นของ PCF8591	47
บทที่ 6 แผงเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์	52
6.1 ชุดแผงติดตั้งเซลล์แสงอาทิตย์	52
6.2 วงจรควบคุมชุดแผงติดตั้งเซลล์แสงอาทิตย์	53
6.3 โปรแกรมควบคุมชุดแผงติดตั้งเซลล์แสงอาทิตย์	61
บทที่ 7 บทวิจารณ์และสรุป	64

กิตติกรรมประกาศ

บรรณานุกรม

ภาคผนวก

ก.โปรแกรมควบคุม

ข.คู่มือไอซี

สารบัญรูปภาพ

รูปที่	หน้า
2.1	3
2.2	4
2.3	5
2.4	6
2.5	7
2.6	9
3.1(ก)	12
3.1(ข)	13
3.2	13
3.3	14
3.4	15
3.5	16
3.6	17
3.7	18
3.8	19
3.9	20
3.10	21
3.11	22
3.12	22
3.13	23
4.1	25
4.2	25
4.3	28
	28
	28
4.4	29
4.5	31
4.6(ก)	32
4.7	33
4.8	34

รูปที่	หน้า
4.9 วงจรสัณญาณแบบ Non-inverting closed loop	35
4.10 วงจรแรงดันอ้างอิงพื้นฐานและกราฟคุณสมบัติ	35
4.11 วงจรแรงดันอ้างอิงใช้ซีเนอร์ไดโอดที่ปรับปรุงคุณสมบัติจากวงจรอ้างอิงพื้นฐาน	36
4.12 แรงดันอ้างอิง (ก)แรงดันอ้างอิงบวก (ข)แรงดันอ้างอิงลบ	36
4.13 แรงดันอ้างอิงปรับค่าได้	37
4.14 วงจร Band gap voltage reference	38
4.15 ADC แบบ Binary weight ladder	39
4.16 วงจร DAC แบบ R-2R ขนาด 4 บิต	40
4.17 วงจรรีซิสทีฟแลดเดอร์ (Resistive LadderX	40
4.18 วงจร Invert R-2R ladder DAC	41
4.19 วิธีการพื้นฐานของ DAC	42
4.20 Tranfer function ของ Comparater	42
4.21 บล็อกไดอะแกรมของ Sucessive appoximation ADC	43
4.22 สัญญาณเวลาของ SAR	43
4.23 บล็อกไดอะแกรมของ Parallel ADC	44
5.1 ไดอะแกรมการทำงานของวงจร ADC แบบซัสเซสซีฟแอปพริอักษิเมชัน	46
5.2 ไดอะแกรมเวลาการทำงานของวงจร ADC แบบซัสเซสซีฟแอปพริอักษิเมชัน	47
5.3 การจัดขาของไอซี ADC\DAC ขนาด 8 บิตผ่านบัส I ² C เบอร์ PCF8591	48
5.4 ตัวอย่างการเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับ PCF8591	51
6.1 ชุดติดตั้งเซลล์แสงอาทิตย์	52
6.2 วงจรควบคุม	53
6.3 วงจรตัวต้านทานเปลี่ยนค่าตามแสง	54
6.4 วงจรควบคุมชุดติดตั้งเซลล์แสงอาทิตย์	55
6.5 ลายทองแดงบนแผ่นวงจรค้ำบน	56
6.6 ตำแหน่งการวางอุปกรณ์บนแผ่นวงจร	57
6.7 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล	58
6.8 วงจรลิมิตสวิดซ์	59
6.9 ชุดขับมอเตอร์พร้อมกลับทิศทางหมุน	60
6.10 วงจร AT89s8252	60
6.11 โปรแกรมควบคุมแผงติดตั้งเซลล์แสงอาทิตย์	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ในโครงการนี้ เป็นการศึกษารูปแบบการควบคุมรีเลย์ เพื่อควบคุมการทำงานของเครื่องใช้ไฟฟ้า อุปกรณ์ไฟฟ้าหรือเครื่องจักรในโรงงานอุตสาหกรรมโดยอัตโนมัติซึ่งมีความจำเป็นมาก โดยเฉพาะอย่างยิ่งงานที่มีความเสี่ยงสูงในโรงงานอุตสาหกรรมดังนั้นการเข้าใจพื้นฐานการควบคุมอุปกรณ์อัตโนมัติจึงเป็นสิ่งจำเป็น

1.2 วัตถุประสงค์ของโครงการ

ในโครงการนี้เป็นการประยุกต์ใช้รีเลย์ควบคุมการทำงานของ มอเตอร์ที่ประกอบอยู่ในชุดแผ่นรับพลังงานแสงอาทิตย์ โดยการตรวจสอบความเข้มของแสงเพื่อปรับให้แผ่นรับพลังงานแสงอาทิตย์หมุนตามดวงอาทิตย์เพื่อให้ได้ปริมาณพลังงานแสงอาทิตย์มากที่สุดเท่าที่เป็นไปได้การควบคุมรีเลย์อาจใช้การควบคุมโดยการเขียนโปรแกรมคอมพิวเตอร์ควบคุม เช่น Microsoft Visual C หรือ Microsoft Visual Basic ผสมกับไมโครคอนโทรลเลอร์ หรือใช้ไมโครคอนโทรลเลอร์เพียงอย่างเดียว ขึ้นอยู่กับความเหมาะสมของงานที่ประยุกต์ใช้ โครงการนี้เลือกการควบคุมผ่านไมโครคอนโทรลเลอร์ 89AT8252 ซึ่งเป็น CPU 16bit ในตระกูล 8051 เพียงอย่างเดียวเนื่องจากแผ่นรับพลังงานแสงอาทิตย์เป็นงานสนามจึงไม่สะดวกในการเชื่อมต่อกับคอมพิวเตอร์

การตรวจสอบค่าความเข้มของแสงโดยใช้ตัวต้านทานปรับค่าตามแสง (LDR) เป็นตัวเปรียบเทียบแสงจากดวงอาทิตย์ ความต้านทานของตัวต้านทานปรับค่าตามแสงจะถูกแปลงเป็นแรงดันเพื่อส่งไปยัง วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลขนาด 8 บิต เปรียบเทียบค่าความแตกต่างของความเข้มของแสง เพื่อควบคุมการหมุนของ มอเตอร์ เพื่อปรับทิศทางแผ่นรับพลังงานแสงอาทิตย์

1.3 ขอบเขตของโครงการ

ขั้นตอนการทำงานทั้งหมด สามารถสรุปได้ดังนี้

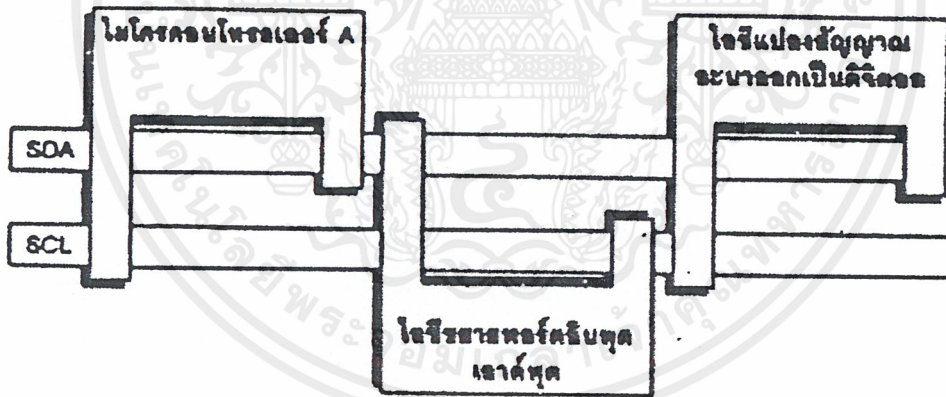
1. ศึกษาทฤษฎีการทำงานของ ไมโครคอนโทรลเลอร์ สถาปัตยกรรม คำสั่ง รวมทั้งการออกแบบวงจรร่วมกับอุปกรณ์ตัวอื่นๆ
2. ศึกษาทฤษฎีการทำงานของวงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิทัลพื้นฐาน การประยุกต์การใช้งาน รวมทั้งการใช้งาน PCF8591 A/D and D/A Converter
3. ออกแบบวงจรควบคุมโดยใช้ทฤษฎีข้างต้นเพื่อควบคุมการทำงานของรีเลย์ พร้อมเขียนโปรแกรม แอสเซมบลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีของ I²C

2.1 ความรู้เบื้องต้นเกี่ยวกับ I²C

I²C ย่อมาจาก Inter-IC Communication หมายถึงการติดต่อสื่อสารระหว่างไอซีโดยบัส I²C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อทำงาน และควบคุมภายใต้สายสัญญาณ 2 เส้นเส้นหนึ่งสายข้อมูลอีกเส้นหนึ่งสายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I²C ทำได้ง่ายมากเพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไปส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัวจะใช้รหัสข้อมูลและการกำหนดสถานะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว สายข้อมูลบนบัส I²C มีชื่อเรียกอย่างเป็นทางการว่าสายข้อมูลอนุกรมหรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่าสายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock Line) ในการอธิบายต่อไปนี้จะเรียกสายสัญญาณทั้งสองว่า สาย SDL และ สาย SCL



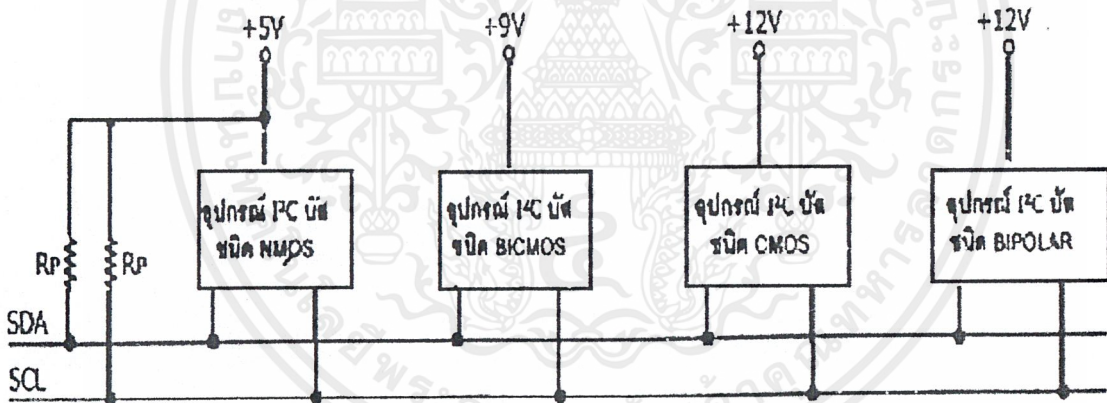
รูปที่ 2.1 แสดงการเชื่อมต่อของอุปกรณ์ต่างๆบนระบบบัส I²C

ในรูปที่ 2.1 แสดงผังการเชื่อมต่ออุปกรณ์ต่างๆบนบัส I²C จะเห็นได้ว่าอุปกรณ์ที่ทำการเชื่อมต่อบนบัส I²C มีหลากหลาย ไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander), ไอซีแปลงสัญญาณอนาลอกเป็นดิจิทัล (ADC) และแปลงสัญญาณดิจิทัลเป็นอนาล็อก(DAC), ไอซีรีลไทม์ล็อก(RTC), ไอซีขับ โมดูล (LCD), หน่วยความจำอีอีพรอม, ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คุณสมบัติโดยทั่วไปของบัส I²C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอัพที่แรงดัน +5 V ไว้ตลอดเวลาเพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งานทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาท์พุทของอุปกรณ์ที่ต่ออยู่บนบัส I²C ต้องมีลักษณะเป็นวงจรเรณเปิด(open-drain)หรือคอลเล็กเตอร์เปิด (open-collector) อัตราการถ่ายเทข้อมูลบนบัส I²C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง(fast mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I²C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400pF การเข้าถึงอุปกรณ์บนบัส I²C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต(7-bit addressing)หรือ 10 บิต(10-bit addressing)ข้อดีอีกประการหนึ่งของบัส I²C คือสามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้โดยอุปกรณ์บนบัส I²C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบน I²C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงต่างกัน กล่าวคือให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัพ (Rp)เข้ากับแรงดัน +5V ไว้เสมอ ดังแสดงในรูปที่ 2.2

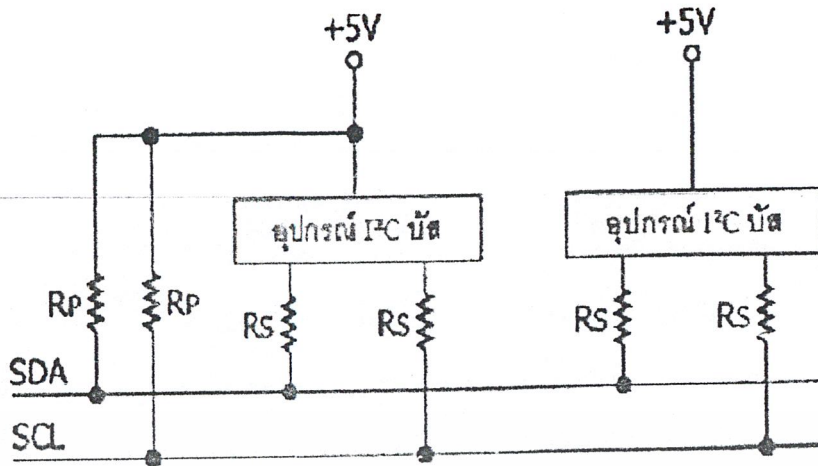


รูปที่ 2.2 การต่อตัวต้านทานพูลอัพบนสายสัญญาณในระบบบัส I²C

ในกรณีที่อาจจะมีแรงดันไฟกระชากขนาดใหญ่ ปะปนเข้ามาในบัส I²C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า Rsก่อนต่อเข้าสู่บัส I²C ดังแสดงในรูปที่

2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การต่อตัวต้านทาน R_s เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส I²C

2.3 หลักการของบัส I²C

บัส I²C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงมีการกำหนดรูปแบบของการติดต่อบนบัสหรือเรียกว่าโปรโตคอล (protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้มีอุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่งต่อไปนี้จะขออธิบายลักษณะ หน้าที่และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่าตัวส่ง (transmitter) อุปกรณ์ที่เป็นผู้รับข้อมูลเรียกว่าตัวรับ (receiver) ในอุปกรณ์บนบัส I²C สามารถเป็นได้ทั้งตัวรับและตัวส่งบางอุปกรณ์ทำหน้าที่เป็นดังรับอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I²C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียวอุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I²C เรียกว่า มาสเตอร์ (master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I²C เรียกว่า สเตฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I²C คือ

- (1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น
- (2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลจะรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

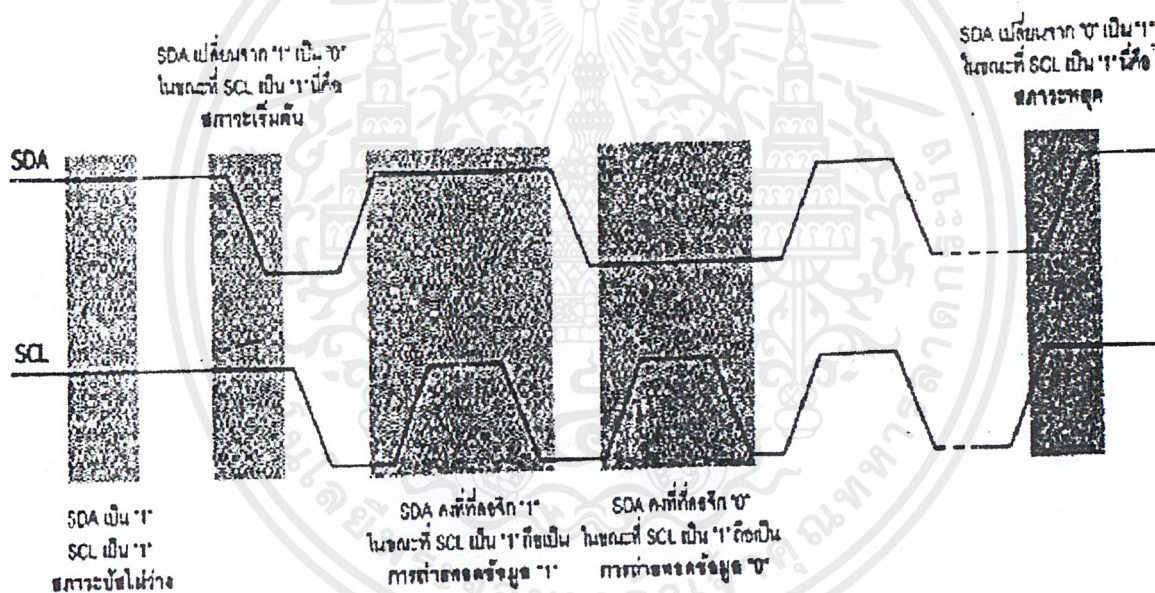
สถานะที่เกิดขึ้นบนบัส I²C มีด้วยกัน 5 สถานะ ดังนี้

- (1) บัสว่าง (bus not busy) สถานะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้
- (2) เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นนี้ว่า สถานะเริ่มต้น (START)
- (3) หยุดการถ่ายทอดข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นนี้ว่า สถานะหยุด (STOP)

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) หยุดการถ่ายทอข้อมูล(stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการแปลงเปลี่ยนระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นว่า สภาวะหยุด (STOP)

(4) ข้อมูลค้างอยู่บนบัส(data valid)สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอ เมื่อสาย SCL เป็นลอจิกสูงสถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น “0”หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ทำให้ข้อมูลที่ทำการถ่ายทออาจผิดพลาดไป



รูปที่ 2.4 ไตอะแกรมเวลาแสดงสถานะต่างๆในบัส I²C

(5) รับรู้ข้อมูล(acknowledge)เกิดขึ้นหลังจากที่การถ่ายทอข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่าบิตรับรู้(acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกาเพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้

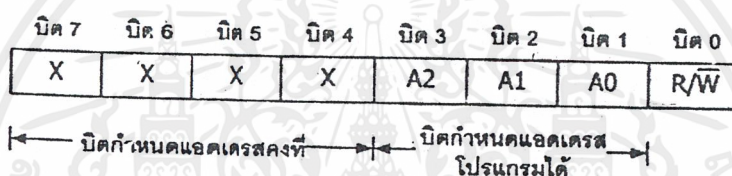
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทราบว่าได้รับข้อมูลในแต่ละบิตเรียบร้อยแล้วในรูปแบบที่ 2.4 เป็นไคอะแกรมเวลาที่แสดงถึงการเกิดสถานะต่างๆบนบัส I²C ไม่ว่าจะ เป็นสถานะบัสว่าง เริ่มต้น ถ่ายทอดข้อมูล รับรู้ และหยุดการถ่ายทอดข้อมูล

2.4 การทำงานบนบัส I²C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างต่างๆที่อยู่บนบัส ต้องมีการอ้างถึงเสียก่อนโดยการอ้างถึงอุปกรณ์บนบัส I²C นั้นจะใช้การอ้างถึงแบบ 7 บิตหรือ 10 บิต ในกรณีที่มีอุปกรณ์ที่อยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามีอุปกรณ์ที่อยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงถึงแบบ 10 บิต หลังจากที่ตั้งคต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้วก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้นหัวใจสำคัญอันดับแรกของการทำงานบนบัส I²C คือการอ้างถึงอุปกรณ์แต่ละตัว ซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ



รูปที่ 2.5 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต

2.4.1 การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ทำการติดต่อหรือข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 2.5 ใน 7 บิตบนรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็น บิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้กับขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I²C ส่วนในแบบ LSB เป็นที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นี้ ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมแตกต่างกันไป ยกตัวอย่าง ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง (data) หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การอ้างถึงแบบ 10 บิต

ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกันแบบ 7 บิต หากแต่จะมีข้อมูลอนุกรมที่เหมือนกันแบบ 7 บิตหากแต่ยังมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสถานะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อกับข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อกับข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะจะเป็นข้อมูลจริงที่ใช้ในการติดต่อ เช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากการถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสถานะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

2.5 อุปกรณ์ที่ใช้การเชื่อมต่อแบบ บัส I²C

ในปัจจุบันบัส I²C ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ด้วยข้อดีที่ชัดเจนคือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้น และการขยายระบบไมโครคอนโทรลเลอร์ ที่มีจำนวนอินพุตเอาต์พุตและหน่วยความจำจำกัดสามารถทำได้ง่ายขึ้นด้วยระบบบัส I²C เมื่อเป็นเช่นนี้จึงมีอุปกรณ์เพอร์IPHERAL ที่ใช้ในการเชื่อมต่อแบบบัส I²C มากมายจากหลายผู้ผลิตออกมาให้ใช้ได้ใช้งาน ดังมีตัวอย่างต่อไปนี้

ไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O expander): PCF8574,PCF8582,PCF8584

ไอซีหน่วยความจำอีอีพรอมอนุกรม (serial EEPROM): 24Cxx,PCF8570,PCF72/73,PCF8582

ไอซี ADC/DAC : PCF8591

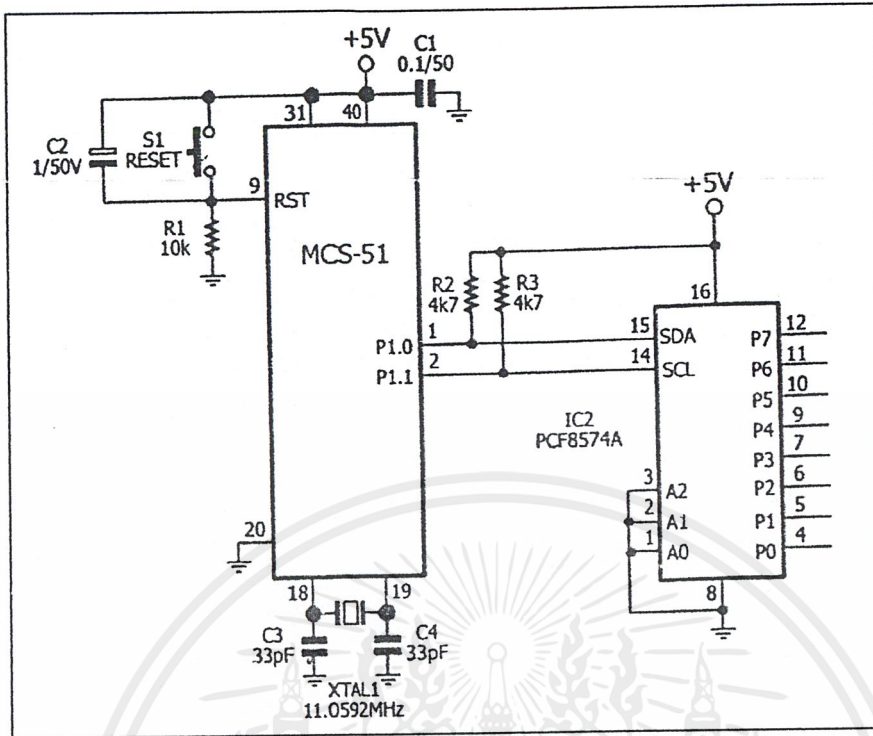
ไอซีไทม์ค็อก (Real-time clock;RTC): PCF8583,PCF8593,PCF8598,41T56c

ไอซีขับ LCD โมดูล (LCD driver) : PCF8466,PCF8576,PCF8577/78,PCF8579,SAA1064

ไอซีกำเนิดสัญญาณ DTMF (DTMF generator) :PCD3311/12

2.5.1 การต่ออุปกรณ์ระบบบัส I²C กับ ไมโครคอนโทรลเลอร์ MCS-51

สามารถทำได้ เพียงใช้ขาพอร์ต 2 ขา โดยกำหนดให้ขาหนึ่งเป็น SDA อีกขาหนึ่งเป็น SCL และต่อตัวต้านทานค่าประมาณ 4.7k พูลอัพที่ขาพอร์ตทั้งสองขา เพียงเท่านี้ก็สามารถติดต่อกับอุปกรณ์รับบัส I²C ได้แล้ว



รูปที่ 2.6 วงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 กับอุปกรณ์ระบบบัส

ในรูป 2.6 เป็นวงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 เข้ากับระบบบัส I²C จากวงจรจะใช้ขาพอร์ต P1.0 เป็นขา SDA และ P1.1 เป็นขา SCL อุปกรณ์ที่ทำการติดต่อก็คือไอซีขยายพอร์ตอินพุทเอาต์พุทเบอร์ PCF8574

2.6 การเขียนโปรแกรมติดต่อบัส I²C

เริ่มต้นด้วยการสร้างสภาวะมาตรฐานของบัส I²C อันประกอบด้วย สภาวะเริ่มต้น สภาวะสิ้นสุดการส่งข้อมูล สภาวะหยุด และสัญญาณนาฬิกาบนขา SCL

การสร้างสภาวะหยุด

1. เมื่อต้องการหยุดส่งข้อมูลจะต้องส่งสภาวะหยุดออกไป โดยในเงื่อนไขแรกต้องกำหนดให้ขา SCL และ SDA เป็นลอจิก “0” ทั้งคู่ก่อน
2. กำหนดให้ขา SCL มีลอจิกเป็น “1” โดย SDA ยังคงมีลอจิกเป็น “0”
3. จากนั้นทำให้ขา SDA มีลอจิกเป็น “1” ซึ่งจะทำให้ระบบบัสกลับเข้าสู่สภาวะสว่างอีกครั้ง พร้อมที่จะรับหรือส่งข้อมูลต่อไป

การส่งข้อมูลลอจิก “0” และลอจิก “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลลोजิก “0” และลोजิก “1”

หลังจากที่ทำการส่งบิตเริ่มต้นแล้ว ลำดับต่อไปคือ จะต้องส่งข้อมูลควบคุมซึ่งจะเป็นขบวนของลोजิก “0” และลोजิก “1” สำหรับการส่งข้อมูลลोजิก “0” ต้องดำเนินการตามขั้นตอนดังนี้

1. ทำให้ขา SDA เป็น “0” สำหรับการส่งข้อมูลลोजิก “0”
2. ทำให้ขา SCL เป็น “1” สำหรับการป้อนสัญญาณนาฬิกา ในขณะที่ขา SDA ยังคงเป็น “0” อยู่
3. จากนั้นทำให้ขา SCL กลับมามีสถานะเป็นลोजิก “0” เหมือนเดิม

ในขณะที่การส่งข้อมูลลोजิก “1” มีขั้นตอนดังนี้

1. ทำให้ขา SDA มีลोजิกเป็น “1” สำหรับการส่งข้อมูลลोजิก “1”
2. ทำให้ขา SCL เป็น “1” สำหรับการส่งสัญญาณนาฬิกา ในขณะที่ขา SDA ยังคงเป็น “1” อยู่
3. จากนั้นทำให้ขา SCL เป็น “1” กลับมามีสถานะเป็นลोजิก “0” เหมือนเดิม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ไมโครคอนโทรลเลอร์ MCS-51

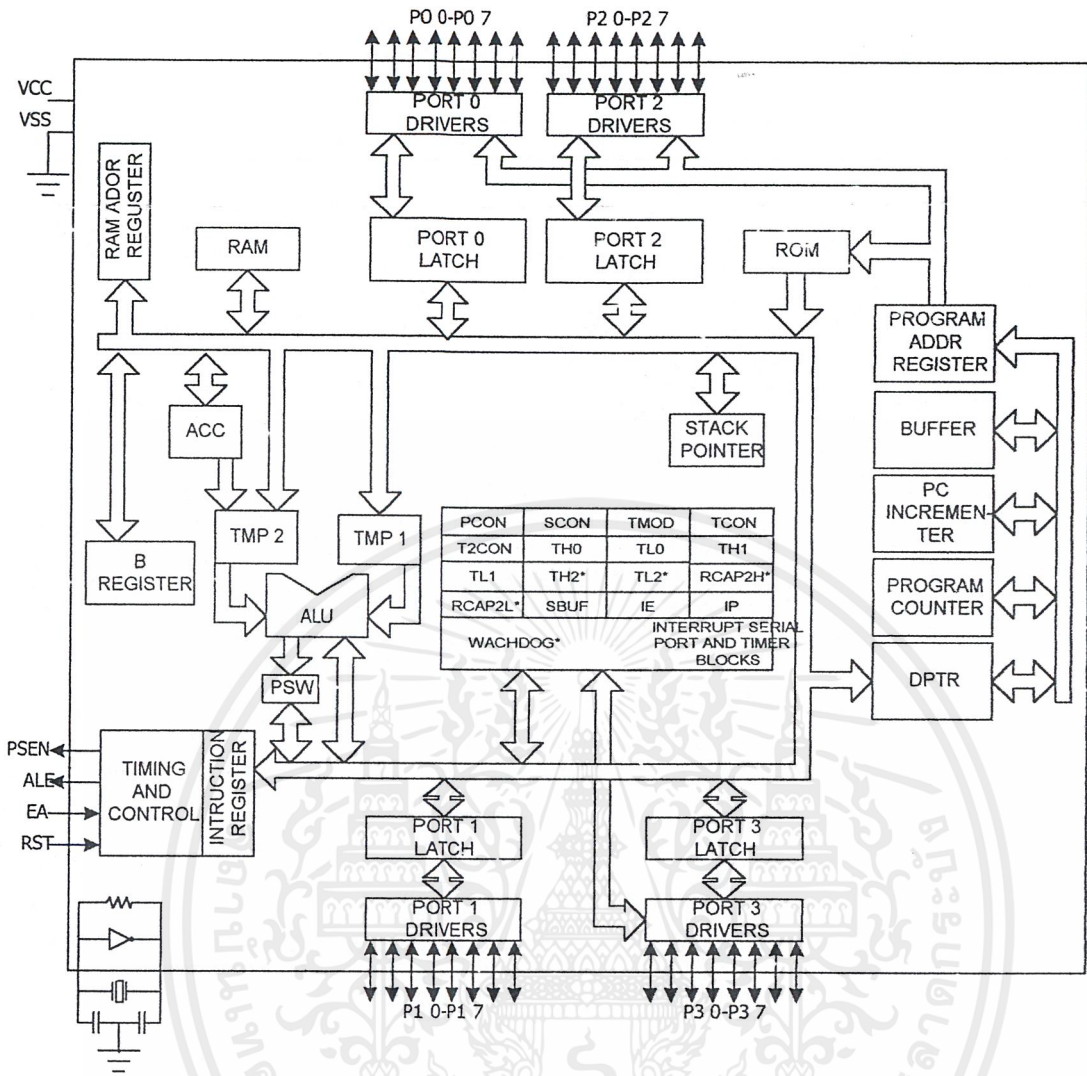
3.1 คุณสมบัติของไมโครคอนโทรลเลอร์

คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ มีรายละเอียดสำคัญดังต่อไปนี้

- แหล่งจ่ายไฟ + 5 V
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 8031 และ 8032 ไม่มีหน่วยความจำชุดนี้ ส่วน 8032 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับ 8052 มีถึง 256 ไบต์
- หน่วยความจำสำหรับ โปรแกรมและค่า (Program Memory และ Data Memory แยกจากกัน อย่างละ 64 กิโลไบต์)
- คำสั่งที่ใช้เวลาน้อยที่สุดประมาณ 1 μ s เมื่อทำงานที่ความถี่ 12 MHz
- มี Timer/Counter ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเทอร์รัพท์ได้ 6 แหล่ง 5 เวกเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ต ทั้งรับและส่งในเวลาเดียวกันได้ (Full Duplex) เลือกรูปแบบการส่งข้อมูลได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR หรือ COMPLEMENT ได้ทั้งแบบ 8 บิตและ 1 บิต

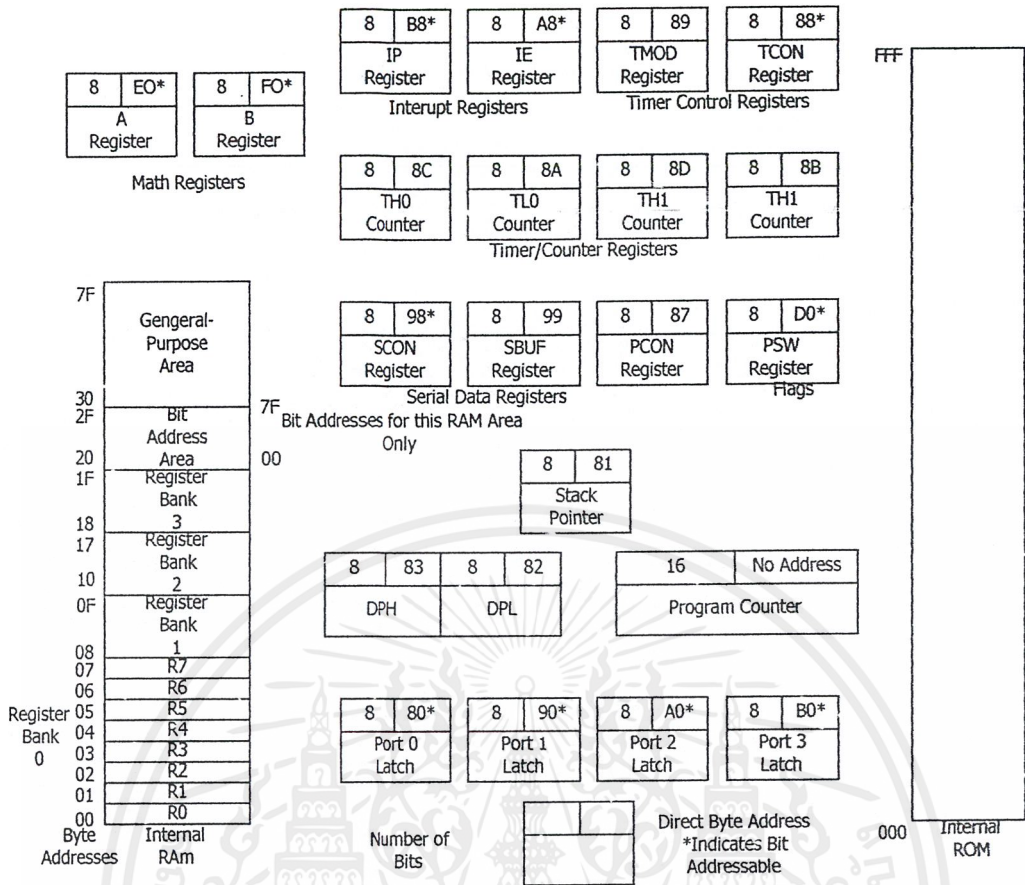
3.2 โครงสร้างภายในของ MCS-51

MCS-51 ใช้เทคโนโลยีในการผลิตแบบ NMOS และ CMOS เบอร์ 8032 และ 8052 จะมี ROM BASIC อยู่ภายในจึงสะดวก สำหรับโปรแกรมเมอร์ที่จะเขียนโปรแกรมด้วยภาษาเบสิก โครงสร้างสำหรับ เบอร์ 8051 ดังแสดงในรูปที่ 3.1 (ก) และ (ข) และหน่วยความจำเพื่อใช้ประกอบการเขียนโปรแกรม

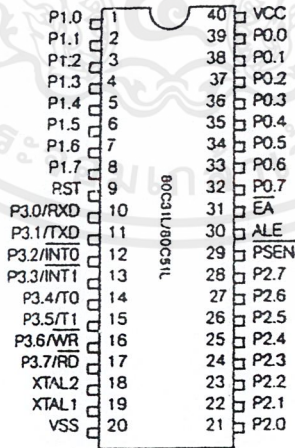


รูปที่ 3.1 (ก) 8051 บล็อกไดอะแกรมของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 (ข) ตำแหน่งของรีจิสเตอร์ต่างๆ



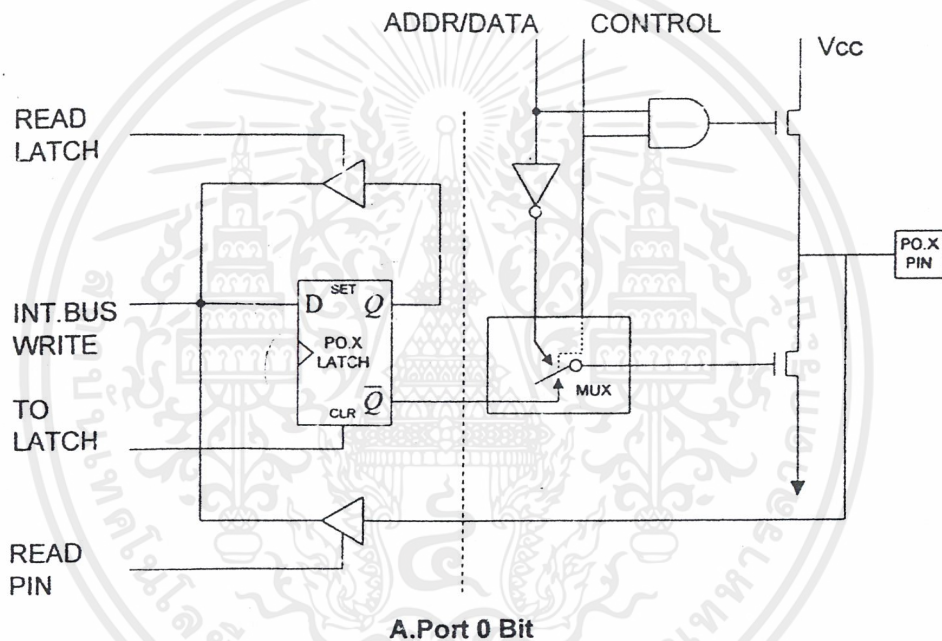
รูปที่ 3.2 การจัดวางขาของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 พอร์ตของ 8051

8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่าง ๆ ดังนี้

- Vcc (ขา 40) ต่อกับ + 5 V
- Vss (ขา 20) เป็นขา GND
- **PORT 0** (ขา 32-39) มีทั้งหมด 8 บิตคือ (P0.0-P0.7) มีโครงสร้างแบบ Open Drain Bi-directional ดังแสดงในรูปที่ 3.3

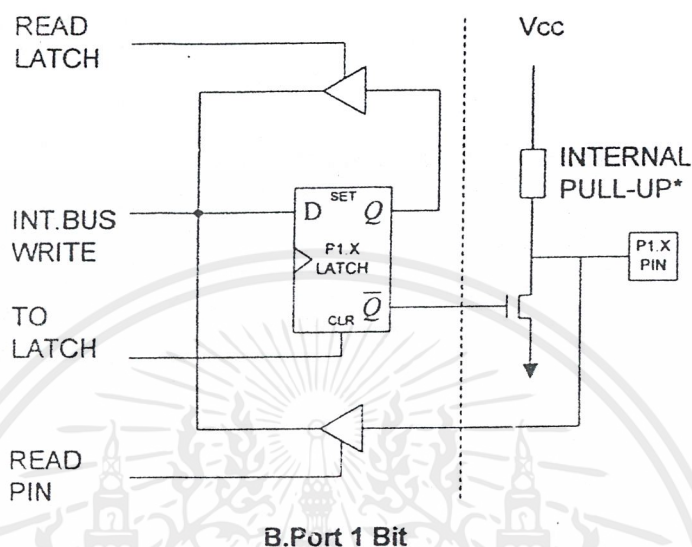


รูปที่ 3.3 โครงสร้าง พอร์ต 0 (บิต)

- พอร์ต 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.0-P0.7) ใช้งานได้ 2 หน้าที่คือ ส่งแอดเดรสและค่าข้อมูลให้หน่วยความจำภายนอกเมื่อทำการเขียนข้อมูลลงในหน่วยความจำภายนอกควบคุมด้วยขา Control รูป 3.3 ประกอบและอีกหน้าที่หนึ่งหน้าที่ก็คือ เป็นพอร์ต I/O ถ้าต้องการให้ทำงานเป็นอินพุตพอร์ตต้องส่งลอจิก “ 1 “ ไปยังพอร์ตนี จะมีผลให้ \bar{Q} ของ D-FF เป็น “ 0 “ ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุตพอร์ต PIN (พอร์ต P0.X PIN) จะใช้สัญญาณ READ LATCH เมื่อถูกกระตุ้นที่ Tri-State Buffer ตัวบน

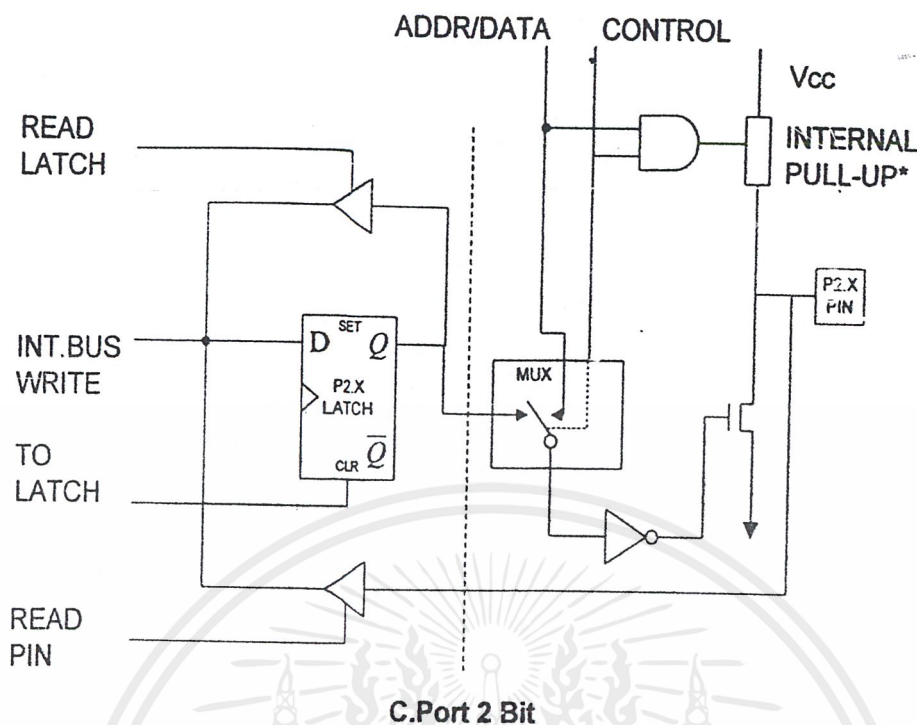
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้าย พอร์ต 0 แต่จะ ใช้ ความต้านทานภายในพูลอัพแทน (Internal Pull Up Register) มีโครงสร้างดังรูปที่ 3.4



รูปที่ 3.4 โครงสร้างของพอร์ต 1 (บิต)

- พอร์ต 2 (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.0-P2.7) มีโครงสร้างคล้าย PORT 0 โดยมี FET ตัวล่างตัวเดียวส่วนตัวด้านบนใช้ความต้านทานพูลอัพแทน (Internal PullUP) พอร์ตนี้ ทำงาน 2 หน้าที่คือ สามารถใช้เป็นพอร์ตสำหรับส่งแอสแตค 8 บิตบน (A8-A15) และเป็น I/O พอร์ตใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุตพอร์ต ต้องส่งลอจิก “ 1 “ มาที่พอร์ตนี้ ก่อนเพื่อบังคับให้ FET อยู่ในสถานะ OFF ดังแสดงในรูปที่ 3.5
- พอร์ต 3 (ขา 10-17) มีทั้งหมด 8 บิต คือขา (P3.0-P3.7) มีโครงสร้างคล้าย พอร์ต 1 พอร์ตนี้ ทำหน้าที่เป็น I/O พอร์ต ถ้าจะให้พอร์ตนี้เป็น I/O PORT ก็ส่งลอจิก “ 1 “ มาที่พอร์ตนี้และ อีกหน้าที่หนึ่งคือ ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไป สัญญาณต่าง ๆ มีดังนี้



รูปที่ 3.5 โครงสร้างของพอร์ท 2 (บิต)

P3.0/RXD (Serial Input Port) เป็นขาที่รับข้อมูลแบบอนุกรม

P3.1/RXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/ $\overline{INT0}$ (External Interrupt) ใ้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/ $\overline{INT1}$ (External Interrupt) ใ้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0 (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับสัญญาณไซเคิลของสัญญาณ T1 นี้หรือสัญญาณนาฬิกาก็ได้

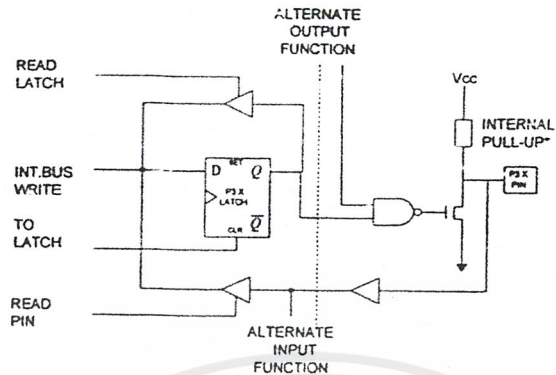
P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter 1 ซึ่งมีการทำงานเหมือน T0

P3.6/ \overline{WR} (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

P3.7/ \overline{RD} (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

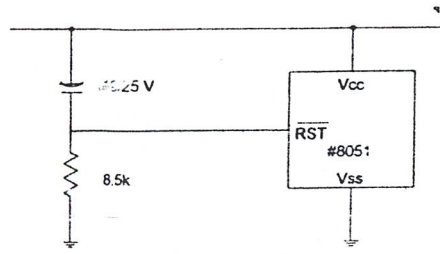
โครงสร้างของ พอร์ต 3 แสดงดังรูปที่ 3.6



รูปที่ 3.6 โครงสร้างของพอร์ต 3 (บิต)

- ALE (ขา 30) เป็นขาส่งสไตรบสำหรับใช้ในการแลตช์แอดเดรสไบร์ต่า (A0-A7) ที่ส่งออกมาจาก (พอร์ต 0) สัญญาณนี้จะแอกทีฟทุก ๆ 2 ครั้ง ใน 1 แมซซึนไซเคิล (1/6 ของสัญญาณนาฬิกา)
- $\overline{\text{PSEN}}$ (ขา 29) เป็นขาที่ใช้ส่งสไตรบสำหรับอ่านข้อมูลจาก Program Memory ภายนอก (หน่วยความจำประเภท ROM EPROM) สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมซซึนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้
- $\overline{\text{EA}}$ (ขา 30) ถ้าป้อนลอจิก “ 0 “ เข้ามาที่ขานี้ซีพียูจะอ่านค่าจาก Program Memory ภายนอกซิพเท่านั้น แต่ถ้าถูกป้อนด้วยลอจิก “ 1 “ ก็จะทำอ่านโปรแกรมภายในซิพ
- RST (ขา 9) เป็นขาเรซีตซีพียูจะเรซีตได้ก็ต่อเมื่อ ป้อนลอจิก “ 1 “ เข้ามาที่ขาอย่างน้อย 2 แมซซึนไซเคิล เมื่อซีพียูถูกรีเซตค่าต่าง ๆ ในรีจิสเตอร์ใด ๆ จะมีค่าตั้งคั้ง Table 3.1
- XTAL1 (ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรรอสซึเลเตอร์
- XTAL2 (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรรอสซึเลเตอร์

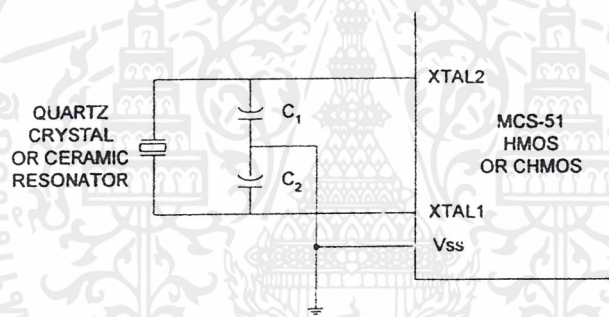
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



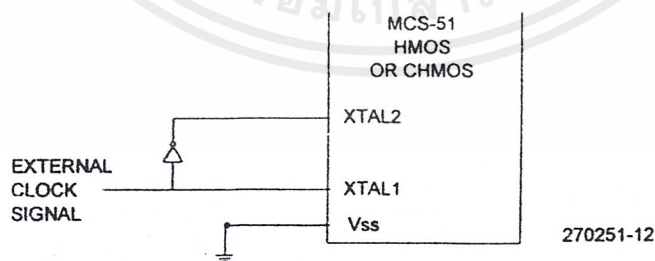
รูปที่ 3.7 การต่อขารีสตให้กับ 8051

3.4 วงจรสร้างสัญญาณนาฬิกาของ 8051

การต่อวงจรสร้างสัญญาณนาฬิกามีอยู่ด้วยกัน 2 รูปแบบ คือแบบภายในและภายนอกแสดงดังได้รูปที่ 3.8

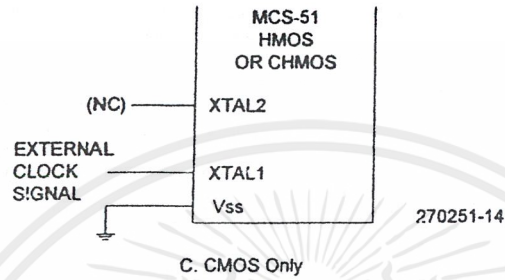
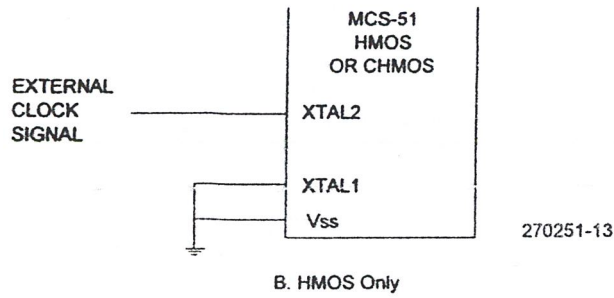


(a) Using the on-chip Oscillator



A. HMOS OR CHMOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

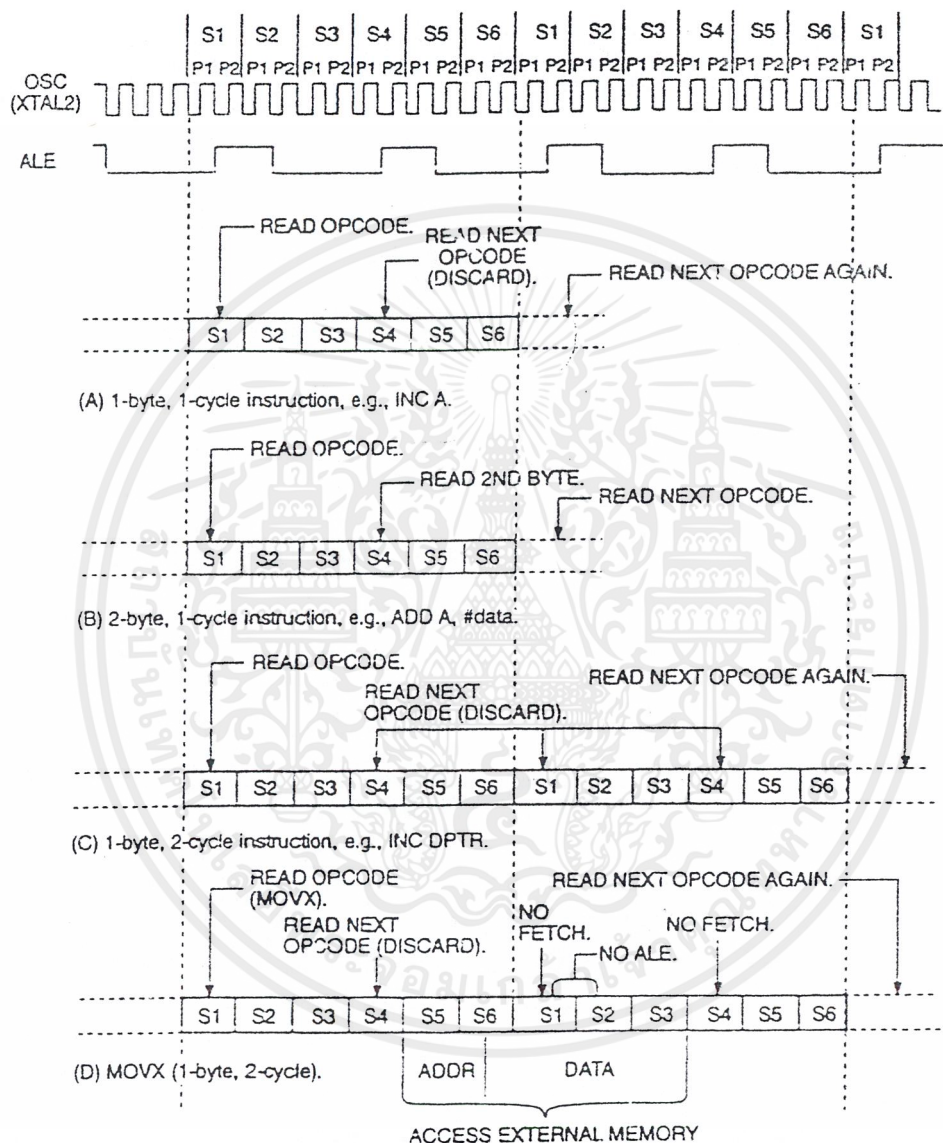


(b) Using the External Clock

รูปที่ 3.8 วงจรสร้างสัญญาณนาฬิกาของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ฟังก์ชันสัญญาณเวลาซีพียู (CPU Timing)



รูปที่ 3.9 ฟังก์ชันการทำงานของคำสั่ง

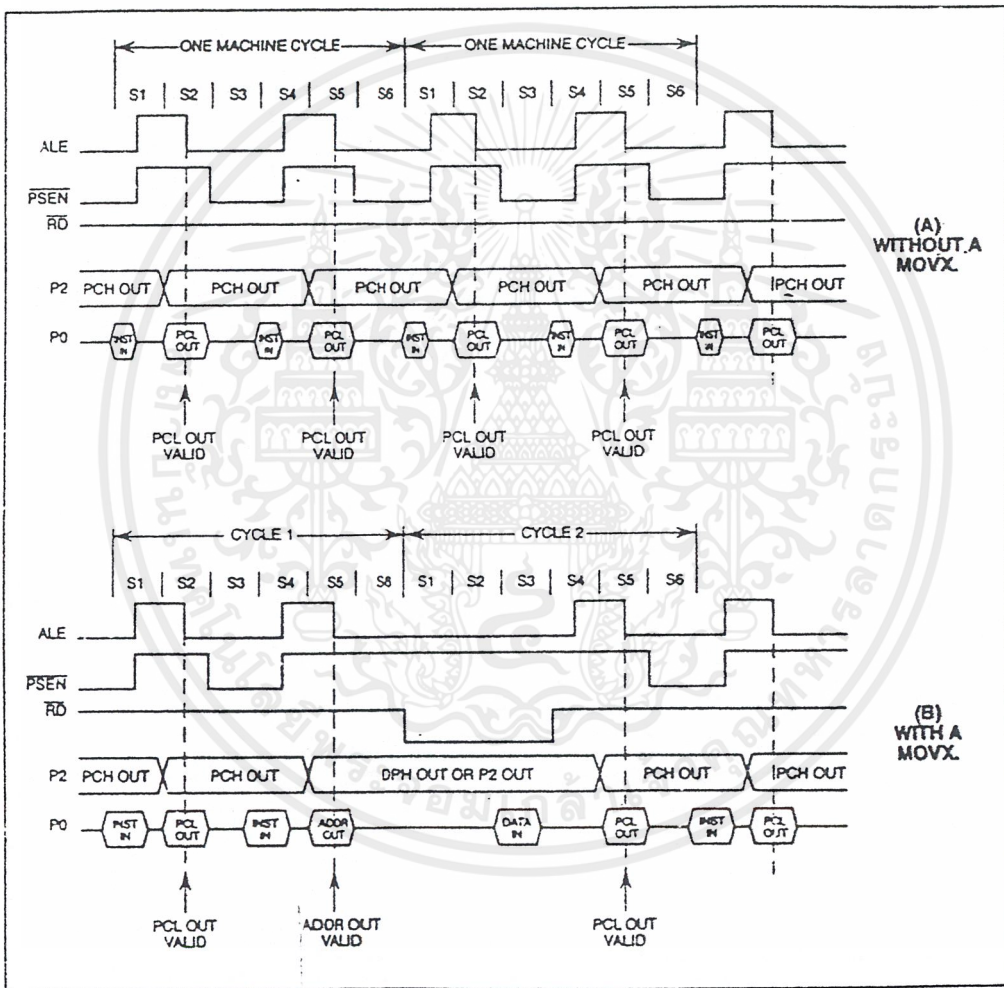
ซีพียูจะ RUN ด้วยความเร็วเท่ากับ 12 MHz ดังนั้น คล็อก 12 ลูกจะกินเวลาเท่ากับ $(1/12) * 12 = 1 \mu S$ ว่า 1 แมกซ์ไซเคิล คือช่วงการทำงานตั้งแต่ S1 จนถึง S6

รูปที่ 3.9 (a) แสดงการทำงานของคำสั่ง INT A ซึ่งเป็นคำสั่ง 1 ไบต์ทำงานเสร็จภายใน 1 แมกซ์ไซเคิล
 เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 (b) แสดงการทำงานของคำสั่ง ADD A.#Data ซึ่งเป็นคำสั่ง 2 ไบต์ทำงานเสร็จใน 1 แมชีน ไซเคิล

รูปที่ 3.9 (c) แสดงการทำงานของคำสั่ง INC DPTR ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 แมชีน ไซเคิล

รูปที่ 3.9 (d) แสดงการทำงานของคำสั่ง MOVX ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 แมชีน ไซเคิล



รูปที่ 3.10 แสดงผังเวลาการติดต่อกับหน่วยความจำภายนอก

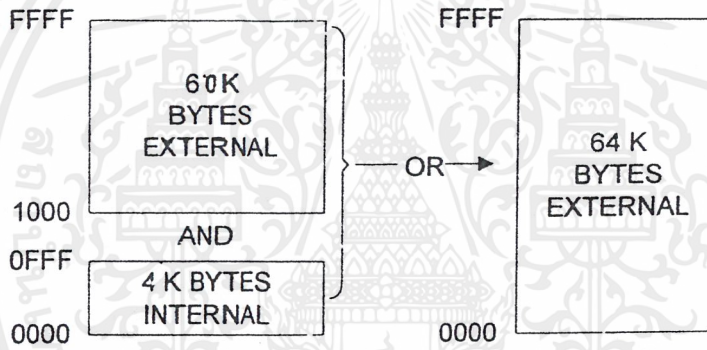
รูปที่ 3.10 (a) เป็นผังเวลาของสัญญาณซึ่งเกี่ยวข้องกับเฟิร์ทซ์เมื่อส่วนของ Program Memory อยู่นอก ภาวนอก คั้งนั้น สัญญาณที่จะนำมาใช้อ่าน OP-Code จาก Program Memory ก็คือ PSEN ซึ่งจะแอดคทีฟ 2 ครั้งใน 1 แมชีน ไซเคิล คั้งนั้นสัญญาณที่ใช้อ่านข้อมูลจาก Program Memory จะใช้สัญญาณ PSEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

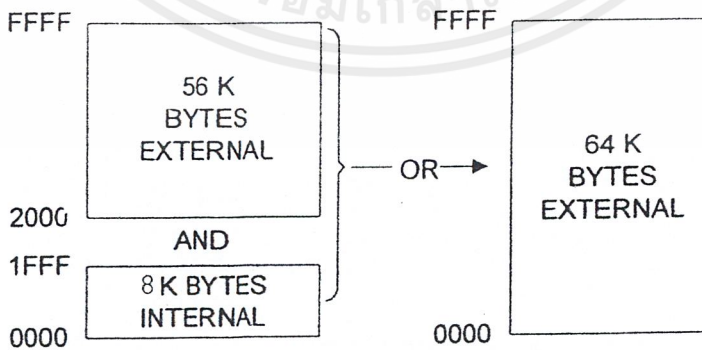
รูปที่ 3.10 (b) เป็นผังเวลาของสัญญาณที่เกี่ยวข้องกับการอ่านข้อมูลจาก Data Memory โดยใช้สัญญาณ \overline{RD} (อาจสรุปได้ง่าย ๆ ว่าการอ่านข้อมูลจาก Program Memory จะใช้สัญญาณ \overline{PSEN} และการอ่านข้อมูลจาก Data Memory จะใช้สัญญาณ \overline{RD} ส่วนสัญญาณ ALE คือสัญญาณที่ใช้ในการ Latch Address A0-A7 นั้นเอง)

3.6 การแบ่งประเภทของหน่วยความจำ

หน่วยความจำที่ใช้กับ MCS-51 มีอยู่ด้วยกัน 2 ชนิด คือ Program Memory และ Data Memory ซึ่งเป็นหน่วยความจำที่ใช้เก็บโปรแกรมทำงานบรรจุอยู่ในชิพ 8051 ส่วนที่เป็น Program Memory ก็คือ ROM ขนาด 4 กิโลไบต์นั่นเอง แต่ถ้าเป็นเบอร์ 8052 ก็คือ ROM ขนาด 8 กิโลไบต์ ดังแสดงในรูปที่ 3.11 และ ที่ 3.12

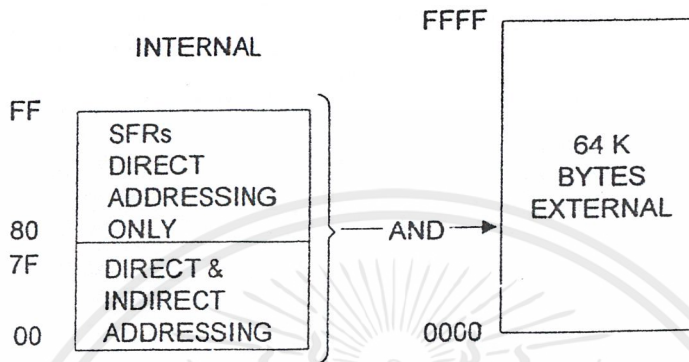


รูปที่3.11 ผังเก็บความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8051



รูป ที่3.12 ผังแสดงหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8052

Data Memory เป็นหน่วยความจำที่ใช้เก็บข้อมูลหน่วยความจำนี้ สามารถเขียนข้อมูลลงไป และอ่านข้อมูลออกมาได้ ซึ่งเป็นหน่วยความจำภายในชิปมีเพียง 128 ไบต์ สำหรับเบอร์ 8051 และ 256 ไบต์ สำหรับเบอร์ 8052 ส่วนหน่วยความจำภายนอกชิปมี 64 กิโลไบต์ ดังแสดงในรูปที่ 2.13 และที่ 3.14



รูปที่ 3.13 ผังหน่วยความจำสำหรับ Data Memory เบอร์ 8051

ทฤษฎีของ ADC และ DAC

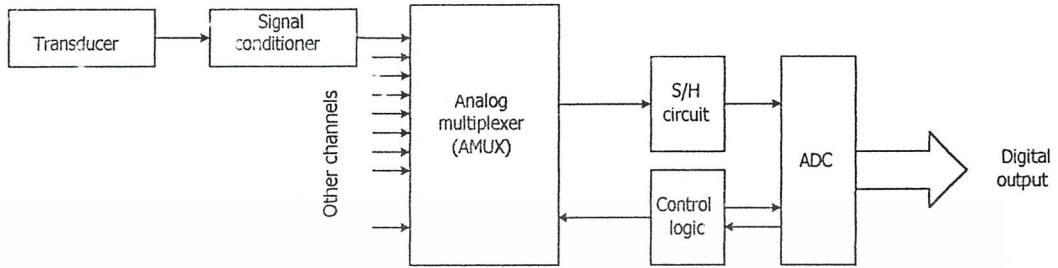
4.1 ทฤษฎีของ Data Acquisition and Conversion

รูปแบบสัญญาณ ไฟฟ้าที่เราพบเห็นและคุ้นเคยในชีวิตประจำวันนั้นจะอยู่ในรูปของสัญญาณที่ต่อเนื่องหรือที่เรียกว่าสัญญาณอนาล็อก (Analog signal) ซึ่งแต่เดิมการนำเอาสัญญาณไฟฟ้าดังกล่าวมาประมวลผล (Process) เพื่อให้มีรูปแบบที่เหมาะสมจะกระทำในแบบอนาล็อกนั่นเองแต่เมื่อเทคนิคและอุปกรณ์ประมวลผลสัญญาณทางดิจิทัลได้รับการพัฒนาขึ้นมา เนื่องจากพบว่าในรูปแบบดิจิทัล การประมวลผลข้อมูล สื่อสารและการนำเสนอกระทำได้ง่ายและมีประสิทธิภาพมากกว่า ดังนั้นการเปลี่ยนรูปแบบของสัญญาณ (Conversion) จึงได้มีความจำเป็นขึ้นมา ในรูปที่ 4.1 เป็นตัวอย่างแสดงระบบควบคุมที่ใช้การประมวลผลข้อมูลในระบบดิจิทัล ในระบบที่ยกมาเป็นตัวอย่างนี้เป็นการเปลี่ยนแปลงทางกายภาพ (Physical process) ในลักษณะใด ๆ ก็ตาม เช่น ความดัน อุณหภูมิ จะถูกเปลี่ยนให้เป็นกำลังไฟฟ้าที่มีความต่อเนื่อง (สัญญาณอนาล็อก) โดยทรานสดิวเซอร์ที่มีคุณสมบัติเหมาะสมกับรูปแบบทางกายภาพนั้น สัญญาณไฟฟ้านั้นจะถูกปรับให้อยู่ในรูปและขนาดที่เหมาะสมก่อน โดยวงจรส่วนที่ทำหน้าที่ปรับแต่งสัญญาณ เช่น วงจรขยาย วงจรฟิลเตอร์ เป็นต้น ADC จะทำหน้าที่เปลี่ยนรูปสัญญาณจากอนาล็อกเป็นดิจิทัล ตัวประมวลผลทางดิจิทัล (Digital processor) เช่น คอมพิวเตอร์ จะจัดการกับข้อมูลเพื่อนำเสนอหรือถูกเปลี่ยนกลับมาอยู่ในรูปแบบของสัญญาณอนาล็อกโดย DAC เพื่อป้อนกลับไปควบคุมการเปลี่ยนแปลงทางกายภาพ ในระบบที่มีข้อมูลที่ต้องประมวลผลในเวลาเดียวกันหลาย ๆ ข้อมูล หาก ADC ทำงานได้เร็วพอก็ไม่จำเป็นต้องใช้ ADC หลาย ๆ ตัวทำงานแยกกันสำหรับข้อมูลแต่ละชุด แต่จะใช้วิธีการแบ่งเวลา (Timesharing) โดยใช้วิธี Multiplexing (รูปที่ 4.1) วงจรสุ่มตัวอย่างและคงค่า (Sampling and hold, S/H) จะสุ่ม (Sample) ขนาดของสัญญาณอนาล็อกมาและเก็บ (Hold) ไว้ในช่วงเวลาเพื่อรอให้ ADC รับไปเปลี่ยนให้เป็นสัญญาณดิจิทัลจนเรียบร้อยแล้วค่อยสุ่มสัญญาณใหม่ ทั้งนี้เพื่อที่ไม่จำเป็นต้องใช้ ADC ที่มีความเร็วสูงราคาแพง ข้อมูลดิจิทัลจะถูกส่งต่อไปยัง System bus และถูกประมวลผลโดยตัวประมวลผลที่ได้ของการประมวลผลจะถูกส่งกลับออกมาเพื่อเปลี่ยนกลับเป็นสัญญาณอนาล็อก DAC เพื่อไปควบคุมกิจกรรมทางกายภาพ ของระบบ Analog actuator

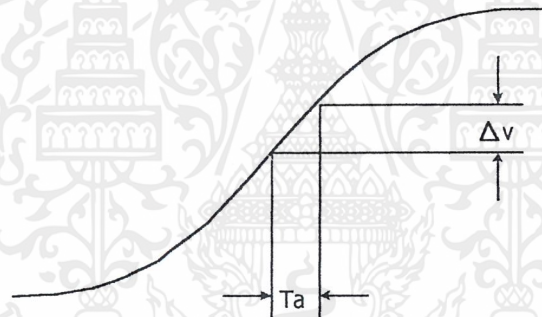
4.2 ทฤษฎีการ Sampling

ในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลนั้น ADC จะต้องใช้เวลาช่วงหนึ่งในการจัดการซึ่งช่วงเวลาดังกล่าวนั้นขึ้นอยู่กับหลาย ๆ ปัจจัย เช่น ความละเอียดของการเปลี่ยนสัญญาณ (จำนวนดิจิทัลบิต) เทคนิคของการเปลี่ยนสัญญาณ และความเร็วในการทำงานของอุปกรณ์ร่วมอื่น ๆ การเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดความเร็วของการแปลงสัญญาณขึ้นอยู่กับการใช้งานเฉพาะอย่างและความแม่นยำที่ต้องการ



รูปที่ 4.1 ระบบควบคุมที่มีการประมวลผลข้อมูลแบบดิจิทัล



รูปที่ 4.2 error จากการวัดใน Aperture time

ช่วงเวลาในการแปลงสัญญาณบางครั้ง อาจเรียกว่า Aperture time ซึ่งความหมายโดยทั่วไป หมายถึงช่วงเวลาที่เกิดความไม่แน่นอนขึ้นในการวัดและประมวลผลก็คือความผิดพลาด (error) ต่อค่าที่วัดได้ในรูปที่ 4.2 สัญญาณอนาลอก $V(t)$ มีอัตราการเปลี่ยนแปลง dv/dt ในช่วง Aperture time, T_a โดยที่ช่วงการเปลี่ยนแปลงอนาลอกจะเท่ากับ Δv นั่นคือ

$$\Delta V = T_a \frac{dV(t)}{dt} \quad (4.1)$$

ดังนั้นหากเวลาที่ ADC ใช้ในการเปลี่ยนสัญญาณในช่วงเวลา T_a นี้สัญญาณดิจิทัลที่ได้อาจจะตรงกับขนาดของสัญญาณอนาลอกค่าใดค่าหนึ่งในช่วงเวลานี้ และจัดทำได้ง่ายและราคาถูกกว่าที่เกิดขึ้นซึ่งแน่นอนเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอนว่าในบางครั้งเป็นไปได้ที่สัญญาณดิจิทัลจะตรงกับขนาดของสัญญาณอนาลอกที่เกิดขึ้นซึ่งเรียกค่าความผิดพลาดที่เกิดขึ้นนี้ว่า Aperture time error ตัวอย่างในกรณีสัญญาณอินพุตเป็นรูปคลื่นชานัน อัตราการเปลี่ยนแปลงของรูปคลื่นจะเกิดสูงสุดตรงบริเวณจุดตัดแกนเวลารอบ ๆ จุดศูนย์โวลต์ (Zero crossing) และ Aperture time error คือ

$$\Delta V = T_a \frac{dV}{dt} (A \sin \omega t)_{t=0} = T_a A \omega \quad (4.2)$$

และ ค่าความผิดพลาดรวม (ε) คิดจากอัตราส่วนของขนาดเต็มสเกล คือ

$$\varepsilon = \frac{\Delta V}{2A} \pi T_a \quad (4.3)$$

ดังนั้น หากต้องการเปลี่ยนสัญญาณเป็นรูปชานันความถี่ 1 กิโลเฮิร์ต ให้เป็นสัญญาณดิจิทัล 10 บิต ซึ่งยอมให้ค่าความผิดพลาดไม่เกินกว่าค่าความละเอียด (Resolution) คือ $1/2^{10}$ LSB หรือ 1.001 ดังนั้นเวลา Aperture time จะต้องอยู่ในช่วง

$$T_a = \frac{\varepsilon}{\pi} = \frac{0.001}{3.14 \times 10^3} = 320 \times 10^{-9} \quad (4.4)$$

จะเห็นว่าแม้สัญญาณ 1 กิโลเฮิร์ต จะไม่ใช่ความถี่สูงก็จริง แต่ ADC ที่ใช้ต้องการเวลาในการเปลี่ยนในเวลา 32 นาโนวินาที ให้เป็นรหัส 10 บิต วิธีอื่นที่ไม่จำเป็นจะต้องใช้ ADC ความเร็วสูงคือการใช้การสุ่มและคงค่าสัญญาณ ซึ่งการสุ่มและคงค่าสัญญาณ ที่มี Aperture time น้อย ๆ นั้นทำได้ง่ายและราคาถูกกว่า

4.3 การสุ่มและคงค่าสัญญาณ (Sample and Hold) และ Aperture error

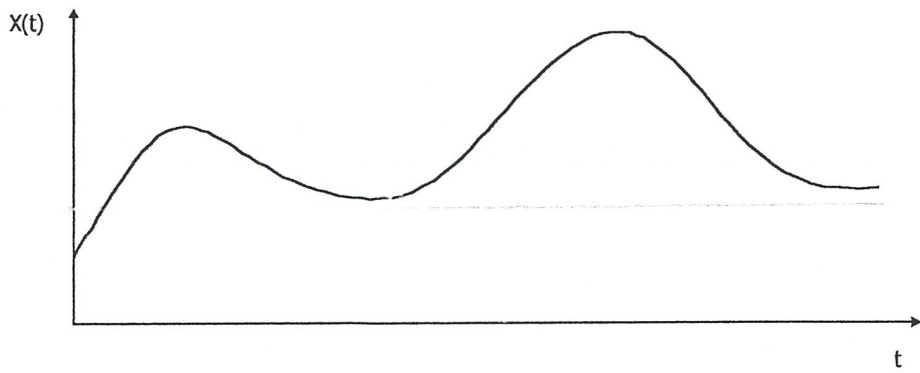
วงจรสุ่มและคงค่าจะทำการสุ่มค่าสัญญาณอินพุต และคงค่าสัญญาณนั้นไว้ในช่วงเวลาหนึ่งได้ ซึ่งส่วนใหญ่จะใช้การประจุแรงดันนั้นไว้ในตัวเก็บประจุที่รั่วไหลต่ำ Aperture time ของการสุ่มและคงค่าสัญญาณ คือเวลาตั้งแต่เริ่มสุ่มสัญญาณจนเก็บประจุค่าแรงดันจนถึงค่าที่สุ่มซึ่งขึ้นอยู่กับแบนด์วิดท์ และช่วงเวลาในการสวิตช์ (Switching time) ในการสุ่มสัญญาณอนาลอกจะถูกสุ่มเป็นระยะ ๆ ซึ่งคงที่ตามรูปที่ 4.3 การสุ่มจะเป็นการตัดต่อสัญญาณอนาลอกในช่วงเวลาอันสั้นด้วยสวิตช์ที่ทำงานด้วยความเร็วสูง ผลของการสุ่มสัญญาณด้วยความเร็วจะเสมือนกับการคูณขบวนสัญญาณพัลส์แคบ ๆ กับสัญญาณอนาลอก ซึ่งจะให้เป็นสัญญาณที่มอดูเลต (Modulate) ระหว่างขบวนพัลส์กับสัญญาณอนาลอกโดยเสมือนว่าสัญญาณอนาลอกจะที่มานับพัลส์ดังแสดงในรูปที่ 4.3(ค) ปัญหาที่ว่าอัตราการสุ่มสัญญาณนั้นควรมีขนาดเท่าใดจึงจะไม่ทำให้ข้อมูลเสียไปเมื่อสัญญาณนั้นถูกเปลี่ยนกลับมาเป็นเช่นเดิม คำตอบคือ ขึ้นอยู่กับความถี่ของสัญญาณ อนาลอกทฤษฎีของการสุ่มกล่าวไว้ว่า “ถ้าสัญญาณต่อเนื่องที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

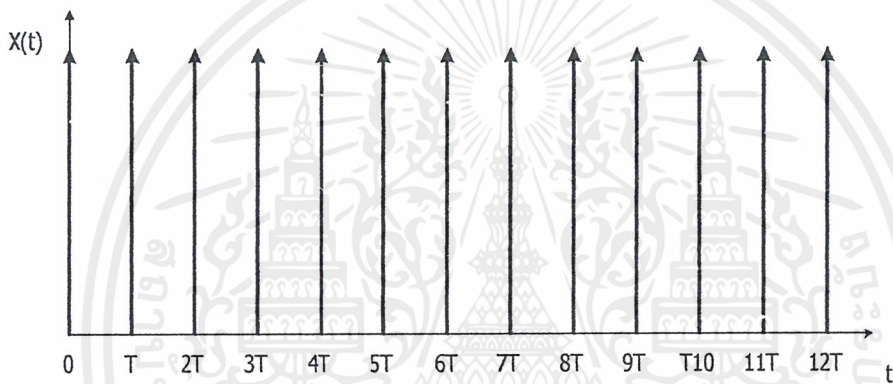
ความถี่และฮาร์โมนิกส์ไม่เกิน f_c ถ้าส่วนคว่ำอัตราการสุ่มไม่น้อยกว่า $2f_c$ แล้วสัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาได้อย่างเดิมโดยไม่สูญเสียรายละเอียดหรือผิดเพี้ยนไป”

4.4 Quantizing Theory

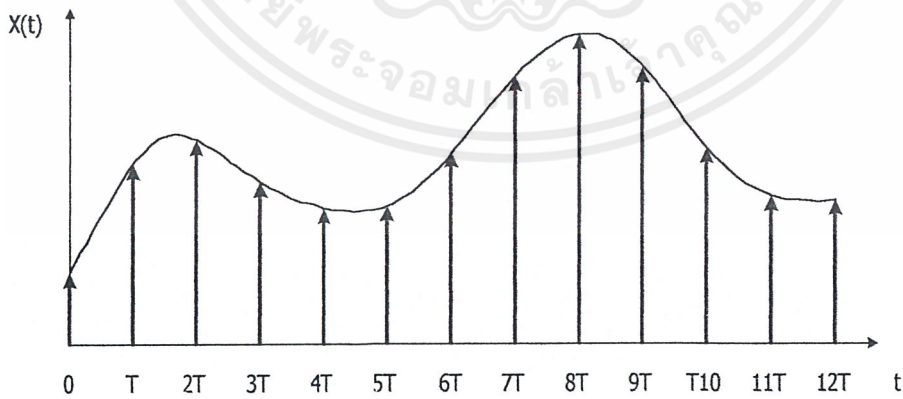
Quantizing Theory เป็นขบวนการที่เปลี่ยนแปลงสัญญาณอนาลอกเป็นสัญญาณที่ไม่ต่อเนื่อง (Discrete signal) หลังการสุ่ม โดยผ่านขบวนการเข้ารหัส (Coding) จัดให้สัญญาณที่ไม่ต่อเนื่องนั้นอยู่ในรูปที่ง่ายต่อการประมวลผลและเป็นสัดส่วนสัมพันธ์กับสัญญาณอนาลอก เช่น ในรูปของรหัสไบนารี (Binary) เป็นต้น หากนำเอาขนาดของสัญญาณอนาลอกและสัญญาณดิจิทัลที่ได้รับการ Quantize มาเขียนกราฟก็จะได้กราฟแสดง Quantize transfer function ดังรูปที่ 4.4 ในรูปกราฟแสดงให้เห็นถึงความสัมพันธ์ระหว่างสัญญาณอนาลอกที่มีขนาดอยู่ระหว่าง 0 ถึง +10 โวลต์ ถูก Quantize และ Encode เป็นรหัสไบนารี (Binary) 3 บิต ได้ 8 ระดับจาก 000 ถึง 111 เนื่องจากในระบบไบนารีรหัสดิจิทัลแต่ละค่าจะแทนขนาดของสัญญาณอนาลอกแต่ละค่าที่เป็นสัดส่วนกับค่าเต็มสเกลโดยค่าสูงสุดของรหัสดิจิทัลคือ ทุกบิตที่เป็น 1 จะเท่ากับสัญญาณอนาลอกเต็มสเกลคูณด้วย $(1-2^{-n})$ โดย n เป็นจำนวนบิตของรหัสดิจิทัลแต่ละบิตที่เป็น 1 จะเท่ากับขนาดเต็มสเกลของสัญญาณอนาลอกคูณกับค่าน้ำหนัก (Weighting) ของรหัสชนิดนั้น หาคด้วย 2^n



(ก)



(ข)



(ค)

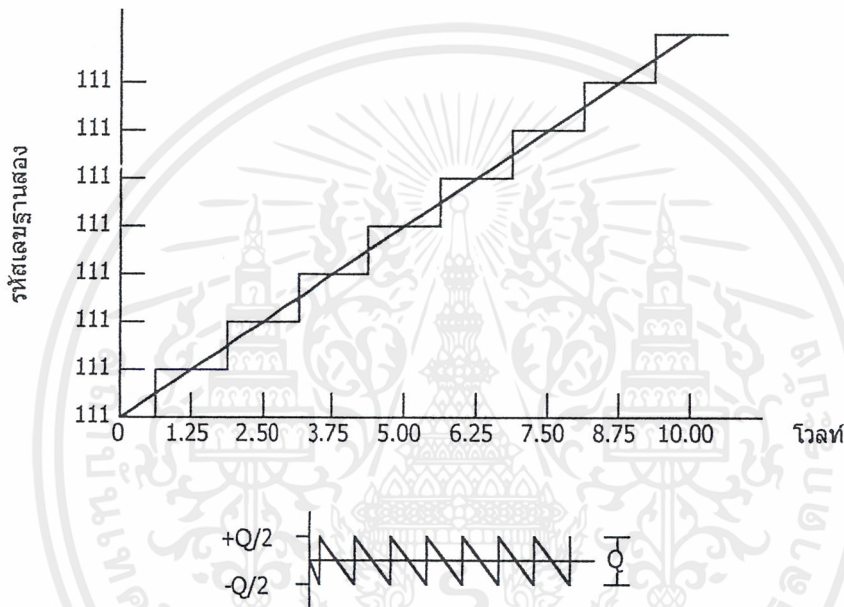
รูปที่ 4.3 การสุ่มสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) สัญญาณอนาล็อกอินพุต

(ข) พัลส์ที่มาสู่สัญญาณ

(ค) สัญญาณอนาล็อกหลังการสุ่ม



รูปที่ 4.4 ทราન્สเฟอร์ฟังก์ชันของ Quantize 3 บิต ตามทฤษฎี

ตัวอย่าง เช่น ค่าเต็มสเกลของสัญญาณอนาล็อกเป็น 10 โวลต์ รหัส 1011 จะแทนขนาดของสัญญาณอนาล็อกอินพุต

$$V_{input} = \frac{R_s}{2^n} \{(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)\} \quad (4.5)$$

จุดสำคัญที่เกี่ยวกับทราન્สเฟอร์ฟังก์ชันในรูปที่ 4.4 อันแรกได้แก่ ความละเอียด (Resolution) ของ Quantizer ซึ่งกำหนดได้จากจำนวนบิตของรหัสดิจิทัล หรือจากกราฟคือขนาดความกว้างของขั้นระดับ (Step) ทางแกนอนาล็อกอินพุตว่าเป็นสัดส่วนเท่าใดระหว่างค่าเต็มสเกลของสัญญาณอนาล็อกกับค่า 2^n จำนวนสถานะเอาต์พุตกำหนดได้จากจำนวนบิตคือ เท่ากับ 2^n สถานะ ตัวอย่างกรณี ADC 8 บิต Quantize จะให้เอาต์พุต 256 สถานะและ 12 บิต ให้ 4096 สถานะต่อค่าเต็มสเกลของอนาล็อกในไดอะแกรมนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกรมทรานเฟอร์ฟังก์ชันจะเห็นจุดแบ่ง ระดับ (Decision point หรือ Theshold level) สัญญาณอนาลอก จะมีจำนวน $2^n - 1$ จุดที่อยู่ที 0.625, 1.875, 3.125, 4.375, 5.625, และ 8.125 โวลต์ ระหว่างจุดดังกล่าว เป็นสัญญาณอนาลอกซึ่งแปลงเป็นสัญญาณดิจิทัล 1 สถานะ ดังนั้น ค่าเหล่านี้จะต้องปรับให้ถูกต้องมากที่สุดเพื่อแปลงขนาดของอนาลอกให้ตรงกับค่าที่ทำการ Quantizer แรงดันที่ 1.25, 2.50, 3.75, 5.0, 6.25, 7.5, 8.75 โวลต์ เป็นจุดกึ่งกลางของรหัสดิจิทัลสถานะสุดท้าย สังเกตว่าในทางทฤษฎีแล้วเส้นตรงนี้จะต้องผ่านจุดกึ่งกลางของรหัสดิจิทัลสถานะสุดท้าย และสังเกตว่าในทางทฤษฎีแล้วเส้นตรงนี้จะต้องผ่านจุดกึ่งกลางของทุกระดับดิจิทัล

4.5 Quantizer Resolution and Error

ในแต่ละสถานะของสัญญาณดิจิทัลเอาต์พุตจะแทนขนาดของสัญญาณอนาลอกค่าใดค่าหนึ่ง ในช่วงเล็ก ๆ ระหว่างจุดแบ่งระดับ เรียกช่วงเล็ก ๆ นี้ว่าเป็นขนาดหนึ่ง Analog quantization หรือหนึ่งควันตัม คือ 1.25 โวลต์ คำนีได้จากการคำนวณจาก

$$Q = \frac{FSR}{2^n} \quad (4.6)$$

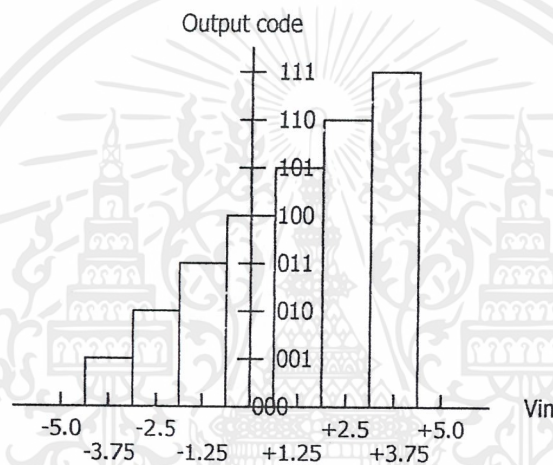
โดยที่ FSR คือ ช่วงเต็มสเกลของแรงดันอนาลอก (Full Scal Range) และ n คือ จำนวนบิตของรหัสดิจิทัลจากสมการจะเห็นว่า หากขนาดของจำนวนบิตมากขนาดของควันตัมก็จะลดลงและถ้าให้สัญญาณอินพุตของ Quantizer กว้างไปตลอดช่วงของสัญญาณอนาลอกก็จะเห็นว่าช่วงผลต่างของสัญญาณอนาลอกอินพุตและดิจิทัลเอาต์พุตเป็นช่วงพล็อตได้เป็นรูปฟันเลื่อยดังรูปที่ 4.4 ด้านล่าง ซึ่งเรียกว่า Quantizing error ซึ่งค่าของความผิดพลาดแบบนี้ก็คือ 1 ช่วงสัญญาณอนาลอกแปลงเป็นรหัสดิจิทัล 1 สถานะ ดังกล่าวมาแล้วนั่นเอง ค่าความผิดพลาดนี้เป็นธรรมชาติของ Quantizing ซึ่งทำการแก้ไขไม่ได้นอกจากการเพิ่ม จำนวนบิตของ Quantizer ให้มากขึ้น และ เอาต์พุตค่าความผิดพลาดจะอยู่ระหว่าง $0-Q/2$ Error อาจจะเป็นที่ศูนย์ เมื่อสัญญาณอนาลอกค่าที่จุดกึ่งกลางของควันตัมพอดี ลักษณะฟังก์ชันของค่าความผิดพลาดจะสามารถพิจารณาจากสัญญาณรบกวนทางอินพุต ซึ่งมีค่าเป็น 0 โวลต์ และค่าเฉลี่ยเป็นศูนย์ ค่าอาร์เอ็มเอส (root mean square, rms) เป็น $Q/2 \sqrt{3}$ ซึ่งจะเป็นการวิเคราะห์รูปคลื่นฟันเลื่อย

4.6 รหัสตัวเลขสำหรับการเปลี่ยนข้อมูล

รหัสตัวเลขที่นิยมนำมาใช้ในการเปลี่ยนข้อมูลได้แก่รหัสไบนารีหรือที่เรียกว่า Straight binary โดยที่รหัสไบนารีสถานะสูงสุดจะแทนสัญญาณอนาลอก FSR $(1 - 2^{-n})$ โวลต์ ตัวอย่างเช่นหากสัญญาณอนาลอกเต็มสเกล (FSR) เท่ากับ 20 โวลต์ สำหรับ ADC ขนาด 12 บิต รหัส 1111 1111 1111 จะแทนสัญญาณอนาลอกขนาด $20(1 - 2^{-12})$ หรือ 19.39951171 โวลต์ นอกจากรหัสไบนารีธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

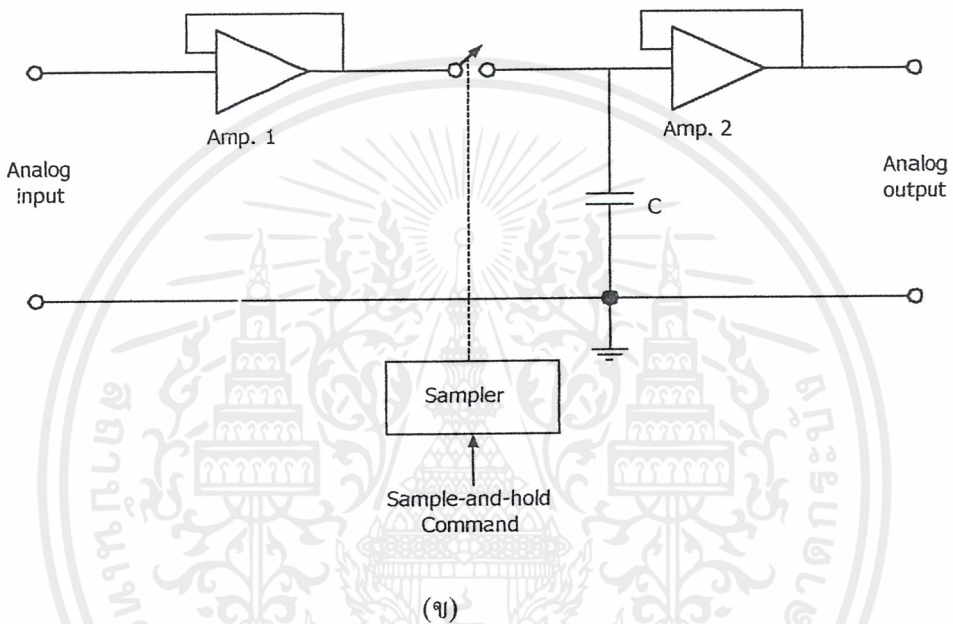
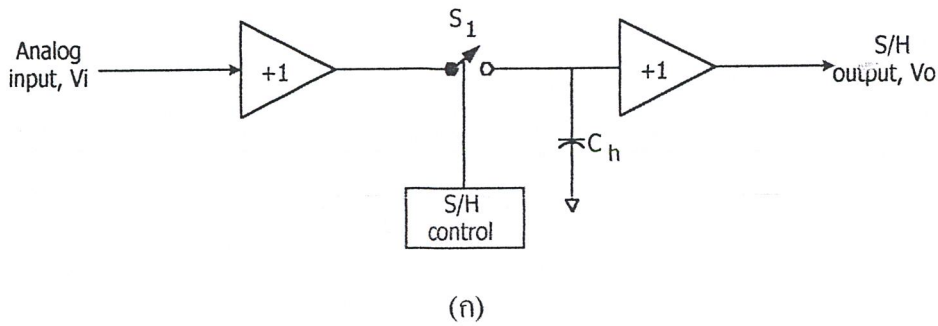
กล่าวยังมีการใช้ระบบการแปลงสัญญาณ ได้แก่ ออฟเซตไบนารี, Two 's complement, BCD ซึ่งแต่ละชนิดมีข้อดีและความเหมาะสมต่างกัน ตัวอย่างเช่น ระบบ BCD เหมาะสำหรับการแสดงเป็นตัวเลขหน้าปัทม์หรือต่อเข้ากับดิจิตอลมิเตอร์ รหัส Two 's complement เหมาะสำหรับการคำนวณทางคณิตศาสตร์ลอจิกและนสำหรับระบบออฟเซตไบนารีนั้นเหมาะสำหรับการแปลงสัญญาณอินพุตที่ทั้งช่วงบวกและลบ ในรูปที่ 3.5 แสดงทรานส์เฟอร์ฟังก์ชันของ ADC 3 บิตที่ใช้รหัสออฟเซตไบนารี นอกจากการใช้มาตรฐานของการใช้รหัสตัวเลขแล้วยังมีมาตรฐานของการเลือกช่วงของขนาดแรงดันอินพุตสำหรับ ADC คือ หากเป็นสัญญาณช่วงบวกหรือลบอย่างเดียวจะใช้ 0-5 โวลต์ หรือ 0-10 โวลต์ แต่ถ้าเป็นช่วงลบจะใช้ -2.5 โวลต์ - 5 โวลต์ และ -10 โวลต์ เป็นมาตรฐาน



รูปที่ 4.5 ทรานส์เฟอร์ฟังก์ชันของ ADC 3 บิต ที่ใช้รหัสออฟเซตไบนารี

4.7 วงจรสุ่มและคงค่าสัญญาณ (Sample and Hold, S&H)

ที่ผ่านมาได้กล่าวถึงจุดมุ่งหมายในการใช้วงจร S&H กับ ADC และต่อจากนี้จะได้กล่าวถึงรายละเอียดของวงจร S&H บางแบบที่ใช้ในปัจจุบันความจริงแล้ววงจร S&H มิได้มีใช้เฉพาะกับ ADC เท่านั้น แต่ก็ยังใช้กันทั่วไปในระบบ Data distribution, Sampling scope, DVM, Reconstruction filter และอนาลอกคอมพิวเตอรืเป็นต้น วงจร S&H โดยพื้นฐานแล้วเป็นอุปกรณ์หรือวงจรเก็บแรงดัน (Voltage memory) ซึ่งใช้อุปกรณ์ร่วมสำคัญคือตัวเก็บประจุ ในรูปที่ 4.6 (ก) แสดงวงจรพื้นฐาน S&H อิเล็กทรอนิกส์ตัวชี้และเวลาในการประจุแรงดันถึงค่าที่ Sample มานั้นเรียกว่า Aperture-time ของ S&H จาก ลักษณะอนาลอกอินพุตสัญญาณการสุ่มและเอาต์พุต



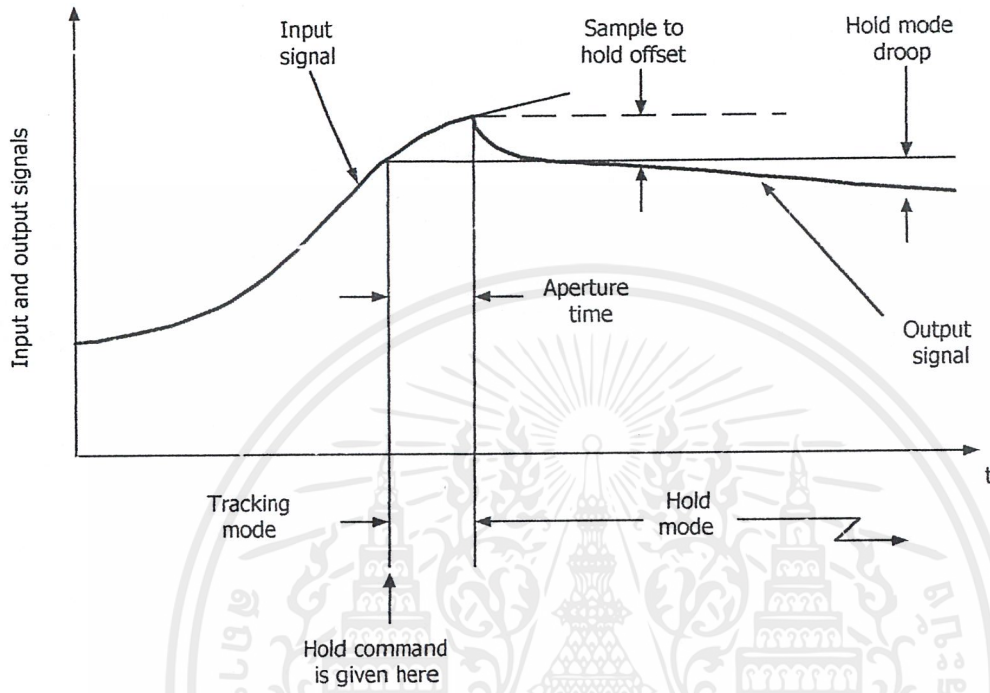
รูปที่ 4.6 (ก) พื้นฐานของ S&H

(ข) ไลอะแกรมของวงจร S&H

รูปที่ 4.6 (ข) แสดงวงจร ที่ใกล้เคียงกับวงจรในทางปฏิบัติโดยเพิ่มเติมบัฟเฟอร์แอมป์ไฟร์เข้าทางส่วนอินพุตและเอาต์พุตของ S&H พื้นฐาน แอมป์ไฟร์ทางด้านอินพุตช่วยทำให้วงจรมีอินพุตอิมพีแดนสูง สะดวกต่อการใช้งาน และสามารถเพิ่มกระแสเพื่อทำให้ประจุ C_h ได้เร็วขึ้น ส่วนทางเอาต์พุตช่วยทำให้เอาต์พุตอิมพีแดนซ์สามารถขับ ADC ได้ง่าย มีจุดสำคัญที่ต้องพิจารณาคือ ในส่วนของแอมป์ไฟร์เหล่านี้ปกติแล้ว ถ้าเป็นจะต้องเป็นแอมป์ไฟร์ที่ใช้กระแสอินพุตต่ำ ทั้งนี้เพื่อคั้งกระแสจากตัวเก็บประจุในช่วงที่คั้งค่าสัญญาณให้น้อยที่สุด มิฉะนั้นแรงดันที่คั้งค่าไว้จะลดระดับเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการโหด (drop) ดังแสดงในรูปที่ 4.7 ซึ่งปกติแล้วจะใช้แอมพลิไฟเออร์ที่มี FET หรือ MOSFET เป็นอินพุต เพราะการไบอัสด้วยแรงดันทำให้กระแสต่ำด้วย



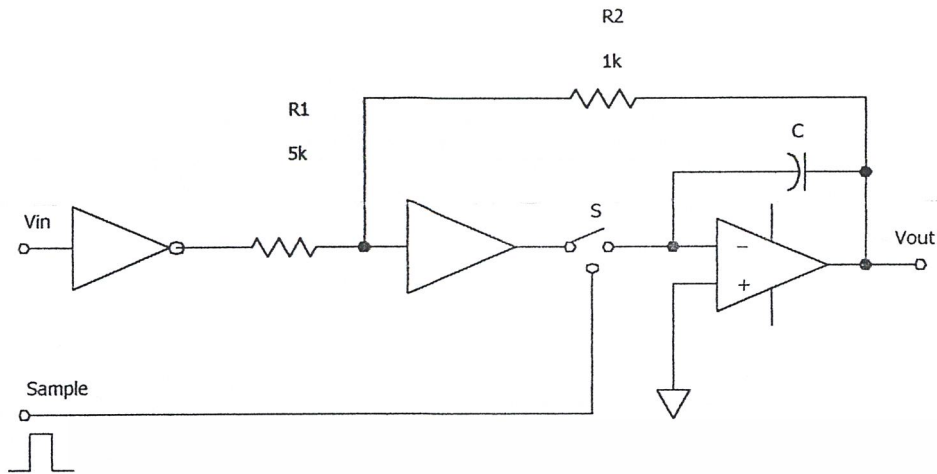
รูปที่ 4.7 รูปคลื่นเอาต์พุตของ S&H

วงจร S&H ในระบบ Data acquisition นิยมใช้สองแบบคือ Sample-hold และ Track-hold วงจร S&H จะใช้วิธีการสุ่มสัญญาณอย่างรวดเร็วแล้วเข้าสู่ Holding period ซึ่งหมายความว่าสวิทช์ควบคุมจะต้องต้นในช่วงเวลาอันสั้นอย่างต่อเนื่อง ส่วน Track-holds จะตัวสวิทช์สุ่มช้ากว่าการจับวงจร S&H มีได้หลายลักษณะ ซึ่งอาจนำไอซี (IC) หรือทรานซิสเตอร์มาประกอบเป็นวงจรตลอดจนการสร้างวงจรทั้งหมดของ S&H ลงชิปไอซีเดี่ยว เช่น เบอร์ LF398

4.8 วงจรสุ่มแบบ Inverting Close Loop

ในวงจรนี้ C จะประจุด้วยอัตรา RC ซึ่งสามารถเพิ่มความเร็วได้โดยใช้ Current boost amplifier อยู่ในลูปฟีดแบ็คดังรูปที่ 4.8 โดยแอมพลิไฟเออร์นี้มีอัตราขยายเท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



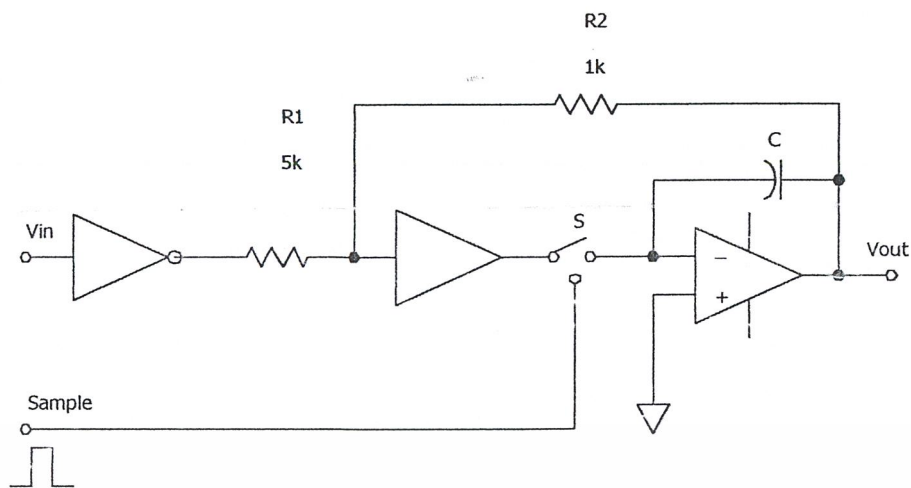
รูปที่ 4.8 วงจรลุ่มสัญญาณแบบ Inverting Close Loop

ข้อดีของวงจรมีคือทำงานได้รวดเร็วและแม่นยำความเร็วในการประจุขึ้นอยู่กับความเร็วของ A1 และความสามารถ ในการจ่ายกระแสของมัน ไดโอดสองตัวจะทำหน้าที่ Clamp สัญญาณเอาต์พุตไปที่ อินพุตอินเวอร์ตติ้งของ A1 เพื่อยังคงให้วงจรมีเสถียรภาพดี เมื่อสวิตช์ (s) เปิด วงจรลักษณะนี้เป็นวงจร พื้นฐานของ ไอซีเบอร์ LF398

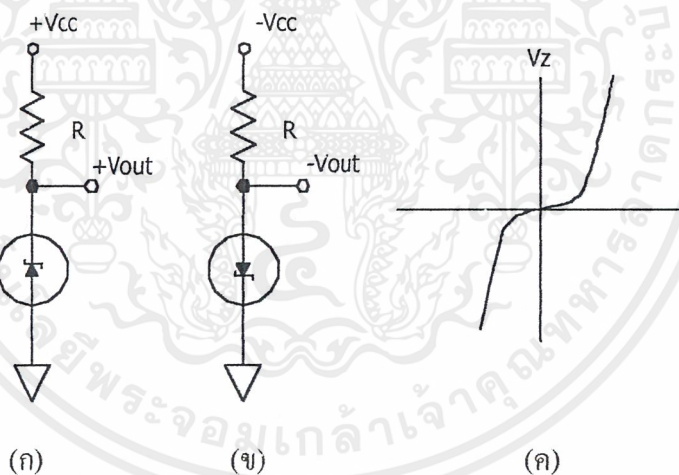
4.9 วงจรแรงดันอ้างอิง (Voltage Reference Circuit)

วงจรแรงดันอ้างอิงนั้นเป็นวงจรที่มีความสำคัญวงจรหนึ่งในระบบ Data Acquisition เนื่องจาก เป็นส่วนสำคัญในการกำหนดคุณภาพของวงจร DAC หรือ ADC วงจรแรงดันอ้างอิงมีทั้งเป็นวงจรอิสระ หรือเป็นวงจรรวมอยู่ในวงจร ADC หรือ DAC

1) Basic voltage reference อุปกรณ์ที่นิยมให้เป็นแหล่งกำเนิดแรงดันอ้างอิงได้แก่ ซินเนอร์ไดโอด ซึ่งเมื่อให้รีเวิร์สไบอัสจนเกิดการเบรคดาวน์แรงดันคร่อมซินเนอร์จะคงที่เท่ากับแรงดันเบรคดาวน์ (V_z) ตัวต้านทาน R (รูปที่ 4.10) ที่ต่ออนุกรมกับซินเนอร์ จะทำหน้าที่กำหนดกระแสไบอัสแก่ซินเนอร์ ให้เบรคดาวน์และจำกัดกระแสรีเวิร์สไม่ให้ไหลมากเกินไปจนเป็นอันตรายแก่ซินเนอร์ ข้อเสียของวงจรมีคือ แรงดันมักเปลี่ยนแปลงตามอุณหภูมิได้ง่ายหรือเรียกว่ามีสัมประสิทธิ์ทางอุณหภูมิสูง และจ่ายกระแสได้จำนวนจำกัด รวมทั้งแรงดันเอาต์พุตจะแปรตามแรงดันอินพุตจึงมักใช้วงจรมีกับ ADC ที่ไม่ต้องการคุณภาพมากนัก



รูปที่ 4.9 วงจรลูปแบบ Non-inverting closed loop

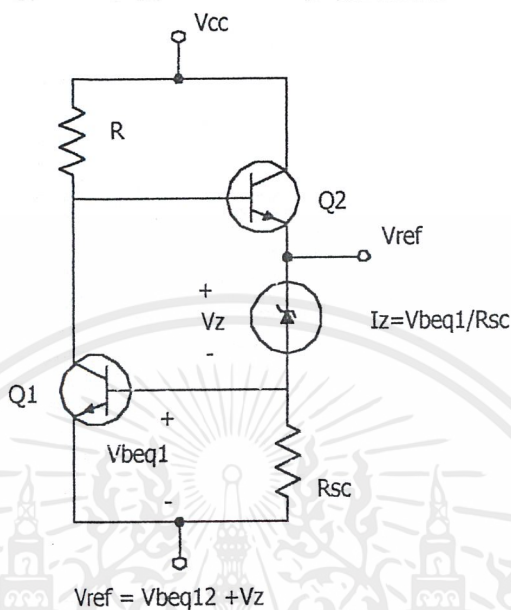


รูปที่ 4.10 วงจรแรงดันอ้างอิงพื้นฐานและกราฟคุณสมบัติ

- (ก) แรงดันอ้างอิงบวก
- (ข) แรงดันอ้างอิงลบ
- (ค) กราฟคุณสมบัติของซีเนอร์

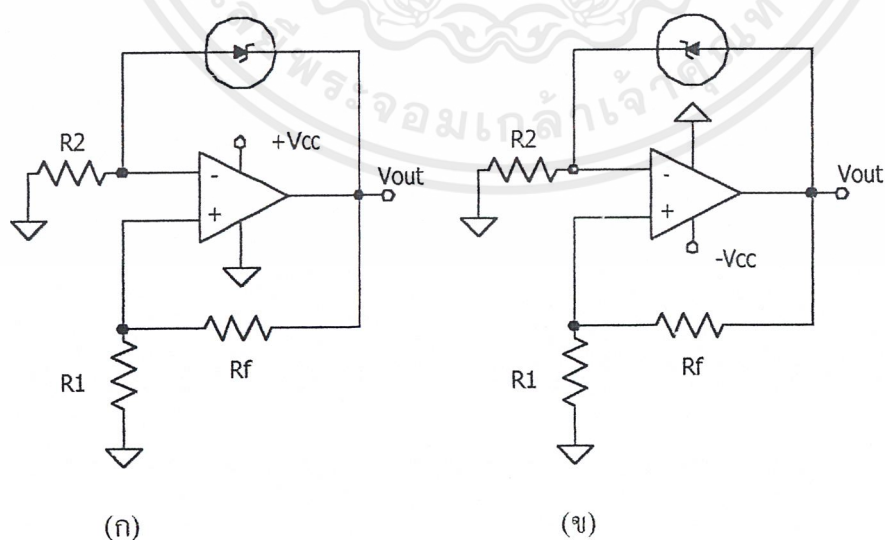
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสียของวงจรอ้างอิงแบบพื้นฐานดังกล่าวสามารถแก้ไขโดยวงจรดังรูปที่ 4.11 ทรานซิสเตอร์ Q_1 และ Q_2 จะทำหน้าที่เป็นแหล่งจ่ายกระแสไบอัสคงที่ให้แก่อิเล็กตรอนไดโอด ลักษณะดังกล่าวทำให้แรงดันเอาต์พุต (V_{cc}) รวมทั้งผลทางด้านอุณหภูมิของวงจรได้รับการปรับปรุงให้ดีขึ้นด้วย



รูปที่ 4.11 วงจรแรงดันอ้างอิงใช้ซีเนอร์ไดโอดที่ปรับปรุงคุณสมบัติจากวงจรแรงดันอ้างอิงพื้นฐาน

2) Precision Voltage Reference แรงดันอ้างอิงที่คุณภาพดีกว่าจะใช้โอปแอมป์ร่วมกับซีเนอร์ ซึ่งนอกจากจะได้แรงดันที่คงที่มากกว่าแล้ว ยังสามารถปรับแรงดันเอาต์พุตให้ได้มากหรือน้อยกว่าแรงดันซีเนอร์ไดโอดด้วย ลักษณะการจัดรูปวงจรต่าง ๆ แสดงได้ดังรูปที่ 4.12



รูปที่ 4.12 แรงดันอ้างอิง (a) แรงดันอ้างอิงบวก (b) แรงดันอ้างอิงลบ

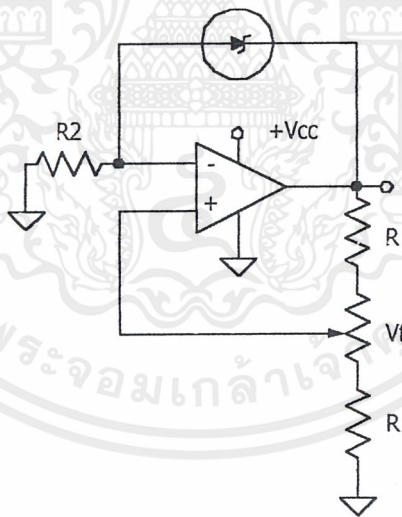
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามปกติแล้วแรงดันอ้างอิงที่ใช้ซีเนอร์ไดโอดจะให้คุณภาพดีก็ต่อเมื่อกระแสที่จ่ายให้ซีเนอร์คงที่ตลอดเวลาและทุกช่วงของอุณหภูมิในวงจรรูปที่ 3.12 ออปแอมป์จะทำหน้าที่จ่ายกระแสที่และมีสัมประสิทธิ์อุณหภูมิต่ำกว่ากระแสที่ผ่านซีเนอร์ขึ้นอยู่กับค่า R_1 , R_2 , R_f และค่า V_o กำหนดจาก R_1 , R_f และ V_o การออกแบบต้องเลือกซีเนอร์ซึ่งรู้ค่า I_z และ V_z ทำการเลือกค่า R_1 และ R_f จากสมการ V_{out}

$$V_o = \frac{V_R(R_1 + R_f)}{R_f} \quad (4.7)$$

$$I_z = \frac{V_o R_1}{R_2(R_1 + R_f)} \cong \frac{V_o R_1}{R_f} \quad (4.8)$$

$$V_1 = \frac{V_o R_1}{(R_1 + R_f)} = I_z R_2 \quad (4.9)$$



รูปที่ 4.13 แรงดันอ้างอิงปรับค่าได้

3) Bandgap Voltage Reference ได้รับการออกแบบเพื่อแก้ไขสัมประสิทธิ์ทางอุณหภูมิ โดยใช้ผลต่างของแรงดันเบส-อิมิตเตอร์ของทรานซิสเตอร์ที่ทำงานที่กระแสต่างกันโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_{Ref} = V_{BEQ3} + I_2 R_2$$

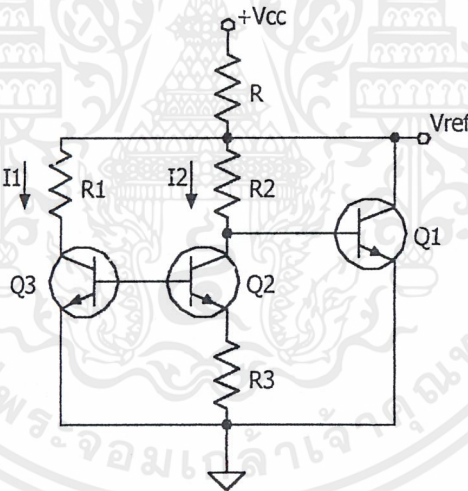
$$V_{Ref} = V_{BEQ3} + \frac{kT_j}{q \left(\ln \frac{I_1}{I_2} \right)} \quad (4.10)$$

เมื่อ k = Boltmann's constant (1.3805×10^{-23} J/K)

T_j = Absolute temperature ของรอยต่อ

q = Electron charge (1.602×10^{-19} C)

แรงดันอ้างอิงแบบแบนด์แกปได้ถูกสร้างโดยใช้วงจรพื้นฐานในรูปที่ 3.14 และมีจำหน่ายตัวดังคล้ายทรานซิสเตอร์ เช่น เบอร์ LM336 สามารถปรับขนาดของ V_{out} ได้



รูปที่ 4.14 วงจร Band gap voltage reference

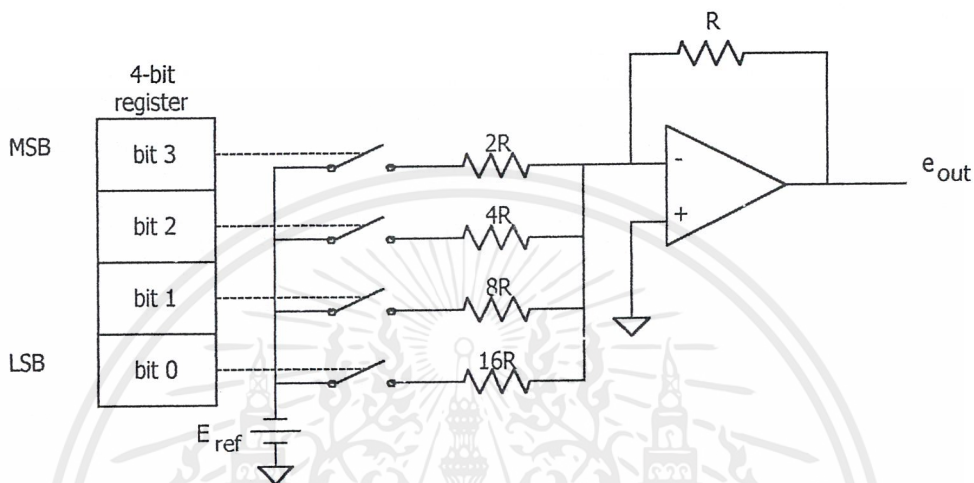
4.10 วงจร Digital to Analogue Converter (DAC)

DAC นับเป็นอุปกรณ์สำคัญที่ทำให้ดิจิทัลคอมพิเตอร์สามารถเชื่อมโยงกับอุปกรณ์หรือวงจรรนาลอกอื่น ๆ ตัวอย่างการใช้งาน DAC คือระบบแสดงผลบนจอภาพ ระบบสังเคราะห์เสียง เป็นต้น และที่สำคัญ DAC ยังเป็นส่วนประกอบที่สำคัญใน ADC ที่ใช้กันอยู่ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10.1 DAC แบบ Binary weight ladder

การจัดวงจร Binary weight ladder มีลักษณะตามรูปที่ 3.15 สวิตช์ $S_1 - S_4$ จะถูกควบคุม เปิด / ปิด ด้วยรหัสดิจิทัล เพื่อตัด/ต่อแรงดันอ้างอิงเข้ากับวงจรรีซิสเตอร์ที่มีค่า $R, 2R, 4R, \dots, (2n)R$ ตัวอย่างเช่นในกรณี DAC แบบ 4 บิต ใช้รีซิสเตอร์เป็น $10k\Omega, 20k\Omega, 30k\Omega, 40k\Omega$ และ $80k\Omega$ เป็นต้น



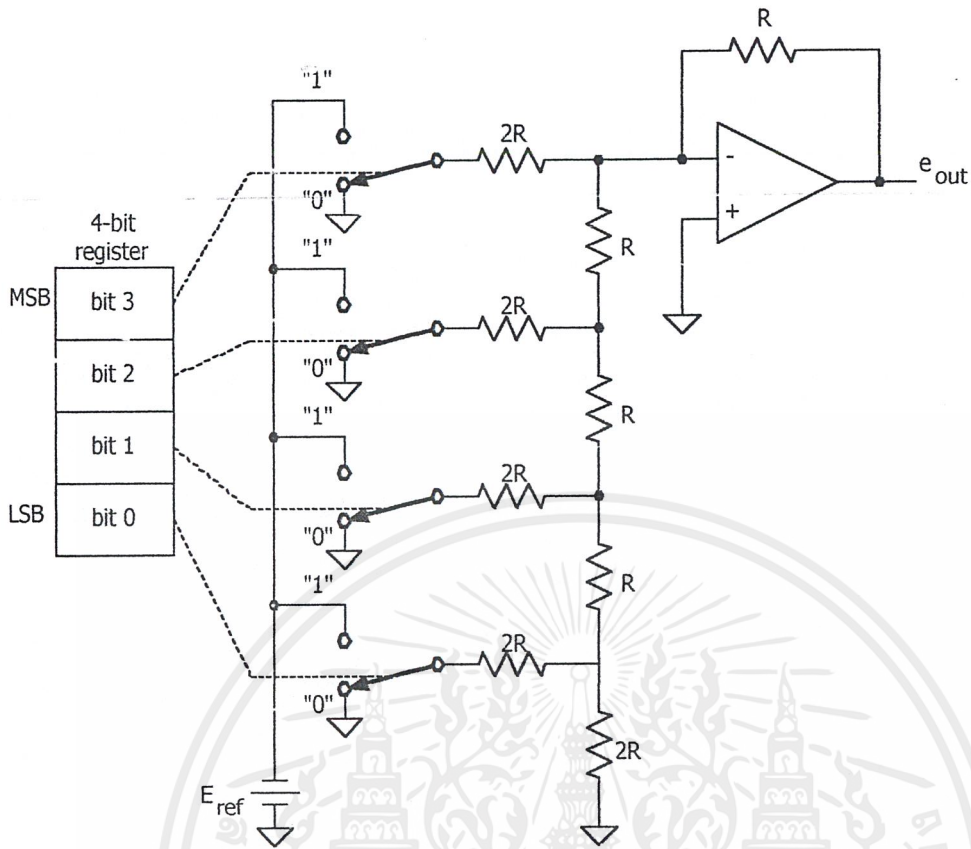
รูปที่ 4.15 DAC แบบ Binary weight ladder

ค่ารีซิสเตอร์ที่ weight ค่าตามรหัสดิจิทัลที่เพิ่มขึ้นจะทำให้กระแสผ่านรีซิสเตอร์เข้าไปรวมกันก่อนเข้าออปแอมป์ลดลงด้วยแฟกเตอร์ 2 ตามค่า R ที่เพิ่มขึ้น เช่นหากแรงดันอ้างอิงเป็น 10 โวลต์ ในกระแสที่ผ่านตัวต้านทานจะเป็น 1.0, 0.5, 0.25, และ 0.125 mA ตามลำดับ ออปแอมป์ที่เอาต์พุตจะทำหน้าที่เปลี่ยนกระแสรวมให้เป็นแรงดันที่เอาต์พุต

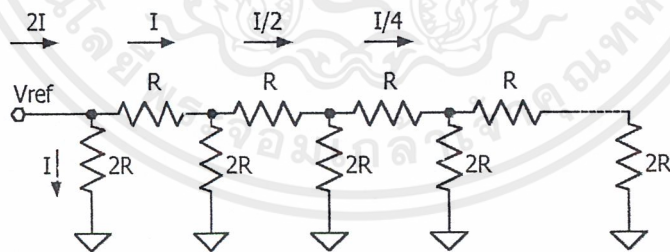
4.10.2 DAC แบบ R-2R ladder

ถึงแม้ว่า DAC แบบ Binary weight จะใช้ค่ารีซิสเตอร์เพียง 4 ค่า ก็ตาม แต่ในการผลิต DAC แบบนี้นับเป็นข้อเสียก็ยังคงเป็นปัญหายุ่งยากในการผลิตอยู่ดี รูปแบบที่ดีกว่าคือการจัดวงจรแบบ R-2R ดังรูปที่ 4.16 ในวงจรนี้จะเห็นว่าสวิตช์จะตัดต่อให้แรงดันอ้างอิงต่อกับวงจร ladder หรือต่อ ladder ลงกราวด์ที่ค่า $2R$ จะเห็นได้ว่า switch input resistor ($2R$) มองเข้าไปจะเห็นคู่ของรีซิสเตอร์ระหว่างจุดต่อ R-2R ที่ติดกัน กระแสจะถูกบั่นทอนไปในอัตรา 2:1 ซึ่งสอดคล้องกับรหัสไบนารี ดังแสดงในรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 วงจร DAC แบบ R-2R ขนาด 4 บิต

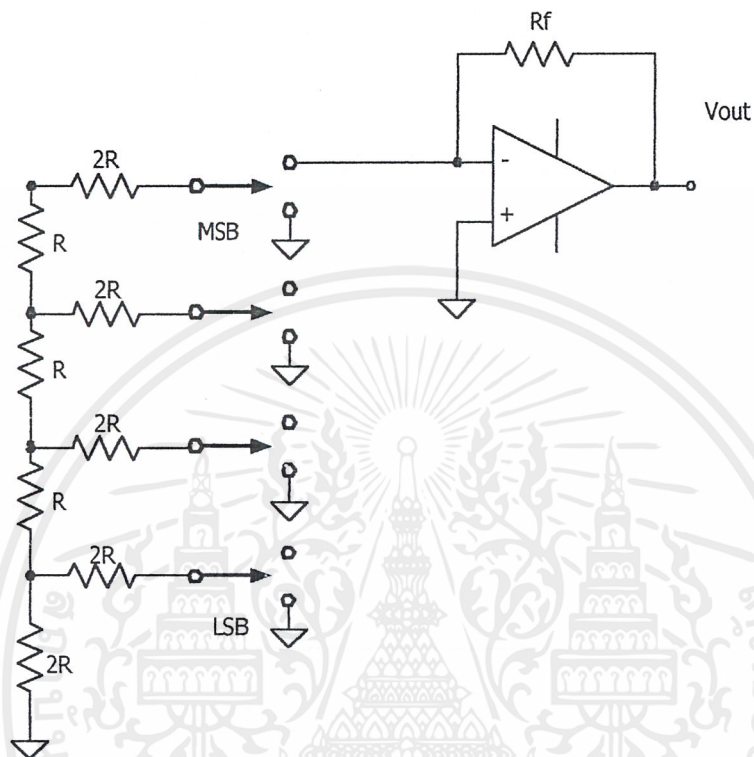


รูปที่ 4.17 วงจรรีซิสทีฟแลดเดอร์ (Resistive Ladder)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.10.3 Inverted R/2R ladder DAC

ลักษณะการจัดวงจรแบบนี้จะคล้ายกับ R-2R ladder เพียงแต่สวิทช์จะตัดต่อขา R กับกราวด์และอินพุตของ Summing amplifier แทนที่จะเป็น V_{ref} วิธีนี้นิยมใช้ในการทำ DAC ในวงจรรวมเพราะสวิทช์จะตัดต่อที่แรงดันตกคร่อมต่ำกว่าซึ่งสร้างได้ง่ายกว่า



รูปที่ 4.18 วงจร Invert R-2R ladder DAC

4.11 Analog to Digital Converter

ลักษณะของการจัดวงจรแบบ DAC มีหลายแบบ แต่ที่นิยมใช้มีเพียงไม่กี่แบบและส่วนใหญ่จะอยู่ในรูปของวงจรรวม

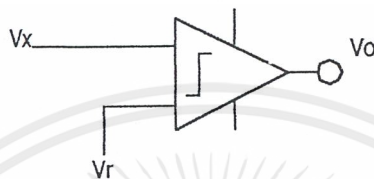
4.11.1 Basic conversion method

วิธีการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลแบบง่าย ๆ แสดงในรูปที่ 4.19 แรงดันอินพุตที่ไม่ทราบค่า V_x จะต่อเข้ากับอินพุตขาหนึ่งของอนาลอกคอมพาราเตอร์ และแรงดันอ้างอิงที่ขนาดแปรตามเวลา V_r ต่อเข้ากับขาอินพุตขาหนึ่งของ คอมพาราเตอร์ ลักษณะของทรานเฟอร์ฟังก์ชันของคอมพาราเตอร์แสดงในรูปที่ 4.20 ถ้าแรงดันอินพุต V_1 มากกว่าแรงดันอินพุต V_2 แล้วแรงดันอินพุตจะเป็นลอจิก 1 ถ้าอินพุต V_1 น้อยกว่า V_2 แล้วเอาต์พุตจะเป็น 0 วิธีการแปลงข้อมูลคือ แรงดันอ้างอิงจะถูกแปรค่าจนกระทั่งรู้ค่าแรงดันอินพุตที่ผิดพลาดไม่เกิน Quantization error ของคอนเวอร์เตอร์ ในแนวความคิด

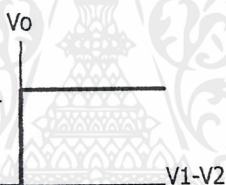
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว ตรรกของ ADC คือพยายามเลือกกลุ่มของสัมประสิทธิ์ ไบนารี a_i เพื่อให้ผลต่างระหว่างแรงดันอินพุต V_x และค่าที่ Quantize ได้ครั้งสุดท้ายน้อยกว่า 0.5 LSB ซึ่งเขียนเป็นสมการได้

$$\left[V_x - V_{FSR} \sum_{j=1}^n a_j 2^j \right] < 0.5 \text{ LSB} \quad (4.11)$$



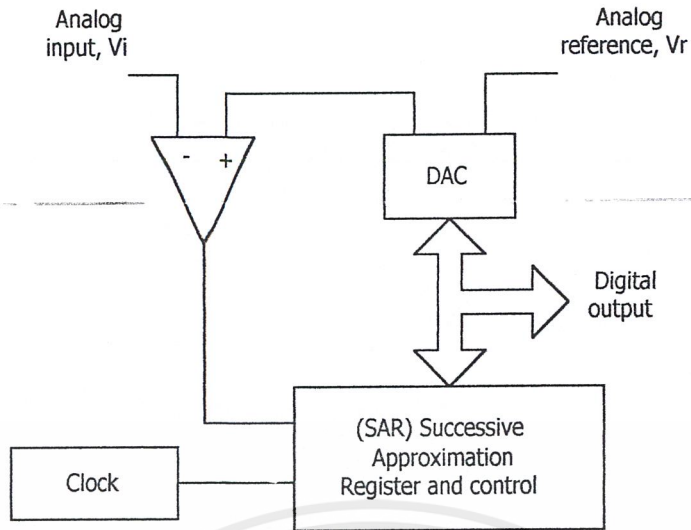
รูปที่ 4.19 วิธีการพื้นฐานของ ADC



รูปที่ 4.20 Transfer function ของ Comparator

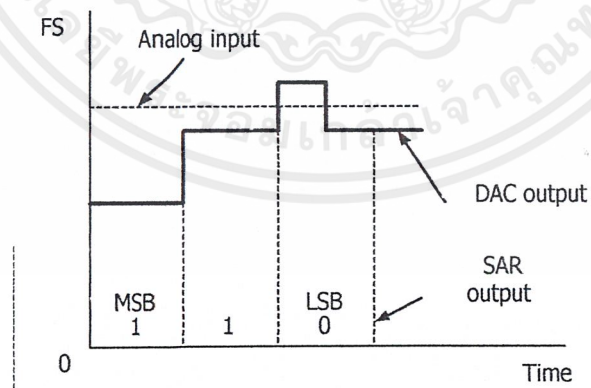
4.11.2 Successive Approximation ADC

วงจร ADC ชนิดนี้ได้รับความนิยมในงานประยุกต์ที่ใช้ความเร็วปานกลางและค่อนข้างสูง การจัดวงจรคล้ายกันกับแบบเคาท์เตอร์ที่ทำงานในลักษณะของการป้อนกลับ ซึ่งบล็อกไดอะแกรมในรูปที่ 4.21 แสดงฟังก์ชันต่าง ๆ ใน ADC ชนิดนี้ คอมพาราเตอร์จะคอยเปรียบเทียบเอาต์พุตจาก ADC กับอนาล็อกอินพุต V_{in} เอาต์พุตจะไปควบคุม SAR (Successive Approximation Register) ซึ่งเป็นไอซี MSI (Medium Scale Integrated circuit) ที่ได้รับการออกแบบพิเศษเพื่อทำหน้าที่นี้โดยเฉพาะ



รูปที่ 4.21 บล็อกไดอะแกรมของ Successive approximation ADC

ในรูปที่ 4.22 แสดงไทมิ่งไดอะแกรมของ ADC ที่มีระดับอนาล็อก 1 และ 2 ที่ระดับ 1 clock เข้าไป 1 ลูกจะทำให้ MSB (Most Significant Bit) (4 บิต) เป็น 1 ส่วนบิตอื่นยังคงเป็น 0 DAC จะเปลี่ยนเอาต์พุตของ SAR เป็นอนาล็อกเปรียบเทียบกับสัญญาณอนาล็อกอินพุต ถ้าผลการเปรียบเทียบที่คอมพาราเตอร์บอกว่ น้อยกว่าอินพุตก็ให้คงค่าบิตนั้นเป็น 1 ไว้ แต่ถ้ามากกว่าจะให้บิตนั้นเป็น 0 จากนั้นทำการทดสอบบิตถัดไปโดยทำให้เป็น 1 หากผลรวมของสองบิตหรือบิตหลังมากกว่าก็จะทำให้บิตนั้นเป็น 0 แต่ถ้าน้อยกว่าให้คงค่า 1 ไว้ แล้วทดสอบบิตถัดไปตามกรรมวิธีดังกล่าวจนครบทุกบิตหรือจนกว่าเอาต์พุตจะต่างจาก V_{in} ไม่เกิน 1 LSB ในตัวอย่างแสดงการทำงานเมื่อ V_{in} ลดต่ำลงมาระดับหนึ่งด้วยเช่นกัน



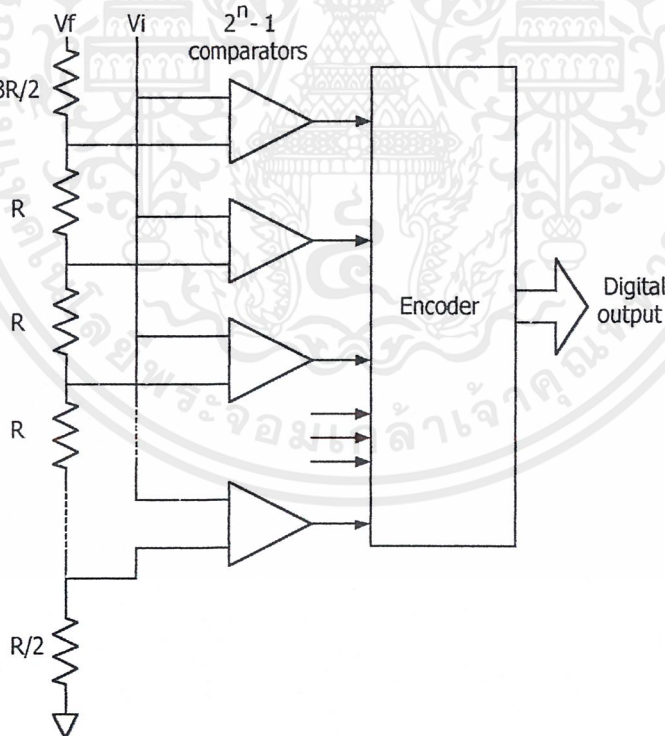
รูปที่ 4.22 สัญญาณเวลาของ SAR

มีข้อจำกัดประการหนึ่งสำหรับการแปลงสัญญาณ คือสัญญาณอนาล็อกอินพุต จะต้องคงที่ในช่วงเวลาที่ทำการเปลี่ยนแปลงสัญญาณได้ไม่เกิน $\frac{1}{2}$ LSB ในช่วงสุดท้ายของการเปลี่ยนสัญญาณดิจิทัล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุตจะออกมาขนานกับทุกบิต แต่บางแบบจะให้เอาต์พุตออกมาในลักษณะอนุกรม วงจร DAC แบบนี้สามารถทำงานได้สองโหมด คือ โหมดที่ทำงานโดยอิสระ (Free run) และโหมดครอคำสั่ง (Start conversion) จากภายนอก เวลาที่ใช้ในการเปลี่ยนสัญญาณใช้ $(n + 1)$ ลูกของสัญญาณนาฬิกา โดยพัลส์ลูกแรกจะใช้ในการรีเซ็ตรีจิสเตอร์ภายใน สุดท้ายคุณภาพของระบบจะขึ้นอยู่กับคุณภาพของ DAC ในระบบเป็นอย่างยิ่ง

4.11.3 Parallel (Flash) ADC

สำหรับการแปลงสัญญาณที่มีความเร็วสูงมาก ๆ เช่นการแปลงสัญญาณภาพโทรทัศน์เรดาห์ จำเป็นต้องใช้ ADC แบบพิเศษที่เรียกว่า Parallel ADC ซึ่งแสดงบล็อกไดอะแกรมดังรูปที่ 4.23 หลักการทำงานคือ จะใช้คอมพาราเตอร์เปรียบเทียบสัญญาณอนาล็อกอินพุตกับแรงดันอ้างอิงที่แบ่งแรงดันให้สอดคล้องกับรหัสดิจิทัล โดยใช้ตัวต้านทานแล้วแปลงเอาต์พุตจากคอมพาราเตอร์ให้ตรงกับรหัสดิจิทัล ซึ่งจะเห็นว่าอุปสรรคทางด้านความเร็วจะถูกจำกัดเพียง Propagation time ของคอมพาราเตอร์เท่านั้น แต่อุปสรรคที่สำคัญในการพัฒนางจรชนิดนี้นับเป็นไอซีคือ วงจรนี้ต้องการคอมพาราเตอร์ถึง $2^n - 1$ ตัว สำหรับ ADC 1 ตัว แต่ก็ได้ ADC ชนิดที่ทำงานรวดเร็วที่สุดเช่นกัน



รูปที่ 4.23 บล็อกไดอะแกรมแสดง Parallel ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

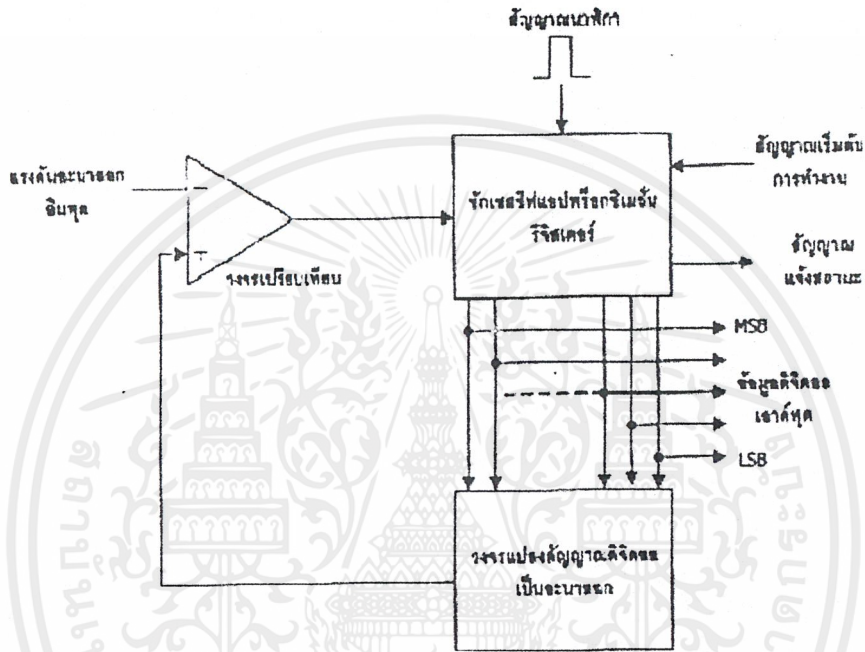
การแปลงสัญญาณอนาล็อกดิจิทัลแบบซัคเซสซีฟแอปพร็อกซิเมชัน

(Success Approximation ADC)

การแปลงสัญญาณอนาล็อกเป็นดิจิทัล (ADC) ที่ได้รับนิยมสูงและมีประสิทธิภาพดีคือการแปลงแบบซัคเซสซีฟแอปพร็อกซิเมชัน ไอซี ADC ที่เลือกมาทำการทดลองนี้ก็ใช้วงจร ADC แบบนี้ ก่อนถ้าจะแปลเป็นไทยอาจเรียกกระบวนการ ADC แบบซัคเซสซีฟแอปพร็อกซิเมชันนี้ว่า เป็นการแปลงแบบประมาณใกล้เคียง โค้ดแอมแกรมของการทำงานของกระบวนการ ADC แบบนี้แสดงในรูปที่ 3.1 ส่วนสำคัญหลักคือ วงจรเปรียบเทียบแรงดัน วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกหรือ DAC สัญญาณนาฬิกา และส่วนควบคุมลอจิกวงจร ADC แบบซัคเซสซีฟแอปพร็อกซิเมชันนี้จะใช้รีจิสเตอร์เลขฐานสองหรือไบนารีรีจิสเตอร์ในการส่งข้อมูลดิจิทัลของวงจร DAC ภายใน แต่ละบิตของรีจิสเตอร์จะเซตหรือรีเซตโดยการควบคุมจากวงจรควบคุม ต่อไปนี้จะอธิบายการทำงานของ ADC แบบนี้ไปที่ละขั้น ขอให้พิจารณาโค้ดแอมแกรมเวลาในรูปที่ 3.2 ประกอบด้วย

1. ส่งสัญญาณเริ่มต้นการทำงาน (start converter) มายังซัคเซสซีฟแอปพร็อกซิเมชันรีจิสเตอร์ (successive approximation register)
 2. ขณะนี้สถานะของรีจิสเตอร์จะไม่ว่าง (busy) สัญญาณนาฬิกาถูกส่งเข้ามาเพื่อกำหนดให้ค่าของรีจิสเตอร์เท่ากับ 0000
 3. เอาท์พุทของ DAC จะเป็น 0 V ส่งไปในวงจรเปรียบเทียบ เพื่อเปรียบเทียบกับ V_{in} ในขณะนี้จะได้เอาท์พุทเท่ากับ -5V กำหนดเป็นลอจิก "0"
 4. เมื่อสัญญาณนาฬิกาถูกส่งเข้ามา จะทำการเซตบิต MSB ของรีจิสเตอร์เป็น "1"
 5. ในกรณีเป็น ADC ขนาด 4 บิต ดังนั้นการที่บิต MSB เซต จะทำให้วงจร DAC แปลงค่าเป็นแรงดัน 8 V นำไปเปรียบเทียบกับวงจรเปรียบเทียบแรงดัน แต่อย่างน้อยก็ว่า V_{in} ดังนั้นเอาท์พุทของวงจรเปรียบเทียบยังคงเป็น "0" ทำให้รีจิสเตอร์ยังคงค่าบิต MSB ให้เป็น "1" ต่อไป
 6. ต่อมาบิต B2 (ถัดจากบิต MSB 1 บิต เนื่องจากมี 4 บิต กำหนดบิต MSB = B3) จะเซตมีค่าเท่ากับ 4 V นำไปรวมกับค่าของบิต MSB ที่มีอยู่ 8 V เช่น 12V นำไปเปรียบเทียบกับ V_{in} ก็ยังน้อยกว่ารีจิสเตอร์จึงยังคงค่า B2 ไว้ที่ "1" เช่นกัน
 7. ต่อมาบิต B1 จะเซตทำให้แรงดัน เอาท์พุทมา DAC กลายเป็น $8+4+2=14$ V ซึ่งมากกว่า V_{in} ทำให้วงจรเปรียบเทียบเกิดการเปลี่ยนสถานะเป็น "1" ซึ่งจะส่งสัญญาณมาควบคุมให้ B1 กลายเป็น "0"
 8. เมื่อบิต LSB ถูกเซต จะมีค่าแรงดัน 1 V เข้ามารวมกับค่าของ B3, B2, B1 เป็น $8+4+0+1=13$ V นำไปเปรียบเทียบกับ V_{in} ปรากฏยังน้อยกว่า V_{in} ทำให้ที่บิต B0 หรือ LSB มีค่าเป็น "1"
 9. ขณะนี้ทุกบิตในรีจิสเตอร์ถูกนำมาแปลงค่าเรียบร้อยแล้ว ทำให้สถานะของรีจิสเตอร์กลับมาเป็น พร้อมการทำงาน (ready)
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

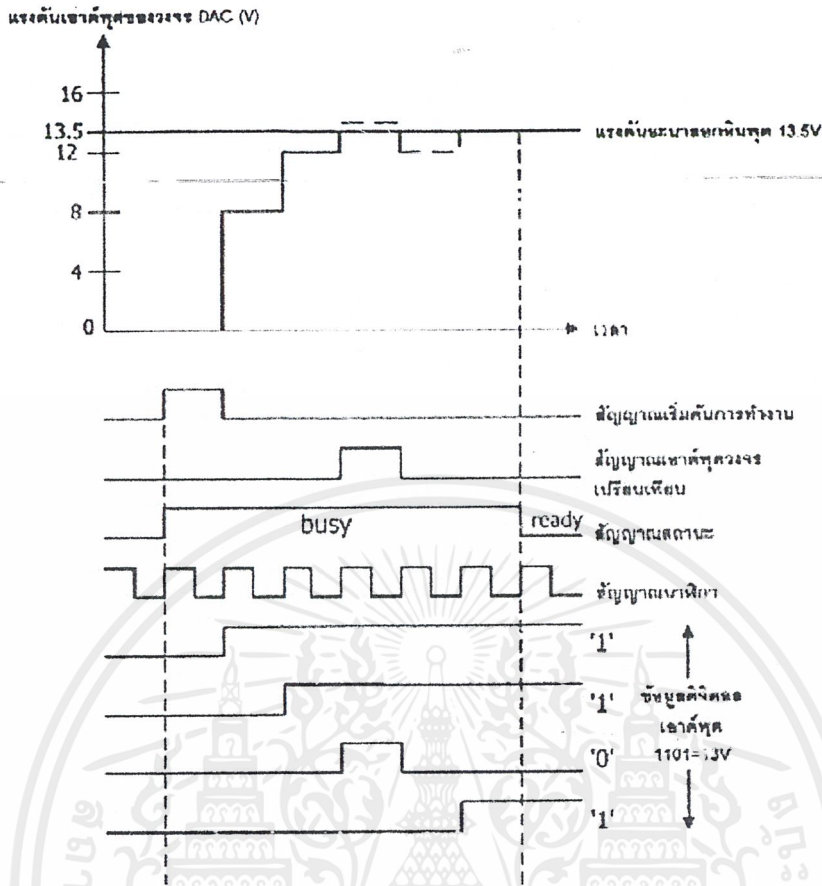
10. ข้อมูลดิจิทัลที่ได้จากการ ADC แบบนี้ จะมีค่า 1101 หรือ 13 V ซึ่งใกล้เคียงกับ V_{in} (13.5V) มากที่สุด ถ้าหากรีจิสเตอร์มีจำนวนบิตมากกว่านี้ ความละเอียดของข้อมูลที่แปลงได้จะมีความใกล้เคียงกันมากขึ้น ช่วงเวลาการแปลงสัญญาณจะเริ่มขึ้นตั้งแต่สัญญาณนาฬิกาถูกส่งเข้าไปเตรียมระบบ ไปจนถึงเมื่อสถานะของรีจิสเตอร์กลับมาเป็น “พร้อมการทำงาน” อีกครั้งหนึ่ง ซึ่งจะต้องใช้จำนวนสัญญาณนาฬิกาเท่ากับ $n+1$ พัลส์ โดย n เท่ากับจำนวนบิตของรีจิสเตอร์



รูปที่ 5.1 ไดอะแกรมแสดงการทำงานของวงจร ADC แบบซิกมาซีฟแอปพร็อกซิเมชัน

5.1 ความเที่ยงตรงของวงจร ADC

เป็นการเปรียบเทียบแรงดันอะนาล็อกของวงจร ADC กับแรงดันที่ควรเกิดขึ้นจริง ยกตัวอย่างที่ข้อมูลดิจิทัลสูงสุดของวงจร ADC ขนาด 8 บิตเมื่อเทียบเป็นแรงดันอะนาล็อกควรมีค่าเท่ากับ 5.0000V แต่จากการคำนวณได้ค่าแรงดัน 4.9804V นั่นคือเกิดความผิดพลาดไป 0.0195V หรือ 19.5mV แต่การบอกค่าความเที่ยงตรงของวงจร ADC มักระบุเป็นจำนวนที่เทียบกับ VLSB ดังนั้นในวงจร ADC ขนาด 8 บิตที่ยกเป็นตัวอย่างนี้จึงมีค่าความเที่ยงตรง (หรือบางที่เรียกเป็นค่าความผิดพลาด) เป็น $(+/-)1/2LSB$



รูปที่ 5.2 ไตอะแกรมเวลาแสดงการทำงานของวงจร ADC แบบซัสเซสซีฟแอปพริอ็อกซิเมชัน

5.2 ค่าเวลาในการแปลงสัญญาณ (conversion time)

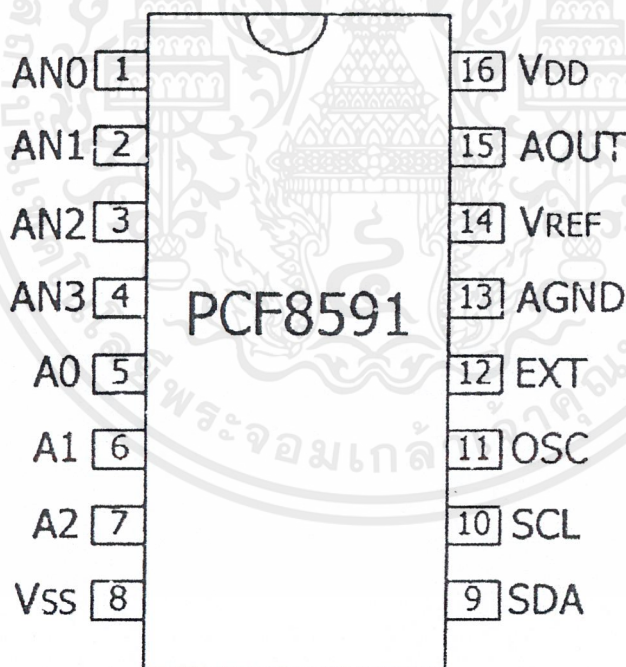
เป็นค่าของเวลาทั้งหมดที่วงจร ADC แบบวงจรมัลติเพล็กซ์และแบบซัสเซสซีฟแอปพริอ็อกซิเมชันใช้ในการแปลงสัญญาณอนาล็อกเป็นดิจิทัลจนเสร็จสิ้น พารามิเตอร์ตัวนี้มักจะปรากฏในคุณสมบัติของไอซีที่ทำงานเป็นวงจร ADC เมื่อไอซีทำการแปลงสัญญาณเสร็จสิ้นลง จะส่งสัญญาณที่เรียกว่า EOC (End of Conversion) ค่าเวลาในการแปลงสัญญาณของวงจร ADC จะขึ้นอยู่กับจำนวนบิตของวงจร, ค่าความถี่ของสัญญาณนาฬิกาที่ใช้ในการแปลงสัญญาณและขนาดของสัญญาณอนาล็อกอินพุต

5.3 ข้อมูลเบื้องต้นของ PCF8591

ในการทดลองนี้จะใช้ไอซี ADC ที่มีความสามารถสูงเบอร์ PCF8591 เนื่องจากในคัมมันมีวงจร ADC แบบซัสเซสซีฟแอปพริอ็อกซิเมชันขนาด 8 บิตสูงถึง 4 ช่อง ทั้งยังมีวงจร DAC อีก 1 ช่องด้วย ระบบการเชื่อมต่อเป็นระบบบัส I²C ทำให้ใช้สายสัญญาณเพียง 2 เส้น ทั้งยังสามารถต่อพ่วงกันได้สูงสุด 8 ตัว ทำให้ได้วงจร ADC รวมสูงถึง 32 ช่อง และวงจร DAC รวม 4 ช่อง ทำให้สามารถนำไปประยุกต์ใช้งานได้กว้างขวาง มีรายละเอียดคุณสมบัติทางเทคนิคดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทำงานโดยใช้แหล่งจ่ายไฟชุดเดียว
2. ทำงานที่แรงดัน 2.5V ถึง 6V
3. กินกระแสขณะอยู่ในสภาวะ (stand by) ต่ำ
4. ติดต่อกับไมโครคอนโทรลเลอร์หรือไมโครคอนโทรลเลอร์ผ่านระบบบัส I²C
5. สามารถเลือกตำแหน่งแอดเดรสทางฮาร์ดแวร์จากขา A0,A1,A2 ทำให้สามารถต่อพ่วงกันได้สูงสุดถึง 8 ตัว
6. อัตราการสุ่มข้อมูล (sampling) ขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาบนบัส บัส I²C
7. วงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัล (ADC) สามารถรับสัญญาณอะนาล็อกได้ 4 ช่อง ทั้งยังเลือกได้ว่าให้ทำงานแบบแยกช่องหรือทำงานเป็นวงจรดิฟเฟอเรนเชียล
8. การอ่านค่าสามารถกำหนดให้เลื่อนช่องอินพุตโดยอัตโนมัติได้
9. สัญญาณอะนาล็อกมีระดับแรงดันตั้งแต่ V_{ss} ไปจนถึง V_{dd}
10. วงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัลเป็นแบบซัสเซสซีฟแอมป์ร็อกซิมเมชันขนาด 8 บิต
11. มีวงจรแปลงสัญญาณดิจิทัลเป็นอะนาล็อกขนาด 8 บิต 1 ช่อง



รูปที่ 5.3 การจัดขาของไอซี ADC/DAC ขนาด 8 บิตผ่านบัส I²C เบอร์ PCF8591

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PCF8591 สามารถทำหน้าที่เป็นไอซีแปลงสัญญาณอะนาล็อกเป็นดิจิทัลขนาด 8 บิต 4 ช่อง และทำหน้าที่เป็นไอซีแปลงสัญญาณดิจิทัลเป็นอะนาล็อกได้ในคราวเดียวกัน ด้วยการควบคุมผ่านระบบบัส I²C ทำให้สามารถต่อพ่วงไอซี PCF8591 ได้สูงสุด 8 ตัว รองรับการอ่านค่าสัญญาณอะนาล็อก อินพุตได้สูงสุดถึง 32 ช่อง และสามารถส่งสัญญาณอะนาล็อกเอาต์พุตสูงสุดได้ 8 ช่อง ด้วยการกำหนดแอดเดรสจากขา A0,A1,A2 การจัดขาของ PCF8591 แสดงในรูปที่ 3.3 ส่วนรายละเอียดตำแหน่งขาต่างๆ มีดังนี้

ขา AN0-AN3 (ขา 1-4) เป็นขาอินพุตสำหรับป้อนสัญญาณอะนาล็อกที่ต้องการแปลงค่า

ขา A0-A2(ขา 5-7) เป็นขาสำหรับกำหนดข้อมูลแอดเดรสทางฮาร์ดแวร์ ปกติต้องลงกราวด์แต่ถ้ามีการใช้งาน PCF8591 มากกว่า 1 ตัวต้องกำหนดการต่อขา A0-A2 ของ PCF8591 ให้ไม่ตรงกัน จึงทำให้สามารถต่อใช้งานร่วมกันได้สูงสุด 8 ตัว

ขา Vss (ขา 8) เป็นขาต่อกราวด์

ขา SDA,SCL (ขา 9 และ ขา 10) เป็นขาสำหรับต่อกับสัญญาณบัส I²C

ขา OSC (ขา 11) เป็นขาสำหรับต่อกับสัญญาณนาฬิกาภายนอกเมื่อขา EXT ต่อกับไฟ +5V และจะทำงานเป็นขาเอาต์พุตสัญญาณนาฬิกาถ้าขา EXT ต่อกับกราวด์

ขา EXT (ขา 12) เป็นขาสำหรับเลือกแหล่งกำเนิดสัญญาณนาฬิกา ถ้าต่อไฟ +5V จะเป็นการเลือกใช้สัญญาณนาฬิกาจากภายใน

ขา AGND (ขา 13) เป็นขากราวด์ของแรงดันอ้างอิง ปกติต้องลงกราวด์

ขา Vref(ขา 15) เป็นขาเอาต์พุตของวงจรแปลงสัญญาณดิจิทัลเป็นอะนาล็อก

ขา Vdd (ขา 16) เป็นขาต่อไฟเลี้ยง จ่ายได้ตั้งแต่ +2V ถึง +6V ปกติใช้ +5V

5.3 รายละเอียดฟังก์ชันต่างๆของ PCF8591

5.3.1 ตำแหน่งแอดเดรส

ในระบบบัส I²C การติดต่อกับอุปกรณ์แต่ละตัวต้องระบุแอดเดรสของอุปกรณ์เหล่านั้นอย่างชัดเจน ถ้าเป็นการอ้างอิงแบบ 7 บิต ข้อมูลกำหนดแอดเดรส 4 บิตบนจะเป็นค่าแอดเดรสเฉพาะของอุปกรณ์ตัวนั้นๆ ที่กำหนดมาจากผู้ผลิต ผู้ใช้งานไม่สามารถเปลี่ยนแปลงได้ สำหรับไอซี PCF8591 จะมีค่าเท่ากับ 1001 (ฐานสิบ) ข้อมูล 3 บิตถัดมาจะเป็นค่าของแอดเดรสที่ผู้ใช้งานสามารถกำหนดได้ทางฮาร์ดแวร์เพื่อเลือกไอซี PCF8591 ที่ต้องการติดต่อกับในกรณีที่มีการต่อใช้งาน PCF8591 มากกว่า 1 ตัว ส่วนบิต LSB ใช้ในการกำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับไอซีตัวนั้นๆ

5.3.2 ข้อมูลควบคุม

หลังจากส่งข้อมูลกำหนดแอดเดรสให้แก่ PCF8591 แล้ว ต้องส่งข้อมูลควบคุมตามไปด้วยเพื่อกำหนดคุณสมบัติวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัลและวงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อกภายใน PCF8591 โดยมีรายละเอียดของข้อมูลในแต่ละบิตดังในรูปที่ 3.4

บิต 6 ของข้อมูลควบคุมใช้สำหรับการอินาเบิลอะนาล็อกเอาต์พุต เมื่อต้องการอินาเบิลต้องกำหนดให้ขานี้เป็น “1”

บิต 4 และ บิต 5 ของข้อมูลควบคุมใช้สำหรับการกำหนดรูปแบบของสัญญาณอนาล็อกอินพุตที่ป้อนให้แก่ PCF8591

บิต 2 ใช้สำหรับการเลือกรูปแบบการอ่านข้อมูลจากขาอินพุตอะนาล็อกกว่าจะเป็นการอ่านจากเพียงอินพุตเดียวหรืออ่านแบบเรียงอันดับทุกอินพุต ถ้าต้องการเลือกให้อ่านแบบเรียงอันดับต้องกำหนดให้บิตนี้เป็น “1”

บิต 0 และ บิต 1 ใช้สำหรับกำหนดช่องอินพุตอะนาล็อกที่ต้องการอ่าน ถ้ากำหนดให้บิต 2 เป็น “1” หลังจากอ่านค่าของบิต “0” และบิต “1” แล้ว ในการอ่านครั้งต่อไปจะเป็นการอ่านค่าอินพุตจากช่องที่ 1

ข้อมูลควบคุมทั้งหมดจะถูกเก็บไว้ในรีจิสเตอร์ควบคุมภายใน PCF8591 เมื่อจ่ายไฟให้แก่ PCF8591 ครั้งแรก บิตต่างๆของข้อมูลภายในรีจิสเตอร์ควบคุมจะเป็น “0”

5.3.3 วงจรออสซิลเลเตอร์

วงจรออสซิลเลเตอร์ภายใน PCF8591 จะสร้างสัญญาณนาฬิกาสำหรับการแปลงสัญญาณอนาล็อกเป็นดิจิทัล เมื่อต้องการใช้วงจรออสซิลเลเตอร์ภายใน ขา EXT ต้องต่อลงกราวด์ ถ้าต้องการ OSC ของ PCF8591 โดยความถี่ของสัญญาณสูงสุดที่ป้อนให้กับออสซิลเลเตอร์เท่ากับ 1.25 MHz

5.3.3 การอ่านค่าข้อมูลอินพุตอะนาล็อกของ PCF8591

มีลำดับขั้นตอนดังนี้

1. เตรียมข้อมูลกำหนดแอดเดรส โดยในที่นี้กำหนดแอดเดรสของ PCF8591 ไว้ที่ 0000 (ขา A0,A1A2 ต่อลงกราวด์ทั้งหมด) และให้ทำงานในโหมดเขียนข้อมูล (ป้อนข้อมูลลอจิก “0” ให้แก่บิต R/W)
2. เรียกโปรแกรมย่อยการติดต่อกับอุปกรณ์สเลฟ
3. ส่งข้อมูลควบคุมไปยัง PCF8591
4. ส่งสัญญาณ STOP
5. เรียกโปรแกรมย่อยการติดต่อกับอุปกรณ์สเลฟ

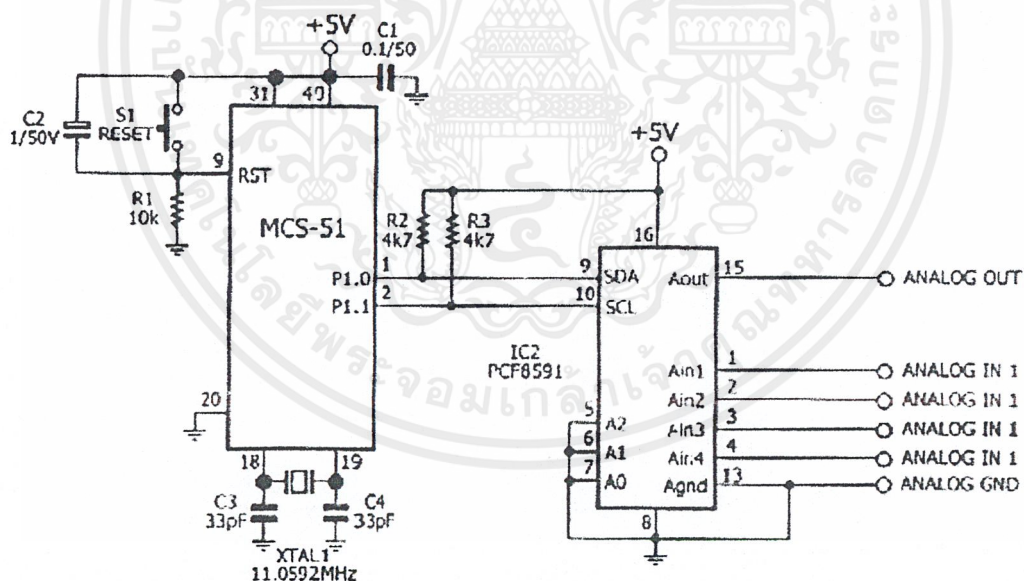
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ส่งข้อมูลกำหนดแอดเดรสอีกครั้งโดยครั้งนี้กำหนดให้เป็นโหมด อ่านข้อมูล (ส่งลอจิก “1” ให้แก่บิต R/W) เพื่อเริ่มต้นการอ่านค่าข้อมูลจากช่องสัญญาณอะนาล็อกอินพุต
7. อ่านค่าจากขาอินพุตของวงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัลของช่องที่ 1
8. หากต้องการอ่านค่าในช่องต่อไปก็ให้เริ่มต้นการติดต่อใหม่ ดังนั้นในการเขียนโปรแกรมเพื่ออ่านค่าต่อเนื่องทั้ง 4 ค่าหรือมากกว่าจึงต้องเขียน โปรแกรมลูปเพื่อกำหนดรอบการทำงาน 4 รอบหรือมากกว่า ก็จะสามารถอ่านได้ครบทุกช่อง

5.3.4 การเขียนข้อมูลไปยังวงจรแปลงสัญญาณดิจิทัลเป็นอะนาล็อกของ PCF8591

การเขียนข้อมูลไปยังขาอะนาล็อกเอาต์พุตมีข้อแตกต่างจากการอ่านข้อมูลดังนี้

1. เรียกโปรแกรมน้อยการติดต่อกับอุปกรณ์สเลฟ
2. ส่งข้อมูลควบคุม 40H ไปยัง PCF8591 เพื่ออีนามัลติอะนาล็อกเอาต์พุต
3. ส่งข้อมูลไปยังเอาต์พุตอะนาล็อก โดยค่าที่ส่งออกไปจะต้องมีค่าอยู่ระหว่าง 0-255
4. ส่งสถานะหยุด



รูปที่ 5.4 ตัวอย่างการเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับ PCF8591

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

แผงเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์

6.1 ชุดแผงติดตั้งเซลล์แสงอาทิตย์

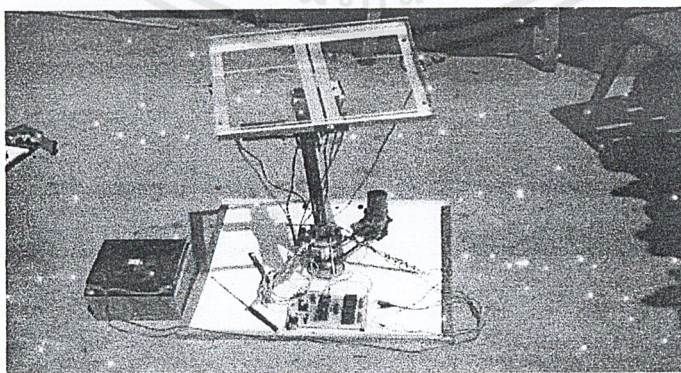
การประยุกต์ใช้งานบอร์ดควบคุมรีเลย์เอนกประสงค์ สามารถนำไปควบคุมติดต่ออุปกรณ์ต่างๆ ได้โดยการเพิ่มเติมเงื่อนไขการทำงานของรีเลย์ ในโครงงานนี้จะนำบอร์ดดังกล่าวมาใช้ควบคุมแผงติดตั้งเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์ มีหลักการและส่วนประกอบดังนี้

1. ชุดแผงติดตั้งเซลล์แสงอาทิตย์ ตรวจสอบความเข้มของแสงโดยการเปรียบเทียบค่าที่ได้จากตำแหน่งของการติดตั้งด้านทานปรับค่าตามแสงโดยติดตั้ง 4 ตัว ตรงกลางของด้านทั้ง 4 ของแผงเซลล์แสงอาทิตย์ตัวด้านทานปรับค่าได้ตามแสง ควรมีฉากกันเพื่อแสงเข้ามาในแนวเส้นตรงทำให้เกิดความแม่นยำในการเปรียบเทียบค่าและลดสัญญาณรบกวนจากภายนอก ตัวด้านทานเปลี่ยนค่าตามแสงมี 4 ตัว แบ่งเป็น 2 ชุด คือ บน - ล่าง 1ชุด และ ซ้าย - ขวา อีก 1ชุด

ชุดที่ 1 บน - ล่าง ใช้ควบคุมมอเตอร์เพื่อขยับแผ่นรองเซลล์แสงอาทิตย์ให้หมุนในแนวตะวันออก-ตะวันตกโดยของขาของการหมุนนั้นสามารถกำหนดข้อจำกัดขององศาในการหมุน โดยใช้ลิมิตสวิตช์ ป้องกันทั้ง 2 ข้าง เพื่อให้ไม่ทำให้แผ่นรองเซลล์แสงอาทิตย์ หมุนในองศาที่ตั้งฉากกับพื้นมากเกินไปจะทำให้แกนหลักไม่สมดุลอาจเกิดความเสียหายได้

ชุดที่ 2 ซ้าย - ขวา ใช้ควบคุมมอเตอร์เพื่อหมุนแกนหลักของชุดแผงติดตั้งเซลล์แสงอาทิตย์ ให้หมุนในแนวแกน เหนือ - ใต้ โดยของขาของการหมุนนั้นสามารถกำหนดข้อจำกัดขององศาในการหมุน โดยใช้ ลิมิตสวิตช์ป้องกันทั้ง 2 ข้างเช่นกัน

ดีซี มอเตอร์ 9 1 โวลต์ พร้อมชุดเฟือง 2 ชุด เป็นตัวหมุนชุดแผงติดตั้ง เซลล์แสงอาทิตย์ทั้ง 2 แกนควบคุมจากวงจรควบคุมที่มีไมโครคอนโทรลเลอร์ MCS51 เป็นตัวควบคุม โดยใช้แบตเตอรี่ภายนอกเป็นแหล่งจ่ายให้กับมอเตอร์



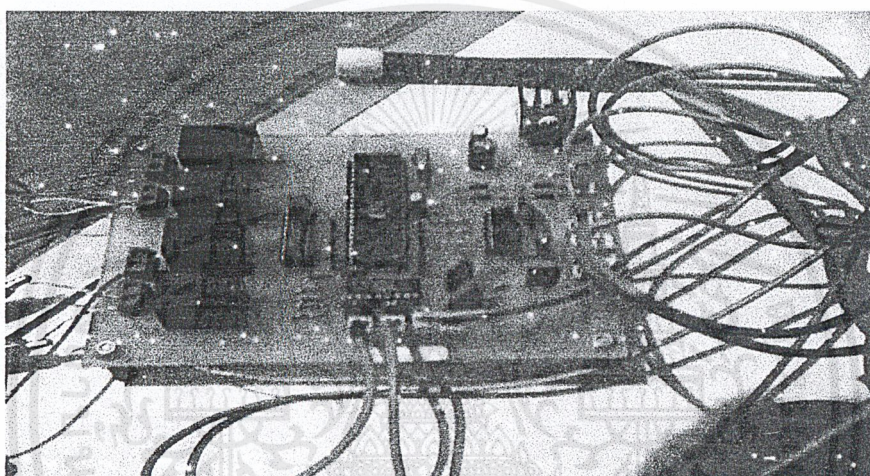
รูปที่ 6.1 ชุดแผงติดตั้งเซลล์แสงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 วงจรควบคุมชุดแผงติดตั้งเซลล์แสงอาทิตย์

วงจรควบคุมประกอบด้วย

1. LDR ตรวจสอบความเข้มของแสง
2. PCF 8591 8 bit A/D And D/A Converter
3. AT89s8252 Microcontroller
4. DN74LS08N And Gate
5. ULN2803A Darlington Arrays
6. L7805CW Voltage Regulator 5 Volt



รูปที่ 6.2 วงจรควบคุม

1. วงจรตัวต้านทานแปลงค่าตามแสง

การทำงานของวงจร

LDR จะแปรค่าตามความเข้มของแสง ค่าความต้านทานของตัวต้านทานแปลงค่าผกผันกับตามแสงจะลดลง โดยเมื่อแสงมีความเข้มมากขึ้นความต้านทานของตัวต้านทานแปลงค่าตามแสงจะลดลง จากค่าที่ทดลองจะได้ประมาณ $1k\Omega$ และเมื่อแสงมีความเข้มลดลงความต้านทานของตัวต้านทานแปลงค่าตามแสงจะมากขึ้น จากค่าที่ทดลองจะได้ประมาณ $100k\Omega$ ดังนั้นการคำนวณค่าความต้านทานที่นำมา ต่อร่วมกับออกแบบตัวต้านทานแปลงค่าตามแสงในลักษณะ Voltage Divider

การคำนวณค่าความต้านทาน

$$V_o = V_{cc} * R_{LDR} / (R + R_{LDR}) \quad (6.1)$$

ความต้านทานที่เหมาะสมจะอยู่ในช่วง $1/20 > R_{LDR(MAX)} > 20R_{LDR(MIN)}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีความเข้มของแสงมากที่สุด

$$R = 20R_{LDR(MIN)}$$

$$V_o = 5V * R_{LDR(MIN)} / (20R_{LDR(MIN)} + R_{LDR(MIN)})$$

$$= 0.24 \text{ mV}$$

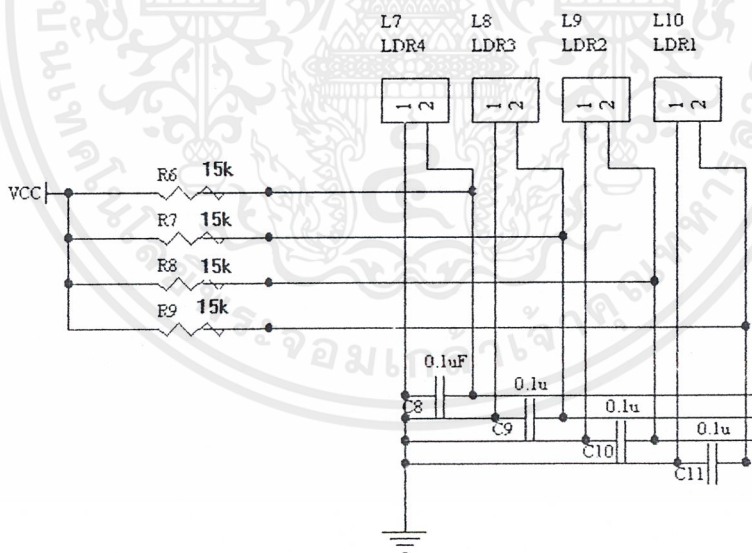
ในกรณีความเข้มของน้อยสุด

$$R = 1/20R_{LDR(MIN)}$$

$$V_o = 5V * R_{LDR(MIN)} / (1/20R_{LDR(MIN)} + R_{LDR(MIN)})$$

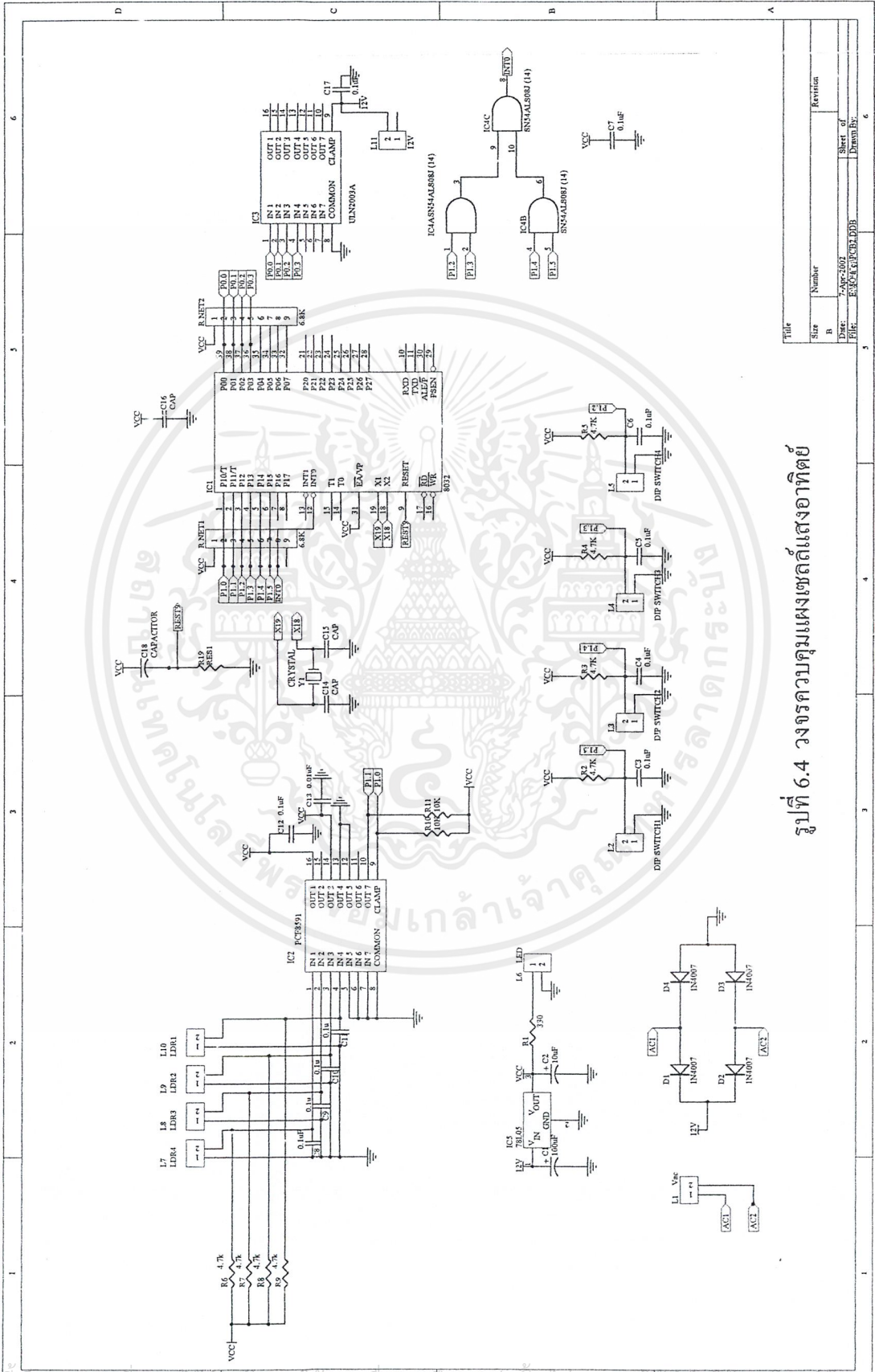
$$= 4.76 \text{ mV}$$

จากค่าที่คำนวณได้อยู่ในช่วง 0-Vcc จึงเลือกค่า R ในการทดลอง 3 ค่า คือ 10 KΩ, 15 KΩ และ 20 KΩ ปรากฏว่าค่าที่ทำให้ได้ผลตอบสนองที่ดีที่สุดอยู่ที่ประมาณ 15 KΩ ผลของการเปลี่ยนแปลงของแรงดัน ทำให้ได้ค่า ดิจิตอลแปรค่าความแตกต่างอย่างเห็นได้ชัด



รูปที่ 6.3 วงจรตัวต้านทานแปรค่าตามแสง

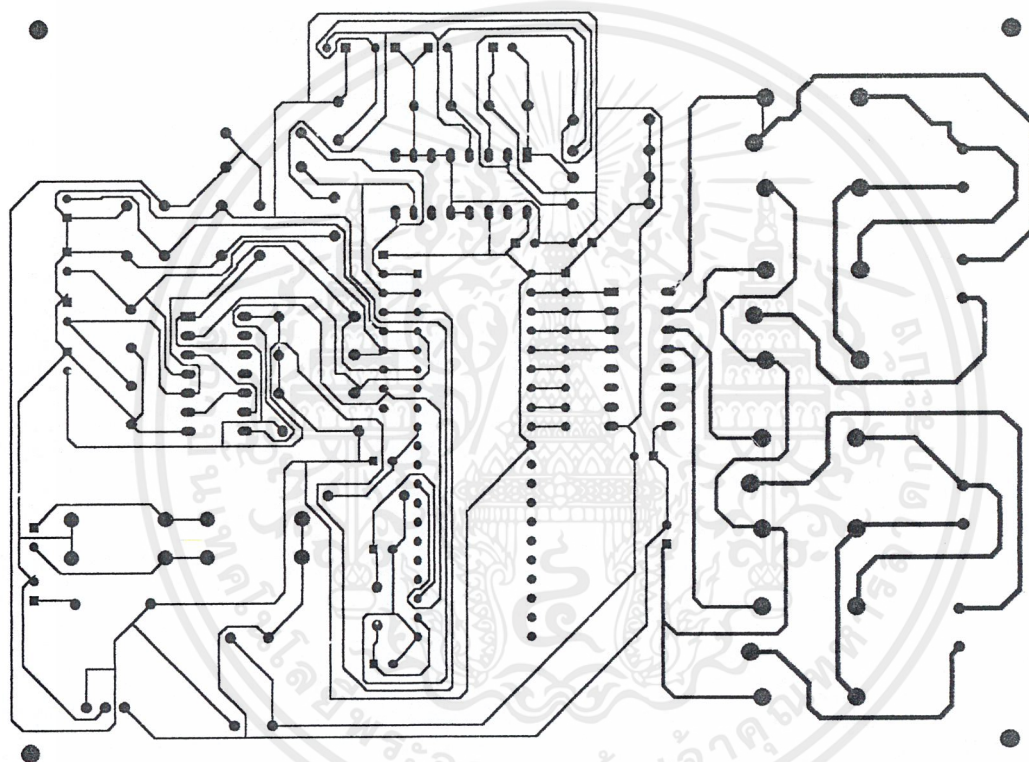
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 วงจรควบคุมแผงเซลล์แสงอาทิตย์

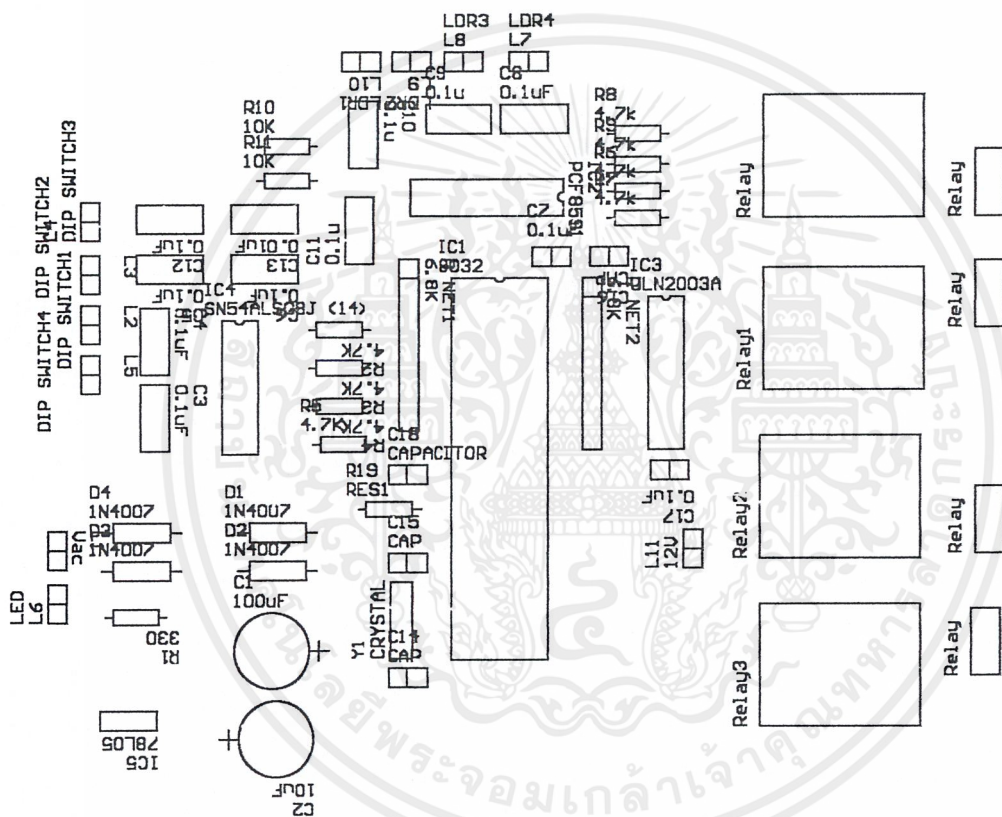
Title	Number	Revision
Size	B	
Date	7-Apr-2001	Sheet of
File	ESP32-LED.DDB	Drawn By

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.5 ลายทองแดงบนแผ่นวงจรต้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

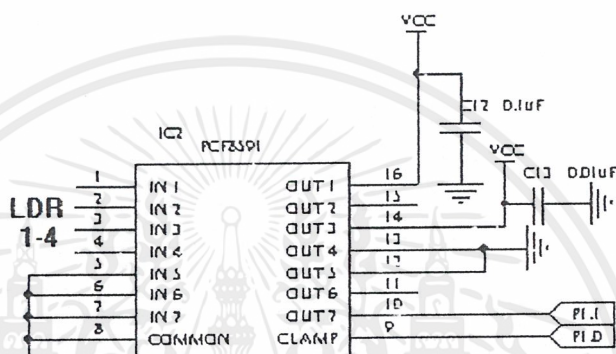


รูปที่ 6.6 ตำแหน่งการวางอุปกรณ์บนแผ่นวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

PCF 8591 เป็นวงจรแปลงสัญญาณอนาล็อกแบบซิงเกิลชิพแอปพลิเคชันที่รับค่าแรงดันที่มาจากวงจรตัวต้านทานแปรค่าตามแสงจาก 0 – 5 โวลต์ เป็น ดิจิตอล 8บิต จาก 0 – 0FFh เพื่อส่งค่าที่ได้ให้กับ AT89s8252 ไอซี PCF 8591 สามารถรับอนาล็อกอินพุตได้ 4แชนแนล การติดต่อกับ AT89s8252 ใช้หลักการของ I²C ของบริษัท Phillip Semiconductor โดยใช้ขาสัญญาณในการติดต่อเพียง 2 ขาเท่านั้น



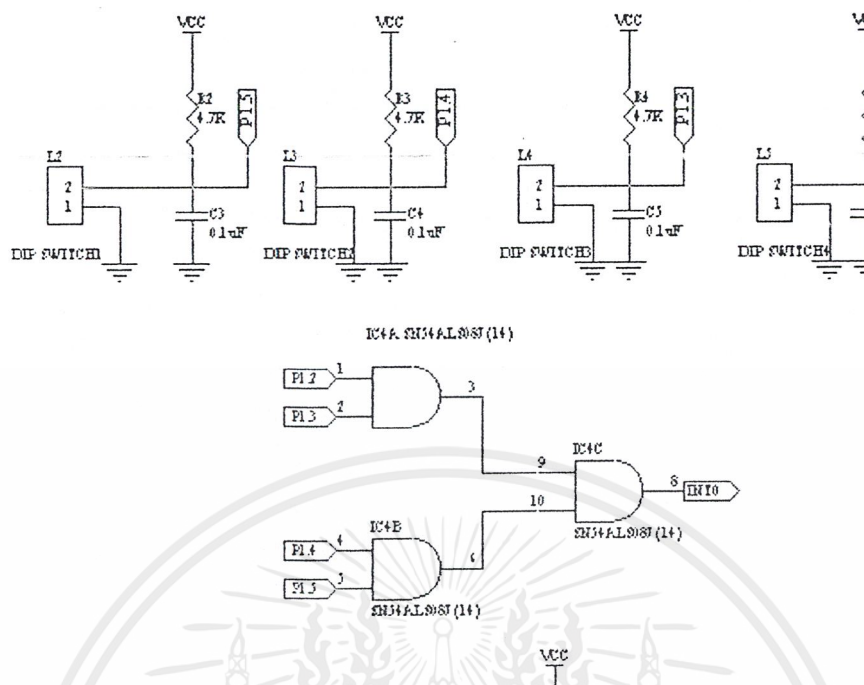
รูปที่ 6.7 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

3. วงจร ลิมิตสวิตช์

ลิมิตสวิตช์ทำหน้าที่กำหนดองศาของการหมุนของแกนแผงรับแสงอาทิตย์ทั้ง 2 แกน แบ่งเป็น 2 ชุดดังนี้

ชุดที่ 1 มีเป็นลิมิตสวิตช์ 2 ตัวทำงานกับแกน ตะวันออก – ตะวันตก เพื่อหมุนตามแสงอาทิตย์ในแต่ละวัน เมื่อแผงรับหมุนตามดวงอาทิตย์ไปจนชน ลิมิตสวิตช์ตัวที่ 2 มอเตอร์จะหยุดทำงานทันทีเพื่อป้องกันแผงรับชนกับแกนจนกว่าดวงอาทิตย์ของวันใหม่จะขึ้นชุดวงจรควบคุมจะเปรียบเทียบแสงเพื่อหมุนมอเตอร์ไปทางตะวันออกจนชนลิมิตสวิตช์ตัวที่ 1 เพราะดวงอาทิตย์ยังขึ้นไม่ตรงกับองศาที่กำหนดเมื่อดวงอาทิตย์ขึ้นจนถึงมุมที่กำหนดก็จะเริ่มควบคุมการทำงานของมอเตอร์อีกครั้ง

ชุดที่ 2 ทำงานร่วมกับแกนเหนือ – ใต้ ซึ่งมีการทำงานน้อยมากเนื่องจากในแต่ละวันความแตกต่างขององศาการขึ้นของดวงอาทิตย์จะไม่ต่างกันมากนัก หลักการการทำงานจะเหมือนกับชุดที่ 1 แต่จะตรวจสอบการขึ้นของดวงอาทิตย์ในแต่ละวันเท่านั้น

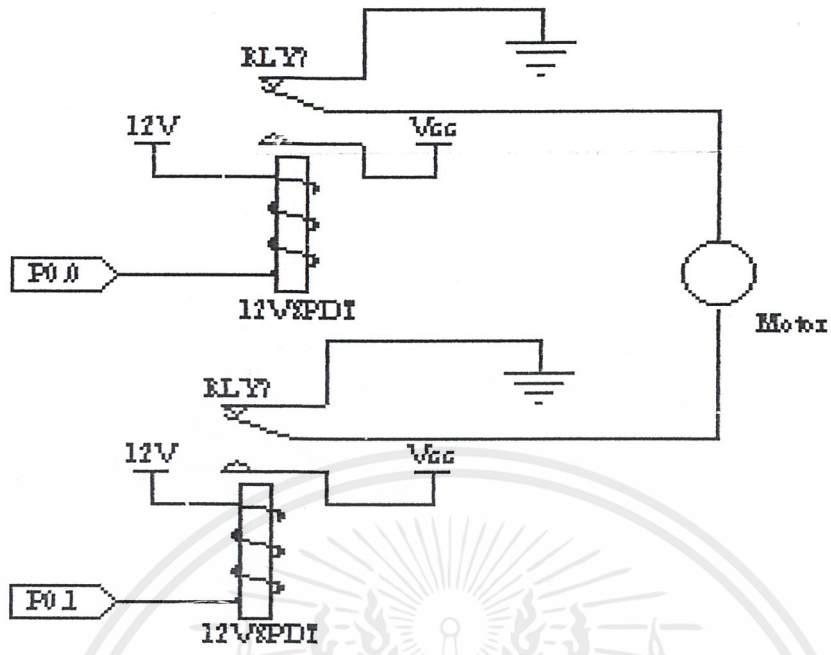


รูปที่ 6.8 วงจรลิมิตสวิตช์

ลิมิตสวิตช์จะต่อแบบ ปกติเปิด โดยเมื่อไม่มีการกดสวิตช์ตัวใดเลยจะทำให้ค่าเอาต์พุตของ AND GATE 7408 ชุดที่ 3 ที่ต่ออยู่กับ INT 0 ของ AT89s8252 เป็น 1 เมื่อมีการกดสวิตช์ตัวใดตัวหนึ่ง ให้ค่าเอาต์พุตของ AND GATE 7408 ชุดที่ 3 ที่ต่ออยู่กับ INT 0 ของ AT89s8252 เป็น 0 จะเกิดการอินเทอร์รัพท์ (Interrupt) ขึ้น เพื่อหยุดการทำงานของมอเตอร์ เนื่องจากสวิตช์แต่ละตัวต่ออยู่กับพอร์ตของ AT89s8252 จึงสามารถอ่านค่าเพื่อรับรู้การอินเทอร์รัพท์ ว่าเกิดการกดสวิตช์ที่ตัวใด เพื่อกำหนดการหยุดการหมุนของมอเตอร์ในทิศทางที่ถูกต้อง

4. ชุดขับมอเตอร์ พร้อมกลับทิศทาง

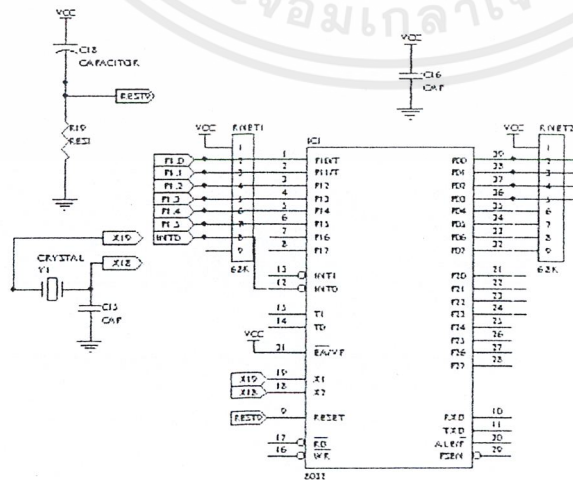
การควบคุมการทำงานของ ดีซีมอเตอร์โดย ULN 2803 เป็นตัวจ่ายกระแสให้ กับรีเลย์ซึ่งจ่ายกระแสสูงสุดได้ 500 mA ทำงานในลักษณะอินเวอร์สค่าที่ส่งมาจากพอร์ต โดยรีเลย์เป็นรีเลย์แบบSP-DT 12Volt 2ตัวต่ออยู่ในลักษณะที่สามารถกลับทางการหมุนของมอเตอร์ได้โดยการสั่งจากพอร์ต



รูปที่ 6.9 ชุดขับมอเตอร์ พร้อมกลับทิศทาง

5. หน่วยประมวลผลกลาง

CPU ในตระกูล MCS 51 เบอร์ AT89s8252 รับค่าจากการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล 8บิต จาก PCF 8591 เพื่อคำนวณค่าความแตกต่างของตัวเลขดิจิทัลที่ได้รับ กำหนดการทำงานของมอเตอร์แต่ละตัว พร้อมทั้งตรวจสอบการทำงานของลิmitsวิตช์ด้วย



รูปที่ 6.10 AT89s8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 โปรแกรมควบคุมชุดแผงติดตั้งเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์

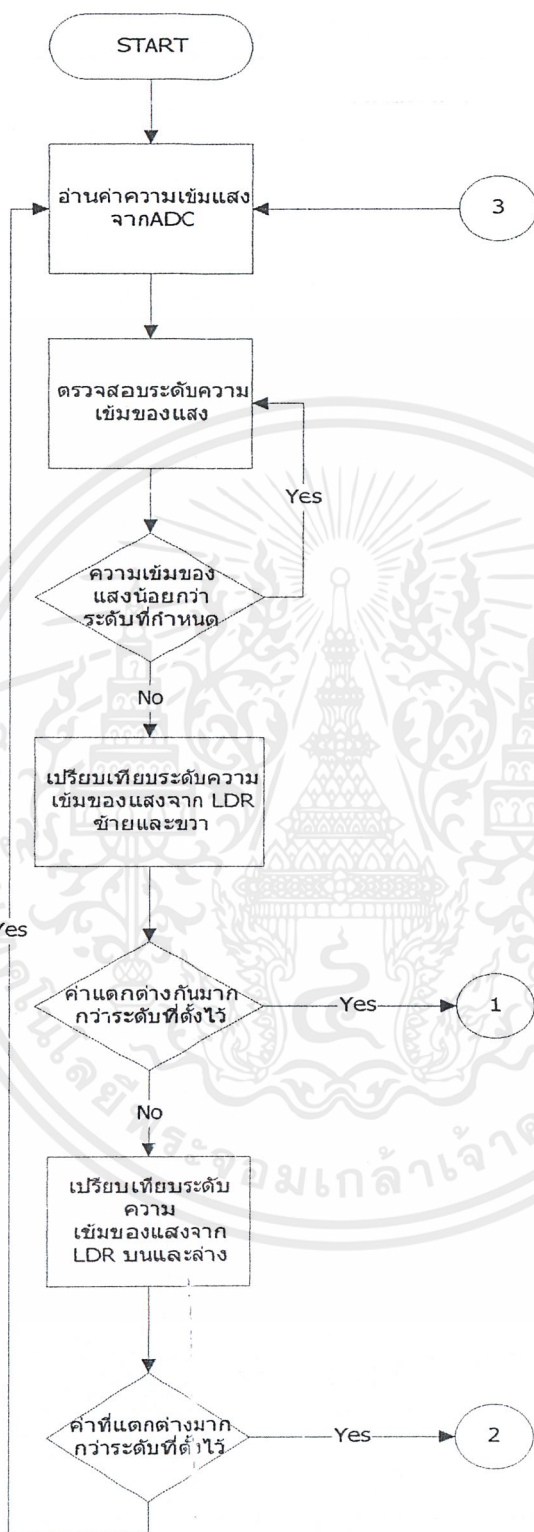
การโปรแกรมโดยใช้ภาษาแอสเซมบลี มีเงื่อนไขการทำงานคือ

1.รับค่าข้อมูลดิจิทัล 8 บิต จาก PCF 8591 โดยใช้การติดต่อกันแบบ I²C เป็นการติดต่อแบบ Synchronous โดยใช้ขา SCL เป็นตัวควบคุมจังหวะการส่งข้อมูล นำค่าของข้อมูล LDR บน-ล่าง โดยกำหนดความต่างของค่าที่ได้รับโดยประมาณค่านี้ขึ้นอยู่กับระยะห่างของการติด LDR และคุณสมบัติของLDR ด้วย การกำหนดค่าความต่างของ LDR ซ้าย-ขวา ก็เช่นเดียวกัน

2.ตรวจสอบลิมิตสวิทช์โดยใช้ L7408CW ซึ่งเป็น And Gate ตรวจสอบค่าที่อ่านจากลิมิตสวิทช์เพื่ออินเทอร์รัพท์ หยุดการทำงานของมอเตอร์เมื่อแกนทั้ง 2 หมุนไปชนลิมิตสวิทช์ โดยมีรายละเอียดทางโปรแกรมควบคุมการทำงานของแผงเซลล์แสงอาทิตย์ได้แสดงไว้ในภาคผนวก ก. และมีโฟลชาร์ตของโปรแกรมมีลำดับขั้นตอนดังนี้

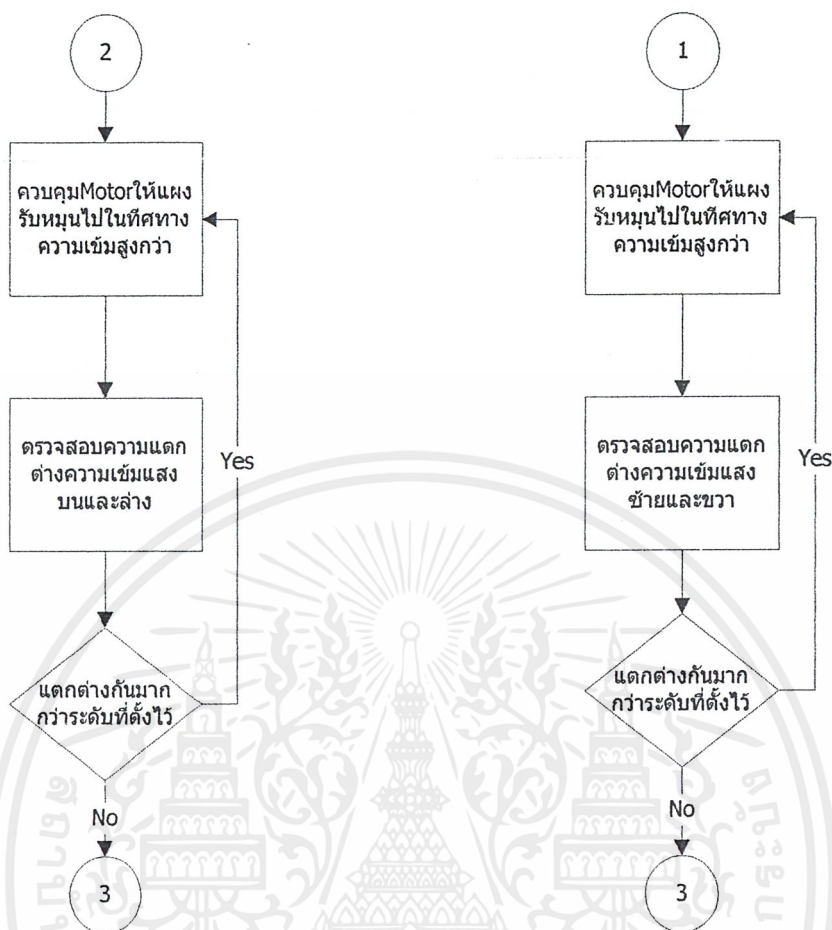


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.11 โปรแกรมควบคุมแผงติดตั้งเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.12 โปรแกรมควบคุมแมงติดตั้งเซลล์แสงอาทิตย์เคลื่อนที่ตามแสงอาทิตย์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทวิจารณ์และสรุป

จากการทดลองค่าความผิดพลาดที่เกิดจากการรับแสงของ LDR เกิดจากการที่แสงตกกระทบกับตัว LDR ไม่อยู่ในแนวมุมฉาก ดังนั้นจึงต้องเพิ่มจากกันแสงของ LDR การตรวจจับแสงจะมีความถูกต้องมากขึ้น เนื่องจากดวงอาทิตย์หมุนไปเล็กน้อยก็สามารถเปรียบเทียบค่าความต่างได้ชัดเจนขึ้น แต่จะทำให้มอเตอร์ตัดต่อถี่ขึ้น ทำให้สิ้นเปลืองพลังงานจึงจำเป็นต้องหาค่าเวลาที่เหมาะสม นอกจากนั้นมอเตอร์ที่เป็นดีซีมอเตอร์ ทำให้มีแรงเฉื่อยในขณะที่สั่งให้มอเตอร์หยุด มอเตอร์จะไม่หยุดทันที จึงแก้ไขโดยการใช้มอเตอร์ที่มีชุดเฟืองทำให้มอเตอร์หมุนช้าลงและมีแรงบิดมากขึ้น โครงการนี้ยังไม่ผ่านเซลล์แสงอาทิตย์มาประกอบหากต้องการที่จะนำไปใช้งานจริงจะต้องมี เซลล์แสงอาทิตย์ และชุดชาร์จประจุเพิ่มเติม

การใช้เซลล์แสงอาทิตย์นั้นจะมีประโยชน์กับท้องถิ่นที่ยังไม่มีไฟฟ้าใช้ ซึ่งเราสามารถประยุกต์ใช้โครงการนี้โดยการทำแผงเซลล์แสงอาทิตย์หลายๆแผงนำมาต่อกัน ก็จะพลังงานมากพอที่จะสามารถนำไปใช้งานได้

กิตติกรรมประกาศ

ปริญญาโทฉบับนี้สำเร็จได้ด้วยดีจากความช่วยเหลือของอาจารย์และบุคคลหลายท่าน ขอขอบคุณอาจารย์รพพงศ์ ตั้งศรีรัตน์ ที่ให้คำแนะนำปรึกษาชี้แนะแนวทางต่างๆคำแนะนำของอาจารย์ทำให้งานง่ายขึ้น

ขอขอบคุณภาควิชาวิศวกรรมระบบควบคุมอาจารย์ทุกท่านที่ให้ความรู้ตลอดเวลาสามปีที่เรียนมาและเจ้าหน้าที่อำนวยความสะดวกในการยื่นคืนอุปกรณ์ทำโครงการ

ขอขอบคุณเพื่อนเพื่อนภาคคอนโทรลทุกคนที่ช่วยเหลือและให้กำลังใจอย่างสม่ำเสมอ

ขอขอบคุณ นายวิรัตน์ ทองจันทร์แก้ว รุ่นพี่วิศวกรรมอิเล็กทรอนิกส์ที่ให้คำแนะนำด้านไมโครคอนโทรลเลอร์และอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในโครงการนี้

ประสบการณ์ ความรู้ ที่ได้จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จะนำไปใช้ในทางที่ดี

กฤษณะ คงวัลย์
อิสรา บุญญานุวัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ยืน ภู่วรรณ “เทคโนโลยีฮาร์ดแวร์ไอพีเอ็มพีซี” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ.2533
- [2] พันจันทร์ ชนวัฒน์เสถียรและคงศักดิ์ ลอรัตนเรืองกิจ “ไมโครโปรเซสเซอร์ทฤษฎีและการประยุกต์ใช้งาน” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ.2537
- [3] สุนทร วิทูรพจน์ “การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ.2533
- [4] ผศ.สมยศ จุณณะปิยะ “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51” คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2541
- [5] ชีรวัฒน์ ประกอบผล “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์” สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น) พ.ศ.2540
- [6] กิตติ องค์กรักษ์ “แนะนำแอสเซบลี” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ.2537
- [7] โยธิน เปรมปราณรัชต์ “ระบบเซอร์และอิเล็กทรอนิกส์คอนโทรลเลอร์” ภาควิชาวิศวกรรมระบบควบคุมคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.

โปรแกรมควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D_ISEC	EQU	-15361	
STATUS	EQU	30H	
AS0	EQU	38H	
AS1	EQU	39H	
STOP_FLAG	BIT	5	
A12_FLAG	BIT	1	;A12_FLAG = 1 A1>A2 ;A12_FLAG = 0 A1<A2
A34_FLAG	BIT	2	;A34_FLAG = 1 A3>A4 ;A34_FLAG = 0 A3<A4
DARK_FLAG	BIT	3	
UP_DOWN	BIT	4	

SW1	BIT	P1.2	
SW2	BIT	P1.3	
SW3	BIT	P1.4	
SW4	BIT	P1.5	
DELTA	EQU	6	
DELTA1	EQU	7	
DARK	EQU	090H	;Contant

```

ORG 0
SJMP MAIN
ORG 0003H
LJMP INTO_INT

MAIN:
MOV SP,#128-32
;CLR STOP_FLAG
LCALL M_STOP
LCALL INIT_RS232
MOV TMOD,#21H
MOV A,#'P'
LCALL SEND

LM:
;
;
;
;
LCALL REC
LCALL SEND
SJMP LM

MOV A,#0AH
LCALL SEND
SETB EA
SETB IT0
SETB EX0
MOV R3,#5 ;Equ 0 When A1>A2
MOV R4,#5
MOV R5,#5
MOV R6,#5
D_LIGH0_0: ;
LCALL DELAY_1S
LCALL ADC_READ

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JNB ADC_ERFLAG,DDP1
MOV A,#'E'
LCALL SEND
SJMP D_LIGH0_0
CPAA:
DDP1:
LCALL SHOW
LCALL CK_DARK
JNB DARK_FLAG,CPAA

:
CPA1_1:
JNB SW2,TURN1
LCALL M_DOWN
CPA1_0:
JB SW2,TNN1
LCALL M_STOP
LCALL DELAY_1S
JNB SW2,TURN1
TNN1:
LCALL M_DOWN
LCALL CK_DARK
MOV A,#'D'
LCALL SEND
JB DARK_FLAG,CPA1_0
LCALL M_STOP
MOV R6,#5
CPA1_2:
LCALL CK_DARK
JB DARK_FLAG,CPA1_1
DJNZ R6,CPA1_2
SJMP CPA

TURN1:
MOV A,#'X'
LCALL SEND
CLR EA
TURN1_2:
LCALL M_UP
JNB SW2,$
LCALL DELAY_1S
SETB EA
TURN1_0:
LCALL M_UP
JB SW1,TUR
LCALL M_STOP
LCALL DELAY_1S
JNB SW1,TURN2
TUR:
LCALL M_UP
MOV A,#'U'
LCALL SEND
LCALL CK_DARK
JB DARK_FLAG,TURN1_0
LCALL M_STOP
MOV R6,#5
TURN1_3:
LCALL CK_DARK
JB DARK_FLAG,TURN1_0
DJNZ R6,TURN1_3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เรียนการสอนในชั้นเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                SJMP CPA
TURN2:         ;
                ;
                ; MOV A,#'Z'
                ; LCALL SEND
                ; LCALL M_DOWN
                ; LCALL DELAY_1S

                MOV R6,#30
TT1:          LCALL M_DOWN
                MOV A,#'D'
                LCALL SEND
                LCALL DELAY_1S
                DJNZ R6,TT1
                LCALL M_STOP
TT2:          MOV R6,#20
TT3:          LCALL CK_DARK
                JB DARK_FLAG,TT2
                DJNZ R6,TT3

                ;
CPA:          LCALL CP_A1A2
                CJNE A,#DELTA,D_LIGH1
                SJMP D_LIGH0
D_LIGH1:     JC D_LIGH0 ;If A<#DELTA
C=1
                JNB A12_FLAG,D_LIGH2 ;If A1 > A2 Then A12_FLAG
=1
                DJNZ R3,D_LIGH0
                LCALL TURN_UP
                LJMP D_LIGH0_0
D_LIGH2:     DJNZ R4,D_LIGH0
                LCALL TURN_DOWN
                LJMP D_LIGH0_0

                ;
D_LIGH0:     ; LJMP D_LIGH0_0
                LCALL ADC_READ
                LCALL SHOW
                LCALL CP_A3A4
                CJNE A,#DELTA1,D_LL1
                LJMP D_LIGH0_0
D_LL1:      JC D_LIGH0A
                JNB A34_FLAG,D_LL2
                DJNZ R5,D_LIGH0A
                LCALL TURN_LI
                LJMP D_LIGH0_0

                ;
D_LL2:      DJNZ R6,D_LIGH0A
                LCALL TURN_RI
D_LIGH0A:   LJMP D_LIGH0_0
;-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
LLP:          LCALL REC
              LCALL SEND
              CJNE A,#'W',LR1
              LCALL M_UP
              SJMP LLP
LR1:         CJNE A,#'Z',LR2
              LCALL M_DOWN
              SJMP LLP
LR2:         CJNE A,#'A',LR3
              LCALL M_LI
              SJMP LLP
LR3:         CJNE A,#'D',LR4
              LCALL M_RI
              SJMP LLP
LR4:         CJNE A,#'S',LLP
              LCALL M_STOP
              SJMP LLP
;
SHOW:        PUSH 0
              MOV A,#0DH
              LCALL SEND
;
ADC_RE:      MOV R0,#ADC_BUFF+1
              MOV A,@R0
              INC R0
              LCALL HEX_ASCII
              MOV A,AS1
              LCALL SEND
              MOV A,AS0
              LCALL SEND
              MOV A,#' '
              LCALL SEND
              CJNE R0,#ADC_BUFF+5,ADC_RE
              POP 0
              RET
;
TURN_UP:     JNB SW1,UP2
              LCALL M_UP
UP0:         JNB SW1,UP2
              LCALL M_UP
              MOV A,#'U'
              LCALL SEND
              LCALL ADC_READ
              LCALL CP_A1A2
              CJNE A,#DELTA,UP1
              SJMP UP0
UP1:         JNC UP0
              LCALL DELAY_1S
UP2:         LCALL M_STOP

```

```

;-----
                                RET
TURN_DOWN:                      JNB SW2,DOWN2
                                LCALL M_DOWN
DOWN0:                          JNB SW2,DOWN2
                                LCALL M_DOWN
                                MOV A,#'D'
                                LCALL SEND
                                LCALL ADC_READ
                                LCALL CP_A1A2
                                CJNE A,#DELTA,DOWN1
                                SJMP DOWN0
DOWN1:                          JNC DOWN0
                                LCALL DELAY1
DOWN2:                          LCALL M_STOP
                                RET

```

```

;-----
TURN_RI:                        JNB SW3,RI2
                                LCALL M_RI
RI0:                            JNB SW3,RI2
                                LCALL M_RI
                                MOV A,#'L'
                                LCALL SEND
                                LCALL ADC_READ
                                LCALL CP_A3A4
                                CJNE A,#DELTA1,RI1
                                SJMP RI0
RI1:                            JNC RI0
                                LCALL DELAY1
RI2:                            LCALL M_STOP
                                RET

```

```

;-----
DELAY1:    PUSH 0
            PUSH 1
            MOV R0,#200
DYY:      MOV R1,#0
            DJNZ R1,$
            DJNZ R0,DYY
            POP 1
            POP 0
            RET

```

```

;-----
TURN_LI:  JNB SW4,LI2
            LCALL M_LI
LI0:      JNB SW4,LI2
            LCALL M_LI
            MOV A,#'R'
            LCALL SEND
            LCALL ADC_READ

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                LCALL CP_A3A4
                CJNE A,#DELTA1,LI1
                SJMP LI0
LI1:            JNC LI0
LI2:            LCALL M_STOP
                RET

;-----
FINE_LIGHT:    ACALL M_UP
                JB SW1,$
                ACALL M_STOP
                ACALL DELAY_1S
                ACALL M_DOWN
                MOV R7,#30
FINE1:         ACALL DELAY_1S
                DJNZ R7,FINE1
                LCALL M_STOP
                RET

;-----
;DARK_FLAG = 1 (DARK) =0 (NOT DARK)
CK_DARK:       PUSH 0
                PUSH 1
                MOV R0,#2
DARK0:         MOV R1,#1
                MOV A,ADC_BUFF+1
                CLR C
                SUBB A,#DARK
                JC DARK1
DARK1_0:      MOV A,ADC_BUFF+2
                CLR C
                SUBB A,#DARK
                JC DARK2
DARK2_0:      MOV A,ADC_BUFF+3
                CLR C
                SUBB A,#DARK
                JC DARK3
DARK3_0:      MOV A,ADC_BUFF+4
                CLR C
                SUBB A,#DARK
                JC DARK4
DARK4_0:      LCALL DELAY_1S          ;Delay 1 second
                LCALL ADC_READ
                DJNZ R0,DARK0
                SETB DARK_FLAG
                POP 1
                POP 0
                RET
DARK1:        DJNZ R1,DARK1_0
                SJMP NOT_DARK
DARK2:        DJNZ R1,DARK2_0

```

เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DARK3:      SJMP NOT_DARK
            DJNZ R1,DARK3_0
            SJMP NOT_DARK
DARK4:      DJNZ R1,DARK4_0
NOT_DARK:   CLR DARK_FLAG
            POP 1
            POP 0
            RET

```

```

;-----
;A12_FLAG = 1 A1>A2
;A12_FLAG = 0 A1<A2
;OUTPUT = A =DIF OF A1,A2
CP_A1A2:    MOV A,ADC_BUFF+1
            CLR C
            SUBB A,ADC_BUFF+2
            JC   CPA1
            SETB A12_FLAG
            RET
CPA1:       CPL A
            INC A
            CLR A12_FLAG
            RET

```

```

;-----
;A34_FLAG = 1 A3>A4
;A34_FLAG = 0 A3<A4
;OUTPUT = A =DIF OF A3,A4
CP_A3A4:    MOV A,ADC_BUFF+3
            CLR C
            SUBB A,ADC_BUFF+4
            JC   CPA3
            SETB A34_FLAG
            RET
CPA3:       CPL A
            INC A
            CLR A34_FLAG
            RET

```

```

;-----
DELAY_1S:   PUSH 0
            MOV R0,#100
DEY1:      MOV TH0,#HIGH D_1SEC
            MOV TL0,#LOW D_1SEC
            SETB TR0
            JNB TF0,$
            CLR TF0
            CLR TR0
            DJNZ R0,DEY1
            POP 0
            RET

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY_1M:    PUSH 0
             MOV R0,#60
DEM1:        ACALL DELAY_1S
             DJNZ R0,DEM1
             POP 0
             RET
;-----
M_UP:        SETB P0.0
             CLR P0.1
             RET
;-----
M_DOWN:      CLR P0.0
             SETB P0.1
             RET
;-----
M_RI:        SETB P0.2
             CLR P0.3
             RET
;-----
M_LI:        CLR P0.2
             SETB P0.3
             RET
;-----
M_STOP:      MOV P0,#0
             RET
;-----
;INPUT A
;OUTPUT AS1,AS0
IEX_ASCII:  PUSH ACC
             ANL A,#0F0H
             SWAP A
             LCALL ASCODE
             MOV AS1,A
             POP ACC
             ANL A,#0FH
             LCALL ASCODE
             MOV AS0,A
             RET

ASCODE:      INC A
             MOVC A,@A+PC
             RET
             DB '0123456789ABCDEF'
;-----Interrupt Service Rutine-
INT0_INT:    ACALL M_STOP
             MOV A,#'A'
             LCALL SEND
             CLR IE0
             RETI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INCL "RS232.ASM"
INCL "PCF8591_1.ASM"
```

```
-----
DC_BUFF      EQU      32H      ;Used 32H-37H
ADC_ERFLAG   BIT      0H
SDA          BIT      P1.0
SCL         BIT      P1.1
-----
```

```
;Reg A
```

```
;Output ADC_BUFF+1 = Output Chennel 1
;      ADC_BUFF+2 = Output Chennel 2
;      ADC_BUFF+3 = Output Chennel 3
;      ADC_BUFF+4 = Output Chennel 4
```

```
;ADC_ERFLAG 0=OK 1=ERROR
```

```
ADC_READ:    PUSH 0
             CLR SDA                      ;Start
             NOP
             NOP
             CLR SCL
             NOP
             MOV A,#10010000B ;Address Byte
             LCALL ADC_WR
             JB ADC_ERFLAG,ADC_EX
             MOV A,#00000100B ;Control Byte
             LCALL ADC_WR
             JB ADC_ERFLAG,ADC_EX
             CLR SDA
             NOP
             NOP
             SETB SCL                      ;Stop
             NOP
             NOP
             SETB SDA
             NOP
             NOP
             CLR SDA                      ;Start
             NOP
             NOP
             CLR SCL
             MOV A,#10010001B ;Address Byte
             LCALL ADC_WR
             JB ADC_ERFLAG,ADC_EX
             MOV R0,#ADC_BUFF
             SETB SDA
ADC1:        LCALL ADC_RD
             MOV @R0,A
             INC R0
             CJNE R0,#ADC_BUFF+4,ADC1
             ;
             MOV R0,#8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADC2:          LCALL SETB_SCL
               MOV C,SDA
               RLC A
               LCALL CLR_SCL
               DJNZ R0,ADC2
               MOV ADC_BUFF+4,A
               LCALL SETB_SCL
               LCALL CLR_SCL
               CLR SDA
               NOP
               NOP
               SETB SCL
               NOP
               NOP
               SETB SDA
               SETB SCL
               SETB SDA
               POP 0
               RET
;-----
ADC_WR:        PUSH 0
               MOV R0,#8
               RLC A
               MOV SDA,C
               LCALL SETB_SCL
               LCALL CLR_SCL
               DJNZ R0,ADC_WR1
               SETB SDA
               LCALL SETB_SCL
               MOV C,SDA
               LCALL CLR_SCL
               JC ADC_WR2
               CLR ADC_ERFLAG
               POP 0
               RET
ADC_WR1:       ;Chack Acknowledge
               LCALL SETB_SCL
               MOV C,SDA
               LCALL CLR_SCL
               JC ADC_WR2
               CLR ADC_ERFLAG
               POP 0
               RET
ADC_WR2:       SETB ADC_ERFLAG
               POP 0
               RET
;-----
ADC_RD:        PUSH 0
               MOV R0,#8
ADC_RD1:       LCALL SETB_SCL
               MOV C,SDA
               RLC A
               LCALL CLR_SCL
               DJNZ R0,ADC_RD1
               CLR SDA
               LCALL SETB_SCL
               ;Send Acknowledge

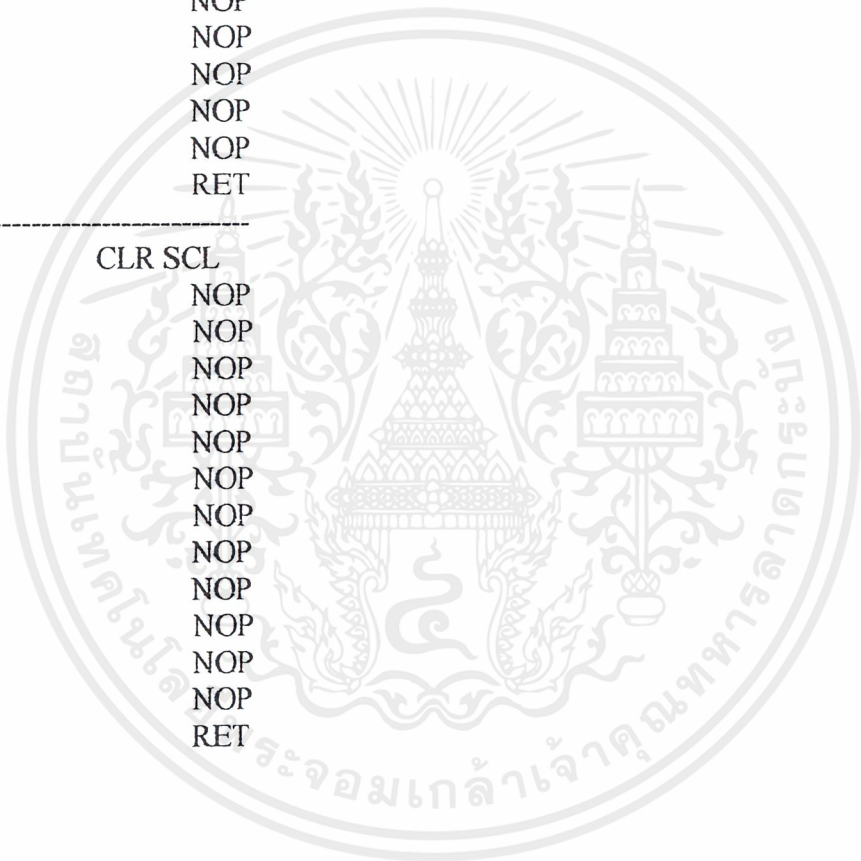
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL CLR_SCL
SETB SDA
POP 0
RET
```

```
-----
SETB_SCL:    SETB SCL
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              RET
```

```
-----
CLR_SCL:     CLR SCL
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              NOP
              RET
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

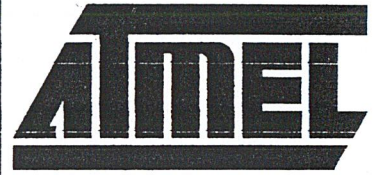
- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
 - SPI Serial Interface for Program Downloading
 - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
- 4.0V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 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 interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



8-Bit Microcontroller with 8K Bytes Flash

AT89S8252

0401D-A-12/97



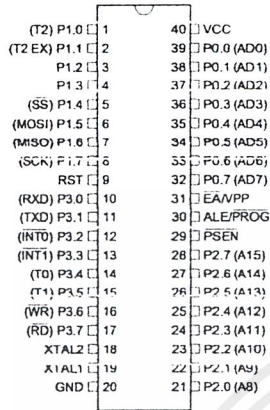
4-105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

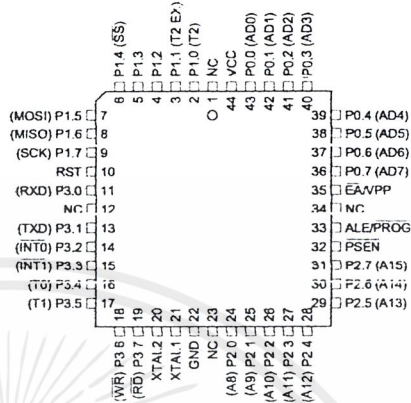


Pin Configurations

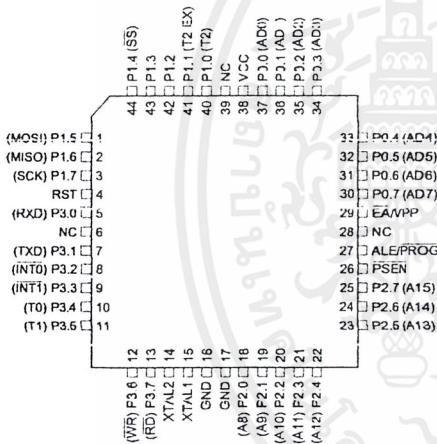
PDIP



PLCC



PQFP/TQFP



Pin Description

V_{CC}
Supply voltage.

GND
Ground.

Port 0
Port 0 is an 8-bit open drain bidirectional 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.

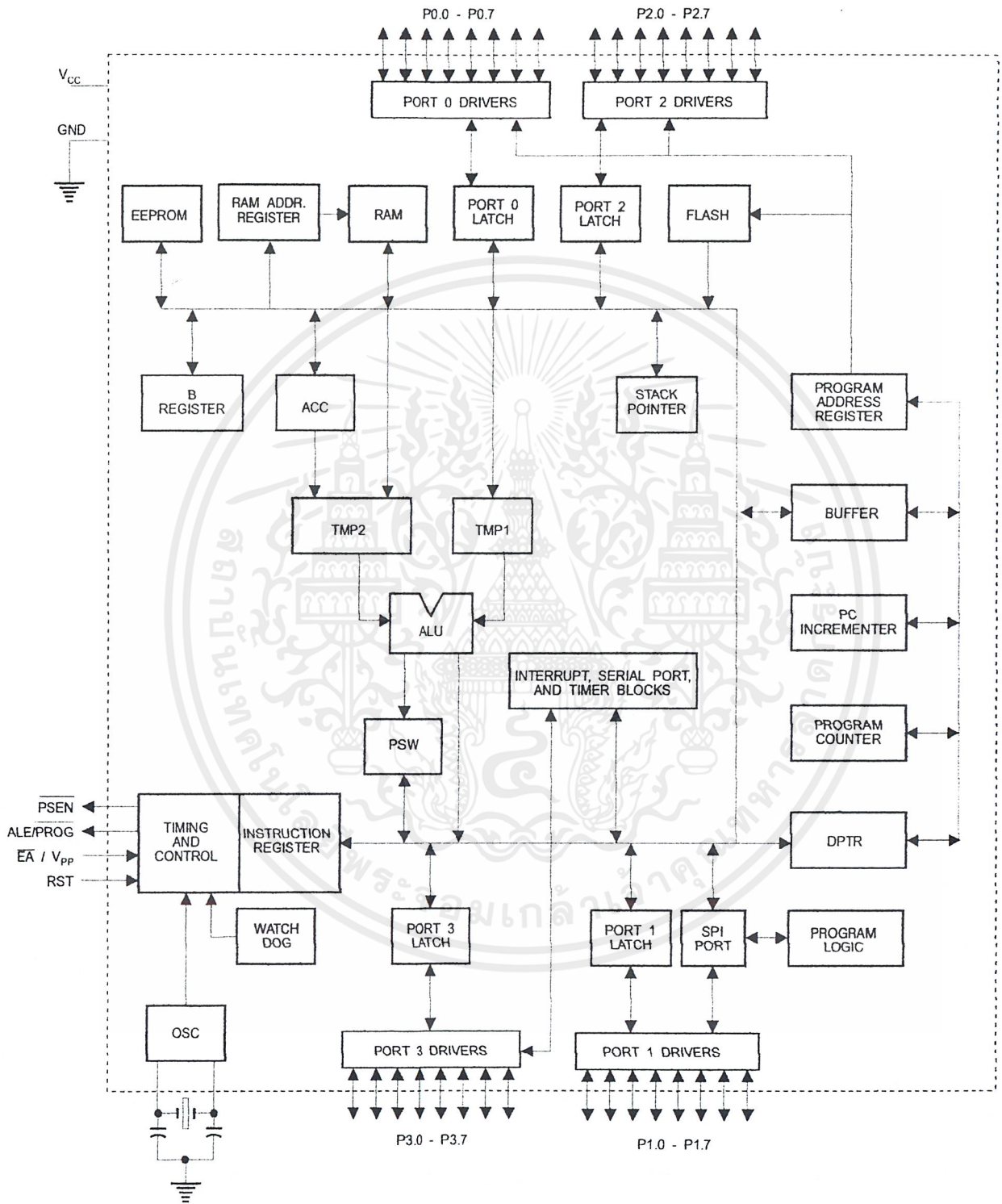
Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

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 bidirectional 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_{IH}) because of the internal pullups.

Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	\overline{SS} (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

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

Port 2

Port 2 is an 8-bit bidirectional 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, Port 2 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 bidirectional 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 AT89S8252, as shown in the following table.

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

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

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 is an 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.

PSEN

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

When the AT89S8252 is executing code from external program memory, \overline{PSEN} is activated twice each machine cycle, except that two \overline{PSEN} activations are skipped during each access to external data memory.

\overline{EA}/V_{PP}

External Access Enable. \overline{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{EA} will be internally latched on reset.

\overline{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 when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Table 1. AT89S8252 SFR Map and Reset Values

0F8H								0FFH	
0F0H	B 00000000							0F7H	
0E8H								0EFH	
0E0H	ACC 00000000							0E7H	
0D8H								0DFH	
0D0H	PSW 00000000					SPCR 000001XX		0D7H	
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH	
0C0H								0C7H	
0B8H	IP XX000000							0BFH	
0B0H	P3 11111111							0B7H	
0A8H	IE 0X000000		SPSR 00XXXXXX					0AFH	
0A0H	P2 11111111							0A7H	
98H	SCON 00000000	SBUF XXXXXXXX						9FH	
90H	P1 11111111						WMCON 00000010	97H	
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH	
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H



User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

SPI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H						Reset Value = 0000 0000B		
Bit Addressable								
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

Power Off Flag The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H				Reset Value = 0000 0010B				
	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.



Table 4. SPCR—SPI Control Register

SPCR Address = D5H		Reset Value = 0000 01XXB						
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
Bit	7	6	5	4	3	2	1	0

Symbol	Function															
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.															
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects \overline{SS} , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.															
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.															
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.															
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.															
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.															
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, F_{OSC} , is as follows: <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>SCK = F_{OSC} divided by</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>0</td> <td>1</td> <td>16</td> </tr> <tr> <td>1</td> <td>0</td> <td>64</td> </tr> <tr> <td>1</td> <td>1</td> <td>128</td> </tr> </tbody> </table>	SPR1	SPR0	SCK = F_{OSC} divided by	0	0	4	0	1	16	1	0	64	1	1	128
SPR1	SPR0	SCK = F_{OSC} divided by														
0	0	4														
0	1	16														
1	0	64														
1	1	128														

Table 5. SPSR—SPI Status Register

SPSR Address = AAH		Reset Value = 00XX XXXXB					
	SPIF	WCOL	—	—	—	—	—
Bit	7	6	5	4	3	2	1

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

Table 6. SPDR—SPI Data Register

SPDR Address = 86H				Reset Value = unchanged				
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Bit	7	6	5	4	3	2	1	0

Data Memory—EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at V_{CC} = 5V) are within ±30% of the nominal.

The WDT is disabled by Power-on Reset and during Power Down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms



Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit $C/\overline{T}2$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Figure 1. Timer 2 in Capture Mode

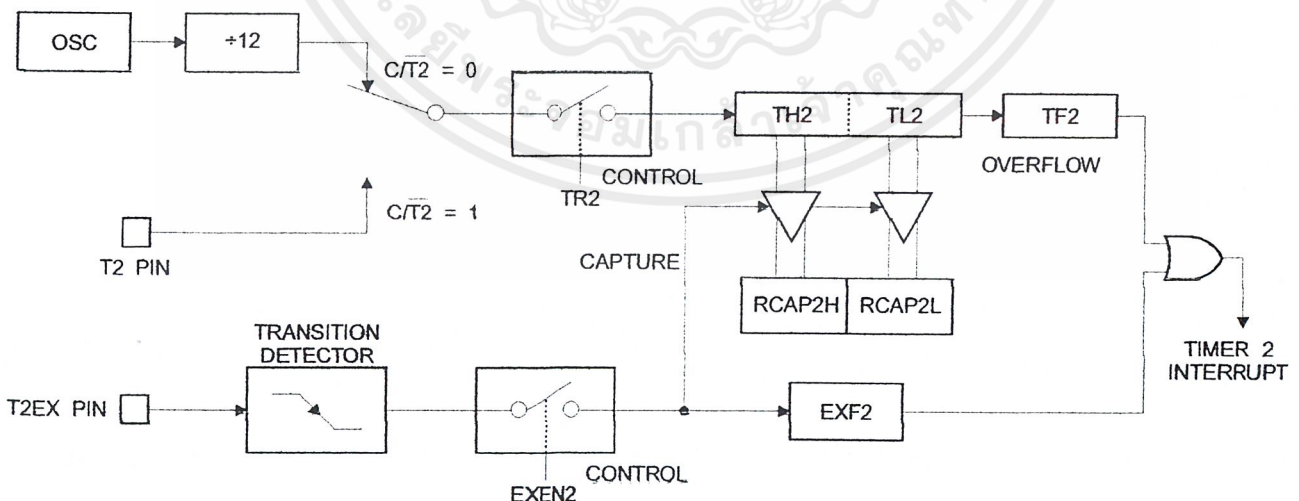


Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-Reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to

0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in

RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 in Auto Reload Mode (DCEN = 0)

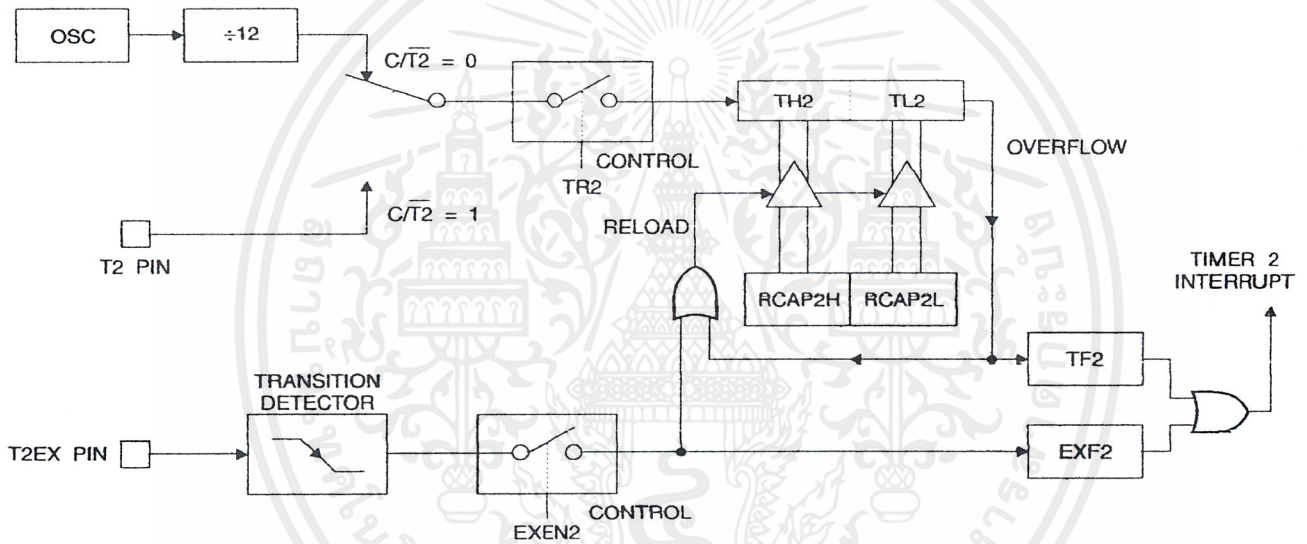


Table 9. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H							Reset Value = XXXX XX00B	
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	T2OE	DCEN
Symbol	Function							
—	Not implemented, reserved for future use.							
T2OE	Timer 2 Output Enable bit.							
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.							



8-bit A/D and D/A converter

PCF8591

1 FEATURES

- Single power supply
- Operating supply voltage 2.5 V to 6 V
- Low standby current
- Serial input/output via I²C-bus
- Address by 3 hardware address pins
- Sampling rate given by I²C-bus speed
- 4 analog inputs programmable as single-ended or differential inputs
- Auto-incremented channel selection
- Analog voltage range from V_{SS} to V_{DD}
- On-chip track and hold circuit
- 8-bit successive approximation A/D conversion
- Multiplying DAC with one analog output.

2 APPLICATIONS

- Closed loop control systems
- Low power converter for remote data acquisition
- Battery operated equipment
- Acquisition of analog values in automotive, audio and TV applications.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCA8591P	DIP16	plastic dual in-line package; 16 leads (300 ml); long body	SOT38-1
PCA8591T	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1



3 GENERAL DESCRIPTION

The PCF8591 is a single-chip, single-supply low power 8-bit CMOS data acquisition device with four analog inputs, one analog output and a serial I²C-bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I²C-bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional I²C-bus.

The functions of the device include analog input multiplexing, on-chip track and hold function, 8-bit analog-to-digital conversion and an 8-bit digital-to-analog conversion. The maximum conversion rate is given by the maximum speed of the I²C-bus.

8-bit A/D and D/A converter

PCF8591

6 PINNING

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

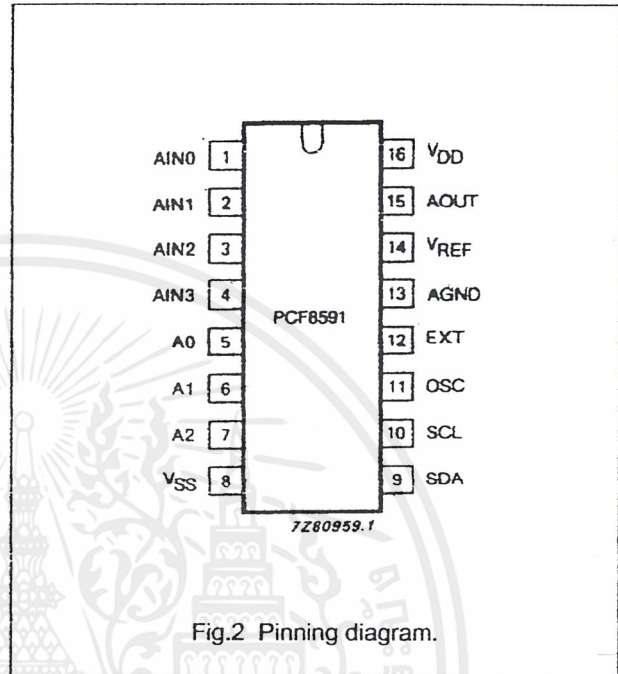


Fig.2 Pinning diagram.

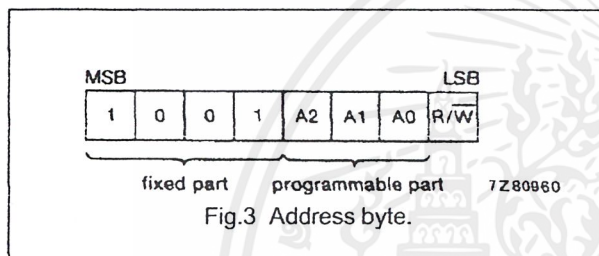
8-bit A/D and D/A converter

PCF8591

7 FUNCTIONAL DESCRIPTION

7.1 Addressing

Each PCF8591 device in an I²C-bus system is activated by sending a valid address to the device. The address consists of a fixed part and a programmable part. The programmable part must be set according to the address pins A0, A1 and A2. The address always has to be sent as the first byte after the start condition in the I²C-bus protocol. The last bit of the address byte is the read/write-bit which sets the direction of the following data transfer (see Figs 3, 15 and 16).



7.2 Control byte

The second byte sent to a PCF8591 device will be stored in its control register and is required to control the device function.

The upper nibble of the control register is used for enabling the analog output, and for programming the analog inputs as single-ended or differential inputs. The lower nibble selects one of the analog input channels defined by the upper nibble (see Fig.4). If the auto-increment flag is set the channel number is incremented automatically after each A/D conversion.

If the auto-increment mode is desired in applications where the internal oscillator is used, the analog output enable flag in the control byte (bit 6) should be set. This allows the internal oscillator to run continuously, thereby preventing conversion errors resulting from oscillator start-up delay. The analog output enable flag may be reset at other times to reduce quiescent power consumption.

The selection of a non-existing input channel results in the highest available channel number being allocated.

Therefore, if the auto-increment flag is set, the next selected channel will be always channel 0. The most significant bits of both nibbles are reserved for future functions and have to be set to 0. After a Power-on reset condition all bits of the control register are reset to 0.

The D/A converter and the oscillator are disabled for power saving. The analog output is switched to a high-impedance state.

8-bit A/D and D/A converter

PCF8591

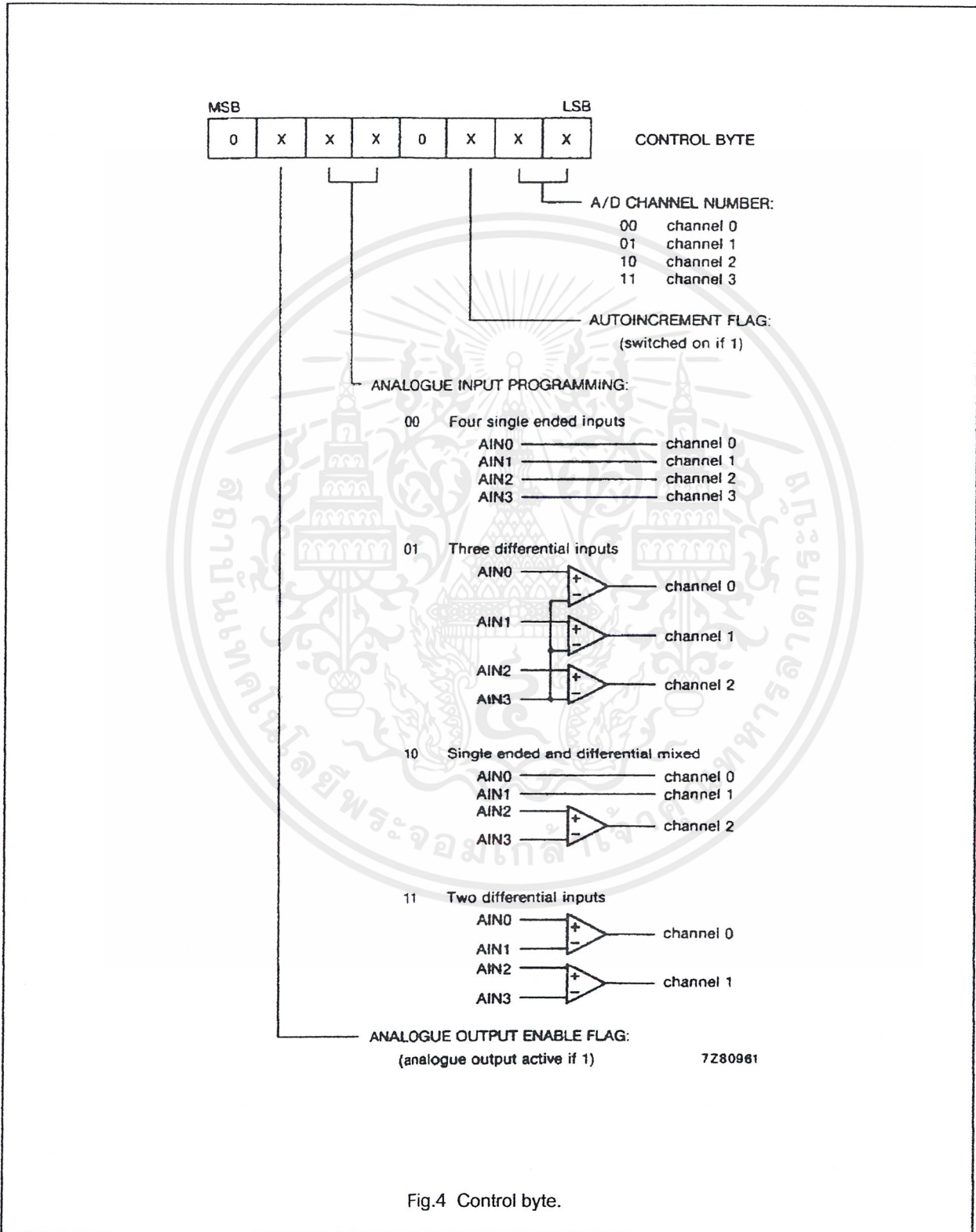


Fig.4 Control byte.

8-bit A/D and D/A converter

PCF8591

8 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for bidirectional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor. Data transfer may be initiated only when the bus is not busy.

8.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as a control signal.

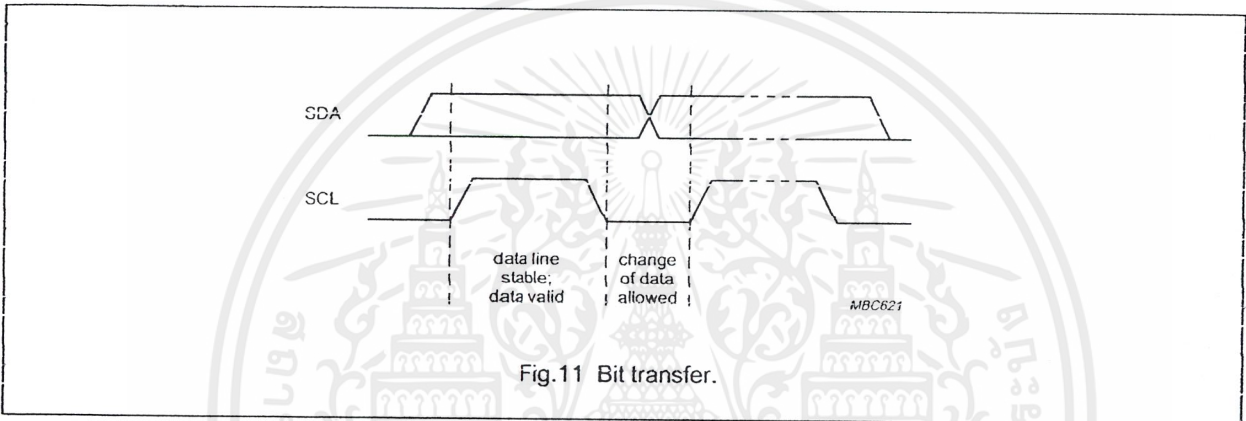


Fig.11 Bit transfer.

8.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH, is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH, is defined as the stop condition (P).

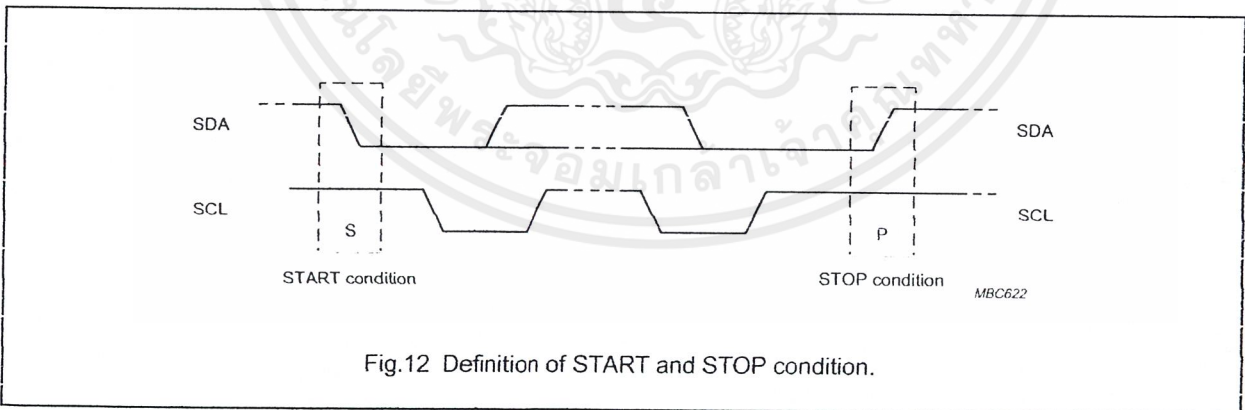


Fig.12 Definition of START and STOP condition.

8-bit A/D and D/A converter

PCF8591

8.3 System configuration

A device generating a message is a 'transmitter', a device receiving a message is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves'.

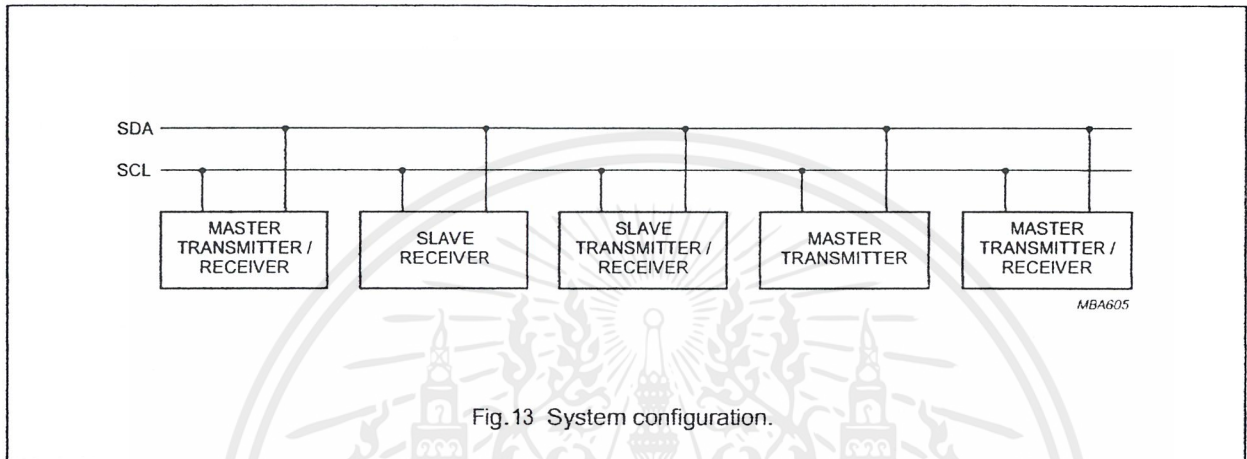


Fig.13 System configuration.

8.4 Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each data byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master also generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. A master receiver must signal an end of data to the transmitter by **not** generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

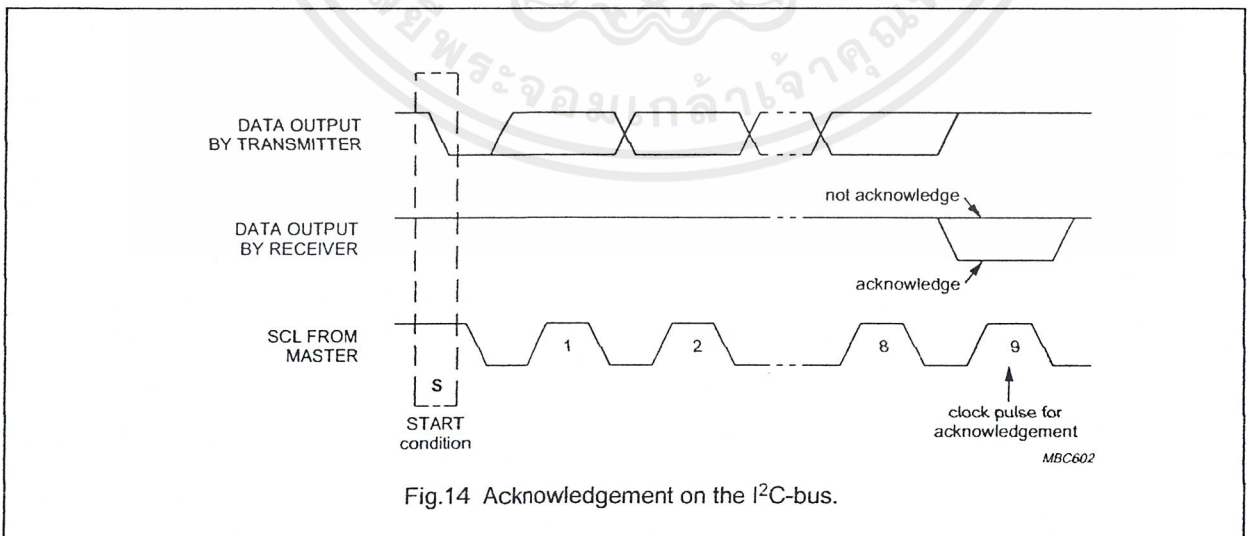


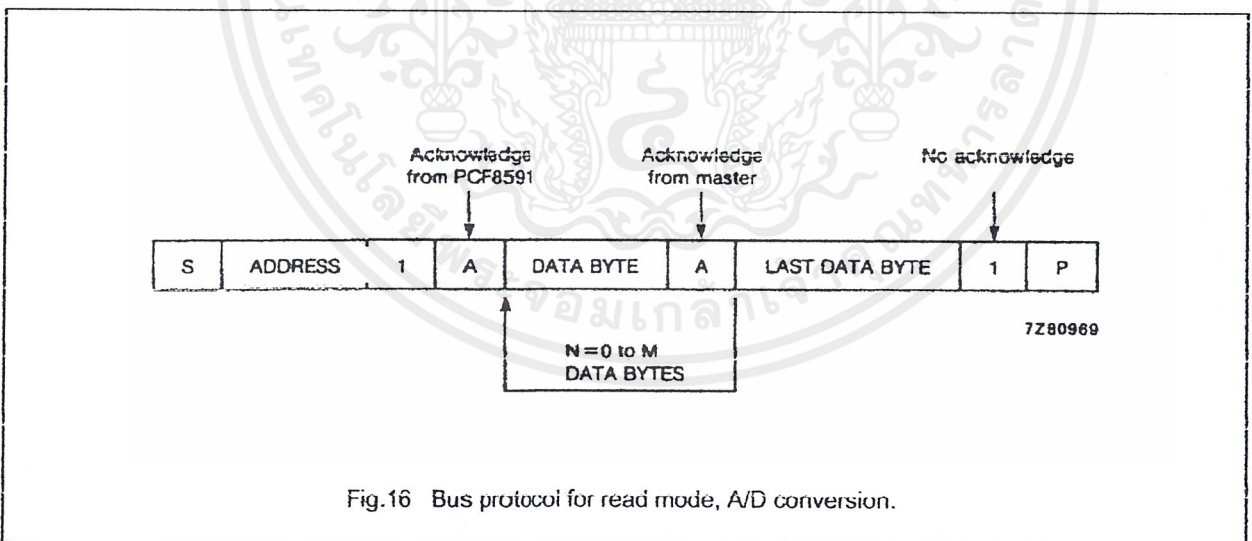
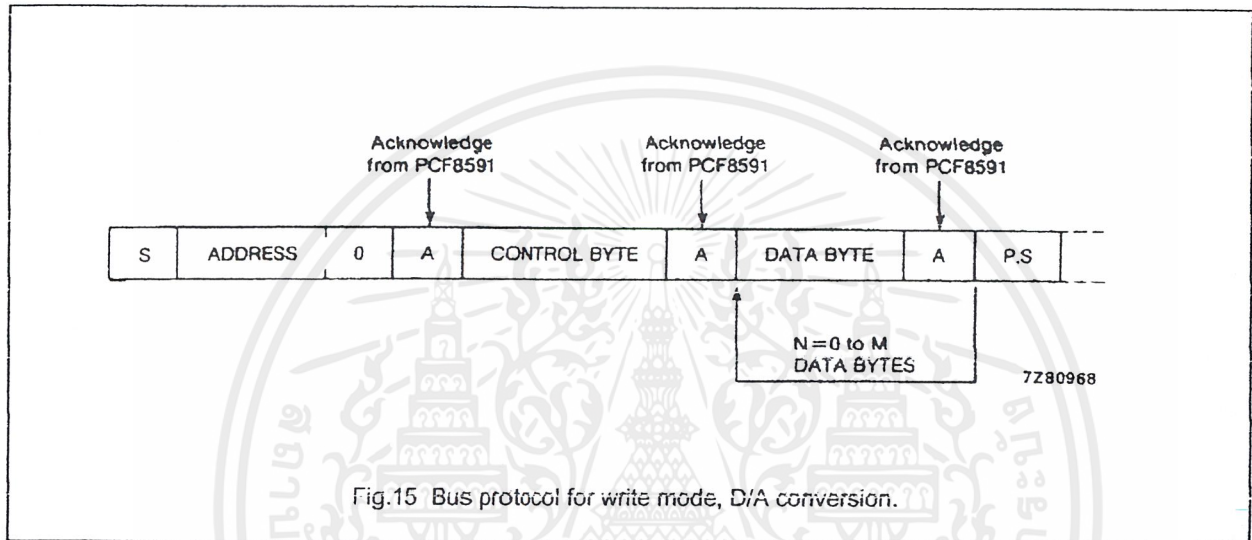
Fig.14 Acknowledgement on the I²C-bus.

8-bit A/D and D/A converter

PCF8591

8.5 I²C-bus protocol

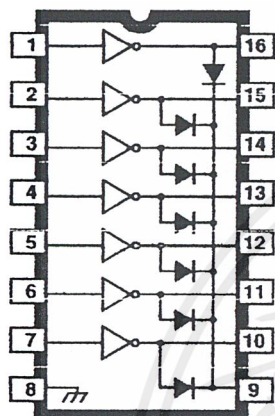
After a start condition a valid hardware address has to be sent to a PCF8591 device. The read/write bit defines the direction of the following single or multiple byte data transfer. For the format and the timing of the start condition (S), the stop condition (P) and the acknowledge bit (A) refer to the I²C-bus characteristics. In the write mode a data transfer is terminated by sending either a stop condition or the start condition of the next data transfer.



2003 THRU 2024

Data Sheet
29304F

HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS



Dwg. No. A-9594

Note that the ULN20xxA series (dual in-line package) and ULN20xxL series (small-outline IC package) are electrically identical and share a common terminal number assignment.

ABSOLUTE MAXIMUM RATINGS

Output Voltage, V_{CE}	
(ULN200xA and ULN200xL)	50 V
(ULN202xA and ULN202xL)	95 V
Input Voltage, V_{IN}	30 V
Continuous Output Current,	
I_C	500 mA
Continuous Input Current, I_{IN}	25 mA
Power Dissipation, P_D	
(one Darlington pair)	1.0 W
(total package)	See Graph
Operating Temperature Range,	
T_A	-20°C to +85°C
Storage Temperature Range,	
T_S	-55°C to +150°C

Ideally suited for interfacing between low-level logic circuitry and multiple peripheral power loads, the Series ULN20xxA/L high-voltage, high-current Darlington arrays feature continuous load current ratings to 500 mA for each of the seven drivers. At an appropriate duty cycle depending on ambient temperature and number of drivers turned ON simultaneously, typical power loads totaling over 230 W (350 mA x 7, 95 V) can be controlled. Typical loads include relays, solenoids, stepping motors, magnetic print hammers, multiplexed LED and incandescent displays, and heaters. All devices feature open-collector outputs with integral clamp diodes.

The ULN2003A/L and ULN2023A/L have series input resistors selected for operation directly with 5 V TTL or CMOS. These devices will handle numerous interface needs — particularly those beyond the capabilities of standard logic buffers.

The ULN2004A/L and ULN2024A/L have series input resistors for operation directly from 6 to 15 V CMOS or PMOS logic outputs.

The ULN2003A/L and ULN2004A/L are the standard Darlington arrays. The outputs are capable of sinking 500 mA and will withstand at least 50 V in the OFF state. Outputs may be paralleled for higher load current capability. The ULN2023A/L and ULN2024A/L will withstand 95 V in the OFF state.

These Darlington arrays are furnished in 16-pin dual in-line plastic packages (suffix "A") and 16-lead surface-mountable SOICs (suffix "L"). All devices are pinned with outputs opposite inputs to facilitate ease of circuit board layout. All devices are rated for operation over the temperature range of -20°C to +85°C. Most (see matrix, next page) are also available for operation to -40°C; to order, change the prefix from "ULN" to "ULQ".

FEATURES

- TTL, DTL, PMOS, or CMOS-Compatible Inputs
- Output Current to 500 mA
- Output Voltage to 95 V
- Transient-Protected Outputs
- Dual In-Line Plastic Package or Small-Outline IC Package

x = digit to identify specific device. Characteristic shown applies to family of devices with remaining digits as shown. See matrix on next page.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะภายในเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2003 THRU 2024
HIGH-VOLTAGE,
HIGH-CURRENT
DARLINGTON ARRAYS**

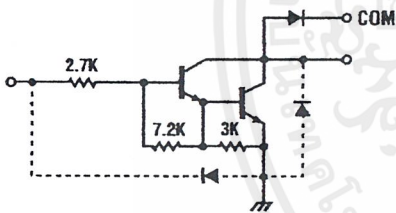
DEVICE PART NUMBER DESIGNATION

$V_{CE(MAX)}$	50 V	95 V
$I_{C(MAX)}$	500 mA	500 mA
Logic	Part Number	
5V TTL, CMOS	ULN2003A* ULN2003L*	ULN2023A* ULN2023L
6-15 V CMOS, PMOS	ULN2004A* ULN2004L*	ULN2024A ULN2024L

* Also available for operation between -40°C and +85°C. To order, change prefix from "ULN" to "ULQ".

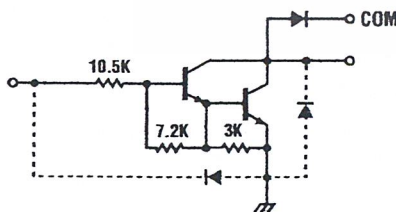
PARTIAL SCHEMATICS

ULN20x3A/L (Each Driver)

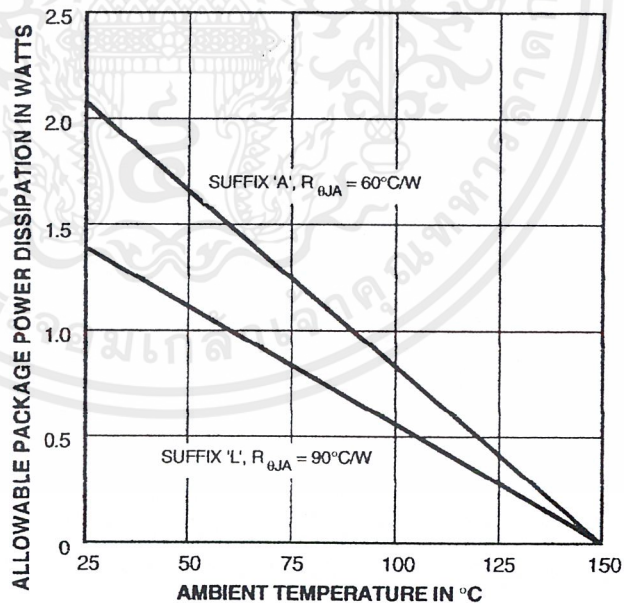


Dwg. No. A-9651

ULN20x4A/L (Each Driver)



Dwg. No. A-9896A



Dwg. GP-006A

X = Digit to identify specific device. Specification shown applies to family of devices with remaining digits as shown. See matrix above.



115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000
Copyright © 1974, 1998 Allegro MicroSystems, Inc.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับ Allegro MicroSystems, Inc. การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2003 THRU 2024
HIGH-VOLTAGE,
HIGH-CURRENT
DARLINGTON ARRAYS

Types ULN2003A, ULN2003L, ULN2004A, and ULN2004L
ELECTRICAL CHARACTERISTICS at +25°C (unless otherwise noted).

Characteristic	Symbol	Test Fig.	Applicable Devices	Test Conditions	Limits			
					Min.	Typ.	Max.	Units
Output Leakage Current	I_{CEX}	1A	All	$V_{CE} = 50\text{ V}, T_A = 25^\circ\text{C}$	—	< 1	50	μA
				$V_{CE} = 50\text{ V}, T_A = 70^\circ\text{C}$	—	< 1	100	μA
		1B	ULN2004A/L	$V_{CE} = 50\text{ V}, T_A = 70^\circ\text{C}, V_{IN} = 1.0\text{ V}$	—	< 5	500	μA
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	2	All	$I_C = 100\text{ mA}, I_B = 250\text{ }\mu\text{A}$	—	0.9	1.1	V
				$I_C = 200\text{ mA}, I_B = 350\text{ }\mu\text{A}$	—	1.1	1.3	V
				$I_C = 350\text{ mA}, I_B = 500\text{ }\mu\text{A}$	—	1.3	1.6	V
Input Current	$I_{IN(ON)}$	3	ULN2003A/L	$V_{IN} = 3.85\text{ V}$	—	0.93	1.35	mA
			ULN2004A/L	$V_{IN} = 5.0\text{ V}$	—	0.35	0.5	mA
				$V_{IN} = 12\text{ V}$	—	1.0	1.45	mA
	$I_{IN(OFF)}$	4	All	$I_C = 500\text{ }\mu\text{A}, T_A = 70^\circ\text{C}$	50	65	—	μA
Input Voltage	$V_{IN(ON)}$	5	ULN2003A/L	$V_{CE} = 2.0\text{ V}, I_C = 200\text{ mA}$	—	—	2.4	V
				$V_{CE} = 2.0\text{ V}, I_C = 250\text{ mA}$	—	—	2.7	V
				$V_{CE} = 2.0\text{ V}, I_C = 300\text{ mA}$	—	—	3.0	V
		ULN2004A/L	$V_{CE} = 2.0\text{ V}, I_C = 125\text{ mA}$	—	—	5.0	V	
			$V_{CE} = 2.0\text{ V}, I_C = 200\text{ mA}$	—	—	6.0	V	
			$V_{CE} = 2.0\text{ V}, I_C = 275\text{ mA}$	—	—	7.0	V	
			$V_{CE} = 2.0\text{ V}, I_C = 350\text{ mA}$	—	—	8.0	V	
Input Capacitance	C_{IN}	—	All		—	15	25	pF
Turn-On Delay	t_{PLH}	8	All	$0.5 E_{IN}$ to $0.5 E_{OUT}$	—	0.25	1.0	μs
Turn-Off Delay	t_{PHL}	8	All	$0.5 E_{IN}$ to $0.5 E_{OUT}$	—	0.25	1.0	μs
Clamp Diode Leakage Current	I_R	6	All	$V_R = 50\text{ V}, T_A = 25^\circ\text{C}$	—	—	50	μA
				$V_R = 50\text{ V}, T_A = 70^\circ\text{C}$	—	—	100	μA
Clamp Diode Forward Voltage	V_F	7	All	$I_F = 350\text{ mA}$	—	1.7	2.0	V

Complete part number includes suffix to identify package style: A = DIP, L = SOIC.

**2003 THRU 2024
HIGH-VOLTAGE,
HIGH-CURRENT
DARLINGTON ARRAYS**

**Types ULN2023A, ULN2023L, ULN2024A, and ULN2024L
ELECTRICAL CHARACTERISTICS at +25°C (unless otherwise noted).**

Characteristic	Symbol	Test Fig.	Applicable Devices	Test Conditions	Limits			
					Min.	Typ.	Max.	Units
Output Leakage Current	I_{CEX}	1A	All	$V_{CE} = 95\text{ V}, T_A = 25^\circ\text{C}$	—	< 1	50	μA
				$V_{CE} = 95\text{ V}, T_A = 70^\circ\text{C}$	—	< 1	100	μA
		1B	ULN2024A/L	$V_{CE} = 95\text{ V}, T_A = 70^\circ\text{C}, V_{IN} = 1.0\text{ V}$	—	< 5	500	μA
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	2	All	$I_C = 100\text{ mA}, I_B = 250\text{ }\mu\text{A}$	—	0.9	1.1	V
				$I_C = 200\text{ mA}, I_B = 350\text{ }\mu\text{A}$	—	1.1	1.3	V
				$I_C = 350\text{ mA}, I_B = 500\text{ }\mu\text{A}$	—	1.3	1.6	V
Input Current	$I_{IN(ON)}$	3	ULN2023A/L	$V_{IN} = 3.85\text{ V}$	—	0.93	1.35	mA
			ULN2024A/L	$V_{IN} = 5.0\text{ V}$	—	0.35	0.5	mA
				$V_{IN} = 12\text{ V}$	—	1.0	1.45	mA
	$I_{IN(OFF)}$	4	All	$I_C = 500\text{ }\mu\text{A}, T_A = 70^\circ\text{C}$	50	65	—	μA
Input Voltage	$V_{IN(ON)}$	5	ULN2023A/L	$V_{CE} = 2.0\text{ V}, I_C = 200\text{ mA}$	—	—	2.4	V
				$V_{CE} = 2.0\text{ V}, I_C = 250\text{ mA}$	—	—	2.7	V
				$V_{CE} = 2.0\text{ V}, I_C = 300\text{ mA}$	—	—	3.0	V
		ULN2024A/L	$V_{CE} = 2.0\text{ V}, I_C = 125\text{ mA}$	—	—	5.0	V	
			$V_{CE} = 2.0\text{ V}, I_C = 200\text{ mA}$	—	—	6.0	V	
			$V_{CE} = 2.0\text{ V}, I_C = 275\text{ mA}$	—	—	7.0	V	
			$V_{CE} = 2.0\text{ V}, I_C = 350\text{ mA}$	—	—	8.0	V	
Input Capacitance	C_{IN}	—	All		—	15	25	pF
Turn-On Delay	t_{PLH}	8	All	$0.5 E_{IN}$ to $0.5 E_{OUT}$	—	0.25	1.0	μs
Turn-Off Delay	t_{PHL}	8	All	$0.5 E_{IN}$ to $0.5 E_{OUT}$	—	0.25	1.0	μs
Clamp Diode Leakage Current	I_R	6	All	$V_R = 95\text{ V}, T_A = 25^\circ\text{C}$	—	—	50	μA
				$V_R = 95\text{ V}, T_A = 70^\circ\text{C}$	—	—	100	μA
Clamp Diode Forward Voltage	V_F	7	All	$I_F = 350\text{ mA}$	—	1.7	2.0	V

Complete part number includes suffix to identify package style: A = DIP, L = SOIC.



115 Northeast Cutoff, Box 15036
Worcester, Massachusetts 01615-0036 (508) 853-5000

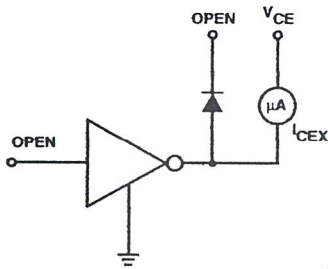
เอเลกรามีเซมิคอนดักเตอร์ที่ผลิตขึ้นเพื่อการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2003 THRU 2024 HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS

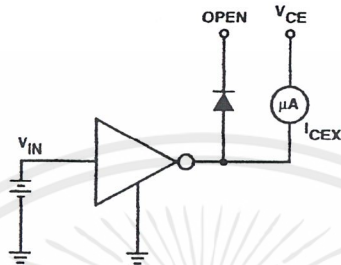
TEST FIGURES

FIGURE 1A



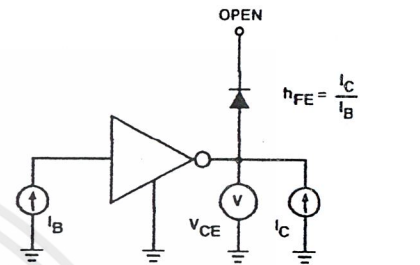
Dwg. No. A-9729A

FIGURE 1B



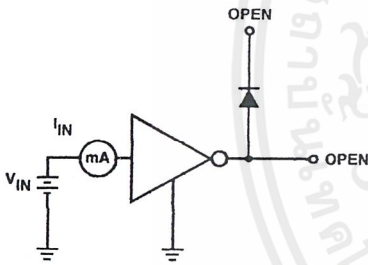
Dwg. No. A-9730A

FIGURE 2



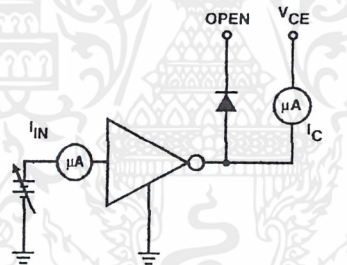
Dwg. No. A-9731A

FIGURE 3



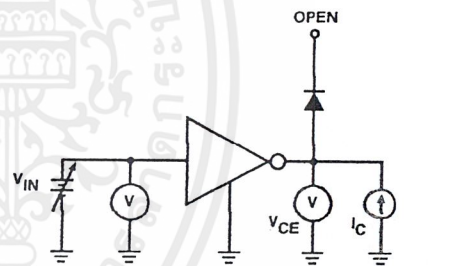
Dwg. No. A-9732A

FIGURE 4



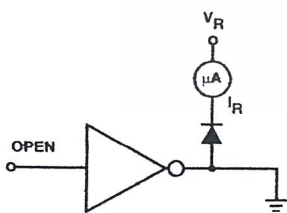
Dwg. No. A-9733A

FIGURE 5



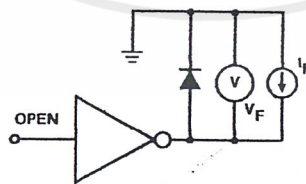
Dwg. No. A-9734A

FIGURE 6



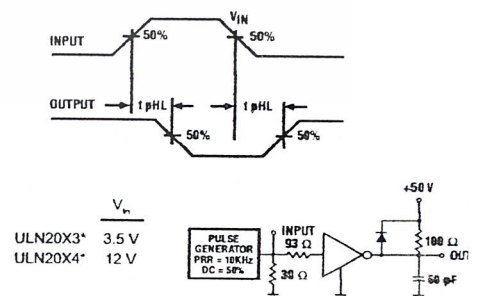
Dwg. No. A-9735A

FIGURE 7



Dwg. No. A-9736A

FIGURE 8



* Complete part number includes a final letter to indicate package.

X = Digit to identify specific device. Specification shown applies to family of devices with remaining digits as shown.

