สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์

ELECTRICAL DEVICES CONTROLLER

T046410

โดย

นายกิตติพงศ์ หนูช่วย

นายศักดิ์ศิริ ดิษฐนวล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีอุตสาหกรรม

ปีการศึกษา 2544

วน.
ก643ก
2544

เลขหมู่.....
เลขทะเบียน 46410
วัน, เดือน, ปี - 1 มิ.ย. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL DEVICES CONTROLLER

BY

Mr.kittipong Knoochuay

Mr.Saksiri Dittanual

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์**
ชื่อนักศึกษา	**นายกิตติพงศ์ หนูช่วย รหัสประจำตัว 43015801**
	นายศักดิ์ศิริ ดิษฐนวล รหัสประจำตัว 43015842
อาจารย์ที่ปรึกษา	**ผศ.อุทัย ศรีธีระวิโรจน์**
	ผศ.ประดิษฐ์ วัชรพิบูลย์
ระดับการศึกษา	**ปริญญาตรีอุตสาหกรรมบัณฑิต**
	สาขาเทคโนโลยีโทรคมนาคม
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ ได้รับอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

(ผศ. อุทัย ศรีธีระวิโรจน์)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Thesis title	ELECTRICAL DEVICES MICROCONTROLLER
Student	Mr.Kittipong Knoochuay
	Mr.Saksiri Dittanual
Advisor	Asst.Prof.Authai Sriteraviroj
Graduat Level	Bachelor'sdegree of industrial tecnology telecommunication
Department	Information Engineerin
Acadamic Year	2001

Accepted by the facuty of Engineering,King Mongkut's Institute of Technology Ladgrabang in fulfillment for the bachelor's degree

Project Report Committee

..... Advisor

(Ass.Prof.Authai Sriteraviroj)

Advisor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์**
ชื่อนักศึกษา	นายกิตติพงศ์ หนูช่วย รหัสประจำตัว 43015801
	นายศักดิ์ศิริ ดิษฐนวล รหัสประจำตัว 43015842
อาจารย์ที่ปรึกษา	ผศ.ประดิษฐ์ วัชรพิบูลย์
	ผศ.อุทัย ศรีธีระวิโรจน์
ระดับการศึกษา	ปริญญาตรีอุตสาหกรรมบัณฑิต
	สาขาเทคโนโลยีโทรคมนาคม
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ในปริญญานิพนธ์นี้กล่าวถึงการออกแบบ และสร้างชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์จะแบ่งการทำงานออกเป็น 2 ส่วนหลักๆคือส่วนของภาคส่งข้อมูลไปควบคุมและตัวรับข้อมูลเพื่อควบคุมอุปกรณ์ไฟฟ้า ส่วนที่เป็นภาคส่งนี้จะรับข้อมูลเข้ามาทางสวิตช์ต่าง ๆ แล้วส่งออกพอร์ทอนุกรม เพื่อส่งให้ภาครับ ส่วนชุดควบคุมทางภาครับนั้นอาจจะมีได้หลายชุด โดยแต่ละชุดจะมีหลายหมายเลขประจำชุดภาครับไม่เหมือนกัน ชุดควบคุมภาครับจะทำงานเมื่อข้อมูลในส่วนที่เป็นหมายเลขชุด ตรงกับภาครับชุดนั้น ๆ ในการควบคุมอุปกรณ์ไฟฟ้าที่กล่าวถึงในที่นี้จะเป็นตัวตวบคุมทิศทางและความเร็วในการหมุนของมอเตอร์ และควบคุมหลอดไฟให้เป็นไปตามชุดควบคุมหลักทางภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis title	ELECTRICAL DEVICS MICROCONTROLLER
Student	Mr.Kittipong Knoochuay
	Mr.Saksiri Dittanual
Advisor	Asst.Prof.Pradit Vatcharapibool
	Asst.Prof.Authai Sriteraviroj
Graduat Level	Bachelor'sdegree of industrial tecnology telecommunication
Department	Information Engineering
Acadamic Year	2001

ABSTRACT

In this thesis described the designing of electrical device controlled by AT89S8252 Microcontroller.There are two main part of this project.

1.Master controlled (Encoder) for sending the controlling data via apair of wire.

2.Slave controlled (Decoder) for control the electrical devices.

By above controller the lamp and the direction of rotate and speed of D.C. motor can be Controlled us desier.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำใคร่ขอขอบพระคุณ ผศ. ประดิษฐ์ วัชรพิบูลย์ และผศ.อุทัย ศรีธีระวิโรจน์ที่ให้คำแนะนำในการทำโครงงานนี้ ให้คำปรึกษาเพื่อเป็นแนวทางในการแก้ไขปัญหาต่างๆ และให้ความช่วยเหลือ ตลอดจนดูแลตรวจสอบโครงงานวิศวกรรมฉบับนี้ลุล่วงได้ด้วยดี รวมทั้งรุ่นพี่เพื่อนๆร่วมสถาบันที่ให้ความช่วยเหลือเกี่ยวกับอุปกรณ์ การทำงานและให้คำปรึกษาในการทำงานตลอดจนขอบคุณภาควิชาอุตสาหกรรมที่ให้โอกาสและสนับสนุนการศึกษามาอย่างสม่ำเสมอ

ท้ายนี้ ผู้จัดทำขอกราบขอบพระคุณ บิดา – มารดา และคณะอาจารย์ทุกท่านที่คอยให้กำลังใจและสนับสนุนมาโดยตลอดจนสำเร็จการศึกษาไปได้ด้วยดี

คณะผู้จัดทำ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ขอบเขตโครงงานในภาคเรียนที่ 1	1
1.2 แผนผังของชุดควบคุมมอเตอร์	2
บทที่ 2 ทฤษฎีทั่วไปไมโครคอนโทรลเลอร์ในตระกูล MCS – 51	3
2.1 คุณสมบัติไมโครคอนโทรลเลอร์ตระกูลเบอร์ AT89S252	3
2.1.1 ทฤษฎีไมโครคอนโทรลเลอร์ในตระกูล MCS – 51	3
2.1.2 คุณสมบัติไมโครคอนโทรลเลอร์เบอร์ AT 89S8252	3
2.1.3 หน่วยความจำของ AT 89S8252	4
2.1.4 หน้าที่ของขาต่างๆ ของ AT 89S8252	6
2.1.5 โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์	8
2.1.6 การทำงานในโหมดต่างๆของ TIMER	20
2.1.7 มาตรฐานการสื่อสารข้อมูลแบบอนุกรม	21
2.2 ทฤษฎี การแปลงสัญญาณอนาล็อกเป็นดิจิตอล	22
2.2.1 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล แบบการเปรียบเทียบแบบขนาน	23
2.2.2 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล แบบดูอัลสโคป	24
2.2.3 วงจรแปลสัญญาณอนาล็อกเป็นดิจิตอล แบบ Successive Approximate	26
2.3 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง	28
2.3.1 การแยกประเภทของมอเตอร์ไฟฟ้ากระแสตรง	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.2 ข้อดีของมอเตอร์แบบฟิลด์แม่เหล็กถาวร	32
บทที่ 3 การออกแบบและสร้างชุดควบคุมภาคส่งและภาครับ	33
3.1 การออกแบบ HARD WARE ชุดควบคุมภาคส่ง	33
3.2 การออกแบบ HARD WARE ชุดควบคุมภาครับ	39
3.3 การออกแบบโปรแกรม	41
บทที่ 4 คุณสมบัติและการใช้งานของชุดควบคุมอุปกรณ์ไฟฟ้าและผลการทดลอง	44
4.1 คุณสมบัติชุดควบคุมอุปกรณ์ไฟฟ้า	44
4.2 ขั้นตอนการใช้ชุดควบคุมอุปกรณ์ไฟฟ้า	44
4.3 วงจรชุดควบคุมทางภาคส่ง (KEY CONTROL)	45
4.4 วงจรขับกระแสไฟมอเตอร์ (DRIVE)	45
4.5 โปรแกรม	48
บทที่ 5 บทสรุป	49
บรรณานุกรม	
ภาคผนวก	
- วงจรของโครงงานทั้งหมด	
- โปรแกรมชุดควบคุมอุปกรณ์ไฟฟ้าทั้งภาคส่งและภาครับ	
- แผนภาพลายทองแดงและการวางอุปกรณ์ของโครงงาน	
- รายละเอียดของข้อมูลไมโครคอนโทรลเลอร์ AT 89S8252	
- รายละเอียดข้อมูลของวงจรขับกระแสมอเตอร์ L293D	
- รายละเอียดข้อมูลของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล ADC0804	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 แผนผังชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทลเลอร์	2
รูปที่ 2.1 สถาปัตยกรรมภายในไมโครคอนโทรลเลอร์	5
รูปที่ 2.2 ไดอะแกรมขาของ AT 89S8252	6
รูปที่ 2.3 วงจรออสซิลเลเตอร์และเวลา	8
รูปที่ 2.4 รีจีสเตอร์ PSW	10
รูปที่ 2.5 หน่วยย่อยต่างๆ ของแรมภายใน	11
รูปที่ 2.6 การทำงานของสแตค	12
รูปที่ 2.7 รีจีสเตอร์ที่ใช้ควบคุมการรับ – ส่งอนุกรม	15
รูปที่ 2.8 รีจีสเตอร์การเลือกโหมดของไทม์เมอร์	16
รูปที่ 2.9 รีจีสเตอร์ที่บอกสถานะและการควบคุมไทม์เมอร์	17
รูปที่ 2.10 รีจีสเตอร์การควบคุมไทม์เมอร์	18
รูปที่ 2.11 แสดงแหล่งกำเนิดสัญญาณขัดจังหวะ	20
รูปที่ 2.12 ชุดข้อมูลอนุกรมในโหมด 1	21
รูปที่ 2.13 ชุดข้อมูลอนุกรมในโหมด 2	22
รูปที่ 2.14 ลักษณะของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล	22
รูปที่ 2.15 วงจรเปลี่ยนแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบเปรียบเทียบขนาน	23
รูปที่ 2.16 วงจรเปลี่ยนแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบดูอัลสโลป	24
รูปที่ 2.17 เอาท์พุทที่ได้จากอินติเกรเตอร์	25
รูปที่ 2.18 วงจรเปลี่ยนแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบ Succesive Approximation	26
รูปที่ 2.19 แผนภูมิแสดงการทำงานของวงจร Succesive Approximation	27
รูปที่ 2.20 แรงบิดในตัวมอเตอร์กระแสตรง	29
รูปที่ 2.21 มอเตอร์ไฟฟ้ากระแสตรงแบบอเมเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก	30
รูปที่ 2.22 คุณสมบัติระหว่างความเร็วและแรงบิดของมอเตอร์ไฟฟ้ากระแสตรงอนุกรมภายใต้แรงดันคงที่	30
รูปที่ 2.23 มอเตอร์ไฟฟ้ากระแสตรงแบบแยกปรับสนามแม่เหล็กได้	31
รูปที่ 2.24 แสดงคุณสมบัติระหว่างความเร็วและแรงบิดของมอเตอร์ขนานภายใต้ภาวะอมาเจอร์แรงดันคงที่และการกระตุ้นสนามแม่เหล็กคงที่	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 2.5 มอเตอร์ไฟฟ้ากระแสตรงแบบฟิลด์เป็นแม่เหล็กถาวร	32
รูปที่ 3.1 การต่อขาใช้งานของไมโครคอนโทรลเลอร์ภาคส่ง	33
รูปที่ 3.2 รูปคีย์สวิตซ์เลือก Slave	35
รูปที่ 3.3 รูปคีย์สวิตซ์ควบคุมอุปกรณ์ไฟฟ้า	36
รูปที่ 3.4 วงจรภาคจ่ายไฟ +5 V	37
รูปที่ 3.5 วงจรภาคจ่ายไฟ +12 V	37
รูปที่ 3.6 วงจรควมความเร็วของมอเตอร์	38
รูปที่ 3.7 การต่อวงจรส่วนแสดงผลชุดควบคุมภาคส่ง	38
รูปที่ 3.8 การต่อขาใช้งานของไมโครคอนโทรลเลอร์ภาครับ	39
รูปที่ 3.9 วงจรขับกำลังมอเตอร์	41
รูปที่ 3.10 FLOWCHAT การออกแบบโปรแกรมชุดควบคุมภาคส่ง	42
รูปที่ 3.11 FLOWCHAT การออกแบบโปรแกรมชุดควบคุมภาครับ	43
รูปที่ 4.1 สัญญาณที่ได้อินพุทมาทางสวิตซ์	45
รูปที่ 4.2 สัญญาณ PWM ที่ออกมาทางที่วงจรขับกระแส ขณะที่ปรับความเร็วน้อยที่สุด	46
รูปที่ 4.3 สัญญาณ PWM ที่ออกมาทางที่วงจรขับกระแส ขณะที่ปรับความเร็วปานกลาง	46
รูปที่ 4.4 สัญญาณ PWM ที่ออกมาทางที่วงจรขับกระแส ขณะที่ปรับความเร็วมากที่สุด	47
รูปที่ 4.5 รูปสัญญาณที่ออกมาทางที่วงจรขับกระแสขณะที่ไม่มีสัญญาณ PWM	47
รูปที่ 4.6 ขณะไม่เปิดสวิตซ์ไฟ L1 และ L2	48
รูปที่ 4.7 ขณะที่เปิดสวิตซ์ไฟ L1 และ L2	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางแสดงรีจีสเตอร์ 13
ตารางที่ 2.2 แสดงการเลือกโหมดการทำงานของพอร์ทอนุกรม 16
ตารางที่ 2.3 แสดงการทำงานของไทม์เมอร์ 2 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

เนื่องจากปัจจุบันนี้มีงานทางด้านอุตสาหกรรมต่างๆ และการขนส่งมวลชนบางชนิดได้มีการใช้มอเตอร์ในงานกันอย่างแพร่หลาย ซึ่งมอเตอร์มีทั้งแบบใช้ไฟกระแสสลับ และใช้ไฟกระแสกระแสตรงซึ่งมอเตอร์ทั้งสองระบบมีการควบคุมที่แตกต่างกันไป แต่เพื่อใช้ในการทำงานที่ต่างกัน เราจำเป็นต้องมีการควบคุมความเร็วในการหมุนของมอเตอร์ตัวต่าง ๆ ซึ่งเป็นการยุ่งยากหากจะใช้บุคลากรหลายคนในการควบคุมความเร็วของมอเตอร์แต่ละตัว ดังนั้นจะเป็นการง่ายถ้าเราใช้ชุดควบคุมความเร็วมอเตอร์ซึ่งเป็นระบบกึ่งอัตโนมัติ มาใช้ในการควบคุมความเร็วมอเตอร์ในงานอุตสาหกรรมต่างๆ ซึ่งทำให้สะดวกและประหยัด

1.1 ขอบเขตของโครงงาน

1. สามารถโปรแกรมไมโครคอนโทรลเลอร์ควบคุมการทำงานของโครงงานได้
2. โครงงานมีการเข้าและถอดรหัสข้อมูลอย่างไม่ผิดพลาด
3. สามารถควบคุมทิศทางในการหมุนของมอเตอร์ไฟฟ้ากระแสตรงได้โดยการสั่งการผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์
4. สามารถควบคุมความเร็วในการหมุนของมอเตอร์ไฟฟ้ากระแสตรงได้โดยการสั่งการผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์

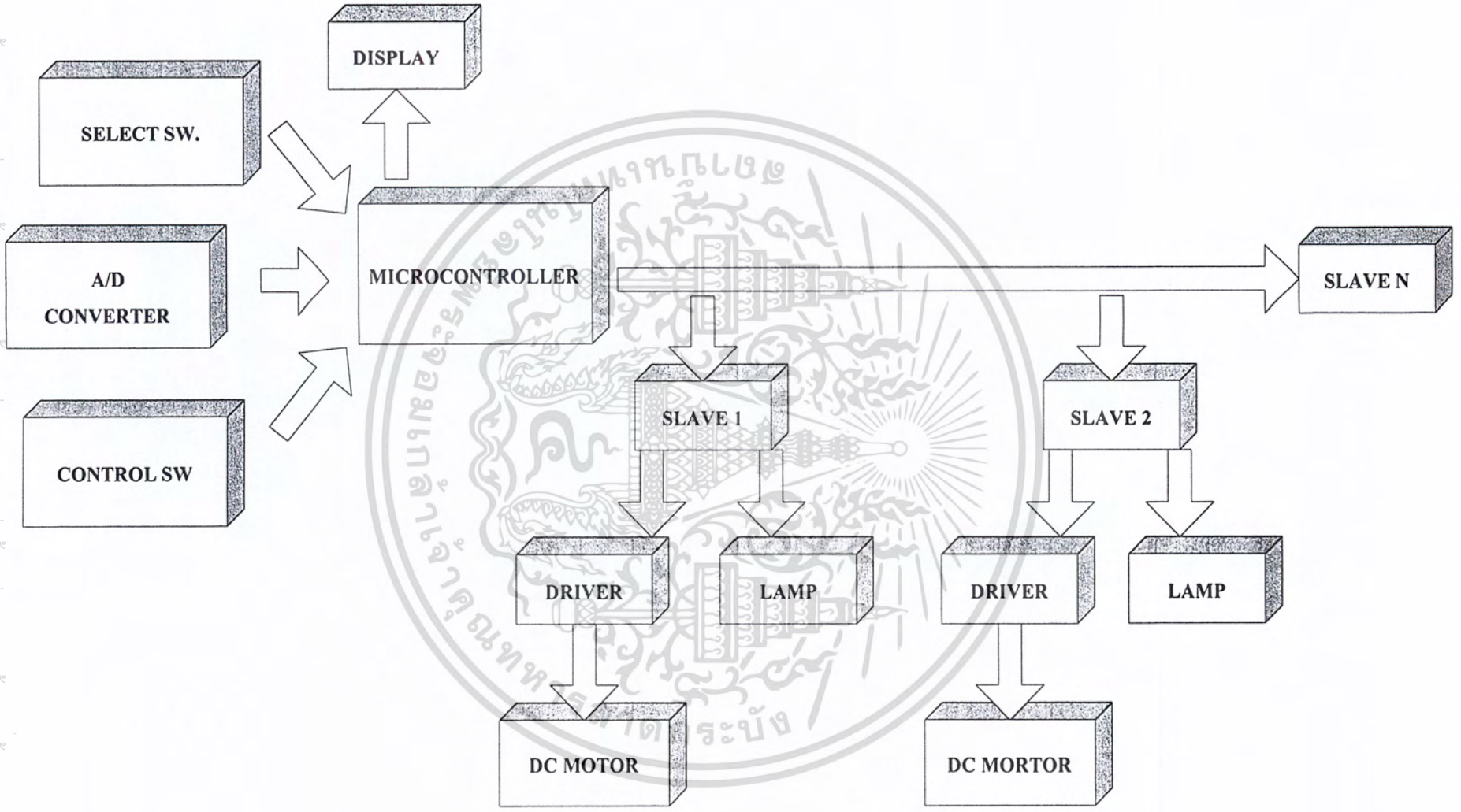
1.2 โครงสร้างของโครงงาน

โครงงานนี้จะประกอบไปด้วยส่วนหลัก ๆ อยู่ 2 ส่วนด้วยกันคือ

1. ส่วนของชุดควบคุมภาคส่ง
2. ส่วนของชุดควบคุมภาครับ

โดยที่แต่ละส่วนจะมีส่วนย่อยๆ ในการทำงานอีกที การทำงานในส่วนของชุดควบคุมภาคส่งจะทำหน้าที่รับข้อมูลที่อยู่ในส่วนของภาคส่งซึ่งมี รหัสประจำตัวของชุดควบคุมภาครับและรับข้อมูลที่รับจากสวิตซ์ต่างๆ แล้วส่งออกพอร์ทอนุกรมไปยังชุดควบคุมภาครับ แล้วชุดควบคุมภาครับจะทำการถอดรหัสเลขประจำตัวและทำการประมวลผลข้อมูลที่ส่ง ดังแสดงในรูปที่ 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แผนผังชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์

บทที่ 2
ทฤษฎีทั่วไป

2.1 สถาปัตยกรรมของ AT89S8252

ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 เป็นส่วนที่ใช้ประมวลผลที่รับคำสั่งมาจากชุดสวิตช์ เนื่องจากไมโครคอนโทรลเลอร์ตระกูลนี้สามาถประยุกต์ใช้งานได้ง่าย และครอบคลุมการใช้งานได้ อย่างกว้างขวาง ซึ่งภายในประกอบไปด้วย หน่วยประมวลผล หน่วยความจำ พอร์ทอินพุต และเอาท์พุต โดยใช้อุปกรณ์ร่วมน้อยมาก ทำให้ประหยัดและสะดวกการใช้งาน โดยโครงงานนี้เลือกใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252

2.1.1 ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล MCS-51

ในการออกแบบการควบคุมนั้น เราใช้ IC ไมโครเมตรคอนโทรลเลอร์ในตระกูล MCS-51 เป็นหัวใจในการควบคุม โดยโดยไมโครคอนโทรลเลอร์ตระกูลนี้สามารถประยุกต์ใช้งานได้ง่าย และครอบคลุมการนำไปใช้งานอย่างกว้างขวางซึ่งภายในประกอบไปด้วย หน่วยการทำงานหลักของระบบคอมพิวเตอร์อย่างครบถ้วน เช่นหน่วยประมวลผล CPU หน่วยความจำ พอร์ตอินพุท และพอร์ทเอาท์พุท โดยใช้อุปกรณ์ภายนอกมาต่อร่วมน้อยมาก ทำให้ประหยัดและสะดวกในการใช้งาน โดยที่ชุดควบคุมนี้ใช้ ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 ของบริษัท ATMEL

2.1.2 คุณสมบัติของไมโครคอนโทรลเลอร์เบอร์ AT89S8252

- มีโครงสร้างและชุดคำสั่งเหมือนกันกับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 มีหน่วยความจำชนิด Flash Memory ขนาด 8 Kbyte สามารถโปรแกรมซ้ำไได้มากกว่าพันครั้ง
- สามารถต่อกับหน่วยความจำภายนอกได้ถึง 64 Kbytes
- มีหน่วยความจำแบบแรม(RAM) 8บิต ขนาด 256 Byte (Intrernal RAM)
- ทำงานที่แรงดัน 2.7 – 6 โวลท์
- มีพอร์ตสื่อสารอนุกรม 1 แชนแนล (UART)
- มีวงจรไทม์เมอร์และวงจรเคาเตอร์ขนาด 16 บิต 3 ชุด เลือกการได้ 4 โหมด
- มีสัญญาณอินเตอร์รัปต์ 6 แหล่ง แบ่งลำดับตามความสำคัญได้ 2 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 หน่วยความจำของ AT89S8252

ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 ประกอบด้วยหน่วยความจำ 2 ส่วน คือ หน่วยความจำโปรแกรมภายนอก (Program Memory) และหน่วยความจำข้อมูล (Data Memory) หน่วยความจำทั้งสองนี้หน้าที่แตกต่างกัน และใช้วิธีการอ้างแอดเดรสสัญญาณติดต่อแยกออกจากกัน

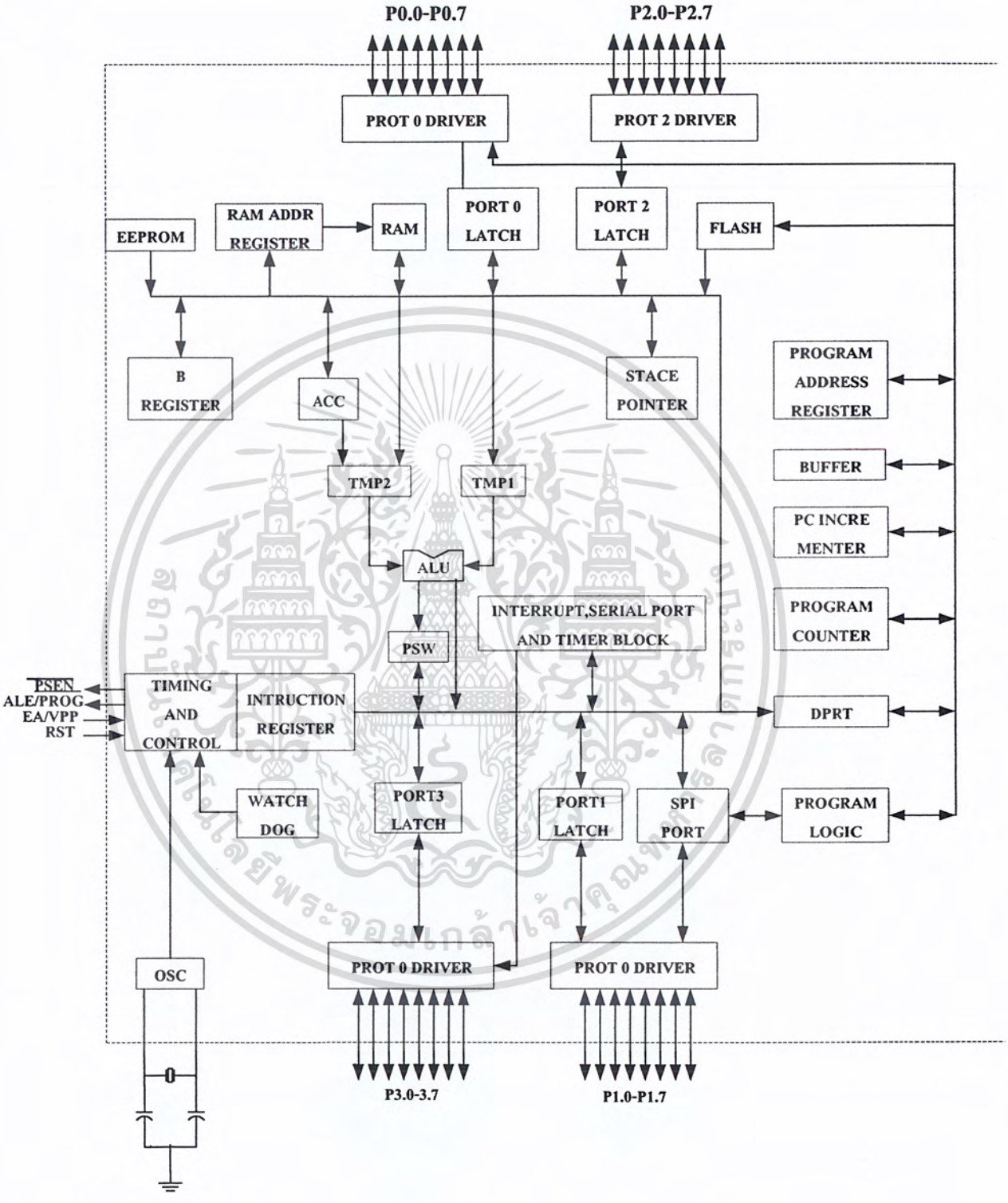
หน่วยความจำโปรแกรม

เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปแบบภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ไมโครคอนโทรลเลอร์ ทำงานเมื่อไมโครคอนโทลเลอร์ทำงานจะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้ไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่น ๆ ตามการทำงนของแต่ละคำสั่งนั้น หน่วยความจำนี้เป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ MCS-51 ตามลำดับการทำงานที่ต้องการแล้วทำการโปรแกรมลงในไมโครคอนโทรลเลอร์ ด้วยครื่องมือพิเศษซึ่งในระหว่างการใช้งานของ MCS – 51 ผู้ใช้จะไม่สามารถเขียนโปรแกรมลงในหน่วยความจำได้ จำนวนตำแหน่งสูงสุดของหน่วยความจำแบบแบบนี้ที่ไมโครคอนโทรลเลอร์ AT89S8252 จะใช้งานได้คือ 64 Kbytes ค่าของตำแหน่ง (Address) จะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000H ถึง FFFFH หน่วยความจำตำแหน่ง 0000H ถึง 0FFFH จำนวน 8 Kbytes นั้น ผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอกถ้าต้องการที่ให้ทำงานใน ROM ที่อยู่ภายในหรือภายนอก ถ้าต้องการที่ให้ทำงานใน ROM ภายในก็ป้อน Logic “1” เข้าที่ขา EA แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอกก็ให้ต่อ Logic “0” เข้าที่ขา EA ส่วนหน่วยความจำที่ตำแหน่ง 2FFFH ถึง FFFFH ต่ออยู่ภายนอกเสมอ

หน่วยความจำข้อมูล

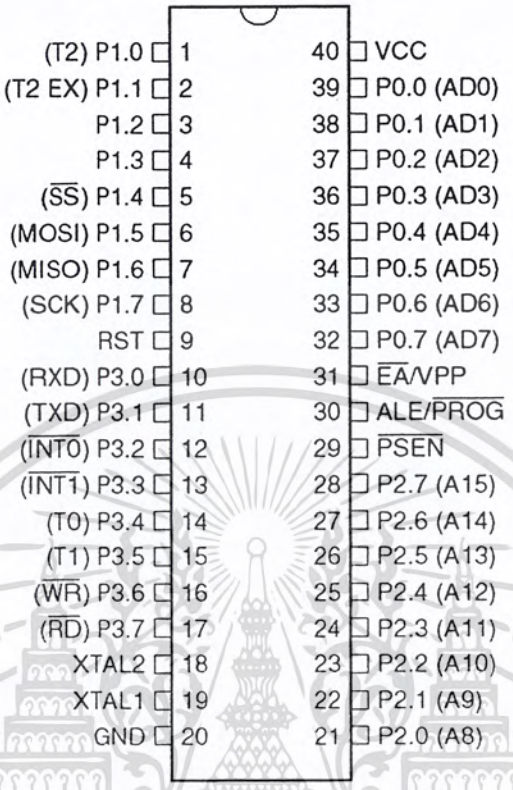
เป็นหน่วยความจำที่ไมโครคอนโทรลเลอร์ สามารถเขียนหรืออ่านข้อมูลได้ ใช้สำหรับเก็บข้อมูลตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลชั่วคราว ซึ่งโดยพื้นฐานแล้วจะเป็นแบบสแตติกแรม การอ่านหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำแบบนี้ประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้ในจะไม่สูญหาย แต่ถ้าปิดเครื่องหรือไม่จ่ายไฟแก่ RAM แล้วข้อมูล RAM จะสูญหายไป การสูญหายของข้อมูลไม่ได้หมายความว่าไม่มีอะไรอยู่เลย แต่เป็นการที่มีข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลเดิมมาแทนที่ หน่วยความจำเป็นแบบ Data Memory ของไมโครคอนโทรลเลอร์เบอร์ AT89S8252 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายในจำนวน 256 Byte อยู่ในตำแหน่ง 00H ถึง FFH ที่ตำแหน่ง 80H ถึง FFH ยังเป็นตำแหน่งของรีจีสเตอร์พิเศษ (SFR) และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกของวงจรรวมแล้วมีได้สูงสุด 64 Kbytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 สถาปัตยกรรมภายในไมโครคอนโทรลเลอร์ตระกูล MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ไดอะแกรมขาของ AT89S8252

2.1.4 หน้าที่ของขาต่างๆ ของ AT89S8252 มีดังนี้

Vcc : ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลท์เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลท์เตจของลอจิก 0 และ 1 จึงต่อเข้ากับอุปกรณ์โดยตรง

Vss : ขา 20 เป็นขาที่ต้องการต่อกับกราวด์ (Ground) ของเหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

Prot 0 : เป็นพอร์ตขนานขนาด 8 บิท อยู่ที่ขา 39- 32 เริ่มจากบิท 0 – 7 ตามลำดับ พอร์ต 0 นี้ใช้ได้ทั้งการรับ – ส่งตำแหน่งข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ตรับ – ส่งข้อมูลก็ได้

Prot 1 : เป็นพอร์ตขนานขนาด 8 บิท อยู่ที่ขา 21 – 28 เริ่มที่บิท 0 – 7 ตามลำดับ พอร์ต 1 นี้ใช้เป็นพอร์ตรับ – ส่งข้อมูลเท่านั้น

Prot 2 : เป็นพอร์ตขนานขนาด 8 บิท อยู่ที่ขา 21 – 28 พเริ่มจากบิท 0 – 7 ตามลำดับ พอร์ต 2 นี้ใช้เพียง 2 ลักษณะคือใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อค่าตำแหน่งนี้ เป็น 8 บิทบนค่าตำแหน่งใช้เป็นพอร์ตรับและส่งข้อมูลกับภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Prot 3 : เป็นพอร์ตขนานขนาด 8 บิท อยู่ที่ขา 10 – 17 เริ่มจากบิท 0 –7 ตามลำดับมีหน้าที่เป็นพอร์ตรับ – ส่งข้อมูล และแต่ละบิทของพอร์ต 3 จะมีฟังก์ชั่นอื่นอีกคือ

P.3/RXD(Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD(Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.3/Int0(External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/Int1(External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0(Time/counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Time/Counter 0 ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T0 นี้ หรือสัญญาณนาฬิกา

P3.5/T1(Time/counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Time/Counter 1 ทำหน้าที่เหมือน กับ T0

P3.7/WR(External Data Memory Write Stobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก AT89S8252

P3.7RD(External Data Memory Read Stobe) ขาสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก

RST : ขารีเซ็ทนี้ จะใช้ในการทำงานของ AT89S8252 ที่ขา RST ภายใน AT89S8252 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปในที่ขานี้จะเป็นการเซ็ตการทำงานของ AT89S8252

ALE : Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์ สัญญาณนี้จะส่งออกมาตลอดเวลายกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก AT89S8252 สัญญาณนี้จะบอกกับอุปกรณ์ภายนอก AT89S8252 ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิทล่างของตำแหน่งหน่วยความจำภายนอก AT89S8252 ที่ต้องการติดต่อออกไปทางพอร์ต 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ต 0 จะใช้รับ – ส่งข้อมูลกับหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ต 0 จะใช้รับ – ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

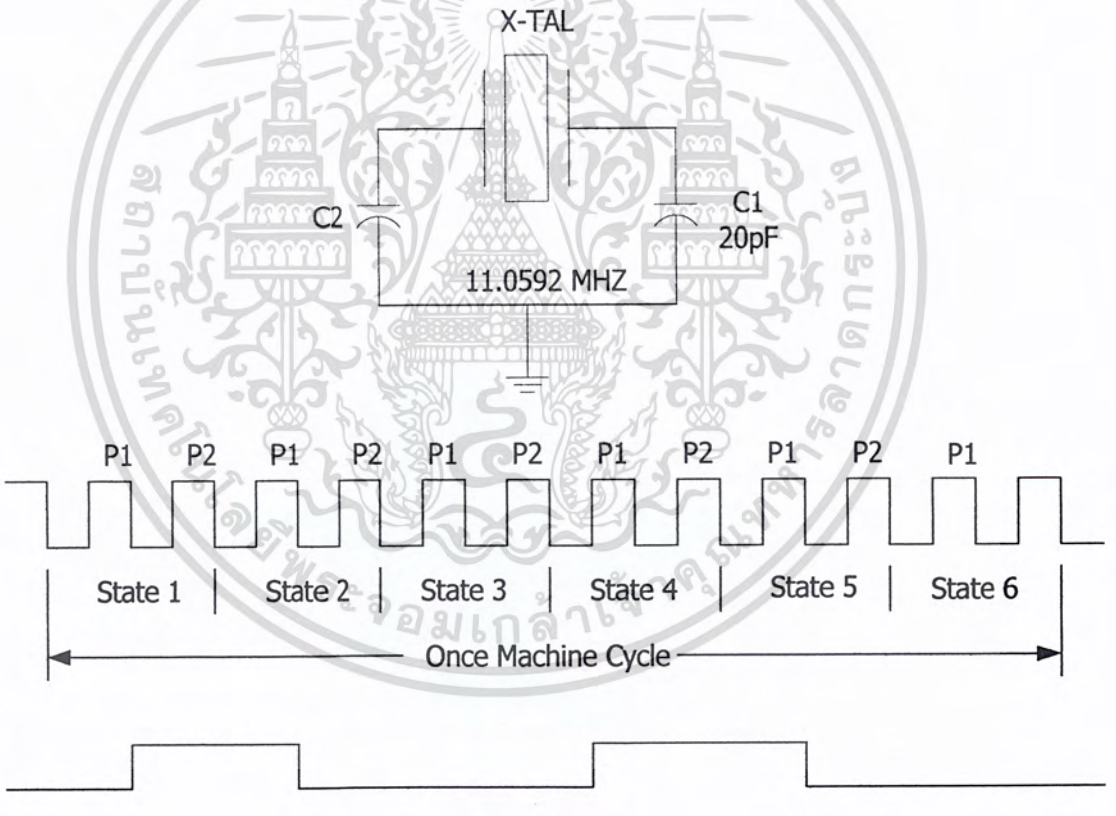
PSEN : Program Store Enable เป็นขาที่ 29 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานจากหน่วยความจำสำหรับโปรแกรมภายนอก AT89S8252 ในกรณีที่อ่านคำสั่งซึ่งเก็บในหน่วยความจำสำหรับโปรแกรมภายใน AT89S8252 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิก 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EA : External Access ขา 31 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าสัญญาณ ลอจิก 0 เข้าไปที่ขา EA นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก AT89S8252 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้าทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน AT89S8252 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่างๆ จาก ROM ภายใน AT89S8252

XTAL,XTAL2 : ขาที่ 19 – 20 ขานี้จะต่อเข้าวงจรออสซิลเลเตอร์ ที่ประกอบไปด้วย CRYSTAL และ CAPCAITOR

2.1.5 โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์ออสซิลเลเตอร์



8051 Timing

รูปที่ 2.3 วงจรออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณเวลาที่ใช้ในเวลาที่ใช้ในคำสั่ง หรือเวลาของโปรแกรมหน่วงเวลาเราต้องทราบความถี่ของพัลซ์นาฬิกาที่ต่อให้กับไมโครโทรลเลอร์ การต่อวงจรกำเนิดพัลซ์นาฬิกาขา STAL1,XTAL2 ต่อกับวงจรรีโซแนนท์เป็นออสซิลเตอร์ เช่นใช้คริสตัล (Crystal) และตัวเก็บประจุ ความถี่คริสตัลเป็นความถี่นาฬิกาภายในของไมโครคอนโทรลเลอร์

เซรามิกรีโซแนนท์อาจถูกนำมาใช้เพราะมีราคาต่ำ แต่เสถียรภาพทางความถี่จะลดลงและความแน่นอนไม่ค่อยดี ถ้าเป็นการสื่อสารอนุกรมด้วยความเร็วสูงก็อาจเกิดภาวะวิกฤตขึ้นได้

ออสซิลเลเตอร์ที่เป็นคริสตัล , ตัวเก็บประจุ และอินเวอร์เตอร์บนชิพ จะสร้างขบวนพัลซ์ความถี่ของคริสตัล ความถี่นาฬิกาต้องมีช่วงเวลาของไมโครคอลโทรลเลอร์ ที่น้อยที่สุดเรียกว่าช่วงเวลาของพัลซ์ช่วงที่น้อยที่สุดที่สามารถทำคำสั่งง่าย หรือซับซ้อน เรียกว่า Machine cycle ซึ่งประกอบด้วย 6 สภาวะ แต่ละภาวะเป็นช่วงเวลาสำหรับการทำงาน ที่แยกจากกันของไมโครคอลโทรลเลอร์ เช่น เฟทซ์คำสั่ง , ทำคำสั่ง หรือเขียนข้อมูลแต่ละสภาวะที่มีสองพัลซ์

โปรแกรมคำสั่งอาจเป็น 1 , 2 หรือ 4 Machine cycle ขึ้นอยู่กับชนิดของคำสั่ง คำสั่งจะถูกเฟทซ์และทำโดยไมโครคอลโทรลเลอร์แบบอัตโนมัติ โดยเริ่มต้นที่ตำแหน่งที่ 0000H ที่เวลาเกิดการรีเซ็ตครั้งแรก

โปรแกรมเคาท์เตอร์ และดาต้าพ้อยเตอร์

AT89S8252 ประกอบด้วยรีจิสเตอร์ขนาด 16 บิต 2 ตัว คือ PC(Program Count) และ DPTR (Data Pointer) แต่ละตัวที่เก็บค่าแอดเดรสขนาด 1 ไบต์ โปรแกรมคำสั่งจะถูกแฟทซ์จากตำแหน่งแอดเดรสใน Program Counter รอมบนชิพอาจมีแอดเดรส 0000H ถึง 0FFFH รอมภายนอกจะมากกว่า 0FFFH โดย Program Counter จะเพิ่มอัตโนมัติ หลังจากเฟทซ์คำสั่ง และอาจเปลี่ยนโดยบางคำสั่ง Program Counter เป็นรีจิสเตอร์ตัวเดียวที่ไม่มีแอดเดรสภายใน Data Pointer (DPTR) ประกอบด้วยรีจิสเตอร์ 8 บิท สองตัวเรียกว่า DPH และ DPL จะเก็บค่าแอดเดรสสุดท้ายหลังจากเข้าถึง โค้ดทั้งในและนอกชิพ ข้อมูลก็เช่นเดียวกัน DPTR อยู่ภายใต้การควบคุมของโปรแกรมคำสั่งและอ้างอิงโดยชื่อของมัน (DPTR) หรือแต่ละไบต์ย่อย DPH และ DPL

รีจิสเตอร์ A และ B

AT89S8252 มีรีจิสเตอร์ที่ใช้งานทั่วไป 34 ตัว และ 2 ตัวในนี้คือ รีจีสเตอร์ A และรีจิสเตอร์ B ใช้เป็นหลักในทางคณิตศาสตร์ของ AT89S8252 ที่เหลือ 32 ตัว ถูกจัดเป็นช่วงของแรมภายใน 4 แบงค์ ถือ B0 – B3 แบงค์ละ 8 ตัว (R0 – R7) รีจิสเตอร์ A หรือ แอคคิวมูเลเตอร์เป็นตัวที่ใช้มากที่สุด เช่น การบวก , การลบ , การคูณ , การหารเลขจำนวนเต็ม , การทำบูลีน รีจิสเตอร์ A ใช้เคลื่อนย้ายข้อมูลระหว่าง MCS-51 และหน่วยความจำภายนอก รีจิสเตอร์ B ใช้กับรีจิสเตอร์ A สำหรับการคูณและการหาร และไม่มีหน้าที่อื่นนอกเหนือจากเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟลกและโปรแกรมสเตตัสเวอร์ด (PSW : Program Status Word)

แฟลกเป็นรีจิสเตอร์ 1 บิต ที่เก็บผลลัพธ์ของบางคำสั่ง คำสั่งอื่นไม่สามารถทดสอบสภาวะของแฟลก และตัดสินใจ แฟลกสามารถที่จะเป็นแอดเดรสที่ต้องการได้ และอยู่ร่วมกันเป็นกลุ่ม PSW และ PCONMCS-51 มีแฟลกทางคณิตศาสตร์ 4 แฟลก ซึ่งให้ผลลัพธ์โดยอัตโนมัติ เมื่อมีการทำงานทางคณิตศาสตร์ และมี 3 แฟลกที่ใช้งานทั่วไป ซึ่งสามารถเซต 1 หรือเคลียร์ 0 ได้ตามต้องการ (P) แฟลกที่ใช้งานทั่วไปมีชื่อว่า F0 , GF0 , GF1 ซึ่งโปรแกรมเมอร์สามารถใช้กำหนดโปรแกรมได้ ความจำว่าแฟลกทั้งหมดสามารถเซต หรือรีเซตโดยโปรแกรม ส่วนแฟลกคณิตศาสตร์มีผลโดยคำสั่งทางคำสั่งทางคณิตศาสตร์ PSW ในรูปประกอบด้วยแฟลกคณิตศาสตร์ แฟลกสำหรับผู้ใช้ (F0) และแฟลกเลือกรีจิสเตอร์ในแบงค์แฟลก 2 ตัวที่เหลือ (GF0 , GF1) เก็บใน PSW ในรูป

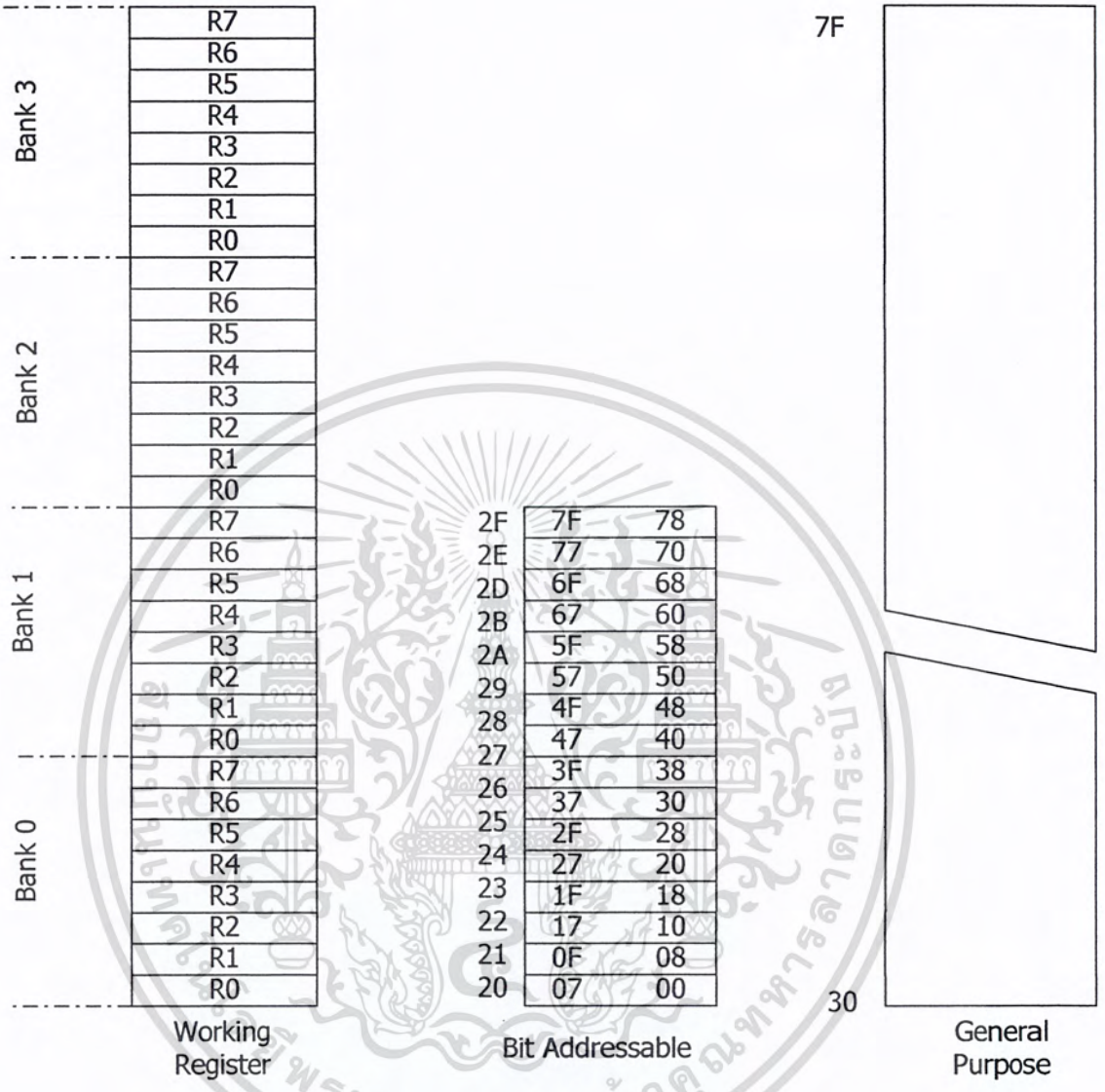
7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	0V	-	P

รูปที่ 2.4 รีจิสเตอร์ PSW

สแตคและสแตคพอยต์เตอร์

สแตกเป็นพื้นที่ของแรมภายในที่ใช้เก็บข้อมูลและส่งออกอย่างรวดเร็ว SP 8 บิตของ AT89S8252 ใช้เก็บแอดเดรสของแรมภายใน โดยเป็นแอดเดรสสุดท้ายของการกำหนดบนสแตค เมื่อข้อมูลเก็บบนสแตค SP จะเพิ่มค่าก่อนเก็บข้อมูลออกจากสแตค ข้อมูลจะถูกอ่านก่อนและ SP จะลดค่าลงเพื่อให้ข้อมูลสามารถเก็บได้

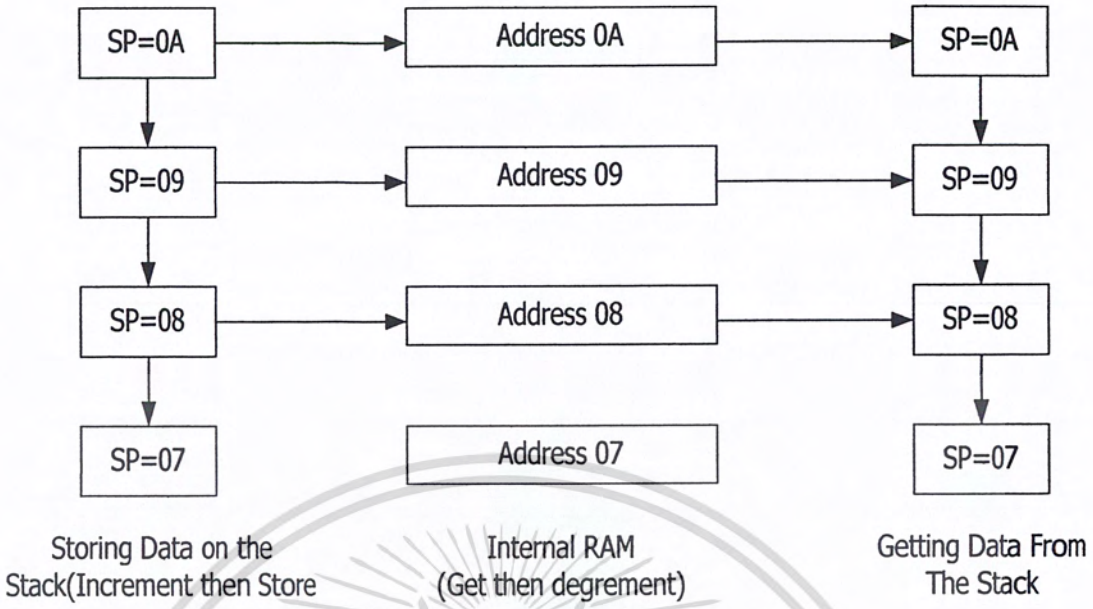
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.5 หน่วยย่อยต่าง ๆของแรมภายใน

การทำงานของสแตคและ SP แสดงในรูปที่ 2.6 เซตที่ 07H เมื่อ AT89S8252 รีเซตและสามารถเปลี่ยนแอดเดรสได้โดยโปรแกรมเมอร์สแตคถูกจำกัดด้วยขนาดของแรมภายในถ้าโปรแกรมเมอร์ไม่ระวังสแตคอาจเก็บข้อมูลเกินพื้นที่แรมที่แอดเดรสบิทได้ และพื้นที่แรมใช้งานทั่วไปโปรแกรมเมอร์ต้องมั่นใจว่าสแตคจะไม่เกินขอบเขตที่กำหนดไว้ก่อนโดยปกติสแตคจะอยู่ส่วนบนของแรมภายในโดยเลือกจำนวนที่อยู่บน SP เพื่อหลีกเลี่ยงการเกินขอบเขตของแรมภายใน ควรจำไว้ว่า PC ไม่ใช่ส่วนของ SFR และไม่มีแอดเดรสแรมภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 การทำงานของสแตค

รีจิสเตอร์ทำหน้าที่

การทำงานของ MCS-51 จะไม่ใช้แรมภายใน 128 ไบต์ตำแหน่ง 00h ถึง 7Fh ซึ่งเป็นกลุ่มของรีจิสเตอร์ภายใน (SFR) ซึ่งอาจทำเป็นแอดเดรสเหมือนกับแรมภายใน โดยใช้แอดเดรสจาก 80H ถึง FFH ชื่อของ SFR แอดเดรสของแรมภายในอยู่ในตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางแสดงรีจิสเตอร์

NAME	FUNCTION	INTERNAL.RAM ADDRESS(HEX)
A	Accumulator	0E0
B	Arithmetic	0F0
DPH	Addressing external memory	83
DPL	Addressing external memory	82
IE	Interrupt enable control	0A8
IP	Interrupt priority	0B8
P0	Input/output port latch	80
P1	Input/output port latch	90
P2	Input/output port latch	A0
P3	Input/output port latch	0B0
PCON	Power control	87
PSW	Program status word	0D0
SCON	Serial port control	98
SBUF	Serial port data buffer	99
SP	Stack pointer	81
TMOD	Time/counter mode control	89
TCON	Time/counter control	88
TLO	Time 0 low byte	8A
THO	Time 0 low byte	8C
TLI	Time 1 low byte	8B
RHI	Time 1 low byte	8D

SFRs เป็นชื่อของบางออฟโค็ดตามชื่อของฟังก์ชั่น เช่น A หรือ THO และถูกอ้างอิงโดยแอดเดรสของมัน เช่น 0E0H หรือ 8CH ควรสังเกตว่าแอกเดรสที่จะเริ่มด้วยตัวเลข ดังนั้นแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

E0H สำหรับ A SFR จึงต้องเริ่มต้นด้วย 0 ข้อผิดพลาดในการใช้ตัวเลขจะมีผลต่อความผิดพลาดของแอสแซมเบอร์เมื่อโปรแกรมนำมาใช้งานร่วมกัน

TIMER

ถ้าเคาน์เตอร์ถูกตั้งเป็นไทม์เมอร์ ก็นับความถี่นาฬิกาภายใน AT89S8252 ออสซิลเลเตอร์หารด้วย 12 เช่น ความถี่คริสตัล 12 MHz จะได้ไทม์เมอร์ 500kHz

วงจรนาฬิกานับที่นับ จะถูกเกทไปที่ไทม์เมอร์โดยใช้วงจรดังรูป สัญญาณนาฬิกาของออสซิลเลเตอร์จะพักเป็นพัลซ์มาสู่ไทม์เมอร์ บิด C/T ใน TMOD ต้องเป็น 0 บิต TRX ใน TCON ต้องเป็น 1 (ไทม์เมอร์ทำงาน) และบิตเกทใน TMOD ต้องเป็น 0 และขา INTX ต้องเป็น 1 ในทางตรงกันข้ามเคาน์เตอร์ทำหน้าที่เป็นตัวนับได้โดยเกทไปที่เคาน์เตอร์ โดยบิตทำงาน(run bit) และเกท (Gate bit) หรือ INTX

PORT 0 ถึง 3

Spectal Function Register ชื่อ P0 , P1 , P2 และ P3 เป็นรีจิสเตอร์ขนาด 8 บิท ของหน่วยความจำ สำหรับข้อมูลภายใน AT89S8252 ที่ตำแหน่ง 80H , 90H , 0B0H ตามลำดับ การเขียนข้อมูลไปยังหน่วยความจำและตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ตนั้น ๆ ของ AT89S8252 ข้อมูลที่เขียนออกไปจะถูก LATCH ค้างไว้และปรากฏที่แต่ละบิทของพอร์ตเช่น MOV 80H , #18H จะปรากฏสภาวะลอจิก LLLHHHLLL ที่ขาบิท 7 ถึง 0 ของพอร์ต 0 ตามลำดับในการอ่านข้อมูลจากพอร์ทจะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ทนั้นๆ เสียก่อน ทุกบิทของพอร์ท 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้โดยคำสั่ง SETB bit และ CLR bit

Serial Data Buffer

รีจีสเตอร์นี้มีขนาด 8 บิทและมีตำแหน่งของหน่วยความจำสำหรับข้อมูลภายในไมโครคอนโทรลเลอร์เบอร์ AT89S252 เท่ากับ 99H โครงสร้างภายในแล้วรีจีสเตอร์นี้มี 2 ตัวที่มีชื่อดียวกันตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรมออกจาก AT89S8252 และอีกตัวสำหรับข้อมูลแบบอนุกรมที่เข้ามา ดังนั้น Serial Port จึงมีการทำงานแบบ Full Duplex เพราะสามารถส่งและรับข้อมูลได้ในเวลาเดียวกันเนื่องจากรีจีสเตอร์แยกออกจากกัน ข้อมูลที่ต้องการส่งออกก็ให้เขียนไปยัง SBUF แล้งสั่งให้ข้อมูลออกมา และจะเริ่มส่งจากบิท 0-7 ข้อมูลที่เข้ามาทางขวา RXD บิทแรกคือบิท 0

SCON (Serial Port Contrrol Register)

เป็นรีจีสเตอร์ 8 บิท ใช้ควบคุมการรับ – ส่งข้อมูลทาง Serial Port แต่ละบิทมีความหมายของข้อมูลในรีจีสเตอร์นี้มีความหมายเฉพาะดังในรูปที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCON:SERIAL PORT CONTROL REGISTER.BIT ADDRESSABLE

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

รูปที่ 2.7 รีจีสเตอร์ที่ใช้ควบคุมการรับ – ส่งอนุกรม

RI (Receive Interrupt Flag)

จะถูกกำหนดโดย ฮาร์ดแวร์ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโหมด 0 นั้น RBS จะมีค่าเป็น 1 ส่วนโหมดอื่นๆ จะเป็น 1 ได้ก็ต่อเมื่อ ข้อมูลเข้ามาถึงครึ่งหนึ่ง ของ Stop bit สามารถ clear ให้เป็น 0 ได้ บิทนี้มีประโยชน์เพื่อให้รู้ว่าข้อมูลที่เข้ามาที่ SBUF นั้นครบแล้ว

TI (Transmit Interrupt Flag)

จะถูกกำหนดโดย ฮาร์ดแวร์ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโหมด 0 นั้น RBS จะมีค่าเป็น 1 ส่วนโหมดอื่นๆ จะเป็น 1 ได้ก็ต่อเมื่อ เริ่มการส่ง Stop bit สามารถ Clear ให้เป็น 0 ได้ บิตนี้มีประโยชน์เพื่อให้รู้ว่าส่งข้อมูลออกไปครบแล้ว

RB8

ในการส่งข้อมูลแบบอนุกรมโหมด 2 และ 3 จะใช้บิทนี้สำหรับเก็บข้อมูลบิทที่ 9 ที่ออกมาทางพอร์ทอนุกรม ส่วนโหมด 1 นั้นบิทนี้จะเก็บ Stop bit ซึ่งมีค่าเป็น 1 นั่นเองในโหมด 0 บิทนี้จะไม่ถูกใช้งานค่าตำแหน่งของบิทนี้คือ 9AH

TB8

ในการส่งข้อมูลแบบอนุกรมโหมด 2 และ 3 จะใช้บิทนี้เก็บข้อมูลบิทที่ 9 ส่วนโหมดอื่นจะไม่ใช้งานในบิทนี้สามารถทำได้โดยใช้คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิทนี้คือ 9BH

REN (Receive Enable)

เป็นบิทที่ใช้กำหนดให้ทำการรับข้อมูลจาก Serial Port ถ้าเป็น 1 ก็จะรับข้อมูลเข้ามา ถ้าเป็น 0 จะไม่รับที่ขา RXD

SM2

เป็นบิทควบคุมการทำงานของฮาร์ดแวร์ที่จะทำให้ RI เป็น 1หรือไม่ถ้าเป็น 0 จะเป็นไปตามปกติ แต่ถ้าเป็น 1 ในโหมด 2 และ 3RI จะเป็น 1 เมื่อข้อมูลบิทที่ 9 เข้ามาเป็น 1 ในโหมด 1 RI จะเป็น 1เมื่อมี Stop bit เข้ามาโหมด 0 เป็น 0 เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SM0 , SM1

ใช้ร่วมกัน เพื่อเลือกการทำงานว่าทำงานในโหมดไหน กำหนดได้ตามตารางนี้

ตารางที่ 2.2 แสดงการเลือกโหมดการทำงานของพอร์ทอนุกรม

SM0	SM1	MODE	Description
0	0	0	Shift register
0	1	1	8-bit UART
1	0	2	9-bit UART
1	1	3	9-bit UART

TMOD (Timer/counter mode register)

เป็นรีจีสเตอร์ขนาด 8 บิท มีหน้าที่ควบคุมการทำงานของ Time 0 และ Time 1

7	6	5	4	3	2	1	0
GATE	C/T	MI	M0	GATE	C/T	MI	M0

รูปที่ 2.8 รีจีสเตอร์การเลือกโหมดของ Timer

ในรูปรีจีสเตอร์นี้จะแบ่งเป็น 2 ชุดๆ ละ 4 บิท ซึ่งมีหน้าที่เหมือนกัน

GATE

เป็นบิตที่ใช้ควบคุมว่า Timer ทำงานหรือไม่ ถ้าเป็น 1 จะทำงานเมื่อ INTx และบิท TRx เป็น 1 ด้วย

C/T

ใช้เลือกการทำงานว่าจะใช้งานเป็น Timer หรือ Counter โดยเป็น 1 เป็น Counter

M0 , M1

เป็น 2 บิท ที่ใช้ร่วมกันเพื่อเลือกโหมดการทำงานของ การทำงานของโหมด 0 และ 2 จะเหมือนกัน แต่ ในโหมด 3 การทำงานของทั้งสองจะแตกต่างกัน ค่าในและจะเลือกโหมดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

TCON (Timer control Register)

เป็นรีจีสเตอร์ขนาด 8 บิท ใช้บอกสภาวะของ Timer 0และ Timer 1 ประกอบไปด้วย

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 2.9 รีจีสเตอร์ที่บอกสถานะและควบคุมไทม์เมอร์

IT0

ใช้กำหนดวิธีการจัดการขัดจังหวะจากสัญญาณที่เข้ามาทาง INTO ถ้าเป็น 0 จะดูที่สภาวะที่เป็นถ้าเป็น 1 ดูที่ของขาลง

IE0

จะเป็น 1 เมื่อ เกิดการขัดจังหวะ เมื่อเสร็จแล้วบิทนี้จะกลับเป็น โดยดูที่ INTO

IT1

เหมือนกับ IT0 แต่จะดูสัญญาณที่เข้ามาทาง INT1

IE1

เหมือนกับ IE0 แต่จะดูสัญญาณที่เข้ามาทาง INT1

TR0

ถ้าบิทนี้เป็น 0 Timer 0 จะ ไม่ทำการนับไม่ว่ากรณีใดๆ แต่ถ้าเป็น 1 จะขึ้นอยู่กับสัญญาณ GATE,INTO

TF0

เมื่อการนับใน Timer 0 เกิด Overflow ขึ้นจะทำให้บิตนี้เป็น 1 แล้วทำการขัดจังหวะจะทำให้บิทนี้กลับเป็น 0

TR1

เหมือนกับ TR0 แต่จะทำงานกับ Timer 1

TF1

เหมือนกับ TF0 แต่จะทำงานกับ Timer 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46410

T2 CON (Timer 2 Control Register) เป็นรีจีสเตอร์ขนาด 8บิท

TF2	EFX2	RCLK	TCLK	EXEN2	TR2	C/T2	CP8RL2
7	6	5	4	3	2	1	0

รูปที่ 2.10 รีจีสเตอร์การควบคุมไทม์เมอร์

TF2

เป็นบิทแสดงการเกิดโอเวอร์โฟล์วของไทม์เมอร์ 2 ถูกเซ็ตโดยฮาร์ดแวร์และเคลียร์โดยใช้คำสั่งในโปรแกรมเท่านั้น และจะไม่ถูกเซตเองเมื่อใช้เป็นตัวกำหนดค่า บอร์ดเรด

EXF2

บิทแสดงการอินเตอร์รัปต์จากภายนอกของไทม์เมอร์ 2 ระหว่างการทำงานในโหมด ออโตรีโหลดและแคปเตอร์ หากบิท EXEN2 เป็น 1และมีการเปลี่ยนสถานะที่ขา T2EX จะทำให้บิทนี้เซต เมื่อบิทควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัปต์ของไทม์เมอร์ 2 และบิท EXF2 เซต จะกระโดดไปทำงานที่โปรแกรมอินเตอร์รัปต์ทันที

RCLK

บิทเลือกใช้ไทม์เมอร์ 2 เป็นตัวกำหนด บอร์ทเรต ในการรับข้อมูล

1 ใช้ไทม์เมอร์ 2 เป็นตัวกำหนด

0 ใช้ไทม์เมอร์ 1 เป็นตัวกำหนด

TCLK

บิทเลือกใช้ไทม์เมอร์ 2 เป็นตัวกำหนด บอร์ทเรต ในการรับข้อมูล

1 ใช้ไทม์เมอร์ 2 เป็นตัวกำหนด

0 ใช้ไทม์เมอร์ 1 เป็นตัวกำหนด

EXEN2

บิทเลือกการทำงานจากสัญญาณที่ขา T2EX

1 พิจารณาสัญญาณที่ขา T2EX

0 ไม่พิจารณาสัญญาณที่ขา T2EX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TR2

บิตที่ใช้ควบคุมการทำงานของไทม์เมอร์ 2

1 เริ่มการนับของไทม์เมอร์ 2

0 หยุดการนับของไทม์เมอร์ 2

C/T2

บิทเลือกการทำงาน

1 เลือกการทำงานแบบ แคปเตอร์

0 เลือกการทำงานแบบไทม์เมอร์

CP/RL2

บิตเลือกการทำงาน

1 เลือกทำงานแบบ แคปเตอร์

0 เลือกการทำงานแบบ ออโต้รีโหลด

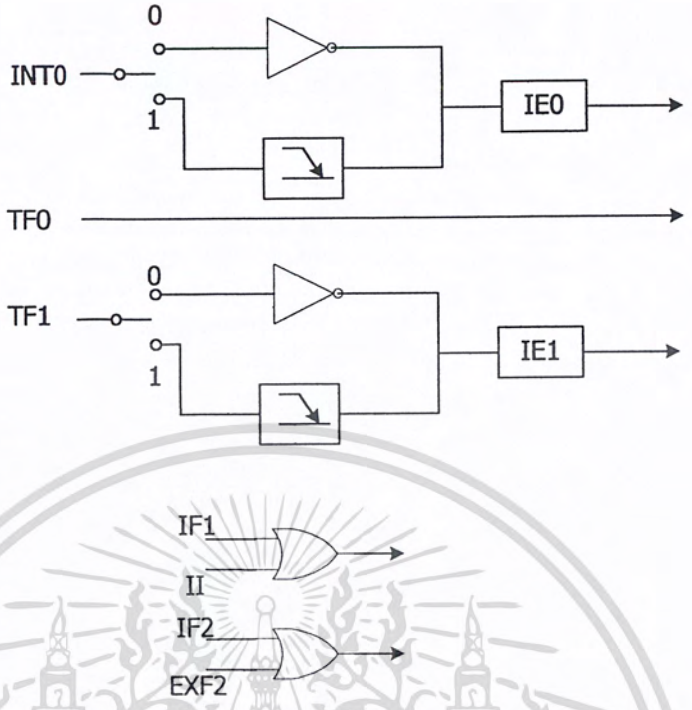
ตารางที่ 2.3 แสดงโหมดการทำงานของไทม์เมอร์ 2

RCLK+TCLK	CP/RL2	TR2	MODE
0	0	1	16 bit Auto-reload
0	1	1	16 bit Capture
1	X	1	Band Rate Generater
X	X	0	(Off)

IE Interrupt Enable Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0A8H

การขัดจังหวะการทำงาน เป็นการที่มีสัญญาณหนึ่งหรือคำสั่งหนึ่งที่ (ไม่ใช่คำสั่ง หรือ)ที่จะทำให้การทำงานการปกติของโปรแกรมถูกขัดจังหวะ แล้วข้ามไปทำงานยังตำแหน่งที่ตำแหน่งใดตำแหน่งหนึ่งที่กำหนดไว้ เมื่อเสร็จการทำงานแล้วจะกลับมาทำงานในโปรแกรมที่ตำแหน่งถัดไป ไมโครคอนโทรลเลอร์สามารถรับสัญญาณ INTERRUPT ได้อย่างน้อย 5 ชนิดด้วยกัน แต่บางเบอร์สามารถรับได้ 6 ชนิดส่วน ไมโครคอนโทรลเลอร์ที่ใช้ในโครงงานนี้สามารถรับสัญญาณของการ INTERRUPT ได้ 6 ชนิด แหล่งกำเนิด สัญญาณ INTERRUPT ทั้ง 5 ชนิดมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงแหล่งกำเนิดสัญญาณขัดจังหวะ

2.1.6 การทำงานในโหมดต่างๆ ของ Timer

โหมด 0

ในโหมดนี้สามารถใช้ได้ทั้ง Timer 0 และ Timer 1 จะทำหน้าที่เป็นตัวนับ 13 บิทโดยมี 5 บิทล่างของ TLx และ THx ในการควบคุมการนับโดยการกำหนดบิท GATE เมื่อเป็น 1 จะถูกควบคุมโดยใช้สัญญาณที่เข้ามาทางขา INTx เป็นตัวเริ่มนับ (จะต้องทำการเซต TRx ก่อน) แต่ถ้าเป็น 0 จะใช้ TRx เป็นตัวควบคุมถ้าเป็น 1 จะเกิดการอินเตอร์รัปต์ ที่ตำแหน่งของการอินเตอร์รัปท์ที่กำหนด

โหมด 1

ในโหมดนี้จะเหมือนกับโหมด 0 ทุกประการแต่จะมีตัวนับเป็น 16 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2

เป็นโหมด Auto Load การนับสูงสุดจะทำได้เพียง 8 บิท เท่านั้นแต่จะมีข้อดีคือเราสามารถที่จะโหลดข้อมูลที่ THx เข้าไปยัง TLx ได้โดยอัตโนมัติ เมื่อการนับเกิดการโอเวอร์โฟลว์ เพื่อทำการรอนับครั้งต่อไป และในส่วนอื่น ๆ จะไม่แตกต่างกัน

โหมด3

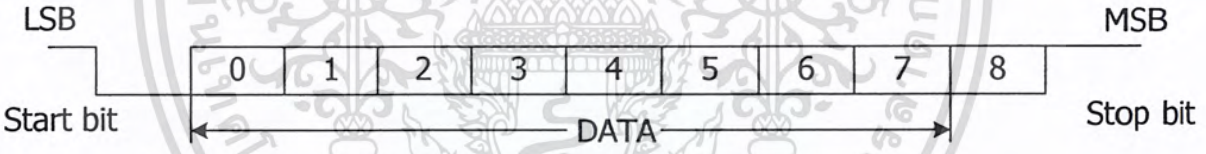
Timer 1จะไม่มีการทำงานในโหมดนี้และโหมด 0 จะมีตัวนับ 2 ตัวคือ TL0 และ TH0 โดย T10 โดยจะใช้ควบคุมเช่นเดิมแต่ใน TH0 สัญญาณการนับจะได้ 1/12 ของความถี่ OSC และใช้ TR1 เป็นตัวควบคุมเท่านั้นและเมื่อเกิดการโอเวอร์โฟลว์จะแยกเป็น 2 ตัวคือ TF0 และ TF1 จาก TL0 และ TH0 ตามลำดับ

2.1.7 มาตรฐานการสื่อสารข้อมูลแบบอนุกรม (UART)

แบ่งการทำงานได้ 4 โหมด

MODE 0 : โหมดนี้จะส่งสัญญาณ Clock ที่ใช้สำหรับเลื่อนข้อมูล 1 ชุด ซึ่งจะมีเพียง 8 บิท เท่านั้น อัตราการส่งจะเท่ากับ 1/12 เท่าของความถี่สัญญาณนาฬิกา

MODE 1: ข้อ 1 ชุดจะมี 10 บิท เริ่มด้วย Start bit (เป็น 0) , ข้อมูล 8 บิท , Stop bit (เป็น 1)



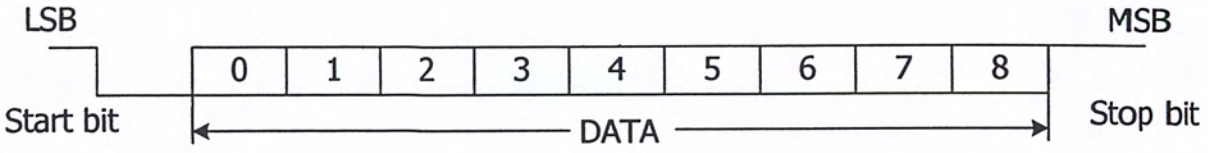
การส่งข้อมูลมีดังรูป

รูปที่ 2.12 ชุดข้อมูลอนุกรมในโหมด 1

เมื่อรับข้อมูลเข้ามา ข้อมูล 8 บิท จะถูกเก็บไว้ใน SBUF และ Stop bit จะถูกเก็บไว้ในบิท RB8 ใน SCON ในการส่งข้อมูลที่ต้องการส่งไปยังรีจีสเตอร์ SBUF อัตราการส่งข้อมูลในโหมดนี้สามารถกำหนดได้โดยจะขึ้นกับการเกิด Overflow ใน Timer 1

MODE 2: ในโหมด 1 จะมี 11 บิตเริ่มด้วย Start bit (เป็น 0), ข้อมูล 8 บิท (เริ่มจากบิท 0) ข้อมูลบิทที่ 9 และ Stop bit บิทที่ 9 นี่ที่จะส่งออกสามารถกำหนดได้ว่าจะเป็น 1 หรือ 0 โดยการกำหนดบิท RBS เมื่อรับข้อมูลเข้ามา บิทที่ 9 ก็จะถูกส่งไปเก็บไว้ใน RBS ดังรูป

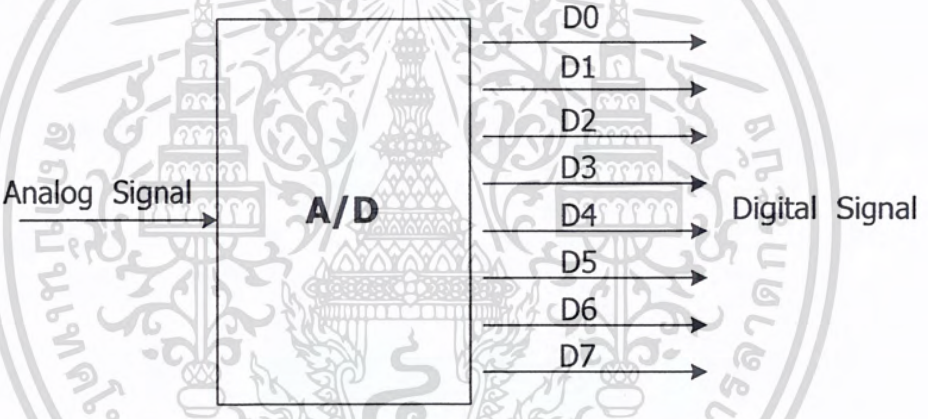
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ชุดข้อมูลอนุกรมในโหมด 2

MODE 3 : จะมี 11 บิท เหมือนโหมด 2 แต่ต่างกันตรงที่อัตราการส่งข้อมูลเท่านั้น คือจะกำหนดได้เหมือนกับโหมด 1

2.2 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (Analog to digital Conversion)

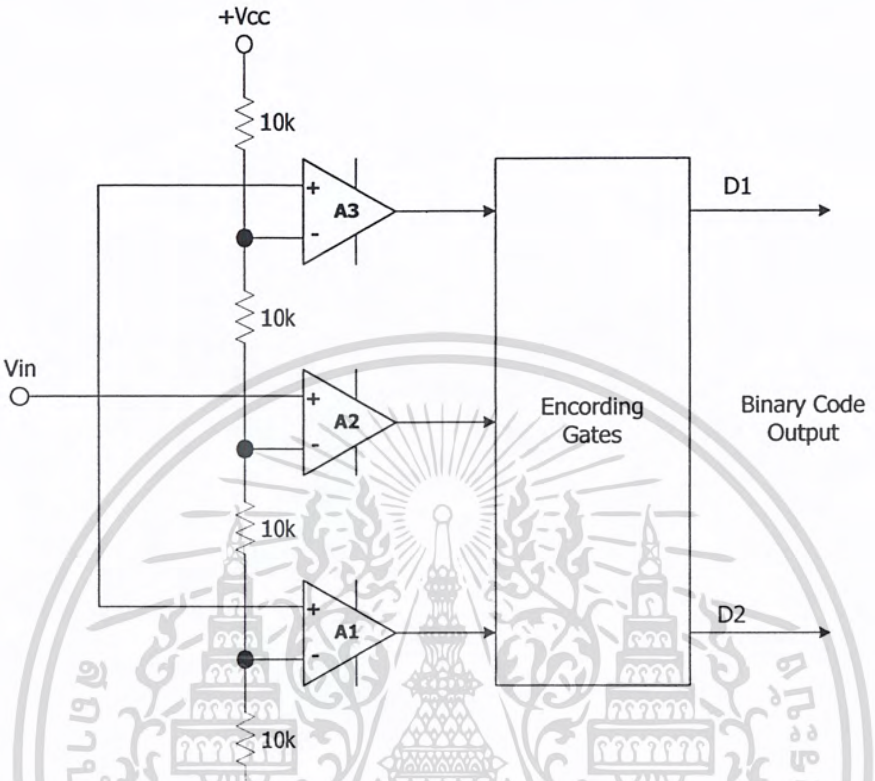


รูปที่ 2.14 ลักษณะของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล

วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล จะมีลักษณะเหมือนกับวงจรแปลงสัญญาณดิจดตแลเป็นอนาล็อกและจะมีคุณสมบัติพิเศษอีก 1 อย่าง คือการเปลี่ยนแปลงเวลา (Conversion Time) ดังนี้หมายถึงช่วงเวลาที่วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล ใช้ในการแปลงสัญญาณอนาล็อกเป็นดิจิตอลได้ 1ค่า วงจรแปลงสัญญาณอนาล็อกเป็นดิจดตอลที่ดีควรเปลี่ยนแปลงเวลาน้อยที่สุดเท่าที่จะน้อยได้ วงจรแปลงสัญญาณอนาล็อกเป็นดิจดตอลมีหลักการได้หลายแบบ โดยจะกล่าวเป็นข้อดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบการเปรียบเทียบแบบขนาน



รูปที่ 2.15 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบการเปรียบเทียบ

จากรูปที่ 2.15 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้ใช้หลักการของการเปรียบเทียบแบบขนาน โดยสัญญาณที่นำมาแปลงนี้จะต่อขนานกับตัวเปรียบเทียบทุกตัวสัญญาณที่จะเข้าแทนที่ขาบวก ส่วนขาลบของตัวเปรียบเทียบจะต่อกับแรงดันอ้างอิงของแต่ละระดับโดยมากได้มาจากการต่อแบ่งของความต้านทาน จากวงจรนี้ถ้าสัญญาณที่มามีค่าเท่ากับจุดเปรียบเทียบใดๆ ก็จะทำให้เอาท์พุทของตัวเปรียบเทียบอันนั้นเป็น 1 เช่นถ้าสัญญาณเข้ามา 2.6 โวลท์ จะทำให้ A1 และ A2 มีค่าเป็น 1 เอาท์พุท A1,A2,A3 นี้ไม่ได้อยู่ในรูปของไบนารี่ ดังนั้นจึงจำเป็นต้องแปลงโดยใช้วงจรทางลอจิกทั่ว ๆไปได้ จุดเด่นของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้ก็คือสามารถทำการเปลี่ยนแปลง (Converter) ได้เร็วมาก ซึ่งค่าทั่วไปอยู่ที่ 20 นาโนวินาที เท่านั้น แต่ข้อเสียก็คือจะต้องสิ้นเปลืองฮาร์ดแวร์เป็นจำนวนมาก รูปที่ 1 คือวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 2 บิท เท่านั้น ซึ่งต้องใช้ตัวเปรียบเทียบถึง 3 ตัวเขียนเป็นสูตรได้ดังนี้

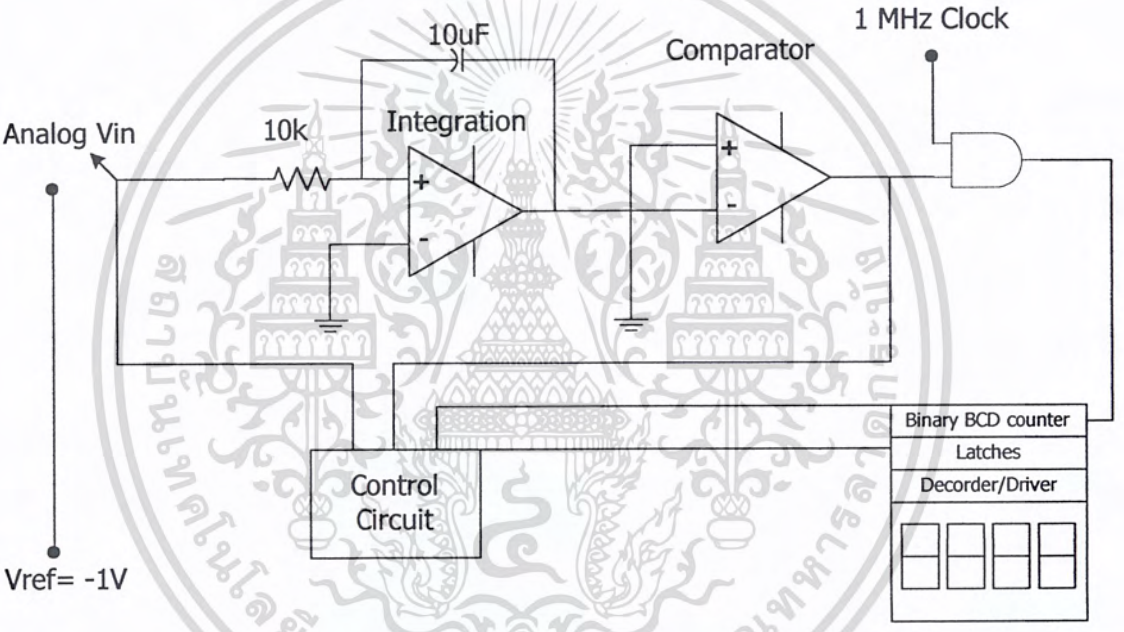
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2^n-1

โดย n คือ จำนวนบิทที่ต้องการ

หมายความว่าถ้าเราต้องการ วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล ขนาด 8 บิท ก็จะต้องใช้ตัวเปรียบเทียบถึง 255 ตัว จึงทำให้วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้ไม่ค่อยที่นิยมมากนัก

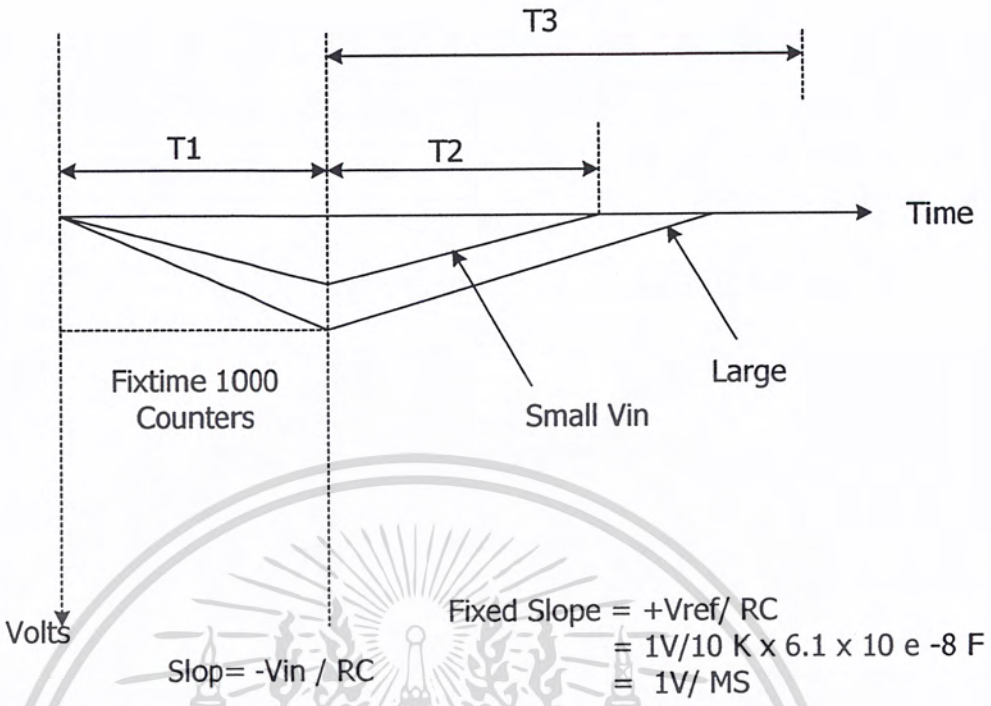
2.2.2 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบดูอัลสโคป



รูปที่ 2.16 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบดูอัลสโคป

วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้ จะใช้กับดิจิตอลมิเตอร์เนื่องจากมีราคาถูก ในขณะที่สามารถให้รายละเอียดได้ดี การทำงานเริ่มต้นด้วยการรีเซ็ทตัวนับ ให้มีค่าเป็น 0 และต่อสัญญาณอนาล็อกเข้ากับอินติเกรเตอร์ (Integrator) เราสมมติว่ามีแรงดันอินพุทมีค่าเป็นบวกเอาท์พุทจากขาอินติเกรเตอร์จะแสดงดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

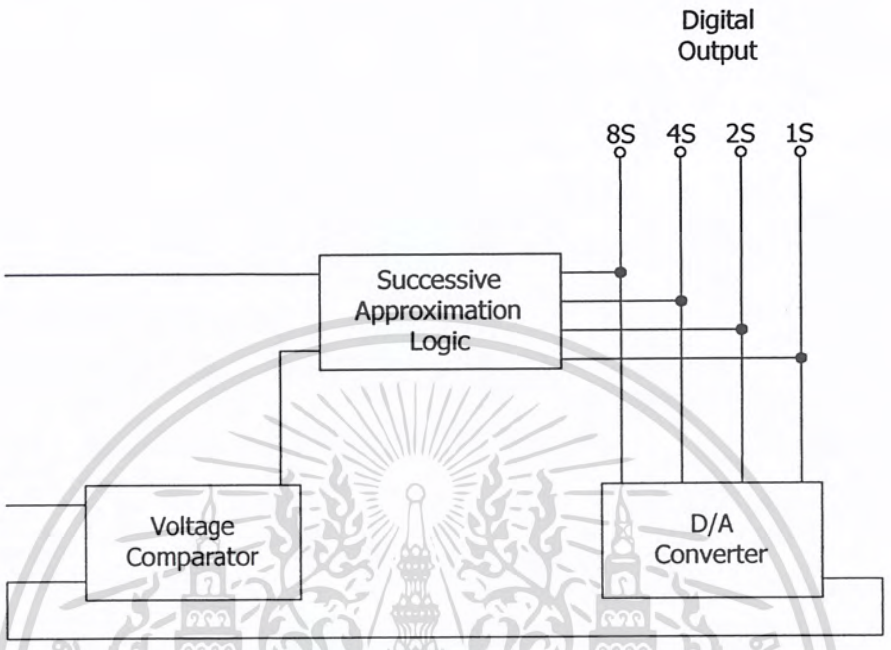


รูปที่ 2.17 เอาท์พุทที่ได้จากอินติเกรเตอร์

เมื่อค่าเอาท์พุทจากอินติเกรเตอร์มีค่าเป็นลบ จะทำให้คอมพิวเตอร์ทำงานส่งผลให้แอนด์เกทปิด และจะทำให้สัญญาณนาฬิกาสามารถผ่านไปยังระบบนับได้ ค่าเอาท์พุทจากอินติเกรเตอร์นี้จะถูกกำหนดระบุเวลา(Fixed Time) จากวงจรควบคุมเมื่อถึงจุดนี้จะทำให้รีเซ็ตนับอีกครั้งและต่อจากสัญญาณอินพุทของอินติเกรเตอร์เข้ากับแรงดันอ้างอิง (V_{REF}) ที่เป็นค่าลงซึ่งยังส่งผลให้เอาท์พุทของอินติเกรเตอร์มีสโลปมากขึ้นเป็นบวก ค่าสโลปตั้งแต่ขึ้นจนกระทั่งถึงค่า 0 นี้เอง ที่จะทำให้ตัวนับ นับความถี่ได้จะสัมพันธ์กับแรงดันอินพุทเริ่มแรกที่วัดได้ วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้มักจะอยู่ในรูปของไอซีสำเร็จรูป และใช้งานด้านดิจิตอลโวลท์มิเตอร์ ซึ่งโดยทั่วไปจะมีการเปลี่ยนแปลงเวลาประมาณ 300 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบ Successive Approximate



รูปที่ 2.18 ผังไดอะแกรมของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบ Successive Approximate

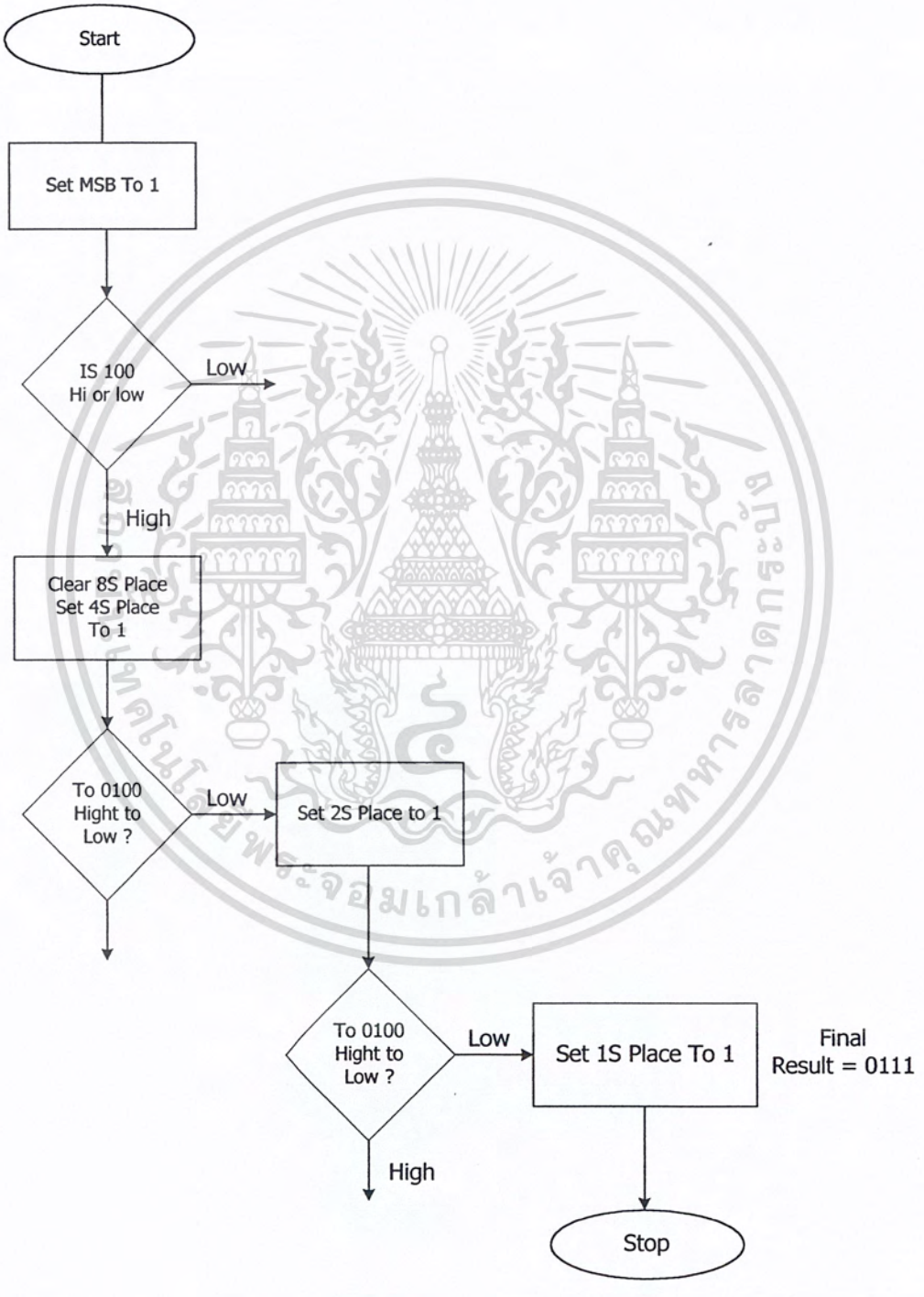
วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้ประกอบด้วย วงจรเปรียบเทียบแรงดัน วงจรเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอล และวงจร Successive Approximate Logic ดังแสดงในรูปที่ 2.18 และแผนภูมิแสดงลำดับการทำงานดังรูปที่ 2.19

การทำงานตามรูปแผนภูมิในรูปที่ 2.19 อธิบายได้ดังนี้

สมมติว่าแรงดันอนาล็อกที่อินพุทเท่ากับ 7 โวลท์ วงจรจะต้องให้บิต MSB เป็นลอจิก "1" ทำให้ได้รหัสไบนารี "1000"ป้อนผ่านวงจรแปลงสัญญาณดิจิตอลเป็นอนาล็อกไปยังอินพุทของวงจรเปรียบเทียบแรงดัน ซึ่งจะเปรียบเทียบว่าแรงดันที่ป้อนกลับสูงหรือต่ำกว่าแรงดันที่อินพุท ถ้าแรงดันเปรียบเทียบสูงกว่าวงจร Successive Approximate Logic ก็จะเคลียร์บิต MSB หรือบิต 2^2 ให้เป็นลอจิก "1" ผลที่ได้คือรหัส"0100" ซึ่งจะเปลี่ยนแรงดันป้อนกลับไปเปรียบเทียบต่ำกว่าแรงดันที่อินพุท วงจรลอจิกก็จะเซ็ตบอร์ดถัดไปคือ บิตที่ 2^1 ให้เป็นลอจิก"1" ผลที่ได้คือรหัส "0110" แล้วนำไปเปรียบเทียบใหม่ ถ้าแรงดันเปรียบเทียบยังต่ำกว่าแรงดันที่อินพุตอีก วงจรลอจิกก็จะเซ็ตบิทถัดไปคือบิทที่ 2^0 ให้เป็นลอจิก "1" ผลที่ได้คือรหัส "0111" เมื่อนำไปเปรียบเทียบกับแรงดันที่อินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรากฏว่าเท่ากันกับวงจรเปรียบเทียบ ก็จะควบคุมให้วงจร Successive Approximate Logic หยุดการทำงาน ข้อมูลรหัสไบนารีที่อยู่ในวงจรลอจิกควบคุมก็คือ “0111” ซึ่งจะเป็นรหัสไบนารีแทนค่าแรงดันอนาล็อกที่อินพุท 7 โวลท์นั่นเอง



รูปที่ 2.19 แผนภูมิแสดงการทำงานของวงจร Successive Approximation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบนี้คือจะใช้เวลาในการเปลี่ยนสัญญาณเร็วมาก เพราะไม่ต้องใช้วิธีนับเรียงลำดับไปเรื่อย ๆ เหมือนกับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบอื่นๆ วงจร Successive Approximation Logic จึงเป็นที่นิยมใช้อย่างมากมายและกว้างขวาง

2.3 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงเป็นทรานสดิวเซอร์แรงบิด ซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษคือ แรงบิดของเพลามอเตอร์ไฟฟ้ากระแสตรงจะเป็นสัดส่วนโดยตรงกับกระแสอาร์มาเจอร์ แรงบิดของเพลาของมอเตอร์ไฟฟ้ากระแสตรงจะได้ผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการนี้แสดงได้ในรูปที่ 2.1 ในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วยเส้นแรงแม่เหล็ก (ϕ) และขดลวดตัวนำเส้นนั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลาและกระแสเท่ากับ

$$T = k\phi I \qquad \text{---(2 - 1)}$$

เมื่อ T คือ แรงบิดของเพลามีหน่วยเป็นนิวตัน – เมตร

ϕ คือ เส้นแม่เหล็กมีหน่วยเป็นเวเบอร์

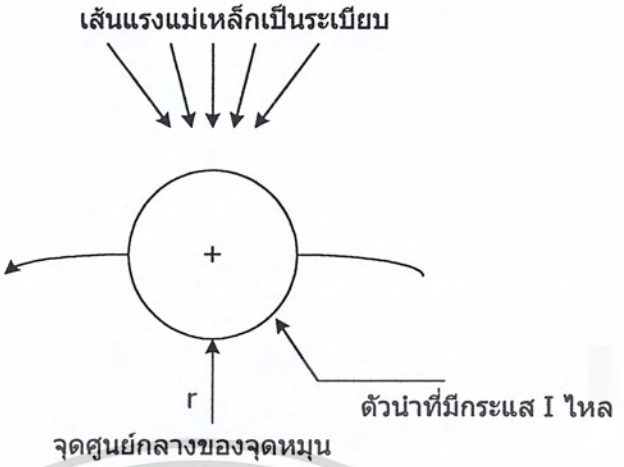
I คือ กระแสเป็นแอมแปร์

k คือ ค่าคงที่

ดังนั้นแรงบิดของเพลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส เมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดแรงดันตกคร่อมตัวมันเอง แรงดันนี้จะเป็นสัดส่วนกับความเร็วของเพลาของมอเตอร์และต้านการไหลของกระแส ความสัมพันธ์ระหว่างแรงดันย้อนกลับนี้และความเร็วของเพลามอเตอร์คือ

$$K = k\phi\omega \qquad \text{---(2 – 2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 เกิดแรงบิดในตัวมอเตอร์ไฟฟ้ากระแสตรง

เมื่อ E คือ แรงดันย้อนกลับ (emf) มีหน่วยเป็นโวลท์

ϕ คือ เส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์

ω คือ ความเร็วของมอเตอร์มีหน่วยเป็นเรเดียนต่อวินาที

สมการที่ (2–1)และ(2–2) เป็นสมการที่แสดงถึงหลักการทำงานพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง

2.3.1 การแยกประเภทของมอเตอร์ไฟฟ้ากรแสตรง

มอเตอร์ไฟฟ้ากระแสตรงสามารถแบ่งออกได้หลายประเภท ขึ้นอยู่กับลักษณะวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์แลขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาร์มาเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกออกได้เป็น 2 แบบ คือ

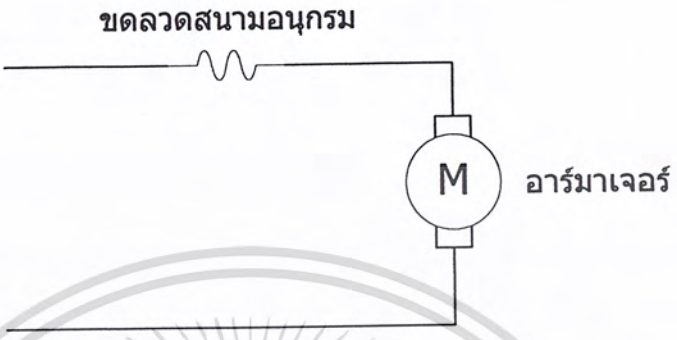
2.3.1.1 มอเตอร์ไฟฟ้ากระแสตรงแบบปรับเส้นแรงแม่เหล็กได้

มอเตอร์ไฟฟ้ากระแสตรงแบบปรับเส้นแรงแม่เหล็กแบ่งแยกได้เป็น 2 แบบคือ

- แบบขดลวดสนามแม่เหล็กต่ออนุกรมกับขดลวดอามาเจอร์
- แบบขดลวดสนามแม่เหล็กแยกกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กอนุกรมแสดงดังรูปที่ 2.21



รูปที่ 2.21 มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์มาเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก

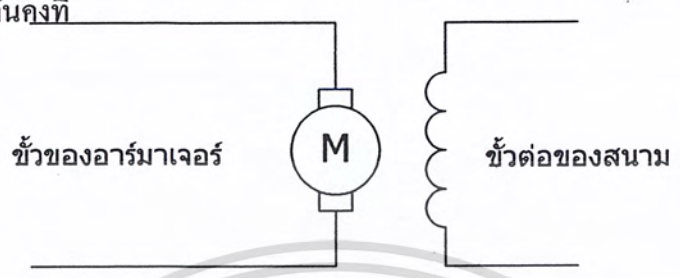
แบบนี้จะมีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแส ดังนั้นเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ และเราจะได้ความสัมพันธ์ระหว่างความเร็วและแรงบิดเป็นแบบไม่เชิงเส้น (non linear) ดังแสดงในรูปที่ 2.22

รูปที่ 2.22 คุณสมบัติระหว่างความเร็วและแรงบิดของมอเตอร์ไฟฟ้ากระแสตรงอนุกรมภายใต้ภาวะแรงดันคงที่

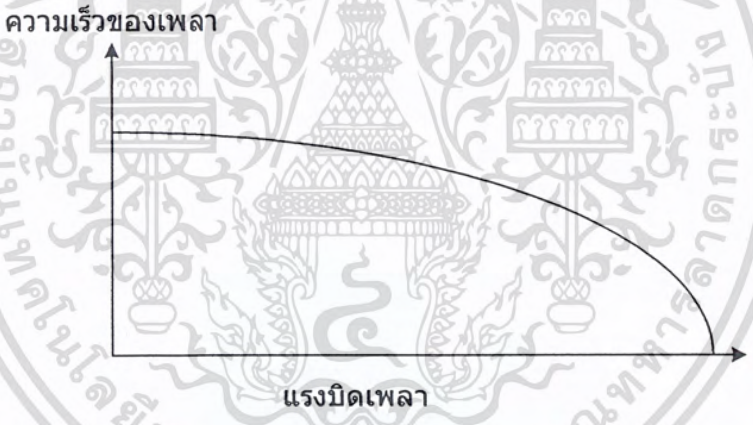
มอเตอร์ดังกล่าวจะใช้งานภาวะเฉพาะเมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดที่ความเร็วสูง เช่น ระบบขับเคลื่อนของรถลาก ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กกระตุ้นแสดงดังในรูปที่ 2.23 มอเตอร์ไฟฟ้ากระแสตรงแบบนี้มักนิยมเรียกกันว่า มอเตอร์ขนาน (Shunt motor) มอเตอร์แบบนี้สามรถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์มาเจอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังผลให้สามารถควบคุมพารามิเตอร์ของมอเตอร์ให้มีค่าคง ที่ได้ตลอดช่วงพิสัยที่กว้าง มอเตอร์นี้มักจะใช้งานในกรณีระบบบังคับการเคลื่อนที่ที่ต้องการแรงบิดสูง ในรูปที่ 2.24 แสดงถึงคุณสมบัติระหว่างแรงบิดกับความเร็วของมอเตอร์ขนานภายใต้ภาวะการกระตุ้นสนามแม่เหล็กคงที่ และอาร์มาเจอร์แรงดันคงที่



รูปที่ 2.23 มอเตอร์ไฟฟ้ากระแสตรงแบบแยกปรับสนามแม่เหล็กได้

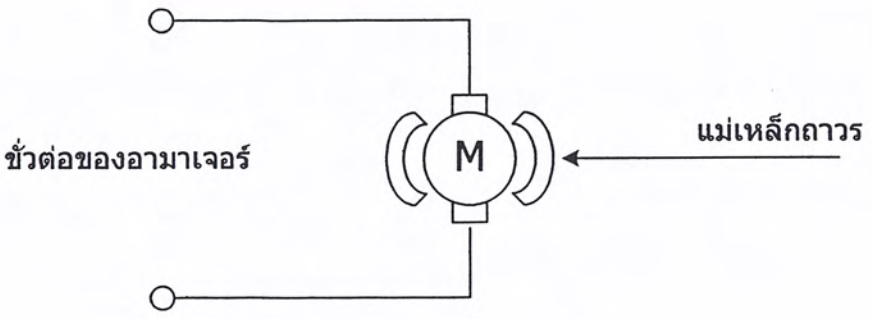


รูปที่ 2.24 คุณสมบัติระหว่างความเร็วและแรงบิดของมอเตอร์ขานภายใต้ภาวะอาร์มาเจอร์แรงดันคงที่และการกระตุ้นสนามแม่เหล็กคงที่

2.3.1.2 มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กคงที่

ระบบกระตุ้นฟิลด์ของมอเตอร์ทั่วไปในปัจจุบัน มักใช้แบบแม่เหล็กถาวรดังแสดงในรูปที่ 2.25 ในระบบนี้เส้นแรงของฟิลด์มีค่าคงที่ ระบบนี้จะให้ความสัมพันธ์ระหว่าง กระแสอาร์มาเจอร์ แรงบิดและความเร็วอยู่ในลักษณะเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 มอเตอร์ไฟฟ้ากระแสตรงแบบฟิลด์เป็นแม่เหล็กถาวร

2.3.2 ข้อดีของมอเตอร์แบบฟิลด์แม่เหล็กถาวรซึ่งเหนือกว่ามอเตอร์แบบมีโครงสร้างฟิลด์

ด้วยการพันของขดลวดคือ ไม่มีกำลังสูญเสียในฟิลด์ มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์ที่มีขนาดของกำลังแรงม้าเท่ากัน นอกจากนั้นความสัมพันธ์เชิงเส้นในสมการที่ 2 ยังให้ค่ากระแสอาร์มาเจอร์สูงกว่ามอเตอร์ไฟฟ้ากระแสตรงแบบฟิลด์เป็นขดลวดการประยุกต์ใช้งานเหมาะสำหรับระบบที่ต้องการแรงบิดของโหลดสูง

ถ้าพิจารณาแยกประเภทตามลักษระการออกแบบโครงสร้างอาร์มาเจอร์สามารถแยกออกได้เป็น 3 แบบคือ

- มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์มาเจอร์เป็นแกนเหล็ก
- มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์มาเจอร์ที่มีขดลวดพันอยู่บนพื้ผิว
- มอเตอร์ไฟฟ้ากระแสตรงแบบอาร์มาเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีมอเตอร์ไฟฟ้ากระแสตรงชนิดพิเศษอีกแบบหนึ่งคือ แบบไม่มีแปรงถ่าน (Brushless dc motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับมีมอเตอร์ไฟฟ้ากระแสตรงชนิดมีมอเตอร์กระแสตรงชนิดมีแปรงถ่าน ยกเว้นการคอมมิวเตชั่น (Commutation) กระทำโดยทางเทคนิคทางอิเล็กทรอนิกส์แทนทีจะกระทำโดยวิธีเชิงกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

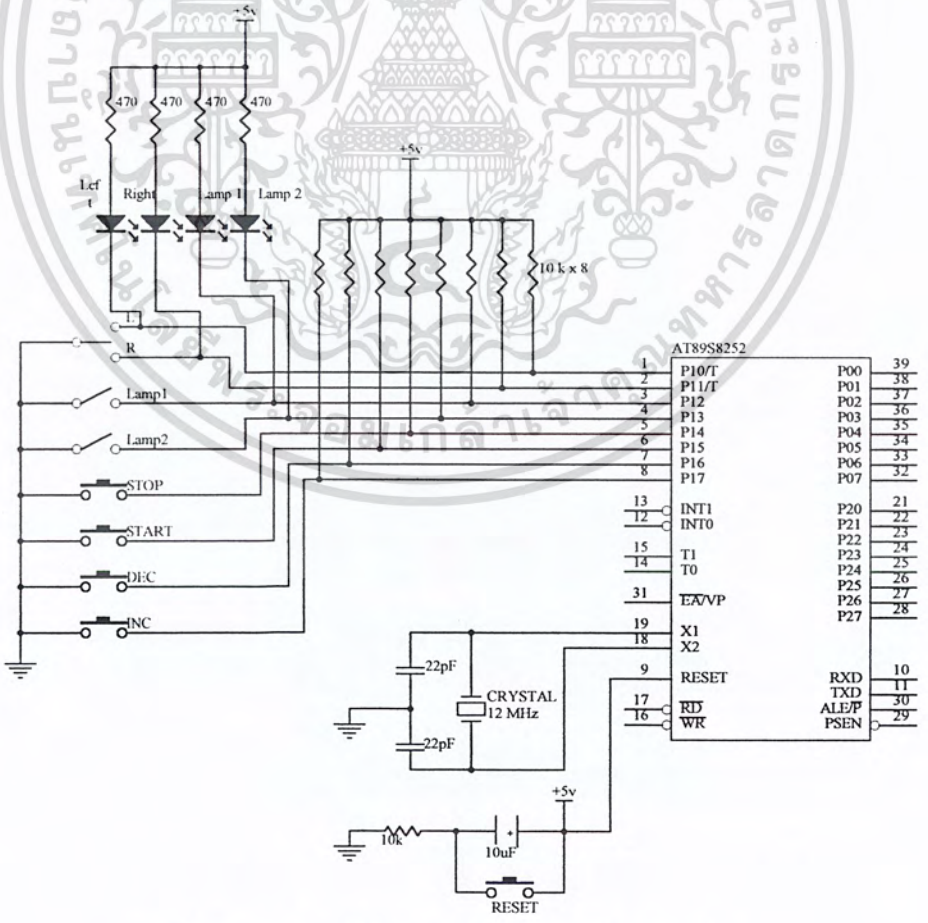
บทที่ 3
การออกแบบและโครงสร้าง

การออกแบบวงจรของชุดควบคุมในส่วนของภาคส่งนี้ แบ่งการออกแบบ 2 ส่วนใหญ่ ๆ คือ ส่วนที่เป็นฮาร์ดแวร์วงจรและซอฟท์แวร์ควบคุมวงจรโดยฮาร์ดแวร์จะใช้ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สำคัญในการประมวลผลและควบคุมการทำงาน ส่วนไมโครคอนโทรลอเลอร์ที่ใช้ในโครงงานนี้จะเลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 เนื่องจากเบอร์นี้มีหน่วยความจำโปรแกรม 8 Kbyte ซึ่งเพียงพอกับโครงงานนี้ ส่วนทางด้านซอฟท์แวร์ควบคุมวงจรนั้นใช้ภาษาแอสแซมบลี้ของ MCS-51 ในการเขียนโปรแกรม

3.1 การออกแบบชุดควบคุม HARD WARE ภาคส่ง

วงจรของชุดควบคุมภาคส่งประกอบไปด้วยส่วนต่างๆดังนี้

3.1.1ส่วนประมวลผล



รูปที่ 3.1 การต่อขาใช้งานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่และการต่อขาต่างๆ กับอุปกรณ์ภายนอกของไมโครคอนโทรลเลอร์ ในส่วนของภาคส่งที่ทำหน้าที่ตัวควบคุมจะมีดังนี้

ขา 1 P1.0 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ L เพื่อกำหนดให้มอเตอร์มีทิศทางการหมุนไปทางซ้าย

ขา 2 P1.1 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ R เพื่อกำหนดให้มอเตอร์มีทิศทางการหมุนไปทางขวา

ขา 3 P1.2 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ L1 เพื่อกำหนดการเปิดหรือปิดไฟดวงที่ 1

ขา 4 P1.3 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ L2 เพื่อกำหนดการเปิดหรือปิดไฟดวงที่ 2

ขา5 P1.4 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ STOP เพื่อกำหนดให้หยุดการส่งข้อมูลแบบอนุกรม

ขา 6 P1.5 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ START เพื่อกำหนดให้ส่งข้อมูลแบบอนุกรม

ขา 7 P1.6 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ DECREASE เมื่อกดแล้ว จะทำหน้าที่ในการลดค่าตำแหน่งของภาครับ

ขา 8 P1.7 บิตนี้ทำหน้าเป็นอินพุทเพื่อรับคำสั่งจากสวิทช์ INCREASE เมื่อกดแล้ว จะทำหน้าที่ในการเพิ่มค่าตำแหน่งของภาครับ

ขา 9 RESET เป็นขารีเซ็ทของวงจร โดยจะต่อตัวต้านทานลงกราวด์ และต่อตัวเก็บประจุคร่อมสวิทช์ที่ต่อกับไฟ +5V

ขาที่ 11 RTX ใช้ทำหน้าที่เป็นตัวส่งข้อมูลอนุกรม เพื่อส่งไปยังตัวภาครับ

ขาที่ 17 RD เป็นขา READ โดยต่อกับขา 2 ของวงจร A/D

ขาที่ 18 XTAL1 เป็นขาที่ต่อกับชุดออสซิลเลเตอร์ โดยใช้ CRYTAL 12MHz และต่อ C 20 pF ลงกราวด์

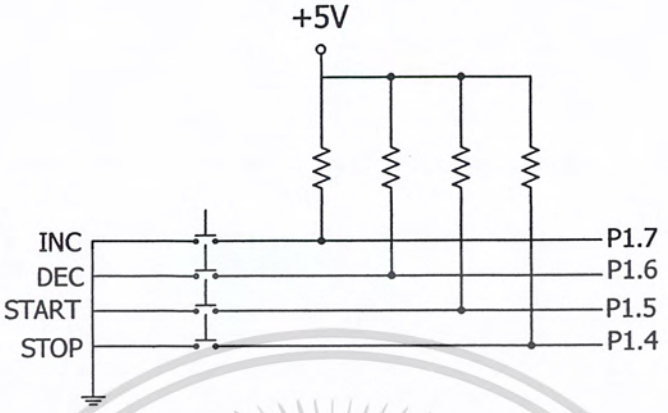
ขาที่ 19 XTAL2 เป็นขาที่ต่อกับชุดออสซิลเลเตอร์ โดยใช้ CRYTAL12MHz และต่อ C 20 pF ลงกราวด์

ขาที่ 32-39 PORT 0 เป็น I/P พอร์ทที่รับข้อมูลมาจากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลโดยการปรับ VR และต่อตัวต้านทาน PULL Up ไว้

ขาที่ 21-28 PORT 2 ทำหน้าที่เป็น Output เพื่อไปยังส่วนของวงจรแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ภาคคีย์สวิทช์ทางด้านอินพุทของภาครับ

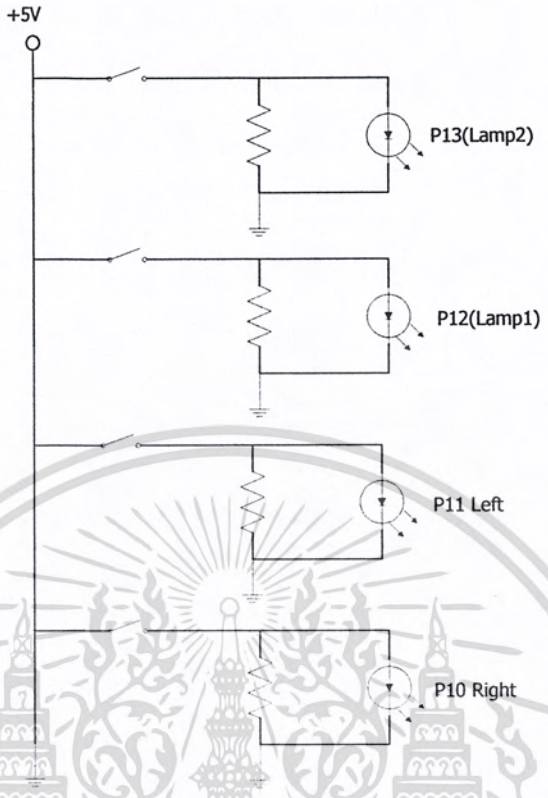


รูปที่ 3.2 รูปคีย์สวิทช์เลือก Slave

รูปที่ 3.2 นั้นได้แสดงสวิตซ์ชุดควบคุมรับโดยมีสวิตซ์ Increase ต่อที่ขา P1.4 จะเป็นการเพิ่มค่าตำแหน่งในการเลือกชุดควบคุมภาครับและสวิตซ์ Decrease ต่อกับที่ขา P1.6 จะเป็นการลดค่าในการเลือกชุดควบคุมภาครับ

เมื่อทำการเลือกตำแหน่งของภาครับได้แล้วก็ทำการกดสวิตซ์ Start อยู่ที่ขา P1.5 เพื่อให้มีการเริ่มส่งข้อมูลผ่านทางพอร์ทอนุกรมเพื่อส่งข้อมูลไปยังชุดควบภาครับ ณ ตำแหน่งที่กำหนดไว้ โดยการส่งข้อมูลอนุกรมจะใช้ในมาตรฐานการรับส่งแบบ UART เมื่อต้องการสั่งให้มีการหยุดการส่งข้อมูลผ่านทางพอร์ทอนุกรม ทำได้โดยการกดสวิตซ์ Stop อยู่ที่ P1.4 จะกดเพื่อให้มีการหยุดส่งข้อมูลผ่านทางพอร์ทอนุกรมเพื่อต้องการจะเปลี่ยนตำแหน่งของภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



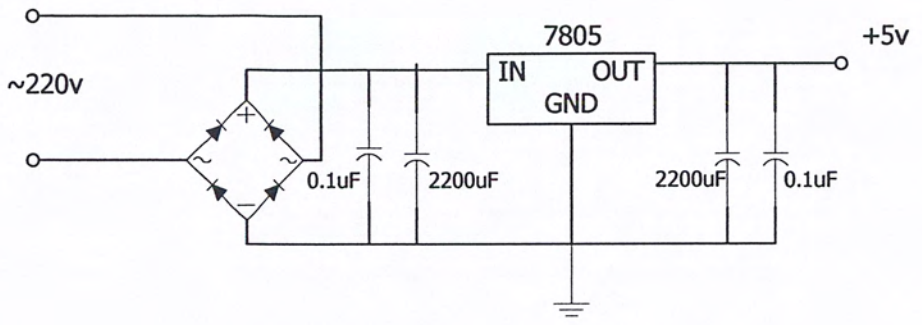
รูปที่ 3.3 คีย์สวิตซ์ควบคุมอุปกรณ์ไฟฟ้า

สวิตซ์อีก 4 ตัว ที่เหลือจะเป็นสวตซ์ที่ใช้กำหนดการทำงานของภาครับซึ่งจะประกอบไปด้วยสวิตซ์ L,R,L1,L2 จะเป็นสวิตซ์เลือกให้มอเตอร์หมุนซ้ายหมุนขวา เปิด-ปิดไฟดวงที่ 1และ 2 ตามลำดับ

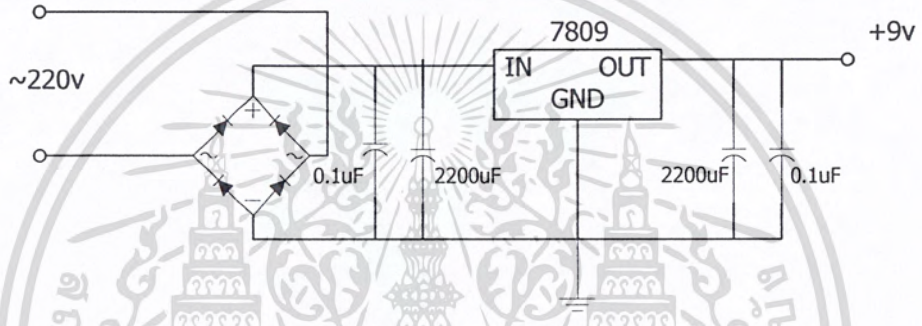
3.1.3 ภาคจ่ายไฟ

ในชุดควบคุมส่วนของภาคส่งนี้จะแบ่งออกเป็นชุดเป็นแหล่งจ่ายไฟ 2 ชุด ได้แก่ +5V และ +9V ซึ่งจะใช้ IC Regulate 2 ตัวคือ #7805,#7809 ตามลำดับเพื่อแปลงแรงดัน 9 โวลท์จากหม้อแปลงให้เป็นแรงดัน 5 โวลท์ โดยที่ใช้สำหรับเป็นไฟเลี้ยงให้กับวงจรค่าต่างๆและ +9V ใช้สำหรับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรภาคจ่ายไฟ +5V

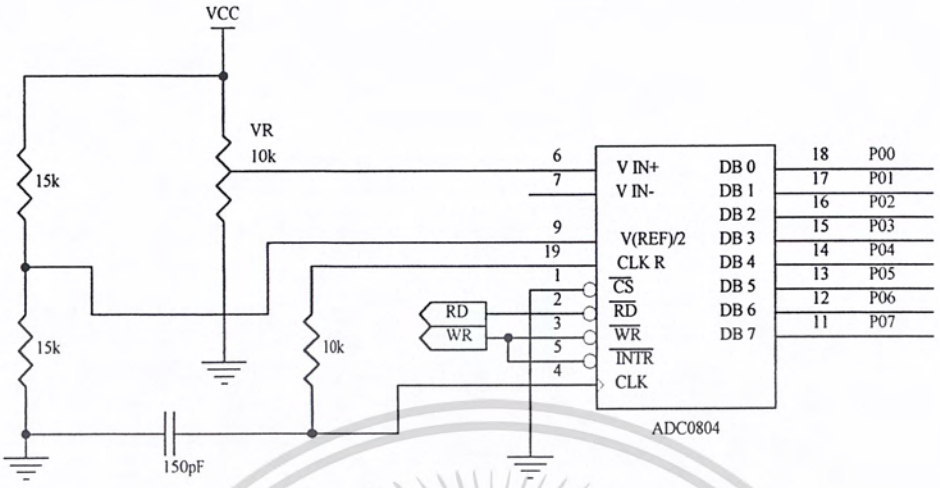


รูปที่ 3.5 วงจรภาคจ่ายไฟ +9V

3.1.4 ภาคควบคุมความเร็วมอเตอร์

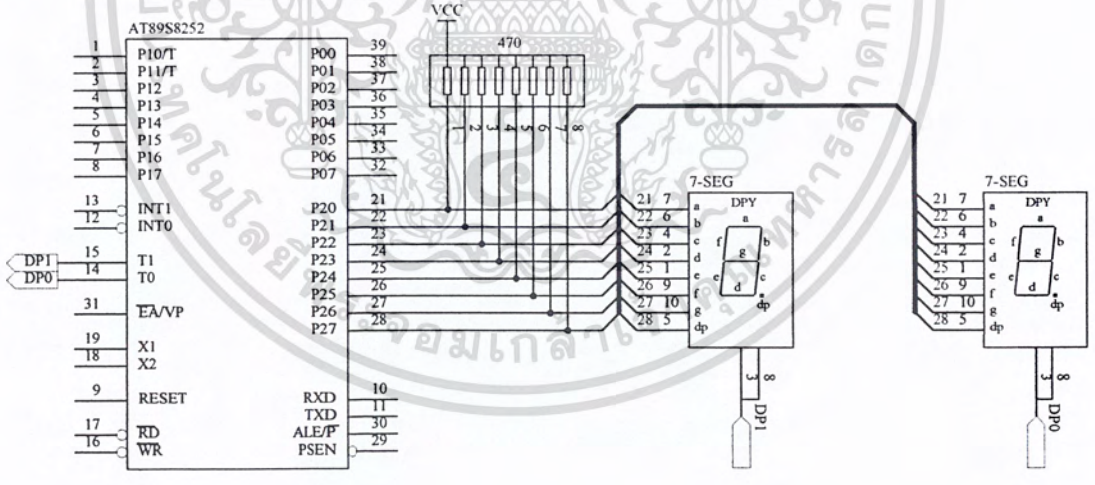
ในส่วนของภาคนี้จะเป็นส่วนการควบคุมความเร็วของมอเตอร์ โดยที่มีตัวต้านทานปรับค่าได้เป็นตัวปรับความเร็วโดยจะเปลี่ยนค่าแรงดันไฟเข้าไปยังวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลแล้วส่งผ่านไปยังไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรควบคุมความเร็วมอเตอร์

3.1.5 ส่วนแสดงผลของภาคส่ง



รูปที่ 3.7 ส่วนแสดงผลของชุดควบคุมภาคส่ง

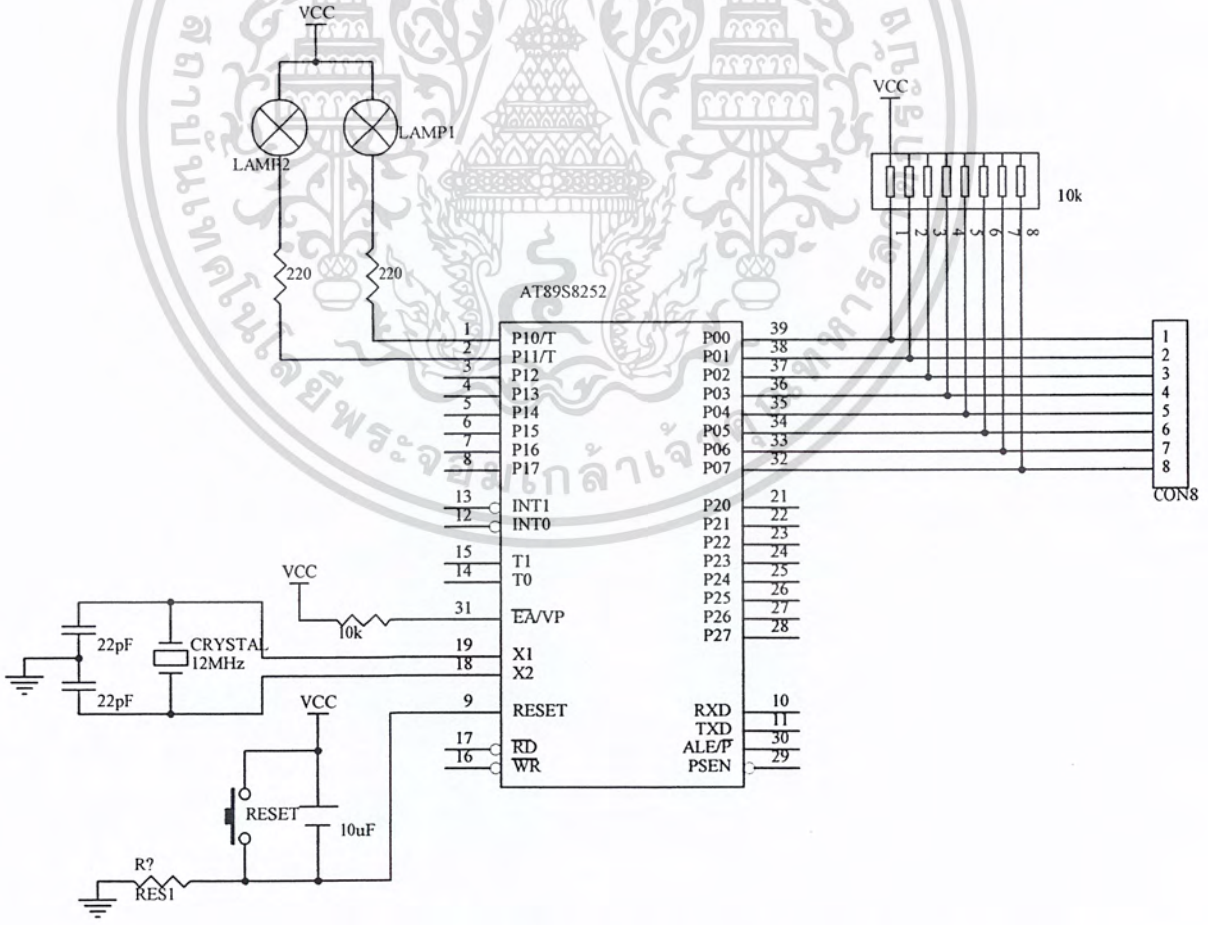
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของวงจรนี้จะเป็นการแสดงผลออกทาง 7-Segment เพื่อเป็นตัวบอกตำแหน่งให้กับผู้ใช้ทราบว่ามีค่าเท่าไร หลังจากที่ทำการกดสวิตซ์เพิ่ม – ลดแล้วจะผ่านส่วนของโปรแกรมที่ทำหน้าที่เปลี่ยนจากเลขฐาน 2 เป็นรหัส BCD แล้วส่งออกเอาต์พุท จากนั้นสัญญาณที่ได้จะส่งไปยังวงจรแปลงรหัส BCD เป็น 7-Segment โดยใช้ไมโครคอนโทรลเลอร์สแกนออกมาแล้วจึงนำออกไปที่ LED 7-Segment 2 หลัก

3.2 การออกแบบ HARD WARE ชุดควบคุมภาครับ

วงจรชุดควบคุมภาครับประกอบไปด้วยส่วนต่าง ๆ ดังนี้

3.2.1 ส่วนประมวลผล



รูปที่ 3.8 ขาการต่อใช้งานของไมโครคอนโทรลเลอร์ภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 1 P1.0 ขานี้จะถูกต่อกับขา 1 ของ 74LS08 เพื่อทำหน้าที่เป็นตัวกำหนดทิศทางการหมุนให้ กับมอเตอร์ว่าจะต้องไปทางซ้าย ซึ่งกำหนดโดยการโยกสวิตซ์ที่ภาคส่ง

ขา 2 P1.1 ขานี้จะถูกต่อกับขา 5 ของ 74LS08 เพื่อทำหน้าที่เป็นตัวกำหนดทิศทางการหมุนให้กับมอเตอร์ว่าจะต้องไปทางขวา ซึ่งกำหนดโดยการโยกสวิตซ์ที่ภาคส่ง

ขา 3 P1.2 ใช้เป็นขา O/P ที่จะนำไปต่อกับไฟต่อกับไฟดวงที่ 1

ขา 4 P1.3 ใช้เป็นขา O/P ที่จะนำไปต่อกับไฟดวงที่ 2

ขา 9 RESET เป็นขารีเซ็ทของวงจร โดยจะต่อตัวต้านทานลงกราวด์ และต่อตัวเก็บประจุคร่อมสวิตซ์ที่ต่อกับไฟ +5V

ขา 10 RTX เป็นขาที่ใช้รับข้อมูลอนุกรมที่ส่งมาจากภาครับ

ขา 18 XTAL1 เป็นขาที่ต่อกับชุด OSCILATOR โดยใช้ CRYSTAL 12 MHz และต่อ C 20 pF ลงกราวด์

ขา 19 XTAL2 เป็นขาที่ต่อกับชุด OSCILATOR โดยใช้ CRYSTAL 12 MHz และต่อ C 20 pF ลงกราวด์

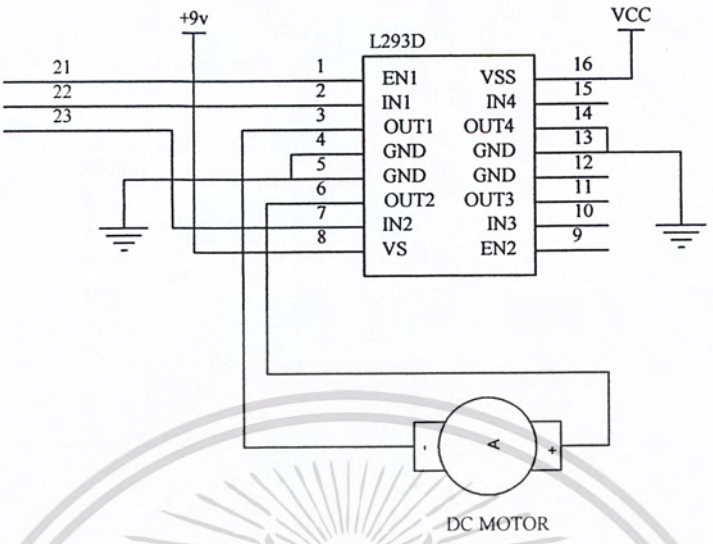
ขา 31 EA จะต่อ +5V เพื่อบอกว่า CPU จะติดต่อกับหน่วยความจำภายในเท่านั้น

ขา 21 P2.0 เป็นเอาท์พุตของวงจร AND GATE ใน PORT นี้บิตอื่นไม่มีการใช้งาน จะถูกต่ออยู่กับขา 2 และ 4 ของ 74LS08

3.2.2 ส่วนของวงจร DRIVER

วงจรส่วนนี้จะใช้ IC #L293D ทำหน้าที่เป็นตัวขับกระแสให้กับมอเตอร์ โดยจะนำเอาขา 3 กับ 6 ของ 74LS08 ที่เป็นเอาท์พุตของวงจร AND GATE มาเข้ากับขาอินพุท คือ ขา 2 และ 7 ตามลำดับ ที่ขา 3 และ 6 จะถูกนำไปต่อกับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



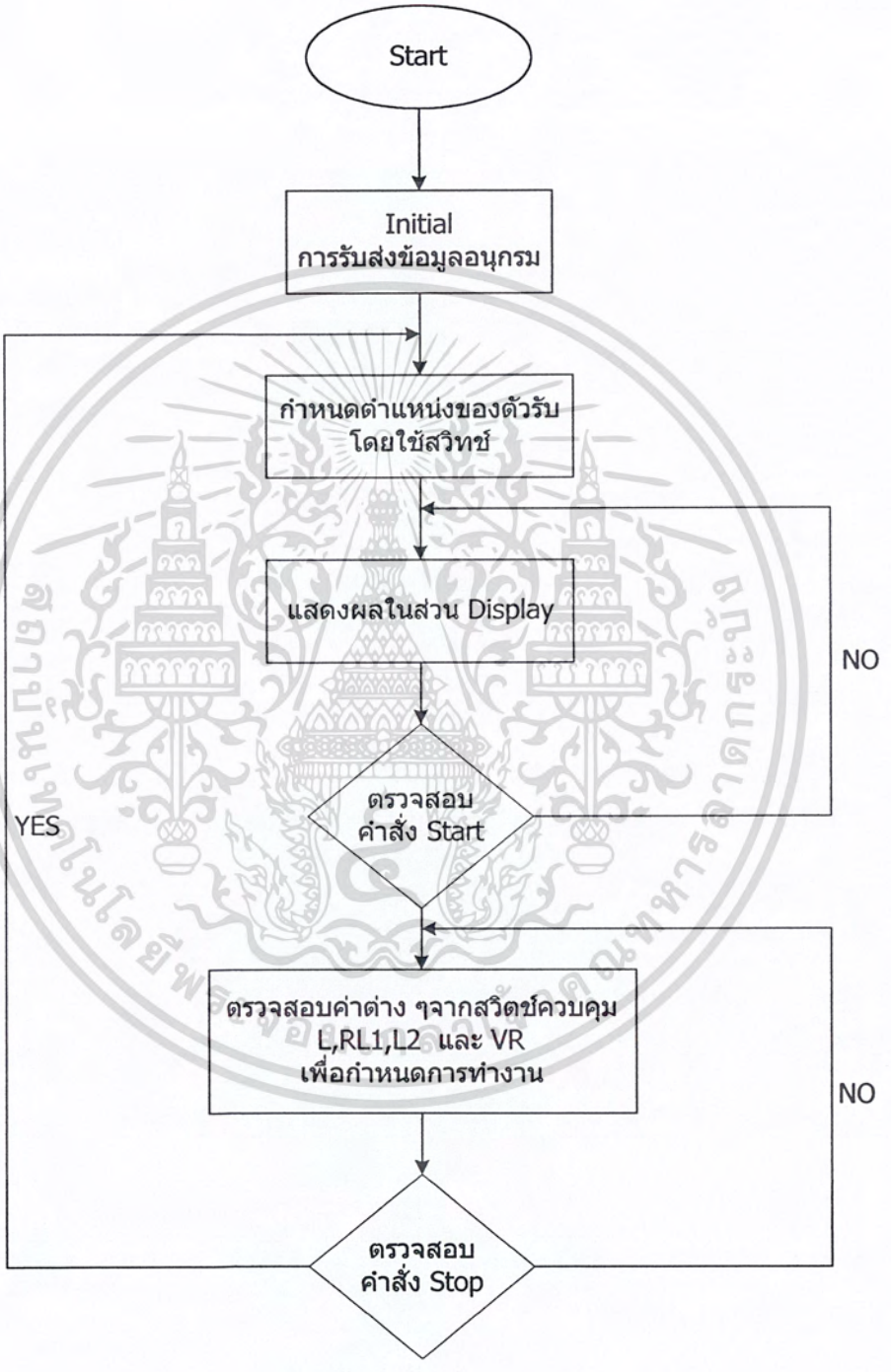
รูปที่ 3.9 วงจรขับกำลังมอเตอร์

3.3 การออกแบบโปรแกรม

ในส่วนของการออกแบบโปรแกรมนั้นได้แบ่งเป็น 2 ส่วนคือ โปรแกรมส่วนของภาคส่งและส่วนของภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

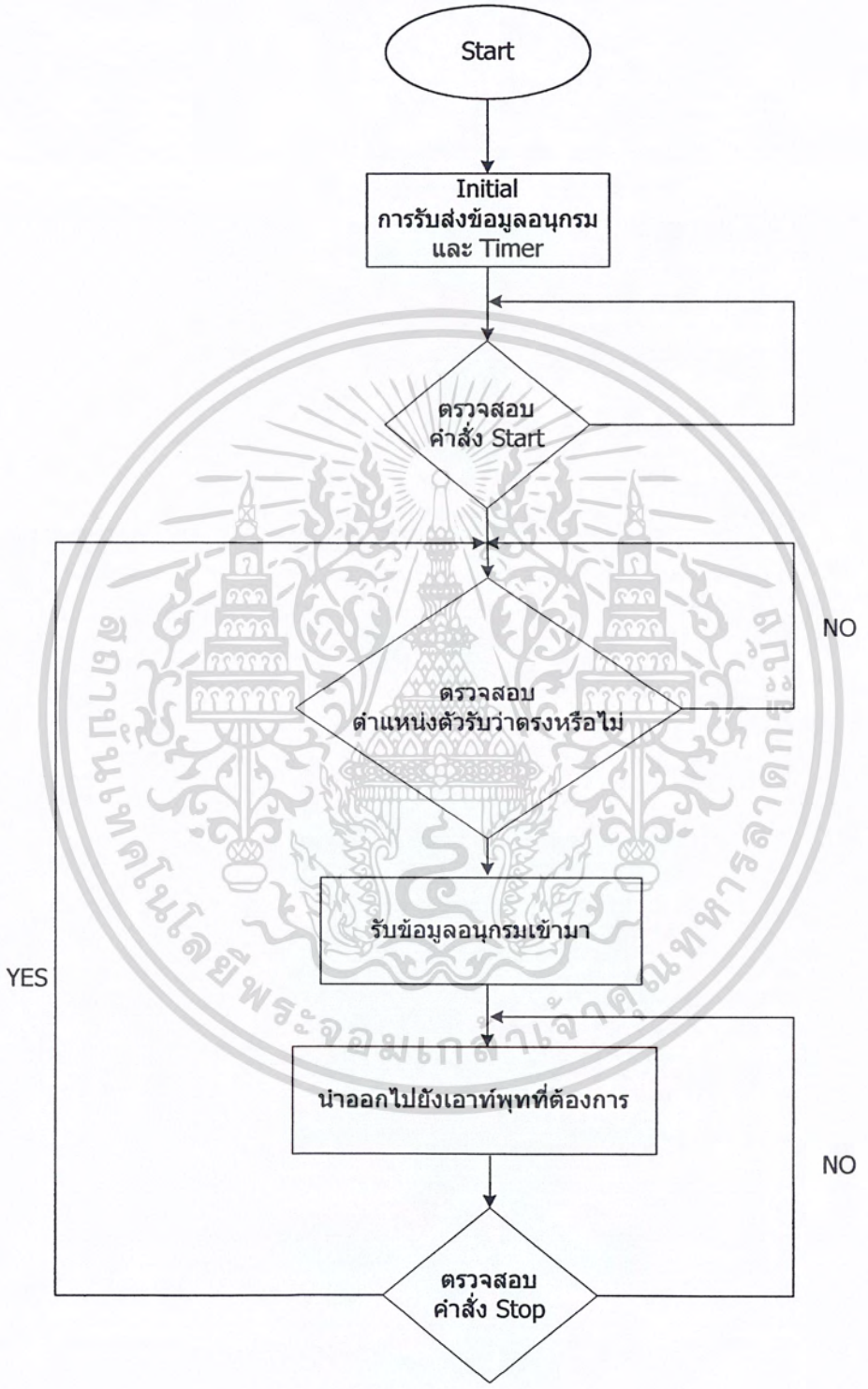
3.3.1 การออกแบบโปรแกรมชุดควบคุมภาคส่ง



รูปที่ 3.10 Flow chart การออกแบบโปรแกรมชุดควบคุมภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การออกแบบโปรแกรมชุดควบคุมภาครับ



รูปที่ 3.11 Flowchart การออกแบบโปรแกรมชุดควบคุมภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4
คุณสมบัติการใช้งานและผลการทดลอง
ชุดควบคุมอุปกรณ์ไฟฟ้า

4.1 คุณสมบัติชุดควบคุมอุปกรณ์ไฟฟ้า

ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอลโทรลเลอร์นี้สามารถที่จะควบคุมอุปกรณ์ไฟฟ้าได้คือ สามารถควบคุมทิศทางในการหมุนและความเร็วในการหมุนของมอเตอร์ และสามารถควบคุมการปิดเปิดไฟในชุดควบคุมตัวรับ การรับส่งข้อมูลผ่านพอร์ทอนุกรม โดยจะสามารถเลือกได้ว่าจะใช้ชุดควบคุมอุปกรณ์ภาครับชุดไหนทำงาน และสามารถที่จะให้การทำงานของแต่ละชุดเป็นอิสระต่อกัน โดยเมื่อเลือกชุดควบคุมภาครับชุดหนึ่งแล้ว ไปเลือกอีกชุดหนึ่ง ชุดที่สั่งให้ทำก่อนนั้นก็ยังคงสภาวะการทำงานได้อยู่เหมือนสภาวะสุดท้ายที่ทำงานอยู่ก่อนที่จะเลือกชุดควบคุมอุปกรณ์ไฟฟ้าภาครับชุดต่อไป

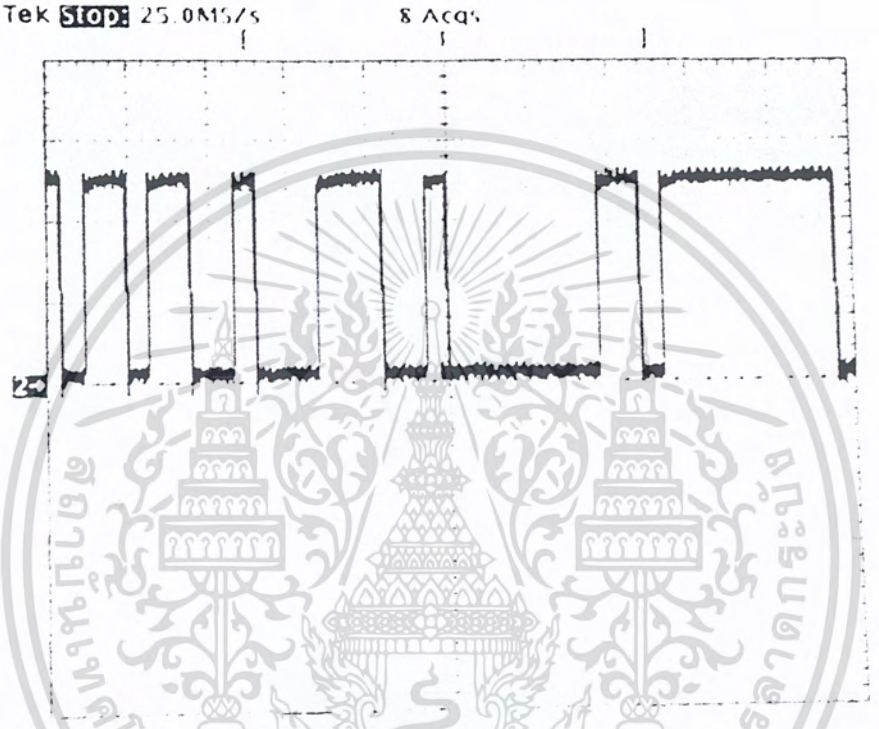
4.2 ขั้นตอนการใช้งานชุดควบคุมอุปกรณ์ไฟฟ้า

1. เปิดสวิตซ์ไฟ POWER ไปที่ ON
2. เลือกคีย์สวิตซ์ว่าจะให้ตำแหน่งของชุดควบคุมภาครับตัวใดทำงาน โดยการ เลือกสวิตซ์ดังนี้
   - INC เพื่อที่จะเพิ่มค่าตำแหน่งชุดควบคุมภาครับให้ทำงาน
   - DEC เพื่อที่จะลดค่าตำแหน่งชุดควบคุมภาครับให้ทำงาน
3. เลือกคีย์สวิตซ์ว่าให้มอเตอร์หมุนไปทางซ้ายหรือทางขวา
4. ปรับตัวต้านทานปรับค่าได้เพื่อเลือกว่าจะให้มอเตอร์หมุนเร็วเท่าไหร่ชุดนี้จะเป็นชุดที่รับส่งข้อมูลผ่านพอร์ทอนุกรมตลอดเวลาที่ทำการเปลี่ยนแปลงค่าที่ตัวต้านทานปรับค่าได้
5. เลือกเปิดสวิตซ์ไฟว่าให้ไฟดวงไหนติด
6. กดคีย์สวิตซ์ START เพื่อที่จะทำการส่งข้อมูลที่ได้รับมาทางคีย์สวิตซ์ต่างๆที่ควบคุมอุปกรณไฟฟ้าผ่านออกไปยังพอร์ทอนุกรมเพื่อส่งให้ภาครับ
7. เลือกการควบคุมตามที่ต้องการในข้อ 3 ข้อ 4 และข้อ 5 ว่าจะให้เป็นอย่างไร
8. หากจะทำการหยุดส่งหรือทำการเปลี่ยนชุดควบคุมภาครับก็ให้ทำการกดคีย์สวิตซ์ STOP เพื่อที่จะเริ่มทำขั้นตอนในข้อ 2 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3วงจรชุดควบคุมภาคส่ง

ทำการทดลองปรับค่าตัวต้านทานปรับค่าได้ ว่ามีผลเปลี่ยนแปลงอย่างไร ทางอินพุทของพอร์ท 0 ของไมโครคอนโทรลเลอร์โดยการดูที่ LED ที่ต่อไว้ในการทดลองว่าเป็นผลอย่างไร

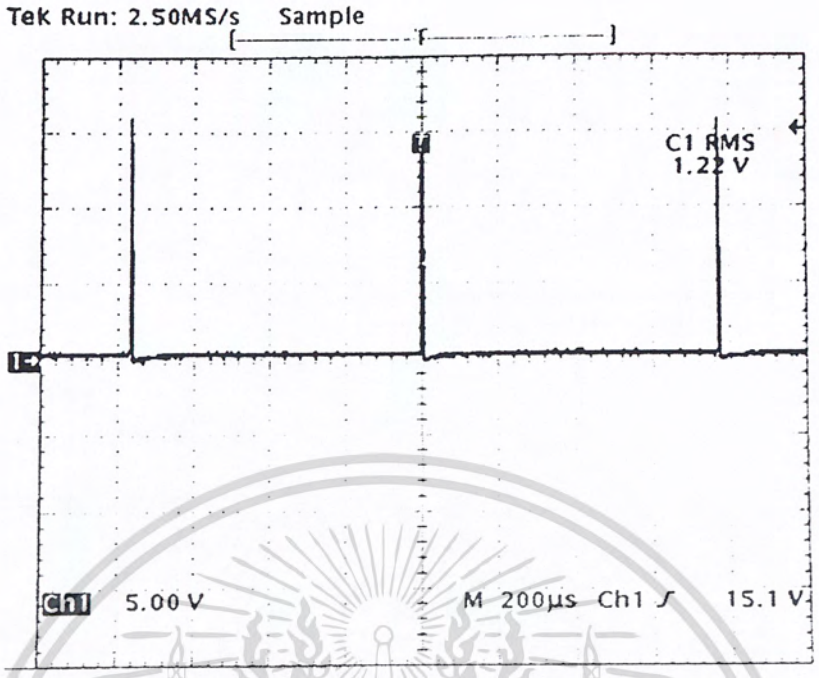


รูปที่ 4.1 สัญญาณที่ได้อินพุทมาทางสวิตซ์และส่งออกทางพอร์ทอนุกรม

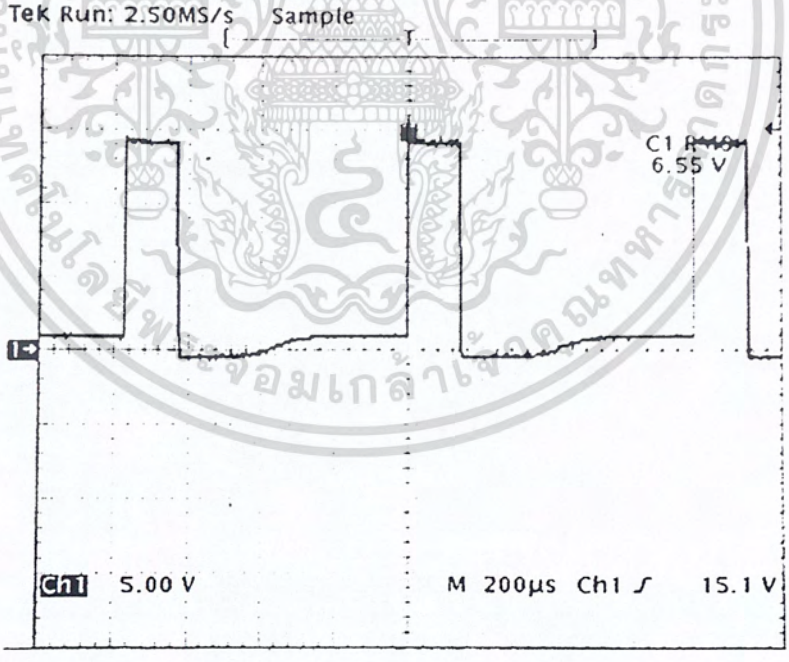
4.4 วงจรขับกระแสไฟมอเตอร์

ทดลองจ่ายไฟให้กับ วงจรขับกำลังงานไฟของ IC L239D ซึ่งเป็นวงจรขับกระแสไฟได้ประมาณ 1 แอมป์ ดูผลการทดลองว่ามอเตอร์สามารถหมุนได้หรือไม่ หากไม่ได้ก็ให้ทำการต่อทรานซิสเตอร์ดาร์ลิงตันแบบบริดจ์เพื่อเพิ่มกระแสไฟก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

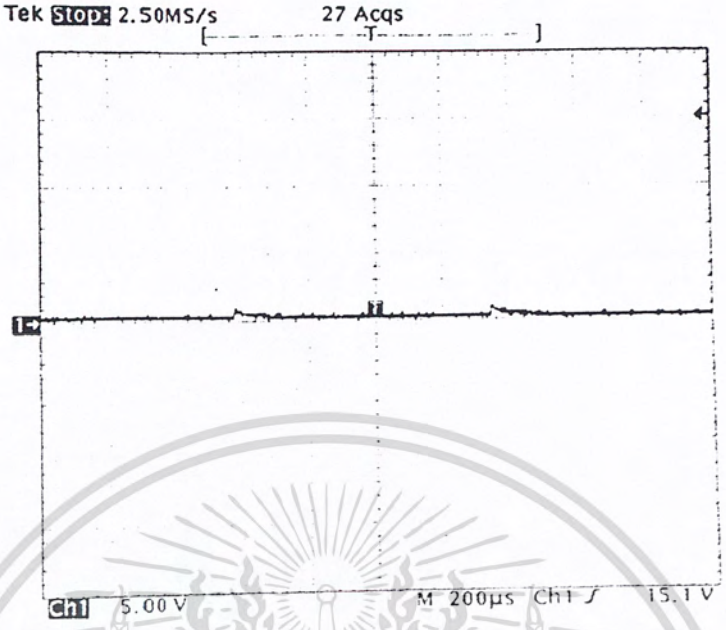


รูปที่ 4.2 สัญญาณ PWM ที่ออกมาทางวงจรขับกระแสขณะที่ปรับความเร็วน้อยที่สุด

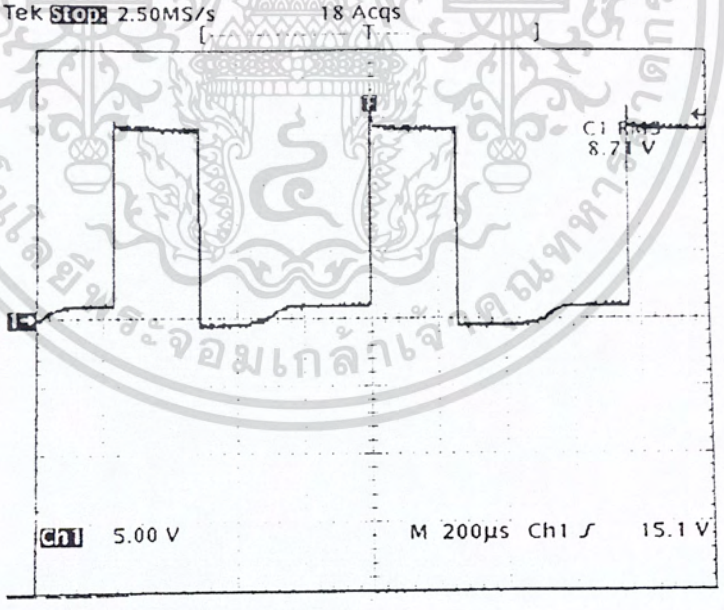


รูปที่ 4.3 สัญญาณ PWM ที่ออกมาทางวงจรขับกระแสขณะที่ปรับความเร็วปานกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 สัญญาณ PWM ที่ออกมาทางวงจรขับกระแสที่ปรับความเร็วมากสุด



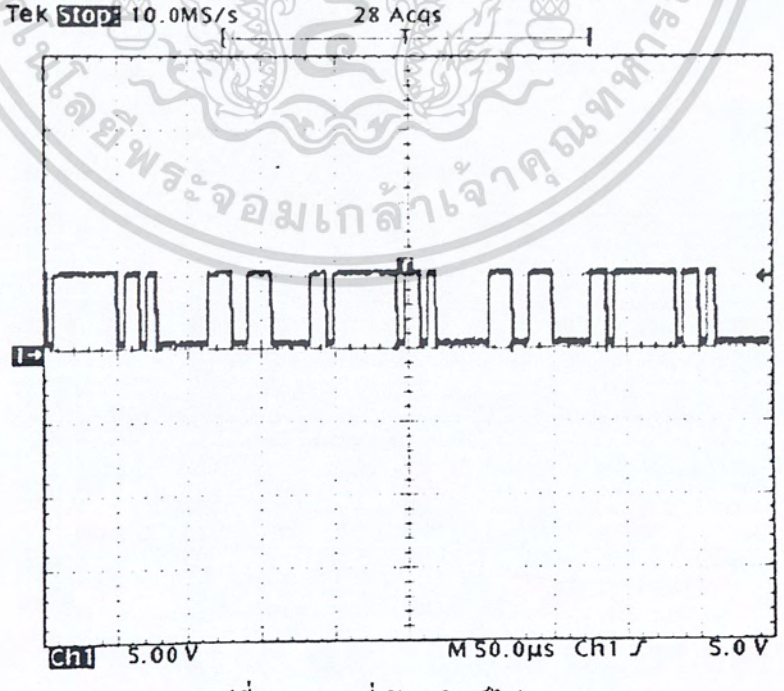
รูปที่ 4.5 รูปสัญญาณที่ออกมาทางที่วงจรขับกระแสขณะที่ไม่มีสัญญาณ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 โปรแกรม

ทำการทดลองรันโปรแกรมว่าเป็นไปตามเงื่อนไขที่เขียนหรือไม่ ถ้าไม่เป็นไปตามที่เขียนให้ทำการทดลองเขียนโปรแกรมแล้วรันใหม่

รูปที่ 4.6 ขณะที่ไม่เปิดสวิตซ์ไฟ L1 และ L2



รูปที่ 4.7 ขณะที่เปิดสวิตซ์ไฟ L1 และ L2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

สรุปผลการทดลอง

ผลการทดลองจากการใช้ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์ยังมีการผิดพลาดของฮาร์ดแวร์เนื่องจากว่าอุปกรณ์บางตัวนั้นมีค่าผิดพลาดเกิดขึ้นรวมไปจนถึงการใช้อุปกรณ์ที่เป็นของเก่าแล้วซึ่งอาจมีการผิดพลาดตรงนี้ได้ แต่อย่างไร ก็ดีชุดควบคุมเครื่องแบบนี้สามารถควบคุมทั้งทิศทางและความเร็วในการหมุนของมอเตอร์ 2 ตัวหรือมากกว่านั้นก็ได้ในกรณีที่ต้องการชุดควบคุมภาครับเพิ่มขึ้น แต่อย่างๆไรก็ดี ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์ชุดนี้มีขีดความสามารถในการควบคุมภาครับได้ 256 ชุด

แนวทางการพัฒนาโครงงาน

แนวทางการพัฒนาโครงงาน สามารถนำไปประยุกต์ใช้ในการควบคุมทิศทางการหมุนของมอเตอร์ในระบบโรงงานอุตสาหกรรมหรืออาจจะนำไปประยุกต์ใช้กับการขนส่งอาทิเช่น รถราง รถไฟฟ้า เพราะโครงงานนี้ส่งผ่านพอร์ทอนุกรมทำให้ใช้สายสัญญาณในการควบคุมมีแค่ 2 เส้น และไม่จำเป็นต้องมีคนควบคุมบนตัวรถเพราะสามารถที่จะควบคุมที่สถานีใหญ่ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.อุดม จีนประดับ. ไมโครคอนโทรลเลอร์ MCS-51 .กรุงเทพฯ: ศูนย์ผลิตตำรา สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ,พิมพ์ครั้งที่ 1

2.สุเจตน์ จันทรังษ์ . ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051.กรุงเทพฯ: โครงการตำราวิชาการวิทยาลัยมหานคร, พิมพ์ครั้งที่ 1

3.Nationnal Data Acquisition Databook.National Semiconductor

4.Texus Instrument:http://www.ti.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

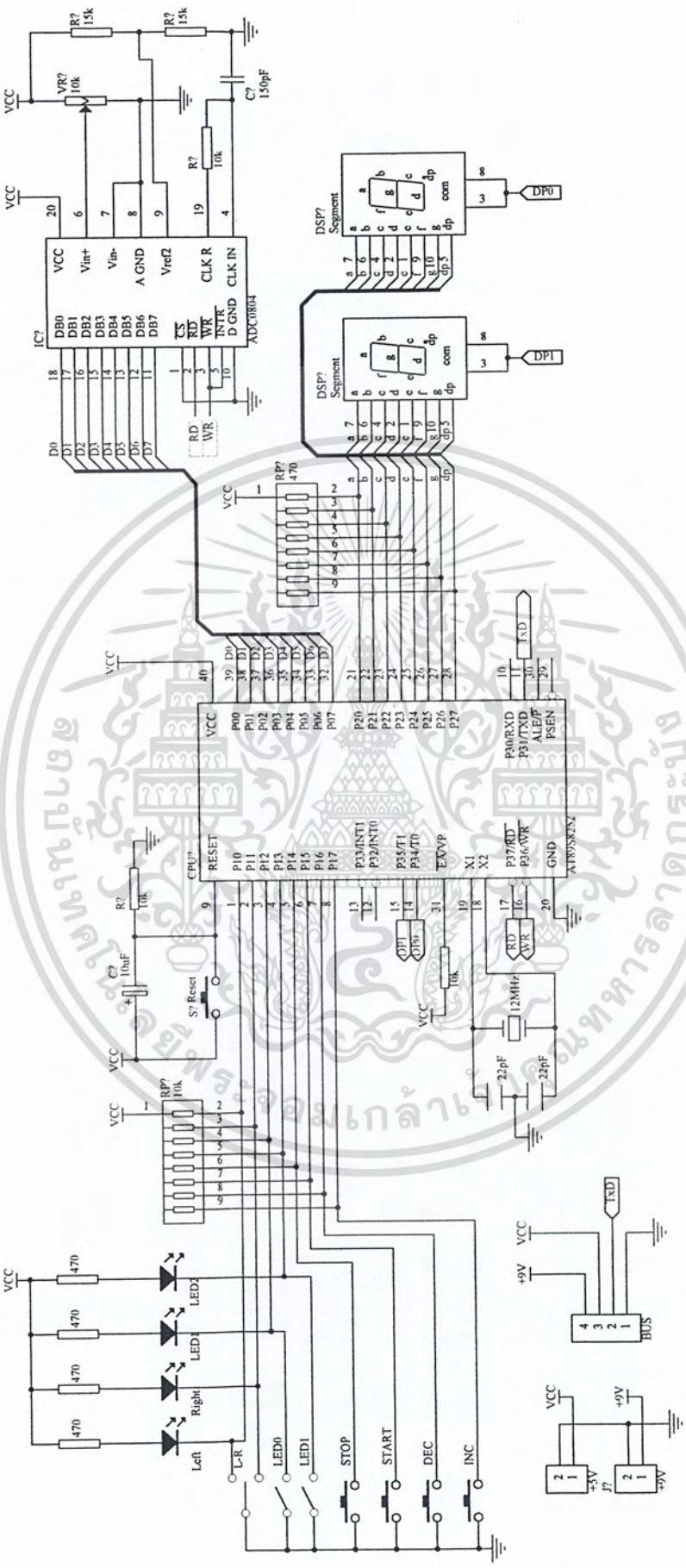
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

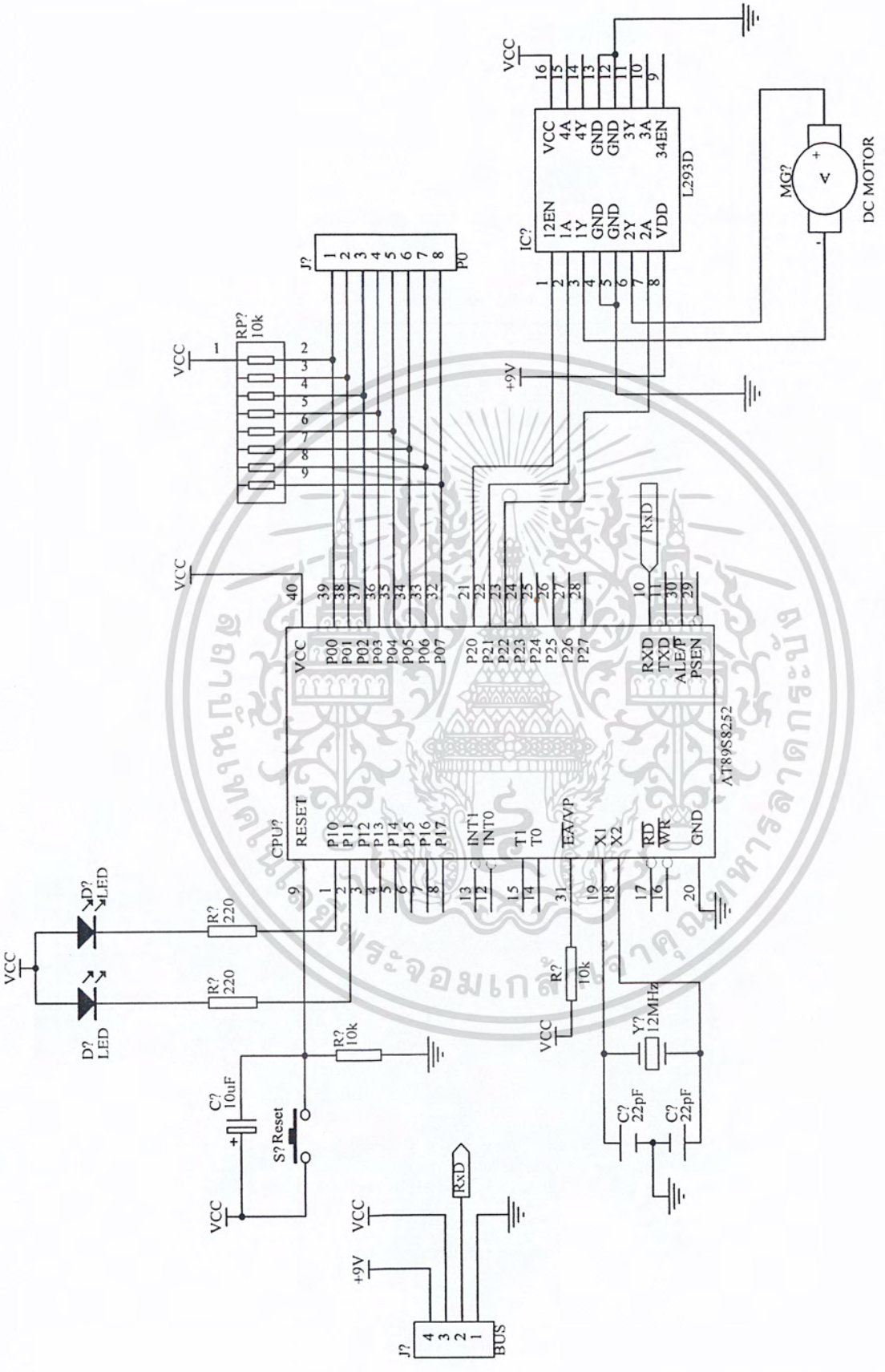
วงจรรวมของโครงงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรชุดควบคุมชุดประมวลผลภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรชุดควบคุมภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
โปรแกรมชุดควบคุมอุปกรณ์ไฟฟ้าภาคส่งและภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมใช้งานในส่วนของภาคส่ง

```
;-----------------------------------
        org     0000h
        ljmp    Main
        org     000Bh
        ljmp    isrTimer0
        org     0030h

;=============================================================
;               initial
;=============================================================
codeReceive     equ     05h     ; โค้ดเริ่มต้นการรับข้อมูล
codeEnd         equ     0Dh
        ; Bit Ram
sKey    equ     20h
SendEnable      equ     08h     ; เป็น "1" ส่งข้อมูลได้
dspEnable       equ     09h
Load    equ     22h

; บิตใน Load
lLed0   equ     0       ; LED0  0 ติด, 1 ดับ
lLed1   equ     1       ; LED1  0 ติด, 1 ดับ
lMotorDirect    equ     2       ; ทิศทางการหมุนของมอเตอร์  0 ซ้าย, 1 ขวา
lMotorEnable    equ     3       ; มอเตอร์ใช้งานได้


        ; Byte Ram
DSP     equ     30h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
T0Interval      equ     31h
ADCResult       equ     32h
SlaveAddr       equ     33h
wswDec          equ     34h
wswInc equ      35h
SpeedMotor      equ     36h
wBlink equ      37h

adcEnable       equ     P3.7
; ---------------------------------------------
iniHardware:    ; RAM ----------------
        mov     SlaveAddr,#1
        mov     DSP,SlaveAddr
        clr     SendEnable
        setb    dspEnable
        mov     wswDec,#00h
        mov     wswInc,#00h
        mov     wBlink,#00h
        mov     Load,#03h

        ; Interrupt ------------
        mov     IE,#10000010b  ; Enable Interrupt Timer0

        ; Initial Timer0 ---------
        mov     TMOD,#21h     ; timer0 ทำงานในโหมด 1 (16 bit), timer1 โหมด 2 (8bitAuto)
        mov     TH0,#0FFh
        mov     TL0,#0FFh     ; ทำให้เกิด overflow ครั้งแรก
        mov     T0Interval,#00h
        setb    TR0     ; Enable Timer0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ; RS-232 & Timer1 ----------
        mov     A,#0E6h         ; Baudrate 1200
        mov     TH1,A
        mov     TL1,A
        mov     SCON,#01000000b   ; 8 bit autoreload
        mov     A,PCON
        clr     ACC.7
        mov     PCON,A          ; กำหนดการหารของ Timer1 (SMOD 0)
        setb    TR1     ; start timer1

        ; ADC ----------------
        clr     adcEnable       ; enable ADC
        clr     P3.6
        setb    P3.6
        ret

;=============================================================
;               Main
;=============================================================
Main:   mov     SP,#3Fh
        lcall   Delay_1S
        lcall   iniHardware

mai1:   lcall   ScanKey
        lcall   ScanLoad
        jnb     SendEnable,mai1
        lcall   SendData
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ljmp    mai1


;=============================================================
;               Master
;=============================================================
SendData:       push    ACC
        mov     A,#codeReceive
        lcall   SendByte
        mov     A,SlaveAddr
        lcall   SendByte
        mov     A,Load
        lcall   SendByte
        mov     A,SpeedMotor
        lcall   SendByte
        mov     A,#codeEnd
        lcall   SendByte
        pop     ACC
        ret


; ส่งข้อมูล ----------------------------------------
; i/p     A
;-------------------------------------------------
SendByte:       mov     SBUF,A
        jnb     TI,$
        clr     TI
        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;=============================================================
;               Display
DataSegment   equ     P2
Digit0  equ     P3.4
Digit1  equ     P3.5
;=============================================================
ScanDisplay:    push    B
        push    ACC
        push    DPH
        push    DPL

        jnb     dspEnable,sdp3

        mov     A,DSP
        mov     B,#10
        div     AB      ; หลักสิบอยู่ใน A, หลักหน่วยอยู่ใน B

        mov     DPTR,#dspTable
        cjne    A,#00h,sdp1
        ljmp    sdp2
sdp1:   movc    A,@A+DPTR
        mov     DataSegment,A
        clr     Digit1
        lcall   Delay_1mS
sdp2:   setb    Digit1

        mov     A,B
        movc    A,@A+DPTR
        mov     DataSegment,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        clr     Digit0
        lcall   Delay_1mS
        setb    Digit0

sdp3:   mov     DataSegment,#00h

        pop     DPL
        pop     DPH
        pop     ACC
        pop     B
        ret

dspTable:       db      3Fh, 06h, 5Bh, 4Fh, 66h, 6Dh, 7Dh, 07h ; 0-7
        db      7Fh, 6Fh, 77h, 7Ch, 39h, 5Eh, 79h, 71h  ; 8-F
        db      40h, 00h                        ; -, ' '

;=============================================================
;       Scan Key Even Driven
nKey    equ     P1
swStop  equ     4
swStart equ     5
swDec   equ     6
swInc   equ     7
;=============================================================
ScanKey:        ; เช็คการกดลง ( Even on key down )
        jb      nKey.swStop,sky1
        jnb     sKey.swStop,sky1
        lcall   swStop_Down
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sky1:   jb      nKey.swStart,sky2
        jnb     sKey.swStart,sky2
        lcall   swStart_Down
sky2:   jb      nKey.swDec,sky3
        jnb     sKey.swDec,sky3
        lcall   swDec_Down
sky3:   jb      nKey.swInc,sky4
        jnb     sKey.swInc,sky4
        lcall   swInc_Down

sky4:   ; เช็คการปล่อยปุ่ม ( Even on key up )
        jnb     nKey.swDec,sky5
        jb      sKey.swDec,sky5
        lcall   swDec_Up
sky5:   jnb     nKey.swInc,sky6
        jb      sKey.swInc,sky6
        lcall   swInc_Up
sky6:
        mov     sKey,nKey
        ret

;------------------------------ make even fPress
EvenKeyPress: push    ACC
        mov     A,#00h
        cjne    A,wswDec,ekp1
        ljmp    ekp2
ekp1:   dec     wswDec
        mov     A,#01h
        cjne    A,wswDec,ekp2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   swDec_fPress
ekp2:   mov     A,#00h
        cjne    A,wswInc,ekp3
        ljmp    ekp4
ekp3:   dec     wswInc
        mov     A,#01h
        cjne    A,wswInc,ekp4
        lcall   swInc_fPress
ekp4:   mov     A,#00h
        cjne    A,wBlink,ekp5
        ljmp    ekp6
ekp5:   dec     wBlink
        mov     A,#01h
        cjne    A,wBlink,ekp6
        lcall   dsp_fBlink
ekp6:   pop     ACC
        ret


;------------------------------ Even Driven
swStop_Down: clr    SendEnable
        mov     wBlink,#00h
        setb    dspEnable
        ret

swStart_Down: setb  SendEnable
        mov     wBlink,#5
        mov     DSP,SlaveAddr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ret

swDec_Down:  push    ACC
        mov     wswDec,#100
        mov     A,#1
        cjne    A,SlaveAddr,sdd1
        ljmp    sdd2
sdd1:   dec     SlaveAddr
        mov     DSP,SlaveAddr
sdd2:   pop     ACC
        ret

swInc_Down:  push    ACC
        mov     wswInc,#100
        mov     A,#99
        cjne    A,SlaveAddr,sid1
        ljmp    sid2
sid1:   inc     SlaveAddr
        mov     DSP,SlaveAddr
sid2:   pop     ACC
        ret

swDec_Up:    mov     wswDec,#00h
        ret

swInc_Up:    mov     wswInc,#00h
        ret

swDec_fPress: lcall    swDec_Down
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     wswDec,#10
        ret

swInc_fPress:   lcall   swInc_Down
        mov     wswInc,#10
        ret

dsp_fBlink:     cpl     dspEnable
        mov     wBlink,#30
        ret

;=============================================================
;               Scan Load
;=============================================================
ScanLoad:       push    ACC
        push    PSW
        mov     C,P1.0
        mov     Load.lMotorDirect,C
        mov     C,P1.2
        mov     Load.lLed0,C
mov     C,P1.3
        mov     Load.lLed1,C
        mov     A,P0
        mov     SpeedMotor,A
        cjne    A,#00h,scl1
        clr     Load.lMotorEnable
        ljmp    scl2
scl1:   setb    Load.lMotorEnable
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
scl2:   pop     PSW
        pop     ACC
        ret

;=============================================================
;       Interrupt Service Rutine
;=============================================================
; ------------------------------------ isr Timer0
isrTimer0:      mov     TH0,#0D8h       ; -10000
        mov     TL0,#0F0h       ; Timer0 overflow ทุก 50mS
        ; Interval ทุกๆ 10mS ----------------

        lcall   ScanDisplay
        lcall   EvenKeyPress

        ;-------------------------------
        inc     T0Interval
        push    ACC
        mov     A,#50
        cjne    A,T0Interval,it01
        mov     T0Interval,#00h
        ; Interval ทุกๆ 500mS ----------------

        ;-------------------------------
it01:   pop     ACC
        reti
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;=============================================================
;               Delay Routine
;=============================================================
Delay_1mS:      push    00h
                push    01h
                mov     R0,#3
d001m2:         mov     R1,#0FFh
d001m1:         djnz    R1,d001m1
                djnz    R0,d001m2
                pop     01h
                pop     00h
                ret

;-------------------------------------
Delay_100mS:    push    00h
                push    01h
                mov     R0,#20h
d1m1:   mov     R1,#0FFh
d1m2:   djnz    R1,d1m2
                djnz    R0,d1m1
                pop     01h
                pop     00h
                ret


;-------------------------------------
Delay_500mS:    push    00h
                mov     R0,#5
d5m1:   lcall   Delay_100mS
                djnz    R0,d5m1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pop     00h
        ret

;-----------------------------------------
Delay_1S:       push    00h
        mov     R0,#10
d1s1:   lcall   Delay_100mS
        djnz    R0,d1s1
        pop     00h
        ret

;-----------------------------------------
Delay_2S:       push    00h
        mov     R0,#20
d2s1:   lcall   Delay_100mS
        djnz    R0,d2s1
        pop     00h
        ret
;-----------------------------------------
Delay_3S:       push    00h
        mov     R0,#30
d3s1:   lcall   Delay_100mS
        djnz    R0,d3s1
        pop     00h
        ret

        end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการใช้งานในส่วนของภาครับ

```
;-------------------------------------
        org     0000h
        ljmp    Main
        org     000Bh
        ljmp    isrTimer0
        org     0030h

;=====================================================
;           initial
;=====================================================
        ; Constant
codeReceive     equ     05h     ; โค้ดเริ่มต้นการรับข้อมูล
codeEnd         equ     0Dh
MyAddr          equ     2       ; ตำแหน่งของ slave แต่ละตัว (1-99)
; บิตใน Load
lLed0   equ     0       ; LED0  0 ติด, 1 ดับ
lLed1   equ     1       ; LED1  0 ติด, 1 ดับ
lMotorDirect    equ     2       ; ทิศทางการหมุนของมอเตอร์  0 ซ้าย, 1 ขวา
lMotorEnable    equ     3       ; มอเตอร์ใช้งานได้

        ; Bit Ram
Load    equ     20h     ; 00h - 07h
PWM     equ     08h

        ; Byte Ram
T0Interval      equ     31h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SpeedMotor     equ      32h      ; 0 หยุด, 1-255 ค่าความเร็ว

; ---------------------------------------
iniHardware:   ; RAM ----------------
        mov     SpeedMotor,#00h
        mov     Load,#00000011b
        clr     PWM

        ; Interrupt ------------
        mov     IE,#10000010b  ; Enable Interrupt Timer0

        ; Initial Timer0 ---------
        mov     TMOD,#21h     ; timer0 ทำงานในโหมด 1 (16 bit), timer1 โหมด 2 (8bitAuto)
        mov     TH0,#0FFh
        mov     TL0,#0FFh     ; ทำให้เกิด overflow ครั้งแรก
        mov     T0Interval,#00h
        setb    TR0     ; Enable Timer0

        ; RS-232 & Timer1 --------
        mov     A,#0E6h         ; Baudrate 1200
        mov     TH1,A
        mov     TL1,A
        mov     SCON,#01010000b    ; 8 bit autoreload, receive enable
        mov     A,PCON
        clr     ACC.7
        mov     PCON,A          ; กำหนดการหารของ Timer1 (SMOD 0)
        setb    TR1     ; start timer1

        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;=============================================================
;               Main
;=============================================================
Main:   mov     SP,#3Fh
        lcall   Delay_100mS
        lcall   iniHardware

        ; รับข้อมูลจาก Master
mai1:   jnb     RI,mai1         ; ถ้าไม่ใช่การรับข้อมูลก็ไม่ต้องทำอะไร
        mov     A,SBUF
        clr     RI
        cjne    A,#codeReceive,mai1

        jnb     RI,$
        mov     A,SBUF          ; PC ส่งข้อมูลมาแล้ว, เอาข้อมูลมา
        clr     RI
        cjne    A,#MyAddr,mai2

        jnb     RI,$
        mov     A,SBUF
        clr     RI
        mov     Load,A ; เก็บเข้าสถานะโหลด

        jnb     RI,$
        mov     A,SBUF
        clr     RI
        mov     SpeedMotor,A  ; เก็บเข้าความเร็วมอเตอร์
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mai2:   jnb     RI,$    ; รอจนกว่าจะมีโค้ดจบการส่ง
        mov     A,SBUF
        clr     RI
        cjne    A,#codeEnd,mai2
        ljmp    mai1


;=============================================================
;  Interrupt Service Rutine PWM Drive Motor and Load
;=============================================================
; ---------------------------------------- isr Timer0
isrTimer0:      push    PSW
        push    ACC
        push    B
        push    00h

        mov     R0,SpeedMotor

        cjne    R0,#00h,it00
        ; ถ้า SpeedMotor เป็น 0 จะหยุดหมุน
        clr     P2.1
        clr     P2.2
        mov     TH0,#0D8h
        mov     TL0,#0F0h
        ljmp    it04

it00:   jb      PWM,it01
        ; ถ้าเป็น "0" สร้างพัลซ์ "1"
        setb    PWM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     A,R0
        cpl     A
        mov     TH0,A
        mov     TL0,#0Fh
        ljmp    it02

it01:   ; ถ้าเป็น "1" สร้างพัลซ์ "0"
        clr     PWM
        mov     A,R0
        mov     TH0,A
        mov     TL0,#0Fh

it02:   jb      Load.lMotorDirect,it03
        ; ถ้า lMotorDirect เป็น 0 จะหมุนซ้าย
        clr     P2.1
        mov     C,PWM
        mov     P2.2,C
        ljmp    it04

it03:   ; ถ้าเป็น 1 จะหมุนขวา
        clr     P2.2
        mov     C,PWM
        mov     P1.1,C

it04:   ; Load Operation
        mov     C,Load.lLed0
        mov     P1.0,C
        mov     C,Load.lLed1
        mov     P1.1,C
```

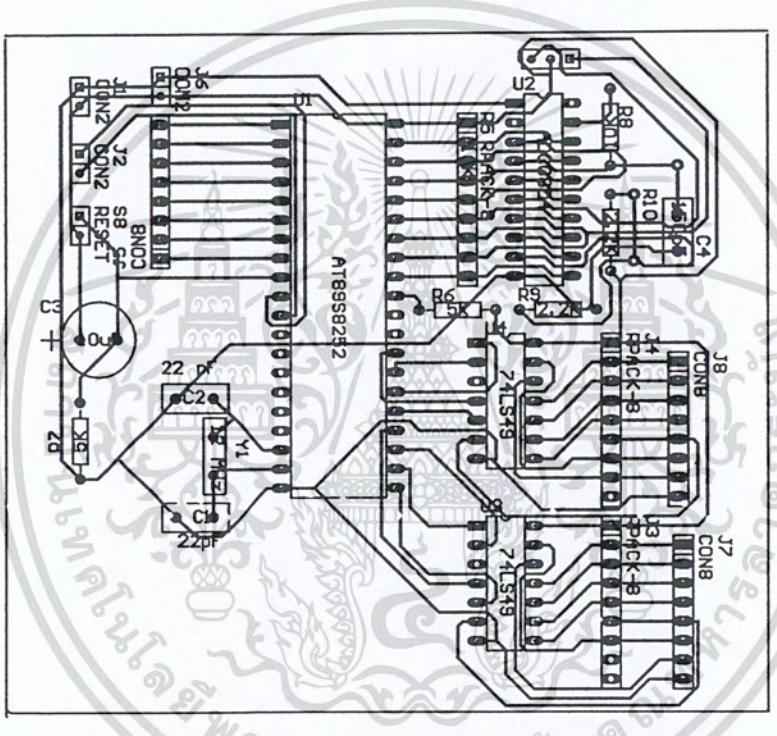
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     C,Load.lMotorEnable
        mov     P2.0,C
        ; Research ส่งข้อมูลไว้ใช้แสดงผล
        mov     A,SpeedMotor
        cpl     A
        mov     P0,A

        pop     00h
        pop     B
        pop     ACC
        pop     PSW
        reti

;=============================================================
;               Delay Routine
;=============================================================
Delay_100mS:  push    00h
        push    01h
        mov     R0,#20h
d1m1:   mov     R1,#0FFh
d1m2:   djnz    R1,d1m2
        djnz    R0,d1m1
        pop     01h
        pop     00h
        ret

        end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

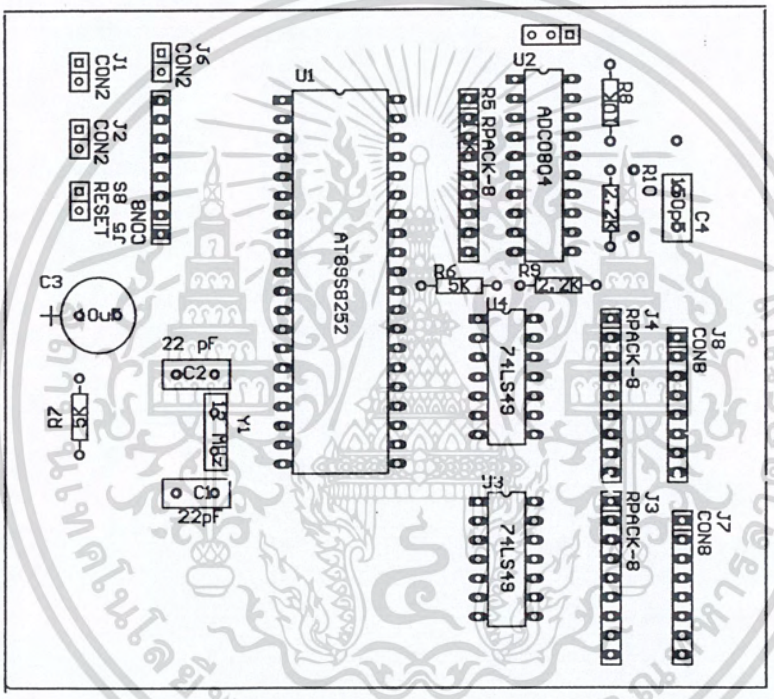
ภาคผนวก ค.
ลายวงจรและการวางอุปกรณ์ของชุดควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



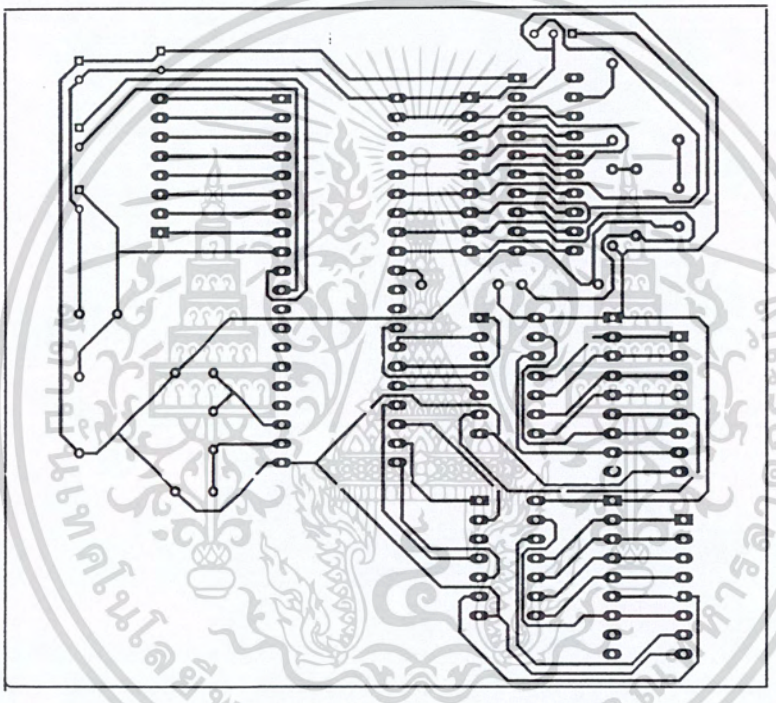
ภาควางอุปกรณ์และลายทองแดงของวงจรชุดควบคุมภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



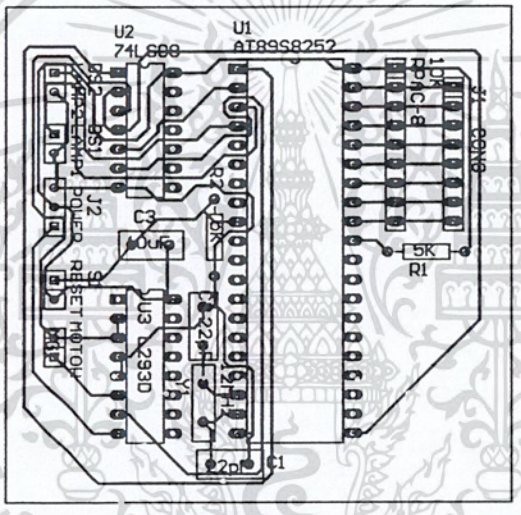
การวางอุปกรณ์ของวงจรชุดควบคุมภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



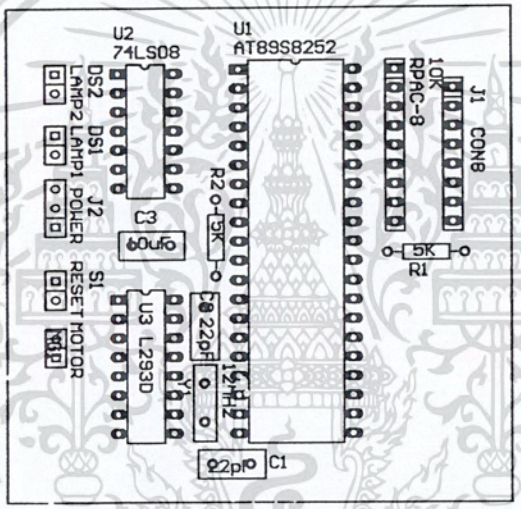
ลายทองแดงของวงจรชุดควบคุมภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



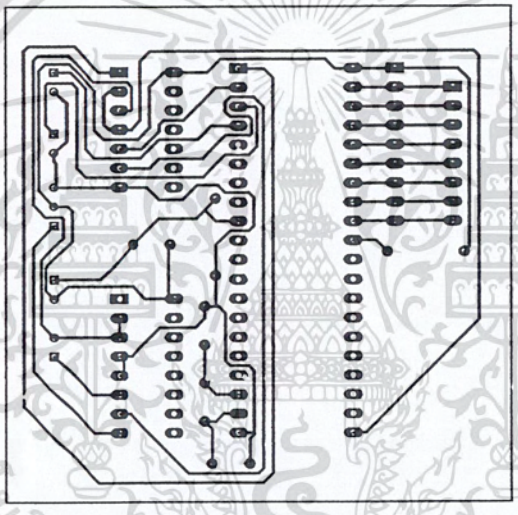
การวางอุปกรณ์และลายทองแดงของวงจรชุดควบคุมภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การวางอุปกรณ์ของวงจรชุดควบคุมภาคส่รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลายทองแดงของวงจรชุดควบคุมภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HARRIS Semiconductor

ADC0802, ADC0803 ADC0804

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

August 1997

Features

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

Description

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

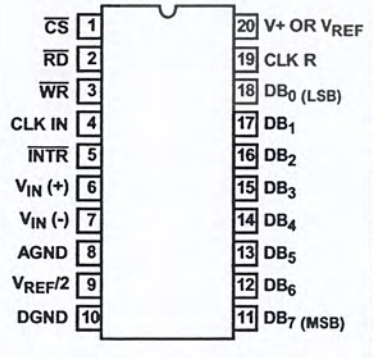
The differential analog voltage input has good common-mode-rejection and permits offsetting the analog zero-input-voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Ordering Information

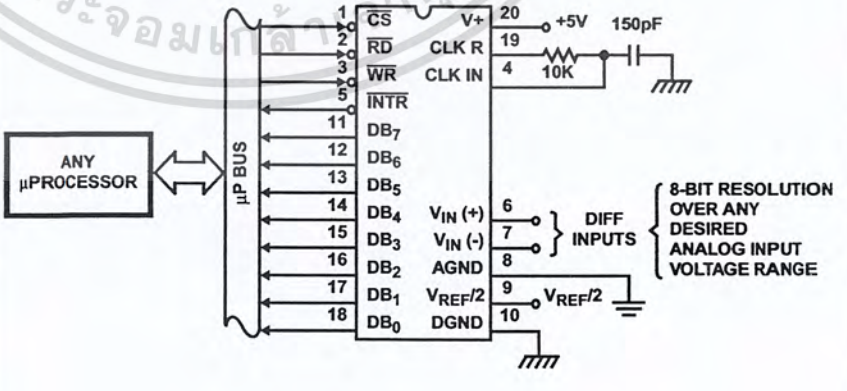
PART NUMBER	ERROR	EXTERNAL CONDITIONS	TEMP. RANGE (°C)	PACKAGE	PKG. NO
ADC0802LCN	$\pm^1/_2$ LSB	$V_{REF}/2 = 2.500V_{DC}$ (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0802LCD	$\pm^3/_4$ LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0802LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0803LCN	$\pm^1/_2$ LSB	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	0 to 70	20 Ld PDIP	E20.3
ADC0803LCD	$\pm^3/_4$ LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0803LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3
ADC0803LD	±1 LSB		-55 to 125	20 Ld CERDIP	F20.3
ADC0804LCN	±1 LSB	$V_{REF}/2 = 2.500V_{DC}$ (No Adjustments)	0 to 70	20 Ld PDIP	E20.3
ADC0804LCD	±1 LSB		-40 to 85	20 Ld CERDIP	F20.3
ADC0804LCWM	±1 LSB		-40 to 85	20 Ld SOIC	M20.3

Pinout

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



Typical Application Schematic



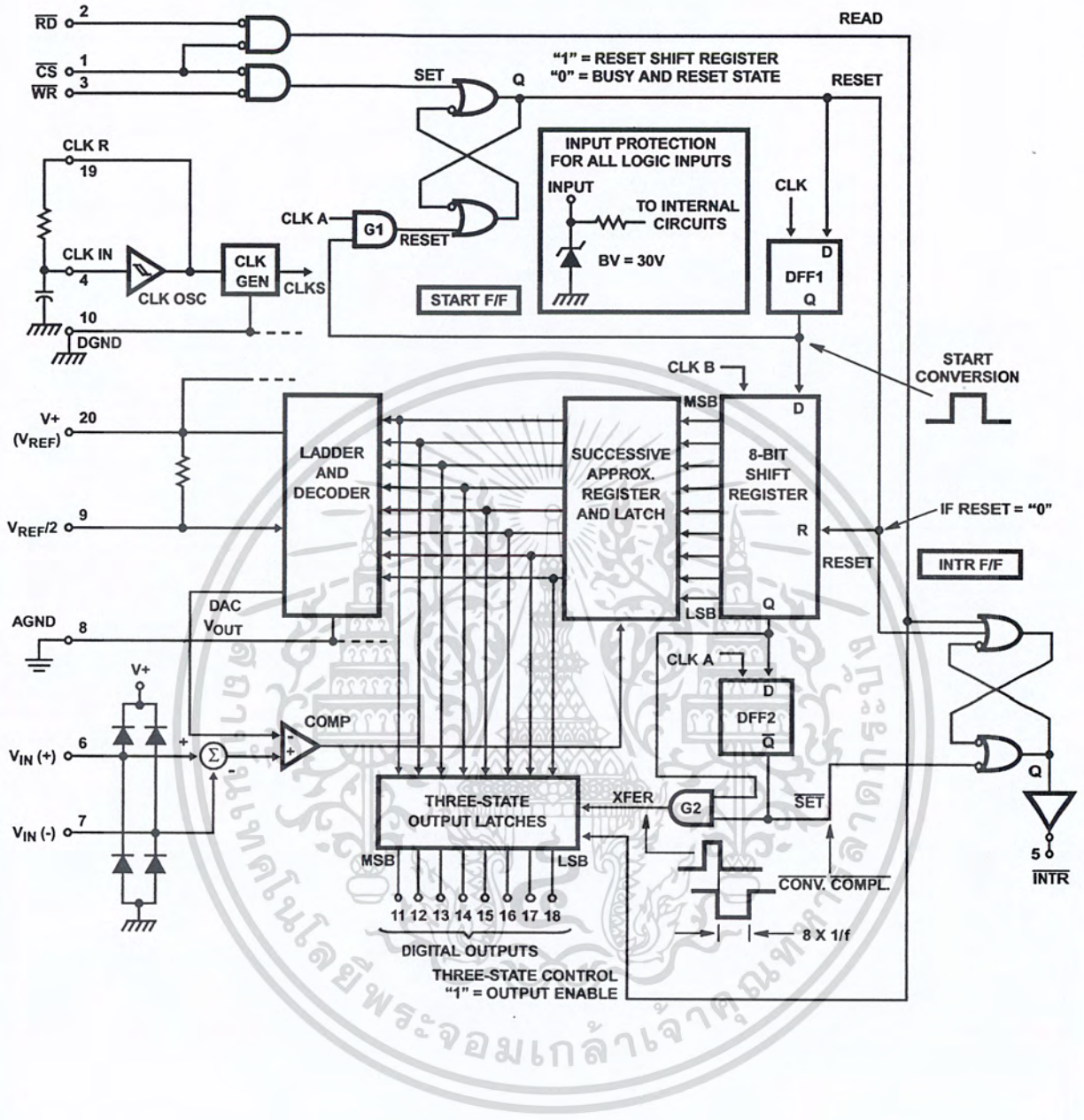
CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper IC Handling Procedures.

Copyright © Harris Corporation 1997

File Number **3094.1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

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

Operating Conditions

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

Thermal Information

Thermal Resistance (Typical, Note 1)	θ_{JA} (°C/W)	θ_{JC} (°C/W)
PDIP Package	125	N/A
CERDIP Package	80	20
SOIC Package	120	N/A

Maximum Junction Temperature
Hermetic Package . . . . . 175°C
Plastic Package . . . . . 150°C
Maximum Storage Temperature Range . . . . . -65°C to 150°C
Maximum Lead Temperature (Soldering, 10s) . . . . . 300°C
(SOIC - Lead Tips Only)

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

NOTE:

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

Electrical Specifications (Notes 1, 7)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS	V+ = 5V, T_A = 25°C and f_{CLK} = 640kHz, Unless Otherwise Specified				
Total Unadjusted Error					
ADC0802	$V_{REF}/2$ = 2.500V	-	-	$\pm^1/_2$	LSB
ADC0803	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	-	-	$\pm^1/_2$	LSB
ADC0804	$V_{REF}/2$ = 2.500V	-	-	±1	LSB
$V_{REF}/2$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm^1/_{16}$	$\pm^1/_8$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm^1/_{16}$	$\pm^1/_8$	LSB
CONVERTER SPECIFICATIONS	V+ = 5V, 0°C to 70°C and f_{CLK} = 640kHz, Unless Otherwise Specified				
Total Unadjusted Error					
ADC0802	$V_{REF}/2$ = 2.500V	-	-	$\pm^1/_2$	LSB
ADC0803	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	-	-	$\pm^1/_2$	LSB
ADC0804	$V_{REF}/2$ = 2.500V	-	-	±1	LSB
$V_{REF}/2$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm^1/_8$	$\pm^1/_4$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm^1/_{16}$	$\pm^1/_8$	LSB
CONVERTER SPECIFICATIONS	V+ = 5V, -25°C to 85°C and f_{CLK} = 640kHz, Unless Otherwise Specified				
Total Unadjusted Error					
ADC0802	$V_{REF}/2$ = 2.500V	-	-	$\pm^3/_4$	LSB
ADC0803	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	-	-	$\pm^3/_4$	LSB
ADC0804	$V_{REF}/2$ = 2.500V	-	-	±1	LSB
$V_{REF}/2$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm^1/_8$	$\pm^1/_4$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm^1/_{16}$	$\pm^1/_8$	LSB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Specifications (Notes 1, 7) **(Continued)**

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
CONVERTER SPECIFICATIONS V+ = 5V, -55°C to 125°C and f_{CLK} = 640kHz, Unless Otherwise Specified					
Total Unadjusted Error ADC0802	$V_{REF}/2$ = 2.500V	-	-	±1	LSB
ADC0803	$V_{REF}/2$ Adjusted for Correct Full Scale Reading	-	-	±1	LSB
$V_{REF}/2$ Input Resistance	Input Resistance at Pin 9	1.0	1.3	-	kΩ
Analog Input Voltage Range	(Note 2)	GND-0.05	-	(V+) + 0.05	V
DC Common-Mode Rejection	Over Analog Input Voltage Range	-	$\pm^1/_8$	$\pm^1/_4$	LSB
Power Supply Sensitivity	V+ = 5V ±10% Over Allowed Input Voltage Range	-	$\pm^1/_8$	$\pm^1/_4$	LSB
AC TIMING SPECIFICATIONS V+ = 5V, and T_A = 25°C, Unless Otherwise Specified					
Clock Frequency, f_{CLK}	V+ = 6V (Note 3)	100	640	1280	kHz
	V+ = 5V	100	640	800	kHz
Clock Periods per Conversion (Note 4), t_{CONV}		62	-	73	Clocks/Conv
Conversion Rate In Free-Running Mode, CR	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}$ = 0V, f_{CLK} = 640kHz	-	-	8888	Conv/s
Width of $\overline{WR}$ Input (Start Pulse Width), $t_{W(WR)I}$	$\overline{CS}$ = 0V (Note 5)	100	-	-	ns
Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid), t_{ACC}	C_L = 100pF (Use Bus Driver IC for Larger C_L)	-	135	200	ns
Three-State Control (Delay from Rising Edge of $\overline{RD}$ to HI-Z State), t_{1H}, t_{0H}	C_L = 10pF, R_L = 10K (See Three-State Test Circuits)	-	125	250	ns
Delay from Falling Edge of $\overline{WR}$ to Reset of $\overline{INTR}$, t_{WI}, t_{RI}		-	300	450	ns
Input Capacitance of Logic Control Inputs, C_{IN}		-	5	-	pF
Three-State Output Capacitance (Data Buffers), C_{OUT}		-	5	-	pF
DC DIGITAL LEVELS AND DC SPECIFICATIONS V+ = 5V, and T_{MIN} to T_{MAX}, Unless Otherwise Specified					
CONTROL INPUTS (Note 6)					
Logic "1" Input Voltage (Except Pin 4 CLK IN), V_{INH}	V+ = 5.25V	2.0	-	V+	V
Logic "0" Input Voltage (Except Pin 4 CLK IN), V_{INL}	V+ = 4.75V	-	-	0.8	V
CLK IN (Pin 4) Positive Going Threshold Voltage, $V+_{CLK}$		2.7	3.1	3.5	V
CLK IN (Pin 4) Negative Going Threshold Voltage, $V-_{CLK}$		1.5	1.8	2.1	V
CLK IN (Pin 4) Hysteresis, V_H		0.6	1.3	2.0	V
Logic "1" Input Current (All Inputs), I_{INHI}	V_{IN} = 5V	-	0.005	1	µA
Logic "0" Input Current (All Inputs), I_{INLO}	V_{IN} = 0V	-1	-0.005	-	µA
Supply Current (Includes Ladder Current), I+	f_{CLK} = 640kHz, T_A = 25°C and $\overline{CS}$ = HI	-	1.3	2.5	mA
DATA OUTPUTS AND $\overline{INTR}$					
Logic "0" Output Voltage, V_{OL}	I_O = 1.6mA, V+ = 4.75V	-	-	0.4	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Specifications (Notes 1, 7) (Continued)

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

NOTES:

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

Timing Waveforms

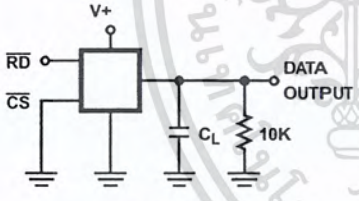


FIGURE 1A. t_{1H}

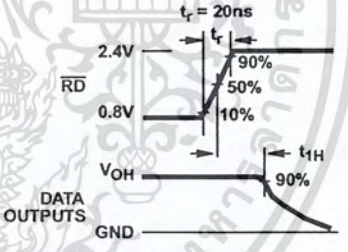


FIGURE 1B. t_{1H}, C_L = 10pF

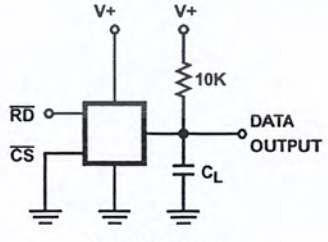


FIGURE 1C. t_{0H}

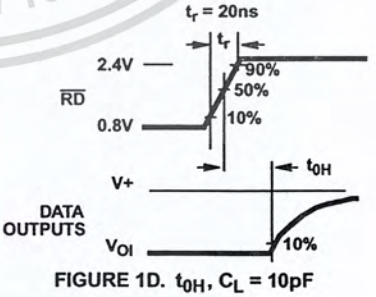


FIGURE 1D. t_{0H}, C_L = 10pF

FIGURE 1. THREE-STATE CIRCUITS AND WAVEFORMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves

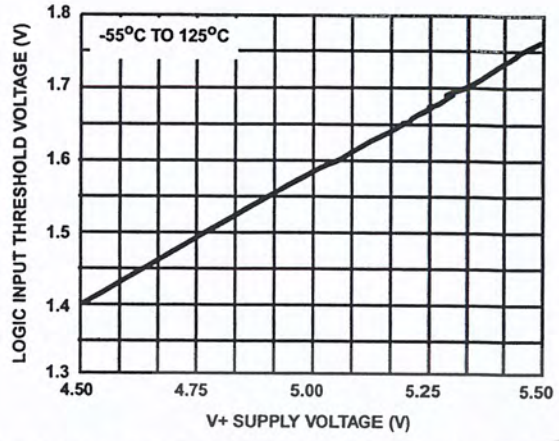


FIGURE 2. LOGIC INPUT THRESHOLD VOLTAGE vs SUPPLY VOLTAGE

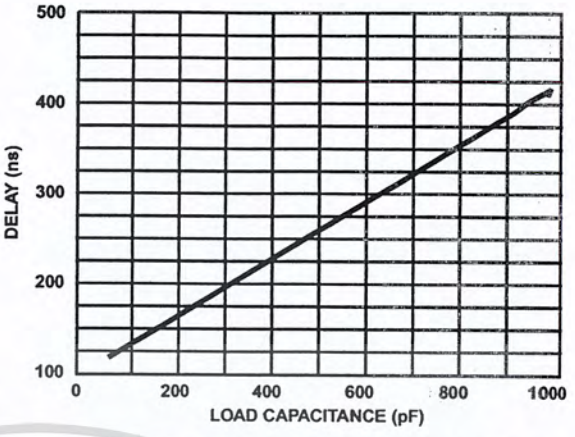


FIGURE 3. DELAY FROM FALLING EDGE OF $\overline{RD}$ TO OUTPUT DATA VALID vs LOAD CAPACITANCE

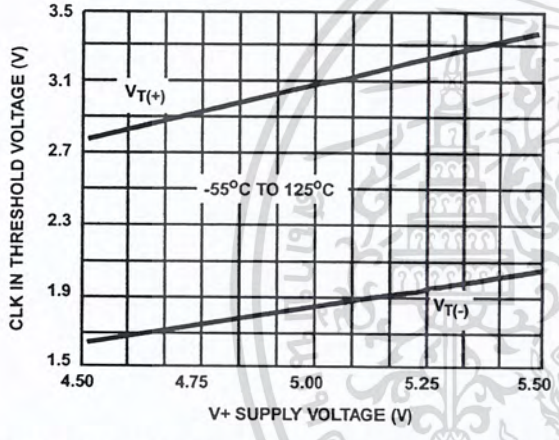


FIGURE 4. CLK IN SCHMITT TRIP LEVELS vs SUPPLY VOLTAGE

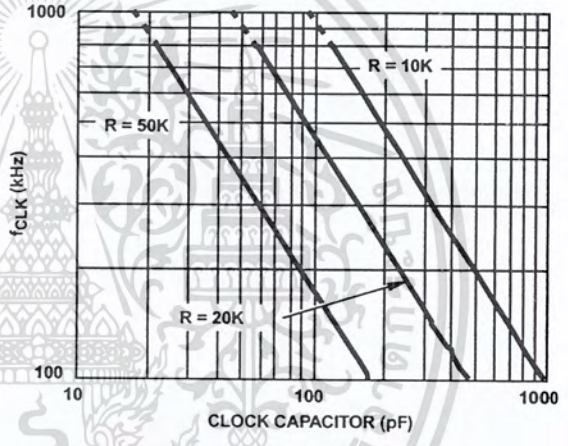


FIGURE 5. f_{CLK} vs CLOCK CAPACITOR

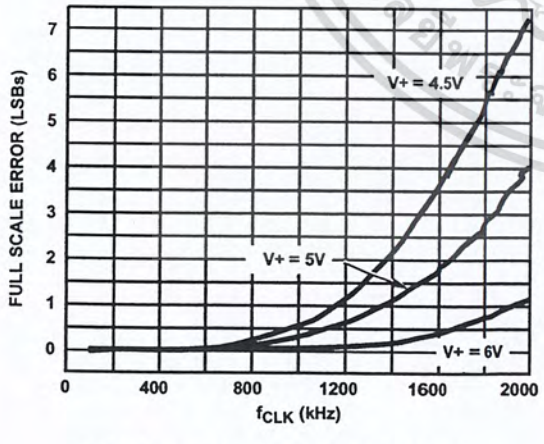


FIGURE 6. FULL SCALE ERROR vs f_{CLK}

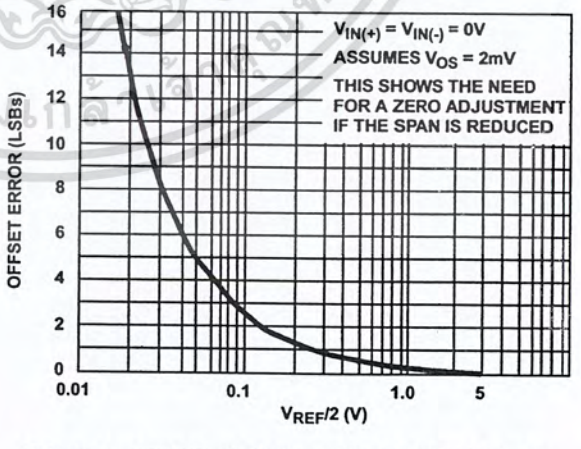


FIGURE 7. EFFECT OF UNADJUSTED OFFSET ERROR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves (Continued)

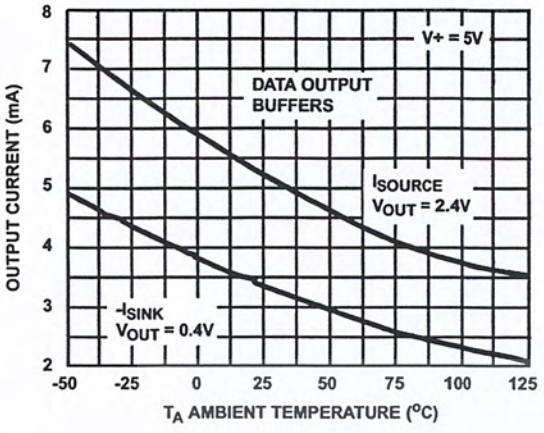


FIGURE 8. OUTPUT CURRENT vs TEMPERATURE

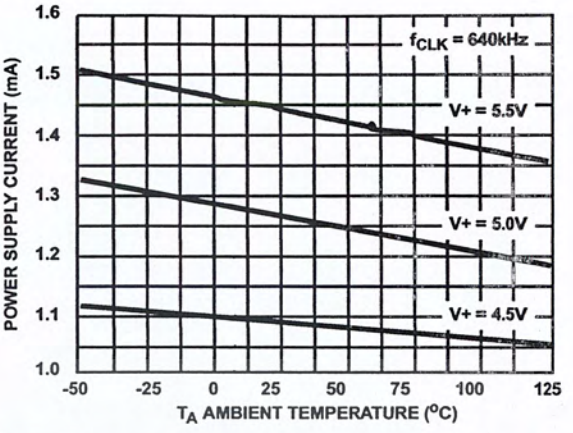


FIGURE 9. POWER SUPPLY CURRENT vs TEMPERATURE

Timing Diagrams

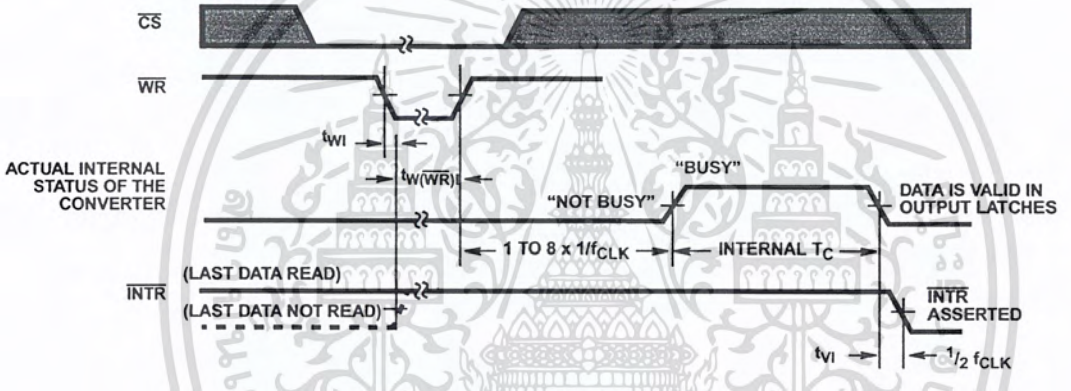


FIGURE 10A. START CONVERSION

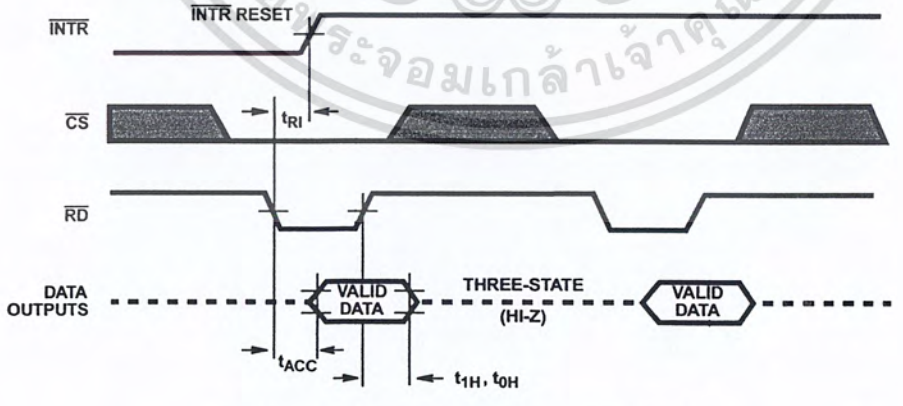


FIGURE 10B. OUTPUT ENABLE AND RESET INTR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้