

เครื่องควบคุมการให้น้ำเกลืออัตโนมัติ
AUTOMATICS SALINE DRIPPING CONTROL



โดย

นางสาวปรีษากรณ์
นายพิมพ์
นายศรายุทธ์

ขวัญชัย
ผลพฤกษา
ขุนณรงค์

อาจารย์ที่ปรึกษา

ผศ. พิชัย คูศิริวานิชกร

เลขหมู่.....
เลขทะเบียน..... 42748
วัน, เดือน, ปี..... 7 ส.ย. 2545

.....
.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๒๕๕๕

ปริญญาานิพนธ์เรื่อง

เครื่องควบคุมการให้น้ำเกลืออัตโนมัติ

AUTOMATICS SALINE DRIPPING CONTROL

จัดทำโดย

นางสาวปริยาภรณ์ ขวัญชัย (41013252)

นายพิมล ผลพฤกษา (41013261)

นายศรายุทธ ขุนณรงค์ (41013267)

อาจารย์ที่ปรึกษา

ผศ. พิชัย กุศิริวานิชกร



ปริญญาานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ

..... อาจารย์ที่ปรึกษา

(พิชัย กุศิริวานิชกร)


วันที่ 30 / 3 / 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


กิตติกรรมประกาศ

ผู้จัดทำรายงานขอขอบพระคุณอาจารย์ที่ปรึกษา ที่ได้ให้คำปรึกษาในการเริ่มต้นทำโครงการ
ชั้นนี้ตลอดจนข้อคิดเห็นในแนวทางในการแก้ไขปัญหาต่างๆ จนสามารถสำเร็จมาได้ด้วยดี และ
อาจารย์ในภาควิชาอิเล็กทรอนิกส์ที่ให้คำแนะนำในการปฏิบัติงานต่างๆรวมถึงการเรียนการสอนใน
ภาคการศึกษาที่ผ่านมา นำมาประยุกต์ใช้ในการสร้างโครงการชั้นนี้ ซึ่งเป็นประโยชน์อย่างมากทำให้
สามารถแก้ไขปัญหาต่างๆสำเร็จลงด้วยดี ซึ่งถ้าขาดบุคคลดังกล่าวมาแล้ว ผู้จัดทำรายงานก็คงไม่
สามารถที่จะทำงานให้สำเร็จลงได้

โดยผู้จัดโครงการได้นำความรู้ที่ได้รับมา นำมาใช้อ้างอิงเป็นเนื้อหาของโครงการโดยสุดท้าย
นี้ผู้จัดทำรายงานใคร่ขอขอบคุณบุคคลดังกล่าวเป็นอย่างสูง ผู้จัดโครงการขอระลึกถึงความ
ขอบคุณยิ่ง


.....
(นางสาวปริยาภรณ์ ขวัญชัย)


.....
(นายพิมล ผลพุกษา)


.....
(นายสรายุทธ์ ขุนณรงค์)

ผู้จัดทำ

เครื่องควบคุมการให้น้ำเกลืออัตโนมัติ

นางสาวปริยาภรณ์ ขวัญชัย

นายพิมล ผลพุกษา

นายศรายุทธ์ ขุนณรงค์

ผศ.พิชัย คูศิริวานิชกร อาจารย์

ที่ปรึกษา ปีการศึกษา 2543

บทคัดย่อ

เครื่องควบคุมการให้น้ำเกลือโดยอัตโนมัติ เป็นอุปกรณ์ที่ใช้ในทางการแพทย์ทางด้านอายุรเวชชนิดหนึ่ง ที่นิยมใช้กันอย่างแพร่หลายในโรงพยาบาลทั่วไป เพราะสามารถควบคุมอัตราการไหลของน้ำเกลือเข้าสู่ตัวของผู้ป่วยได้อย่างคงที่และอย่างง่ายดาย ซึ่งจะทำให้การรักษาและดูแลผู้ป่วยเป็นไปอย่างมีประสิทธิภาพหลักการการทำงานของเครื่องคือไมโครคอนโทรลเลอร์ (MCS 51) เป็นตัวควบคุมการทำงานของมอเตอร์ให้หมุนไปขับเคลื่อนกลไกการบีบสายน้ำเกลือที่ต่อจากขวดน้ำเกลือไปยังตัวผู้ป่วยให้มีอัตราการไหลตามที่เรากำหนดไว้ สำหรับการตรวจเช็คและป้อนกลับเพื่อทำให้มีอัตราการไหลคงที่ที่ค่าหนึ่งๆนั้น มาจากวงจรตรวจจับหยดน้ำเกลือ นำมาเปรียบเทียบกับค่าที่ตั้งไว้และคำนวณภายในโปรแกรมการทำงานจากนั้นจะส่งค่าไปควบคุมการทำงานของมอเตอร์ นอกจากนี้ ยังมีระบบส่งสัญญาณเตือนเมื่อน้ำเกลือหมด หรือครบกำหนดตามจำนวนที่ตั้งไว้ โดยสามารถตั้งโปรแกรมให้หยุดการให้น้ำเกลือ หรือให้ต่อไปในลักษณะต่อเนื่องในอัตราการไหลที่แตกต่างกัน

AUTOMAICS SALINE DRIPPING CONTROL

Miss.Pariyaporn Kwansai

Mr.Phimon Phonphruksa

Mr.Sarayut Kunnarong

Assist.Prof.PichaiKoosirivanishkorn

(A d v i s o r)

2st Semester,Educationnal Year 2000

Abstract

Automatic dripping control machine is one of the widely used equipment in the medication. It can be used to control flow rate of saline dripping accurately for better result of patient treatment. The system uses micro-controller (MCS 51) to control motor in the saline tube compressing mechanism. Flow rate of saline will be controlled to have the defined value. In controlling of the flow rate drop of saline is detected and calculated to the value of flow rate then compared to the defined value and the data is sent to control the motor. Besides that the system will generate alarm signal when the saline is finished or exceed the defined volume. Control program can be set to make the system stop working or continue working in differences of flow rate after alarm.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
Abstract	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์และแนวคิดโครงการ	1
1.2 รายละเอียดของเนื้อหาในรายงาน	2
บทที่ 2 แนวความคิดพื้นฐานในการออกแบบ	3
2.1 กลไกการให้น้ำเกลือที่มีลักษณะแบบการรีด	3
2.2 กลไกการรีดแบบปั๊มต้นตะขาบ (FINGER PUMEP)	3
บทที่ 3 การออกแบบและทดลองชุดตรวจจับหยดน้ำ	5
3.1 การออกแบบวงจรสร้างพัลส์ โดยใช้ Ic Timer 555	6
3.2 การออกแบบวงจร Drive ชุด Infrared sensor	7
3.2.1 ภาควงของ Infrared Photo Transistor	7
3.2.2 ภาควงรับ ใช้ Infrared Photo Transistor	9
3.3 การออกแบบวงจร Band pass Filter	9
3.4 การออกแบบวงจร Non Inverting Amplifier	10
3.5 การออกแบบวงจร Detector	11
3.6 การออกแบบวงจร Mono Stable	12
บทที่ 4 วงจรขั้วดีซีมอเตอร์และวงจรตรวจสอบฟองอากาศ	15
4.1 วงจรขั้วดีซีมอเตอร์	15
4.2 วงจรตรวจจับฟองอากาศ	16
4.2.1 การทำงานของวงจร	16
4.2.2 การออกแบบวงจรส่วน Air Detector	16
4.2.3 การออกแบบวงจร Comparator	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
4.2.4 การออกแบบวงจร Latch	18
บทที่ 5 การเขียนโปรแกรมควบคุมการทำงานของเครื่องให้น้ำเกลือ	21
Flow chart ของ โปรแกรม	22
บทที่ 6 การทดลอง	29
6.1 การทดลองการเปลี่ยนแปลงของอัตราการไหลต่อปริมาณของน้ำเกลือ	29
6.2 การทดลองวัดอัตราการไหลที่ค่าต่างๆ เปรียบเทียบกับค่าที่ตั้งไว้	30
6.3 การทดลองการวัดปริมาตรของน้ำเกลือที่ต้องการให้กับปริมาตรที่ออกมาจริง	33
6.4 การทดลองการทำงานของชุดตรวจจับหยดน้ำเกลือเมื่อความเข้มแสงเปลี่ยนไป	36
สรุปและวิจารณ์ผลการทดลอง	38
บทที่ 7 สรุปและข้อเสนอแนะ	39
7.1 ชุดกลไกการบีบสาย	39
7.2 ชุด Power Supply และแบตเตอรี่สำรอง	39
ภาคผนวก	
เอกสารอ้างอิง	

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงกลไกการให้น้ำเกลือแบบรีด	3
รูปที่ 2.2 แสดงกลไกการให้น้ำเกลือแบบปั๊มตีนตะขาบ	4
รูปที่ 2.3 แสดงบล็อกไดอะแกรมอย่างง่ายของเครื่องให้น้ำเกลือ	4
รูปที่ 3.1 แสดงส่วนประกอบของชุดตรวจจับหยดน้ำ	5
รูปที่ 3.2 แสดงชุดตรวจจับหยดน้ำ	5
รูปที่ 3.3 แสดงบล็อกไดอะแกรมของวงจรตรวจจับหยดน้ำ	6
รูปที่ 3.4 แสดงการสร้างพัลส์โดยใช้ Ic 555	7
รูปที่ 3.5 แสดงวงจร Drive ชุด Infrared sensor	7
รูปที่ 3.6 แสดงวงจรขับ IR Photo Transistor ภาครีบ	8
รูปที่ 3.7 แสดงวงจร Band pass filter	10
รูปที่ 3.8 แสดงวงจรขยายแบบ Non Inverting Amplifier	11
รูปที่ 3.9 แสดงวงจรส่วน Detector และ Comparator	12
รูปที่ 3.10 แสดงวงจร Mono Stable Ic 555	13
รูปที่ 3.11 แสดงวงจรตรวจจับหยดน้ำ	14
รูปที่ 4.1 Block diagram ของวงจรตรวจจับฟองอากาศ	16
รูปที่ 4.2 วงจรส่วนตรวจจับฟองอากาศ	16
รูปที่ 4.3 แสดงวงจร Comparator	17
รูปที่ 4.4 แสดงวงจร Latch โดยใช้ Ic 555	18
รูปที่ 4.5 แสดงวงจรควบคุมมอเตอร์	19
รูปที่ 4.6 แสดงวงจรตรวจจับฟองอากาศ	20
รูปที่ 5.1 แสดงการต่อขาใช้งานของ MCS-51	21
รูปที่ 5.2 แสดง Flow Chart โปรแกรมควบคุมการทำงานของเครื่องให้น้ำเกลือ	22
รูปที่ 5.3 แสดง Key board	28
รูปที่ 6.1 แสดงความสัมพันธ์ปริมาณน้ำเกลือกับอัตราการไหลของน้ำเกลือ	30
รูปที่ 6.2 แสดงความสัมพันธ์ระหว่างอัตราการไหลที่ค่าต่างๆ เปรียบเทียบกับ ค่าที่ตั้งไว้	32
รูปที่ 6.3 แสดง ค่าความผิดพลาดเฉลี่ย(%)ต่ออัตราการไหลที่ตั้งไว้	33

	หน้า
รูปที่ 6.4 แสดงความสัมพันธ์ระหว่างปริมาตรกับอัตราการไหลของน้ำเกลือ ที่ค่าต่างๆ	35
รูปที่ 6.5 แสดงเปอร์เซ็นต์ความผิดพลาดเทียบกับอัตราการไหลของน้ำเกลือ ที่ค่าต่างๆ	35
รูปที่ 6.6 กราฟแสดงการทำงานของ Sensor หยดน้ำเกลือ เมื่อความเข้มของ แสงสว่างเปลี่ยนไป	37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 4.1 ตารางแสดงความสัมพันธ์ของ R_D และ V_{CE} ขณะมี ฟองอากาศและไม่มีฟองอากาศ	17
ตารางที่ 4.2 แสดงสถานะการทำงานของ IC Timer 555	18
ตารางที่ 6.1 แสดงผลการทดลองการเปลี่ยนแปลงของการไหล ของน้ำเกลือเมื่อปริมาณของน้ำเกลือลดลง	29
ตารางที่ 6.2 แสดงผลการทดลองการทดลองวัดอัตราการไหลที่ค่าต่างๆ เปรียบเทียบกับค่าที่ตั้งไว้	31
ตารางที่ 6.3 แสดงผลการทดลองการวัดปริมาตรของน้ำเกลือที่ต้องการให้ กับปริมาตรที่ออกมาจริง	34
ตารางที่ 6.4 แสดงผลการทดลองการทำงานของชุดตรวจจับหยดน้ำเกลือ เมื่อความเข้มข้นเปลี่ยนไป	37

บทที่ 1

บทนำ

การให้สารละลายทางเส้นเลือดแก่ร่างกายในทางการแพทย์ทางด้านอายุรเวทนั้นนับว่ามีความจำเป็นและเป็นวิธีการที่ใช้กันมานานหลายศตวรรษ แต่ปัญหาที่พบบ่อยๆในการให้น้ำเกลือแก่ผู้ป่วยก็คือ อัตราการไหลนั้นพยาบาลจะใช้วิธีการประมาณ ในการนับจำนวนหยดต่อนาทีโดยปรับแถบบังคับจำนวนหยดที่ติดมากับเซตสายน้ำเกลือ และเมื่อครบกำหนดปริมาณที่ต้องการให้น้ำหรือน้ำเกลือหมดก็จะมีเสียงเตือน พยาบาลผู้ดูแลจึงจำเป็นต้องเดินดูทุกระยะๆซึ่งไม่สะดวก นอกจากนี้เครื่องมือที่ใช้ในการควบคุมการให้น้ำเกลือ ซึ่งส่วนใหญ่ต้องนำเข้าจากต่างประเทศมักมีราคาค่อนข้างแพง จึงเป็นจุดเริ่มต้นให้ดำเนินโครงการทำการศึกษาค้นคว้าวิธีการที่จะสร้างเครื่องขึ้นขึ้น ประกอบกับการนำเอาความรู้ในสาขาวิชาอิเล็กทรอนิกส์ไปประยุกต์ใช้ในทางการแพทย์ เพื่อวิจัยและพัฒนาเครื่องมือทางการแพทย์ที่ใช้หลักการง่ายและราคาถูกโดยจะเน้นให้มีการควบคุมการทำงานของเครื่องโดยอัตโนมัติสามารถใช้งานได้ง่าย และมีความปลอดภัยในการใช้งานสูง

1.1 วัตถุประสงค์ของโครงการ

- 1.1.1. เพื่อสร้างเครื่องให้น้ำเกลืออัตโนมัติ ที่ใช้หลักการง่าย ๆ มีประสิทธิภาพ สร้างได้ง่ายและมีราคาถูก
- 1.1.2. เพื่อสร้างเครื่องต้นแบบในการพัฒนาเครื่องมือทางการแพทย์ ขึ้นใช้เองภายในประเทศ ที่ขาดแคลนเครื่องมือ ซึ่งส่วนใหญ่ต้องนำเข้าจากต่างประเทศ
- 1.1.3. เพื่อเป็นการนำเทคโนโลยีและนำความรู้ที่ได้เรียนมาในสาขาวิชาอิเล็กทรอนิกส์ นำมาประยุกต์ใช้งานทางการแพทย์

1.2 รายละเอียดของเนื้อหาในรายงาน

ในปฏิญานิพนธ์ฉบับนี้ได้แบ่งแนวทางในการสร้างเครื่องควบคุมการให้น้ำเกลือออกเป็นหัวข้อต่างจำนวน 7 บท โดย

บทที่ 1 เป็นการกล่าวถึงที่มาและความจำเป็นของเครื่องให้น้ำเกลือซึ่งเป็นแรงจูงใจให้ทำการศึกษาค้นคว้า

บทที่ 2 เป็นการกล่าวถึงแนวความคิดพื้นฐานในการออกแบบ

บทที่ 3 , บทที่ 4 และบทที่ 5 ได้ แสดงถึงการออกแบบวงจร อุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการควบคุมการทำงานของระบบทั้งหมด

บทที่ 6 เป็นการทดลองและผลการทดลองที่แสดงถึงประสิทธิภาพและข้อจำกัดในการใช้งาน
เครื่องให้น้ำเกลือที่ได้สร้างขึ้น

บทที่ 7 เป็นข้อสรุปและแนวทางในการพัฒนาเครื่องนี้ต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

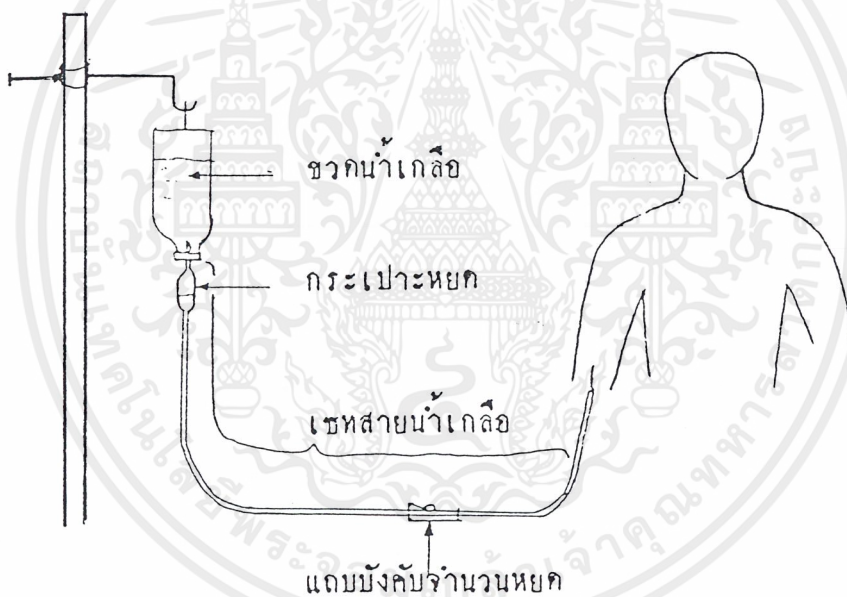
บทที่ 2

แนวความคิดพื้นฐานในการออกแบบ

การให้น้ำเกลือโดยใช้เครื่องควบคุมอัตโนมัติที่มีใช้ตามโรงพยาบาลทั่วไป ในปัจจุบันรูปแบบต่าง ๆ กัน แต่หลักการสำคัญของเครื่องนี้ก็คือการใช้ส่วนวงจรอิเล็กทรอนิกส์ควบคุมการทำงานของกลไกการฉีดสายน้ำเกลือให้เป็นไปตามต้องการ ซึ่งจากการศึกษาและหาข้อมูลตามโรงพยาบาลและวารสารทางวิชาการแพทย์ รวมทั้งเครื่องควบคุมการให้น้ำเกลือหลายยี่ห้อที่ใช้ในโรงพยาบาลสามารถแบ่งลักษณะการทำงานได้เป็น 2 กลุ่มใหญ่ๆคือ

2.1 กลไกการให้น้ำเกลือที่มีลักษณะแบบการฉีด

กลไกการฉีดสายแบบนี้ส่วนใหญ่จะติดมากับชุดชุดสายน้ำเกลือทำให้สะดวกและมีความคล่องตัว แต่โดยมากมักต้องปรับจำนวนหยดน้ำเกลือด้วยมือ



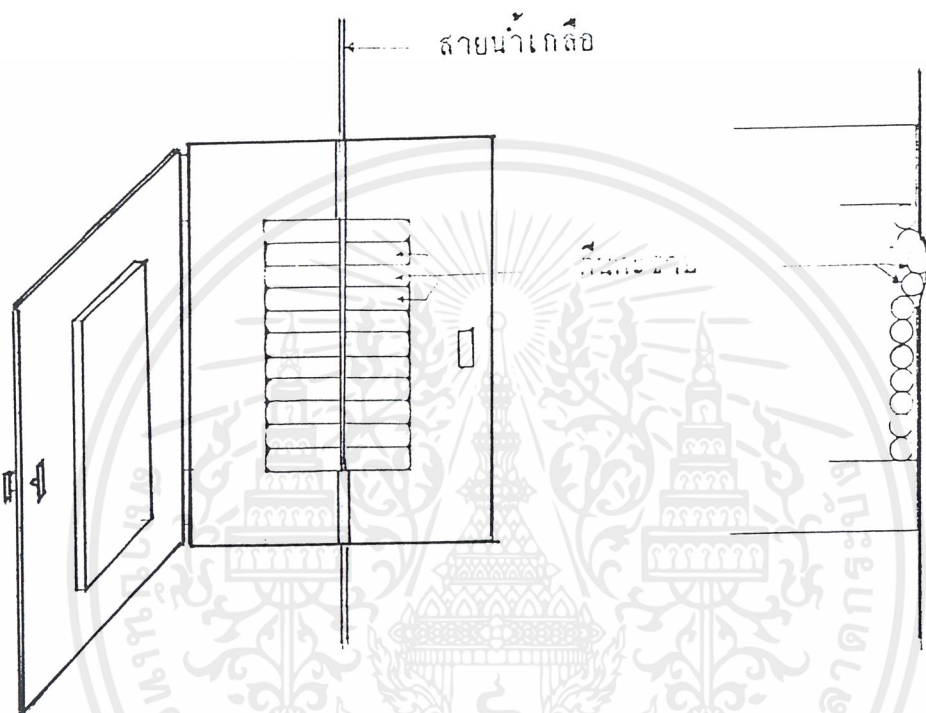
รูปที่ 2.1 แสดงกลไกการให้น้ำเกลือแบบฉีด

2.2 กลไกการฉีดแบบปั๊มตีนตะขาบ (FINGER PUMEP)

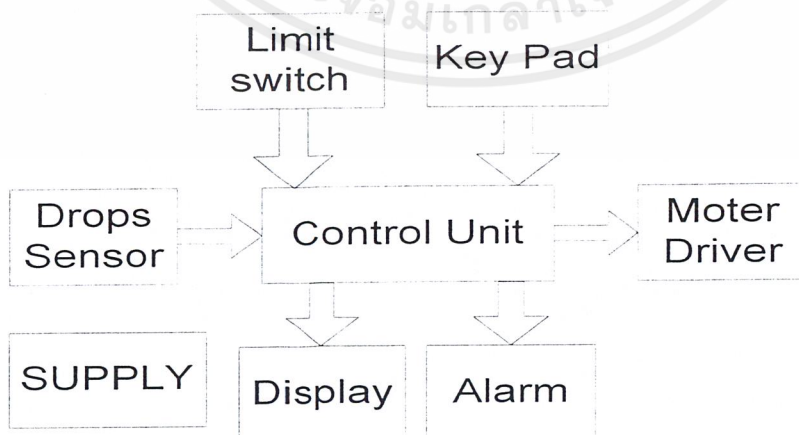
เป็นกลไกที่นิยมใช้กันมากที่สุด สามารถใช้งานได้สะดวกและมีความแม่นยำค่อนข้างสูงดังรูปที่ 2 แสดงการทำงานของกลไกอย่างคร่าวๆคือ ชุดฉีดสายน้ำเกลือแต่ละอันจะวางเรียงเป็นหน้ากระดานและจะเคลื่อนตัวกดสายเรียงกันไปทีละอันคล้ายกับการใช้ล้อกดบีบสายแล้วยกกลับมากดบีบครั้งต่อไปทำให้เกิดการฉีดน้ำเกลือออกไป ชับซ้อนมากพอสมควร ซึ่งไม่อาจใช้เวลาอันไม่มากนักในการทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลไกที่มีลักษณะการรีด ซึ่งจะได้นำสู่การออกแบบในบทต่อไป ส่วนทางด้านวงจรอิเล็กทรอนิกส์ เพื่อใช้ในการควบคุมการทำงานของชุดรีดสายน้ำเกลือ(การควบคุมการหมุนของมอเตอร์และการตรวจเช็คจำนวนหยดน้ำเกลือ) จะใช้ไมโครคอนโทรลเลอร์ (MCS 51) ควบคุมการทำงานของมอเตอร์ซึ่งจะได้นำสู่ในส่วนการออกแบบวงจรควบคุมการทำงานในบทต่อไป



รูปที่ 2.2 แสดงกลไกการให้น้ำเกลือแบบปั๊มตี้นตะขบ



รูปที่ 2.3 แสดงบล็อกไดอะแกรมอย่างง่ายของเครื่องให้น้ำเกลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

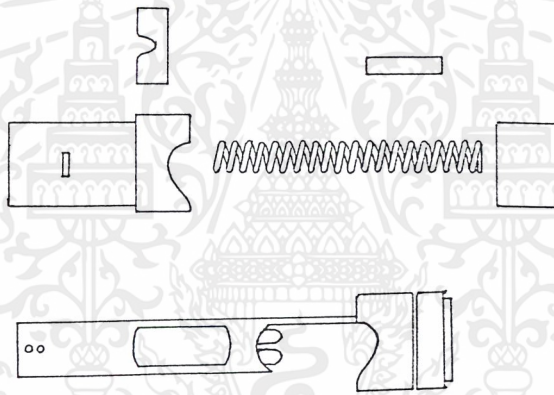
บทที่ 3

การออกแบบและทดลองชุดตรวจจับหยดน้ำ

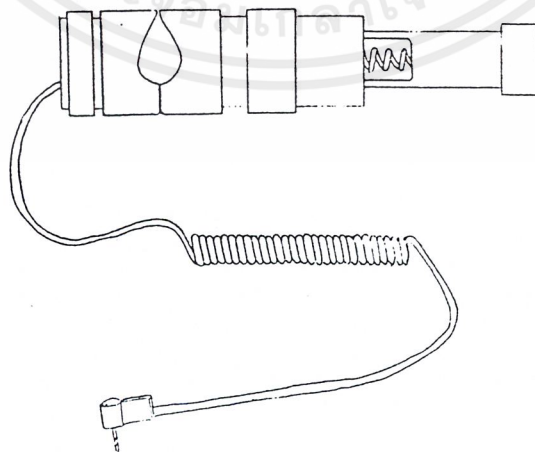
จากข้อมูลผลการทดลองในภาคการศึกษาที่ผ่านมาทำให้ทราบปัญหาและข้อบกพร่องหลายประการจากชุดตรวจจับหยดน้ำ เช่น

1. ไม่สามารถตรวจจับได้ ถ้าหากระยะระหว่างตัวรับและตัวส่งเปลี่ยนไป
2. ชุดอินฟราเรดเซ็นเซอร์มีอายุการใช้งานต่ำเนื่องจากใช้ไฟกระเสตกรงในการไปอัส
3. มีความไวต่อสัญญาณรบกวนสูง เช่น จากไฟบ้าน 50 HZ ทำให้จำนวนหยดน้ำที่ได้มีความผิดพลาด

ดังนั้น ทางผู้จัดทำจึงได้ออกแบบชุดตรวจจับหยดน้ำ ทั้งส่วนแมคคาณิกส์ และวงจรใหม่ ซึ่งมีรูปร่างและบล็อคไดอะแกรม ดังรูป

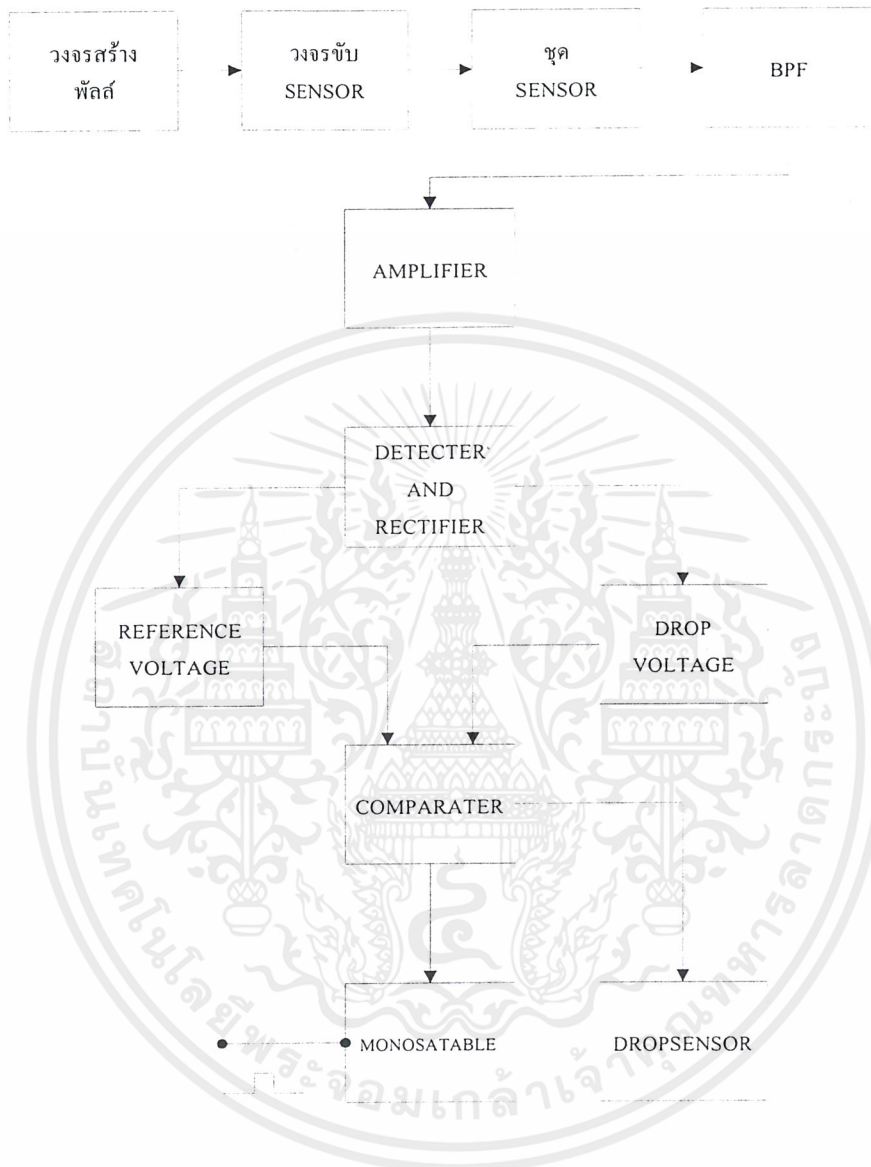


รูปที่ 3.1 แสดงส่วนประกอบของชุดตรวจจับหยดน้ำ



รูปที่ 3.2 แสดงชุดตรวจจับหยดน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

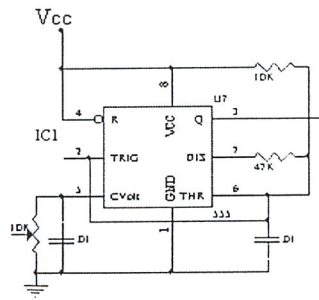


รูปที่ 3.3 แสดงบล็อกไดอะแกรมของวงจรตรวจจับหยดน้ำ

3.1 การออกแบบวงจรสร้างพัลส์ โดยใช้ Ic Timer 555

จากการทดลองวัดระยะเวลาที่หยดน้ำผ่านจุด sensor ซึ่งจะใช้เวลาประมาณ 5-10 ms การคั้งนี้วงจรสร้างพัลส์จึงไม่ควรมีความถี่ต่ำกว่า 1 KHZ ถ้าเราต้องการให้แต่ละหยดน้ำมีพัลส์ออกมาไม่น้อยกว่า 5 พัลส์ ซึ่งในการทดลองนี้จะใช้ความถี่ประมาณ 1600 HZ ซึ่งจะได้วงจรดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงการสร้างพัลส์โดยใช้ Ic 555

ซึ่งจากการคำนวณโดยใช้ค่าผิดพลาดของอุปกรณ์มากที่สุดทั้งทางบวกและทางลบ จะได้ค่าความถี่อยู่ในช่วง 1198 Hz – 1619 Hz

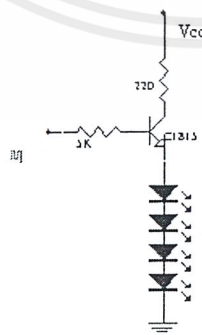
จากสูตรการคำนวณความถี่ออสซิลเลต ของ 555 คือ

$$f = \frac{1}{T} = 1.44 / (RA + 2RB) C$$

และเพื่อง่ายในการปรับ duty cycle จึงต่อ VR 10 k (ที่ขาเทสโอสของ Ic 555 คือ ขา 5 โดยวงจรสามารถปรับ duty cycle ได้ ตั้งแต่ 5% ไปจนถึง 40% ซึ่งในการทดลองนี้ ปรับ duty cycle ไว้ที่ประมาณ 20% ซึ่งเพียงพอในการทำให้จุด sensor ทำงานได้อย่างมีประสิทธิภาพ

3.2 การออกแบบวงจร Drive จุด Infrared sensor

3.2.1 ภาคส่ง ของ Infrared LED 4 ตัวต่ออนุกรมกันเพื่อเพิ่มกำลังแสงให้เพียงพอในกรณีที่กระเปาะน้ำเกลือมีไอน้ำเกาะทำให้ไม่สามารถตรวจจับหยดน้ำได้ และเพื่อให้กระแสที่ไหลผ่าน LED มีค่าไม่สูงเกินไป เป็นการยืดอายุการใช้งานของ Infrared LED ซึ่งมีวงจรดังรูป



รูปที่ 3.5 แสดงวงจร Drive จุด Infrared sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถคำนวณค่ากระแส Peak และค่ากระแสเฉลี่ยที่ไหลผ่าน LED ได้ โดยเริ่มจากพิจารณาว่า Tr. c1815 ทำงานในย่าน cutoff และ Saturated

$$\text{จาก } I_b = \frac{12 - 0.7 - 1.8(4)}{5k} = 0.82 \text{ mA}$$

2SC1815 มีค่า $\beta_{\min} = 200$ ดังนั้น $I_c = \beta I_B$

$$I_c = 200 \times 0.52 \text{ mA} = 164 \text{ mA}$$

จะได้ $V_c = 164 \text{ mA} \times 220 \Omega = 36.08 \text{ V}$

ดังนั้น Transistor ทำงานในย่าน Saturated แน่นอน

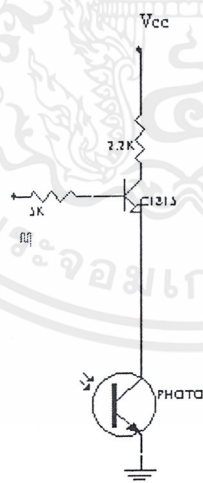
$$\text{ทำให้ได้ } I_{D_{\text{peak}}} = 12 - 0.2 - 1.8(4) = 20.9 \text{ mA}$$

แต่พัลส์ที่ Turn on ทรานซิสเตอร์ มี duty cycle 20%

ดังนั้น กระแสเฉลี่ยที่ไหลผ่าน LED

$$I_{D_{\text{avg}}} = \frac{20.9 \times 20}{100} = 4.18 \text{ mA}$$

3.2.2 ภาครับ ใช้ Infrared Photo Transistor เป็นตัวรับ ซึ่งมีวงจรดังรูป



รูปที่ 3.6 แสดงวงจรรับ IR Photo Transistor ภาครับ

จากที่ได้คำนวณในภาคส่งทำให้ทราบว่า Tr. C1815 ทำงานในย่าน saturated และ cutoff mode

ดังนั้นเราสามารถคำนวณค่ากระแส Peak และกระแสเฉลี่ยที่ไหลผ่าน อินฟราเรด โฟโต ทรานซิสเตอร์ได้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_{TR \max} = \frac{12 - 0.2}{2.2 \text{ k}} = 5.36 \text{ mA}$$

แต่พัลส์ที่ Turn on ทราานซิสเตอร์ มี duty cycle 20%

$$\text{ดังนั้น } I_{TR \text{avg}} = 5.36 \text{ mA} \times 0.2 = 1.07 \text{ mA}$$

3.3 การออกแบบวงจร Band pass Filter

จาก Transfer function ของวงจร Band pass Filter

$$T(s) = \frac{Kas}{s^2 + as + b} = \frac{K \omega_p S}{\frac{Q_p}{s^2 + \omega_p s + \omega_p^2}}$$

และ Transfer function ในรูปวงจร Rc คือ

$$T(s) = \frac{V_o}{V_{in}} = \frac{S}{\frac{R_1 C_1}{\frac{S^2 + S \left(\frac{1}{R_2 C_2} + \frac{1}{R_2 C_1} \right) + \frac{1}{R_1 R_2 C_1 C_2} + \frac{1}{R_2 R_3 C_1 C_2}}}}$$

ในการออกแบบเพื่อความง่ายเราจะเลือกใช้การป้อนกลับแบบลบ และเลือกค่า pass band

$$\text{Gain} = 1, Q = 10, f_p = 1600 \text{ HZ}$$

จาก Normalized second order BPF ที่มีการป้อนกลับแบบลบ เราจะได้ค่า

$$R_2 = 2Q$$

$$\text{Gain เมื่อ } R_3 = \alpha = 2Q^2 \text{ แต่เราต้องการค่า gain} = 1$$

$$\text{ดังนั้น } \frac{R_3}{R_1 + R_3} = \frac{1}{200} = 0.005$$

$$\text{และ } R_1 // R_3 = \frac{1}{2Q}$$

$$\text{จะได้ } \left. \begin{array}{l} \frac{R_1}{Q} = 100 \\ R_2 = 2Q \\ \frac{R_3}{Q} \end{array} \right\} \text{ที่ค่า } C_1 = C_2 = 1$$

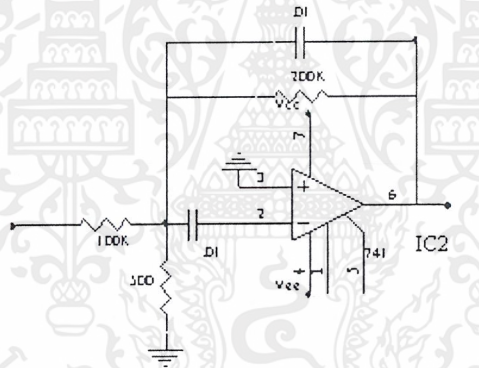
Scale frequency ลดค่า C ลง $2\pi \times 1600$ เท่า = $99.47 \mu\text{F}$

Scale Impedance โดยเลือกค่า C = $0.01 \mu\text{F}$ หรือลดค่า C ลงมา $\frac{99.47}{0.01} \approx 10,000$ เท่า

ดังนั้น จึงต้องเพิ่ม ค่า R 10,000 เท่า จะได้

$$R_1 = 100 \text{ k}\Omega, R_2 = 200 \text{ k}\Omega, R_3 = 500 \Omega$$

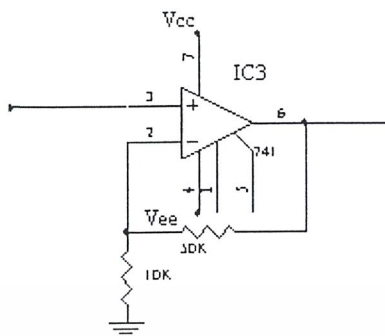
ซึ่งจะได้วงจรดังรูป



รูปที่ 3.7 แสดงวงจร Band pass filter

3.4 การออกแบบวงจร Non Inverting Amplifier

จากการทดลองสัญญาณที่ได้จากวงจร Band pass filter มีค่าสัญญาณในช่วง 0.3 V เมื่อไม่มีกระเปาะน้ำเกลือ และไม่มีหยดน้ำ และไม่เกิน 1.3 V เมื่อกระเปาะน้ำเกลือมีไอน้ำเกาะ และมีหยดน้ำ ดังนั้น จึงเลือกวงจร Non Inverting Amplifier ที่มี gain = 6 เพื่อสัญญาณที่ขยายได้ จะไม่ Saturated ซึ่งจะได้วงจรดังรูป



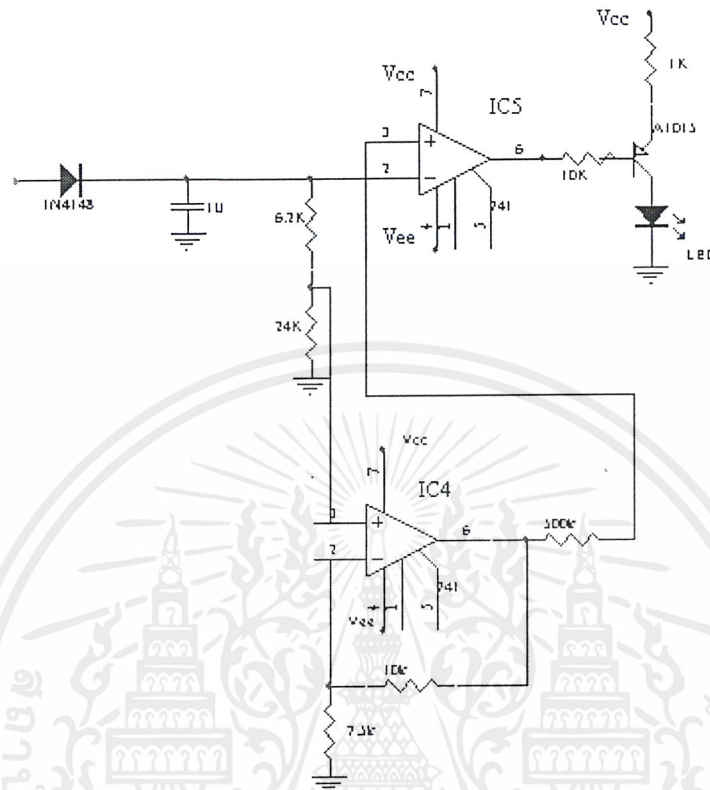
รูปที่ 3.8 แสดงวงจรขยายแบบ Non Inverting Amplifier

3.5 การออกแบบวงจร Detector

ในการตรวจจับหยดน้ำ เนื่องจาก Voltage ที่ได้เมื่อระยะหรือขนาดของกระเปาะน้ำเกลือเปลี่ยนแปลงไป หรือกระเปาะน้ำเกลือมีไอน้ำเกาะ มีค่าไม่เท่ากัน ดังนั้นเราจึงกำหนดค่าแรงดัน reference ให้สามารถเปลี่ยนแปลงตามค่าแรงดันที่เปลี่ยนแปลงไป โดยการแปลงสัญญาณที่ได้จากวงจรขยายเป็นแรงดันไฟ DC โดยผ่าน Diode IN4148 และกรองด้วย C ขนาด 1 F จากนั้นจะแยกออกเป็น 2 ส่วนเป็นแรงดัน reference และสัญญาณหยดน้ำเกลือ ส่วนของแรงดัน reference จะถูกขยายด้วย Non Inverting Amplifier เพื่อให้มีค่ามากกว่าสัญญาณขณะไม่มีหยดน้ำตลอดเวลา ซึ่งค่ากระแสสูงสุดที่ผ่าน Diode และ op Amplifier ต้องจ่ายโดยคิดที่

$$V_p \text{ สูงสุด คือ } \frac{12 \text{ V}}{30 \text{ k}} = 12 \text{ } = 0.4 \text{ mA}$$

จากนั้นจะทำการเปรียบเทียบโดยใช้ Comparator และแสดงผลด้วย LED ซึ่งค่ากระแสสูงสุดที่ไหลผ่าน LED มีค่าประมาณ 10 mA ซึ่งจะได้วงจรดังรูป



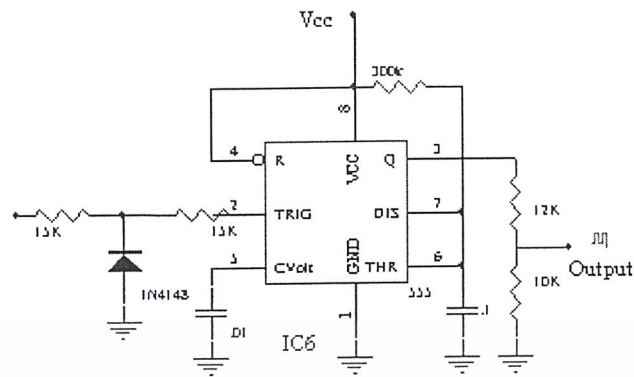
รูปที่ 3.9 แสดงวงจรส่วน Detector และ Comparator

3.6 การออกแบบวงจร Mono Stable

เพื่อให้พัลส์ของหยดน้ำเกลือมีขนาดเท่า ๆ กัน จึงใช้ Ic Timer 555 ต่อเป็นวงจร Mono Stable โดยให้ค่าความกว้างพัลส์ประมาณ 30 ms

$$\begin{aligned} \text{ซึ่งคำนวณได้จากสูตร } T &= 1.1 RC = 1.1 (300 \text{ k}) (0.1 \mu\text{F}) \\ &= 33 \text{ ms} \end{aligned}$$

เนื่องจากพัลส์ที่ได้จากการ Comparator ด้วย op Amplifier มีทั้งบวกและลบ ดังนั้นจึงต่อ Diode ลง GND ไว้ เพื่อตัดไฟลบทิ้งไป และที่เข้าที่พุทของ Ic 555 ได้ต่อ R เพื่อลดแรงดันจาก 12 volt ให้เหลือ 5 volt เข้าสู่ Micro controller ต่อไป

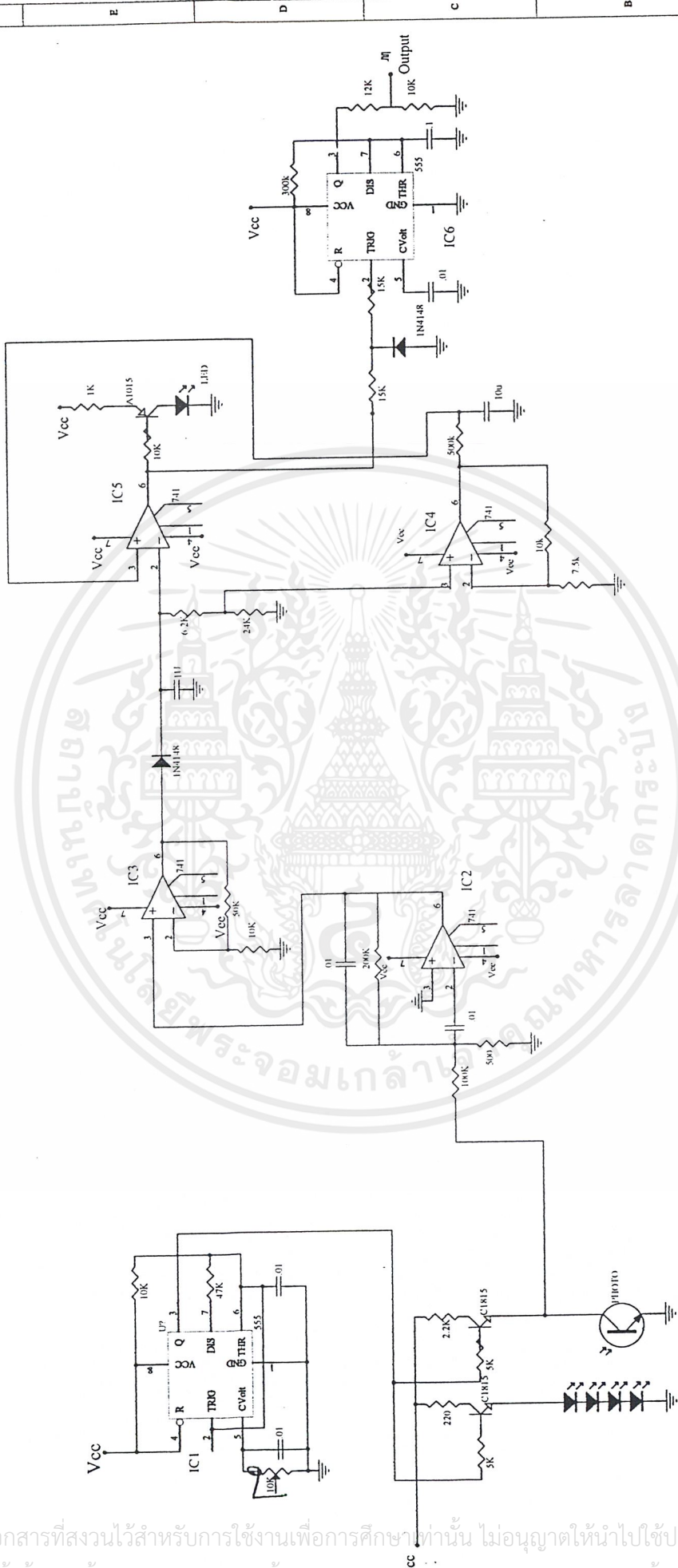


รูปที่ 3.10 แสดงวงจร Mono Stable Ic 555



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DROPS SENSOR CIRCUIT



รูปที่ 3.11 แสดงวงจรตรวจจับหยดน้ำ

Title		DROPS SENSOR CIRCUIT	
Size	Number	Revision	
B			
Date:	13/05/2011	Sheet #	8
File:	D:\PROJECT\CIRCUIT\KOPROSS.SCH	Drawn By:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 มีค่าธรรมเนียมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

วงจรขั้วดีซีมอเตอร์และวงจรตรวจสอบฟองอากาศ

4.1 วงจรขั้วดีซีมอเตอร์

วงจรขั้วดีซีมอเตอร์จะรับสัญญาณควบคุมการทำงานมาจาก 3 แหล่งด้วยกัน คือ

1. สัญญาณควบคุมการหมุนจาก CPU โดยผ่านทาง Opto Isolated เบอร์ 4N25 ซึ่งจะส่งรหัสควบคุมดังนี้

0 0 มอเตอร์หยุดหมุน

1 1 มอเตอร์หยุดหมุน

0 1 มอเตอร์หมุนทางซ้าย

1 0 มอเตอร์หมุนทางขวา

2. สัญญาณควบคุมจาก Limit Switch เมื่อดีซีมอเตอร์หมุนเข้าหรือออกจนสุดจะชนกับ Limit Switch ซึ่งจะทำการหยุดมอเตอร์และรอคำสั่งให้หมุนกลับอีกทางหนึ่ง

3. สัญญาณควบคุมจากวงจรตรวจจับฟองอากาศ เมื่อมีฟองอากาศเกิดขึ้นมอเตอร์จะหมุนเข้า โดยอัตโนมัติ เพื่อหยุดไม่ให้มีน้ำเกลือไหลเข้าสู่ตัวคนไขโดยทันที

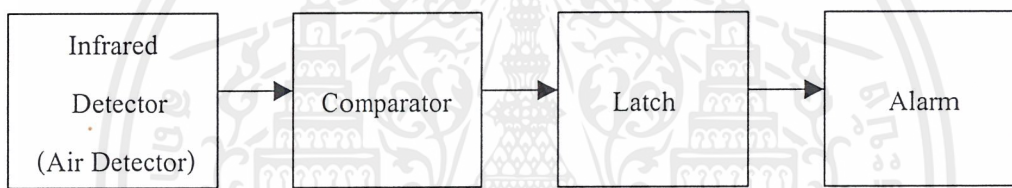
การทำงานของวงจรขั้วดีซีมอเตอร์เริ่มจาก CPU ส่งสัญญาณพัลส์ผ่านทาง Opto Isolate ผ่านชุดช่วงเวลา R5, R6, C1 และ D1 โดย R5 และ C1 จะทำหน้าที่ช่วงเวลาช่วงมอเตอร์ Turn on และในช่วง Turn off จะช่วงเวลาผ่านทาง R6 และ D1 ในกรณีของ R7, R8, C2 และ D2 ก็ทำหน้าที่เดียวกัน IC4049 จะทำหน้าที่ป้องกันไม่ให้ Transistor ภาค output Turn on พร้อมกัน IC 4012 ทำหน้าที่รับสัญญาณจาก CPU รวมทั้งสัญญาณ Switch ตรวจเช็ค IN/OUT Limit และสัญญาณจากวงจรตรวจจับฟองอากาศ ซึ่งสัญญาณทั้งหมดจะส่งผ่านเข้ามายัง IC Op-Amp 741 ซึ่งต่อไปเป็น Diff-Amp โดยให้ output เป็นบวกลบ หรือ 0 ขึ้นอยู่กับสัญญาณควบคุมทั้งหมดซึ่งจะทำให้ Transistor output Turn on และ turn off สลับกันหรือหยุดทำงานพร้อมกันในกรณี ต้องการให้มอเตอร์หยุดหมุน

จากการทดลองดีซีมอเตอร์กินกระแสสูงสุดประมาณ 700 mA ซึ่งในวงจรใช้ Transistor เบอร์ B507 และ D131 ซึ่งทนกระแสได้ถึง 4 Amp ซึ่งเพียงพอในการขั้วดีซีมอเตอร์หรือในกรณีที่ต้องการเปลี่ยนมอเตอร์ที่กินกระแสสูงขึ้นก็สามารถทำได้ ส่วน Diode D3 และ D4 ทำหน้าที่ป้องกันการ Turn on ของ Transistor output อันเนื่องมาจากแรงดันออฟเซตของ Op-Amp ซึ่งอาจเกิดขึ้นได้ซึ่งจะทำให้ดีซีมอเตอร์หมุนผิดพลาดได้

4.2 วงจรตรวจจับฟองอากาศ

การทำงานของวงจรตรวจจับฟองอากาศ

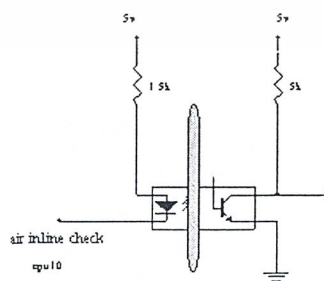
4.2.1 การทำงานของวงจรเริ่มจากชุด Infrared sensor ทำการตรวจจับฟองอากาศจากนั้นจะทำการเปรียบเทียบกับแรงดันอ้างอิงที่ตั้งไว้โดย Comparator ซึ่งในกรณีไม่มีฟองอากาศ Output ของ Comparator จะเป็นบวกและมีฟองอากาศเกิดขึ้น Output ของ Comparator จะเป็นลบ จึงใช้ Diode 1N4148 ตัดส่วนลบทิ้งให้ Output เหลือเป็น 0 Volt ไปทริก IC Timer 555 ซึ่งต่อเป็นวงจร Latch โดยจะทำให้ Output ของ IC 555 ที่ขา 3 เป็น High ตลอดไปจนกว่า จะมีการแก้ไขให้ฟองอากาศหมดไป และมีการกด Reset Switch ที่ขา 5 ของ IC 555 ซึ่งพัลส์ที่ได้จากวงจรตรวจจับฟองอากาศจะไปควบคุมให้วงจรขับดีซีมอเตอร์หมุนเข้าเพื่อหยุดอัตราการไหล โดยอัตโนมัติ รวมทั้งไปขับให้ส่วน LED แสดงผลและ Alarm ดัง โดยทันที



รูปที่ 4.1 Block diagram ของวงจรตรวจจับฟองอากาศ

4.2.2 การออกแบบวงจรส่วน Air Detector

ในการออกแบบชุดตรวจจับฟองอากาศจะใช้ Infrared LED และ Infrared Photo Transistor ซึ่งมีวงจรดังรูป



รูปที่ 4.2 วงจรส่วนตรวจจับฟองอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

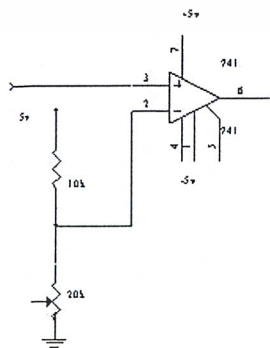
ซึ่งจากการทดลองโดยการเปลี่ยนค่าความต้านทานหลายๆค่า เพื่อให้ได้แรงดันขณะมีฟองอากาศและไม่มีฟองอากาศอยู่ในช่วงที่เหมาะสมซึ่งผลการทดลองเป็นดังตารางจึงเลือกใช้ค่า R_D ที่ 1.5 K และเลือกค่า R_C ที่ 5K เพื่อให้ทนกระแสสูงสุดที่ไหลผ่าน Infrared Photo Transistor มีค่าไม่สูงเกินไปและต่ำเกินไปซึ่งจะได้ V_{ce} ขณะไม่มีฟองอากาศและมีฟองอากาศอยู่ในช่วง 2 Volt ถึง 3 Volt ซึ่งเป็นค่าที่อยู่ในช่วงกลางๆจึงเลือกใช้นี้

$R_D (\Omega)$	$R_C (\Omega)$	V_{ce} (Volt)	
		มีฟองอากาศ	ไม่มีฟองอากาศ
1k	5k	0.22	1.1
1.2k	5k	0.53	1.3
1.5k	5k	1.78	3.1
1.8k	5k	2.51	3.82
2k	5k	2.94	4.1
2.2k	5k	3.45	4.82

ตารางที่4.1 ตารางแสดงความสัมพันธ์ของ R_D และ V_{CE} ขณะมีฟองอากาศและไม่มีฟองอากาศ

4.2.3 การออกแบบวงจร Comparator

จากการทดลองเราจะได้ Voltage ขณะมีฟองอากาศและไม่มีฟองอากาศอยู่ในช่วง 2 Volt ถึง 3 Volt ดังนั้น จึงตั้งแรงดัน Reference ไว้ที่ประมาณ 2.5 V แต่ถ้าหากค่าของ Voltage ขณะมีฟองอากาศเปลี่ยนแปลงไปก็สามารถปรับค่าโดยปรับ VR ที่ขา 2 ของ Op-Amp ซึ่งมีวงจรดังรูป



รูปที่ 4.3 แสดงวงจร Comparator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

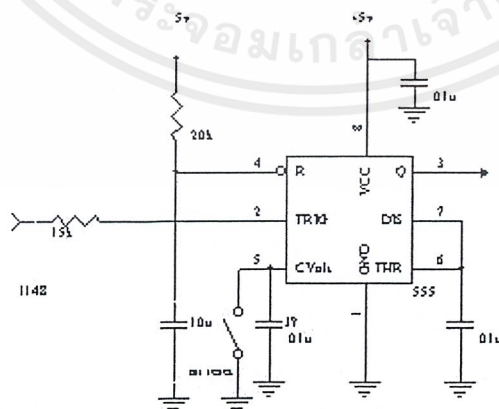
4.2.4 การออกแบบวงจร Latch

ในการออกแบบครั้งนี้จะใช้ IC timer 555 ต่อในลักษณะ Mono Stable แต่ให้ค่าความต้านทานที่ขา 6 และขา 7 ซึ่งเป็น Threshold และ discharge เป็น Infinity ทำให้ Output ที่ขา 3 เป็น High ตลอดไป เมื่อมีพัลส์เข้ามาที่ขา 2 จนกว่าแรงดันที่ขา 2 จะเป็น High อีกครั้งหนึ่งและมีการทำให้ Voltage ที่ขา Threshold มีค่าสูงกว่าค่าแรงดัน V_{UT} ซึ่งมีค่าประมาณ $2/3 V_{cc}$ Output ของ IC 555 จึงจะกลับเป็น 0 อีกครั้งหนึ่ง ซึ่งในวงจรขณะใช้ Switch ต่อที่ขา 5 ซึ่งเป็นขาควบคุมแรงดัน ทำให้แรงดันที่ขา Threshold (ขา 6) สูงกว่า V_{UT}

ขาทรigger 2	ขาเทสโสล 6	ผลลัพธ์	
		ขา Output ขา 3	ขา discharge ขา 7
ต่ำกว่า V_{LT}	ต่ำกว่า V_{UT}	High	Open Circuit
ต่ำกว่า V_{LT}	สูงกว่า V_{UT}	สถานะเดิม	
สูงกว่า V_{LT}	ต่ำกว่า V_{UT}	สถานะเดิม	Ground
สูงกว่า V_{LT}	สูงกว่า V_{UT}	Low	

$$\text{โดยที่ } V_{UT} = \frac{2 V_{cc}}{3} ; V_{LT} = \frac{V_{cc}}{3}$$

ตารางที่ 4.2 แสดงสถานะการทำงานของ IC Timer 555



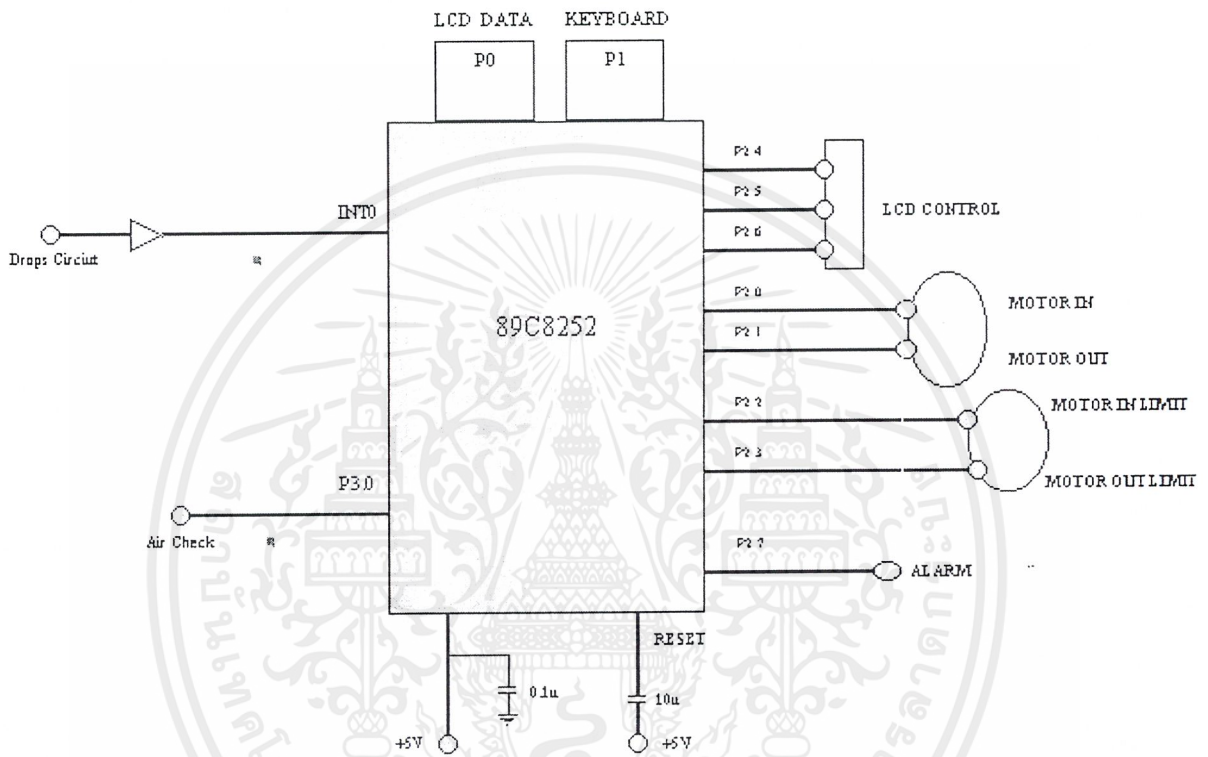
รูปที่ 4.4 แสดงวงจร Latch โดยใช้ Ic 555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การเขียนโปรแกรมควบคุมการทำงานของเครื่องให้น้ำเกลือ

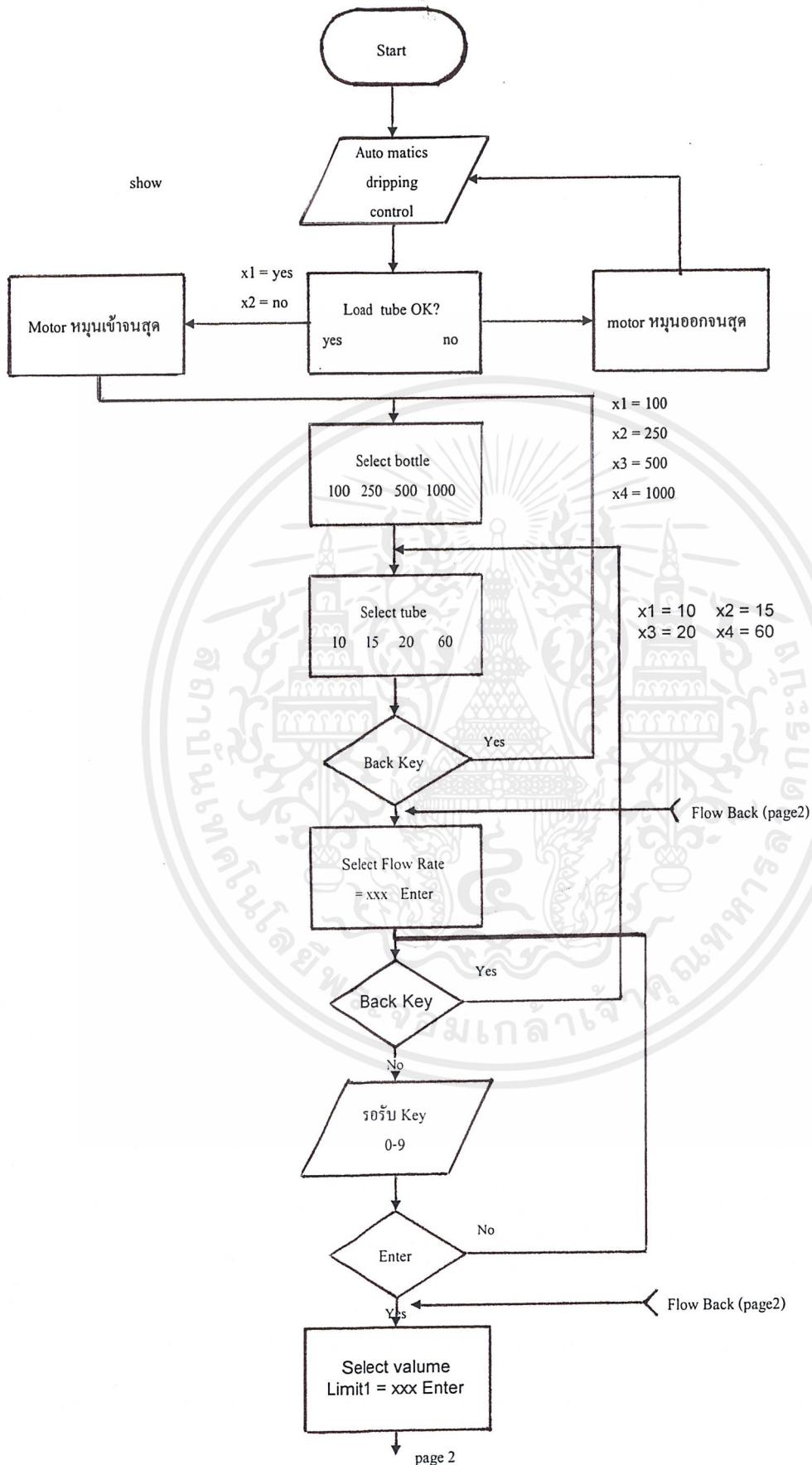
การควบคุมการทำงานของเครื่องควบคุมการให้น้ำเกลือจะใช้ CPU ในตระกูล MCS-51 ซึ่งต่อใช้งานต่างๆ ดังรูป



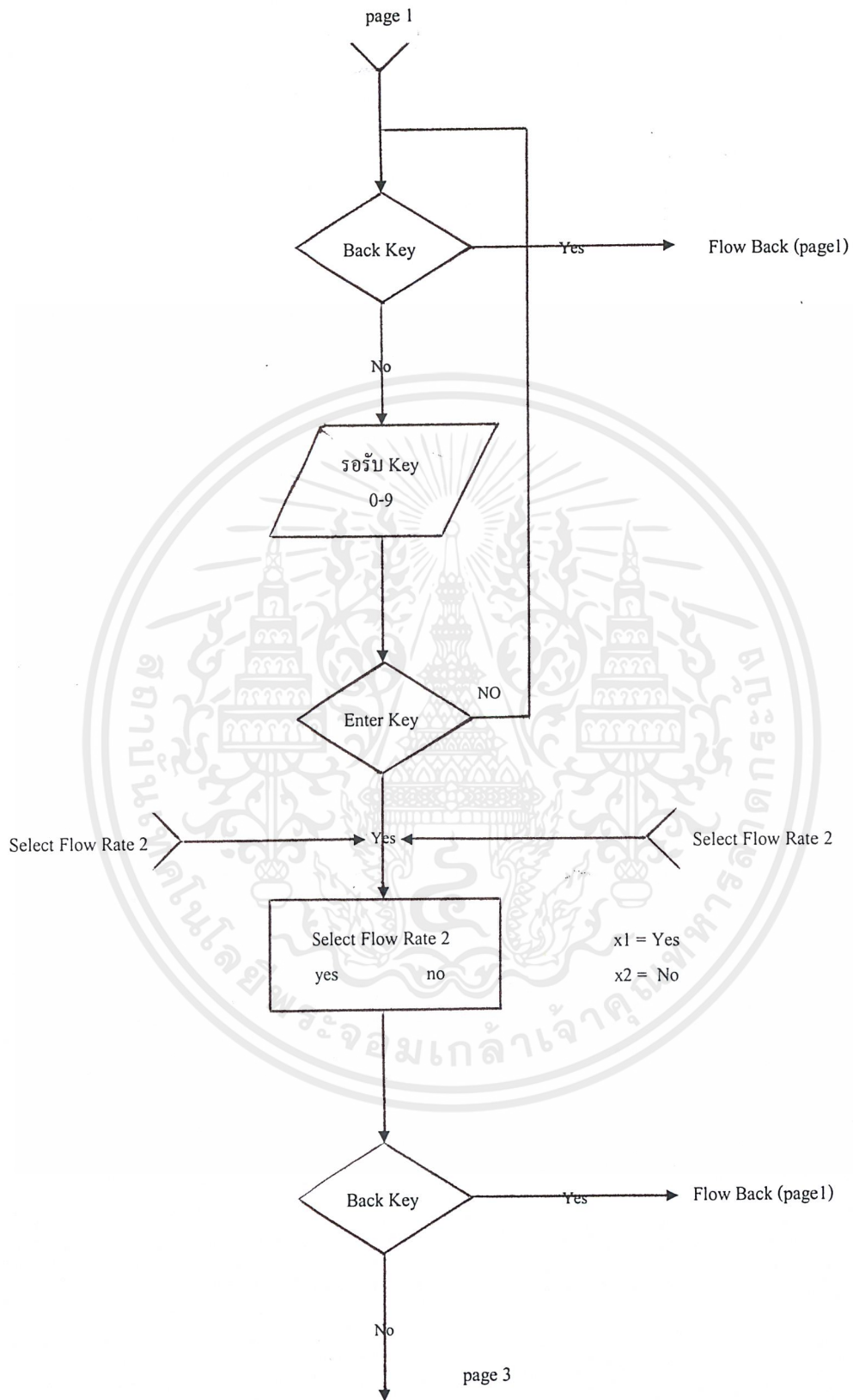
รูปที่ 5.1 แสดงการต่อขาใช้งานของ MCS-51

การเขียนโปรแกรมควบคุมการทำงานของเครื่องเริ่มจากรับอินพุตต่างๆทาง Key board และแสดงผลที่ LCD จากนั้นจะควบคุมการหมุนของ ดีไซน์มอเตอร์ ให้ได้อัตราไหลตามที่ตั้งไว้ ซึ่งการควบคุมอัตราการไหลจะเป็นไปในลักษณะ การตรวจเช็คระยะเวลาระหว่างหยุด จากนั้นนำมาคำนวณเปรียบเทียบกับค่าที่ตั้งไว้แล้วจึงสั่งให้มอเตอร์หมุนเข้าหรือหมุนออกตามระยะห่างระหว่างหยุด ในส่วนแสดงผลในลักษณะเฉลี่ยใน 1 นาที

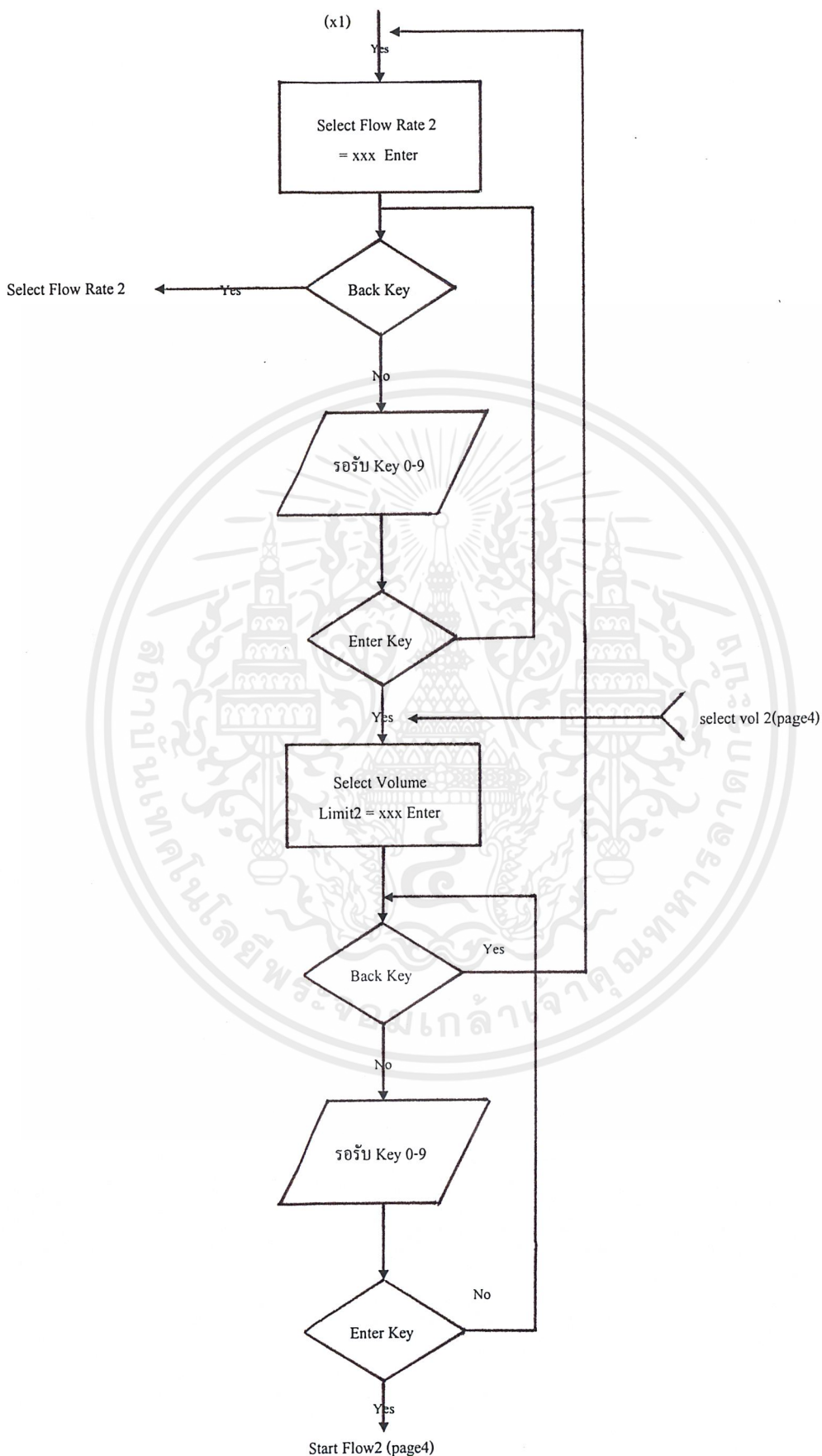
ในส่วนของการแสดงผลปริมาตรที่ให้ไปแล้วนั้นจะแสดงผลเป็น CC และจะแสดงผลทุกๆ 1 CC ดังนั้นจะขึ้นอยู่กับขนาดของสายที่เลือกไว้ เช่น สายขนาด 15 หยด/ซีซี เมื่อครบ 15 หยดก็จะแสดงผล 1 ครั้ง และในส่วนของการรับค่าอัตราการไหลจะกำหนดให้ใส่ค่าได้ไม่เกิน 250 หยดต่อนาที เนื่องจากการทดลองใช้เครื่องให้น้ำเกลือควบคุมอัตราการไหลมีโอกาสที่น้ำเกลือจะไหลเป็นสายต่อเนื่องจนไม่สามารถควบคุมได้ เมื่ออัตราการไหลมากกว่า 250 หยดต่อนาที จะมี Flow Chart การทำงานดังนี้



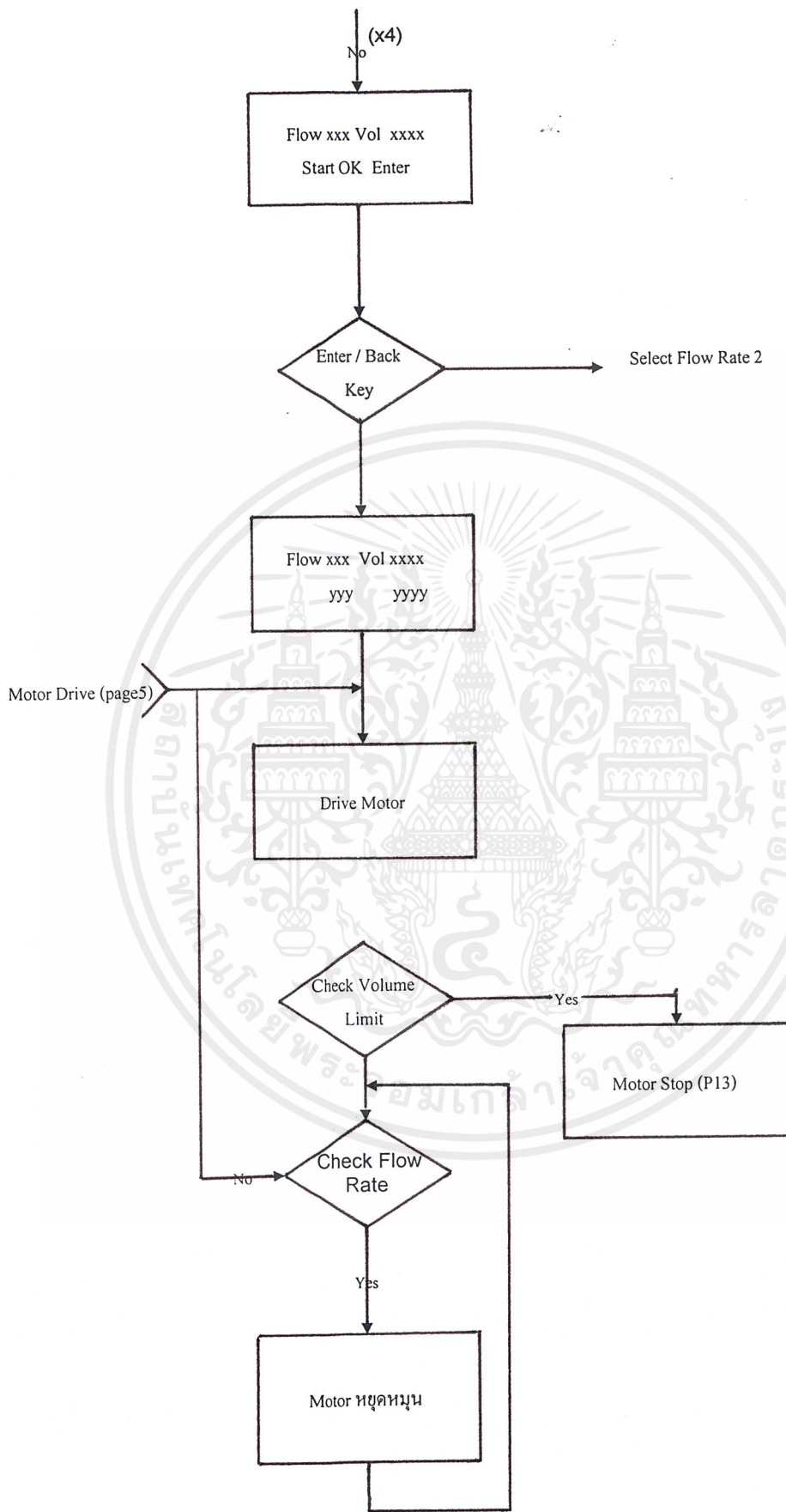
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า. ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



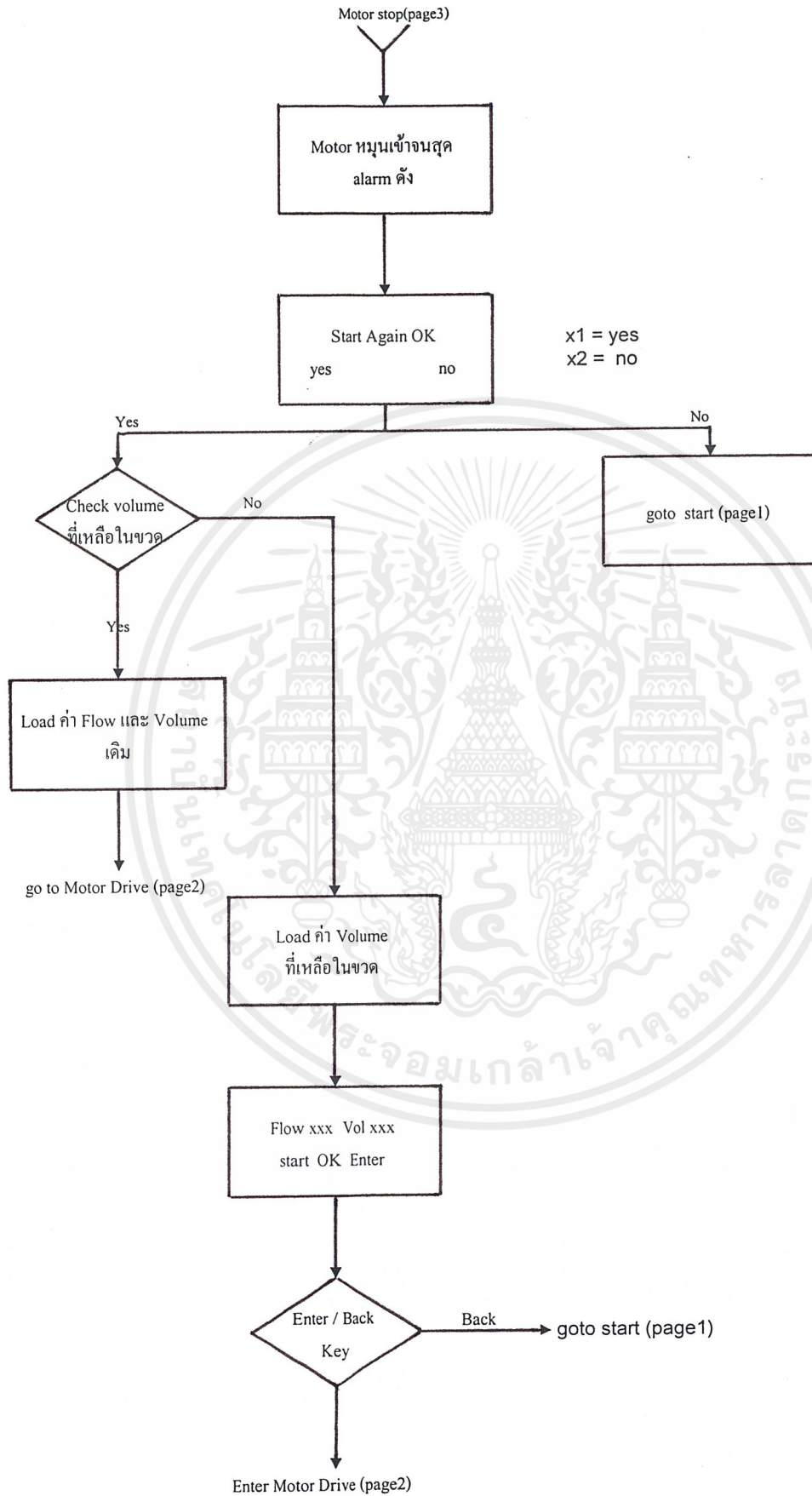
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



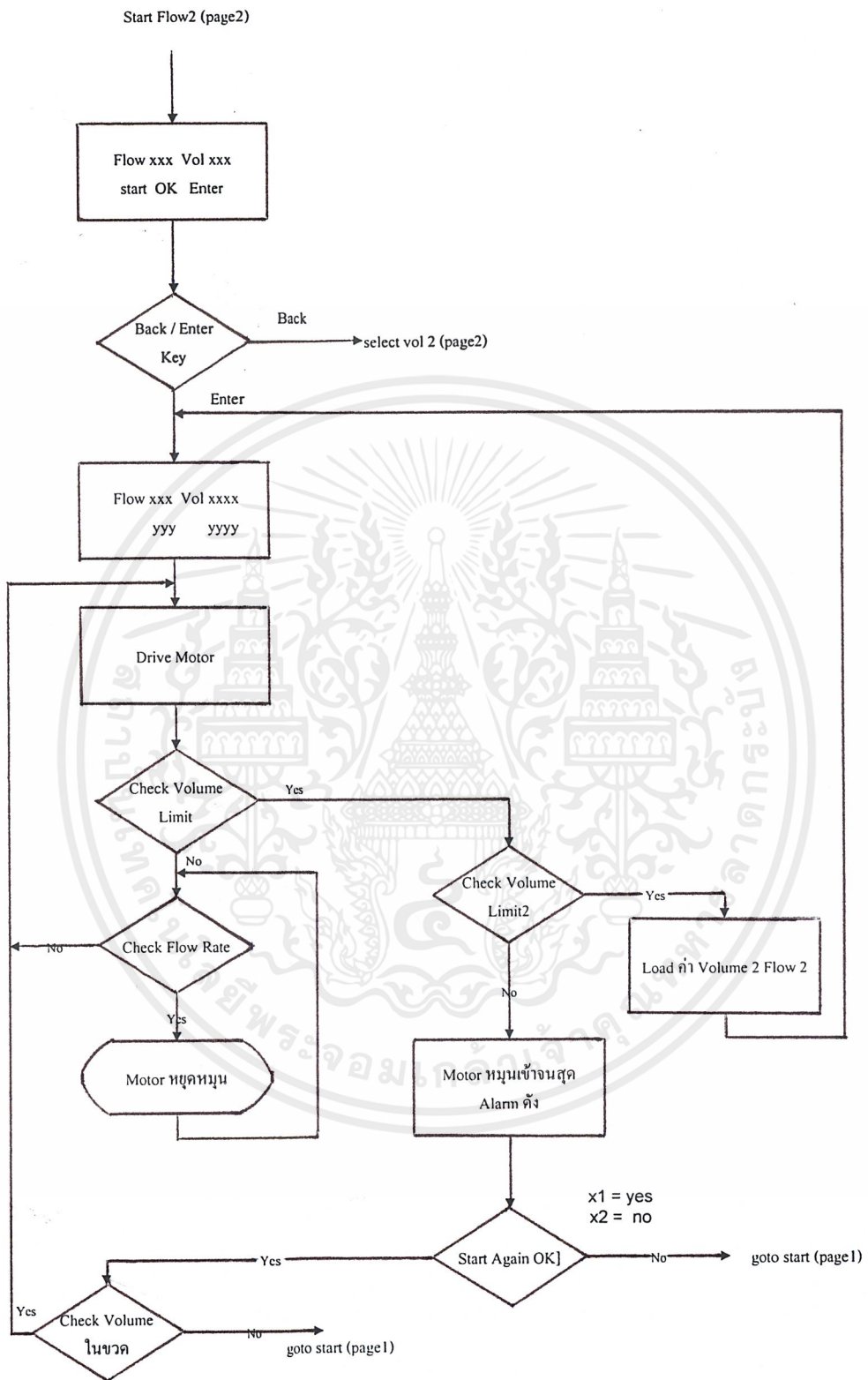
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Key Board

^	^	^	^
0	1	2	3
4	5	6	7
8	9	Back <	Enter

รูปที่ 5.3 แสดง Key Broad

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลอง

การทดลองที่ 6.1 การเปลี่ยนแปลงของอัตราการไหลต่อปริมาณของน้ำเกลือ

จุดประสงค์ ศึกษาการเปลี่ยนแปลงของการไหลของน้ำเกลือเมื่อปริมาณของน้ำเกลือลดลง

อุปกรณ์ประกอบการทดลอง

1. ชุดเซตน้ำเกลือขนาด 500 มิลลิลิตร , 15 หยก/ซีซี
2. นาฬิกาจับเวลา

ขั้นตอนการทดลอง

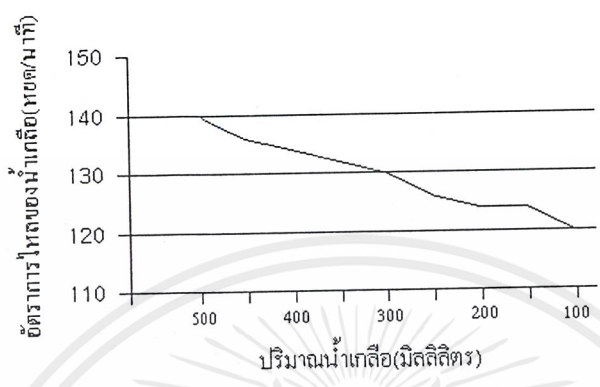
1. ต่อชุดเซตน้ำเกลือ
2. ตั้งอัตราการไหลของน้ำเกลือไว้ที่ 140 หยด/นาที
3. วัดอัตราการไหลของหยดน้ำเกลือเมื่อปริมาณของน้ำเกลือลดลงทุกๆ 50 มิลลิลิตร
4. สังเกตการเปลี่ยนแปลงอัตราการไหล
5. บันทึกผลการทดลอง

ปริมาณของน้ำเกลือ (มิลลิลิตร)	อัตราการไหล (หยด/นาที)
500	140
450	140
400	136
350	134
300	132
250	130
200	126
150	124
100	124
50	120

ตารางที่ 6.1 แสดงผลการทดลองการเปลี่ยนแปลงของการไหลของน้ำเกลือเมื่อปริมาณของน้ำเกลือลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองจะพบว่าเมื่อปริมาณน้ำเกลือมีปริมาณลดลงนั้นจะทำให้อัตราการไหลของน้ำเกลือมีค่าต่ำลงไปด้วย ดังนั้นในการควบคุมอัตราการไหลของน้ำเกลือคอลโทรลเลอร์จึงจำเป็นที่จะต้องตรวจสอบอัตราการไหลของน้ำเกลืออยู่ตลอดเวลา



รูปที่ 6.1 แสดงความสัมพันธ์ปริมาณน้ำเกลือกับอัตราการไหลของน้ำเกลือ

การทดลองที่ 6.2 การทดลองวัดอัตราการไหลที่ค่าต่างๆ เปรียบเทียบกับค่าที่ตั้งไว้จุดประสงค์

เพื่อให้ทราบค่าความเที่ยงตรงของอัตราการไหลที่ค่าต่าง ๆ

อุปกรณ์ประกอบการทดลอง

1. ชุดชุดสายน้ำเกลือขนาด 15 หยด/ซีซี
2. ขวดน้ำเกลือ
3. เครื่องควบคุมการให้น้ำเกลือ
4. นาฬิกาจับเวลา

ขั้นตอนการทดลอง

1. ต่อชุดสายน้ำเกลือ ขวดน้ำเกลือเข้ากับเครื่องควบคุมการให้น้ำเกลือ
2. นับจำนวนหยดน้ำเกลือเปรียบเทียบกับที่เครื่องควบคุมการให้น้ำเกลือนับได้
3. บันทึกอัตราการไหลที่ค่าต่าง ๆ
4. ทำการทดลองซ้ำจำนวน 5 ครั้ง และหาค่าเฉลี่ยต่างๆ
5. นำค่าที่ได้ไปพลอตกราฟ

ลำดับ ที่	Flow Rate(หยด/ นาที)	อัตราการไหลของน้ำเกลือที่วัดได้จริง(หยด/นาที)										
		1 (นาที)	2 (นาที)	3 (นาที)	4 (นาที)	5 (นาที)	6 (นาที)	7 (นาที)	8 (นาที)	9 (นาที)	10 (นาที)	%er (Ave)
1	5	9	8	8	7	8	8	7	8	9	10	18
% Error		80	60	60	40	60	60	40	60	80	100	
2	10	11	12	11	11	11	12	11	10	12	11	1.25
% Error		10	20	10	10	10	20	10	0	20	10	
3	20	18	21	22	18	16	19	20	19	16	19	9
% Error		10	5	10	10	20	5	0	5	20	5	
4	30	31	29	34	32	29	32	31	32	32	30	5.33
% Error		3.83	3.33	13.33	6.67	3.33	6.67	3.33	6.67	6.67	0.00	
5	40	32	42	45	46	44	45	45	46	45	44	12.5
% Error		20	5	12.5	15	10	12.5	12.5	15	12.5	10	
6	50	49	50	51	49	48	49	55	49	48	51	3
% Error		36	16	10	8	12	10	10	8	10	12	
7	60	54	56	55	56	54	55	54	56	55	54	8.50
% Error		10	6.67	8.33	6.67	10	8.33	10	6.67	8.33	10	
8	70	63	69	67	66	68	65	64	66	68	67	5.29
% Error		5	2.50	3.75	2.50	1.25	2.50	3.75	1.25	0	1.25	
9	80	76	78	77	78	81	78	77	81	80	79	2.38
% Error		5	2.50	3.75	2.50	1.25	2.50	3.75	1.25	0	1.25	
10	90	101	86	80	88	89	91	93	87	88	91	4.22
% Error		12.22	4.44	11.11	2.22	1.11	1.11	3.33	3.33	2.22	1.11	
11	100	97	99	96	93	98	97	95	98	96	99	3.00
% Error		3	1	4	7	2	3	5	2	4	1	
12	120	106	114	112	114	115	114	117	119	114	116	4.92
% Error		11.67	5	6.67	5	4.17	5	2.5	0.83	5	3.33	
13	140	134	136	142	137	142	137	142	136	142	137	2.22
% Error		4.29	2.86	1.43	2.14	1.43	2.14	1.43	2.86	1.43	2.14	
14	160	152	154	159	160	158	157	161	158	159	162	1.63
% Error		8.89	2.22	1.67	2.22	1.67	0.56	1.67	2.22	1.11	0	

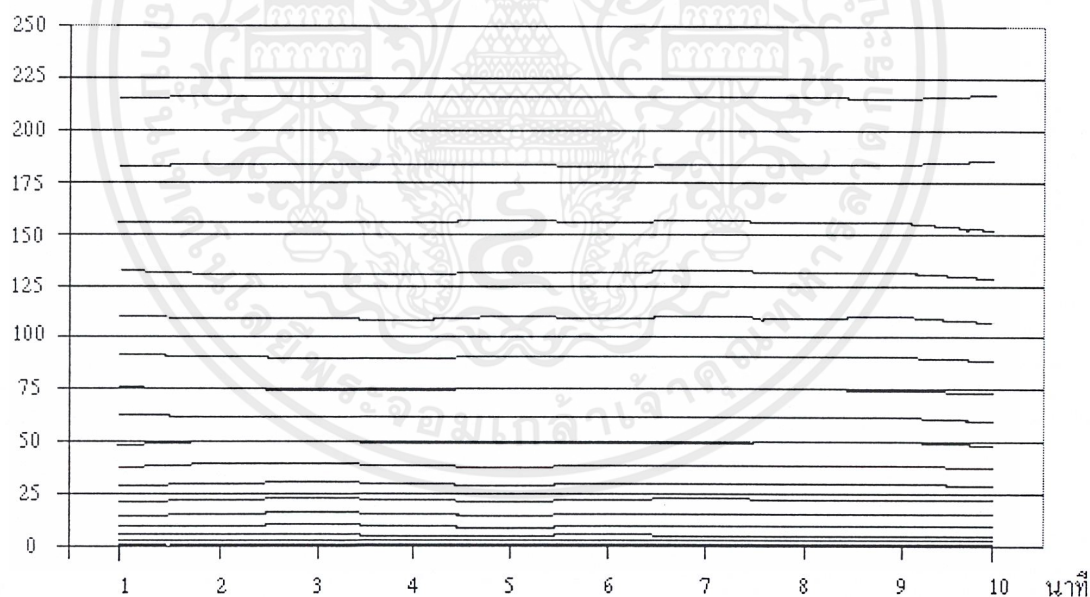
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้สำหรับครูอาจารย์ในการศึกษาเท่านั้น ไม่สามารถนำข้อมูลไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ ที่	Flow Rate(หยด/ นาທີ)	อัตราการไหลของน้ำเกลือที่วัดได้จริง(หยด/นาທີ)										
		1 (นาທີ)	2 (นาທີ)	3 (นาທີ)	4 (นาທີ)	5 (นาທີ)	6 (นาທີ)	7 (นาທີ)	8 (นาທີ)	9 (นาທີ)	10 (นาທີ)	%er (Ave)
15	180	164	184	183	184	177	179	183	184	178	180	2.22
% Error		8.89	2.22	1.67	2.22	1.67	0.56	1.67	2.22	1.11	0	
16	200	194	202	203	207	200	202	201	202	201	203	1.35
% Error		3	1	1.5	3.5	0	1	0.5	1	0.5	1.5	
17	240	232	239	241	240	238	242	239	243	240	239	0.92
% Error		3.33	0.42	0.42	0	0	0.83	0.83	1.25	0	0.42	
18	250	245	249	249	249	248	251	249	249	252	250	0.60
% Error		2	0.40	0.40	0.40	0.80	0.40	0.40	0.40	0.80	0	

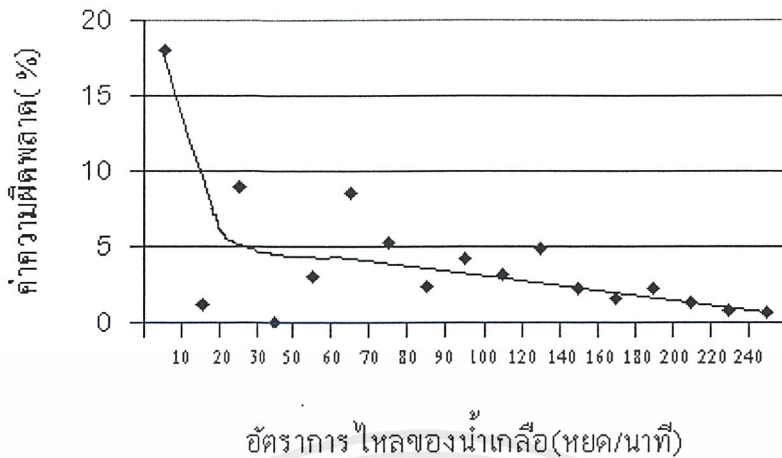
ตารางที่ 6.2 แสดงผลการทดลองการทดลองวัดอัตราการไหลที่ค่าต่างๆ เปรียบเทียบกับค่าที่ตั้งไว้

อัตราไหลของน้ำเกลือ(หยด/นาທີ)



รูปที่ 6.2 แสดงความสัมพันธ์ระหว่างอัตราการไหลที่ค่าต่างๆ เปรียบเทียบกับค่าที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดง ค่าความผิดพลาดเฉลี่ย(%)ต่ออัตราการไหลที่ตั้งไว้

ผลการทดลอง

เครื่องควบคุมการให้น้ำเกลือสามารถควบคุมอัตราการไหลได้ดีที่ค่าอัตราการไหลสูง ๆ แต่ในอัตราการไหลที่ต่ำกว่า 5 หยดต่อนาที จะมีค่าความผิดพลาดสูงกว่า 10 % และไม่สามารถตั้งอัตราการไหลได้เกินกว่า 250 หยดต่อนาที เพราะน้ำเกลืออาจไหลเป็นสายอย่างต่อเนื่อง จนทำให้ชุดตรวจจับหยดไม่สามารถตรวจนับหยดได้

การทดลองที่ 6.3 การวัดปริมาตรของน้ำเกลือที่ต้องการให้กับปริมาตรที่ออกมาจริง จุดประสงค์

เพื่อทราบค่าความผิดพลาดของปริมาณน้ำเกลือที่ตั้งและที่ให้ออกมาจากเครื่องให้น้ำเกลือ

อุปกรณ์ประกอบการทดลอง

1. ชุดเซตสายน้ำเกลือ ขนาด 15 หยด/ซีซี
2. ขวดน้ำเกลือ
3. เครื่องควบคุมการให้น้ำเกลือ
4. หลอดฉีดยาที่มีแถบบอกปริมาตร

ขั้นตอนการทดลอง

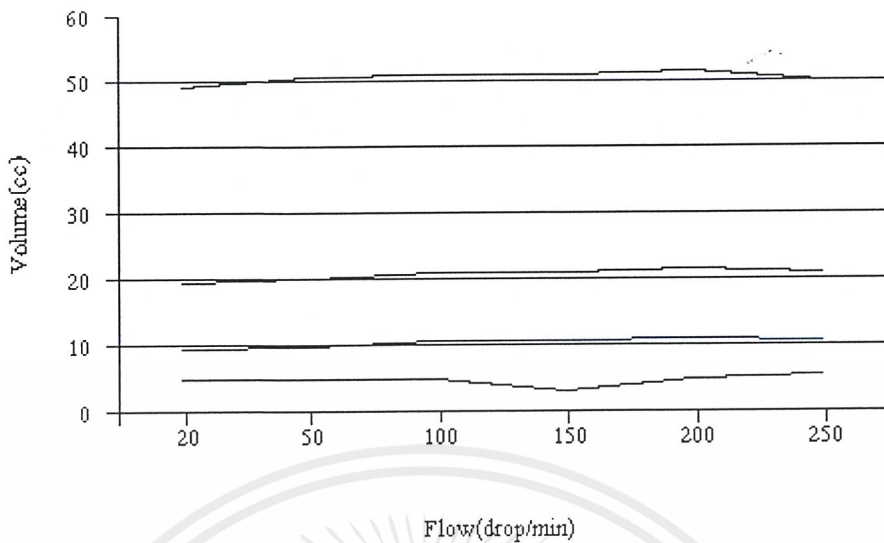
1. ต่อชุดเซตสายน้ำเกลือ ขวดน้ำเกลือเข้ากับเครื่องควบคุมการให้น้ำเกลือ
2. ตั้งอัตราการไหลคงที่ ที่ค่าหนึ่ง ๆ
3. เปลี่ยนปริมาตรที่ต้องการให้โดยให้อัตราการไหลคงที่ หลาย ๆ ค่า
4. เปลี่ยนอัตราการไหล และทำซ้ำข้อ 3.
5. บันทึกผลการทดลองและเขียนกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

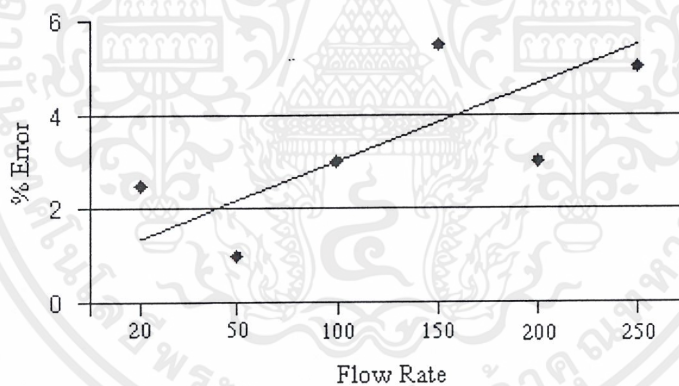
Flow Rate	ปริมาตร(cc)	ปริมาตรที่วัดได้	Error(%)	Average(Error %)
20	5	5	0	2.5
	10	9.5	5	
	20	19.5	3	
	50	49	2	
50	5	5	0	1
	10	9.75	3	
	20	20	0	
	50	50.5	1	
100	5	5	0	3
	10	10.5	5	
	20	21	5	
	50	51	2	
150	5	5.5	10	5.5
	10	10.5	5	
	20	21	5	
	50	50	2	
200	5	5	0	3
	10	11	10	
	20	21.5	8	
	50	51.5	3	
250	5	5.5	10	5
	10	10.5	5	
	20	21	5	
	50	50	0	

ตารางที่ 6.3 แสดงผลการทดลองการวัดปริมาตรของน้ำเกลือที่ต้องการให้กับ ปริมาตรที่ออกมาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 แสดงความสัมพันธ์ระหว่างปริมาตรกับอัตราการไหลของน้ำเกลือที่ค่าต่างๆ



รูปที่ 6.5 แสดงเปอร์เซ็นต์ความผิดพลาดเทียบกับอัตราการไหลของน้ำเกลือที่ค่าต่างๆ

ผลการทดลอง

เครื่องควบคุมการให้น้ำเกลือสามารถควบคุมปริมาตรของน้ำเกลือที่ออกมาจริงได้ใกล้เคียงกับค่าที่ตั้งไว้ ซึ่งขึ้นกับจำนวนหยดของชุดเซต น้ำเกลือว่ามีขนาดตรงกับที่บอกไว้เพียงใด เช่น 15 หยด/ซีซี ได้ 15 หยด ทุกๆ ครั้งหรือไม่เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.4 การทดลองการทำงานของชุดตรวจจับหยดน้ำเกลือเมื่อความเข้มแสงเปลี่ยนไป จุดประสงค์ เพื่อทราบว่าเมื่อใช้งานชุดตรวจจับหยดน้ำเกลือในบริเวณที่มีแสงสว่างมาก เช่นในห้องผ่าตัดสามารถใช้งานได้ตามปกติ

อุปกรณ์ประกอบการทดลอง

1. ชุดชุดสายน้ำเกลือขนาด 15 หยด/ลูกบาศก์เซนติเมตร
2. ขวดน้ำเกลือ
3. ชุดตรวจจับหยดน้ำเกลือ
4. ออสซิลโลสโคป
5. หลอดไฟขนาด 100 W พร้อมชุดหรีไฟ
6. นาฬิกาจับเวลา

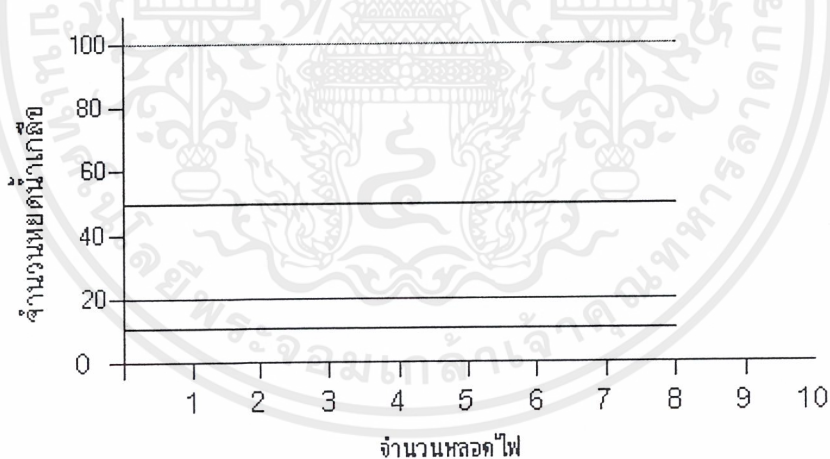
ขั้นตอนการทดลอง

1. ต่อชุดสายน้ำเกลือ ขวดน้ำเกลือ และนำชุดตรวจจับหยดน้ำเกลือที่บริเวณกระเปาะน้ำเกลือ
2. ใช้ออสซิลโลสโคปวัดสัญญาณเอาท์พุทของวงจรตรวจจับหยดน้ำเกลือ สังเกตการเปลี่ยนแปลงของเอาท์พุท
3. ปรับความสว่างของหลอดไฟไปที่ระดับต่างๆ
4. บันทึกความสัมพันธ์ของอัตราการไหลเมื่อความสว่างของหลอดไฟเปลี่ยนแปลงไป
5. เปลี่ยนอัตราการไหลไปที่ระดับต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนหลอดไฟ	จำนวนหยดน้ำที่นับได้(หยด)		
1	20	50	100
2	20	50	100
3	20	50	100
4	20	50	100
5	20	50	100
6	20	50	100
7	20	50	100
8	20	50	100
9	20	50	100
10	20	50	100

ตารางที่ 6.4 แสดงผลการทดลองการทำงานของชุดตรวจจับหยดน้ำเกลือเมื่อความเข้มแสงเปลี่ยนแปลงไป



รูปที่ 6.6 กราฟแสดงการทำงานของ Sensor หยดน้ำเกลือ เมื่อความเข้มของแสงสว่างเปลี่ยนแปลงไป ผลการทดลอง

ชุดตรวจจับหยดน้ำเกลือสามารถทำงานได้ดีโดยไม่มี Pulse ขาดหายไป แม้ว่าจะปรับความสว่างสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปและวิจารณ์ผลการทดลอง

จากการทดลองจะเห็นว่า การควบคุมอัตราการไหลให้หยดน้ำเกลือแต่ละหยดมีระยะเวลาห่างเท่า ๆ กันนั้น ทำได้ยาก แต่เมื่อเฉลี่ยจำนวนหยดต่อนาทีจะมีค่าผิดพลาดไม่มากนักยกเว้นเมื่ออัตราการไหลมีค่าต่ำ ๆ เช่น น้อยกว่า 5 หยดต่อนาที ในส่วนการทดลองวัดปริมาตรที่ตั้งไว้กับค่าจริงที่ออกมา นั้น มีค่าใกล้เคียงกันไม่เกิน 10 % ทุก ๆ ค่าของอัตราการไหลและทุก ๆ ค่าของปริมาตรที่ตั้งไว้และเมื่ออัตราการไหล มีค่ามากกว่า 250 หยดต่อนาที จะมีโอกาสที่หยดน้ำเกลือจะไหลเป็นสายต่อเนื่องทำให้ไม่สามารถตรวจจับหยดน้ำเกลือได้ ทำให้เครื่องให้น้ำเกลือทำงานผิดพลาด จึงไม่ควรตั้งอัตราการไหลเกินกว่า 250 หยดต่อนาที

และจากการทดลองจะเห็นว่าเครื่องให้น้ำเกลือที่สร้างขึ้นสามารถใช้งานได้ดีในที่มีแสงสว่างมาก ๆ โดยไม่ทำให้จำนวนปริมาณของน้ำเกลือและจำนวนหยดเปลี่ยนแปลงไป

สรุปจากผลการทดลอง เครื่องควบคุมการให้น้ำเกลือสามารถควบคุมปริมาตรที่ออกมาได้ดีในทุก ๆ ค่าและจำนวนหยดเฉลี่ยต่อนาที มีค่าผิดพลาดไม่มากนักในอัตราการไหลสูง ๆ มากกว่า 5 หยดต่อนาที

บทที่ 7

สรุปและข้อเสนอแนะ

โครงการนี้ กลุ่มนักศึกษาได้เน้นที่จะสร้างชุดตรวจจับหยดน้ำ และชุดกลไกการควบคุมอัตราการไหลของน้ำเกลือ รวมทั้งวงจรที่มีประสิทธิภาพสามารถออกแบบสร้างได้ง่ายและมีราคาถูก ซึ่งจากการทดลองที่ผ่านมาในส่วนของวงจรตรวจจับหยดน้ำ มีประสิทธิภาพได้ผลเป็นที่น่าพอใจ แต่ในส่วนของชุดกลไกการควบคุมอัตราการไหล ของน้ำเกลือ ยังมีประสิทธิภาพ และความเป็นเชิงเส้นต่ำ ทำให้ยากแก่การใช้ ไมโครคอนโทรลเลอร์ ในการควบคุมอัตราการไหล ให้แต่ละหยดน้ำมีระยะเวลาห่างหยดเท่า ๆ กัน ดังนั้นทางกลุ่มนักศึกษา จึงได้ออกแบบ ให้ไมโครคอนโทรลเลอร์ ควบคุมจำนวนหยดต่อนาที ให้มีค่าใกล้เคียงกับค่าที่ตั้งไว้มากที่สุด ซึ่งจากการทดลองได้ผลเป็นที่น่าพอใจในอัตราการไหลสูง แต่ที่อัตราการไหลต่ำ เช่น ต่ำกว่า 5 หยดต่อนาที จะมีค่าผิดพลาดสูงมากกว่า 10 % ดังนั้น ถ้าหากมีการพัฒนาและวิจัยกันต่อไป ให้ชุดกลไกมีประสิทธิภาพมากขึ้น ก็จะทำให้เครื่องควบคุมการให้น้ำเกลือมีความสมบูรณ์ จนสามารถผลิตและจำหน่ายได้ในอนาคต

ข้อเสนอแนะเพื่อการพัฒนาต่อไป

7.1 ชุดกลไกการบีบสาย

ในการค้นคว้าและพัฒนาที่ผ่านมา ทำให้ได้เครื่องควบคุมการให้น้ำเกลือที่มีประสิทธิภาพในการใช้งานได้พอสมควรที่ยังไม่เหมาะที่จะนำไปใช้งานกับคนไข้ ซึ่งปัญหาทั้งหมด เกิดจากชุดกลไกการบีบสายน้ำเกลือ ทางผู้จัดทำได้พยายามทำชุดบีบสายหลาย ๆ แบบ แต่ยังไม่เป็นที่น่าพอใจ

ดังได้กล่าวแล้วว่า ชุดกลไกที่มีประสิทธิภาพสูง ในเครื่องมือแบบนี้ คือ กลไกแบบตีนตะขาบ แต่การออกแบบสร้างทำได้ยากและทำให้เครื่องมือมีความซับซ้อนมากขึ้น แต่กลไกการบีบสายเป็นกลไกที่ไม่ซับซ้อนสามารถสร้างได้ง่าย ดังนั้น การพัฒนาต่อไปจึงควรพิจารณาถึงชุดกลไกควบคุมอัตราการไหลเป็นอันดับแรก

7.2 ชุด Power Supply และแบตเตอรี่สำรอง

เครื่องควบคุมการให้น้ำเกลือที่ได้ออกแบบและพัฒนามานั้น จึงไม่สะดวกในการนำไปใช้งานจริง และมีปัญหาเมื่อเกิดไฟฟ้าดับเครื่องจะไม่สามารถทำงานได้ เนื่องจากไม่มีแบตเตอรี่สำรองไฟ ดังนั้นหากมีการพัฒนาและวิจัยต่อไป เครื่องควรมีชุดจ่ายไฟรวมอยู่ในตัวเครื่อง ซึ่งอาจเป็นแบบสวิตซ์ซึ่งเพราะมีขนาดเล็ก รวมทั้งควรมีระบบจ่ายไฟสำรองในกรณีไฟฟ้าดับ ซึ่งจะทำให้เครื่องมีความสมบูรณ์ ในการนำไปใช้งานจริงมากขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

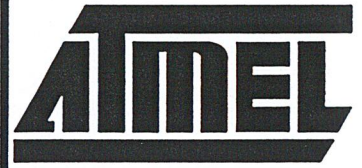
- Compatible with MCS-51™ Products
- 8K bytes of In-System Reprogrammable Downloadable Flash Memory
SPI Serial Interface for Program Downloading
Endurance: 1,000 Write/Erase Cycles
- 2K bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8 bit Internal RAM
- 32 Programmable I/O Lines
- Three 16 bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

Description

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

The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16 bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a



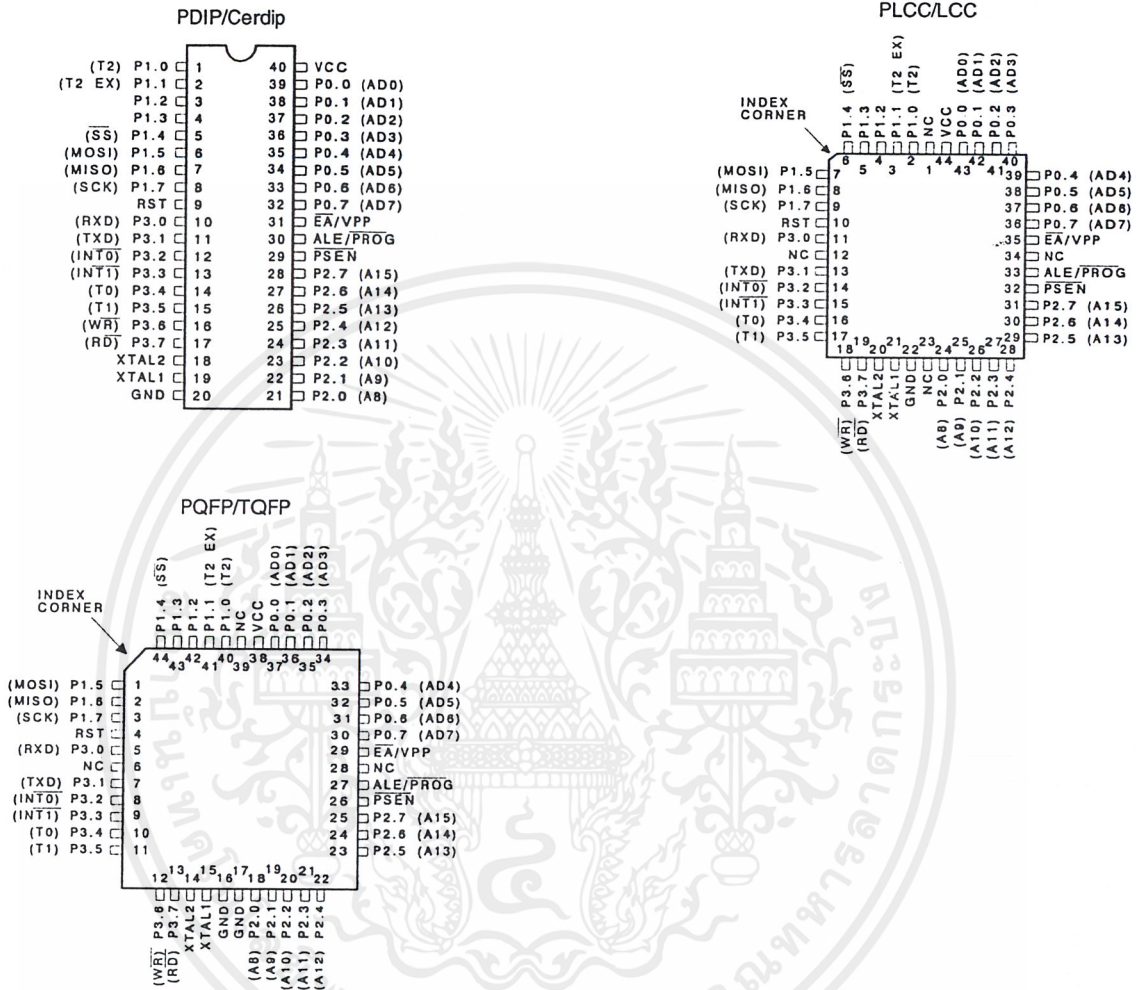
**8 bit
Microcontroller
with 8K bytes
Flash**

**AT89S8252
Preliminary**





Pin Configurations



Pin Description

Vcc

Supply voltage.

GND

Ground.

Port 0

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

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

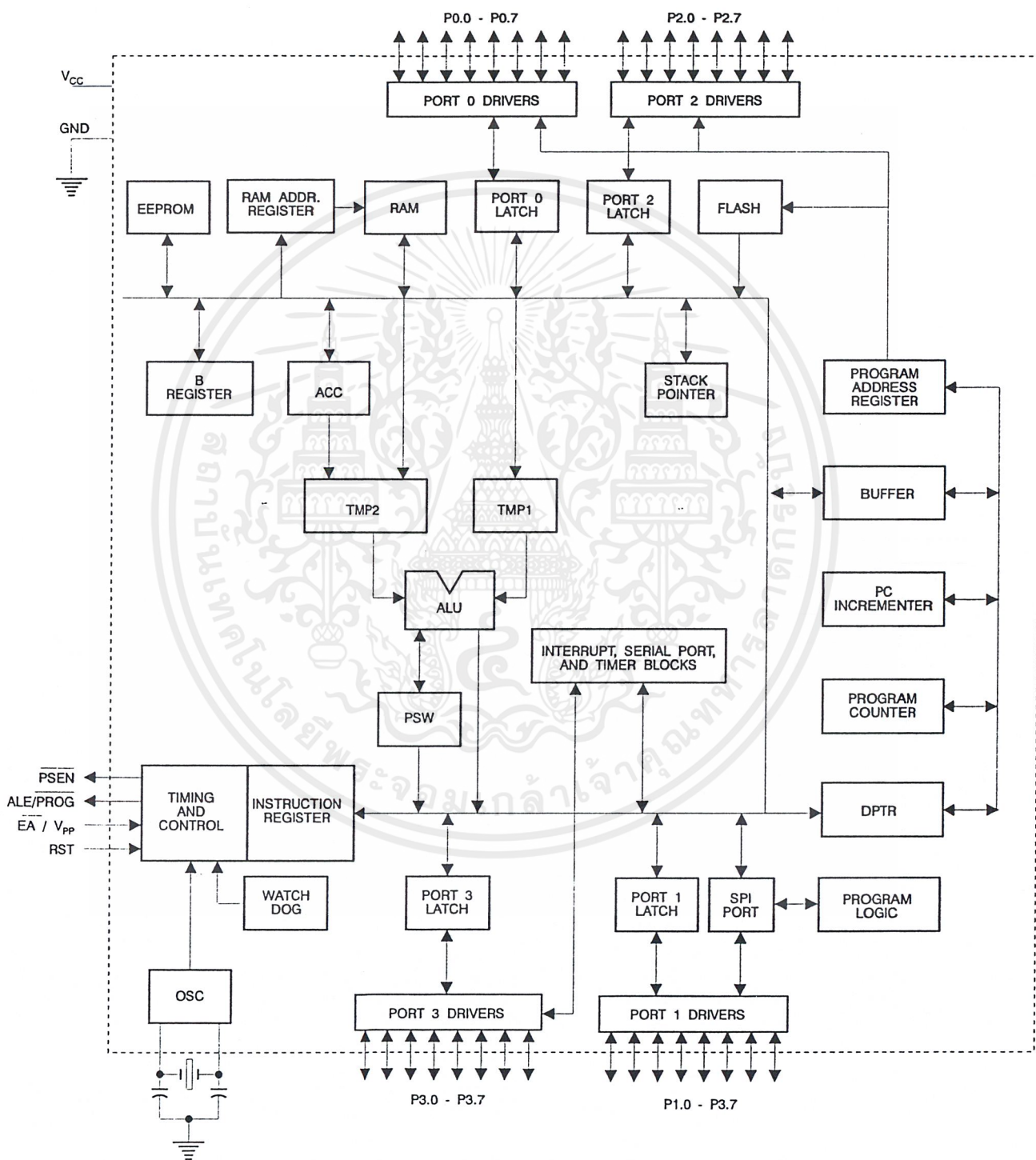
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

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

AT89S8252

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ  เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pin Description (Continued)

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

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

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

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

Port 2

Port 2 is an 8 bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

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

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

Port 3

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

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

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

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

RST

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

ALE/PROG

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

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

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

PSEN

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

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

\overline{EA}/PP

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

\overline{EA} should be strapped to VCC for internal program executions.

(continued)

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ AT89S8252 ใช้เฉพาะเพื่อการศึกษารายละเอียดเท่านั้น ไม่สามารถนำข้อมูลไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

Pin Description (Continued)

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

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

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

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke (continued)

Table 1. AT89S8252 SFR Map and Reset Values

0F8H								0FFH	
0F0H	B 00000000							0F7H	
0E8H								0EFH	
0E0H	ACC 00000000							0E7H	
0D8H								0DFH	
0D0H	PSW 00000000					SPCR 000001XX		0D7H	
0C8H	T2CON 00000000	T2MOD XXXXXX0	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH	
0C0H								0C7H	
0B8H	IP XX000000							0BFH	
0B0H	P3 11111111							0B7H	
0A8H	IE 0X000000		SPSR 00000000					0AFH	
0A0H	P2 11111111							0A7H	
98H	SCON 00000000	SBUF XXXXXXXX						9FH	
90H	P1 11111111						WMCON 00000010	97H	
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH	
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H



Special Function Registers (Continued)

new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16 bit auto-reload mode.

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

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

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

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H		Reset Value = 0000 0000B							
Bit Addressable		TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$
Bit		7	6	5	4	3	2	1	0
Symbol	Function								
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.								
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).								
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.								
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.								
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.								
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.								
$C/\overline{T2}$	Timer or counter select for Timer 2. $C/\overline{T2}$ = 0 for timer function. $C/\overline{T2}$ = 1 for external event counter (falling edge triggered).								
$CP/\overline{RL2}$	Capture/Reload select. $CP/\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. $CP/\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.								

AT89S8252

Table 3. WMCON—Watchdog and Memory Control Register

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

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





Table 4. SPCR—SPI Control Register

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

Symbol	Function															
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.															
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects \overline{SS} , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.															
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.															
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.															
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.															
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.															
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, Fosc., is as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>SPR1</th> <th>SPR0</th> <th>SCK = Fosc. divided by</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>4</td> </tr> <tr> <td>0</td> <td>1</td> <td>16</td> </tr> <tr> <td>1</td> <td>0</td> <td>64</td> </tr> <tr> <td>1</td> <td>1</td> <td>128</td> </tr> </tbody> </table>	SPR1	SPR0	SCK = Fosc. divided by	0	0	4	0	1	16	1	0	64	1	1	128
SPR1	SPR0	SCK = Fosc. divided by														
0	0	4														
0	1	16														
1	0	64														
1	1	128														

Table 5. SPSR—SPI Status Register

SPCR Address = AAH				Reset Value = 0000 0000B				
	SPIF	WCOL	—	—	—	—	—	
Bit	7	6	5	4	3	2	1	0

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

เอกสารนี้เป็นเอกสารของบริษัท ATMEL 8252 ซึ่งรวมเนื้อหาที่อาจมีลิขสิทธิ์ของผู้อื่น ไม่อนุญาตให้แก้ไขได้โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 6. SPDR—SPI Data Register

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

Data Memory—EEPROM and RAM

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

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

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

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

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

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

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

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

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

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

Table 7. Watchdog Timer Period Selection

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

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available





Timer 0 and 1

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

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit $C/\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

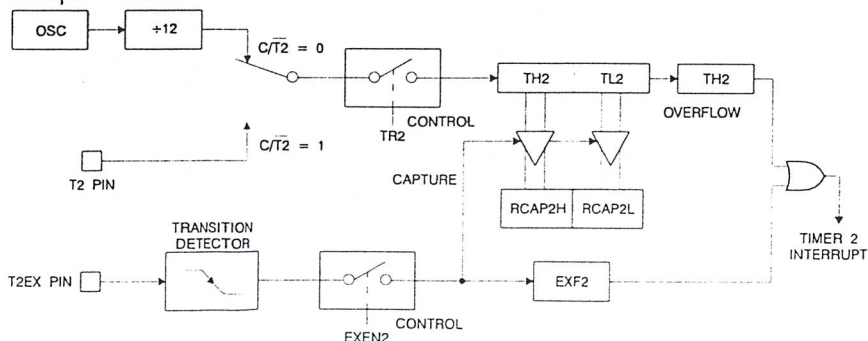
Timer 2 consists of two 8 bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the os-

Table 8. Timer 2 Operating Modes

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

Figure 1. Timer 2 in Capture Mode



Capture Mode

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

Auto-Reload (Up or Down Counter)

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

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

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

(continued)

AT89S8252

Auto-Reload (Up or Down Counter) (Continued)

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

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in

RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

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

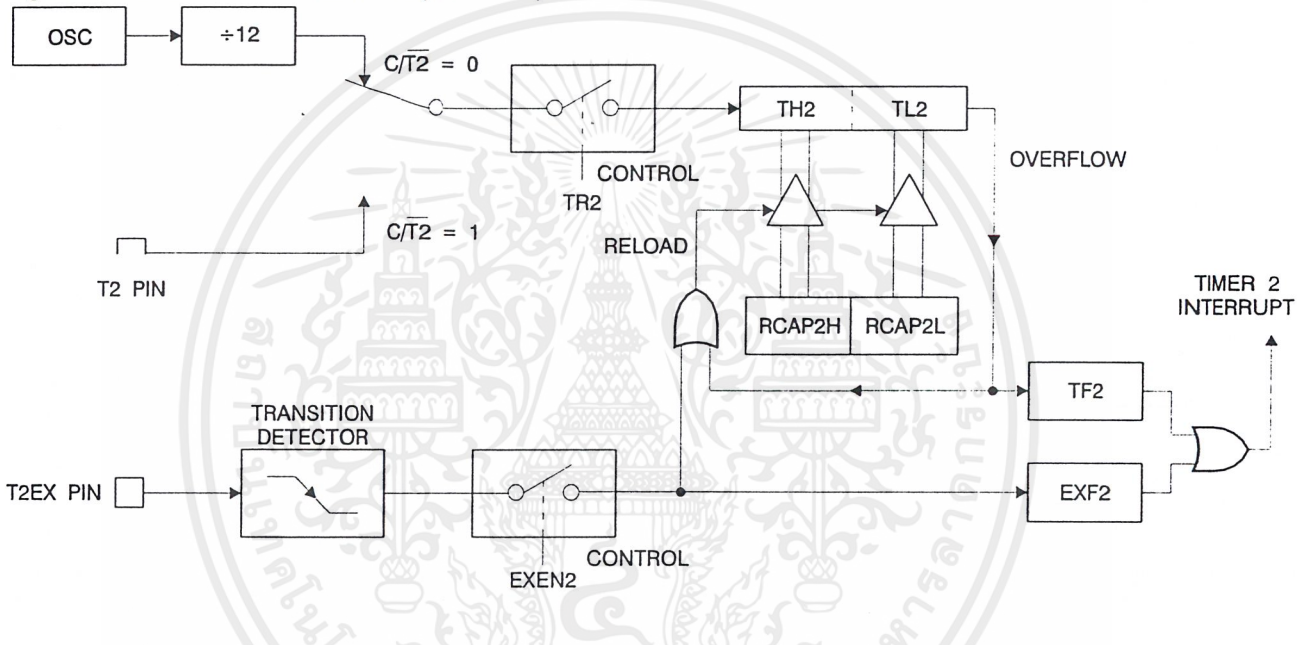


Table 9. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	T2OE	DCEN

Symbol	Function
—	Not implemented, reserved for future use.
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.



Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

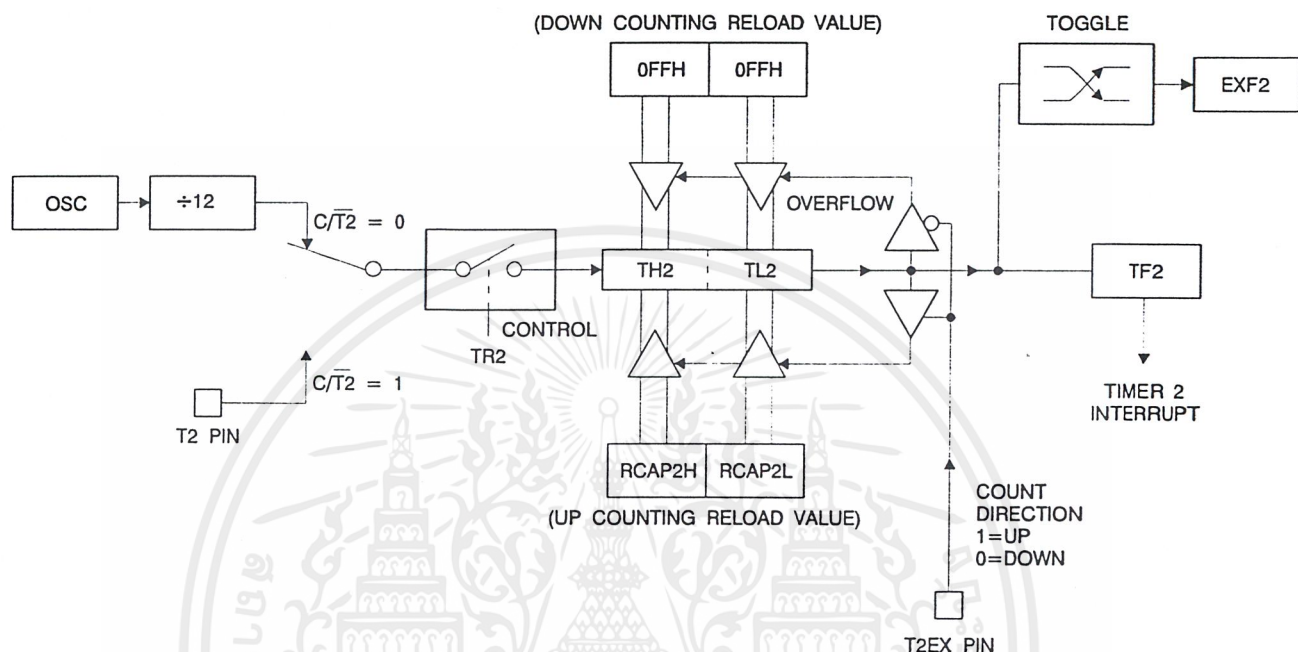
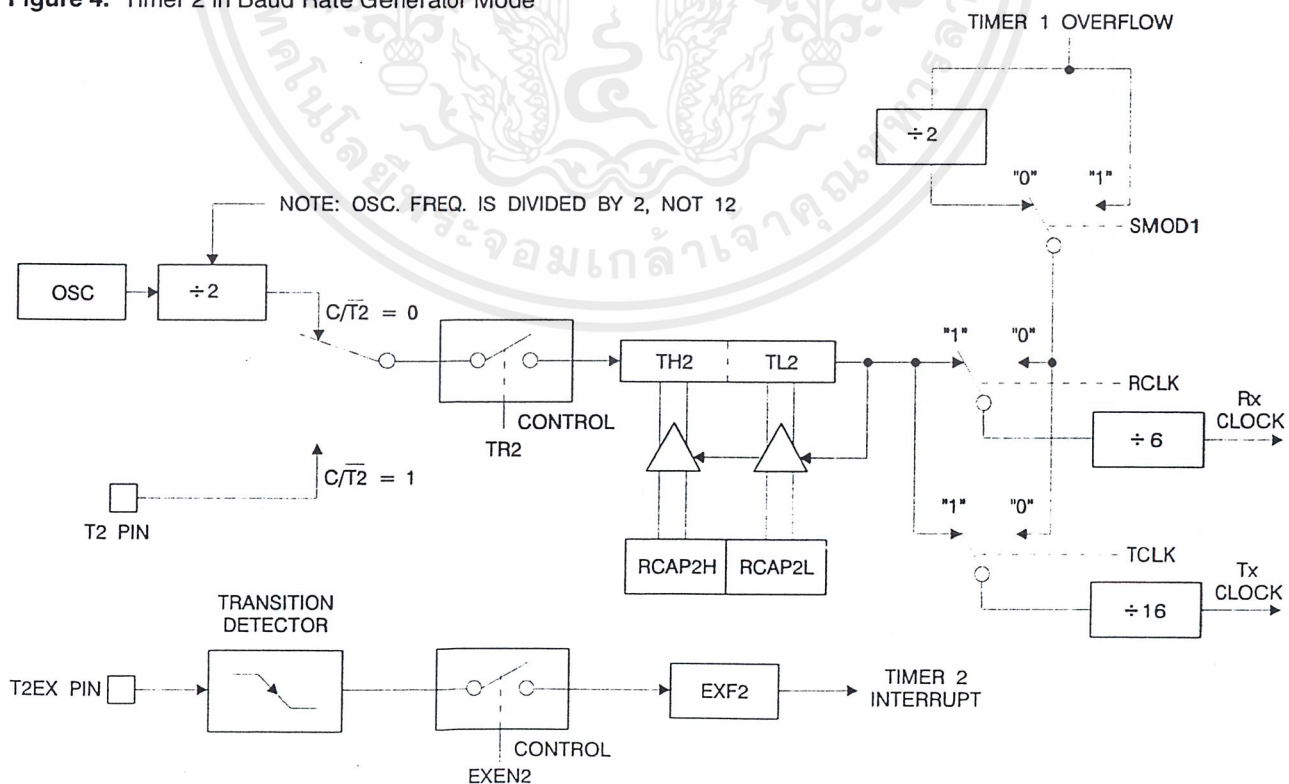


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at

1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

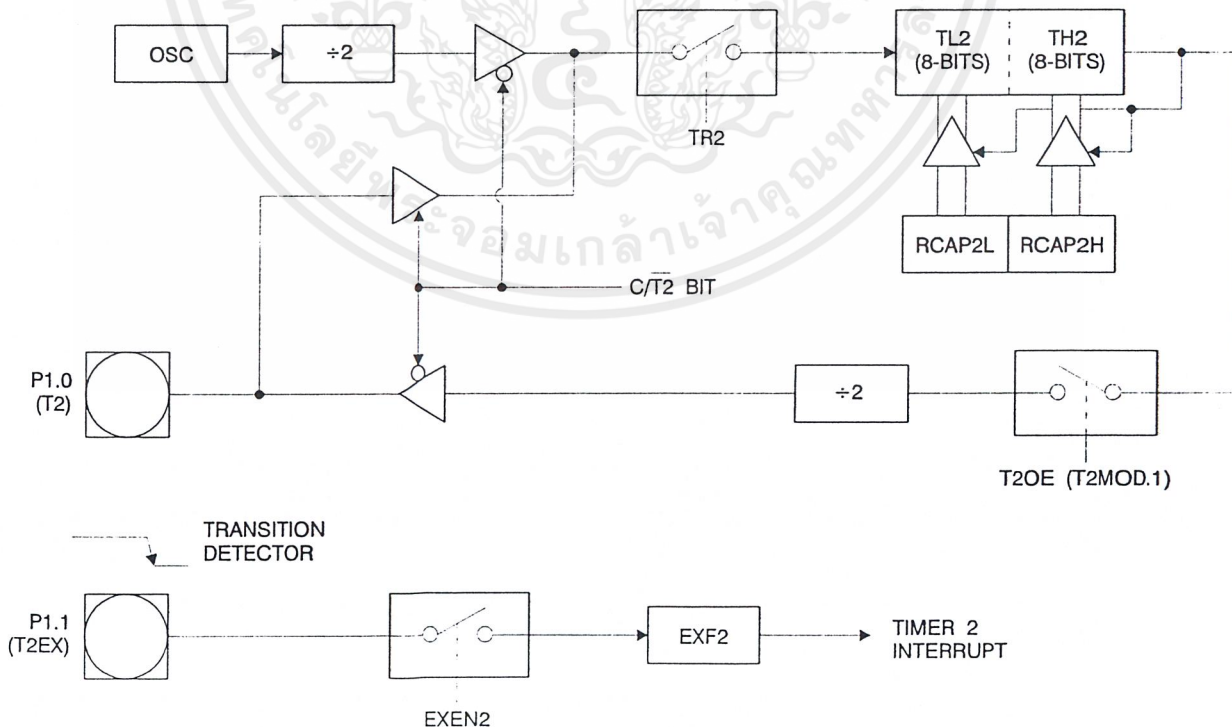
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a l-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might

Figure 5. Timer 2 in Clock-Out Mode





Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out

UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

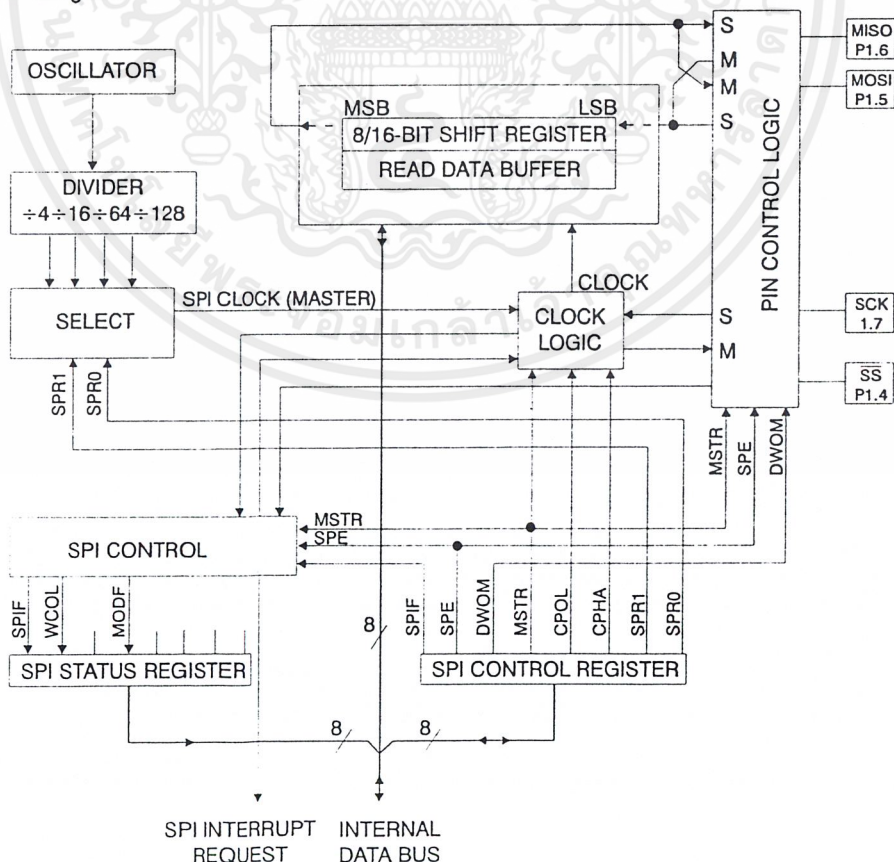
Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 6 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

(continued)

Figure 6. SPI Block Diagram



Serial Peripheral Interface (Continued)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MISO pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}/P1.4$, is set low to select an individual SPI device as a slave. When $\overline{SS}/P1.4$ is set high, the SPI port is deactivated and the MOSI/P1.6 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figures 8 and 9.

Figure 7. SPI Master -Slave Interconnection

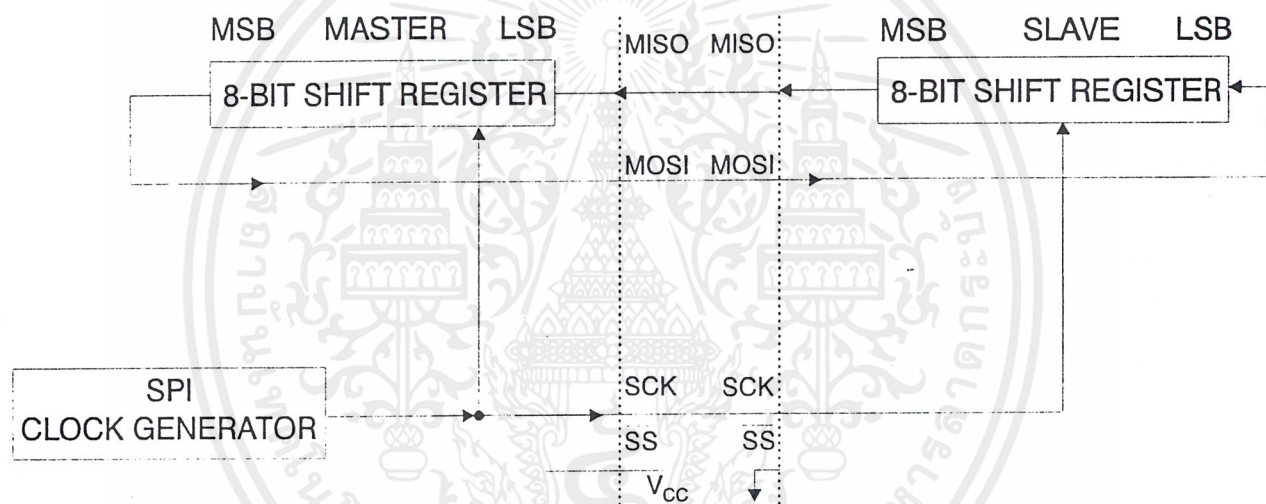
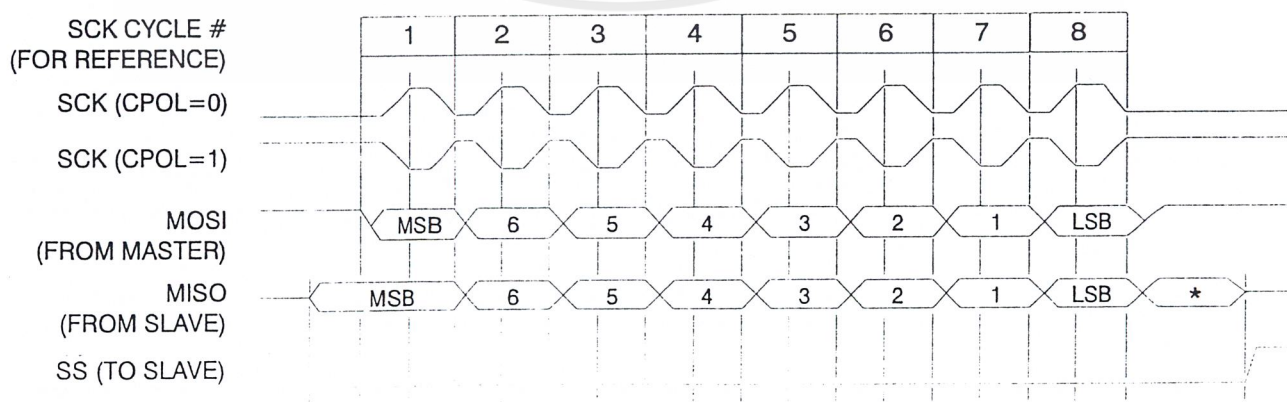


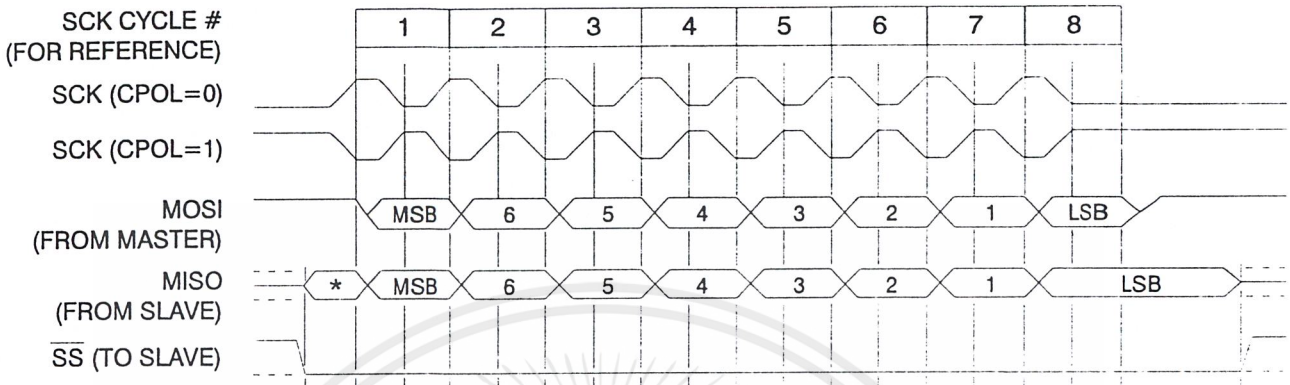
Figure 8. SPI Transfer Format with CPHA = 0



* Not defined but normally MSB of character just received.



Figure 9. SPI Transfer Format with CPHA = 1



* Not defined but normally LSB of previously transmitted character.

Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, AT89C52 and AT89C55, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Table 10. Interrupt Enable (IE) Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables the interrupt.

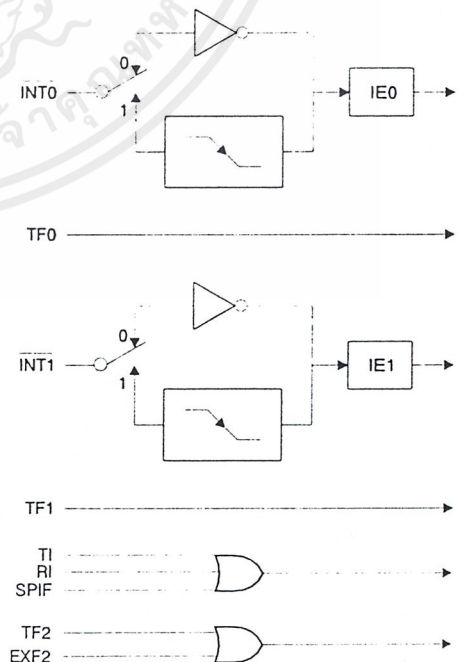
Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit..
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Figure 10. Interrupt Sources



AT89S8252

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

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

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

Power Down Mode

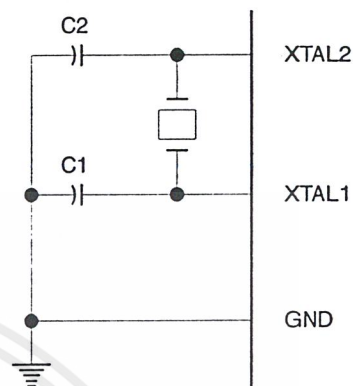
In the power down mode, the oscillator is stopped and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. Exit from power down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power down via an interrupt, the external interrupt must be enabled as level sensitive before entering power

Status of External Pins During Idle and Power Down

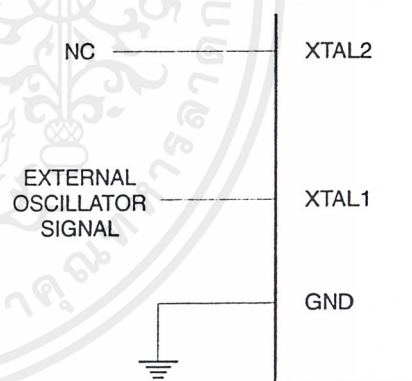
Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Figure 11. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration





down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Lock Bit Protection Modes^(1, 2)

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is

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

Notes: 1. U = Unprogrammed.
2. P = Programmed.

powered up without a reset, the latch initializes to a random value and

holds that value until reset is activated. The latched value of \overline{EA} must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable flash PEROM Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip PEROM Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming modes. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to

reprogram any memory location in the serial programming mode.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

(continued)

AT89S8252

Programming the Flash and EEPROM (Continued)

Parallel Programming Algorithm

To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between V_{CC} and GND pins with all other pins floating.
 - Set RST pin to 'H'.
 - Apply a 4 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Set PSEN pin to 'H'
ALE pin to 'H'
EA pin to 'H' and all other pins to 'H'.
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
Apply data to pins P0.0 to P0.7 for Write Code operation.
5. Raise EA/VPP to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/PROG once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to 'L' and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
 - Set XTAL1 to 'L'.
 - Set RST and EA pins to 'L'.
 - Float all other I/O pins.
 - Turn V_{CC} power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

DATA Polling

The AT89S8252 features DATA Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial

programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. DATA Polling may begin any time after a write cycle has been initiated.

Ready/Busy

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate BUSY. P3.4 is pulled High again when programming is done to indicate READY.

Programming Interface

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

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFFH for Code memory and 0000H to 7FFFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 KHz. *(continued)*



Serial Downloading (Continued)

Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:

Apply power between V_{CC} and GND pins with all other pins floating.

Set RST pin to 'H'.

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 4 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.

Instruction Set

Instruction	Input Format		Operation
	MSB	LSB	
Programming Enable	1010 0101 xxxx	1100 0011 xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 xxxx xxxx	1100 x100 xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa low xxxx	a001 addr xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa low data	a010 addr in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa low xxxx	a101 addr xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa low data	a110 addr in	Write data to Data memory location at selected address.
Write Lock Bits	1010 LLLx BBB 123 xxxx	1100 x111 xxxx	Write lock bits. Set LB1, LB2 or LB3 = '0' to program lock bits.






Notes: 1. $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle, which typically takes less than 2.5 ms.
2. 'aaaaa' = high order address.
3. 'x' = don't care.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

AT89S8252

Flash and EEPROM Parallel Programming Modes

Mode	RST	$\overline{\text{PSEN}}$	$\overline{\text{ALE/PROG}}$	$\overline{\text{EA/VPP}}$	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0	
Serial Prog. Modes	H	h ⁽¹⁾	h	x	(See detail timing)						
Chip Erase	H	L	 ⁽²⁾	12V	H	L	L	L	X	X	
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR	
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR	
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X	
Bit - 1									P0.7 = 0	X	
Bit - 2									P0.6 = 0	X	
Bit - 3									P0.5 = 0	X	
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X	
Bit - 1									@P0.2	X	
Bit - 2									@P0.1	X	
Bit - 3									@P0.0	X	
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H	
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H	
Serial Prog. Enable	H	L	 ⁽²⁾	12V	L	H	L	H	P0.0 = 0	X	
Serial Prog. Disable	H	L	 ⁽²⁾	12V	L	H	L	H	P0.0 = 1	X	
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X	

- Notes: 1. 'h' = weakly pulled 'High' internally. 3. P3.4 is pulled Low during programming to indicate RDY/BSY.
 2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH. 4. 'X' = don't care.



Figure 13. Programming the Flash Memory

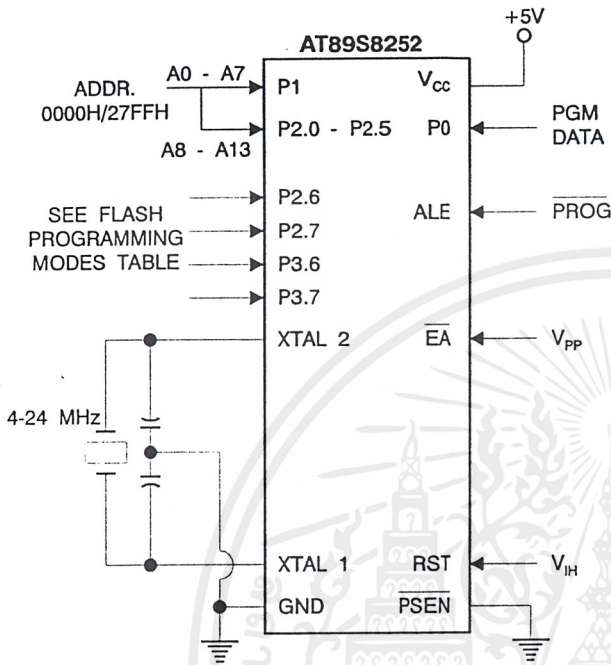


Figure 15. Flash/EEPROM Serial Downloading

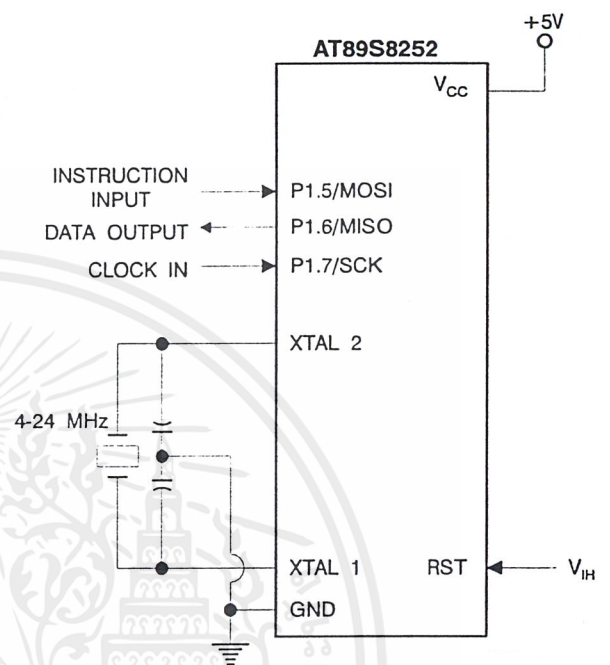
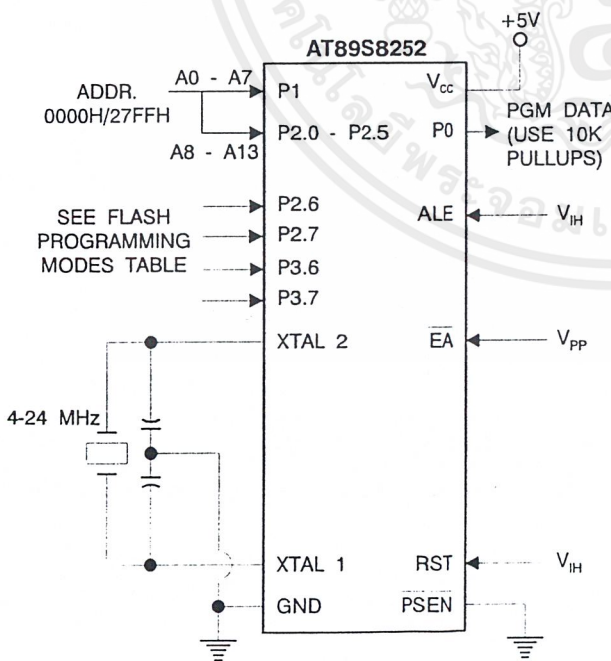


Figure 14. Verifying the Flash Memory



AT89S8252

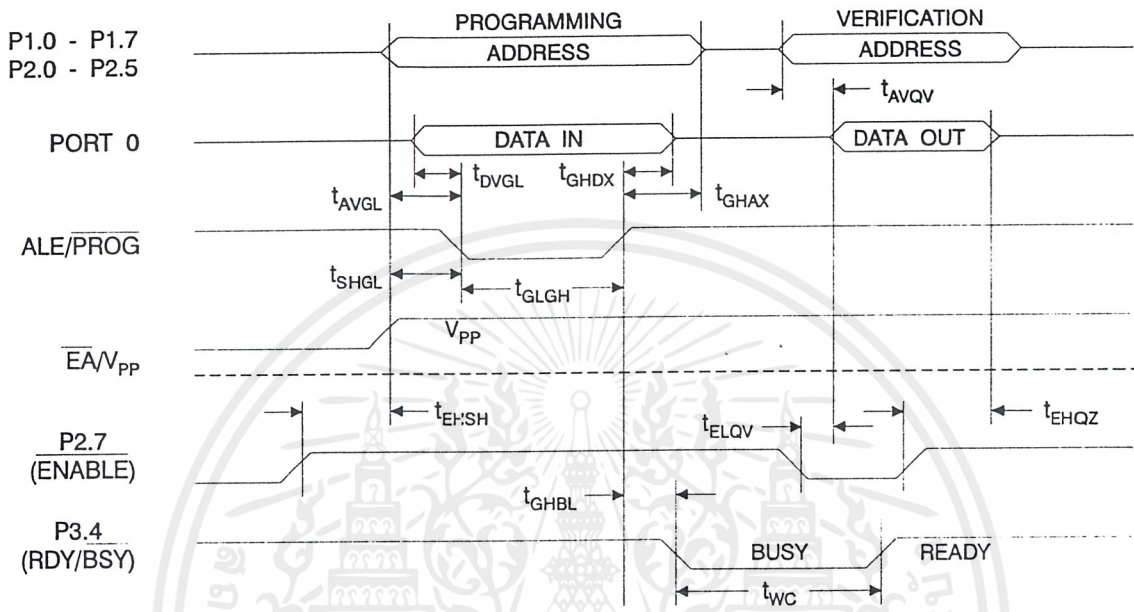
Flash Programming and Verification Characteristics

$T_A = 21^\circ\text{C}$ to 27°C , $V_{CC} = 5.0\text{V} \pm 10\%$

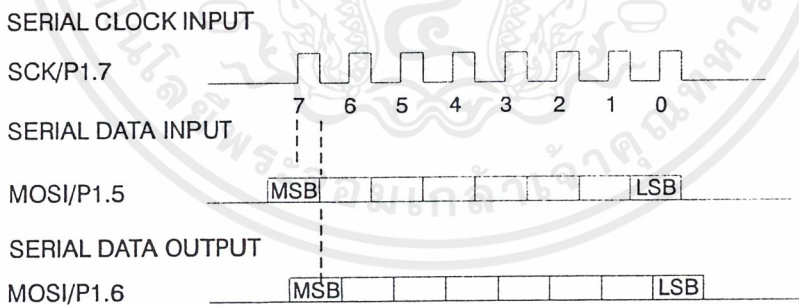
Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{ESHSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQV}	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to BUSY Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms



Flash/EEPROM Programming and Verification Waveforms - Parallel Mode



Serial Downloading Waveforms



AT89S8252

Absolute Maximum Ratings*

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

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

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 2.7\text{V}$ to 6.0V , unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage ($\overline{\text{EA}}$)		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.5	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$		-650	μA
I_{LI}	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pulldown Resistor		50	300	$\text{k}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 10 mA
 Maximum I_{OL} per 8 bit port:
 Port 0: 26 mA
 Ports 1,2,3: 15 mA
 Maximum total I_{OL} for all output pins: 71 mA

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

2. Minimum V_{CC} for Power Down is 2V.



AC Characteristics

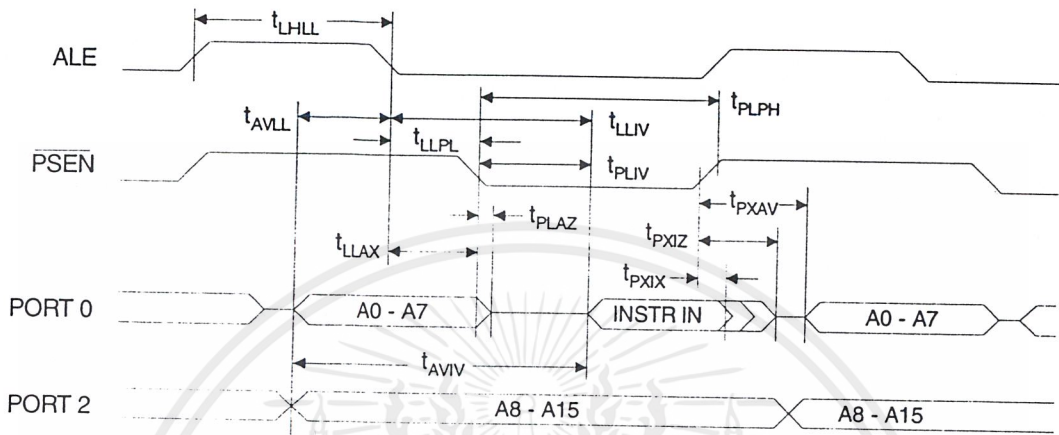
Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

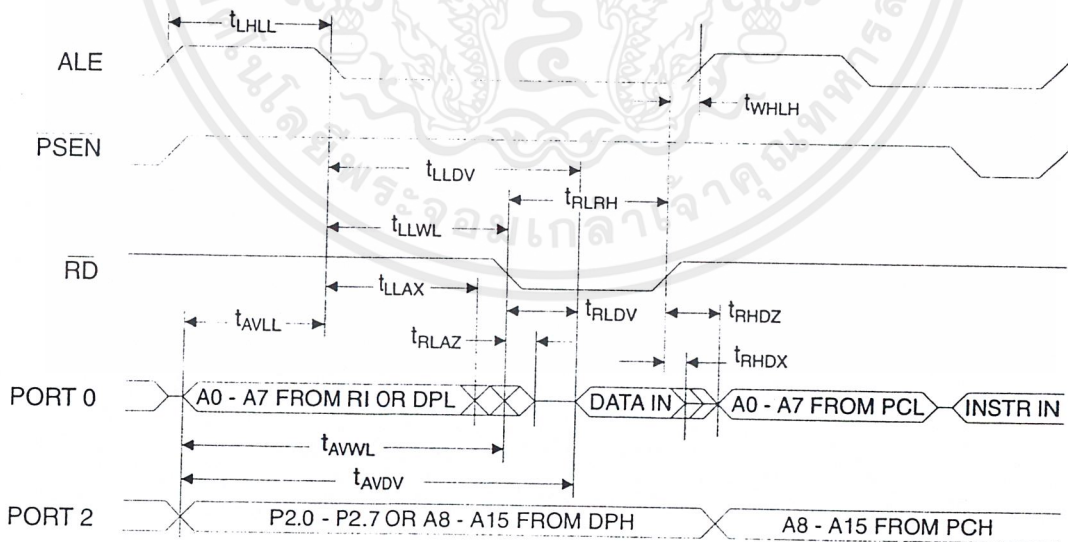
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/tCLCL	Oscillator Frequency			0	24	MHz
tLHLL	ALE Pulse Width	127		2tCLCL - 40		ns
tAVLL	Address Valid to ALE Low	28		tCLCL - 13		ns
tLLAX	Address Hold After ALE Low	48		tCLCL - 20		ns
tLLIV	ALE Low to Valid Instruction In		233		4tCLCL - 65	ns
tLLPL	ALE Low to $\overline{\text{PSEN}}$ Low	43		tCLCL - 13		ns
tPLPH	$\overline{\text{PSEN}}$ Pulse Width	205		3tCLCL - 20		ns
tPLIV	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		3tCLCL - 45	ns
tPXIX	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
tPXIZ	Input Instruction Float After $\overline{\text{PSEN}}$		59		tCLCL - 10	ns
tPXAV	$\overline{\text{PSEN}}$ to Address Valid	75		tCLCL - 8		ns
tAVIV	Address to Valid Instruction In		312		5tCLCL - 55	ns
tPLAZ	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
tRLRH	$\overline{\text{RD}}$ Pulse Width	400		6tCLCL - 100		ns
tWLWH	$\overline{\text{WR}}$ Pulse Width	400		6tCLCL - 100		ns
tRLDV	$\overline{\text{RD}}$ Low to Valid Data In		252		5tCLCL - 90	ns
tRHDX	Data Hold After $\overline{\text{RD}}$	0		0		ns
tRHDX	Data Float After $\overline{\text{RD}}$		97		2tCLCL - 28	ns
tLLDV	ALE Low to Valid Data In		517		8tCLCL - 150	ns
tAVDV	Address to Valid Data In		585		9tCLCL - 165	ns
tLLWL	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	3tCLCL - 50	3tCLCL + 50	ns
tAVWL	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		4tCLCL - 75		ns
tQVWX	Data Valid to $\overline{\text{WR}}$ Transition	23		tCLCL - 20		ns
tQVWH	Data Valid to $\overline{\text{WR}}$ High	433		7tCLCL - 120		ns
tWHQX	Data Hold After $\overline{\text{WR}}$	33		tCLCL - 20		ns
tRLAZ	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
tWHLH	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	tCLCL - 20	tCLCL + 25	ns

AT89S8252

External Program Memory Read Cycle

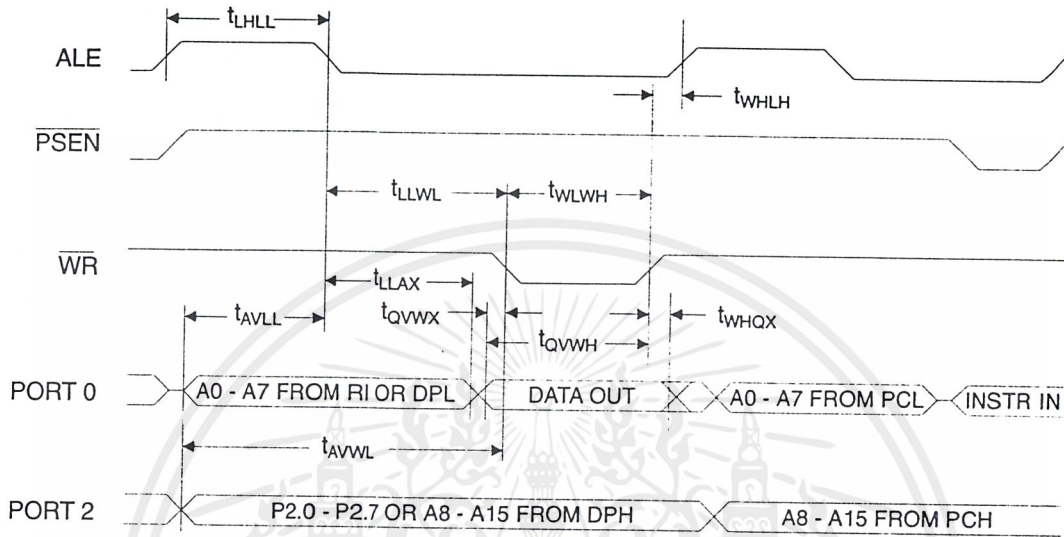


External Data Memory Read Cycle

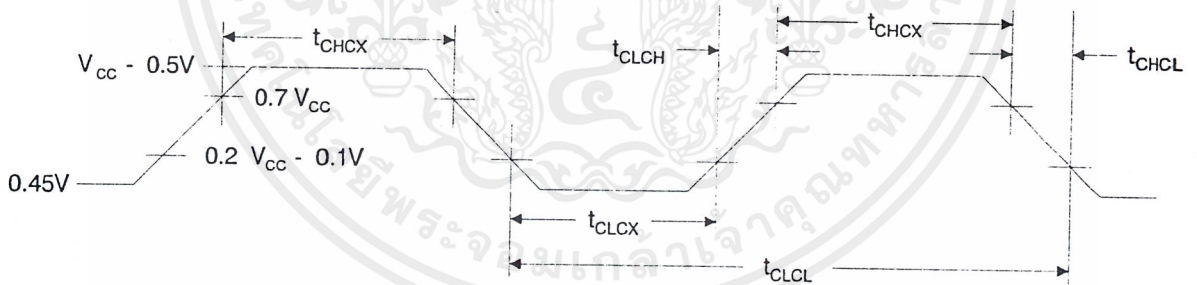




External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	$V_{CC} = 4.0V$ to $6.0V$		Units	$V_{CC} = 2.7V$ to $6.0V$	
		Min	Max		Min	Max
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz	0	12
t_{CLCL}	Clock Period	41.6		ns	83.3	
t_{CHCX}	High Time	15		ns	30	
t_{CLCX}	Low Time	15		ns	30	
t_{CLCH}	Rise Time		20	ns		20

เอกสารนี้เป็นเอกสาร AT89S8252... ไม่สงวนลิขสิทธิ์...
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

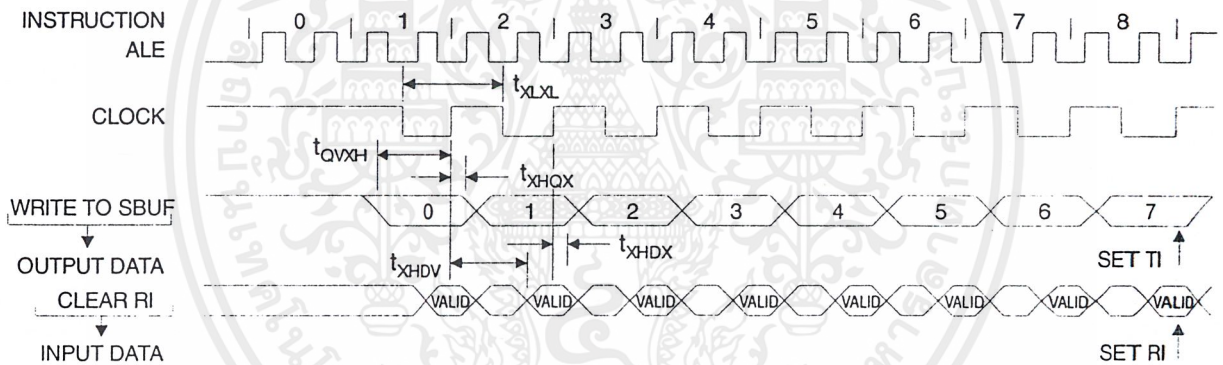
AT89S8252

Serial Port Timing: Shift Register Mode Test Conditions

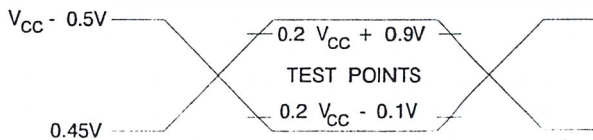
The values in this table are valid for $V_{CC} = 2.7V$ to $6V$ and Load Capacitance = 80 pF .

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL} - 133$		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL} - 33$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL} - 133$	ns

Shift Register Mode Timing Waveforms

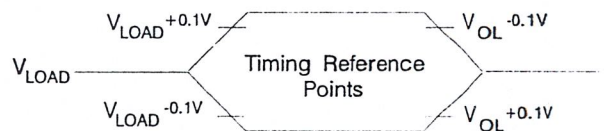


AC Testing Input/Output Waveforms ⁽¹⁾



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

Float Waveforms ⁽¹⁾

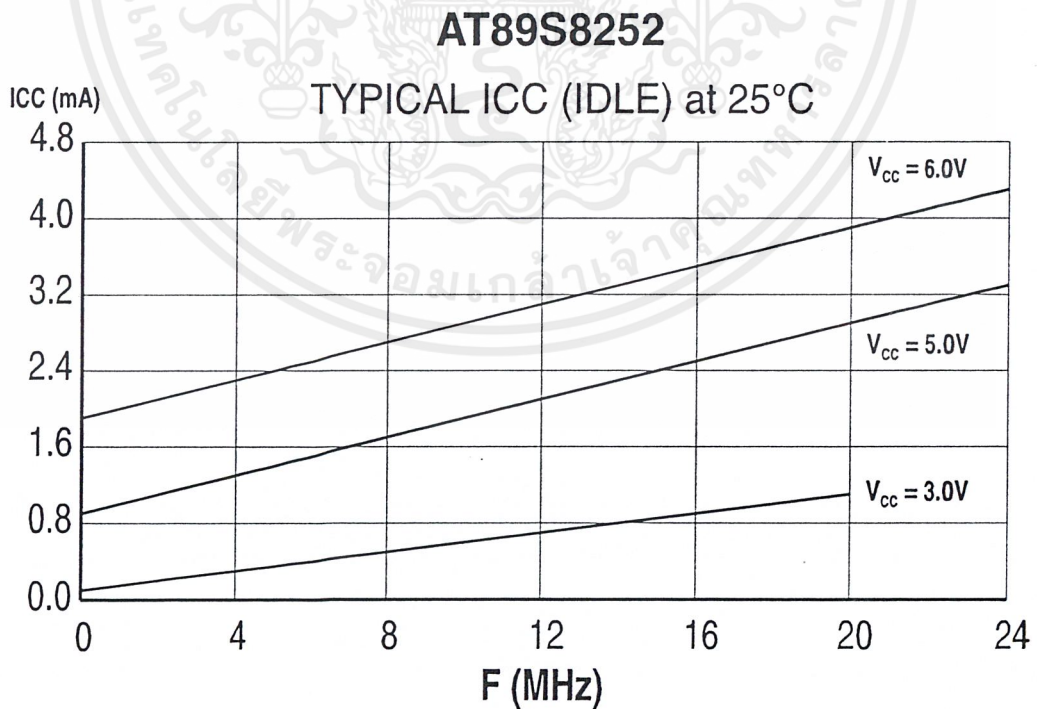
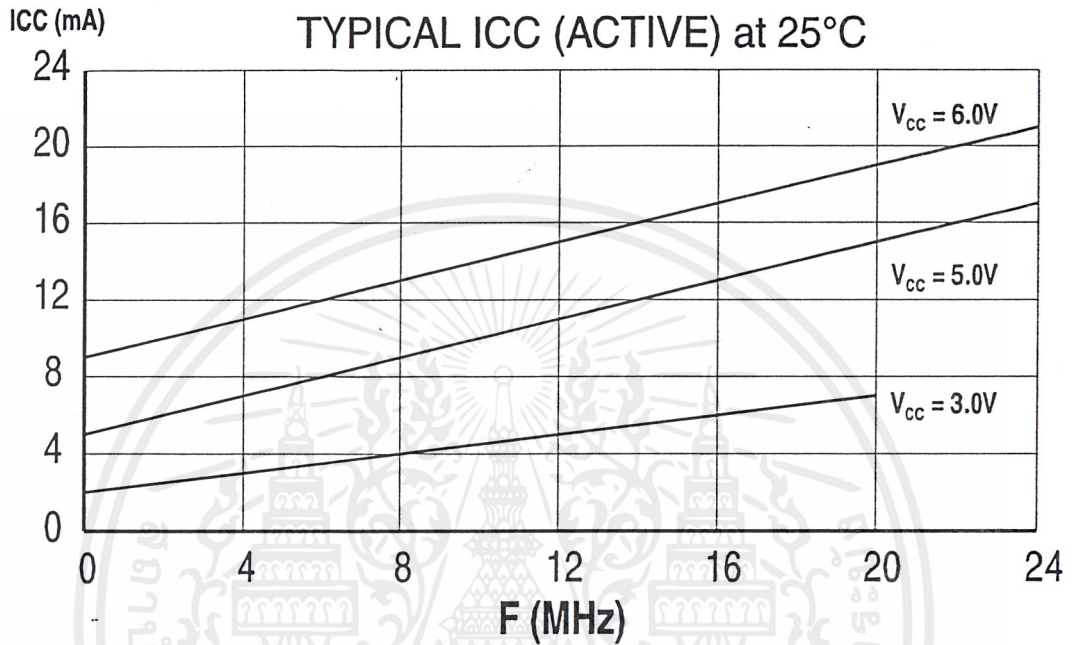


Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.





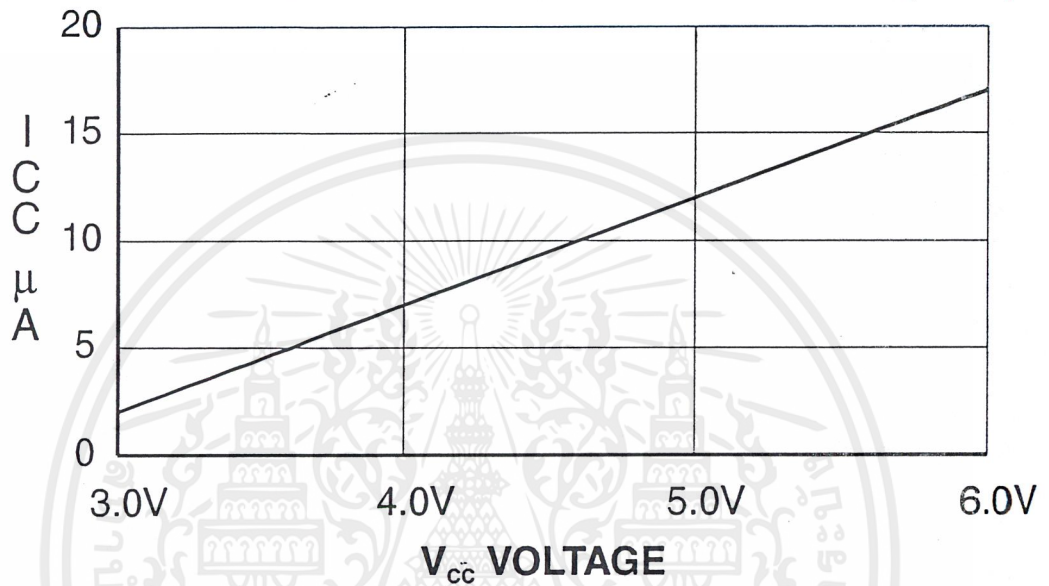
AT89S8252



AT89S8252

AT89S8252

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)





Ordering Information

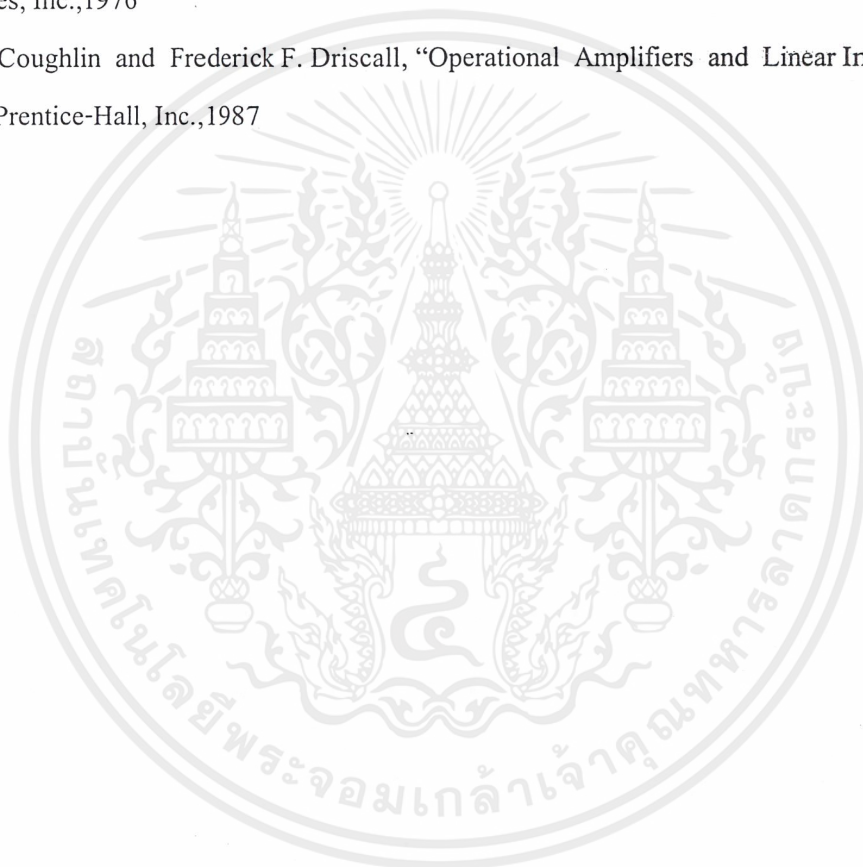
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89S852-12AC AT89S852-12JC AT89S852-12PC AT89S852-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S852-12AI AT89S852-12JI AT89S852-12PI AT89S852-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
16	4.0V to 6.0V	AT89S852-16AA AT89S852-16JA AT89S852-16PA AT89S852-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	4.0V to 6.0V	AT89S852-20AC AT89S852-20JC AT89S852-20PC AT89S852-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S852-20AI AT89S852-20JI AT89S852-20PI AT89S852-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	4.0V to 6.0V	AT89S852-24AC AT89S852-24JC AT89S852-24PC AT89S852-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89S852-24AI AT89S852-24JI AT89S852-24PI AT89S852-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
AT89S8252
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. สุปราณี วศินอมร, “การพยาบาลพื้นฐาน”, จุดทองการพิมพ์, กรุงเทพฯ, 2536.
2. อุดม จีนประดับ , “ไมโครคอนโทรลเลอร์” , MCS-S1, สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ , 2541
3. GOBIND DARYANANI, “Principles of active Network Synthesis and Design” , Bell Laboratories, Inc., 1976
4. Robert F. Coughlin and Frederick F. Driscall, “Operational Amplifiers and Linear Integrated Circuits”, Prentice-Hall, Inc., 1987



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้