

เครื่องับจำนวนหยคน้ำเกลือ



นางสาวนาริมล ทองศรีคำ
นางสาวยุพวรรณ ชำรงศิลป์



T031643

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาคามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาฟิสิกส์ประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

รับ,
๑๖ ๕๕๖๓
๕๕๕๐

เลขหมึ.....

เลขทะเบียน...31643...

วัน, เดือน, ปี 19 พ.ค. 2541

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRIP WATCHER

MISS NAREEMOL THONGSRIKUM
MISS YUPAWAN THUMRONGSILAPA

**A Special Project Submitted in Fulfillment of the Requirement for
the Degree of Bachelor of Science
Department of Applied Physics
Faculty of Science**

King Mongkut's Institute of Technology Ladkrabang

1997

หัวข้อโครงการพิเศษ

เครื่องนับจำนวนหยดน้ำเกลือ

โดย

นางสาวนารีมล ทองศรีคำ

นางสาวบุษพรพรรณ ชำรงศิลป์

ภาควิชา

ฟิสิกส์ประยุกต์

อาจารย์ที่ปรึกษา

ดร.รัชภักย์ จิตต์อารี

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้า
คุณทหารลาดกระบัง อนุมัติให้นับโครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลัก
สูตรวิทยาศาสตร์บัณฑิต



(รศ. สุรพล รักวิชัย)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

คณะกรรมการโครงการพิเศษ

ดร.รัชภักย์ จิตต์อารี

(ดร. รัชภักย์ จิตต์อารี)

ประธานกรรมการ



(ผศ. อนุพงศ์ ทรงประภา)

กรรมการ

(รศ. สุวรรณ คู่อาราย)

กรรมการ



(อ. วิษณุ เพชรภา)

กรรมการ

หัวข้อโครงการพิเศษ	เครื่องนับจำนวนหยดน้ำเกลือ
นักศึกษา	นางสาวนาวิมล ทองศรีคำ นางสาวยุพวรรณ ชำรงศิลป์
อาจารย์ที่ปรึกษา	ดร. รัชภักย์ จิตต์อารี
ภาควิชา	ฟิสิกส์ประยุกต์
ปีการศึกษา	2540

บทคัดย่อ

เครื่องนับจำนวนหยดน้ำเกลือเป็นเครื่องอำนวยความสะดวกในทางการแพทย์ และสาขาที่เกี่ยวข้องทั่วไป กล่าวคือ สามารถนับจำนวนหยดน้ำเกลือเป็นอัตราส่วนจำนวนหยดต่อ 1 นาทีและยังสามารถส่งเสียงเตือน เมื่อน้ำเกลือใกล้หมดขวดได้ พื้นฐานการทำงานโดยทั่วไปของเครื่องนี้จะใช้ “ Infrared Sensor ” เป็นตัวตรวจจับหยดน้ำเกลือทุกๆ หยด และใช้ไมโครคอนโทรลเลอร์ 89C2051 เป็นตัวประมวลผล โดยป้อนคำสั่งทางโปรแกรมภาษาแอสเซมบลี ควบคุมการทำงานของเครื่องให้เป็นไปตามที่ต้องการ และตรงตามวัตถุประสงค์

Special Project Title	Drip Watcher
Name	Miss Nareemol Thongsrikum Miss Yupawan Thumrongsilapa
Special Project Advisor	Dr. Ratchapak Chitaree
Department	Applied Physics
Academic Year	1997

Abstract

A drip watcher is one of useful medical equipment widely used in hospitals. Its basic functions are to count saline drops and give a warning signal to a nearby nurse when the solution bottle is nearly empty. The fundamental operation of the drip watcher built in this project relies on an infrared sensor head which provides signals for a further processing by a Microcontroller 89C2051.

กิติกรรมประกาศ

ขอขอบพระคุณอาจารย์ทุกท่านที่ได้ให้คำแนะนำต่างๆ ในการทำโครงการพิเศษ
นี้ และขอขอบคุณเป็นพิเศษสำหรับ

บิดามารดาที่ให้การสนับสนุนในทุกเรื่องที่ทำให้โครงการพิเศษนี้สำเร็จลงไปได้ด้วยดี

ดร. รัชภักย์ จิตต์อารี ที่ให้คำปรึกษาในการทำโครงการพิเศษนี้

ขอขอบพระคุณแพทย์หญิงมินะ ตระกูลสุนและแพทย์หญิงจิราภา ที่ให้ข้อมูล
เรื่องเข็มน้ำเกลือ

โรงพยาบาลภูมิพลฯ และ โรงพยาบาลลาดกระบังตลอดจนเจ้าหน้าที่ที่เกี่ยวข้องที่
ให้ข้อมูลและสาริตเครื่องนับจำนวนหยดน้ำเกลือ

คุณอนันตชัย นัยจิตรและคุณกัญชชด ที่ให้คำแนะนำด้านซอฟต์แวร์

นารีมล ทองศรีคำ
ยุพวรรณ ชำรงศิลป์

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการเบื้องต้นและอุปกรณ์	2
	■ น้ำเกลือ	2
	■ ชนิดของเข็มน้ำเกลือและการใช้งาน	8
	■ ชนิดของภาชนะบรรจุน้ำเกลือและการใช้งาน	14
	■ พิสิกส์ของน้ำเกลือ	15
	■ อินฟราเรด	15
	■ ออปโตอิเล็กทรอนิกส์	16
	■ การเชื่อมต่อกับ LED 7 ส่วน	18
บทที่ 3	การวิจัยและการดำเนินงาน	26
	■ ศึกษาค้นคว้า	26
	■ การทดลอง	27
	■ ประมวลผลและแสดงผล	30
	■ การประยุกต์ใช้	31
บทที่ 4	ผลการศึกษาจากการดำเนินงาน	33
	■ เครื่องนับจำนวนหยดน้ำเกลือที่สร้างขึ้น	33
	■ ผลการทดลอง	36
บทที่ 5	สรุปผลการวิจัย	38
	ภาคผนวก	
	เอกสารอ้างอิง	
	บรรณานุกรม	

สารบัญรูป

		หน้า
รูปที่ 2.1	แสดงเครื่องนับจำนวนหยดน้ำเกลือแบบที่ 1	5
รูปที่ 2.2	แสดงเครื่องนับจำนวนหยดน้ำเกลือแบบที่ 2	6
รูปที่ 2.3	แสดงชุดให้อาหารทางหลอดเลือดดำ	7
รูปที่ 2.4	แสดงชุดเข็มน้ำเกลือขนาดมาตรฐาน	9
รูปที่ 2.5	แสดงชุดเข็มน้ำเกลือชนิดไมโครครอป	10
รูปที่ 2.6	แสดงชุดเข็มน้ำเกลือพร้อมเข็มฉีดยา ขนาด 100 มิลลิลิตร	12
รูปที่ 2.7	สเปกตรัมของคลื่นแม่เหล็กไฟฟ้าและแสดงย่านการใช้งาน ออพโตอิเล็กทรอนิกส์	16
รูปที่ 2.8	วงจรใช้งาน LED	17
รูปที่ 2.9	เมื่อใช้ LED กับแรงดันไฟสลับ	18
รูปที่ 2.10	แสดงสัญลักษณ์แต่ละเซกเมนต์	19
รูปที่ 2.11	แสดงการต่อ LED 7 ส่วนเข้ากับ MCS-51	21
รูปที่ 2.12	แสดงการต่อ LED 7 ส่วน 4 ตัว	23
รูปที่ 3.1	แสดงแผนภาพการดำเนินการ	26
รูปที่ 3.2	แสดงวงจรตัวตรวจจับทางแสง(Sensor)	27
รูปที่ 3.3	แสดงวงจรส่วนประมวลผล	28
รูปที่ 3.4	แสดงผลของวงจรทั้งหมด	29
รูปที่ 3.5	แสดงวงจรขับลำโพง	30
รูปที่ 4.1ก, 4.1ข	เครื่องนับจำนวนหยดน้ำเกลือ	33
รูปที่ 4.2	แสดงชุดของเข็มน้ำเกลือ	35

สารบัญตาราง

		หน้า
ตารางที่ 2-1	แสดงตำแหน่งขาที่จะต่อกับพอร์ต	19
ตารางที่ 2-2	แสดงข้อมูลที่ส่งให้ LED 7 ส่วนแสดงเป็นเลขต่างๆ	20
ตารางที่ 4-1	แสดงผลการทดลอง	36



บทที่ 1

บทนำ

เครื่องนับจำนวนหยดน้ำเกลือ

ที่มาของโครงการพิเศษ

เป็นที่ทราบกันโดยทั่วไปว่า “น้ำเกลือ” นำมาใช้กันอย่างกว้างขวางในทางการแพทย์การสาธารณสุข และมีความสำคัญต่อการบำบัดรักษาผู้ป่วยเป็นอย่างมาก การปฏิบัติงานของแพทย์พยาบาล เจ้าหน้าที่อนามัยต่อผู้ป่วย จำเป็นอย่างยิ่งที่จะต้องควบคุมปริมาณน้ำเกลือให้เหมาะสมกับความต้องการของร่างกายของผู้ป่วยในกรณีแตกต่างกันไป ซึ่งการควบคุมดังกล่าวนี้ มีหลักเกณฑ์การวัดหลายรูปแบบ เช่น จำนวนหยดต่อหนึ่งหน่วยเวลา ปริมาณต่อหนึ่งหน่วยเวลา เป็นต้น และเพื่อความถูกต้องแม่นยำในการควบคุมปริมาณน้ำเกลือ จึงมีผู้คิดเครื่องนับจำนวนหยดน้ำเกลือขึ้น ซึ่งจะเห็นได้ว่าปัจจุบันนี้ตามโรงพยาบาลต่างๆ นิยมใช้เครื่องมือชนิดนี้ แต่เป็นลักษณะของการนำเข้าจากต่างประเทศและมีราคาสูง ดังนั้นเพื่อเป็นการสร้างอุปกรณ์ขึ้นใช้ได้เอง อีกทั้งยังเป็นการประหยัดและลดการสั่งซื้อจากต่างประเทศ จึงได้ริเริ่มโครงการพิเศษนี้ขึ้นมา

วัตถุประสงค์ของโครงการพิเศษ

1. อำนวยความสะดวกให้กับแพทย์ พยาบาล ตลอดจนเจ้าหน้าที่อนามัยต่างๆ ให้มากที่สุด
2. เพื่อประดิษฐ์เครื่องนับจำนวนหยดน้ำเกลือขึ้นเอง โดยมีคุณภาพเชื่อถือได้
3. ประยุกต์ไมโครคอนโทรลเลอร์ใช้กับตัวตรวจจับทางแสง (infrared sensor) ให้มีประสิทธิภาพมากที่สุด

ประโยชน์ที่ได้รับจากการทำโครงการพิเศษ

1. เข้าใจการทำงานของเครื่องนับจำนวนหยดน้ำเกลือ ตลอดจนหาวิธีแก้ไขจุดบกพร่องต่างๆ ที่เกิดขึ้นแล้วพัฒนาให้มีประสิทธิภาพดียิ่งขึ้นต่อไป
2. ทำให้ทราบถึงการประยุกต์ใช้ไมโครคอนโทรลเลอร์ให้เป็นประโยชน์ในชีวิตประจำวัน
3. ได้ประสบการณ์ในการค้นคว้าและการปฏิบัติงานจริง

ขอบเขตของการทำโครงการพิเศษ

1. สามารถนับปริมาณน้ำเกลือเป็นจำนวนหยดต่อนาทีได้
2. สามารถส่งเสียงเตือนเมื่อน้ำเกลือใกล้จะหมดขวดได้

บทที่ 2

ทฤษฎีและหลักการเบื้องต้นของอุปกรณ์

2.1 น้ำเกลือ

ในที่นี้จะขอกล่าวถึง ชนิดของน้ำเกลือและหลักการให้น้ำเกลือพอสังเขป ดังนี้
ชนิดของน้ำเกลือ ที่ใช้บ่อยได้แก่

1. นอร์มัลซาลิน (Normal saline solution / NSS) หมายถึง น้ำเกลือธรรมดาที่มีความเข้มข้น 0.9% ซึ่งเท่ากับ ความเข้มข้นของเกลือในกระแสเลือดของคนปกติ มีอย่างขนาด 500 มล. และ 1000 มล.
2. 5% เดกซ์โทรส (5% Dextrose in water หรือ 5% D/W) หมายถึง น้ำตาลเดกซ์โทรสที่มีความเข้มข้น 5% ไม่มีเกลือแร่ผสม มีอย่างขนาด 500 มล. และ 1000 มล.
3. 5% เดกซ์โทรสในนอร์มัลซาลิน (5% Dextrose in NSS หรือ 5% D/NSS) หมายถึง น้ำตาลเดกซ์โทรสเข้มข้น 5% ผสมกับเกลือธรรมดา
4. 5% เดกซ์โทรสใน 1/3 นอร์มัลซาลิน (5% Dextrose in 1/3 NSS หรือ 5%D/(1/3)NSS) หมายถึง น้ำตาลเดกซ์โทรสเข้มข้น 5% ผสมกับน้ำเกลือที่มีความเข้มข้น 0.3% (เข้มข้นเพียง 1/3 ของน้ำเกลือธรรมดา) มีอย่างขนาด 500 มล. และ 1000 มล.

ข้อบ่งใช้และข้อควรระวัง

ข้อบ่งใช้ น้ำเกลือที่จะให้ผู้ป่วยที่มีอาการดังนี้

1. ขาดน้ำ (Dehydration) เนื่องจาก ท้องเดิน, อาเจียนรุนแรง (เช่น ภาวะทะเลาะไข้สุดต้น) , หอบ (เช่น หืด ปอดอักเสบ หลอดลมฝอยอักเสบ) ควรให้น้ำเกลือที่มี NSS ผสม เช่น NSS,5%D/NSS และ 5% D 1/3 NSS
2. ช็อก (Shock) เนื่องจาก เสียเลือด,เสียน้ำ หรือจากสาเหตุอื่น ควรให้น้ำเกลือที่มี NSS ผสมเช่นเดียวกับข้อ 1
3. หมดสติ หรือกินข้าวและน้ำไม่ได้มานานๆ ควรให้น้ำเกลือที่มีเดกซ์โทรสผสมกับน้ำเกลือ เช่น 5%D/NSS,5% D 1/3 NSS
4. น้ำตาลในเลือดต่ำ (Hypoglycemia) เนื่องจาก อดอาหารนานๆ, คีมี療法จัด, ไข้รักษาเบาหวานเกินขนาด ควรให้น้ำเกลือที่มีเดกซ์โทรสผสม เช่น 5% D/W,5% D/NSS และ D 1/3 NSS
5. ผู้ป่วยที่อดอาหารและน้ำ ก่อนและหลังการผ่าตัดควรให้น้ำเกลือที่มีเดกซ์โทรสผสมเช่นเดียวกับข้อ 3

5. ผู้ป่วยที่อดอาหารและน้ำ ก่อนและหลังการผ่าตัดควรให้น้ำเกลือที่มีเดกซ์โทรสผสมเช่นเดียวกันกับในข้อ 3

6. ผู้ป่วยที่ต้องฉีดยาเข้าทางหลอดเลือดดำวันละหลายๆ ครั้ง เลือกให้น้ำเกลือชนิดโคชนิคหนึ่งก็ได้ โดยให้ช้าๆ เพียงเพื่อให้มีสายน้ำเกลือคาบกับเส้นเลือด เพื่อให้เกิดความสะดวกในการฉีดยา

ข้อควรระวัง

1. ผู้ป่วยที่เป็นโรคหัวใจ โรคไตวาย หรือมีอาการบวมทั่วตัว ควรระมัดระวังเป็นพิเศษ และควรหลีกเลี่ยงการใช้น้ำเกลือที่มีความเข้มข้นของเกลือมาก เพราะอาจทำให้หัวใจวายหรือ เกิดภาวะน้ำคั่งในปอด หรือปอดบวมน้ำ (pulmo nary edema) ได้

2. ผู้ป่วยที่เป็นโรคความดันโลหิตสูง ควรหลีกเลี่ยงการใช้น้ำเกลือที่มีความเข้มข้นของเกลือมากๆ (5% D/NSS,NSS) ยกเว้นในกรณีที่มีภาวะการขาดเกลือร่วมด้วย

3. ผู้ป่วยที่เป็นเด็กเล็ก ไม่ควรใช้น้ำเกลือที่มี NSS ผสม ควรใช้น้ำเกลือที่มีความเข้มข้นน้อย ได้แก่ น้ำเกลือที่มีความเข้มข้น 0.3% เช่น 5% D/(1/3) NSS

การเลือกใช้น้ำเกลือ

1. นอร์มัลซาลิน (NSS)

ข้อบ่งใช้

1. สำหรับผู้ป่วยที่มีภาวะขาดน้ำหรือช็อก

2. เด็กที่มีภาวะขาดน้ำรุนแรงและไม่สามารถให้น้ำเกลือทางหลอดเลือดดำได้ (หลอดเลือดดำมีขนาดเล็กแทงไม่ได้) ให้ฉีด NSS เข้าได้หนังตรงหน้าขาหรือสะบักหลังขนาด 20 -30 มล.ค่อน้ำหนักตัว 1 กก

ข้อควรระวัง

1. ห้ามให้ NSS ทางหลอดเลือดดำในเด็กเล็กเพราะปริมาณของเกลือเข้มข้นกว่าเกลือที่มีอยู่ในกระแสเลือดของเด็ก อาจเป็นอันตรายได้

2. ผู้ป่วยที่มีภาวะ หัวใจวาย, ไตวาย, ความดันโลหิตสูง หรือบวมทั่วตัว ควรหลีกเลี่ยงการให้ NSS ยกเว้นในผู้ป่วยที่มีภาวะการขาดเกลือ

ขนาดที่ใช้

1. ผู้ใหญ่ 1000 มล. - 2000 มล. ใน 12-24 ชม. ในผู้ป่วยที่มีภาวะขาดน้ำรุนแรงหรือช็อกในระยะแรกควรให้เร็วๆ จนอาการดีขึ้น จึงค่อยลดปริมาณลงให้ช้าๆ

2. 5% เดกซ์โทรสในนอร์มัลซาลิน (5% D/NSS)

ข้อบ่งใช้

สำหรับผู้ป่วยที่มีภาวะขาดน้ำ ช็อก อดอาหาร หรือมีภาวะน้ำตาลในเลือดต่ำ

ข้อควรระวัง

เช่นเดียวกับ นอร์มัลซาไลน์ (NSS)

ขนาดที่ใช้

เช่นเดียวกับ นอร์มัลซาไลน์ (NSS)

3. 5% เดกซ์โทรส (5% D/W)

ข้อบ่งใช้

สำหรับเด็กอ่อนและผู้ใหญ่ที่มีภาวะน้ำตาลในเลือดต่ำ หรืออดอาหาร

ขนาดที่ใช้

ผู้ใหญ่ 1000 มล. - 2000 มล. ใน 12-24 ชม.

เด็ก 100 มล. ต่อน้ำหนักตัว 1 กก. ใน 24 ชม.

4. 5% เดกซ์โทรสใน 1/3 นอร์มัลซาไลน์ (5% D/ (1/3) NSS)

ข้อบ่งใช้

สำหรับเด็กที่มีภาวะขาดน้ำ ช็อก ใช้เลือดออก อดอาหาร หรือมีภาวะน้ำตาลในเลือดต่ำ

ขนาดที่ใช้

100 มล. ต่อน้ำหนักตัว 1 กก. ใน 24 ชม. ถ้ามีภาวะขาดน้ำรุนแรง หรือช็อก ใน 2 ชม.

แรก ควรให้ขนาด 20-40 มล. ต่อน้ำหนักตัว 1 กก.

อันตรายจากน้ำเกลือ

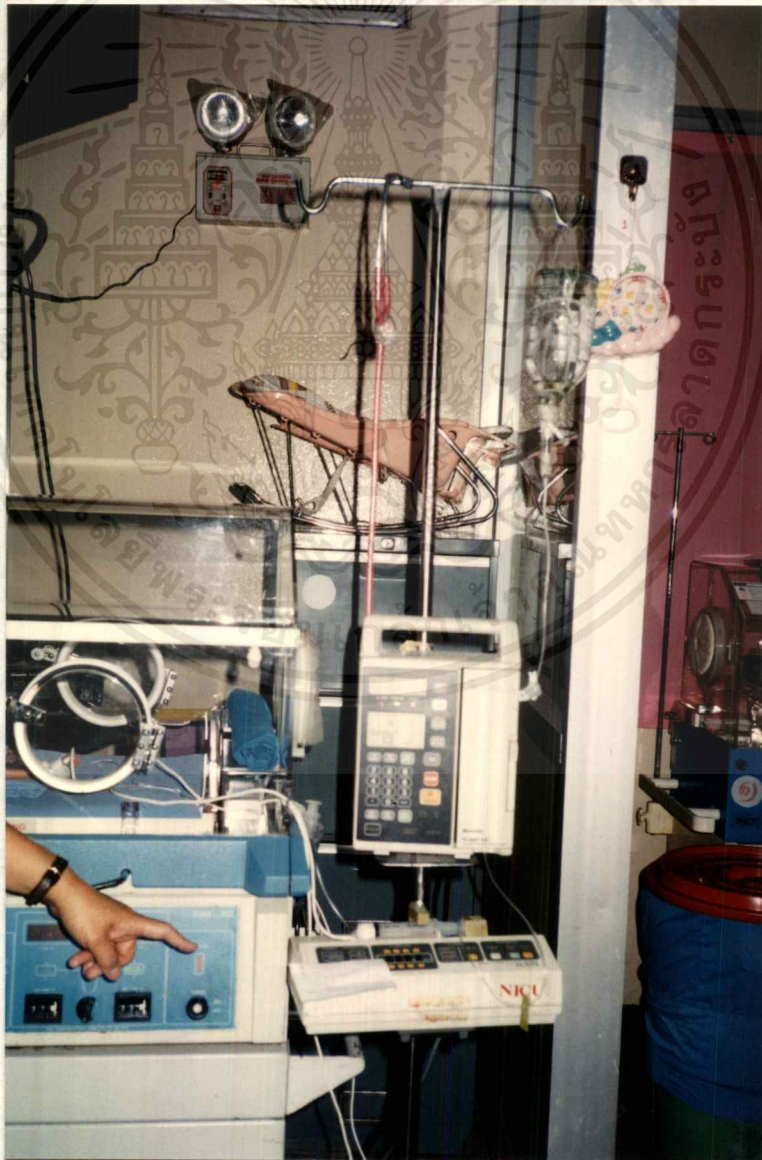
1. ถ้าเครื่องใช้และน้ำยาไม่สะอาดหรือเทคนิคการให้ไม่ถูกต้อง อาจทำให้ผิวหนังบริเวณที่แทงเข็ม น้ำเกลืออักเสบติดเชื้อได้
2. ถ้ามีฟองอากาศ เพราะไล่อากาศจากสายน้ำเกลือไม่หมดฟองอากาศจะเข้าไปในเส้นเลือดดำและเข้าสู่หัวใจอาจเป็นอันตรายได้
3. มีอาการหนาวสั่น เนื่องจากการแพ้ น้ำเกลือ
4. ถ้าใช้น้ำเกลือชนิดที่มีความเข้มข้นของเกลือมากกว่าความเข้มข้นของเกลือในเลือด อาจเป็นอันตรายถึงตายได้ โดยเฉพาะอย่างยิ่งในเด็กเล็ก
5. ถ้าให้น้ำเกลือมาก หรือเร็วเกินไป อาจทำให้บวมมีน้ำคั่งในปอดหรือหัวใจวายถึงตายได้ โดยเฉพาะอย่างยิ่งในเด็กเล็ก คนสูงอายุ คนที่เป็นโรคหัวใจ หรือโรคไตอยู่ก่อน

ข้อควรปฏิบัติ

1. ควรให้น้ำเกลือเฉพาะในผู้ป่วยที่มีความจำเป็นจริงๆ เท่านั้น น้ำเกลือไม่ใช่ยาบำรุง ยาเพิ่มเลือด หรือยาเพิ่มแรงและก็ไม่ใช่ยาที่ใช้แทนอาหาร จึงไม่ควรให้อย่างพร่ำเพรื่อ
2. ควรเตรียมเครื่องใช้ให้สะอาดปราศจากเชื้อโรค และทำตามวิธีที่ถูกต้อง

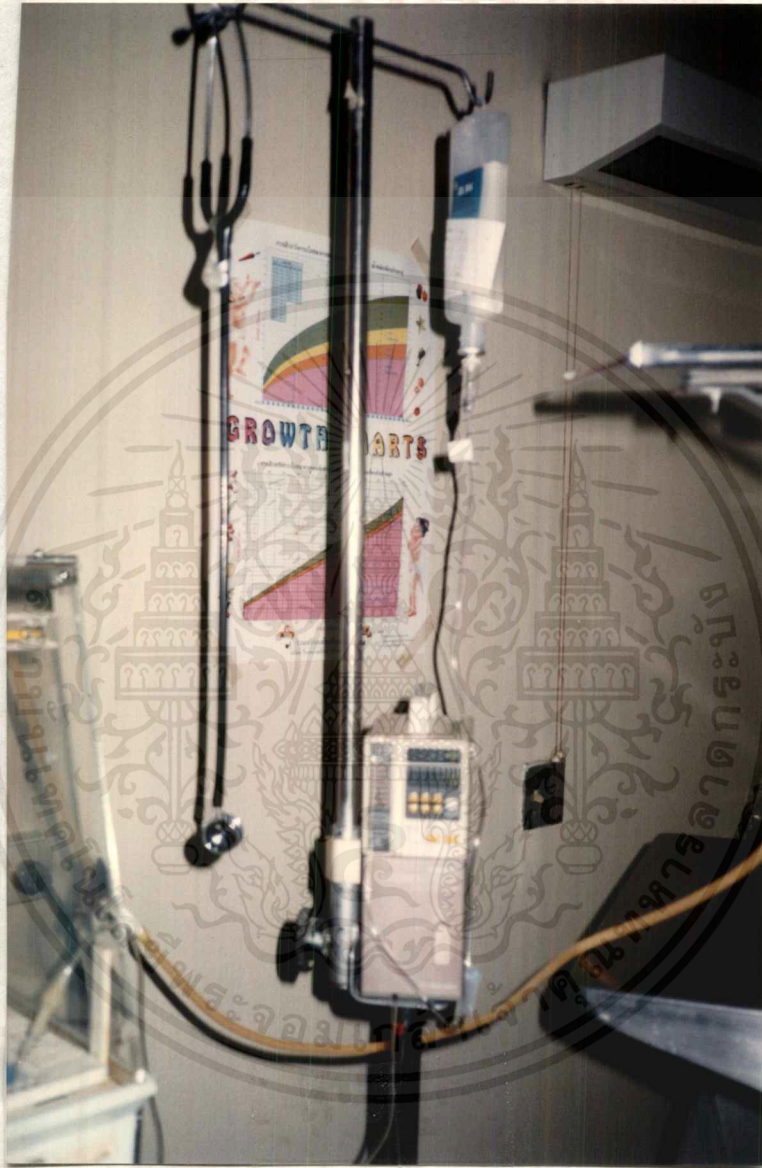
3. ควรให้น้ำเกลือช้าๆ หรือน้อยๆ ไว้ก่อน เมื่อเห็นว่าน้อยเกินไปก็เพิ่มเติมในภายหลังได้อย่าให้เร็วเกินไป (ยกเว้นในกรณีที่มีภาวะขาดน้ำรุนแรงหรือช็อก ควรให้เร็วๆ ใน 1-2 ชม. แรก) มิฉะนั้นอาจทำให้บวมมีน้ำคั่งในปอด หรือหัวใจวายได้
4. หมั่นตรวจผู้ป่วยอย่างใกล้ชิด (โดยเฉพาะในเด็กเล็ก คนสูงอายุ คนที่เป็นโรคหัวใจหรือไตวาย) ถ้าหากมีอาการหายใจหอบ และฟังปอดมีเสียงกรอบแกรบ (crepitation) แสดงว่ามีน้ำคั่งในปอด เนื่องจากให้น้ำเกลือมากเกินไป ต้องหยุดน้ำเกลือ และพิจารณาขับปัสสาวะ เช่น ลาซิกซ์ 1-2 หลอดเข้าเส้นเลือดทันที หากยังหอบอยู่ให้รีบพาไปโรงพยาบาล
5. ถ้ามีอาการหนาวสั่นระหว่างให้น้ำเกลือ แสดงว่าผู้ป่วยแพ้น้ำเกลือ ให้ถอดเข็มน้ำเกลือออก และพิจารณาแก้แพ้ เช่น คลอเฟนิรามีน 1/2 - 1 หลอดเข้ากล้ามเนื้อทันที ถ้าจำเป็นต้องให้น้ำเกลือต่อ ควรเปลี่ยนขวดใหม่

เครื่องนับจำนวนหยดน้ำเกลือที่ใช้ในโรงพยาบาล มีลักษณะดังรูป



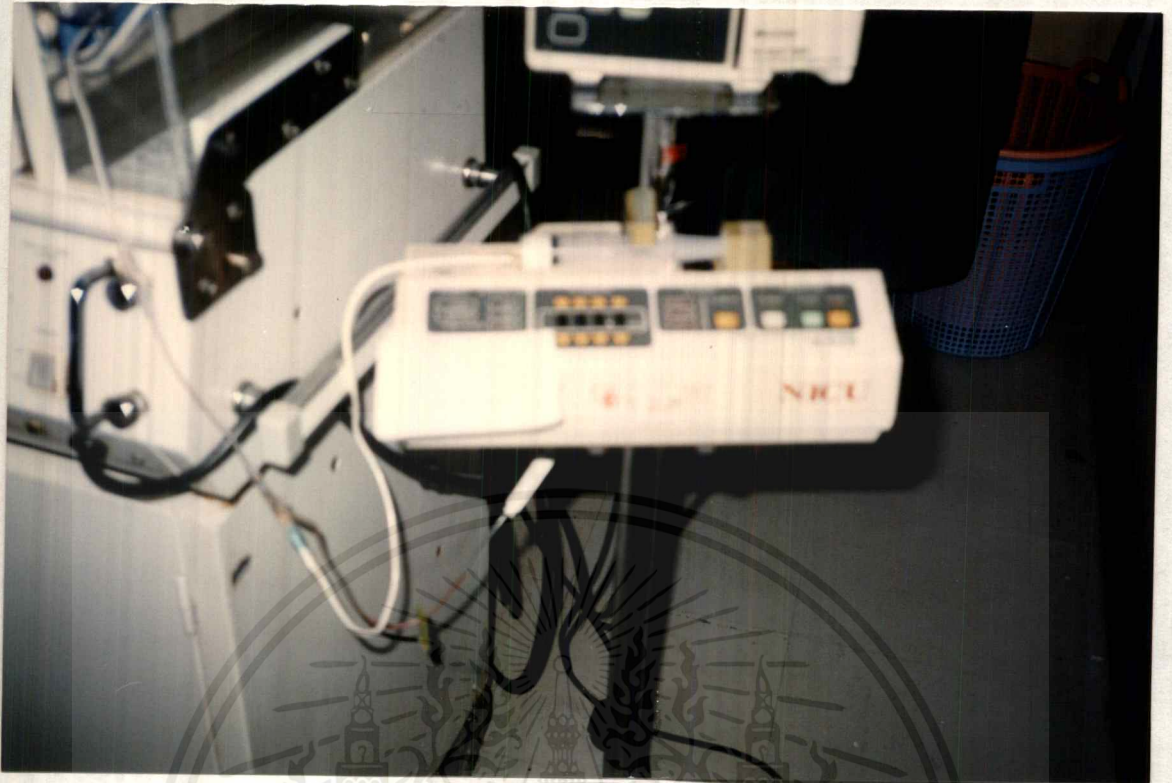
รูปที่ 2.1 แสดงเครื่องนับจำนวนหยดน้ำเกลือ แบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้ขาดเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงเครื่องนับจำนวนหยดน้ำเกลือ แบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงชุดให้อาหารทางหลอดเลือด (Terufusion Syringe Pump)

ชุดให้อาหารทางหลอดเลือด (Terufusion Syringe Pump) มักใช้ร่วมกับเครื่องนับจำนวนหยดน้ำเกลือต่างๆ ไป มีข้อแตกต่าง คือ รูปแบบนี้จะใช้กระบอกฉีดยา (Syringe) มีขนาด 10, 20, 30, 50 และ 100 มิลลิลิตรซึ่งภายในกระบอกจะบรรจุสารอาหาร(ของเหลว) หรือน้ำเกลือ ที่มีปริมาตรตามแต่ขนาดของกระบอกฉีดยา เช่น ถ้าเป็น 10 มิลลิลิตร ผู้ป่วยก็จะได้รับ สารอาหารที่มีปริมาณตั้งแต่ 0 มิลลิลิตรถึง 10 มิลลิลิตร หรือถ้าต้องการให้ผู้ป่วยรับน้ำเกลือตั้งแต่ 100 มิลลิลิตรถึง 1000 มิลลิลิตร ก็ต่อชุดน้ำเกลือเข้ากับถุงน้ำเกลือขนาดมาตรฐานได้เลย เป็นต้น

เครื่องให้อาหารผู้ป่วยนี้ มีข้อดีคือ ในกรณีที่เรต้องการให้ปริมาณน้ำเกลือหรือสารอาหารผู้ป่วยน้อยกว่า 100 มิลลิลิตร เช่นผู้ป่วยที่เป็นเด็กหรือท้องร่วง ก็สามารถใช้เครื่องมือดังกล่าวนี้ได้

เครื่องนับจำนวนหยดน้ำเกลือที่ใช้ในโรงพยาบาลเป็นเครื่องที่สั่งจากต่างประเทศ มีลักษณะพิเศษดังนี้ คือ

1. สามารถควบคุมจำนวนหยดน้ำเกลือได้ ควบคุมการไหลของน้ำเกลือให้ได้จำนวนหยดที่ต้องการ
2. มีหน้าปัทม์แสดงผลการนับหยดน้ำเกลือทุกหยด
3. สามารถนับจำนวนหยดน้ำเกลือต่อนาทีได้
4. สามารถป้อนค่าปริมาณของน้ำเกลือที่ใช้ได้
5. กล่องควบคุมอยู่ติดกับเสาน้ำเกลือเป็นการถาวร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ก็มีข้อเสียดังนี้

1. มีน้ำหนักมาก
2. ต้องมีชุดน้ำเกลือสำหรับเครื่องโดยเฉพาะ ถ้าไม่ใช่ชุดดังกล่าว เครื่องจะไม่สามารถควบคุม การไหลของหยดน้ำเกลือได้
3. ราคาสูงมาก

2.2 ชนิดของเข็มน้ำเกลือและการใช้งาน

การเทียบระหว่างเข็มน้ำเกลือมาตรฐานกับเข็มชนิดไมโครดรอป ซึ่งคำนวณตามความต้องการ แคลลอรี่ ใน 1 วัน ถ้าผู้ป่วยมีน้ำหนัก 50 กิโลกรัม $10 \text{ กิโลกรัม} \times 100 = 1000 \text{ แคลลอรี่}$

$10 \text{ กิโลกรัม} \text{ ต่อไป} \times 50 = 500 \text{ แคลลอรี่}$

$10 \text{ กิโลกรัม} \text{ ต่อไป} \times 20 = 200 \text{ แคลลอรี่}$

$10 \text{ กิโลกรัม} \text{ ต่อไป} \times 10 = 100 \text{ แคลลอรี่}$

$10 \text{ กิโลกรัม} \text{ ต่อไป} \times 10 = 100 \text{ แคลลอรี่}$

จะได้ว่า ถ้าผู้ป่วยหนัก 50 กิโลกรัม ต้องใช้แคลลอรี่ ประมาณ 2000 แคลลอรี่ ใน 1 วัน หรือ 24 ชั่วโมง

เข็มน้ำเกลือมาตรฐาน น้ำเกลือ 20 drop จะมีปริมาณน้ำเกลือ 1 cc.

เข็มน้ำเกลือชนิดไมโครดรอป น้ำเกลือ 60 drop จะมีปริมาณน้ำเกลือ 1 cc.

ถ้าใน 24 ชั่วโมง ต้องการให้น้ำเกลือ 2000 cc.

ใน 1 ชั่วโมง ต้องการให้น้ำเกลือ 88.88 cc.

ใน 1 นาที ต้องการให้น้ำเกลือ 1.47 cc. ประมาณ 1.5 cc.

ดังนั้นใน 1 นาทีต้องการให้น้ำเกลือ 30 drop เมื่อใช้ เข็มน้ำเกลือแบบมาตรฐาน

ใน 1 นาทีต้องการให้น้ำเกลือ 90 microdrop เมื่อใช้เข็มน้ำเกลือแบบไมโครดรอป

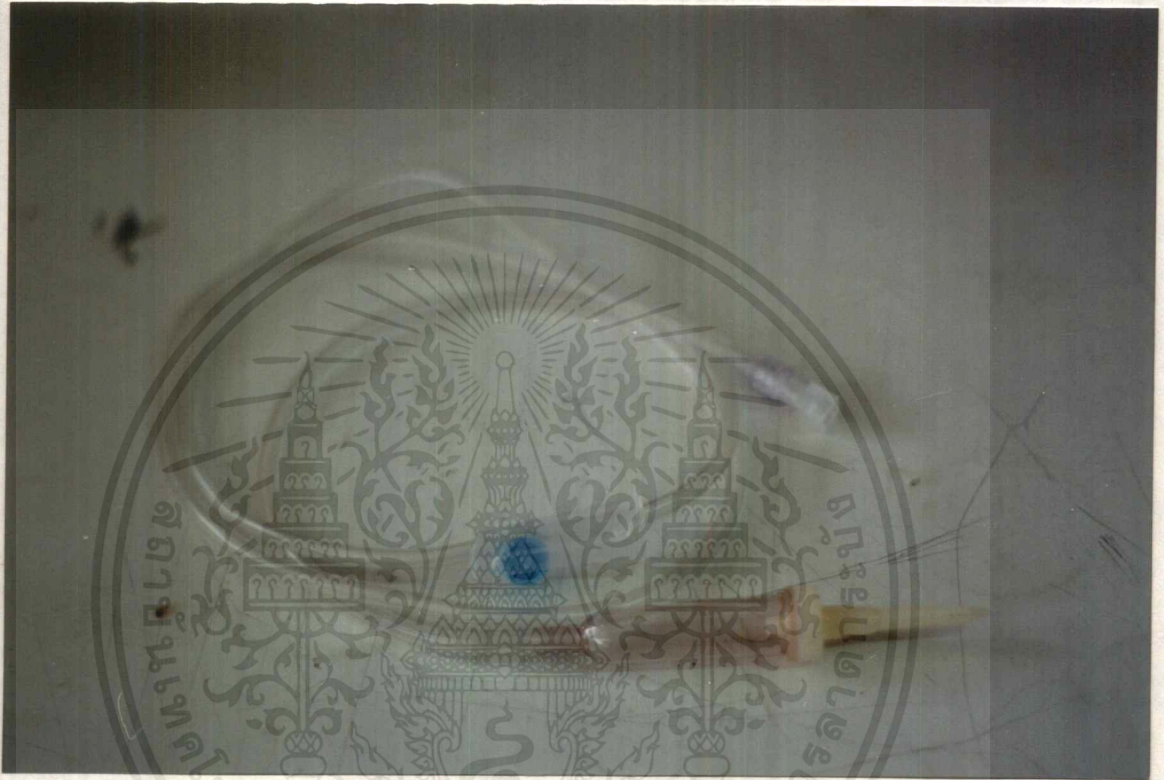
ส่วนในผู้ป่วยที่เป็นเด็กนั้น ต้องการแคลลอรี่ 1000 cc. ต่อ 1 วัน

ดังนั้นใน 1 นาทีต้องการให้น้ำเกลือ 15 drop เมื่อใช้เข็มน้ำเกลือแบบมาตรฐาน

ใน 1 นาทีต้องการให้น้ำเกลือ 45 microdrop เมื่อใช้เข็มน้ำเกลือแบบไมโครดรอป

1. ชุดน้ำเกลือขนาดมาตรฐาน (Standard Infusion Set)

ชุดน้ำเกลือชนิดนี้ประกอบด้วยเข็มน้ำเกลือ ขนาดมาตรฐานทั่วไปใช้กับผู้ใหญ่ มีลักษณะดังรูป



รูปที่ 2.4 แสดงชุดเข็มน้ำเกลือขนาดมาตรฐาน (Standard Infusion Set)

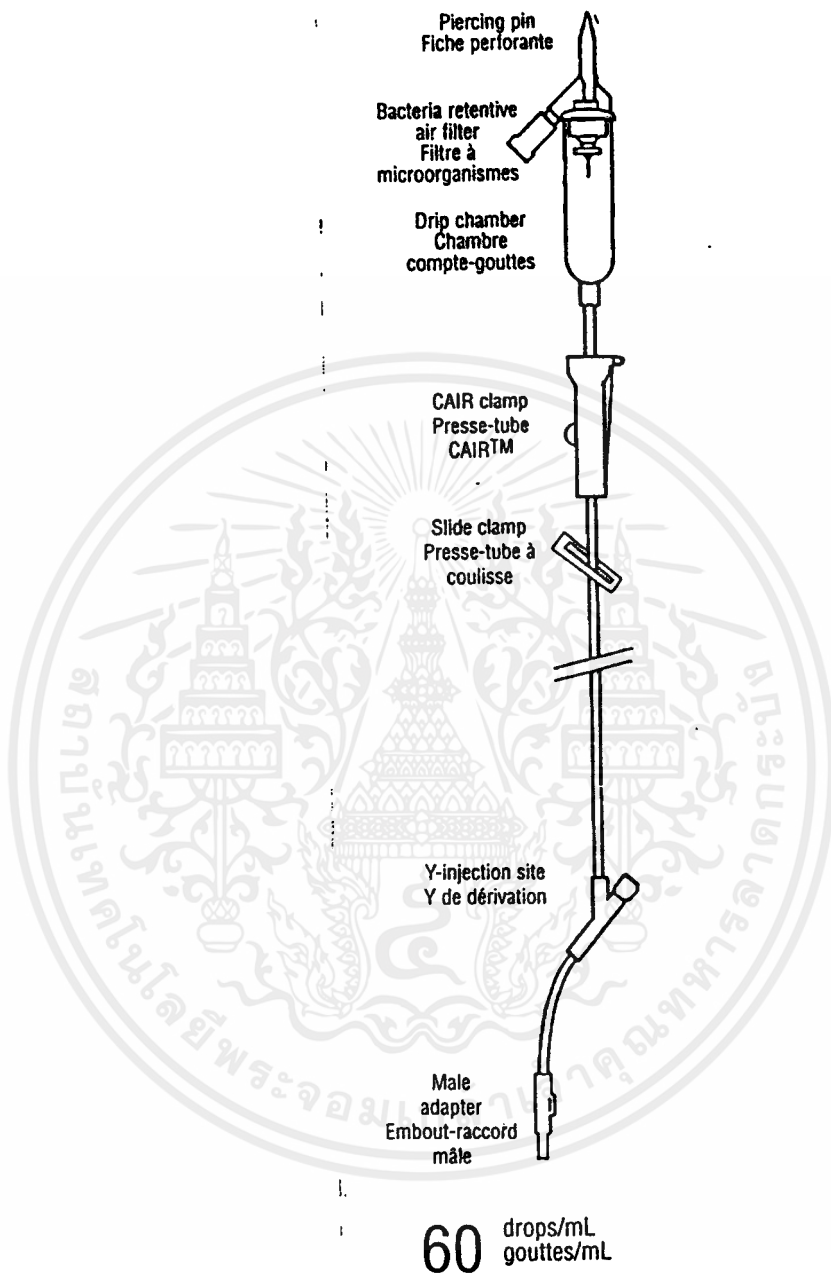
วิธีการใช้งานจะกล่าวในรายละเอียดของบทที่ 4 ต่อไป

คำเตือน

1. ชุดเข็มน้ำเกลือใช้เพียงครั้งเดียวเท่านั้น
2. ไม่ควรเก็บไว้ที่มีแสงแดดและอุณหภูมิสูงๆ
3. ไม่ควรใช้ถ้าบรรจุภัณฑ์ฉีกขาด

2. ชุดน้ำเกลือชนิดไมโครครอป

ชุดเข็มน้ำเกลือชนิดนี้ประกอบด้วยส่วนต่างๆ ดังรูป คือ



รูปที่ 2.5 แสดงชุดเข็มน้ำเกลือชนิดไมโครครอป

วิธีการใช้

1. ใช้เทคนิคการปลอดเชื้อ ให้ถอดส่วนที่คลุมออก
2. ทำความสะอาดบริเวณจุดงู้น้ำเกลือ
3. ปิด clair clamp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สำหรับน้ำเกลือที่บรรจุขวด (Rigid container) ทำความสะอาดขวด โดยตั้งขวดหงายขึ้น แล้วแทงเข็มเข้าตรงกลางของจุกขวด อย่าบิด หมุนเข็มหรือทะแยงเข็ม แล้วขยักขวดแขวนกับเสาทันที ปลดขั้วน้ำเกลือหยดลง chamber (ประมาณครึ่งหนึ่งของ chamber)

5. สำหรับน้ำเกลือที่บรรจุถุง (collapsible) ให้เตรียมส่วนที่จะใส่เข็ม ถอดส่วน air filter ออกแล้วเปลี่ยน pin cover ใส่แทน แทงเข็มโดยการหมุน แขนงถุง แล้วบีบส่วน chamber เพื่อให้ น้ำเกลือไหลลงมาประมาณครึ่ง chamber

6. ค่อยกับส่วนที่แทงกับเส้นเลือด (Venipuncture)

7. เปิด clamp แล้วปลดขั้วน้ำให้ผ่านไล่ฟองอากาศ จากท่อ จนถึงปลายเข็มที่แทงเส้นเลือด แล้วปิดเข็ม

8. หลังจากแทงเส้นเลือดเรียบร้อยแล้ว ให้ปรับความเร็วของน้ำเกลือโดย clamp โดย 60 หยด จะมีปริมาณ 1 มิลลิลิตร

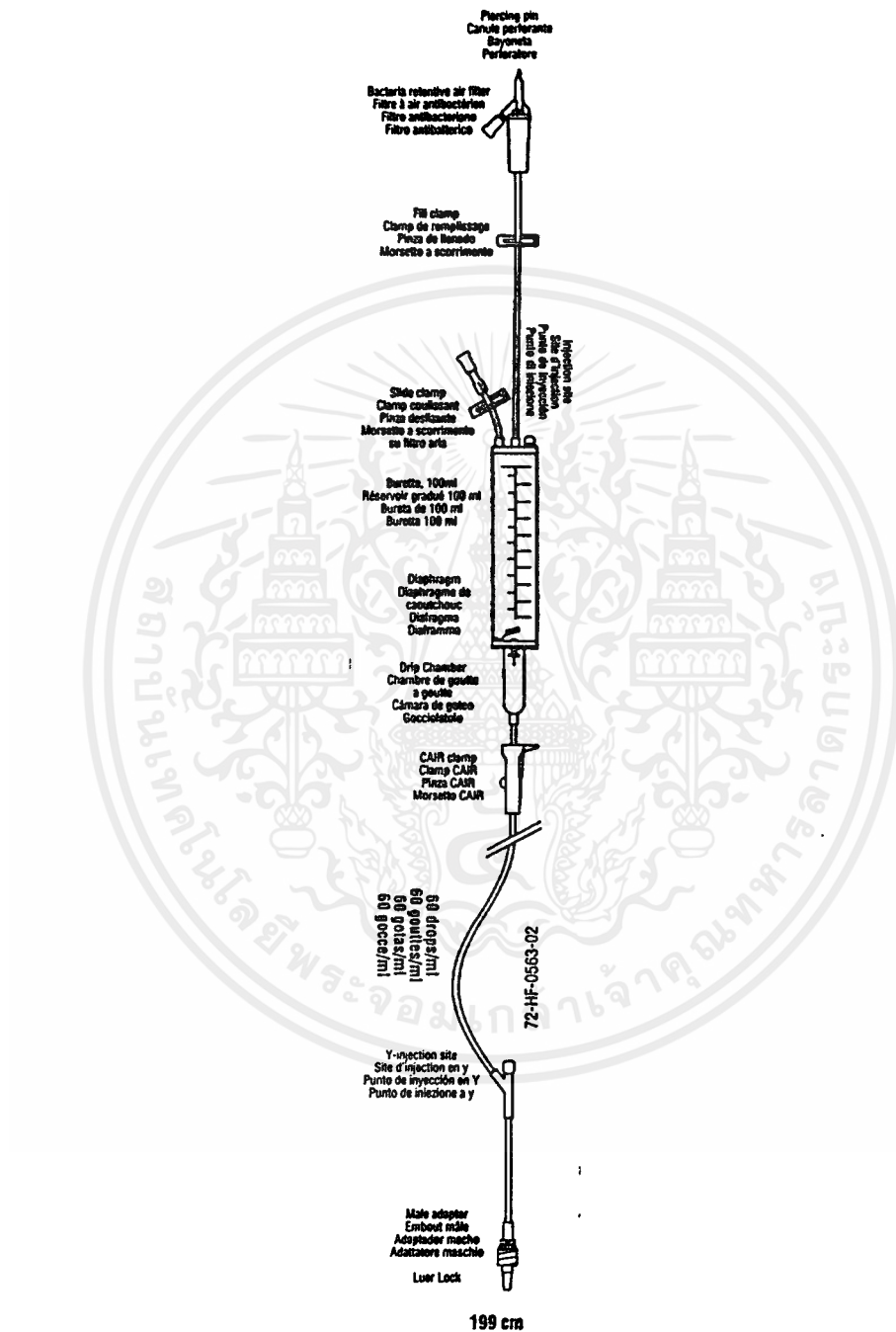
9. ถ้าต้องการหยุดไหล โดยไม่รบกวน flow control setting ให้ขยับเลื่อนสายท่อเข้าไปในรูปิด

หมายเหตุ ใช้เข็มขนาด 1.2 มิลลิเมตร (18G) หรือเล็กกว่า สำหรับแทงถุงในจุดอื่น เมื่อท่อสายดึงหรือถูกดึงลากอาจจะทำให้ flow control clamp ลดประสิทธิภาพลงได้

คำแนะนำ

1. ชุดน้ำเกลือนี้ใช้เพียงครั้งเดียวเท่านั้น
2. ก่อนใช้จะฆ่าเชื้อโดยการอบแก๊ส

3. ชุดเข็มน้ำเกลือไมโครครอปพร้อมเข็มฉีดยา 100 มิลลิลิตร (Solutes 100 × 60)
ชุดเข็มน้ำเกลือนี้ ใช้ในการให้อาหารผ่านทางหลอดศีดยาร่วมกับเครื่องนับจำนวนหยดน้ำเกลือ
แสดงได้ดังรูป



รูปที่ 2.6 แสดงชุดเข็มน้ำเกลือพร้อมเข็มฉีดยา ขนาด 100 มิลลิลิตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้

1. ใช้เทคนิคการปลดออกเชื้อ ถอดส่วนที่หุ้มออกแล้วดำเนินการต่อ
 2. ปิด fill clamp และ cair clamp
เปิด slide clamp
 3. สำหรับขวดแก้ว (Rigid) ทำความสะอาดส่วนที่จะแทงเข็ม (outlet port) ตั้งขวดขึ้น
แทงเข็มตรงๆ เข้ายังส่วนตรงกลางของ port อย่าบิดหมุนหรือทะแยงเข็ม แล้วยกขวดแขวนทันที
 4. สำหรับถุง (Collapsible) ทำความสะอาดส่วน outlet port ให้ถอด Bacteria retentive
air filter ออก แล้วใส่ Pin cover แทน แทงเข็มโดยการหมุนจนขอบของ pin ชิดกับขอบของ port
แล้วแขวนถุง
 5. เปิด fill clamp ให้น้ำอย่างน้อย 35 มิลลิลิตร ไหลสู่หลอด (Burette)
ปิด fill clamp
บีบ drip chamber จนให้น้ำไหลลงมาประมาณครึ่งหนึ่งของ chamber
 6. ถอดส่วนที่ปิด male adapter
เปิด cair clamp แล้วปล่อยให้ น้ำไหลเพื่อไล่อากาศออกจากท่อ
ปิด cair clamp
 7. ต่อ male adapter กับส่วนที่แทงเข้าเส้นเลือด (Venipuncture) ให้หมุนเกลียวตามเข็มนาฬิกา เพื่อให้ล็อก
- การให้สารอาหาร (solution) โดย
1. วัดปริมาณน้ำใน burette (ที่ต้องการให้)
 2. ปิด fill clamp
 3. เปิด slide clamp
 4. ให้ประมาณปริมาณของน้ำที่จะให้โดนปรับจาก cair clamp 60 drop จะเท่ากับ 1
มิลลิลิตร (โดยประมาณ)
- ส่วนของ diaphragm จะปิดโดยอัตโนมัติ เมื่อไม่มีน้ำใน burette continuous น้ำจากขวดน้ำ
เกลือโดยตรง
1. ปิด slide clamp
 2. เปิด fill clamp
 3. ปรับอัตราการไหลของน้ำจาก cair clamp เพื่อให้ diaphragm เปิดตลอด ให้น้ำใน
burette อย่างน้อย 35 มิลลิลิตร

-refill burette

1. เปิด slide clamp
2. cair clamp เหมือนเดิม

3. เปิด fill clamp ให้น้ำไหลลง burette ตามปริมาณที่กำหนดแล้ว
4. ปิด fill clamp เริ่มให้น้ำเกลือโดยบีบ drip chamber เพื่อให้ diaphragm เปิด ถ้า drip chamber น้ำท่วม (น้ำเต็ม) ให้ปิด clamp ทุกอัน แล้วคว่ำ burette ลง บีบ drip chamber เพื่อไล่น้ำส่วนเกินกลับยัง burette แล้วแขวนใหม่ ปรับ clamp เหมือนเดิม

การให้ยา

1. ถ้าให้ในขวดน้ำเกลือ ให้เติมยาลงในขวดก่อนที่จะแทงเข็ม (set)
2. ถ้าให้ใน burette

เปิด slide clamp ปิด fill clamp

ทำความสะอาด injection site เติมน้ำโดยใช้ syringe หรือ I.V. set กับเข็ม

3. ถ้าใน Y-Injection

ทำความสะอาด

เติมน้ำโดยใช้ syringe และ I.V. set กับเข็ม

คำเตือน

1. ถ้าสายคิ่งหรือเอียงจะทำให้ clamp ประสิทธิภาพลดลง
2. ไม่ใช้กับเลือดหรือ plasma
3. เปลี่ยนทุก 24 ชั่วโมง
4. ใช้ครั้งเดียว

2.3 ชนิดของภาชนะบรรจุน้ำเกลือและการใช้งาน

ภาชนะบรรจุน้ำเกลือแบ่งออกได้ดังนี้ คือ

1. ชนิดขวด

ภาชนะใส่น้ำเกลือชนิดนี้ มีลักษณะเป็นขวดแก้ว มีปริมาตรเพียงขนาดเดียว คือ 1000 มิลลิลิตร ก่อนใช้ต้องผ่านการฆ่าเชื้อโรคก่อน ส่วนการใช้งานดังกล่าวแล้วข้างต้นในส่วนของชุดเข็มน้ำเกลือ

2. ชนิดถุง

ภาชนะใส่น้ำเกลือชนิดนี้ มีปริมาตรใช้งานได้แก่ 1000 มิลลิลิตรและ 500 มิลลิลิตร ในการใช้งานนั้น กรณี 1000 มิลลิลิตรนิยมใช้กับผู้ป่วยที่เสียน้ำในร่างกายมาก จากการอ่อนเพลียจากโรคภัยไข้เจ็บต่างๆ ส่วน 500 มิลลิลิตรนั้น นิยมใช้กับผู้ป่วยที่ต้องควบคุมปริมาณน้ำเกลือมากเป็นพิเศษ เช่น โรคเบาหวาน ท้องเสีย เป็นต้น ส่วนการวินิจฉัยโรคนั้นให้อยู่ในดุลพินิจของแพทย์ต่อไป

2.4 ฟิสิกส์ของน้ำเกลือ

แรงดึงผิว

การทำให้น้ำเกลือไหลผ่านเข็มน้ำเกลือลงมาเป็นหยดๆ นั้น สามารถอธิบายได้จากหลักการทางฟิสิกส์ ในเรื่องกลศาสตร์ของไหล ซึ่งของไหลเป็นสารที่ไม่มีรูปร่างแน่นอนเปลี่ยนแปลงไปตามสภาพแวดล้อม ของไหลที่พบโดยทั่วไปแบ่งเป็น ของเหลวและก๊าซ ซึ่งน้ำเกลือที่เป็นของเหลวอยู่รวมกันนั้น จะไหลผ่านเข็มน้ำเกลือลงมาเป็นหยดได้นั้น เกิดมาจากความตึงผิวของน้ำเกลือ ซึ่งภายในของเหลวแต่ละโมเลกุลจะถูกแรงดึงร่วม (cohesive force) กระทำเท่ากันหมดในทุกทิศทาง เนื่องจากโมเลกุลข้างเคียง แต่โมเลกุลที่ผิวของของเหลวนอกจากจะถูกกระทำด้วยแรงดึงแยก (adhesive force) เนื่องจากภาชนะที่บรรจุ หรือ อากาศด้วยทำให้ผิวของของเหลวเปรียบเสมือนเป็นแผ่นเยื่อยืดหยุ่นที่ถูกดึงด้วยแรงทั้ง 2 ข้างดัน โดยมีทิศตั้งฉากกับแผ่นเยื่อ ถ้าแรงดึงร่วมมีค่ามากกว่าแรงดึงแยกของเหลวนั้นก็ไม่น่าเปียก หรือเกาะติดภาชนะที่บรรจุ เช่น หยดปรอทบนกระดาษหนังสือพิมพ์ แต่ถ้าแรงดึงร่วมมีค่าน้อยกว่าแรงดึงแยก ของเหลวนั้นก็เปียกหรือเกาะติดภาชนะที่บรรจุ เช่น หยดน้ำบนกระดาษหนังสือพิมพ์

เมื่อมีแรงภายนอกดึงแผ่นเยื่อตามแนวสัมผัสกับแผ่นเยื่อ เนื่องจาก แผ่นเยื่อมีความยืดหยุ่นจึงเกิดแรงต้านแรงภายนอกนั้นด้วยขนาดที่เท่ากันแต่ทิศทางตรงกันข้าม เรียกแรงที่เกิดในแผ่นเยื่อของเหลวว่าแรงดึงผิว (surface tension force)

กำหนดให้ความตึงผิวเป็นอัตราส่วนของแรงดึงผิวขณะแผ่นเยื่อถูกดึงขาด ต่อความยาวของแผ่นเยื่อด้านที่ถูกแรงกระทำ

$$\beta \equiv F_s / 2L$$

F_s แทนแรงดึงผิวขณะแผ่นเยื่อขาด (N)

L แทนความยาวด้านที่ถูกแรงกระทำ (m)

β แทนความตึงผิว (N/m)

2.5 อินฟราเรด (Infrared)

ในการศึกษาเรื่องธรรมชาติของแสง แต่เดิมในปี 1801 เริ่มศึกษาถึงแสงที่มองเห็นได้ (visible light) และพบว่าเมื่อเปรียบว่าเอา ซึ่งเรียกแสงในช่วงนั้นว่า "Infra-Red" จากนั้นจึงมีการศึกษาถึง Infrared อีกหลายครั้ง

"Heinrich Rubens" ศาสตราจารย์แห่งมหาวิทยาลัยเบอร์ลินเป็นผู้หนึ่งที่ได้ริเริ่มค้นคว้าแสง Infrared โดยมีวิธีศึกษาโดยทั่วไปและแยกศึกษาถึงการแผ่รังสี Infrared เช่นการดูดกลืนเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ การส่งผ่านแสง Infrared ผ่านตัวกลาง และได้ประดิษฐ์เครื่องมือวัดคุณสมบัติต่าง ๆ ของแสง Infrared ด้วย

ในปี 1900 "Coblentz" (ชาวอังกฤษ) เป็นผู้คำนวณหาค่า Transmittance ของแสง Infrared ที่ผ่านสารตัวกลางต่าง ๆ และเป็นผู้กำหนดค่าสเปกตรัมของแสง Infrared หรือ ค่าความยาวคลื่นของแสง Infrared

- Infrared Sources

แหล่งกำเนิดแสง Infrared โดยทั่วไป จะมีค่าความร้อนทางไฟฟ้าอยู่ประมาณ 1800 °C เช่น

"Nernst glower" หรือ "Nernst filament" (เฝ้ามักทำกับออกไซด์จาก Zr, Th, Ce, Y, Er)

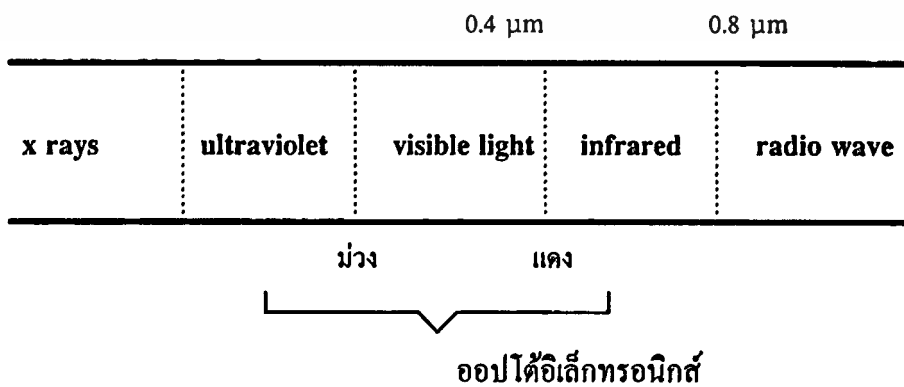
"Globar" (silicon carbide)

หรือวัสดุที่มาจากเซรามิคต่าง ๆ

โดยแสง Infrared ที่ออกมาจากแหล่งกำเนิดดังกล่าวนี้จะปล่อยความเข้มแสง Infrared อยู่ในช่วงความถี่ 5000 - 7000 cm⁻¹ ซึ่งขึ้นอยู่กับชนิดของตัวกำเนิดแสง

2.6 ออปโตอิเล็กทรอนิกส์ (Optoelectronics)

แสงเป็นคลื่นแม่เหล็กไฟฟ้าแบบหนึ่ง มีความเร็ว 3 x 10⁸ m/s แสงที่มองเห็นได้ (visible light) เป็นสเปกตรัมหนึ่งของคลื่นแม่เหล็กไฟฟ้าโดยมีความยาวคลื่นอยู่ระหว่าง 0.4 μm (แสงสีม่วง) ไปจนถึง 0.8 μm (แสงสีแดง) ทว่าอุปกรณ์อิเล็กทรอนิกส์ที่ทำงานเกี่ยวกับแสงหรือเรียกว่า "optoelectronics" นั้นจะมีย่านการใช้งานมากกว่า ย่านแสงที่มองเห็น เพราะมันสามารถทำงานได้ในย่านเหนือม่วง หรือ อัลตราไวโอเล็ต และ ได้แดงหรืออินฟราเรด ในรูปที่ 2.7 แสดงสเปกตรัมของคลื่นแม่เหล็กไฟฟ้า และสเปกตรัมย่านการใช้งานของอุปกรณ์ออปโตอิเล็กทรอนิกส์



รูปที่ 2.7 สเปกตรัมของคลื่นแม่เหล็กไฟฟ้า และแสดงย่านการใช้งานของออปโตอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์อปโตอิเล็กทรอนิกส์ที่จะกล่าวถึงในที่นี้แบ่งออกเป็น อุปกรณ์ตรวจจับแสง หรือเซนเซอร์ และอุปกรณ์กำเนิดแสง หรือ อิมิตเตอร์

อุปกรณ์ตรวจจับแสง

-ตัวรับแสง (Receiver)

โฟโตไดโอด (Photodiode)

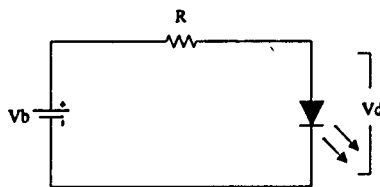
เป็นไดโอดที่จะทำงานเมื่อมีแสงมาตกกระทบ ในสภาวะที่การนำโฟโตไดโอดไปต่อใช้งานต้องต่อในลักษณะไบอัสกลับคือ ต่อแอโนดเข้ากับไฟลบ แคโทดเข้ากับไฟบวก จนเมื่อมีแสงมาตกกระทบจึงจะสามารถนำกระแสได้ แต่อย่างไรก็ตามในขณะที่ไม่มีแสงมาตกกระทบ ก็จะมีกระแสรั่วไหลอยู่ค่าประมาณ $10 \mu A$ เมื่อมีแสงมาตกกระทบกระแสจะไหลเพิ่มเป็นประมาณ $100 \mu A$

โฟโตไดโอดเป็นอุปกรณ์ที่มีกำลังไฟฟ้าต่ำ อิมพีแดนซ์สูง และเมื่อมีการนำมาใช้จำเป็นต้องมีการออกแบบวงจรที่ซับซ้อนกว่า LDR แต่ทว่าการตอบสนองในการทำงานของโฟโตไดโอดเป็นเชิงเส้น และมีความไวในการทำงานสูงกว่า LDR มาก เพราะใช้เวลาดตอบสนองในการทำงานเพียง 200 ns เท่านั้น มักนำโฟโตไดโอดไปใช้ในตัวอ่านเทปความเร็วสูง และในออปโตไอโซเลเตอร์

ตัวกำเนิดแสง (Transmitter)

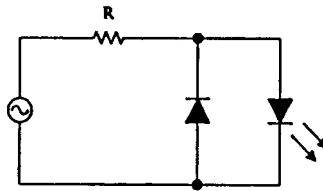
-LED (light emitting diode)

LED เป็นไดโอดที่สามารถเปล่งแสงได้ในขณะที่มันนำกระแส เนื่องจากเป็นไดโอดแบบหนึ่ง ดังนั้นการทำให้มันนำกระแสได้ก็ต้องให้ไบอัสเช่นเดียวกับไดโอดคือ ต่อไฟบวกเข้ากับแอโนด ไฟลบเข้ากับแคโทด แต่จะต้องต่อตัวต้านทานอนุกรมกับ LED ด้วยเพื่อจำกัดกระแสไม่ให้ไหลผ่าน LED มากเกินไปดังรูป 2.8



รูปที่ 2.8 วงจรใช้งาน LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 เมื่อใช้ LED กับแรงดันไฟสลับ

LED จะสามารถนำกระแสได้เมื่อมีแรงดันตกคร่อมตัวมันตั้งแต่ 2-5 V กระแสที่ไหลผ่านจะอยู่ระหว่าง 5-30 mA ไม่ควรเกินกว่านี้ เพราะอาจทำให้ LED เสียหาย ได้การเลือกค่าความต้านทานที่เหมาะสมให้แก่ LED เป็นไปตามสูตรดังนี้

$$R = (V_s - V_d) / I_d$$

โดยที่ V_s คือ แรงดันไฟเลี้ยง มีหน่วยเป็น โวลต์

V_d คือ ค่าแรงดันไบแอสตรง ที่ตกคร่อม LED แล้วทำให้มันทำงานมีหน่วยเป็น โวลต์

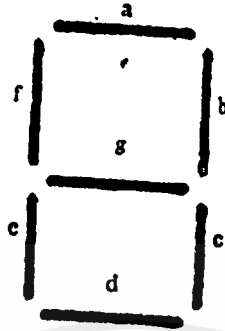
I_d คือ ปริมาณกระแสไฟตรง

LED เหมาะที่จะใช้กับวงจรไฟตรง แต่ถ้าหากต้องการนำไปใช้ในวงจรไฟสลับ ก็สามารถทำได้โดยจัดวงจรป้องกัน LED ไม่ให้เสียหาย เมื่อ LED ได้รับแรงดันไบแอสกลับ ด้วยการต่อไดโอดกลับขั้วขนานกับ LED ดังรูปที่ 2.9

LED มีให้เลือกใช้งานหลายแสงสี ที่มีในปัจจุบันได้แก่ สีแดง, เหลือง, เขียว, ส้ม และ อินฟราเรด นอกจากนั้น LED ยังสามารถใช้งานร่วมกับอุปกรณ์ตรวจจับแสง เช่น โฟโตเซลล์ ได้อีกด้วย

2.7 การเชื่อมต่อกับ LED 7 ส่วน

ต่อไปจะกล่าวถึงการเชื่อมต่อกับ LED ที่แสดงผลแบบตัวเลขได้ ที่เรียกว่า LED 7 ส่วน (7 - segment display) หรือ เจ็ดเซกเมนต์ ซึ่งมีทั้งแบบคาโทดร่วม (Common - cathode) และแบบอโนดร่วม (Common - anode) LED 7 ส่วนนี้จะเป็นการร่วม LED 7 หลอดประกอบกันให้สามารถแสดงเป็นตัวเลขได้ ถ้าเป็นแบบนำขาคาโทด ของ LED ทุกตัวมาร่วมกันเรียกว่าแบบคาโทดร่วม แบบอโนดร่วมก็ทำนองเดียวกัน แต่ละเซกเมนต์จะมีชื่อขาดังรูปที่ 2.10



รูปที่ 2.10 แสดงสัญลักษณ์ของแต่ละเซกเมนต์

ถ้าเราเชื่อมต่อแต่ละขา กับ Data Bus ของไมโครคอนโทรเลอร์โดยแต่ละ bit มีความสัมพันธ์กับขาดังตารางที่ 2-1 ถ้าหากต้องการให้ LED 7 ส่วน แสดงตัวเลขและตัวอักษรต่างๆ เราจะต้องส่งข้อมูล ให้แต่ละเซกเมนต์สว่างหรือดับให้ประกอบเป็นอักษรต่างๆ ดังนั้นข้อมูลที่ส่งไปที่พอร์ตจะเป็นตัวกำหนดตัวอักษรที่จะแสดงบน LED 7 ส่วน ตัวอักษรและค่าต่าง ๆ ที่ส่งออกมา มีความสัมพันธ์กันดังแสดงในตารางที่ 2-2

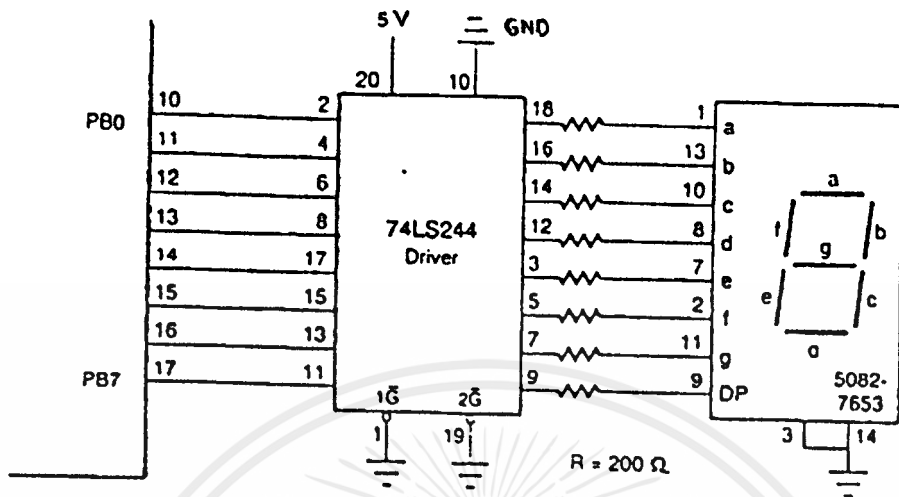
ตำแหน่งบิต	ตำแหน่งเซกเมนต์
7	-
6	g
5	f
4	e
3	d
2	c
1	b
0	a

ตารางที่ 2-1 แสดงตำแหน่งขาที่จะต่อกับพอร์ต

แสดงผล	อาโนคร่วม	คาโทคร่วม	แสดงผล	อาโนคร่วม	คาโทคร่วม
0	C0	3F	L	E1	1E
1	F9	06	O	C7	38
2	A4	5B	P	C0	3F
3	B0	4F	U	8C	73
4	99	66	Y	C1	3E
5	92	6D	b	99	66
6	82	7D	c	83	7C
7	F8	07	d	A7	58
8	80	7F	h	A1	5E
9	98	6F	n	88	74
A	88	77	o	AB	54
C	C6	39	r	A3	5C
E	86	79	u	AF	50
F	8E	71	-	E3	1C
G	82	70	?	BF	40
H	89	76	BLANK	AC	53
I	F9	06		FF	00

ตารางที่ 2-2 แสดงข้อมูลที่ส่งให้ LED 7 ส่วนแสดงเป็นเลขต่าง ๆ

พิจารณาวงจรตามรูปที่ 2.11 ถ้าหากต่อ LED 7 ส่วน กับพอร์ต A ของ 8255 ซึ่งถอดรหัสไว้ที่หมายเลข 0FC00H - 0FC03H โดยข้อมูลที่ส่งให้ LED 7 ส่วน จะส่งทางพอร์ต A ส่วนขาคาโทคร่วมจะต่อกับบิต PC0 ซึ่งมีเกตบัฟเฟอร์ที่เอาต์พุตเป็นแบบ OC ต่ออยู่ด้วย ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 แสดงการต่อ LED 7 ส่วน เข้ากับ MCS-51

ดังที่กล่าวมาแล้วการที่จะให้ LED 7 ส่วน แสดงเป็นตัวเลขต่าง ๆ จะต้องส่งข้อมูลออกมาให้ LED แต่ละส่วนสว่างเป็นเลขนั้น ในที่นี้จะเก็บข้อมูลที่จะแสดงตัวเลข 0 -9 ไว้ในตาราง และใช้การเขียนโปรแกรมแบบเปิดตาราง ขั้นแรกจะเป็นการโปรแกรมให้ MCS-51ทำงานเป็นอินพุตทุกพอร์ต จากนั้นหา DPTR เก็บค่าตำแหน่งของตารางเอาไว้ จากนั้นถ้าจะแสดงตัวเลขอะไรให้เก็บค่านั้นไว้ในรีจิสเตอร์ A แล้วเขียนโปรแกรมให้นำค่าที่อยู่ในตำแหน่งที่ DPTR ซ้ำอยู่บวกกับค่าใน A ซึ่งจะได้ข้อมูลที่จะทำ ให้ แสดงเป็นตัวเลขที่อยู่ใน A แล้วส่งค่าออกไปที่ LED 7 ส่วน ซึ่งสามารถเขียนโปรแกรมได้ดังนี้

**** LED 7 SEGMENT SHOW ****

```

PORT_A    EQU 0FC00H
PORT_B    EQU 0FC01H
PORT_C    EQU 0FC02H
PORT_CON  EQU 0FC03H

```

```

MAIN:  MOV  A,#80H
        MOV  DPTR,#PORT_CON
        MOVX @DPTR,A    ;โปรแกรมให้ทุกพอร์ตเป็นเอาท์พุท

SHOW :

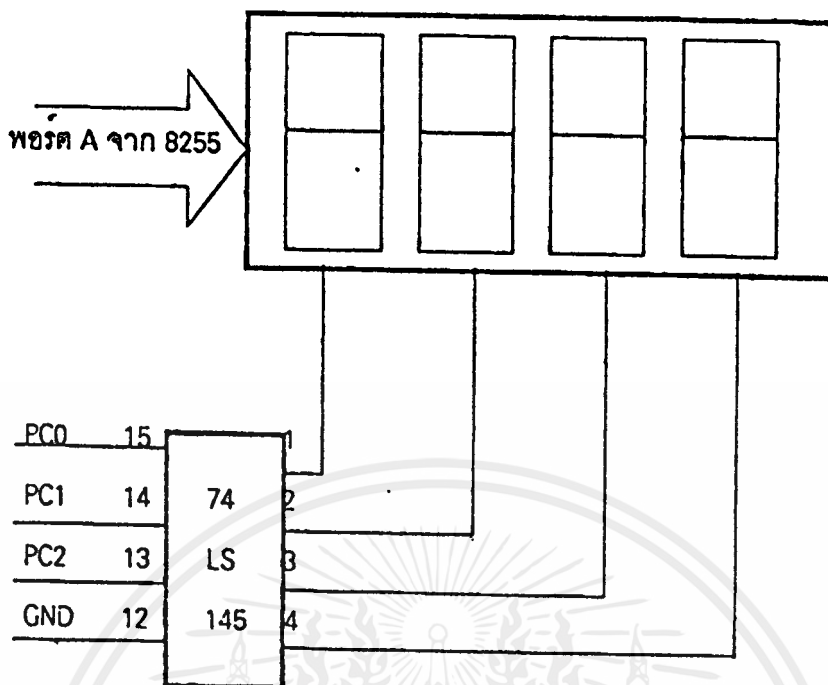
        MOV  DPTR,#SEG_CODE ; นำ DPTR  ที่ไปที่ตารางเก็บค่าแสดงผล
        MOV  A,#03          ; ใส่เลข 3  ไปใน A
        MOVC A,@A+DPTR     ; อ่านค่าในตารางช่องที่ 4 มาใส่ใน A
        MOV  DPTR,#PORT_A
        MOVX @DPTR,A       ; ส่งค่าออกไปที่ LED 7 ส่วน
        MOV  A,#00H        ; ทำให้ PC0 เป็น " 0 "
        MOV  DPTR,#PORT_C
        MOVX @DPTR,        ; LED 7 ส่วนสว่าง

SEG_CODE:
        DB 3FH,06H,58H,4FH,66H ; ตารางเก็บรูปแบบการแสดงผล
        DB 6DH,7DH,07H,7FH,6FH ; เรียงตามตัวเลข 0 ถึง 9

```

- การแสดงผลกับ LED 7 ส่วน หลาย ๆ ตัว

ระบบไมโครคอนโทรลเลอร์หากต้องการแสดงผลเป็นตัวเลขหลาย ๆ หลัก จำเป็นต้องใช้ LED 7 ส่วน หลาย ๆ ตัว จากที่กล่าวมาข้างต้น LED 7 ส่วน หนึ่งตัวจะต้องใช้พอร์ตขนาด 8 bit หนึ่งพอร์ต หากต้องการแสดงตัวเลข 3 หลัก จะต้องใช้พอร์ตถึง 3 พอร์ต ซึ่งจะเห็นว่าถ้าจะแสดงหลายหลักจะใช้พอร์ตเปลือง มีวิธีหนึ่งที่ จะแสดงผลหลายหลักได้ โดยจะประหยัดจำนวนพอร์ตและเป็นวิธีที่ใช้กัน โดยทั่ว ๆ ไป เรียกว่า Multiplexed displays โดยต่อขาแต่ละ เซกเมนต์เข้าด้วยกัน จากนั้นจะใช้วิธีสแกนให้ LED ติดทีละหลัก โดยการสแกนแต่ละหลัก จะต้องเร็วจนตาไม่สามารถมองการดับของ LED 7 ส่วนได้ทัน จนดูเหมือนว่า LED 7 ส่วนทุกตัวติดพร้อมกัน



รูปที่ 2.12 แสดงการต่อ LED 7 ส่วน 4 ตัว

จากรูปที่ 2.12 เป็นการต่อ LED 7 ส่วน 4 ตัวเข้ากับพอร์ตเพื่อที่จะแสดงผลแบบ multiplexed ในการต่อวงจรจะต่อพอร์ต A ของ MCS-51 เข้ากับทุกเซกเมนต์ของ LED 7 ส่วน จากนั้นจะใช้พอร์ต C เป็นตัวสแกนทางหลัก โดยใช้ไอซีเบอร์ 74LS145 เป็นตัวถอดรหัส ซึ่งไอซีเบอร์นี้สามารถต่อ LED 7 ส่วน ได้ถึง 10 ตัว และเอาท์พุทของไอซีเบอร์นี้จะเป็นแบบ open collector ซึ่งสามารถรับกระแสเข้าได้มากเมื่อ LED ทุกเซกเมนต์สว่างพร้อมกัน จากวงจรข้างต้นถ้าหาก MCS-51 ถอดรหัสพอร์ตไว้ที่หมายเลข 0FC00H-0FC03H เราจะเขียนโปรแกรมให้แสดงผลในหลักแรกและหลักที่สองได้ดังโปรแกรมข้างล่างนี้ ซึ่งจะแสดงตัวเลข 0 และ 1 โดยนำค่าตัวเลขที่จะแสดงผลใส่ใน R2 ที่ละค่าแล้วเขียนโปรแกรมให้ทำงาน จากนั้นให้โปรแกรมวนลูปแสดงผลให้ตาสามารถมองเห็นได้

ตัวอย่างโปรแกรมที่ 1

```

PORT_A    EQU    0FC00H
PORT_B    EQU    0FC01H
PORT_C    EQU    0FC02H

```

```

*****

```

```

**** โปรแกรมให้ MCS-51 ทำงาน ****

```

```

*****

```

```

MAIN:      MOV     A,#88H
           MOV     DPTR,#PORT_CON
           MOVX    @DPTR,A

DISPLAY:   MOV     R3,#09H

SHOW1:    MOV     R2,#00H      ; นำค่า 0 ใส่ใน R2 เพื่อที่จะแสดงเลข 0
           MOV     DPTR,#SEG_CODE
           MOV     A,R2
           MOVC   A,@A+DPTR   ; นำค่า 0 ไปเปิดตาราง
           MOV     DPTR,#PORT_A
           MOVX   @DPTR,A     ; ส่งข้อมูลออกไปที่พอร์ต A
           MOV     A,#00H
           MOV     DPTR,#PORT_C
           MOVX   @DPTR,A     ; ส่งข้อมูลออกไปที่พอร์ต C เพื่อให้หลักที่ 1 สว่าง
           CALL    DELA       ; เรียกโปรแกรมหน่วงเวลา

```

```

*****

```

```

**** สิ้นสุดการแสดงผลหลักที่ 1 ****

```

```

*****

```

```

SHOW:     MOV     R2,#01H      ; นำเลข 1 ใส่ไว้ใน R2 เพื่อจะแสดงผล
           MOV     DPTR,#SEG_CODE
           MOV     A,R2
           MOVC   A,@A+DPTR
           MOV     DPTR,#PORT_A
           MOVX   @DPTR,A

```

```

MOV   DPTR,#PORT_C
MOVX  @DPTR,A
CALL  DELAY           ; เรียกโปรแกรมย่อย

```

***** สิ้นสุดการแสดงผลหลักที่ 2 *****

```

SJMP  DISPLAY       ; วนรอบใหม่เพื่อให้มองเห็น

```

SEG_CODE:

```

DB    3FH,06H,5BH,4FH,66H

```

```

DB    6DH,7DH,07H,7FH,6FH

```

***** โปรแกรมย่อยหน่วงเวลา *****

```

DELAY:  MOV   R1,#0FFH
DELAY1: DJNZ  R1,DELAY1
        RET

```

บทที่ 8

การวิจัยและการดำเนินงาน

ขั้นตอนการดำเนินงานโครงการพิเศษนี้สามารถแสดงให้เห็นดังแผนภาพ



รูปที่ 3.1 แสดงแผนภาพการดำเนินงาน

จากแผนภาพแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

1. ศึกษา ค้นคว้า และเลือกใช้ให้เหมาะสม ทำการศึกษา ค้นคว้าหาข้อมูลเกี่ยวกับอุปกรณ์ที่นำมา ใช้ในการทำโครงการพิเศษนี้ ได้แก่
 - การเลือกใช้ตัวตรวจจับทางแสง (sensor) ให้เหมาะสมตามที่ต้องการ ในที่นี้เลือกใช้ LED และ photodiode ชนิด Infrared ซึ่งเหตุผลที่เลือกใช้มีดังนี้ คือ
 1. หาซื้อง่าย และง่ายต่อการใช้งาน
 2. ไม่ต้องการให้มีแสงเล็ดลอดออกมาภายนอกเหมือน LED โดยทั่วไป ซึ่งอาจทำให้เกิดการหักเหรบกวนจากแสงธรรมชาติได้
 - การเลือกใช้ไมโครคอนโทรลเลอร์ให้เหมาะสม มีเหตุผลดังนี้
 1. มีรูปร่างภายนอกที่เหมาะสม กล่าวคือ มีขนาดเล็ก แต่มีฟังก์ชันมากมาย ทำให้ประหยัดเนื้อที่ ในการลงอุปกรณ์
 2. สามารถประมวลผลได้อย่างรวดเร็ว เพราะมี RAM และ ROM อยู่ภายในตัว
 3. มีฟังก์ชันที่พอเพียง เช่น Port interrupt เป็นต้น
- พร้อมกันนั้น ก็ทำการศึกษา ข้อจำกัด ตลอดจนการนำไปใช้งาน ในการทดลองได้อย่างมีประสิทธิภาพ

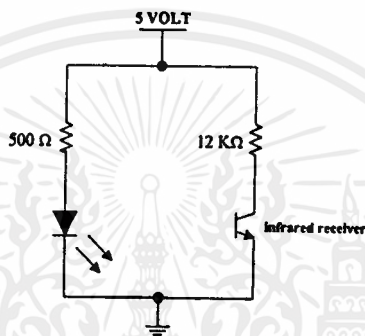
เมื่อทำการศึกษาในขั้นตอนนี้เสร็จเรียบร้อยแล้ว ต่อไปก็จะทำขั้นตอนการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2, การทดลอง

เมื่อเลือกใช้อุปกรณ์ที่เหมาะสมได้แล้ว ก่อนที่ทำการทดลองจะอธิบายส่วนต่างๆ ของวงจรโดยรวม ทั้งหมดของโครงงานพิเศษนี้ ดังรายละเอียดต่อไปนี้ คือ

1. ส่วน heading sensor



รูปที่ 3.2 แสดงวงจรตัวตรวจจับทางแสง (sensor)

ในส่วนนี้ได้ใช้ LED ย่าน Infrared เป็นตัวส่งแสงและใช้ infrared receiver เป็นตัวรับแสง จากรูปที่ 3.2 วงจรดังกล่าวมีหลักการทำงานดังต่อไปนี้

1. เมื่อมีแสงจาก LED ตกกระทบตัวรับ ก็จะทำให้ แรงดันเอาต์พุตที่ได้ มีค่าประมาณ 0 โวลต์ ซึ่งจะเป็นสถานะลอจิก "0"
2. แต่ถ้าไม่มีแสงมาตกกระทบกับ infrared receiver เช่น กรณีที่มีการตัดแสงเกิดขึ้น อันเนื่องมาจาก หดน้ำเกลือ หรือการใช้กระดาษเล็กๆ ตัดแสง จะพบว่าแรงดันเอาต์พุตที่ได้มีค่าประมาณ 4.26 โวลต์ ซึ่งเปรียบได้กับสถานะลอจิก "1"
3. เมื่อมีการสลับกันไปมาระหว่างลอจิก "1" กับ ลอจิก "0" อย่างรวดเร็ว จะเกิดเป็นพัลส์ ที่ได้นี้ไปควบคุมไมโครคอนโทรลเลอร์ต่อไป

จากหลักการข้างต้น ในการทดลองต้องคำนึงถึงสิ่งต่อไปนี้ คือ

1. การจัดตำแหน่งของตัวรับตัวส่ง ต้องอยู่ในตำแหน่งที่ตรงกัน มากที่สุดเพื่อจะได้ผลการทดลองที่ถูกต้องแม่นยำ กล่าวคือ แรงดันเอาต์พุตที่ได้จะมีค่าเป็น 0 โวลต์ หรือ 5 โวลต์ ซึ่งเป็นสถานะลอจิกที่แน่นอน

หมายเหตุ ในทางปฏิบัติแล้ว แรงดันที่ได้อยู่ในช่วง 0.5-4.26 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นได้ว่า

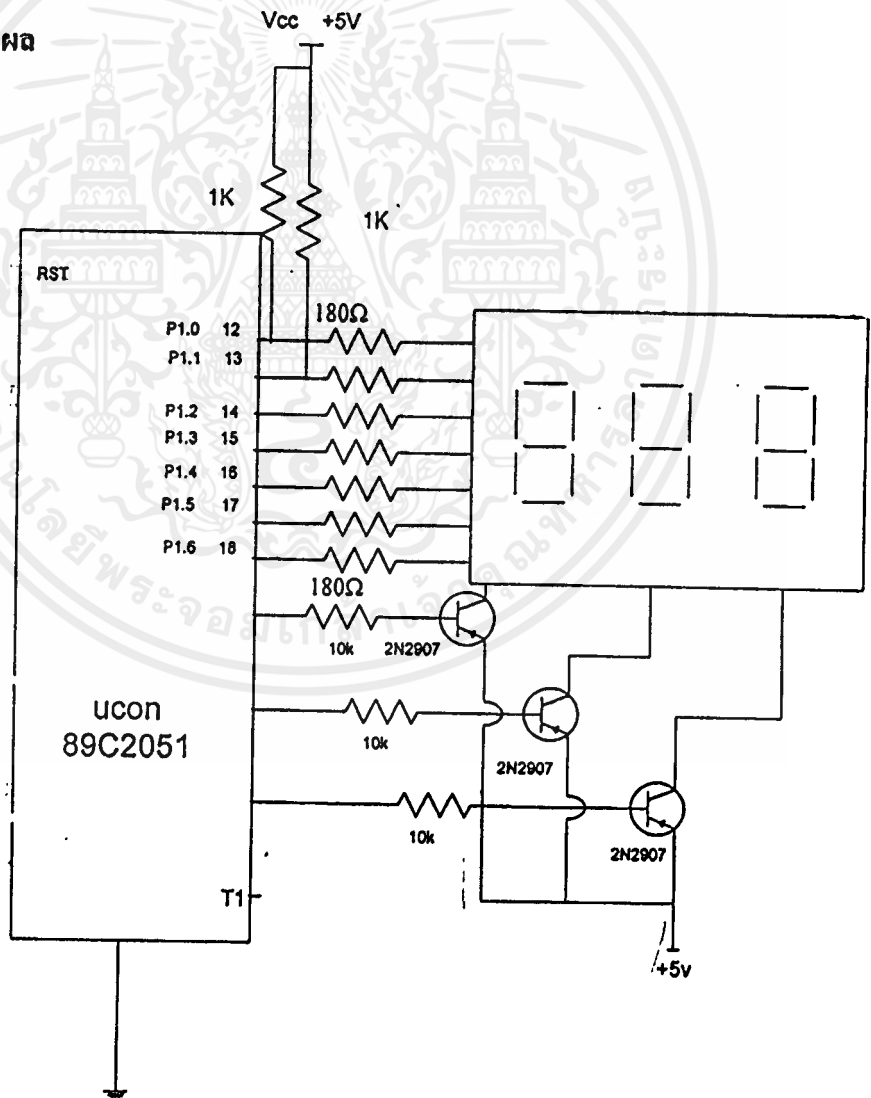
- วงจร Regulator ใช้ทำหน้าที่แปลงแรงดันที่มีค่ามากกว่า 5 โวลต์ ให้เป็นแรงดัน 5 โวลต์ ซึ่งจำเป็นสำหรับ 89C2051 โดยใช้ Regulator 7805 มีลักษณะ 3 ขา ขา input คือ ขาที่ 1, output คือ ขาที่ 3 และกราวด์เป็นขาที่ 2 ซึ่งขาที่ 3 นี้ ทำหน้าที่เป็นไฟเลี้ยงของวงจรทั้งหมด

- 74LS14 เป็น NOT GATE ประเภท open collector ทำหน้าที่กลับสัญญาณพัลส์ ที่เข้ามาให้เป็นพัลส์ขาขึ้น

- Crystal Oscillator ที่ต่อบริเวณขา 4 และ ขา 5 ของ 89C2051 ทำหน้าที่ป้องกันความถี่ให้กับส่วนประมวลผลใช้ค่า 11.0592 MHz

- สวิตช์ RESET อยู่ที่ขา 1 ของ 89C2051 และสวิตช์อินเทอร์รัพต์จะอินเทอร์รัพต์ที่ขา 7 ของ 89C2051

3. ส่วนแสดงผล



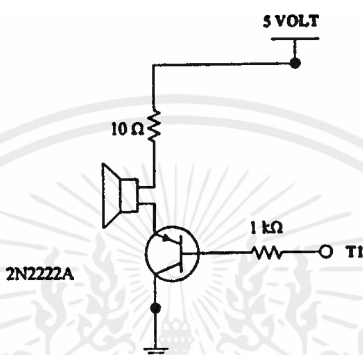
รูปที่ 3.4 แสดงผลของวงจรทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งออกเป็นารแสดงผล 2 ส่วน

- ทาง LED 7 ส่วน จะแสดงผลเป็นจำนวนหยดค่อนาที โดยการที่ทำให้ LED 7 ส่วนแต่ละตัวติดสว่าง นั้นจะใช้ pnp Transistor 2N2907 เป็นตัวขับ สามารถทดสอบโดยการเขียนโปรแกรมภาษาแอสเซมบลี ประมวลผลเป็นจำนวนหยดค่อนาที

- แสดงผลทางลำโพง เป็นเสียงที่ดังขึ้นจากลำโพงขนาด 10 โอห์ม ดังรูปที่ 3.5



รูปที่ 3.5 แสดงวงจรขับลำโพง

จากรูปที่ 3.5 จะเห็นได้ว่าใช้ Transistor npn 2N2222 ขับให้ลำโพงมีเสียงดังเกิดขึ้น และจะนำขาข้างหนึ่งของลำโพงต่อเข้ากับแรงดัน 5 โวลต์ ส่วนขาอีกข้างหนึ่งของลำโพงต่อเข้ากับขา 19 หรือ P1.7 ของ 89C2051 เพื่อควบคุมการเกิดเสียงของลำโพง รายละเอียดการส่งเสียงเตือน เมื่อน้ำเกลือจะหมดขวด จะกล่าวถึงหัวข้อการประยุกต์ใช้ต่อไป

4. ประมวลผลและแสดงผล

เพื่อแน่ใจว่าสัญญาณพัลส์ที่ได้เป็นสัญญาณที่ถูกต้อง (ที่เกิดจากการที่หยดน้ำเกลือหยดตัด ผ่านตัวตรวจจับทางแสงทุกหยด) ทำการตรวจสอบหลายๆครั้ง โดยใช้ logic probe จับที่ขา 1 และ ขา 2 ของ 74LS14 และ ขา 6 ของ 89C2051 พบว่าเมื่อน้ำเกลือหยดตัดแสง logic probe จะแสดงสถานะสลับกันไปมาระหว่าง logic high กับ logic low ดังนั้นจึงนำสัญญาณดังกล่าวนี้ เข้าสู่ส่วนประมวลผล

ทำการตรวจสอบส่วนของวงจรทั้งหมดโดยการเขียนโปรแกรมภาษาแอสเซมบลี ตรวจสอบวงจรทีละส่วนก่อน ดังต่อไปนี้

1. วงจรควบคุมการนับจำนวนหยดน้ำเกลือ

โดยการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ 89C2051 นับจำนวนหยดน้ำเกลือทุกหยดที่ตัดผ่าน แล้วแสดงผลออกทาง 7-segment เป็นเลขฐานสิบ ถ้าพบว่าไมโคร

คอนโทรลเลอร์ไม่สามารถนับได้แสดงว่า แรงดันเอาต์พุตที่ตรวจจับทางแสงไม่สามารถทำให้เกิด ลอจิก "0" หรือ "1" ได้

2. วงจรทั้งหมด คือ การเขียนโปรแกรมประมวลผล เป็นจำนวนหยดต่อ 1 นาที ซึ่ง สามารถเปลี่ยนไปตามการปรับอัตราน้ำเกลือของผู้ใช้ เช่น ให้หยดช้าลง หยดเร็วขึ้น อัตราเร็วของ หยดน้ำเกลือก็จะต่างกัน โดยการเขียนโปรแกรมมีหลัก ดังนี้

น้ำเกลือหยดที่ 1 ที่ตัดผ่านแสง ไมโครคอนโทรลเลอร์จับเวลา t_1
 " ----- " 2 " ----- " t_2

∴ จะมีน้ำเกลือหยดผ่านใน 1 นาที = $60 / (t_2 - t_1)$ จำนวนหยด/นาที

ในทางปฏิบัติมีความคลาดเคลื่อนเกิดขึ้นได้ เช่น เมื่อไมโครคอนโทรลเลอร์ทำตาม คำสั่งการแล้วเกิดเหลือเศษของผลหารขึ้น ไมโครคอนโทรลเลอร์จะไม่นำเศษของการหาร มา ประมวลผลอีก สาเหตุสำคัญอีกประการคือไมโครคอนโทรลเลอร์จะประมวลผลออกมาเป็น ทศนิยม 1 ตำแหน่งเท่านั้น สามารถแก้ไขได้โดยพัฒนาโปรแกรมให้ตัดขึ้นต่อไป

เมื่อทำการประมวลผล โดยทุกๆ ไปของวงจรทั้งหมดแล้ว ต่อไปก็จะกล่าวถึงส่วนของ การประยุกต์ใช้

5. การประยุกต์ใช้

การประยุกต์ใช้โดยรวมจะเป็นในลักษณะให้มีการส่งเสียงดังตามที่ต้องการ ดังนี้

- ส่งเสียงเตือนเมื่อน้ำเกลือใกล้จะหมดขวด

ในที่นี้จะให้มีการส่งเสียงเตือน เมื่อน้ำเกลือในขวดมีปริมาณ = 26.7 มล. มีหลัก ในการพิจารณาดังนี้ คือ

เมื่อทำการทดลองแล้ว หากค่าเฉลี่ยในการนับจำนวนหยดน้ำเกลือที่หยดลงมาทั้งหมด 500 หยด จำนวน 5 ครั้งจะได้ค่าเฉลี่ย = 26.7 มล.

ดังนั้น ถ้าน้ำเกลือปริมาตร 1000 มล.จะได้จำนวนหยดทั้งหมด = 18726.59 หยด ซึ่งใน โครงการพิเศษนี้จะให้ Infrared sensor นับจำนวนหยดน้ำเกลือไป 18716 หยด แล้วเกิดเสียง เตือนดังขึ้น ดังนั้น 18716 หยด เมื่อแปลงเป็นเลขฐานสิบหก จะได้ = 419C หยด

เมื่อไมโครคอนโทรลเลอร์นับจำนวนหยดจนถึงจำนวน 18716 หยด แล้ว ไมโคร คอนโทรลเลอร์ก็จะทำตามโปรแกรมตอบสนองการอินเทอร์รัพต์ โดยให้มีเสียงดังเกิดขึ้น

ดังนั้น เมื่อพิจารณาถึงวงจรโดยรวมทั้งหมดแล้ว สามารถอธิบายการทำงานจากรูปวงจร ทั้งหมด (ในภาคผนวก) ได้ดังนี้คือ

1. เมื่อหยดน้ำเกลือหยดแรกตัดผ่านตัวจับแสง ก็จะเป็นพัลส์ขาลงผ่าน 74LS14 และออกมาเป็นพัลส์ขาขึ้นเข้าสู่ตัวไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์จะทำหน้าที่ เป็นตัว timer จับเวลาไว้ (t_1)

2. เมื่อหยดน้ำเกลือหยดที่ 2 ตัดผ่านตัวตรวจจับทางแสง ไมโครคอนโทรลเลอร์ก็จะทำหน้าที่เป็นตัว timer อีกเช่นกัน (t_2)

3. ไมโครคอนโทรลเลอร์ทำการประมวลผลเป็นจำนวนหยดต่อ 1 นาที

4. เมื่อผู้ใช้งาน (แพทย์, พยาบาล) ปรับน้ำเกลือให้ได้อัตราตามที่ต้องการแล้ว (ตั้งแสดงผลทาง 7-segment)

5. ทำการกดปุ่มอินเทอร์รัพต์ เมื่อกดปุ่มนี้จะอินเทอร์รัพต์ที่ P3.5 ของไมโครคอนโทรลเลอร์ ทำให้ไมโครคอนโทรลเลอร์ทำหน้าที่เป็น counter เริ่มนับจนกระทั่งครบจำนวนหยด ตามที่ได้เขียนโปรแกรมไว้

6. ไมโครคอนโทรลเลอร์ทำงานตอบสนองโปรแกรมอินเทอร์รัพต์เกิดเสียงดังขึ้นที่ลำโพงทั้งสองตัวพร้อมกัน

6. ส่วนของโปรแกรม

สำหรับโครงการพิเศษนี้ใช้โปรแกรมภาษาแอสเซมบลีในการควบคุมส่วนต่างๆ สรุปได้ดังนี้

6.1 ส่วนประมวลผลเป็นอัตราจำนวนหยดต่อนาที แสดงออกทาง 7-segment

6.2 ส่วนโปรแกรมการนับจำนวนหยด

6.3 ส่วนโปรแกรมการส่งเสียงเตือนออกทางลำโพง

ซึ่งส่วนโปรแกรมทั้งหมดแสดงได้ดังภาคผนวก

บทที่ 4

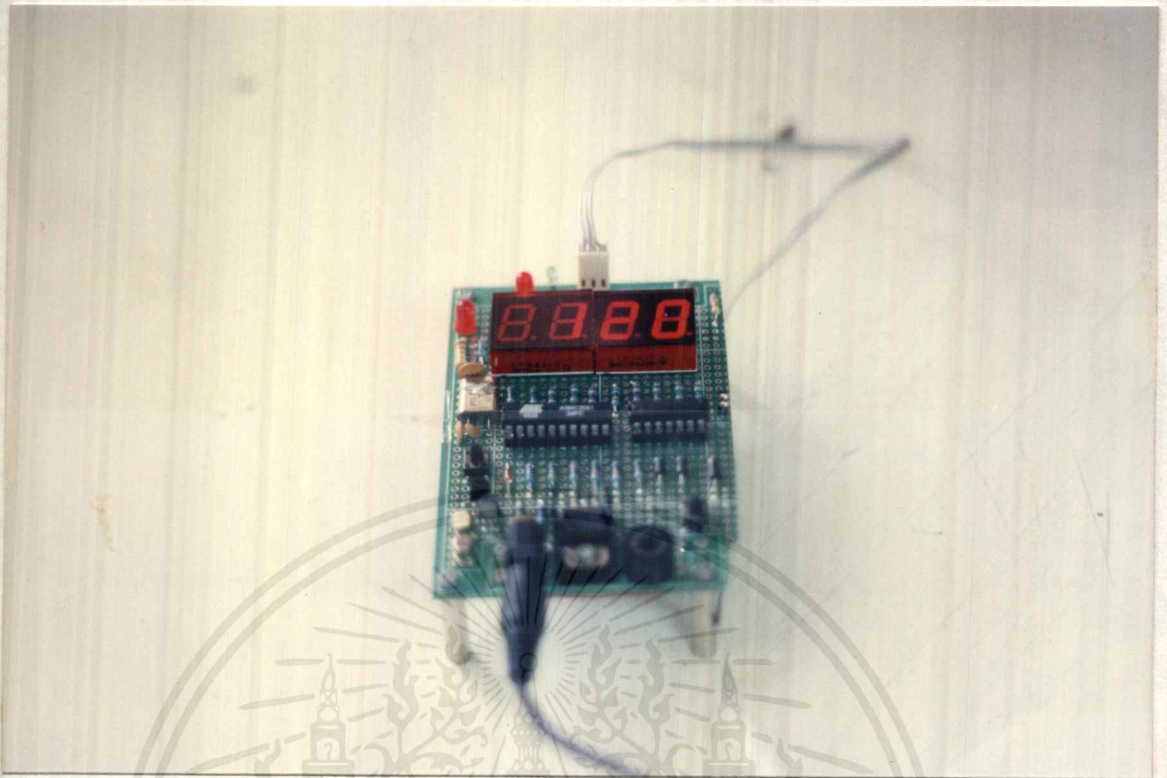
ผลการศึกษาจากการดำเนินงาน

ในบทนี้จะกล่าวถึงผลของการสร้างเครื่องนับจำนวนหยดน้ำเกลือ
เครื่องนับจำนวนหยดน้ำเกลือที่สร้างขึ้นมีลักษณะดังแสดงในรูป 4.1



รูปที่ 4.1ก เครื่องนับจำนวนหยดน้ำเกลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1x เครื่องนับจำนวนหยดน้ำเกลือ

จากรูปที่ 4.1 และ รูปที่ 4.2 มีการดำเนินงานอย่างละเอียดต่อไปนี้ คือ

1. นำชุดน้ำเกลือ ซึ่งประกอบไปด้วยถุงน้ำเกลือและชุดเข็มที่เป็นส่วนกระเปาะน้ำเกลือ นำเข็มดังกล่าวแทงเข้าไปที่ปลายของถุงน้ำเกลือ พร้อมกันนั้นก็นำเข็มปลายแหลม(ที่มีลักษณะเหมือนเข็มฉีดยา) ทิ่มลงในบริเวณสูญญากาศของน้ำเกลือ ขณะคว่ำขวดน้ำเกลือแล้ว
2. ทำการหมุนปรับ roller clamp ดังรูป ให้น้ำเกลือหยดลงมาอยู่ที่กระเปาะของน้ำเกลือเพียงเล็กน้อย



รูปที่ 4.2 แสดงชุดของเข็มน้ำเกลือ

3. นำส่วนของตัวตรวจจับทางแสง(ที่ได้ ออกแบบไว้แล้ว) หนีบเข้ากับกระเปาะน้ำเกลือ แล้วต่อเข้ากับส่วนประมวลผล
4. ทำการหมุน roller clamp อีกครั้งหนึ่ง ขณะที่ทำการหมุน ตัวเลขที่ปรากฏบน LED 7 ส่วนจะเปลี่ยนไปเรื่อยๆ(เมื่อหมุนหยดน้ำเกลือความเร็วแตกต่างกัน) จนกระทั่งได้อัตราตามต้องการแล้ว ก็หยุดปรับ
5. กดปุ่ม สวิตช์อินเทอร์รัพท์ เพื่อให้ 89C2051 ทำหน้าที่เป็น counter นับจำนวนหยดน้ำเกลือทุกหยดที่ตัดผ่านแสงจนกระทั่งได้จำนวนหยดดังที่ได้โปรแกรมไว้ ก็จะมีเสียงเตือนเกิดขึ้นที่ลำโพง ซึ่งแสดงว่าน้ำเกลือกำลังจะหมดขวดแล้ว

ผลการทดลอง

เครื่องนับจำนวนหยดน้ำเกลือที่จัดทำขึ้นนี้ใช้ได้ในกรณีเฉพาะเท่านั้น กล่าวคือในปัจจุบันมีขวดน้ำเกลือที่ใช้และเข็มหลากหลายขนาดมาก ดังนั้นจะใช้กับขวดน้ำเกลือที่มีขนาดใหญ่และปริมาณน้ำเกลือต่างๆ ที่เหมาะสมจะอยู่ในความควบคุมของผู้เชี่ยวชาญ ดังมีหลักเกณฑ์ในทฤษฎีบทที่ 2 การแสดงผลการทดลอง จึงขึ้นอยู่กับการนำไปใช้งานจริงนั่นเอง

อย่างไรก็ตาม ผู้จัดทำได้ทำการทดลองแสดงผลได้ดังตารางข้างล่างนี้

เครื่องนับแสดงผล 84 หยด/นาที

ครั้งที่	อัตรา หยด/นาที
1	108
2	112
3	108
4	110
5	110

เครื่องแสดงผล 110 หยด/นาที

ครั้งที่	อัตรา หยด/นาที
1	100
2	120
3	90
4	130
5	110

□ เครื่องแสดงผล 130 หยค/นาทึ

ครั้งที่	อัตราหยค/นาทึ
1	120
2	110
3	120
4	120
5	130

ตารางที่ 4-1 แสดงผลการทดลอง

เมื่อนำมาจับเวลาเปรียบเทียบกัน พบว่ายังมีความคลาดเคลื่อนอยู่บ้าง ดังนั้นจึงต้องทำการพัฒนาโปรแกรมให้ดียิ่งขึ้นต่อไป

บทที่ 5

สรุปผลการวิจัย

โครงการพิเศษเครื่องนับจำนวนหยดน้ำเกลือนี้ จัดทำขึ้นเพื่ออำนวยความสะดวกแก่แพทย์และพยาบาลตลอดจนทุกสาขาวิชาที่เกี่ยวข้อง ซึ่งจะพบว่าจะยังมีข้อจำกัดอยู่บ้าง ดังที่ได้กล่าวมาแล้ว แต่อย่างไรก็ตามโครงการนี้ก็ยังมีข้อแตกต่างบางประการเมื่อเทียบกับเครื่องที่มีผู้ประดิษฐ์ไว้แล้วและเครื่องที่สั่งซื้อจากต่างประเทศ ได้แก่

1. มีน้ำหนักเบากว่า
2. มีเสียงเตือนเมื่อน้ำเกลือใกล้จะหมดขวด
3. ต้นทุนต่ำ
4. ใช้งานง่าย

อย่างไรก็ตามโครงการนี้ก็ยังไม่สามารถใช้งานได้สมบูรณ์แบบ อันเนื่องมาจากสาเหตุหลายประการ เช่น

- ใช้งานได้เข็มได้ขนาดเดียวเท่านั้น

- ไม่สามารถควบคุมหยดน้ำเกลือได้ กล่าวคือ ไม่สามารถควบคุมกลไกทาง

กลศาสตร์ของไหลในสายน้ำเกลือให้บังคับจำนวนหยดที่หยดออกมาเป็นอัตราเร็วตามที่ต้องการได้

- ปัญหาอีกประการหนึ่งก็คือ ถ้าตัวรับ-ส่งอินฟราเรดไม่อยู่ในตำแหน่งที่ตรงกันพอดี ก็จะทำให้ไมโครคอนโทรลเลอร์ไม่สามารถประมวลผลได้

ดังนั้นจึงต้องมีการพัฒนาเครื่องให้เหมาะสมต่อไป ได้แก่

- พัฒนาให้มีการส่งรหัสจากเตียงของผู้ป่วยแต่ละเตียง เมื่อน้ำเกลือจะหมดขวด ทำ

ให้พยาบาลหรือแพทย์ทราบว่า เตียงผู้ป่วยใดที่น้ำเกลือจะหมดขวด

- พัฒนาโปรแกรมให้ดียิ่งขึ้น

- พัฒนาให้มีการควบคุมกลไกของหยดน้ำเกลือได้

- พัฒนารูปแบบของอุปกรณ์ภายนอก เช่น กล่องใส่วงจร ตัวขีดตัวตรวจจับทางแสง

สวิตซ์ต่างๆ ให้เหมาะสมยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

สรุปคำสั่งของ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ในการควบคุมอุปกรณ์ภายนอก จะต้องมีการส่งข้อมูลออก และรับข้อมูลเข้าระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การส่งและรับข้อมูลนั้นจะเป็นการถ่ายเทข้อมูลระหว่างรีจิสเตอร์กับตำแหน่งของพอร์ต (หน่วยความจำของพอร์ตนั้น) จากการศึกษาเรื่อง MCS-51 พบว่าคำสั่งในการย้ายข้อมูลมีอยู่ 3 ชุด คือ

- MOV เป็นการย้ายข้อมูลระหว่างหน่วยความจำภายใน
- MOVC เป็นการย้ายข้อมูลระหว่างหน่วยความจำโปรแกรมกับ MCS-51
- MOVX เป็นการย้ายข้อมูลระหว่างหน่วยความจำภายนอกกับ MCS-51

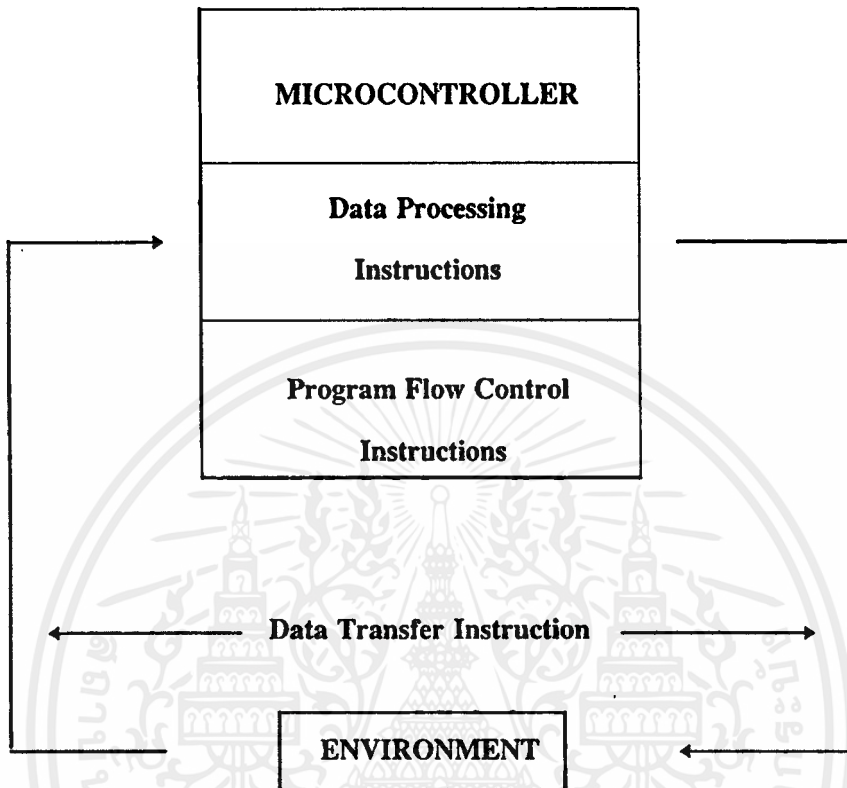
นอกจากนี้การทำงานของ MCS-51 อาจมีการประมวลผลข้อมูลด้วยคำสั่งให้ MCS-51 ทำงานจะต้องมีชุดคำสั่งของมัน และเราต้องแปลงชุดคำสั่งเหล่านี้ให้เป็นรหัสคำสั่งจากนั้นบรรจุลงไป ในหน่วยความจำโปรแกรม MCS-51 จึงทำงานได้ ซึ่งชุดคำสั่งแต่ละคำสั่งนำมาเรียงกันจะได้อำนาจภาษาแอสเซมบลี การแปลงภาษาแอสเซมบลีให้เป็นรหัสคำสั่งจะทำได้สองวิธีคือ วิธีแรกใช้วิธีเปิดคำสั่งชุดตารางคำสั่งของ MCS-51 อีกวิธีหนึ่งจะใช้คอมพิวเตอร์ในการแปลง

ระบบไมโครคอนโทรลเลอร์ถ้าหากต้องการติดต่อกับอุปกรณ์ภายนอกอาจเขียนเป็นไคอะแกรมได้ดังรูปที่ 1 โดยขณะที่ MCS-51 ทำงานจะต้องมีคำสั่งควบคุมโดย MCS-51 จะอ่านรหัสคำสั่งที่เรียกว่า Fetch cycle โดยอ่านจากตำแหน่งที่ PC ซ้ำอยู่ จากนั้นเพิ่มค่า PC ขึ้นอีก 1 และกระทำตามคำสั่งนั้นซึ่งเขียนเป็นไคอะแกรมดังรูป

ถ้าหากเขียนคำสั่งว่า MOV, R1 01 MOV R2, #02 02 R2 MCS-51

- Rn หมายถึงติดต่อกับรีจิสเตอร์ 8 ตัวใน bank ที่ active อยู่โดย n มีค่าเท่ากับ 0
- direct หมายถึงตำแหน่งข้อมูลของ RAM ภายในขนาด 8 บิต
- @Ri หมายถึงการอ้างตำแหน่งที่เก็บอยู่ใน R0 หรือ R1
- #data หมายถึงข้อมูลขนาด 8 บิต
- #data16 หมายถึงข้อมูลขนาด 16 บิต
- addr16 หมายถึงการอ้างตำแหน่งโดยอ้างแบบ 16 บิต ซึ่งทำได้ทั้งหมด 64 K
- addr11 หมายถึง การอ้างตำแหน่งแบบ 11 บิต ซึ่งจะอ้างได้ 2 K
- rel หมายถึงการอ้างแบบ relative โดยใช้ตัวเลขแบบ 2's complement โดยอ้างไปข้างหน้าได้ -128 ตำแหน่ง และอ้างไปข้างหน้าได้ 127 ตำแหน่ง จากตำแหน่งที่ PC ซ้ำอยู่

- bit หมายถึงการอ้างตำแหน่งแบบบิต



รูปที่ 1 การติดต่อระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก

ชุดคำสั่งในการย้ายข้อมูล

กลุ่มคำสั่งในการย้ายข้อมูลของ MCS-51 แสดงได้ดังตาราง โดย byte หมายถึงการกำหนดที่อยู่ (addressing mode) แบบต่างๆ คำสั่งในการโอนย้ายข้อมูลนั้นจะเป็นการกระทำระหว่างรีจิสเตอร์กับรีจิสเตอร์ หรือรีจิสเตอร์กับหน่วยความจำ มีรูปแบบเป็น destination ,source หรือ source,destination

ตัวอย่าง เช่น MOV R3,32H

เป็นการย้ายข้อมูลของหน่วยความจำข้อมูลภายในตำแหน่งที่ 32H มาเก็บไว้ในรีจิสเตอร์ที่ 3 ซึ่งจะเห็นว่า source จะเป็นตำแหน่งของหน่วยความจำโดยตรง

□ The immediate Constant Addressing Mode

การอ้างตำแหน่งนี้เป็นการอ้างข้อมูลที่ source เป็นค่าคงที่โดยตรง โดยจะใช้เครื่องหมาย # นำหน้าเพื่อบอกว่าตัวเลขที่ตามหลังมาเป็นค่าคงที่ เช่น

```

MOV R4,#3
MOV R5,#2
MOV R6,#1
MOV R7,#0
LJMP 0

```

ถ้าหากเป็นการย้ายข้อมูลแบบ 16 บิตด้วย เช่น การเขียนคำสั่ง MOV DPTR ,#1234H การเก็บข้อมูลแบบนี้จะเก็บค่า 12H ในรีจิสเตอร์ DPH และ 34H ใน DPL

□ The direct Addressing Mode

เป็นการอ้างตำแหน่งที่อยู่ของข้อมูลโดยตรง ซึ่งจะย้ายข้อมูลระหว่างรีจิสเตอร์กับตำแหน่งของ RAM ภายในรวมทั้งตำแหน่งของ Special Function Register (SFR) ด้วยเช่นถ้าหากเขียนข้อมูลไปที่ตำแหน่ง 90H ซึ่งเป็นตำแหน่งของพอร์ต 1 ดังนั้นการเขียนข้อมูลไปที่ตำแหน่ง 90H จะเป็นการส่งข้อมูลออกไปที่พอร์ต 1 เช่น

```

MOV A,#70H      อ่านข้อมูลจากตำแหน่ง 70H ของ RAM มาเก็บใน A
MOV A,#0        clear ค่าในรีจิสเตอร์ A
MOV 90H,A       นำค่าใน A ส่งไปที่พอร์ต 1

```

ถ้าหากจะออกแบบระบบโดยต่อ LED กับพอร์ต 1 และพอร์ต 4 ต่อกับสวิทช์ และถ้าจะเขียนโปรแกรมให้อ่านค่าจากสวิทช์ แล้วแสดงผลทาง LED จะเขียนโปรแกรมได้ ดังนี้

```

ORG 8000H
MOV A,0E8H      อ่านค่าจากตำแหน่ง 0E8H มาเก็บใน A
MOV 90H,A       นำค่าจาก A ส่งไปที่พอร์ต 1
LJMP 8000H      กระโดดไปที่จุดเริ่มต้น

```

□ The Register Addressing Mode

การอ้างตำแหน่งแบบนี้เป็นการใช้รีจิสเตอร์ในการอ้างตำแหน่ง โดย source และ destination จะเป็นรีจิสเตอร์ เช่น ตัวอย่างต่อไปนี้ จะเป็นการติดต่อกับรีจิสเตอร์ในแบงค์ 2 ซึ่งจะนำข้อมูลใน A ไปเก็บใน R0 และนำค่า B ไปเก็บใน R1

```

ORG 8000H
MOV PSW,#10H   เลือกค่าในรีจิสเตอร์ bank 2
MOV R0,A       นำค่าใน A ไปเก็บใน R0
MOV R1,B       นำค่าใน B ไปเก็บใน R1
LJMP 0

```

```

ORG 8000H
MOV PSW,#18H      คิดต่อกับรีจิสเตอร์ bank 3
MOV A,R0          นำค่า R0 ไปเก็บในรีจิสเตอร์ A
MOV R1,A          นำค่าในรีจิสเตอร์ A ไปเก็บใน R1
LJMP 0

```

□ The Register-Specific Addressing Mode

เป็นการใช้รีจิสเตอร์พิเศษในการอ้างตำแหน่ง เช่น ถ้าหากเขียนข้อมูลลงไปในรีจิสเตอร์ A จะเท่ากับการเขียนข้อมูลลงไปตำแหน่ง 0E0H พิจารณาคำสั่ง MOV สองคำสั่งต่อไปนี้

```

MOV A,#01          นำค่า 1 ไปเก็บใน A
MOV 0E0H,#01      นำค่า 1 ไปเก็บตำแหน่ง 0E0H

```

คำสั่ง MOV ทั้งสองคำสั่ง ผลที่ได้เท่ากันเพราะรีจิสเตอร์ A จะอยู่ที่ตำแหน่ง 0E0H แต่ถ้าหากแปลงเป็นภาษาเครื่องแล้วผลที่ได้จะไม่เหมือนกัน คือคำสั่ง MOV A,#01 จะมีขนาด 2 ไบท์ คือ 74 และ 01 แต่ถ้าเป็น MOV 0E0H,#01E0 จะมีขนาด 3 ไบท์ คือ 75 E0 และ 01 ซึ่งจะให้ความยาวของรหัสคำสั่งไม่เท่ากัน

□ The Register Indirect Addressing Mode

การอ้างตำแหน่งแบบนี้จะใช้รีจิสเตอร์เก็บค่าตำแหน่งที่จะติดต่อเอาไว้ หรืออาจกล่าวได้ว่าค่าตำแหน่งของ source หรือ destination จะเก็บไว้ในรีจิสเตอร์ โดยรีจิสเตอร์ที่ใช้มีสองตัวคือ R0 และ R1 พิจารณาดังตัวอย่างต่อไปนี้

```

ORG 8000H
MOV PSW,#0        เลือกรีจิสเตอร์ bank 0
MOV R0,#78H      เก็บค่า 78H ลงในรีจิสเตอร์ 0
MOV @R0,#1       เก็บค่า 1 ลงในตำแหน่งที่ R0 ชี้อยู่
LJMP 0

```

จากโปรแกรมจะเก็บค่า 78 ลงใน R0 บรรทัดต่อมาจะเป็นการใส่ค่า 1 ลงในตำแหน่งที่ R0 ชี้อยู่ ซึ่งขณะนั้น R0 จะชี้อยู่ที่ตำแหน่ง 78H จึงทำให้ค่าตำแหน่งที่ 78H เก็บค่า 1 พิจารณาไปตามโปรแกรมต่อไปนี้

ORG 8000H	
MOV PSW,#0	เลือกรีจิสเตอร์ bank 0
MOV R0,90H	ใส่ค่า 90 ลงใน R0
MOV @R0,#01	ใส่ค่า 01 ลงในตำแหน่งที่ R0 ชี้อยู่
LJMP 0	

การเขียนโปรแกรมข้างบน R0 จะเก็บค่า 90H จากนั้นจะเก็บข้อมูล 01 ลงในตำแหน่งที่ R0 ชี้อยู่ ซึ่งจะเท่ากับนำค่า 01 ไปใส่ในตำแหน่ง 90H ซึ่งเป็นตำแหน่งของพอร์ต 1 นั่นเอง

□ The Register Indexed Addressing Mode

การอ้างตำแหน่งแบบนี้จะเป็นลักษณะการเปิดตารางโดยมีรีจิสเตอร์ DPTR หรือ รีจิสเตอร์ PC เป็นตัวชี้ตำแหน่งเริ่มต้นของตาราง ตัวอย่างต่อไปจะเป็นการเขียนโปรแกรมนำรูปแบบของการแสดงผลของ LED 7 ส่วน ที่มีความสัมพันธ์กับเลขฐานสิบ 0 ถึง 9

ORG 8000H	ตำแหน่งเริ่มต้น โปรแกรม
MOV DPTR,#8100H	เก็บค่าตำแหน่งของตารางลง DPTR
MOV A,#0	เก็บค่าตัวเลขที่จะแสดงผล
MOVC A,@A+DPTR	นำค่าใน A ไปเปิดตาราง
LJMP 0	
ORG 8100H	ตารางรูปแบบการประมวลผล
DB 0C0H	0
DB 0F9H	1
DB 0A4H	2
DB 0B0H	3
DB 99H	4
DB 92H	5
DB 82H	6
DB 0F0H	7
DB 80H	8
DB 90H	9

Stack-Oriented Data Transfer

รีจิสเตอร์ Stack Pointer (SP) จะใช้ในการเก็บค่าของโปรแกรมเคาน์เตอร์ (PC) ซึ่งมีความสัมพันธ์กับคำสั่ง push และ pop รีจิสเตอร์ SP นี้จะเก็บอยู่ที่ตำแหน่ง 81H ของหน่วยความจำภายใน ถ้าหากต้องการให้หน่วยความจำ stack อยู่ที่ใด จะต้องโปรแกรมให้รีจิสเตอร์ SP ก่อนพิจารณาตัวอย่างต่อไปนี้ จะกำหนดให้ SP ชี้อยู่ที่หน่วยความจำ 4FH

ORG 8000H

MOV SP,#4FH

กำหนดค่าตำแหน่งเริ่มต้นของstack pointer

MOV A,#45H

เก็บค่า 45H ลงในรีจิสเตอร์ A

PUSH A

PUSH ค่าในรีจิสเตอร์ A

MOV B,#0

clear ค่าใน B

POP B

คืนค่าจากstack มาในรีจิสเตอร์ B

LJMP 0

ชุดคำสั่งการประมวลผลข้อมูล (Data Processing Instructions)

ชุดคำสั่งทางคณิตศาสตร์

ชุดคำสั่งทางคณิตศาสตร์ของ MCS-51 แสดงให้เห็นได้ดังตารางชุดคำสั่งซึ่งจะเห็นได้ว่าสามารถอ้าง addressing ได้หลายโหมด

ชุดคำสั่งบวก-ลบ

คำสั่งบวกมี 2 คำสั่ง คือ add (ADD) และ add with carry (ADDC) โดยคำสั่ง ADDCจะนำค่าใน carry flag เข้ามาบวกโดยผลลัพธ์จากการบวกจะเก็บในรีจิสเตอร์ A การบวกเลขสองจำนวน ถ้าผลลัพธ์มีค่าเกิน FFH ตัว OV flag จะเซต ดังตัวอย่าง

ORG 8000H

MOV A,#1

นำค่า 1 เก็บในรีจิสเตอร์A

ADD A,#2

บวกค่า 2 กับA และเก็บค่าในA

MOV 78H,#3

นำค่า 3 เก็บใน78H ของ RAM ภายใน

ADD A,78H

นำค่าในตำแหน่ง 78H บวกกับค่าใน A

MOV 79H,#4

นำค่า 4 เก็บในตำแหน่งที่ 79H

MOV R0,#79H

นำค่า 79H เก็บใน R0

ADD A,@R0

นำค่าจากตำแหน่งที่ชี้โดย R0 บวกกับ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV R5,#5

นำค่า 5 เก็บใน R5

ADD A,R5

บวกค่าใน R5 กับ A

สำหรับคำสั่งการบวกเลขขนาด 16 บิต เช่น บวกค่า X และ Y จะทำได้โดยเก็บค่าแต่ละค่าไว้ในหน่วยความจำขนาด 2 ไบต์ โดยแบ่งเป็น XH และ XL และแบ่งค่า Y เป็นค่า YH และ YL จากนั้นให้บวกไบต์ต่ำบวกไบต์ข้อมูลทั้งสองและไปบวกไบต์สูงของข้อมูลทั้งสอง

ถ้าหากการบวกไบต์ต่ำมีตัวทศเกิดขึ้นจะนำไปบวกกับไบต์สูงด้วยคำสั่ง ADDC ตัวอย่างต่อไปนี้จะบวกเลข 1234H กับ 12EFH

```
ORG 8000H          กำหนดค่า X ,Y
MOV 78H,#34H      X=1234H,XL=34H
MOV 79H,#12H      XH=12H
MOV 7AH,#0EFH     Y=12EH,YL=EFH
MOV 7BH,#12H      YH=12H
MOV A,78H
ADD A,7AH
MOV 79H,A
MOV A,79H         เก็บค่าไบต์สูงไว้ใน A
ADDC A,7BH
MOV 79H,A         นำผลบวกที่ได้เก็บไว้ใน XH
```

จากโปรแกรมนำข้อมูลผลลัพธ์ 16 บิต เก็บใน XH และ XL ตัวอย่างต่อไปจะลบค่า Y และ X โดยตัวแปร X และ Y เป็นตัวเลข 16 บิต

```
ORG 8000H
MOV 78H,#34H      X=1234H,XL=34H
MOV 79H,#12H      XH=12H
MOV 7AH,#35H      Y=1135H,YL=35H
MOV 7BH,#11H      YH=11H
CLR C             เคลียร์ carry flag
MOV A,79H         นำค่า XH ไว้ใน A
SUB A,78H
MOV 79H,A         เก็บค่าผลลัพธ์ไว้ใน XH
```

ถ้าหากต้องการกระทำทางคณิตศาสตร์กับข้อมูลหลายๆ ไบต์อาจต้องเขียน โปรแกรมวนลูป
 เพิ่มให้บวกกันทีละไบต์ เช่น ถ้าตัวแปร X และ Y เก็บข้อมูลขนาด 4 ไบต์ ถ้าจะนำมาบวกกันเขียน
 โปรแกรมได้ดังนี้

```

ORG      8000H
MOV      70H,#67H          X=01234567H
MOV      71H,#45H
MOV      72H,#23H
MOV      73H,#01H
MOV      78H,#0EFH        Y=89ABCDEFH
MOV      79H,#0CDH
MOV      7AH,#0AH
MOV      7BH,#89H
MOV      R3,#4             กำหนดค่า pointers R0 และ R1
MOV      R0,70H           ให้ R0 ชี้ตำแหน่งของ X
MOV      R1,78H           ให้ R1 ชี้ตำแหน่งของ Y
CLR      C
LOOP:    MOV      A,@R0    อ่านค่าไบต์ของ X
        ADDC     A,@R1    บวกกับค่าที่ชี้โดย R1 กับ Y
        MOV      @R0,A    เก็บค่าผลลัพธ์ใน X
        INC     R0
        INC     R1
        DJNZ    R3,LOOP
    
```

□ คำสั่งการคูณและการหาร

ในการคูณและการหารรีจิสเตอร์ที่ใช้คือ A และ B ถ้าเป็นการคูณเลข 8 บิตในรีจิสเตอร์ A และ B
 ผลลัพธ์ที่ได้จะมีขนาด 16 บิต โดยไบต์ค่าของผลลัพธ์จะเก็บใน A และไบต์ค่าจะเก็บใน B

MUL AB

สำหรับการหารเลข จะหารค่าในรีจิสเตอร์ B ออกจากรีจิสเตอร์ A ผลลัพธ์ที่ได้เก็บใน A
 และรีจิสเตอร์ B จะเก็บเศษ จะเขียนชุดคำสั่งได้ดังนี้

DIV AB

```

ORG 8000H
MOV A,#9
MOV B,#5
MUL AB
MOV A,#99
MOV B,#5
MUL AB
MOV A,#10
MOV B,#5
DIV AB
MOV A,#99
MOV B,#5
DIV AB
MOV A,#99
MOV B,#0
DIV AB

```

□ ชุดคำสั่ง AND ,OR, Exclusive OR

ชุดคำสั่งเหล่านี้สามารถอ้างตำแหน่งแบบบิต หรือแบบไบท์ก็ได้ ถ้าเป็นการกระทำแบบไบท์จะเป็นการกระทำลอจิกกับบิตต่อไบต์ที่กำลังติดต่อยู่

```

MOV A,#18H
ORL PSW,A

```

□ ชุดคำสั่งกระทำลอจิกกับบิต

ชุดคำสั่งนี้อ้างข้อมูลระดับบิต ซึ่งอาจมองได้ว่าบิต carry flag (c) เป็นเสมือน bit-accumulator ได้ พิจารณาโปรแกรมต่อไปนี้

```

ORG 8000H
CLR A
SETB ACC0
ORL C,ACC.0
SETB C
ORL C,ACC.0

```

กลุ่มคำสั่งกระโดด

กลุ่มคำสั่งนี้ จะใช้ในการย้ายโปรแกรมเคาน์เตอร์ไปยังตำแหน่งที่ต้องการซึ่งประกอบไปด้วยชุดคำสั่งดังต่อไปนี้

- ชุดคำสั่งกระโดดแบบไม่มีเงื่อนไข
- ชุดคำสั่งกระโดดแบบมีเงื่อนไข
- ชุดคำสั่งเรียกโปรแกรมย่อย

ชุดคำสั่งกระโดดแบบมีเงื่อนไข เมื่อโปรแกรมทำถึงชุดคำสั่งนี้จะกระโดดไปยังตำแหน่งที่กำหนดทันที ชุดคำสั่งนี้ ได้แก่

`SJMP<rel add>` เป็นคำสั่งกระโดดที่อ้างตำแหน่งที่จะกระโดดแบบ 2's complement โดยจะกระโดดไปข้างหน้าได้ 127 กิโลไบต์ และกระโดดไปข้างหลัง 128 ไบต์

`AJMP<add 11>` เป็นคำสั่งกระโดดที่อ้างตำแหน่งได้ 11 บิต ซึ่งจะทำให้กระโดดได้ในช่วง 2 กิโลไบต์

`LJMP<add16>` เป็นคำสั่งที่อ้างตำแหน่งได้ 16 บิต ซึ่งสามารถกระโดดไปยังตำแหน่งใดๆ ได้ตลอดที่หน่วยความจำมีอยู่

`JMP @A+DPTR` จะกระโดดไปยังตำแหน่งที่อยู่ใน data pointer บวกกับค่าในรีจิสเตอร์ A ตัวอย่างต่อไปจะประยุกต์ใช้คำสั่งกระโดดแบบไม่มีเงื่อนไข เพื่อเขียนโปรแกรมในลักษณะเลือกทำ โดยในโปรแกรมจะหาค่าผลคูณของค่าในตัวแปร A กับค่าต่างๆ โดยถ้าในตัวแปร A เป็น 0 จะคูณค่า A กับค่าที่อ่านได้จาก P0 ถ้าค่าใน A เป็น 1 จะคูณค่าใน A กับค่าที่อ่านจากพอร์ต P1 โดยการทำงานของโปรแกรมโดยเขียนโปรแกรมในลักษณะ โปรแกรมย่อย 4 โปรแกรม โดยแต่ละโปรแกรมมีขนาด 4 ไบต์ เมื่อรับค่า A มาคูณด้วย 4 จากนั้นใช้คำสั่งกระโดดแบบเปิดตาราง โดยค่ารีจิสเตอร์ DPTR ซึ่งที่ตำแหน่งคั่นของตาราง ตัวอย่างเช่นถ้า A เป็น 2 จะกระโดดไปโปรแกรมย่อยที่ 3 คือ routine 2 เป็นต้น

```
ORG    8000H
MOV    A,#2
RL     A
RL     A
MOV    DPTR,#TABLE
JMP    @A+DPTR
```

TABLE:

```
MOV    B,P0
```

```
SJMP RUN
MOV B,P1
SJMP RUN
MOV B,P2
SJMP RUN
MOV B,P3
SJMP RUN
```

RUN:

```
MOV A,P1
MUL AB
LJMP 0
```

ชุดคำสั่งกระโดดแบบมีเงื่อนไข ชุดคำสั่งนี้จะตรวจสอบเงื่อนไขก่อนกระโดดโดยกระโดดแบบ relative ได้แก่ คำสั่งต่อไปนี้

○ กระโดดโดยทดสอบบิต

```
JZ<rel add>
JNZ<rel add>
JC<rel add>
JNC<rel add>
JB<bit>,<rel add>
JNB<bit> ,<rel add>
JBC<bit>,<rel add>
```

○ กระโดดโดยเปรียบเทียบค่า ถ้าไม่เท่าจะกระโดด

```
CJNE A,direct,<rel add>
CJNE Rn,#data,<rel add>
CJNE @Ri,#data,<rel add>
```

○ ลดค่าจากตำแหน่งที่กำหนด ถ้าไม่เป็นศูนย์จะกระโดด

DJNZ Rn,<rel add>

DJNZ direct,<rel add>

○ ชุดคำสั่งเรียกโปรแกรมย่อย ในชุดคำสั่งนี้จะประกอบไปด้วยคำสั่ง CALL และ RETURN โดยมีคำสั่งต่างๆ ดังนี้

ACALL<address 11> เป็นคำสั่งเรียกโปรแกรมย่อยที่อ้างตำแหน่งของโปรแกรมย่อยแบบ 11 บิต ซึ่งสามารถเรียกโปรแกรมย่อยที่อยู่ห่างจาก PC ได้ไม่เกิน 2 กิโลไบต์

LCALL<address 16> เป็นคำสั่งเรียกโปรแกรมย่อยที่อ้างตำแหน่งของโปรแกรมย่อยแบบ 16 บิต ซึ่งสามารถเรียกโปรแกรมย่อยที่อยู่ตำแหน่งใดๆ ที่ MCS-51 มีอยู่ได้

RET จะใช้ท้ายโปรแกรมย่อย เพื่อกลับสู่โปรแกรมหลัก

RETI จะใช้ท้ายโปรแกรมย่อยที่เป็น โปรแกรมตอบสนองการอินเตอร์รัปต์

SW	EQU	21H
COUNT_TEN	EQU	23H
LS1	EQU	24H
MS1	EQU	25H
LS	EQU	26H
MS	EQU	27H
DISPLAY_POINTER	EQU	20H
DISPLAY_BUFFER	EQU	30H
SEC14	EQU	34H
DBUFF	EQU	35H
SEC	EQU	36H
T__2	EQU	40H
BUFF1	EQU	41H
BUFF2	EQU	42H
CHK	EQU	43H
DISPLAY_BUF	EQU	44H
SNDCON	EQU	0A0H

```

ORG 000H
LJMP    MAIN

```

```

ORG    03H
LJMP    SERVICE_INT0

```

```

ORG    0BH
LJMP    SERVICE_T0

```

```
ORG      013H
LJMP     SERVICE_INT1
```

```
ORG      100H
```

```
; ***** SOUND SUB ***** 36
```

```
; SOUND      GENERATOR
```

```
; IN      = R2 FREQUENCY
```

```
;          = R3 LENGTH
```

```
; REG = A,R2,R3,R4,R5,DPTR
```

```
SOUND:    MOV    R5,#0      ; END FLAG
```

```
MOV       R4,#SNDCON      ; DELAY CONSTANT
```

```
SOUND1:   LCALL  SOUNDS
```

```
CJNE     R5,#1,SOUND1
```

```
RET
```

```
SOUNDS:
```

```
SETB     P1.7             ; OUT 1 (P3.4) pin out put
```

```
LCALL    SOUNDX
```

```
CLR      P1.7             ; OUT 0
```

```
LCALL    SOUNDX
```

```
RET
```

```
SOUNDX:   MOV    A,R2      ; FREQUENCY DELAY
```

```
SOUNDX1:  LCALL  SOUNDX
```

```
DEC      A
```

```
JNZ     SOUNDX1
```

```
RET
```

```
SOUNDX:   DJNZ  R4,SOUNDX1 ; LENGTH COUNT DOWN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      R4,#SNDCON
DJNZ    R3,SOUNDY1
MOV      R5,#1
SOUNDY1:  RET

```

```

; ***** nBEEP SUB *****

```

```

; ANY BEEP SUBROUTINE

```

```

; REG = A,R2,R3,R4,R5,DPTR

```

```

UBEEP:   MOV  R2,#0DH      ;USER BEEP (37)

```

```

MOV      R3,#60H

```

```

LCALL   SOUND

```

```

RET

```

```

SCAN_DISP:

```

```

CLR     P3.7

```

```

SETB   P3.1

```

```

SETB   P3.0

```

```

MOV     A,@R0

```

```

MOV     P1,A

```

```

INC     R0

```

```

LCALL   DELAY

```

```

SETB   P3.7

```

```

CLR     P3.1

```

```

SETB   P3.0

```

```

MOV     A,@R0

```

```

MOV     P1,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INC R0
LCALL DELAY

SETB P3.7

SETB P3.1

CLR P3.0

MOV A,@R0

MOV P1,A

INC R0

LCALL DELAY

RET

TRANSFER:

MOV R1,#DISPLAY_BUFFER

TRAN1:

MOV A,@R0

MOV R6,A

ANL A,#0FH ; MASK 4 BIT LOW

LCALL NIBBLE_7SEG

MOV @R1,A

INC R1

MOV A,R6

SWAP A

ANL A,#0FH

LCALL NIBBLE_7SEG

MOV @R1,A

INC R0

INC R1

```

MOV     A,@R0
ANL     A,#0FH
LCALL   NIBBLE_7SEG
MOV     @R1,A
RET

```

NIBBLE_7SEG:

```

MOV     DPTR,#TABLE1
MOVC    A,@A+DPTR
RET

```

DELAY:

```

MOV     R1,#0FFH
DJNZ    R1,$
RET

```

TABLE1:

```

DB      40H
DB      79H
DB      24H
DB      30H
DB      19H
DB      12H
DB      02H
DB      78H
DB      00H
DB      10H
DB      08H
DB      03H
DB      46H
DB      21H
DB      06H
DB      0EH

```

SCANSEG:

```
SCANSEG1:  MOV      R0,DISPLAY_POINTER
            LCALL   TRANSFER
            MOV     R0,#30H
            LCALL   SCAN_DISP
            RET
```

```
LOUND_SPERKER:  LCALL   UBEEP
                RET
```

```
PROG_CAL:      MOV     A,#14D
                MOV     B,T__2
                DIV    AB
                MOV     BUFF1,A
                MOV     A,B
                MOV     B,#6D
                MUL    AB
                MOV     BUFF2,A
                MOV     A,BUFF1
                MOV     B,#60D
                MUL    AB
                ADD    A,BUFF2
                MOV     DPL,A
                MOV     DPH,#00H
                CLR    C
                CALL   HTOD
                RET
```

; ***** HTOD SUB ***** 62

; HEX TO DECIMAL

; IN = DPTR

; OUT = R1,R2,R3

; REG = A,R0,R1,R2,R3,R4,R5,DPTR

HTOD:

CLR A ;CLEAR OUTPUT

MOV R1,A

MOV R2,A

MOV R3,A

MOV R4,#16 ;SHIFT 16 BIT

HTOD1:

MOV A,DPL

RLC A

MOV DPL,A

MOV A,DPH

RLC A

MOV DPH,A

MOV R5,#3 ;ADD DECIMAL

MOV R0,#3 ;INDEX TO R3

HTOD2:

MOV A,@R0

ADDC A,ACC

DA A

MOV @R0,A

DEC R0

DJNZ R5,HTOD2

DJNZ R4,HTOD1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     LS,R3
MOV     MS,R2
RET

```

```

;#####33

```

```

MAIN_COUNT1:  MOV     A,CHK
              CK1:  JNZ     CK2
              INC     CHK
              SETB   TR0
              SJMP   BACK_MAIN1
              CK2:  CLR     TR0
              MOV    T__2,SEC14

```

```

MOV     CHK,#00H
LCALL   PROG_CAL
SETB   TR0
MOV     SEC14,#00H

```

```

BACK_MAIN1:  RET

```

```

CHK_TEN:
INC     DPTR
MOV     A,DPH
CJNE   A,#49H,PASS
MOV     A,DPL
CJNE   A,#1CH,PASS

```

```

; INC     COUNT_TEN
; MOV     A,COUNT_TEN
; CJNE   A,#0AH,PASS

```

```

        LCALL  LOUND_SPERKER
: MOV    COUNT_TEN,#00H
PASS:   RET

```

SERVICE_INT0:

```

        PUSH   PSW
        PUSH   ACC
        PUSH   DPH
        PUSH   DPL
        PUSH   00H
        PUSH   01H
        PUSH   02H

        JB     SW,SW1
        CALL   MAIN_COUNT1
        MOV    DPTR,#0000H
; MOV    COUNT_TEN,#00H
        SJMP   SW2
SW1:    LCALL   CHK_TEN

SW2:    POP    02H
        POP    01H
        POP    00H
        POP    DPL
        POP    DPL
        POP    ACC
        POP    PSW
        RETI

```

SERVICE_T0:

```

        PUSH   PSW
        PUSH   ACC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PUSH    DPH
PUSH    DPL
MOV     PSW,#00011000B
```

SERVICE_BODY:

```
INC     SEC14

POP     DPL
POP     DPH
POP     ACC
POP     PSW
RETI
```

SERVICE_INT1:

```
PUSH    PSW
PUSH    ACC
PUSH    DPH
PUSH    DPL
SETB    SW
MOV     C,SW
JC      STOP_INT1
```

```
; JB    P3.5,SW3
; JNB   SW,SW3
; CLR   SW
; SJMP  STOP_INT1
```

;SW3:

```
; SETB SW
```

STOP_INT1:

```
POP     DPL
POP     DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
POP      ACC
POP      PSW
RETI
```

```
;%%%%%%%%%%% PROGRAM MAIN
```

```
%%%%%%%%%%
```

```
MAIN:
```

```
MOV      SP,#60H
MOV      LS,#00H
MOV      MS,#00H
; MOV    LS1,#00H
; MOV    MS1,#00H
MOV      COUNT_TEN,#00H
MOV      DISPLAY_POINTER,#LS
SETB     PT0
SETB     EA
SETB     ET0
SETB     IT0
SETB     EX0
SETB     EX1
MOV      CHK,#00H

MOV      TMOD,#0000001B
MOV      TL0,#00H
MOV      TH0,#00H
SETB     P3.5          ; SET BIT FOR BUZZER ON

CLR      SW
```

```
HEAR:
```

```
; JNB    SW,HEAR
LCALL    SCANSEG
; LCALL  SERVICE_INT1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SJMP

HEAR

END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้ำเกลือ

ในที่นี้จะขอลำดับ ชนิดของน้ำเกลือและหลักการให้น้ำเกลือพอสังเขป ดังนี้

ชนิดของน้ำเกลือ ที่ใช้บ่อยได้แก่

1. **นอร์มัลซาลิน (Normal saline solution/NSS)** หมายถึงน้ำเกลือธรรมดาที่มีความเข้มข้น 0.9% ซึ่งเท่ากับ ความเข้มข้นของเกลือในกระแสเลือดของคนปกติ มีค่า pH 7.35 และ 1,000 มล.

2. **5% เดกซ์โทรส (5% Dextrose in water หรือ 5% D/W)** หมายถึงน้ำตาลเดกซ์โทรสที่มีความเข้มข้น 5% ไม่มีเกลือผสม มีค่า pH 7.35 และ 1,000 มล.

3. **5% เดกซ์โทรสในนอร์มัลซาลิน (5% Dextrose in NSS หรือ 5% D/NSS)** หมายถึงน้ำตาลเดกซ์โทรสเข้มข้น 5% ผสมกับน้ำเกลือธรรมดา

4. **5% เดกซ์โทรสใน 1/3 นอร์มัลซาลิน (5% Dextrose in 1/3 NSS หรือ 5% D/1/3 NSS)** หมายถึงน้ำตาลเดกซ์โทรสเข้มข้น 5% ผสมกับน้ำเกลือที่มีความเข้มข้น 0.3% (เข้มข้นเพียง 1/3 ของน้ำเกลือธรรมดา) มีค่า pH 7.35 และ 1,000 มล.

ข้อบ่งชี้และข้อควรระวัง

ข้อบ่งชี้ น้ำเกลือจะให้ในผู้ป่วยที่มีอาการดังนี้

1. **ขาดน้ำ (Dehydration)** เนื่องจากท้องเดิน, อาเจียนรุนแรง (เช่น ภาวะพาราไอโธไทรอยด์ก่อนในสมอง), หอบ (เช่น หืดปอดอักเสบ หลอดลมฝอยอักเสบ) ควรให้น้ำเกลือที่มี NSS ผสม เช่น NSS, 5% D/NSS, 5% D/1/3 NSS

2. **ช็อก (Shock)** เนื่องจากเสียเลือด, เสียน้ำ, หรือจากสาเหตุอื่น ๆ ควรให้น้ำเกลือที่มี NSS ผสมเช่นเดียวกับข้อ 1

3. **หมดสติ หรือกินข้าวและน้ำไม่ได้นาน ๆ** ควรให้น้ำเกลือที่มีเดกซ์โทรสผสมกับน้ำเกลือ เช่น 5% D/NSS, 5% D/1/3 NSS

4. **น้ำตาลในเลือดต่ำ (Hypoglycemia)** เนื่องจากอดอาหารนาน ๆ, คัมเสียด, ไข้ยารักษาเบาหวานเกินขนาด ควรให้น้ำเกลือที่มีเดกซ์โทรสผสม เช่น 5% D/W, 5% D/NSS 5% D/1/3 NSS

5. **ผู้ป่วยที่อดอาหารและน้ำก่อนและหลังผ่าตัด** ควรให้น้ำเกลือที่มีเดกซ์โทรสผสมกับน้ำเกลือเช่นเดียวกับข้อ 3

6. **ผู้ป่วยที่ต้องฉีดยาเข้าทางหลอดเลือดดำวันละหลาย ๆ ครั้ง** เลือกให้น้ำเกลือชนิดหนึ่งชนิดใดก็ได้ โดยให้ช้า ๆ เพียงเพื่อให้มีสภาพน้ำเกลือคาบกับเส้นเลือด เพื่อให้เกิดความสะดวกในการฉีดยา

ข้อควรระวัง

1. **ผู้ป่วยที่เป็นโรคหัวใจ โรคไตวาย หรือมีอาการบวมทั่วตัว** ควรระวังเป็นพิเศษ และควรหลีกเลี่ยงการใช้น้ำเกลือที่มีความเข้มข้นของเกลือมาก เพราะอาจทำให้หัวใจวายหรือเกิดภาวะน้ำคั่งในปอดหรือปอดบวม (pulmonary edema) ได้

2. **ผู้ป่วยที่เป็นโรคความดันโลหิตสูง** ควรหลีกเลี่ยงการใช้น้ำเกลือที่มีความเข้มข้นของเกลือมาก (เช่น NSS, 5% D/NSS) ยกเว้นในรายที่มีภาวะขาดเกลือร่วมด้วย

3. **ผู้ป่วยที่เป็นเด็กเล็กไม่ควรใช้น้ำเกลือที่มี NSS ผสม** ควรใช้น้ำเกลือที่มีความเข้มข้นน้อย ได้แก่ น้ำเกลือที่มีความเข้มข้น 0.3% เช่น 5% D/1/3 NSS

การเลือกใช้น้ำเกลือ

1. NSS (นอร์มัลซาลิน)

ข้อบ่งชี้

1. สำหรับผู้ป่วยที่มีภาวะขาดน้ำหรือช็อก
2. เด็กที่มีภาวะขาดน้ำรุนแรง และไม่สามารถให้น้ำเกลือทางหลอดเลือดดำได้ (เช่น หลอดเลือดดำมีขนาดเล็กแทงไม่ได้) ให้ฉีด NSS เข้าใต้หนังตรงหน้าขาหรือสะบักหลัง ขนาด 20-40 มล. ค่อย ๆ ให้น้ำหนักตัว 1 กก.

ข้อควรระวัง

1. ห้ามให้ NSS ทางหลอดเลือดดำในเด็กเล็ก เพราะมีปริมาณของเกลือเข้มข้นกว่าเกลือที่มีอยู่ในกระแสเลือดของเด็ก อาจเป็นอันตรายได้
2. ผู้ป่วยที่มีภาวะหัวใจวาย ไตวาย ความดันโลหิตสูงหรือบวมทั่วตัว ควรหลีกเลี่ยงการให้ NSS (ยกเว้นในรายที่มีภาวะขาดเกลือ)

ขนาดที่ใช้

1. ผู้ใหญ่ 1,000-2,000 มล. ใน 12-24 ชม. ในรายที่มีภาวะขาดน้ำรุนแรงหรือช็อก ในระยะแรก ควรให้เร็ว ๆ จนอาการดีขึ้น จึงให้ช้า ๆ

2. 5% D/NSS (5% เดกซ์โทรสในนอร์มัลซาลีน)

ข้อบ่งชี้

สำหรับผู้ป่วยที่มีภาวะขาดน้ำ, ช็อก อดอาหาร, หรือมีภาวะน้ำตาลในเลือดต่ำ

ข้อควรระวัง เช่นเดียวกับ NSS

ขนาดที่ใช้ เช่นเดียวกับ NSS

3. 5% DW (5% เดกซ์โทรส)

ข้อบ่งชี้

สำหรับเด็กและผู้ใหญ่ที่มีภาวะน้ำตาลในเลือดต่ำ หรืออดอาหาร

ขนาดที่ใช้

ผู้ใหญ่ 1,000-2,000 มล. ใน 12-24 ชม.

เด็ก 100 มล. ต่อน้ำหนักตัว 1 กก. ใน 24 ชม.

4. 5% D $\frac{1}{3}$ NSS (5% เดกซ์โทรสใน $\frac{1}{3}$ นอร์มัลซาลีน)

ข้อบ่งชี้

สำหรับเด็กที่มีภาวะขาดน้ำ, ช็อก, ไข้เลือดออก, อดอาหาร หรือมีภาวะน้ำตาลในเลือดต่ำ

ขนาดที่ใช้

100 มล. ต่อน้ำหนักตัว 1 กก. ใน 24 ชม. ถ้ามีภาวะขาดน้ำรุนแรงหรือช็อก ใน 2 ชม. แรก ควรให้ขนาด 20-40 มล. ต่อน้ำหนักตัว 1 กก.

อาการแสดงว่าให้น้ำเกลือแล้วดีขึ้น

ผู้ป่วยที่มีภาวะขาดน้ำหรือช็อก เมื่อให้น้ำเกลือแล้วมีอาการดังต่อไปนี้ แสดงว่าอาการดีขึ้น

1. มีความรู้สึกตัวดีขึ้น พูดคุยได้ดีขึ้น หน้าตาดูมีขึ้น ผิวหนังเต่งตึงขึ้น หอบน้อยลงและกระสับกระส่ายน้อยลง
2. ความดันเลือดที่เคยตก เริ่มกลับคืนสู่ระดับปกติ
3. ชีพจรที่เคยเต้นเบาและเร็ว กลับคืนแรงขึ้นและช้าลง
4. มีปัสสาวะออกมากขึ้น โดยให้ผู้ป่วยปัสสาวะลงกระโถนหรือขวด แล้วดวงจะพบว่าปัสสาวะออกอย่างน้อย 1 มล. ต่อน้ำหนักตัว 1 กก. ต่อชั่วโมง เช่น ผู้ป่วยหนัก 30 กก. ใน 1 ชั่วโมง ควรมีปัสสาวะออกอย่างน้อย 30 มล.

น้ำเกลือไม่ใช่ยาบำรุง ยาเพิ่มเลือด หรือยาเพิ่มแรง ควรใช้เมื่อยามจำเป็นจริง ๆ เท่านั้น.

อันตรายจากน้ำเกลือ

1. ถ้าเครื่องใช้และน้ำยาไม่สะอาด หรือเทคนิคการให้ไม่ถูกต้อง อาจทำให้เกิดการอักเสบหรือติดเชื้อได้
2. ถ้ามีฟองอากาศ เพราะไล่อากาศจากสายน้ำเกลือไม่หมด ฟองอากาศจะเข้าไปในเส้นเลือดดำและเข้าสู่หัวใจอาจเป็นอันตรายได้
3. มีอาการไตและหนาวสั่นจากการแพ้ น้ำเกลือ
4. ถ้าใช้น้ำเกลือชนิดที่มีความเข้มข้นของเกลือมากกว่าความเข้มข้นของเกลือในเลือดอาจเป็นอันตรายถึงตายได้ โดยเฉพาะอย่างยิ่งในเด็กเล็ก
5. ถ้าให้น้ำเกลือมากหรือเร็วเกินไป อาจทำให้บวม มีน้ำคั่งในปอดหรือหัวใจวายถึงตายได้ โดยเฉพาะอย่างยิ่งในเด็กเล็ก คนสูงอายุ คนที่เป็นโรคหัวใจ หรือโรคไตอยู่ก่อน

ข้อควรปฏิบัติ

1. ควรให้น้ำเกลือเฉพาะในรายที่มีความจำเป็น (มีข้อบ่งชี้) จริง ๆ เท่านั้น น้ำเกลือไม่ใช่ยาบำรุง ยาเพิ่มเลือด หรือยาเพิ่มแรง และก็ไม่ใช่ว่าใช้แทนอาหาร จึงไม่ควรให้อย่างพร่ำเพรื่อ
2. ควรเตรียมเครื่องใช้ให้สะอาดปราศจากเชื้อโรค และทำความสะอาดเทคนิคที่ถูกต้อง
3. ควรให้น้ำเกลือช้า ๆ หรือน้อย ๆ ไว้ก่อน เมื่อเห็นว่าน้อยไปก็เพิ่มเติมในภายหลังได้ อย่าให้เร็วเกินไป (ยกเว้นในกรณีที่มีภาวะขาดน้ำรุนแรงหรือช็อก ควรให้เร็ว ๆ ใน 1-2 ชม.แรก) มิฉะนั้นอาจทำให้บวม น้ำคั่งในปอดหรือหัวใจวายได้
4. หมั่นตรวจดูผู้ป่วยอย่างใกล้ชิด (โดยเฉพาะในเด็กเล็ก คนสูงอายุ คนที่เป็นโรคหัวใจหรือโรคไตวาย) ถ้าหากมีอาการบวม หายใจหอบและฟังปอดมีเสียงกรอบแกรบ (crepitation) แสดงว่ามีน้ำคั่งในปอดเนื่องจากให้น้ำเกลือมากไป ต้องหยุดน้ำเกลือ และฉีดยาขับปัสสาวะ เช่น ลาสิกซ์ (ย2 1. 1) 1-2 หลอดเข้าเส้นเลือดทันที หากยังหอบอยู่ให้รีบพาไปโรงพยาบาล
5. ถ้ามีอาการหนาวสั่นระหว่างให้น้ำเกลือ แสดงว่าผู้ป่วยแพ้ น้ำเกลือ ให้ถอดเข็มน้ำเกลือออก และฉีดยาแก้แพ้ เช่น คลอร์เฟนิรามีน (ย7. 1) $\frac{1}{2}$ - 1 หลอดเข้ากล้ามเนื้อ ถ้าจำเป็นต้องให้น้ำเกลือต่อ ควรเปลี่ยนขวดใหม่

4.1.6 ความตึงผิว (surface tension)

ภายในของเหลวแต่ละโมเลกุลจะถูกแรงดึงดูดรวม (cohesive force) กระทำ
 เท่ากันหมดในทุกทิศทาง เนื่องจากโมเลกุลข้างเคียง แต่โมเลกุลที่ผิวของของเหลวนอกจาก
 จะถูกกระทำด้วยแรงดึงดูดรวมแล้วยังถูกกระทำด้วยแรงดึงดูดแยก (adhesive force) เนื่องจาก
 จากภาชนะที่บรรจุ หรืออากาศด้วยคิดให้ผิวของของเหลวเปรียบเสมือนเป็นแผ่นเยื่อยืดหยุ่นที่
 ถูกดึงด้วยแรงดึง 2 ข้างกัน โดยมีทิศตั้งฉากกับแผ่นเยื่อ ถ้าแรงดึงดูดรวมมีค่ามากกว่าแรงดึงดูด
 แยกของเหลวนั้นก็จะเป็นเยือกหรือเกาะติดภาชนะที่บรรจุ เช่นหยดปรอทลงบนกระดาษหนังสือพิมพ์
 แต่ถ้าแรงดึงดูดรวมมีค่าน้อยกว่าแรงดึงดูดของเหลวนั้นก็จะเป็นเยือกหรือเกาะติดภาชนะที่บรรจุ
 เช่นหยกน้ำบนกระดาษหนังสือพิมพ์

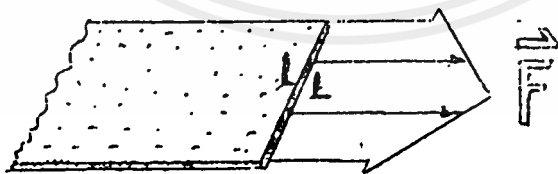
เมื่อมีแรงภายนอกดึงแผ่นเยื่อตามแนวสัมผัสกับแผ่นเยื่อ เนื่องจากแผ่นเยื่อมีความยืด
 หยุ่นจึงเกิดแรงต้านแรงภายนอกนั้นด้วยขนาดที่เท่ากันแต่ทิศทางตรงกันข้าม เรียกแรงที่เกิดขึ้น
 ในแผ่นเยื่อของเหลวว่าแรงตึงผิว (surface tension force)

กำหนดให้ความตึงผิวเป็นอัตราส่วนของแรงตึงผิวขณะแผ่นเยื่อถูกดึงขาด ต่อความยาว
 ของแผ่นเยื่อค้ำที่ถูกแรงกระทำ

$$\beta \equiv F_s / 2L$$

(4.7)

- F_s แทนแรงตึงผิวขณะแผ่นเยื่อขาด (N)
- L แทนความยาวค้ำที่ถูกแรงกระทำ (m)
- β แทนความตึงผิว (N/m)



รูป 4.1 ความยาวของแผ่นเยื่อค้ำที่ถูกแรง F ดึงมีค่าเท่ากับ $2L$

ตัวอย่าง 4.1 วงแหวนบางรัศมี $3.5 \times 10^{-2} m$ มวล $10^{-3} kg$ จุ่มในน้ำมัน
 เบนซิน เมื่อดึงวงแหวนขึ้นพบว่าขอบวงแหวนพอกันหลุดจากผิวของเหลวในแนวระดับที่ต้องใช้
 แรง $2.3 \times 10^{-2} N$ จงหาความตึงผิวของน้ำมันเบนซิน

2 Light

Table 1.1 Electromagnetic spectrum

Type of radiation	Wavelength	Frequency (Hz)	Quantum energy (eV)
Radio waves	100 km	3×10^3	1.2×10^{-11}
Microwaves	300 mm	10^9	4×10^{-6}
Infrared	0.3 mm	10^{12}	4×10^{-3}
Visible	0.7 μm	4.3×10^{14}	1.8
Ultraviolet	0.4 μm	7.5×10^{14}	3.1
X rays	0.03 μm	10^{16}	40
Gamma rays	0.1 nm	3×10^{18}	1.2×10^4
	1.0 pm	3×10^{20}	1.2×10^6

Note: The divisions into the various regions are for illustration only; there is no firm dividing line between one region and the next. The numerical values are only approximate; the upper and lower limits are somewhat arbitrary.

speed c of $3 \times 10^8 \text{ m s}^{-1}$, a frequency of some $5 \times 10^{14} \text{ Hz}$ and a wavelength of about 400 nm. Maxwell's theory suggested the possibility of producing electromagnetic waves with a wide range of frequencies (or wavelengths). In 1887 Hertz succeeded in generating non-visible electromagnetic waves, with a wavelength of the order of 10 m, by discharging an induction coil across a spark gap thereby setting up oscillating electric and magnetic fields. Visible light and Hertzian waves are part of the *electromagnetic spectrum* which, as we can see from Table 1.1, extends approximately over the wavelength range of 1.0 pm to 100 km. The wave theory thus became the accepted theory of light. However, while the wave theory, as we shall see below, provides an explanation of optical phenomena such as interference and diffraction, it fails completely when applied to situations where energy is exchanged, such as in the emission and absorption of light and the photoelectric effect. The photoelectric effect, which is the emission of electrons from the surfaces of solids when irradiated, was explained by Einstein in 1905. He suggested that the energy of a light beam is not spread evenly but is concentrated in certain regions, which propagate like particles. He called these 'particles' *photons*. Einstein was led to the concept of photons by the work of Planck on the emission of light from hot bodies. Planck found that the observations indicated that light energy is emitted in multiples of a certain minimum energy unit. The size of the unit, which is called a *quantum*, depends on the wavelength λ of the radiation and is given by

2 Light

Table 1.1 Electromagnetic spectrum

Type of radiation	Wavelength	Frequency (Hz)	Quantum energy (eV)
Radio waves	100 km	3×10^3	1.2×10^{-11}
	300 m	10^9	4×10^{-6}
Microwaves	0.3 m	10^{12}	4×10^{-3}
Infrared	0.7 μm	4.3×10^{14}	1.8
Visible	0.4 μm	7.5×10^{14}	3.1
Ultraviolet	0.03 μm	10^{16}	40
X rays	0.1 nm	3×10^{18}	1.2×10^4
γ rays	1.0 pm	3×10^{20}	1.2×10^6

Note: The divisions into the various regions are for illustration only; there is no firm dividing line between one region and the next. The numerical values are only approximate; the upper and lower limits are somewhat arbitrary.

speed c of $3 \times 10^8 \text{ m s}^{-1}$, a frequency of some $5 \times 10^{14} \text{ Hz}$ and a wavelength of about 400 nm. Maxwell's theory suggested the possibility of producing electromagnetic waves with a wide range of frequencies (or wavelengths). In 1887 Hertz succeeded in generating non-visible electromagnetic waves, with a wavelength of the order of 10 m, by discharging an induction coil across a spark gap thereby setting up oscillating electric and magnetic fields. Visible light and Hertzian waves are part of the *electromagnetic spectrum* which, as we can see from Table 1.1, extends approximately over the wavelength range of 1.0 pm to 100 km. The wave theory thus became the accepted theory of light. However, while the wave theory, as we shall see below, provides an explanation of optical phenomena such as interference and diffraction, it fails completely when applied to situations where energy is exchanged, such as in the emission and absorption of light and the photoelectric effect. The photoelectric effect, which is the emission of electrons from the surfaces of solids when irradiated, was explained by Einstein in 1905. He suggested that the energy of a light beam is not spread evenly but is concentrated in certain regions, which propagate like particles. He called these 'particles' *photons*.

Einstein was led to the concept of photons by the work of Planck on the emission of light from hot bodies. Planck found that the observations indicated that light energy is emitted in multiples of a certain minimum energy unit. The size of the unit, which is called a *quantum*, depends on the wavelength λ of the radiation and is given by

4.4 Display devices

Table 4.2

Material	Dopant	Peak emission (typical values) (nm)	Color	External quantum efficiencies (commercial diodes) (%)
GaAs	Si	910 → 1020	Infrared	10
GaP	N	570	Green	0.1
GaP	N, N	590	Yellow	0.1
GaP	Zn, O	700	Red	4
GaAs _{0.6} P _{0.4}		650	Red	0.2
GaAs _{0.35} P _{0.65}	N	632	Orange	0.2
GaAs _{0.15} P _{0.85}	N	589	Yellow	0.05
Ga _{0.6} Al _{0.4} As	Zn	650	Red	3
Ga _x Al _{1-x} As (1 < x < 0.7)	Si	870 → 890	Infrared	15

band transitions The indirect bandgap material can also be made to emit visible radiation by doping with N; for example, orange and yellow emission may be obtained from diodes of composition GaAs_{0.35}P_{0.65}:N and GaAs_{0.15}P_{0.85}:N respectively.

Gallium aluminum arsenide (Ga_xAl_{1-x}As). A fairly recent newcomer to the LED scene is Ga_xAl_{1-x}As, which is used in highly efficient red and near-infrared LEDs. In the former, a heterojunction is formed between *n*-type Ga_{0.3}Al_{0.7}As and *p*-type Ga_{0.6}Al_{0.4}As. The *n* layer is the surface layer and, in operation, electrons are injected into the *p* material with efficiencies that approach 100%. Radiative recombination (via acceptor levels) gives rise to radiation with a wavelength of 650 nm. This radiation passes through the surface *n*-type layer with little attenuation, because its bandgap is larger than that of the *p*-layer. The high efficiencies of such 'super-bright' red LEDs may well cause them to replace the 'standard' types based on GaAsP. As far as near-infrared diodes based on GaAlAs are concerned, the basic structure is again a *p-n* junction with the *n* layer uppermost. As with GaAs, doping with Si is used to form both donor and acceptor states. The bandgap is varied through the device by varying the Al concentration from 30% at the surface to 0% some way beyond the junction into the *p* material. Peak emission occurs between 870 and 890 nm. Again, the efficiencies of these LEDs are better than those made from the 'standard' material, GaAs. However, the GaAs diodes do have the advantage that they can withstand higher peak currents.

Table 4.2 summarizes the characteristics of the most commonly used LED materials.

4.6.4 LED construction

A typical LED construction is shown in Fig. 4.16. It is obviously advantageous if most of the radiative recombinations take place from the side of the junction nearest the surface, since then the chances of re-absorption are lessened. We may ensure this by arranging that most of the current flowing across the diode is carried by those carriers that are injected into the surface layer. We assume that, as in Fig. 4.16, the surface

1.4 Display devices

Table 4.2

Material	Dopant	Peak emission (typical values) (nm)	Color	External quantum efficiencies (commercial diodes) (%)
GaAs	Si	910 → 1020	Infrared	10
GaP	N	570	Green	0.1
GaP	N, N	590	Yellow	0.1
GaP	Zn, O	700	Red	4
GaP		650	Red	0.2
GaAs _{0.6} P _{0.4}	N	632	Orange	0.2
GaAs _{0.35} P _{0.65}	N	589	Yellow	0.05
GaAs _{0.15} P _{0.85}	N	650	Red	3
Ga _{0.6} Al _{0.4} As	Zn	650	Red	3
Ga _x Al _{1-x} As (1 < x < 0.7)	Si	870 → 890	Infrared	15

band transitions. The indirect bandgap material can also be made to emit visible radiation by doping with N; for example, orange and yellow emission may be obtained from diodes of composition GaAs_{0.35}P_{0.65}:N and GaAs_{0.15}P_{0.85}:N respectively.

Gallium aluminum arsenide (Ga_xAl_{1-x}As). A fairly recent newcomer to the LED scene is Ga_xAl_{1-x}As, which is used in highly efficient red and near-infrared LEDs. In the former, a heterojunction is formed between *n*-type Ga_{0.7}Al_{0.3}As and *p*-type Ga_{0.6}Al_{0.4}As. The *n* layer is the surface layer and, in operation, electrons are injected into the *p* material with efficiencies that approach 100%. Radiative recombination (via acceptor levels) gives rise to radiation with a wavelength of 650 nm. This radiation passes through the surface *n*-type layer with little attenuation, because its bandgap is larger than that of the *p*-layer. The high efficiencies of such 'super-bright' red LEDs may well cause them to replace the 'standard' types based on GaAsP. As far as near-infrared diodes based on GaAlAs are concerned, the basic structure is again a *p-n* junction with the *n* layer uppermost. As with GaAs, doping with Si is used to form both donor and acceptor states. The bandgap is varied through the device by varying the Al concentration from 30% at the surface to 0% some way beyond the junction into the *p* material. Peak emission occurs between 870 and 890 nm. Again, the efficiencies of these LEDs are better than those made from the 'standard' material, GaAs. However, the GaAs diodes do have the advantage that they can withstand higher peak currents. Table 4.2 summarizes the characteristics of the most commonly used LED materials.

4.6.4 LED construction

A typical LED construction is shown in Fig. 4.16. It is obviously advantageous if most of the radiative recombinations take place from the side of the junction nearest the surface, since then the chances of re-absorption are lessened. We may ensure this by arranging that most of the current flowing across the diode is carried by those carriers that are injected into the surface layer. We assume that, as in Fig. 4.16, the surface

1.34 Display devices

Table 4.2

Material	Dopant	Peak emission (typical values) (nm)	Color	External quantum efficiencies (commercial diodes) (%)
GaAs	Si	910 → 1020	Infrared	10
GaP	N	570	Green	0.1
GaP	N, N	590	Yellow	0.1
GaP	Zn, O	700	Red	4
GaAs _{0.6} P _{0.4}		650	Red	0.2
GaAs _{0.35} P _{0.65}	N	632	Orange	0.2
GaAs _{0.15} P _{0.85}	N	589	Yellow	0.05
Ga _{0.6} Al _{0.4} As	Zn	650	Red	3
Ga _x Al _{1-x} As (1 < x < 0.7)	Si	870 → 890	Infrared	15

band transitions The indirect bandgap material can also be made to emit visible radiation by doping with N; for example, orange and yellow emission may be obtained from diodes of composition GaAs_{0.35}P_{0.65}:N and GaAs_{0.15}P_{0.85}:N respectively.

Gallium aluminum arsenide (Ga_xAl_{1-x}As). A fairly recent newcomer to the LED scene is Ga_xAl_{1-x}As, which is used in highly efficient red and near-infrared LEDs. In the former, a heterojunction is formed between *n*-type Ga_{0.3}Al_{0.7}As and *p*-type Ga_{0.6}Al_{0.4}As. The *n* layer is the surface layer and, in operation, electrons are injected into the *p* material with efficiencies that approach 100%. Radiative recombination (via acceptor levels) gives rise to radiation with a wavelength of 650 nm. This radiation passes through the surface *n*-type layer with little attenuation, because its bandgap is larger than that of the *p*-layer. The high efficiencies of such 'super-bright' red LEDs may well cause them to replace the 'standard' types based on GaAsP. As far as near-infrared diodes based on GaAlAs are concerned, the basic structure is again a *p-n* junction with the *n* layer uppermost. As with GaAs, doping with Si is used to form both donor and acceptor states. The bandgap is varied through the device by varying the Al concentration from 30% at the surface to 0% some way beyond the junction into the *p* material. Peak emission occurs between 870 and 890 nm. Again, the efficiencies of these LEDs are better than those made from the 'standard' material, GaAs. However, the GaAs diodes do have the advantage that they can withstand higher peak currents.

Table 4.2 summarizes the characteristics of the most commonly used LED materials.

4.6.4 LED construction

A typical LED construction is shown in Fig. 4.16. It is obviously advantageous if most of the radiative recombinations take place from the side of the junction nearest the surface, since then the chances of re-absorption are lessened. We may ensure this by arranging that most of the current flowing across the diode is carried by those carriers that are injected into the surface layer. We assume that, as in Fig. 4.16, the surface

population density Δp gives rise to $\Delta p/\tau_c$ recombinations per second per unit volume, we must have

$$\Delta p = \frac{J\tau_c}{et} \quad (4.17)$$

Eliminating Δp from eqs. (4.16) and (4.17), we obtain

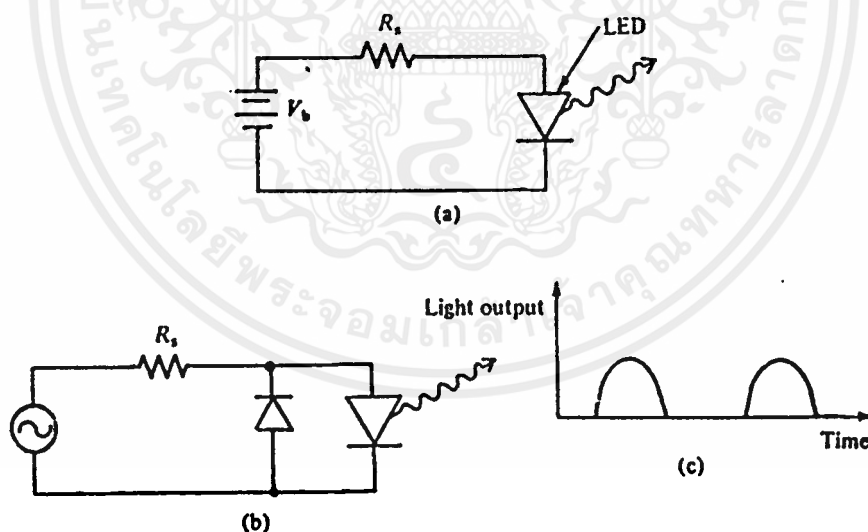
$$\tau_c = \left(\frac{et}{JB}\right)^{1/2} \quad (4.18)$$

We see that in this case τ_c may be reduced by reducing t and increasing J . However, since the lifetime is now current dependent, this approach can lead to signal distortion.

4.6.6 LED drive circuitry

As we have seen, the electrical characteristics of LEDs are essentially those of ordinary rectifying diodes. Typical operating currents are between 20 mA and 100 mA, whilst the forward voltages vary from 1.2 V for GaAs to 2 V for GaP. (The operating voltage is approximately equal to the built-in diode potential, which in turn is slightly less than the energy gap expressed in eV.) Simple drive circuits for d.c. and a.c. voltage operation are shown in Figs. 4.19(a) and (b) respectively. The current through the diode is limited by a series resistance R_s whose value may be calculated from

$$R_s = \frac{V_b - V_d}{i_d} \quad (4.19)$$



population density Δp gives rise to $\Delta p/\tau_c$ recombinations per second per unit volume, we must have

$$\Delta p = \frac{J\tau_c}{el} \tag{4.17}$$

Eliminating Δp from eqs. (4.16) and (4.17), we obtain

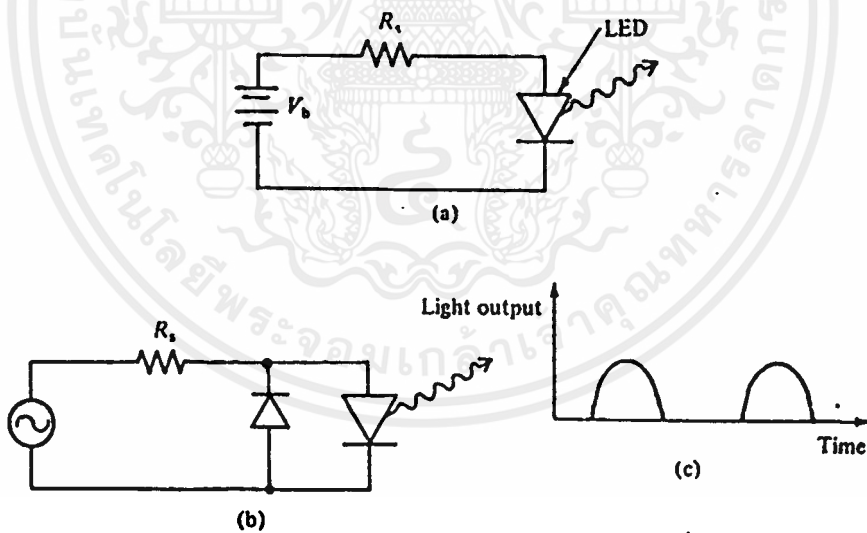
$$\tau_c = \left(\frac{el}{JB}\right)^{1/2} \tag{4.18}$$

We see that in this case τ_c may be reduced by reducing l and increasing J . However, since the lifetime is now current dependent, this approach can lead to signal distortion.

4.6.6 LED drive circuitry

As we have seen, the electrical characteristics of LEDs are essentially those of ordinary rectifying diodes. Typical operating currents are between 20 mA and 100 mA, whilst the forward voltages vary from 1.2 V for GaAs to 2 V for GaP. (The operating voltage is approximately equal to the built-in diode potential, which in turn is slightly less than the energy gap expressed in eV.) Simple drive circuits for d.c. and a.c. voltage operation are shown in Figs. 4.19(a) and (b) respectively. The current through the diode is limited by a series resistance R_s , whose value may be calculated from

$$R_s = \frac{V_b - V_d}{i_d} \tag{4.19}$$



Features

- Compatible with MCS-51™ Products
- 2 Kbytes of Reprogrammable Flash Memory
Endurance: 1,000 Write/Erase Cycles
- 2.7 V to 6 V Operating Range
- Fully-Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Five Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes

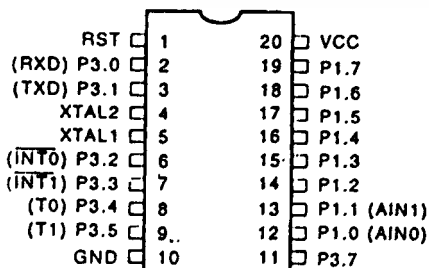
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2 Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2 Kbytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five source two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Configuration

PDIP/SOIC



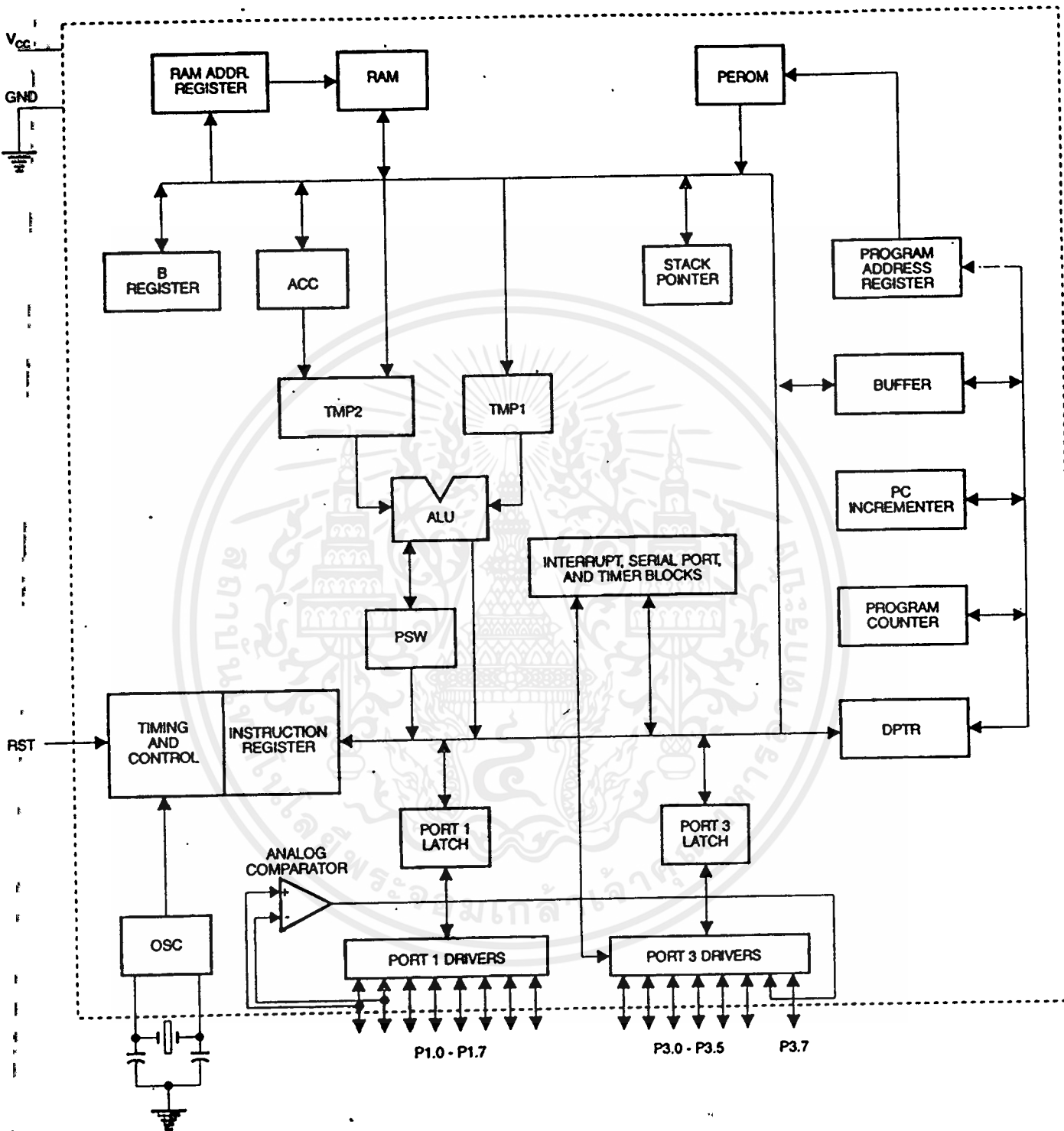
8-Bit
Microcontroller
with 2 Kbytes
Flash

AT89C2051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา หรือส่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Pin Description

Vcc
Supply voltage.

GND
Ground.

Port 1
Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins; they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (IIL) because of the internal pullups.

Port 1 also receives code data during Flash programming and program verification.

Port 3
Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. 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 (IIL) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

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

RST
Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

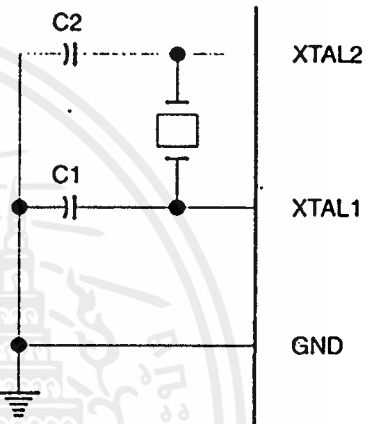
XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier.

Oscillator Characteristics

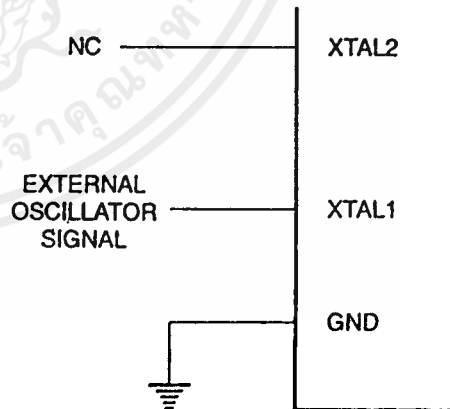
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



Special Function Registers

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

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

Table 1. AT89C2051 SFR Map and Reset Values

0F8H							
0F0H	B 00000000						
0E8H							
0E0H	ACC 00000000						
0D8H							
0D0H	PSW 00000000						
0C8H							
0C0H							
0B8H	IP X0000000						
0B0H	P3 11111111						
0A8H	IE 00000000						
0A0H							
98H	SCON 00000000	SBUF XXXXXXXX					
90H	P1 11111111						
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	
80H		SP 00000111	DPL 00000000	DPH 00000000			PCON 00000000

Restrictions on Certain Instructions

The AT89C2051 and AT89C1051 are economical and cost-effective members of Atmel's growing family of microcontrollers. They contain 2 Kbytes and 1 Kbyte of flash program memory, respectively. Both are fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program these devices.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LCALL 7E0H would be valid on the AT89C2051 (with 2K of memory), but LCALL 900H would not.

Branching instructions:

CALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the AT89C2051). Violating the physical space limits may cause unknown program behavior.

JNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above ap-

plies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory, while the AT89C1051 has 64 bytes. Thus, in the AT89C1051 the stack depth is limited to 64 bytes, the amount of available RAM. External DATA memory access is not supported in either device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

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

Lock Bit Protection Modes⁽¹⁾

Program Lock Bits	Lock Bits		Protection Type
	LB1	LB2	
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation

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.

PI.0 and PI.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

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

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. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before VCC is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

PI.0 and PI.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

Programming The Flash

The AT89C2051 is shipped with the 2 Kbytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
Apply power between VCC and GND pins
Set RST and XTAL1 to GND
With all other pins floating, wait for greater than 10 milliseconds
 2. Set pin RST to 'H'
Set pin P3.2 to 'H'
 3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
- To Program and Verify the Array:
4. Apply data for Code byte at location 000H to P1.0 to P1.7.
 5. Raise RST to 12V to enable programming.
 6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
 7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropriate levels. Output data can be read at the port P1 pins.
 8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
 9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2 Kbytes array or until the end of the object file is reached.
 10. Power-off sequence:
set XTAL1 to 'L'
set RST to 'L'
Float all other I/O pins
Turn Vcc power off

Data Polling: The AT89C2051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2 Kbytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (000H) = 1EH indicates manufactured by Atmel
- (001H) = 21H indicates 89C2051

Programming Interface

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

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

Flash Programming Modes

Mode	RST	P3.2/ PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data ^(1,3)	12V		L	H	H	H
Read Code Data ⁽¹⁾	H	H	L	L	H	H
Write Lock Bit - 1	12V		H	H	H	H
Bit - 2	12V		H	H	L	L
Chip Erase	12V	⁽²⁾	H	L	L	L
Read Signature Byte	H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.

2. Chip Erase requires a 10 ms PROG pulse.
3. P3.1 is pulled Low during programming to indicate RDY/BSY.

Figure 3. Programming the Flash Memory

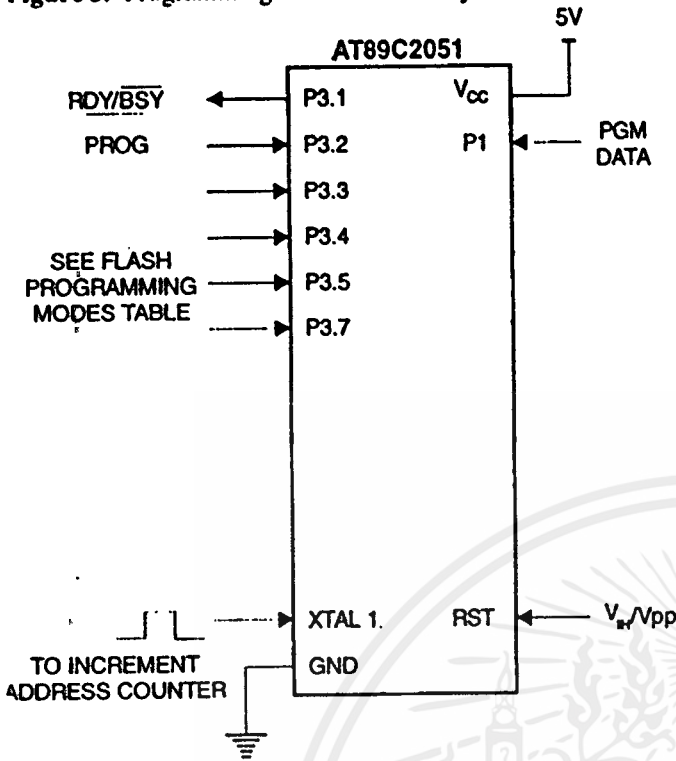
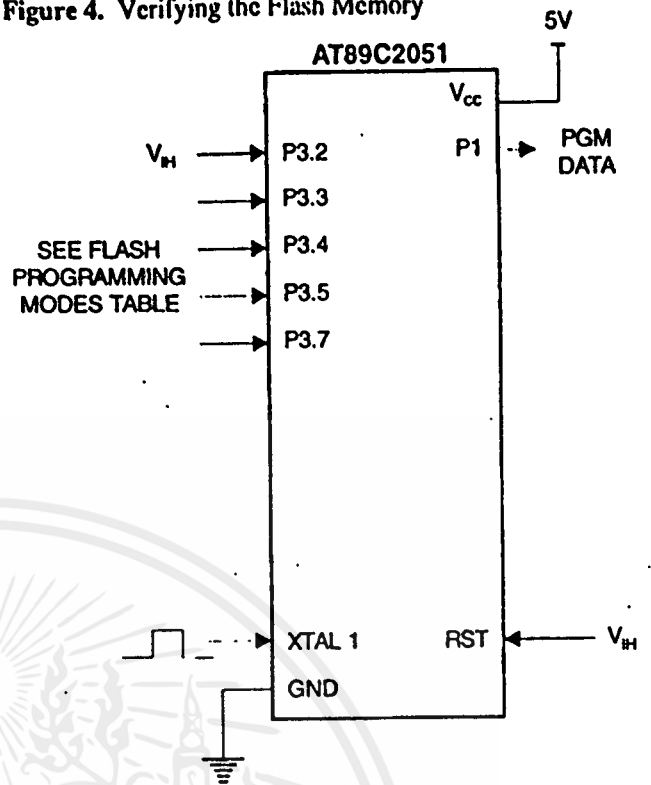


Figure 4. Verifying the Flash Memory



Flash Programming and Verification Characteristics

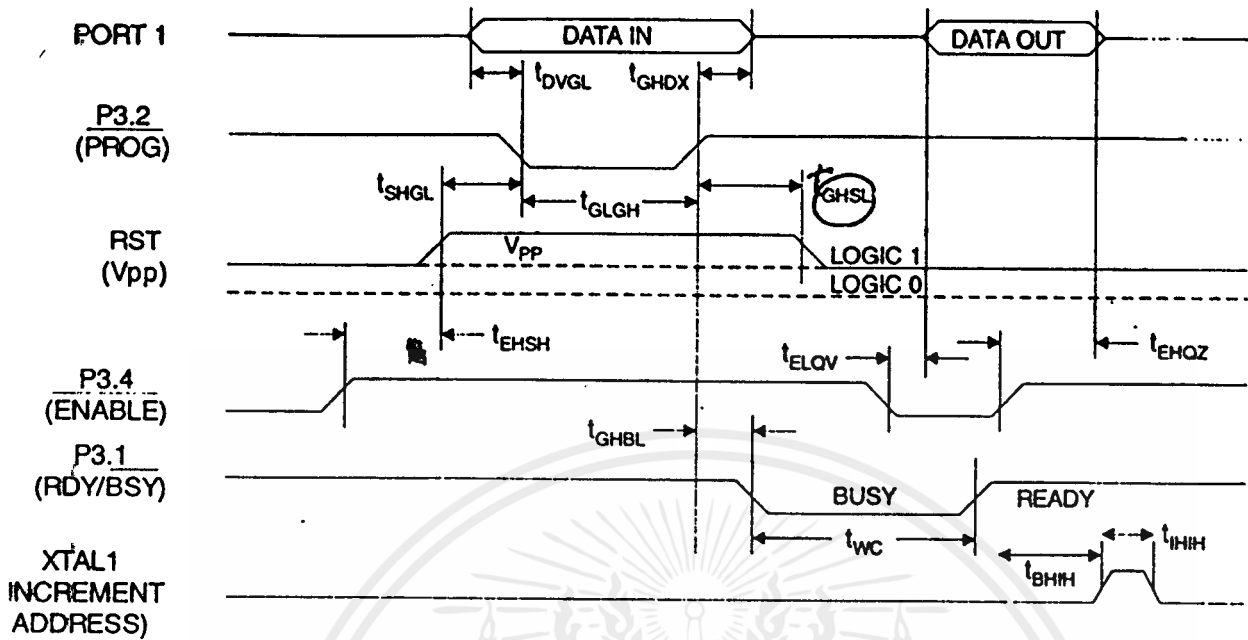
$T_A = 21^\circ\text{C to } 27^\circ\text{C}$, $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	μa
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	1.0		μs
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	1.0		μs
t_{EHSH}	P3.4 ($\overline{\text{ENABLE}}$) High to V_{PP}	1.0		μs
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		1.0	μs
t_{EHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	1.0	μs
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	$\overline{\text{RDY/BSY}}$ to Increment Clock Delay	1.0		μs
t_{IHIL}	Increment Clock High	200		ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms



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	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current.....	25.0 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.



D.C. Characteristics

$T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 2.7\text{ V}$ to 6.0 V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage		-0.5	$0.2 V_{CC}-0.1$	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, 3)	$I_{OL} = 20\text{ mA}$, $V_{CC} = 5\text{ V}$ $I_{OL} = 10\text{ mA}$, $V_{CC} = 2.7\text{ V}$		0.5	V
V_{OH}	Output High Voltage (Ports 1, 3)	$I_{OH} = -80\ \mu\text{A}$, $V_{CC} = 5\text{ V} \pm 10\%$	2.4		V
		$I_{OH} = -30\ \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12\ \mu\text{A}$	$0.9 V_{CC}$		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)	$V_{IN} = 2\text{ V}$		-750	μA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		± 10	μA
V_{OS}	Comparator Input Offset Voltage	$V_{CC} = 5\text{ V}$		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
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, $V_{CC} = 6\text{ V}/3\text{ V}$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6\text{ V}/3\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power Down Mode ⁽²⁾	$V_{CC} = 6\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		100	μA
		$V_{CC} = 3\text{ V}$ P1.0 & P1.1 = 0V or V_{CC}		20	μA

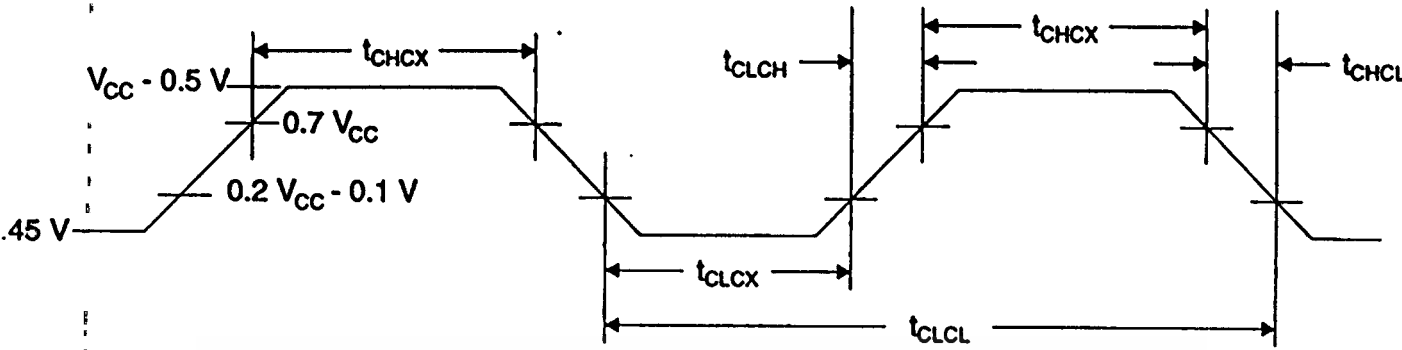
Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 20 mA
Maximum total I_{OL} for all output pins: 80 mA

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 2 V.

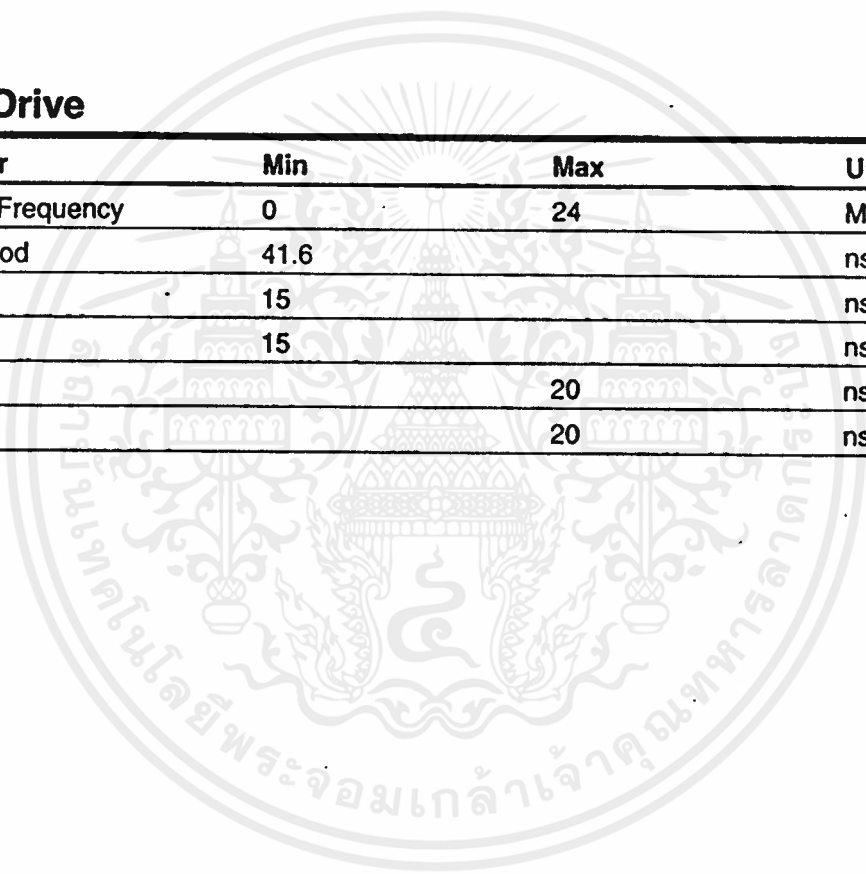
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
1/CLCL	Oscillator Frequency	0	24	MHz
tCLCL	Clock Period	41.6		ns
tCHCX	High Time	15		ns
tCLCX	Low Time	15		ns
tCLCH	Rise Time		20	ns
tCHCL	Fall Time		20	ns



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น จึงขอสงวนสิทธิ์ในเอกสารฉบับนี้ และขอสงวนลิขสิทธิ์ในเอกสารฉบับนี้ไว้ด้วย



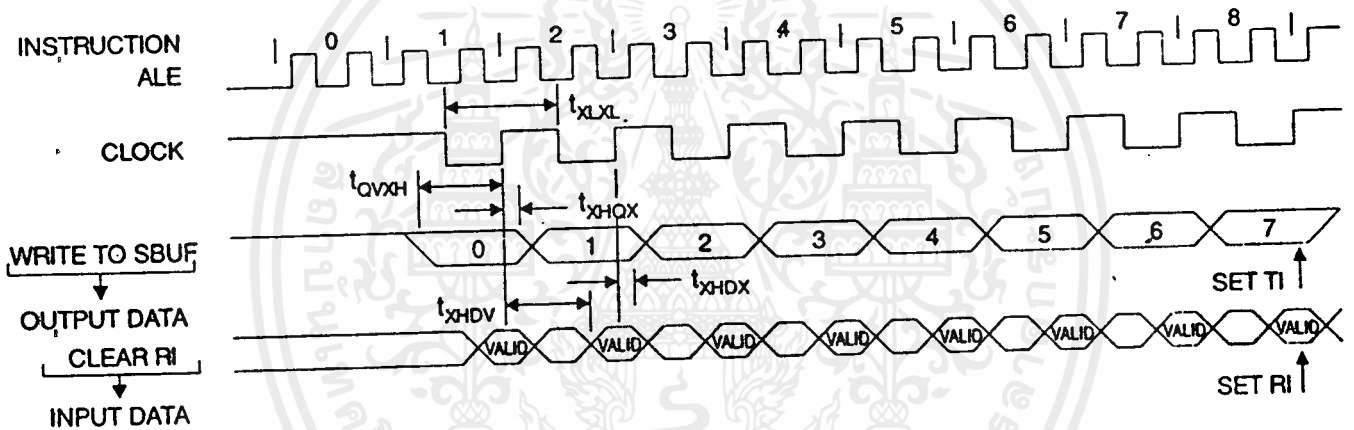


Serial Port Timing: Shift Register Mode Test Conditions

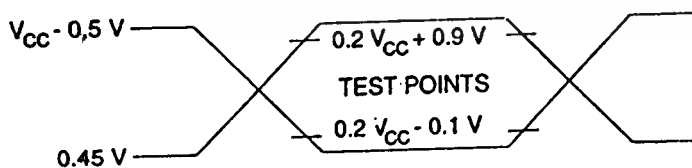
$V_{CC} = 5.0 \text{ V} \pm 20\%$; Load Capacitance = 80 pF)

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

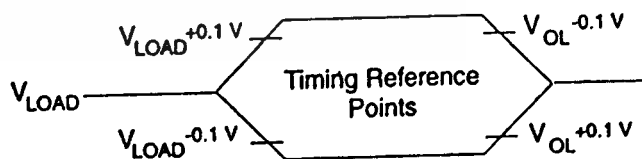
Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms⁽¹⁾ Float Waveforms⁽¹⁾



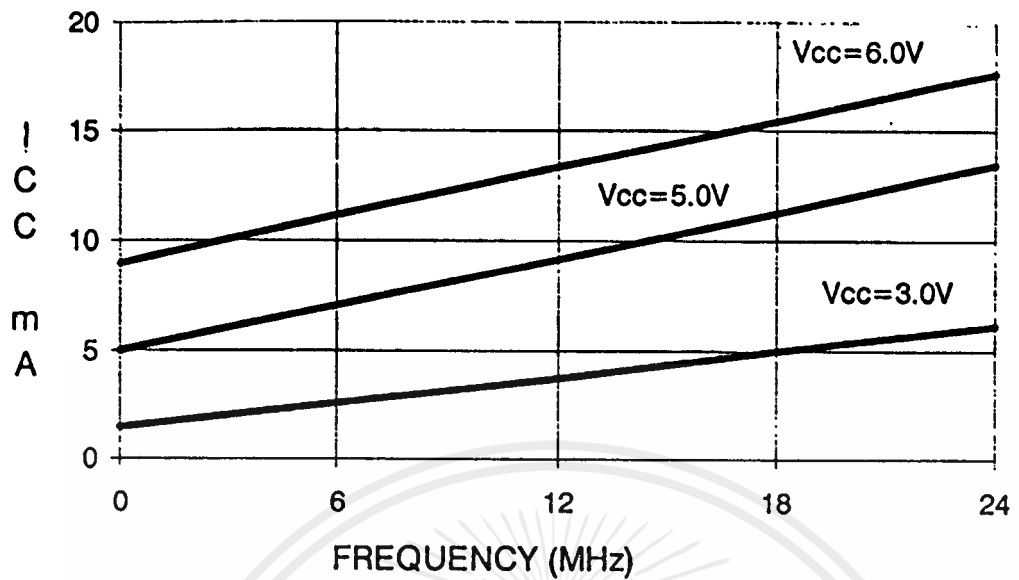
Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5 \text{ V}$ for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at $V_{IH \text{ min.}}$ for a logic 1 and $V_{IL \text{ max.}}$ for a logic 0.



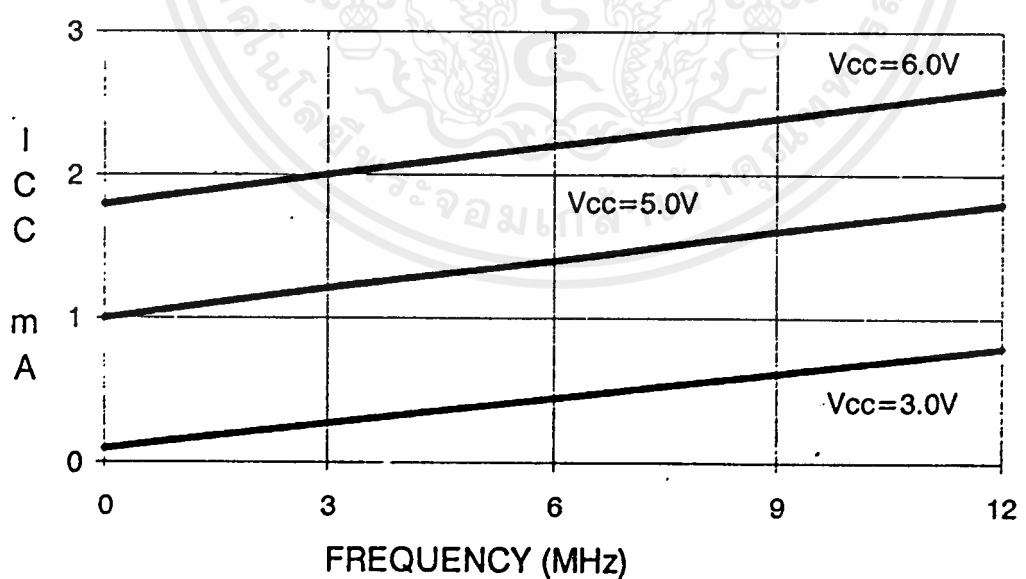
Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89C2051 TYPICAL ICC - ACTIVE (85°C)

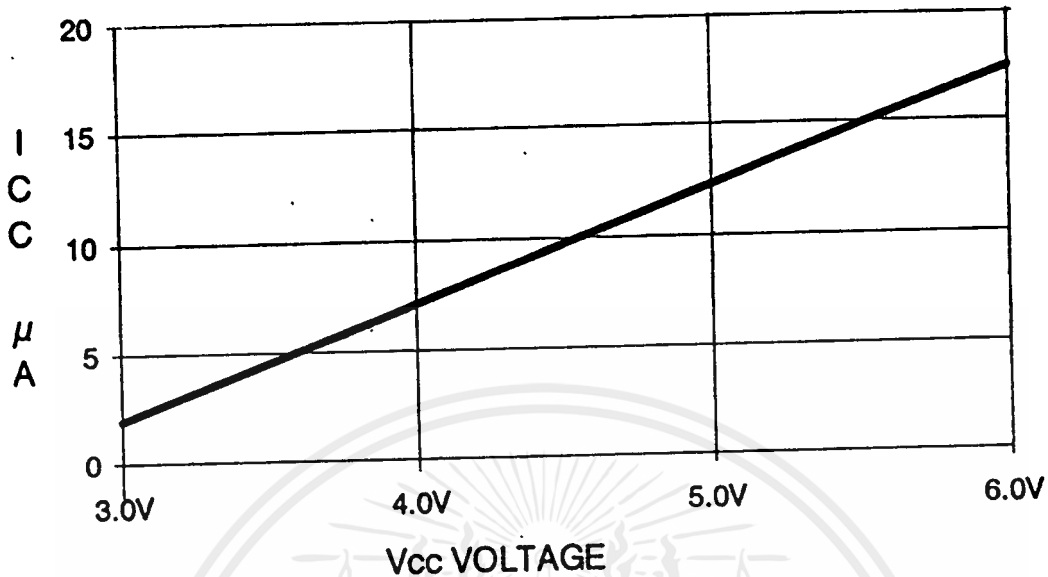


AT89C2051 TYPICAL ICC - IDLE (85°C)

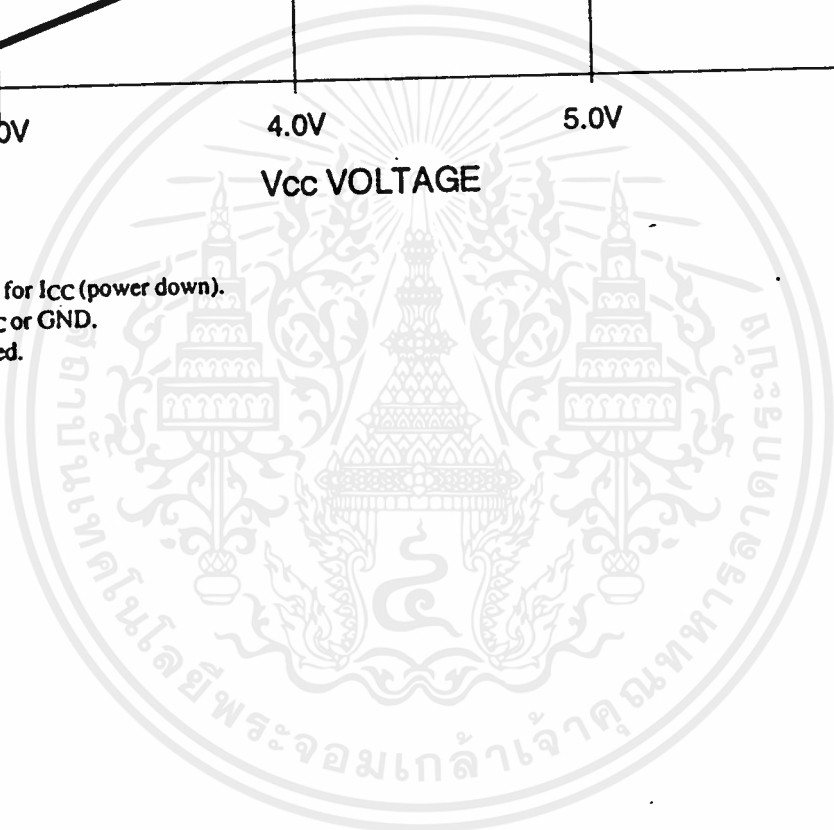


AT89C2051

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



1. XTAL1 tied to GND for ICC (power down).
2. P.1.0 and P.1.1 = VCC or GND.
3. Lock bits programmed.



Ordering Information

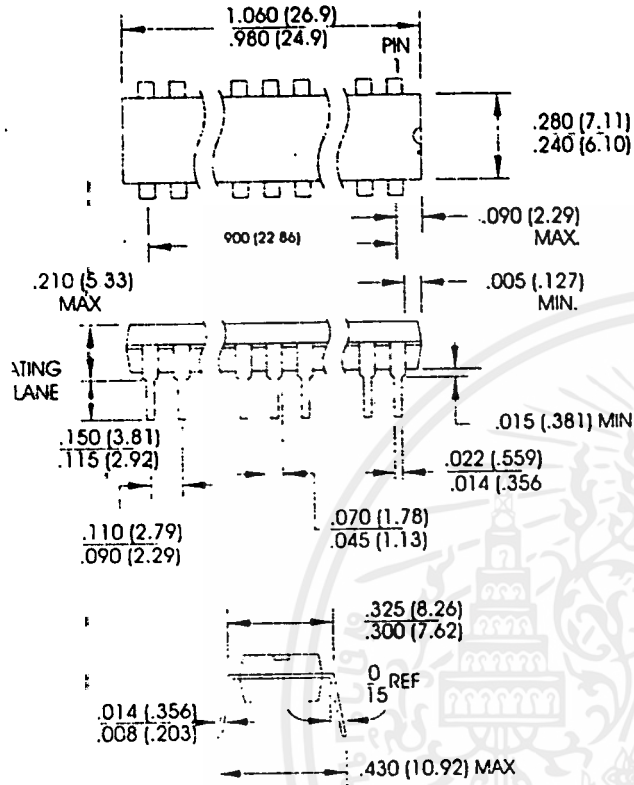
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V to 6.0 V	AT89C2051-12PC AT89C2051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-12PI AT89C2051-12SI	20P3 20S	Industrial (-40°C to 85°C)
24	4.0 V to 6.0 V	AT89C2051-24PC AT89C2051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-24PI AT89C2051-24SI	20P3 20S	Industrial (-40°C to 85°C)



Package Type	
3	20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

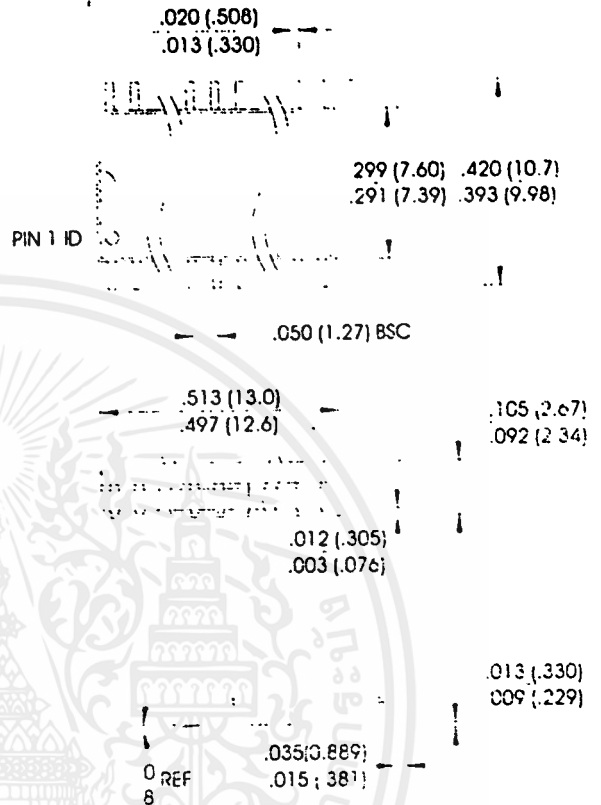
Packaging Information

P23, 20 Lead, 0.300" Wide,
Plastic Dual Inline Package (PDIP)



Controlling dimension: millimeters

20S, 20 Lead, 0.300" Wide,
Plastic Gull Wing Small Outline (SOIC)



บรรณานุกรม

1. อ.ธีรวัฒน์ ประกอบผล “ การเชื่อมต่อไมโครคอนโทรลเลอร์ ” , พิมพ์ครั้งที่ 1 , หน้า 1-29, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2539.
2. สมโภชน์ อิมเอิบ, เสาวณีย์ รักรวิชัย “ ฟิสิกส์ทั่วไป 1 ” ภาควิชาวิทยาศาสตร์ คณะครุศาสตร์ อุดสาหกรรมและวิทยาศาสตร์ พระจอมเกล้า พระนครเหนือ.
3. สุนทร วิฑูรพจน์ “ การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 ” , พิมพ์ครั้งที่ 1, หน้า 63-72, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), กรุงเทพมหานคร, พ.ศ. 2537.
4. นายแพทย์สุรเกียรติ์ อาชานานุภาพ “ ตำราตรวจรักษาโรคทั่วไป 2 : 250 โรคและการดูแลรักษา ” , พิมพ์ครั้งที่ 2, หน้า 755-756, สำนักพิมพ์หมอชาวบ้าน, กรุงเทพมหานคร, พ.ศ. 2532.
5. Pidrotti, F.L., and L.S. Pedrotti. Introduction to optics, Prentice-Hall, 1st ed. 1987, 2nd ed. 1993.