สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องล้างผัก

VEGETABLE WASHING MACHINE

T042628

โดย

นาย จักรพันธ์ จันทนา
นาย ธีระวุฒิ จินายิ้ม
นาย ไมตรี เพ็ชรเกลี้ยง

เลขหมู่ TJ225 .D2 2543
เลขทะเบียน 42628
วัน, เดือน, ปี 5 มิ.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องล้างผัก

VEGETABLE WASHING MACHINE

ชื่อนักศึกษา นาย จักรพันธ์ จันทนา 41013244

นาย ธีระวุฒิ จินายิ้ม 41013251

นายไมตรี เพ็ชรเกลี้ยง 41013263

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(ผศ. พิชัย คูศิริวานิชกร)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องล้างผัก

VEGETABLE WASHING MACHINE

นาย จักรพันธ์ จันทนา 41013244

นาย ธีระวุฒิ จินายิ้ม 41013251

นาย ไมตรี เพ็ชรเกลี้ยง 41013263

รายงานฉบับนี้ได้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

.....

(ผศ. พิชัย คูศิริวานิชกร)

30 / 3 / 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ ผศ. พิชัย คูศิริวานิชกร ที่ให้โอกาสทำโครงงานและให้คำปรึกษา จึงขอขอบพระคุณท่านอาจารย์เป็นอย่างสูง และขอขอบคุณ คุณวิรัตน์ ทองจันทร์แก้ว รวมถึงเพื่อนๆที่ให้ยืมอุปกรณ์ในการทำงาน ที่ให้คำแนะนำที่ดีและคอยเป็นกำลังใจมาโดยตลอด ทำให้โครงงานชิ้นนี้สำเร็จลุล่วงดังที่ตั้งเป้าหมายเอาไว้

จักรพันธ์ จันทนา
(นายจักรพันธ์ จันทนา)

ธีระวุฒิ จันยิ้ม
(นายธีระวุฒิ จินายิ้ม)

ไมตรี เพ็ชรเกลี้ยง
(นายไมตรี เพ็ชรเกลี้ยง)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องล้างผัก

นายจักรพันธ์ จันทนา
นายธีระวุฒิ จินายิ้ม
นายไมตรี เพ็ชรเกลี้ยง
ผศ. พิชัย คูศิริวานิชกร (อาจารย์ที่ปรึกษา)

บทคัดย่อ

เครื่องล้างผักเป็นอุปกรณ์ที่ใช้ในครัวเรือนหรือร้านอาหาร เนื่องจากผู้ที่ใช้สามารถใส่ผักไว้แล้วจากนั้นให้เครื่องทำงานเองโดยอัตโนมัติ และผู้ใช้สามารถไปทำภารกิจอย่างอื่นได้ หลังจากที่เครื่องทำการล้างผักเสร็จสิ้นแล้ว จะมีเสียงเตือนเพื่อบอกว่าขณะนี้ เครื่องได้ทำการล้างผักเสร็จสิ้นตามที่เราได้โปรแกรมไว้เรียบร้อยแล้ว

หลักการทำงานของเครื่องล้างผักนั้น คือนำผักที่ต้องการที่จะทำความสะอาดใส่ลงไปในเครื่องซึ่งในเครื่องล้างผักนั้นจะมีภาชนะ สำหรับรองรับผักที่จะล้างทำความสะอาด หลังจากนั้นทำการตั้งโปรแกรมทำงานตามที่เราต้องการ ว่าต้องการจะตั้งเวลาล้างเป็นเวลานานเท่าไหร่จำนวนกี่ครั้งที่จะทำการล้างและปริมาณของผักที่จะล้างนั้นมากน้อยแค่ไหนเราก็สามารถ โปรแกรมได้ตามที่เราต้องการหลังจากนั้นก็กดปุ่มเริ่มให้เครื่องทำงาน ระบบน้ำไหลเวียนภายในเครื่องจะชะล้างทำความสะอาดผัก หลังจากนั้นเครื่องจะทำการ ถ่ายน้ำนั้นทิ้งและเครื่องปล่อยน้ำใหม่เข้ามาทำการล้างครั้งใหม่จนเสร็จตามโปรแกรมที่เราได้ตั้งไว้

ในการควบคุมการทำงานของเครื่องนั้นจะใช้ ไมโครคอนโทรเลอร์ของตระกูล MCS- 51ในการควบคุมระบบทั้งหมดโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VEGETABLE WASHING MACHINE

Mr.Jakapan Jantana

Mr.Theerawut Jinayim

Mr.Maitree Phechklieng

Asst.Prof. Pichai Kusirivanichakorn(Advisor)

Abstract

Vegetable washing machine can be used in restaurant and home. This machine is convenient for the user because it can set to automatic modes after the user put the vegetable inside and then the user can do something else. There is a warning sound to alarm the user when the vegetable has been washed.

The operations of the vegetable washing machine can be described as follows First, put the unwashed vegetables into the sieve machine. Second, set the timer, how many times, and how long that you want to wash the vegetable, which depend on the quantity of vegetable. Then, the user can press the start bottom and the machine will stop itself. The water ventilation system will take all dusts, and chemical from the vegetable ,then the system will drain the waste water .

MCS-51 is used for automatic control of the operations of the machine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

กิตติกรรมประกาศ I

บทคัดย่อ II

Abstract III

สารบัญ IV

สารบัญภาพ V

สารบัญตาราง VI

บทที่ 1 บทนำ 1

บทที่ 2 การออกแบบ 3

- แบบแสดงลักษณะโครงสร้างของเครื่อง 3
- Block Diagram ของการควบคุมเครื่องล้างผัก 6
- การทำงานของเครื่อง 7
- การควบคุมการทำงานของส่วนต่าง ๆ 8
- หน้าปัดและการใช้งานของปุ่มต่าง ๆ ของเครื่องล้างผัก 10

บทที่ 3 การออกแบบระบบควบคุมการทำงาน 11

- คุณสมบัติของไมโครคอนโทรลเลอร์ 11
- การติดต่ออุปกรณ์ภายนอก 19
- การเอาไมโครคอนโทรลเลอร์ ไปใช้งาน 19
- บล็อกไดอะแกรมการควบคุมของเครื่อง 20
- วงจรสวิตช์การควบคุมต่างๆของเครื่อง 20
- วงจร Sensor ระดับน้ำ 23
- วงจรแสดงผลทาง 7-Segment ของเครื่องล้างผัก 24
- วงจรขับ Pump และวาล์วไฟฟ้า 25
- วงจรเสียง 26

บทที่ 4 ซอฟต์แวร์ควบคุมการทำงานของเครื่องล้างผัก 27

- การกำหนด Port ต่างๆเป็นอินพุทและเอาท์พุทของ MCS-51 27
- โฟล์วชาร์ทการทำงานของซอฟต์แวร์การควบคุมการทำงานของเครื่องล้างผัก... 28

บทที่ 5 สรุปและวิจารณ์ผลการทดลอง 33

หนังสืออ้างอิง 34

ภาคผนวก 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 2.1 แสดงภาพรวมภายนอก 4
รูปที่ 2.2 แสดงการปล่อยน้ำไหลเข้าตัวถัง 4
รูปที่ 2.3 แสดงการทำงานตามระยะเวลาที่ได้กำหนด 5
รูปที่ 2.4 แสดงการปล่อยน้ำทิ้งออกจากตัวถัง 5
รูปที่ 2.5 แสดงบล๊อกไดอะแกรมควบคุมเครื่องล้างผัก 6
รูปที่ 2.6 แสดงหน้าจอของส่วนควบคุมทั้งหมด 10
รูปที่ 3.1 แสดงโครงสร้างของพอร์ต 0 12
รูปที่ 3.2 แสดงโครงสร้างของพอร์ต 1 13
รูปที่ 3.3 แสดงโครงสร้างของพอร์ต 2 13
รูปที่ 3.4 แสดงโครงสร้างของพอร์ต 3 14
รูปที่ 3.5 แสดงหน่วยความจำของโปรแกรมและหน่วยความจำของข้อมูลตามลำดับ 15
รูปที่ 3.6 แสดงหน่วยความจำ 128 ไบต์ของแรมที่เข้าถึงข้อมูลแบบทางตรงและทางอ้อม 17
รูปที่ 3.7 แสดงการประยุกต์ใช้ไมโครคอนโทรลเลอร์ 19
รูปที่ 3.8 แสดงบล๊อกไดอะแกรมควบคุมเครื่องล้างผัก 20
รูปที่ 3.9 แสดงวงจรสวิตซ์ 1-5 21
รูปที่ 3.10 แสดงลักษณะการต่อ Nor Gate เข้าขา interrupt 0 ของ MCS-51 22
รูปที่ 3.11 แสดงวงจรสวิตซ์ 6-8 22
รูปที่ 3.12 แสดงวงจรการ Sensor ระดับน้ำของเครื่องล้างผัก 23
รูปที่ 3.13 แสดงวงจรการต่อวงจรแสดงการตั้งเวลา 24
รูปที่ 3.14 แสดงการขับปั๊มและวาลว์ 25
รูปที่ 3.15 แสดงวงจรเสียง 26
รูปที่ 4.1 แสดงโฟล์วชาร์ทโปรแกรมหลัก 1 29
รูปที่ 4.2 แสดงโฟล์วชาร์ทโปรแกรมหลัก 2 30
รูปที่ 4.3 แสดงโฟล์วชาร์ทโปรแกรม interrupt 1 31
รูปที่ 4.4 แสดงโฟล์วชาร์ทโปรแกรม interrupt 0 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงหน่วยความจำของรีจิสเตอร์ 16
ตารางที่ 3.2 แสดงรีจิสเตอร์ที่มีการใช้งานเฉพาะ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของหัวข้อโครงการ

ในปัจจุบันในการปลูกผักของชาวสวน ได้ใช้สารพิษและยาปราบศัตรูพืชเพื่อให้ผลผลิตของชาวสวนมีความสวยงามเป็นที่ต้องการของตลาดทำให้ผลผลิตมีสารพิษ และนอกจากนี้ภายในผักหรือผลผลิตของชาวสวนยังมีเชื้อโรคที่ติดมากับดินทำให้ผักมีแต่สารมีพิษ และเนื่องด้วยในปัจจุบันผู้บริโภคมีความสนใจในสุขภาพอนามัย จึงหันมาบริโภคผักแทนเนื้อสัตว์มากขึ้นดังนั้นในการบริโภคจึงต้องมีการล้างเสียก่อนจะนำมาบริโภคหรือประกอบอาหารเพื่อกำจัดสารพิษและสิ่งสกปรกต่างๆภายในผักและในอนาคตเครื่องล้างผักน่าจะเป็นเครื่องครัวที่จำเป็นอย่างหนึ่ง เช่นเดียวกับหม้อหุงข้าวหรือเครื่องซักผ้าเพราะเครื่องล้างผักจะช่วยประหยัดเวลาในการประกอบอาหารในครัวเรือน ดังนั้นผู้ดำเนินโครงการจึงคิดที่จะสร้างเครื่องล้างผักต้นแบบ ขึ้นมาโดยทำการคิดค้นอุปกรณ์ที่จะมาทำเครื่องล้างผักโดยนำความรู้ทางอิเล็กทรอนิกส์มาประยุกต์ในการควบคุมการทำงานของเครื่องเพื่อใช้ในการล้างผักเหล่านั้น

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อประยุกต์ความรู้ทางด้านอิเล็กทรอนิกส์และไมโครคอนโทรเลอร์มาใช้งานจริง
2. เพื่อทำให้ผู้ประกอบอาหารไม่ต้องเสียเวลาในการล้างผักเอง
3. สามารถที่จะเป็นต้นแบบในการออกแบบประยุกต์ใช้ในระดับต่อไป

1.3 เนื้อหาของโครงงาน

- การออกแบบระบบ
- แบบแสดงลักษณะโครงสร้างของตัวเครื่อง
- การทำงานของเครื่อง
- การควบคุมการทำงานของส่วนต่างๆ
- การออกแบบระบบควบคุมการทำงาน
- คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51
- บล็อกไดอะแกรมการควบคุมเครื่อง
- วงจร Sensor ระดับน้ำ
- วงจรแสดงผลทาง 7- Segments ของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วงจรขับ Pump & Value
- Software ควบคุมการทำงานของเครื่องล้างผัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การออกแบบระบบ

ในบทนี้จะได้กล่าวถึงในส่วนของระบบการควบคุม การไหลของน้ำภายในตัวเครื่องระบบการไหลของน้ำภายในตัวเครื่องนั้น จะเป็นแบบระบบน้ำไหลเวียน ซึ่งหมายความถึงหลังจากที่ได้มีน้ำไหลเข้าไปในตัวเครื่องแล้ว จะนำน้ำในส่วนนั้นใช้ในการทำความสะอาดผักและในส่วนนี้ยังได้กล่าวถึงระบบควบคุมการจ่ายน้ำเข้าสู่ระบบของตัวเครื่องนั่นเอง ซึ่งในการควบคุมการจ่ายน้ำมีความสำคัญต่อการออกแบบในส่วนของการควบคุม กลไกลในส่วนระบบการควบคุมการจ่าย น้ำจากการล้างนั้นทิ้ง ในส่วนของระบบการฉีดน้ำเพื่อทำความสะอาดผักนั้นจะใช้ในส่วนของมอเตอร์พร้อมด้วยปั้มในการเพิ่มในส่วนของแรงดันน้ำ หากว่าเราต้องการเพิ่มความสะอาดในการล้างก็สามารถที่จะทำได้โดยการเพิ่มในส่วนของแรงดันน้ำนั่นเอง

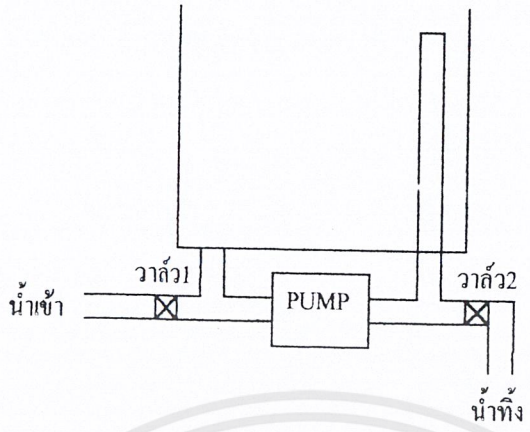
จากระบบการจ่ายน้ำภายในเครื่องและแรงดันของน้ำจากปั๊มที่เราใช้น้ำไหลเวียนภายในเครื่องจะทำการเปลี่ยนแปลงตำแหน่งของผักเพื่อเพิ่มความสะอาดในการล้างมากยิ่งขึ้น สำหรับในส่วนของผักที่ต้องการจะล้างนั้น จะมีภาชนะสำหรับรองรับผักอยู่แล้ว ภาชนะในส่วนนี้จะมีลักษณะเป็นตะแกรง เพื่อให้น้ำสามารถที่จะฉีดไปถูกผักได้อย่างทั่งถึงและเพิ่มความสะอาดในการล้างนั่นเอง

2.1 แบบแสดงลักษณะโครงสร้างของตัวเครื่อง

ดังที่ได้กล่าวไว้แล้วว่าในการทำความสะอาดหรือการล้างผักนั้น จะใช้ในส่วนของระบบน้ำไหลวนดังนั้นในการออกแบบตัวถัง ของเครื่องจะเป็นไปในลักษณะของรูปถังที่สามารถที่จะรองรับน้ำได้ซึ่งภาชนะส่วนนี้พลาสติกที่มีความแข็งแรง และสามารถที่จะรองรับน้ำหนักของน้ำได้ ภาชนะส่วนนี้จะตั้งอยู่บนโครงสร้างของเหล็กซึ่งสามารถที่จะเคลื่อนย้ายได้โดยสะดวกนั่นเอง

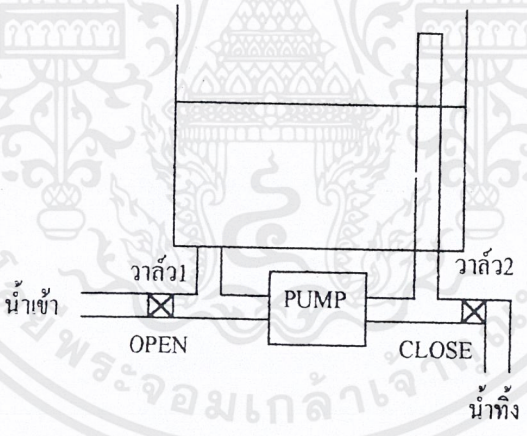
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อท่อให้น้ำวน



รูปที่ 2.1 แสดงภาพรวมภายนอก

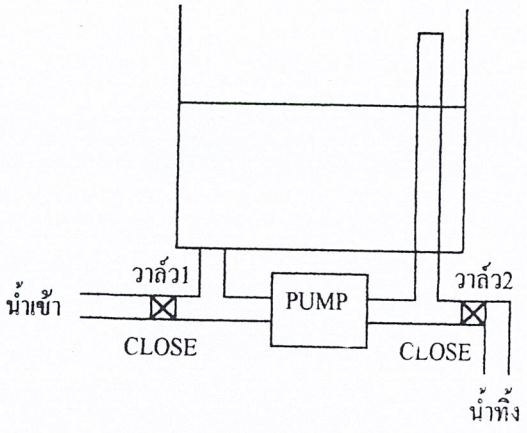
1. วาล์ว ตัวที่ 1 จะ เปิดให้น้ำเข้ามาในถังให้ ได้ระดับที่เราเลือกไว้ แล้ววาล์ว ตัวที่ 1 ปิด เมื่อถึงระดับที่ต้องการในขณะเดียวกัน วาล์วตัวที่ 2 ก็ปิดอยู่ ดังรูป



รูปที่ 2.2 แสดงการปล่อยน้ำไหลเข้าตัวถัง

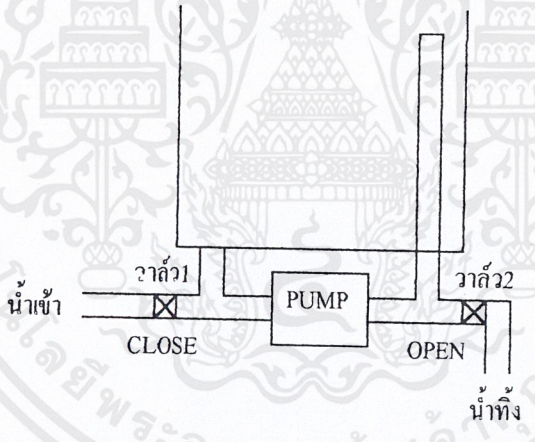
2. ขั้นที่ 2 วาล์ว ตัวที่ 1 , 2 จะ ปิด และ Pump จะเริ่มทำงาน โดยจะดูดน้ำในถังให้วน เมื่อทำการล้างตามที่เราตั้งเวลาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงการทำงานตามระยะเวลาที่ได้กำหนด

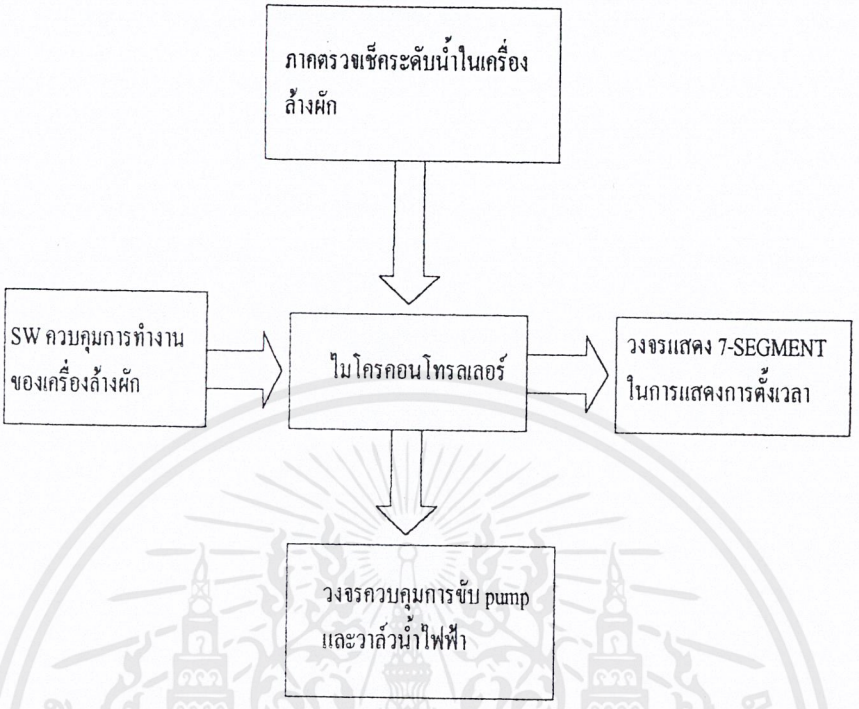
3. ขั้นที่ 3 ช่วงนี้ เราเอาน้ำทิ้งออกจากตัวถัง เราให้น้ำวนในถังเพื่อล้างผักตามเวลาที่ตั้งไว้แล้วเราจะเอาน้ำทิ้งโดยวาลว์ ตัวที่ 1 ปิดและไปควบคุมให้ไมโครคอนโทรเลอร์ เปิดวาล์วตัวที่ 2 เพื่อทิ้งน้ำและให้ Pump ทำงานอยู่เพื่อจะได้ให้ Pump ดูดน้ำออกได้เร็วขึ้น



รูปที่ 2.4 แสดงการปล่อยน้ำทิ้งออกจากตัวถัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Block Diagram ของการควบคุมเครื่องล้างผัก



รูปที่ 2.5 แสดงบล๊อกไดอะแกรมการควบคุมเครื่องล้างผัก

จากบล็อกไดอะแกรมเราสามารถอธิบายแต่ละบล็อกได้ดังนี้

1. Switch ควบคุมการทำงานเมื่อมีการกดสวิตซ์ตัวใดตัวหนึ่งเพื่อให้ทำงานตามที่เราต้องการจะมีการส่งสัญญาณไปให้ไมโครคอนโทรเลอร์ ประมวลผล ทำให้ทำอะไรขึ้นอยู่กับที่เราเขียนโปรแกรมไว้ โดย Switch มีทั้งหมด 8 ตัว โดย Switch ตัวที่ 1 และ Switch ตัวที่ 2 เป็นตัวเพิ่ม ตัวลด ในการล้างผัก Switch ตัวที่ 3 เป็นจำนวนครั้งในการล้าง Switch ตัวที่ 4 เป็นระดับน้ำ Switch ตัวที่ 5 เป็นการ start การล้างผัก Switch ตัวที่6 มีไว้เมื่อเราต้องการล้างผักเอง Switch ตัวที่7 เป็นโปรแกรมที่1 สำหรับการล้างผักน้อย Switch ตัวที่8 เป็นโปรแกรมที่2 สำหรับการล้างผักมาก

2. ตัวการตรวจระดับน้ำโดยจะส่งข้อมูลไปที่ ไมโครคอนโทรเลอร์ให้หยุดให้น้ำเข้าเมื่อได้ระดับที่เราตั้งไว้ โดยระดับน้ำจะมีทั้งหมด 4 ระดับ

3. วงจรแสดงผล 7 Segment ไมโครคอนโทรเลอร์ จะส่งข้อมูลไปยัง Porf 2 เมื่อแสดงผลในการตั้งเวลาออกทาง Segment โดยจะใช้ IC เบอร์ 74247 เป็นตัวช่วยขับ 7 Segment

4. วงจรขับ Pump และ วาล์ว ไมโครคอนโทรเลอร์ จะส่งข้อมูลไปยังวงจรขับ เมื่อเราต้องการให้ Pump หรือ วาล์ว ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.ไมโครคอนโทรเลอร์เป็นศูนย์กลางการควบคุมทั้งหมด จะเป็นตัวจัดลำดับการทำงาน ให้เครื่องทำงานไปตามขั้นตอนที่เราต้องการต่อไป

2.3 การทำงานของเครื่อง

ลักษณะการทำงานของเครื่องนั้นเริ่มจากการที่ผู้ใช้นำผักที่ล้างใส่ลงไปในเครื่องซึ่งจะมีภาชนะสำหรับรองรับผักในส่วนที่จะล้างนี้อยู่แล้ว ในส่วนของตัวเครื่องเองจะมีชุดที่สามารถโปรแกรมทำงานได้ดังนี้คือ

1. POWER SWITCH เป็นสวิตซ์ที่ใชสำหรับเปิดปิดให้ส่วนต่างๆของเครื่องและส่วนของวงจรพร้อมที่จะทำงานได้

2. สวิตซ์ในส่วนที่ใช้สำหรับโปรแกรมในส่วนของจำนวนผักที่ต้องการจะล้างในส่วนนี้จะแบ่งออกได้เป็นสามระดับคือ

2.1 จำนวนผักที่ต้องการที่จะล้างน้อย

2.2 จำนวนผักที่ต้องการจะล้างปานกลาง

2.3 จำนวนผักที่ต้องการล้างมาก

2.4 โปรแกรม 1 และโปรแกรม 2 ของการล้างผัก

เมื่อเปิดเครื่องครั้งแรกส่วนแสดงผลของจำนวนผักที่จะล้างนี้จะแสดงผลที่จำนวนผลในการล้างเล็กน้อย แต่เมื่อกดสวิตซ์ตัวนี้ส่วนแสดงผลจะเลื่อนตำแหน่งไปที่จำนวนผักปานกลางและห่างกดครั้งต่อไปส่วนแสดงผลจะเลื่อนไปที่จำนวนผักปานกลางและหากกดครั้งต่อไปส่วนแสดงผลจะเลื่อนไปที่จำนวนผักมากและเมื่อได้ตามที่เราต้องการแล้วก็จะสามารถที่จะโปรแกรมในส่วนต่อไปได้ โปรแกรมในส่วนนี้โปรแกรมจะรู้เองโดยอัตโนมัติ ว่าระดับน้ำแค่ไหนจึงจะเหมาะสมกับจำนวนผักที่เราได้ทำการเลือกไว้

3. สวิตซ์ที่ใช้เลือกจำนวนครั้งในการล้าง ซึ่งสามารถที่จะทำการล้างได้ตั้งแต่ 1-3 ครั้งเราสามารถที่จะโปรแกรมได้ตามที่เราต้องการซึ่งจะแสดงผลการตั้งโปรแกรนในส่วนนี้ทาง LED DISPLAY นั่นเองเมื่อเปิดเครื่องจะแสดงผลที่ 1 ครั้ง

4. สวิตซ์ที่ใช้สำหรับตั้งเวลา สวิตซ์ในส่วนนี้จะแบ่งออกเป็นสองชุดคือชุดสำหรับกดเพื่อเพิ่มเวลาล้าง และชุดที่ใช้สำหรับที่ลดเวลาล้าง ในส่วนนี้สามารถที่จะตั้งเวลาล้างได้ตั้งแต่ 00-99 นาที แสดงผลการตั้งเวลาด้วย LED 7- SEGMENTS 2 หลักเมื่อเปิดเครื่องครั้งแรกแล้ว DISPLSY จะแสดงผลที่ 00 เวลาล้างคิดเป็นนาที

5. START/PAUSE SWITCH หลังจากที่เราได้ทำการโปรแกรมในส่วนต่างๆเรียบร้อยแล้ว เมื่อกดสวิตซ์ตัวนี้ครั้งแรก จะเป็นการสั่งให้เครื่องเริ่มทำงานตามโปรแกรมที่ได้ตั้งไว้เมื่อต้องการที่จะหยุดการทำงานของเครื่องชั่วคราวก็สามารถที่จะกดสวิตซ์ตัวนี้อีกครั้งและหากต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะให้เครื่องทำงานต่อไปก็กดสวิตซ์ตัวนี้อีกครั้งเครื่องจะทำงานต่อจากค่าเดิมที่ได้หยุดไว้ครั้งแรกจนเสร็จตามโปรแกรมที่เราได้ตั้งไว้นั่นเอง

หลังจากที่เราได้ทำการโปรแกรมการทำงานของเครื่องเสร็จเมื่อหลังกดสวิตซ์สตาร์ทครั้งแรก ตัวควบคุมการทำงานในที่นี้ก็คือไมโครคอนโทรลเลอร์MCS-51 จะส่งสัญญาณควบคุมไปสั่งให้วาล์วตัวนี้จะเปิดน้ำเข้ามาในตัวเครื่องจนได้ระดับกับจำนวนผักที่เราได้ทำการโปรแกรมมาไว้ครั้งแรกโดยอัตโนมัติและก็หยุดปล่อยน้ำเข้ามาเมื่อระดับน้ำนั้นเหมาะสมแล้ว

หลังจากที่วาล์วปล่อยน้ำหยุดปล่อยน้ำแล้ว ตัวควบคุมการทำงานของระบบจะส่งสัญญาณควบคุมไปให้มอเตอร์หมุน ในส่วนของมอเตอร์นี้จะต่ออยู่กับปั๊มน้ำเพื่อเพิ่มแรงดันของน้ำให้แรงขึ้นก่อนที่จะนำน้ำในส่วนนี้ไหลเวียนไปในส่วนต่างๆ ของระบบการล้าง ซึ่งได้วางสำหรับให้น้ำไหลเวียนภายในตัวเครื่องอย่างเหมาะสมแล้วแรงดันน้ำในส่วนนี้จะทำการชะล้างทำความสะอาดผักพร้อมกับทำการเปลี่ยนแปลงตำแหน่งของผักเพื่อเพิ่มความสะอาดในการล้างมากยิ่งขึ้น เวลาในการทำงานของมอเตอร์นี้จะทำงานตามเวลาที่เราได้โปรแกรมระยะเวลาในการล้างครั้งแรก

หลังจากที่มอเตอร์หยุดทำงานตัวควบคุมจะส่งสัญญาณควบคุมไปสั่งให้วาล์วปล่อยน้ำทิ้งทำงานและวาล์วตัวนี้จะหยุดทำงานก็ต่อเมื่อน้ำภายในตัวเครื่องถูกปล่อยทิ้งออกไปจากตัวเครื่องจนหมดและตัวควบคุมการทำงานก็จะสั้งให้วาล์วเปิดน้ำปล่อยน้ำเข้ามาในระบบการล้างอีกครั้ง และจะมีขั้นตอนการทำงานตามลำดับเหมือนเดิม

หลังจากที่เครื่องได้ทำการล้างครบตามจำนวนครั้ง ตามที่ผู้ใช้ได้ทำการโปรแกรมไว้แล้วตัวควบคุมการทำงานจะส่งสัญญาณไปยังส่วนของวงจรกำเนิดสัญญาณเสียงเพื่อส่งสัญญาณเสียงบอกเตือนให้ผู้ใช้ทราบว่าขณะนี้เครื่องได้ทำการล้างเสร็จสิ้นเรียบร้อยแล้วนั่นเอง

2.4 การควบคุมการทำงานของส่วนต่าง ๆ

ส่วนประกอบพื้นฐานที่สำคัญในการควบคุมระบบนั้น สามารถที่จะแบ่งออกเป็นส่วนๆ ต่าง ๆ ที่สำคัญได้ดังนี้คือ

2.3.1 ตัวคอนโทรลเลอร์ (Controller)

2.3.2 วงจรไดรเวอร์ (Driver) หรือวงจรแอมป์ปลิไฟเออร์ (Amplifier)

2.3.3 ส่วนของวงจรป้อนกลับ (Feed back)

2.3.4 โหลด (Lode) ในที่นี้ก็คือวาล์วและมอเตอร์

สำหรับรายละเอียดและหน้าที่ของส่วนต่างๆนั้น สามารถที่จะอธิบายได้ดังนี้

1. Controller เป็นส่วนของระบบที่ทำหน้าที่ในการกำเนิดสัญญาณควบคุมไปบังคับการทำงานของส่วนต่าง ๆของระบบในที่นี้ก็คือวาล์วและมอเตอร์และยังควบคุมการทำงานของระบบให้เป็นไปตามขั้นตอนที่กำหนดด้วย ในการทำงานของเครื่องในที่นี้จะใช้ MCS-51 เป็นตัวคอนโทรลเลอร์การทำงานของระบบซึ่งรายละเอียดจะได้กล่างถึงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Driver วงจรไดรเวอร์หรือวงจรแอมปลิไฟเออร์ (Amplifier) ในส่วนของวงจรส่วนนี้จะทำหน้าที่ในการขยายสัญญาณควบคุมให้มีขนาดที่เหมะสมก่อนที่จะป้อนไปยังโหลด

3. Feed back หรือส่วนของวงจรป้อนกลับเป็นส่วนที่ใช้ในการรับรู้หรือ detect สัญญาณเอาท์พุทที่ต้องการวงจรในส่วนนี้จะต้องไม่มีผลต่อการLoading และสัญญาณส่วนนี้จะถูกเปรียบเทียบกับสัญญาณอ้างอิงและนำสัญญาณที่ได้จากการเปรียบเทียบ (Error) ไปใช้งานต่อไป

4. Load คือส่วนของระบบที่ถูกควบคุมนั่นเอง

ในส่วนของวงจรไดรเวอร์นั้นจะใช้ RELAY เป็นตัวควบคุมการทำงานของวาล์วรวมทั้งมอเตอร์ด้วยในส่วนนี้จะเอาสัญญาณควบคุมจากตัวคอนโทรลเลอร์มาทำการขยายด้วยวงจร Amplifier ก่อนที่จะนำมาในส่วนของภาคนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 หน้าปัดและการใช้งานของปุ่มต่างๆของเครื่องล้างผัก

SW1, SW2 เป็นสวิตซ์เพิ่มและลดเวลาในการล้างผักตามลำดับ

SW3 เป็นสวิตซ์เลือกระดับน้ำในการล้างซึ่งมี3ระดับคือระดับ1,2,และระดับ3

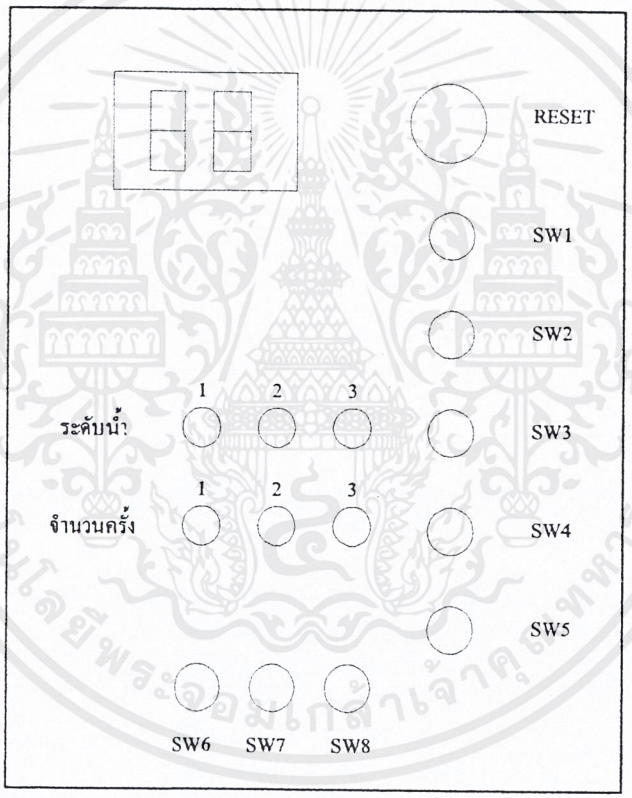
SW4 เป็นสวิตซ์เลือกจำนวนครั้งในการล้างซึ่งมี3ระดับคือ1,2และ3ครั้ง

SW5 เป็นสวิตซ์กด Start และ Stop ในการล้างผัก

SW6เป็นสวิตซ์กดเพื่อตั้งโปรแกรมในการล้างผักด้วยตัวเอง โดยการเลือกระดับน้ำ,จำนวนครั้งและเวลาในการล้างเอง

SW7 เป็นสวิตซ์โปรแกรม1 สำหรับล้างผักน้อยคือเวลา 10 ,ระดับน้ำ1และล้าง2ครั้ง

SW8 เป็นสวิตซ์โปรแกรม1 สำหรับล้างผักน้อยคือเวลา 20 ,ระดับน้ำ3และล้าง3ครั้ง



รูปที่ 2.6 แสดงหน้าจอของส่วนควบคุมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการควบคุมการทำงาน

วงจรที่ใช้ควบคุมการทำงานของเครื่องล้างผักนี้เป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งในตัวไมโครคอนโทรลเลอร์จะมี (CPU) หน่วยความจำ หน่วยส่งและรับสัญญาณ (I/O Unit) และหน่วยพิเศษที่ใช้เฉพาะงานในตัวไมโครคอนโทรลเลอร์การออกแแบบไมโครคอนโทรลเลอร์เพื่อใช้ในการควบคุมนั้นสามารถทำได้ง่ายมีความคล่องตัวสูงเพราะในบางครั้ง เราอาจจะเปลี่ยนรูปแบบ การทำงานของเครื่องล้างผักใหม่โดยเราเปลี่ยนแปลงที่ตัวโปรแกรมใน MCS-51อย่างเดียวเท่านั้นซึ่งเราไม่ต้องไปเปลี่ยนแปลงวงจรอิเล็กทรอนิกส์ส่วนอื่นๆเลย ซึ่งในที่นี้เราจะใช้ไมโครคอนโทรลเลอร์เบอร์ 89C51 ทั้งนี้เนื่องจากเบอร์นี้เราสามารถเขียนโปรแกรมและบันทึกด้วยตัวเราเองได้เลย

คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- มีหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลภายในตัว
- ต้องการแหล่งจ่ายไฟ 5V เพียงชุดเดียว
- หน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มิไพม์เมอร์ เคาน์เตอร์ ขนาด 16 บิต 2 ชุด
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART)2 พอร์ตแบบ Full Duplex เลือกรูปแบบได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต

ในการที่จะนำเอา MSC-51 มาใช้งานจำเป็นต้องศึกษาและทำความเข้าใจถึงโครงสร้างและองค์ประกอบของ MCS-51 เสียก่อนแล้วถึงจะเขียนโปรแกรมเพื่อควบคุมการทำงานของ MCS-51

8051 ไมโครคอนโทรลเลอร์ที่บรรจุในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 8051 มีอยู่ข้างละ 20 ขา รวมทั้งหมด 40ขา นั้นจะใช้งานต่างกันดังนี้คือ

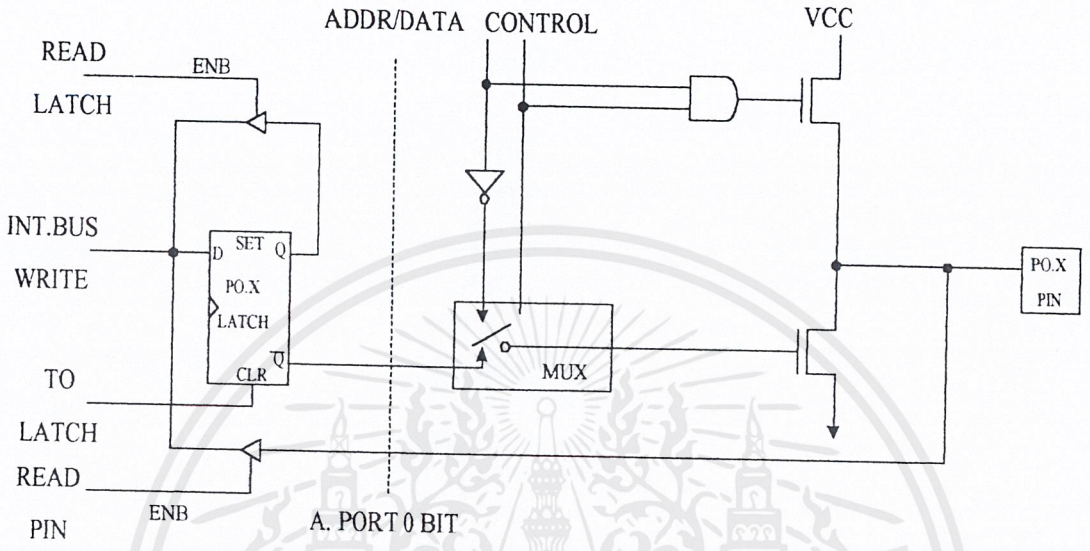
Vcc: ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง 5V เข้าไปเพื่อให้วงจรรวมทำงานได้

Vss: ขา20 เป็นขาที่ต้องต่อกับกราวด์ของแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT 0

เป็นพอร์ตขนาน 8 บิตอยู่ที่ขา 39 ถึง 32 พอร์ต 0 นี้ใช้ได้ทั้งการรับส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือเป็นพอร์ตรับส่งข้อมูล และข้อมูลที่ส่งออกทางพอร์ต 0 และถูก Latch ไว้ที่ขาของพอร์ตแต่ละบิตเป็นแบบ Open Drain Bidirectional ดังรูปที่ 3.1



รูปที่ 3.1 แสดงโครงสร้างของพอร์ต 0

ถ้าต้องการให้พอร์ต 0 เป็นอินพุตพอร์ตต้องส่ง ลอจิก 1 ไปยังพอร์ตนี้ จะมีผลต่อ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุตพอร์ตแลทซ์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port(pin) จะใช้สัญญาณ Read(pin)

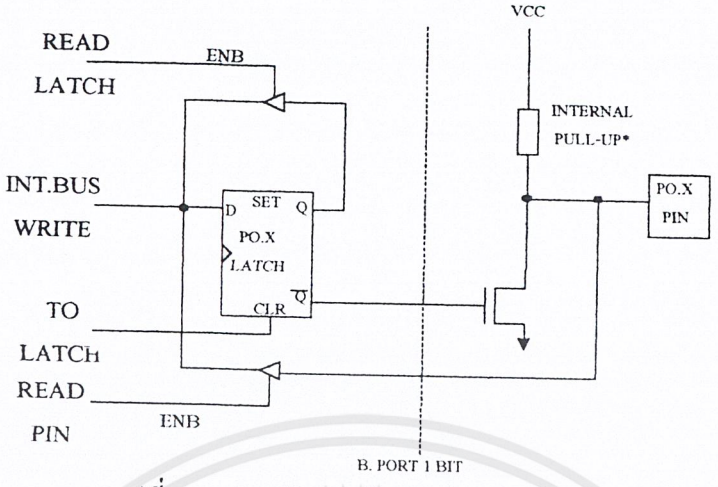
หน้าที่ของพอร์ต 0 มีดังนี้

1. ใช้ส่งค่าตำแหน่งของหน่วยความจำภายนอก
2. ใช้รับส่งข้อมูลกับหน่วยความจำข้อมูลหรือใช้รับข้อมูลจากหน่วยความจำโปรแกรม
3. ใช้รับส่งข้อมูลผ่านพอร์ตโดยตรงในกรณีที่ไม่ได้ใช้หน่วยความจำใดๆ

PORT 1

เป็นพอร์ตขนานแบบ 8 บิต อยู่ที่ขา 1 ถึง 8 มีโครงสร้างของพอร์ต 1 ดังรูปที่ 3.2 พอร์ต 1 ใช้ทำหน้าที่เป็น ตัวรับส่งข้อมูลเท่านั้นข้อมูลที่ถูกส่งลงมาทางพอร์ต 1 จะถูก Latch และการให้พอร์ต 1 เป็นอินพุตก็ต้องป้อน ลอจิก 1 ไปทุกบิตเช่นเดียวกับพอร์ต 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



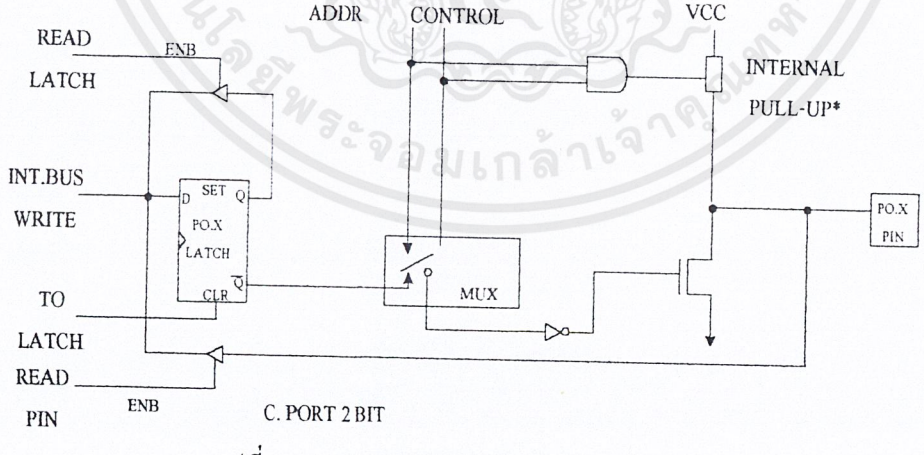
รูปที่ 3.2 แสดงโครงสร้างของพอร์ต1

Port 2

เป็นพอร์ตขนานขนาด 8 บิต การทำงานจะแบ่งได้เป็น 2 แบบคือ

1. ใช่ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ
2. ใช้เป็นพอร์ตรับส่งข้อมูลภายนอก

พอร์ตนี้มีโครงสร้างคล้ายพอร์ต 0 แต่จะใช้ความต้านทานภายในพูลอัพแทนเมื่อใช้งานเป็นอินพุตจะต้องป้อนลอจิก 1 ให้แก่ทุกบิตพอร์ต 2 รูปการแสดงพอร์ต 2 แสดงได้ดังรูปที่ 3.3



รูปที่ 3.3 รูปแสดงโครงสร้างของพอร์ต 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT 3

เป็นพอร์ตขนาด 8 บิตคือขา 10 ถึง 17 มีโครงสร้างคล้ายพอร์ต 1 ทำงานได้ 2 หน้าที่คือเป็นไอโอพอร์ตถ้าจะโปรแกรมเป็นอินพุตต้องป้อน ลอจิก 1 ให้กับทุกบิตของพอร์ต 3 และอีกหน้าที่หนึ่งคือใช้ส่งสัญญาณควบคุมออกมาและรับสัญญาณเข้าไปสัญญาณต่างๆมีดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)

P3.2/INT0 (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0

P3.3/INT1 (External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1

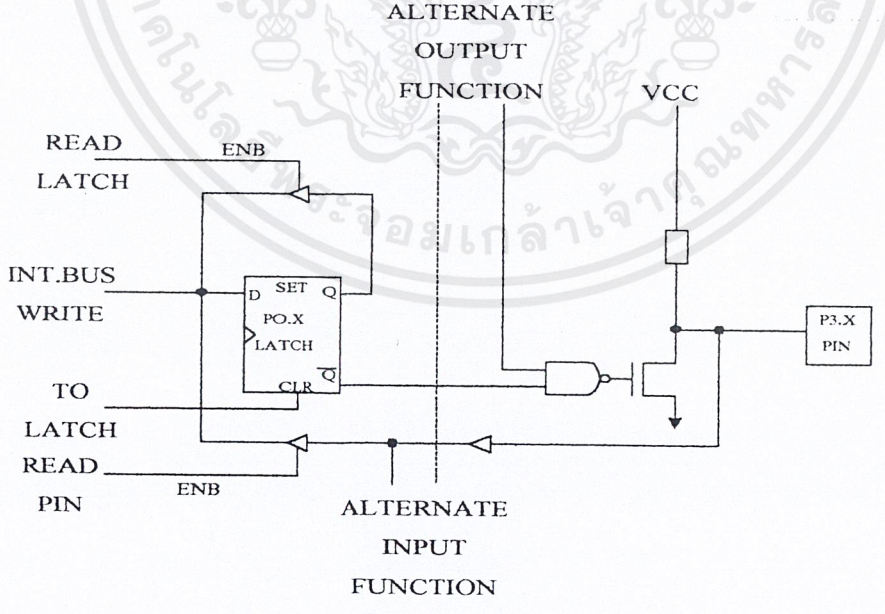
P3.4/TO (Counter External Input) ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 0 (เป็นอินพุต Counter)

P3.5/T1 (Counter External Input) ขารับสัญญาณพัลซ์อินพุตเข้าไปยังวงจร Counter 1

P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก

P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

โครงสร้างของ Port 3



รูปที่ 3.4 แสดงโครงสร้างของพอร์ต 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALE: (Address Latch Enable) ขาที่ 30 ขาสัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 8051 ว่าขณะนี้จะมีส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ต 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ต 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้นซึ่งเวลาต่อมาพอร์ต0 จะใช้รับส่งข้อมูลกับหน่วยความจำภายนอกต่อไป

PSEN: (Program STORE Enable) เป็นขาที่ 29 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่งจะนำไปทำงานมาจากหน่วยความจำโปรแกรมภายนอก

EA: (External Access) เป็นขาที่ 31 ใช้เลือกหน่วยความจำโปรแกรมภายนอก

ป้อน 1 จะอ่านโปรกแรมบนชิพ

ป้อน 0 จะอ่านโปรแกรมภายนอกชิพ

RST: (RESET) เป็นขาที่ 9 ขารีเซ็ทจะรีเซ็ทได้ก็ต่อเมื่อป้อนลอจิก 1 เข้าไปนานอย่างน้อย 2 แมชชีนไซเคิล

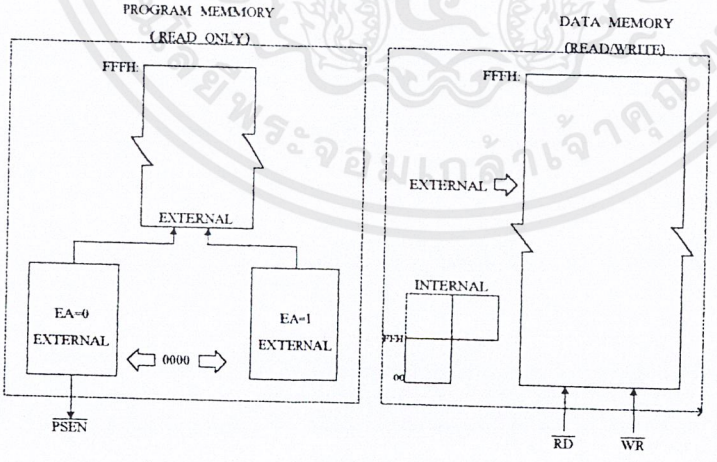
XTAL1: เป็นขาที่ 19 ใช้คริสตัลภายนอกโดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์ภายใน

XTAL2: เป็นขาที่ 18 ใช้คริสตัลภายนอกโดยเป็นเอาท์พุตของวงจรออสซิลเลเตอร์

การแบ่งประเภทของหน่วยความจำของ MCS-51

หน่วยความจำของ MCS-51 มีอยู่ด้วยกัน 2 ชนิดคือ

1. หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory)
2. หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)



รูปที่ 3.5 แสดงหน่วยความจำของโปรแกรมและหน่วยความจำของข้อมูลตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำโปรแกรม ใช้สำหรับเก็บโปรแกรมภาษาเครื่องที่ 8051 จะทำงานเมื่อเริ่มป้อนไฟเลี้ยงให้ 8051 หรือมีการรีเซต 8051 จะทำให้เริ่มการทำงานจากคำสั่งในหน่วยความจำโปรแกรมตำแหน่งที่ 0000H นั่นเอง

หน่วยความจำสำหรับข้อมูล ใช้สำหรับเก็บหรือพักข้อมูลระหว่างที่ทำงานหน่วยความจำสำหรับข้อมูลมี 2 แบบแบบหนึ่งอยู่ในชิพ 8051 ขนาด 128 ไบต์อีกแบบหนึ่งขนาด 64 กิโล-ไบต์ต้องต่อเข้าไปเพิ่มเติมภายนอก 8051 หน่วยความจำภายในตำแหน่ง 0 ถึง 7F นี้สามารถอ้างถึงได้โดยตรงมีการสั่งให้อ่านหรือเขียนข้อมูลไปยังตำแหน่งนั้นได้โดยตรง แต่ตำแหน่ง 80 ถึง FFH นั้นเป็นรีจิสเตอร์ฟังก์ชั่นพิเศษ หน่วยความจำภายในช่วงนี้ใช้เป็นรีจิสเตอร์สำหรับงานเฉพาะอย่าง

หน่วยความจำสำหรับข้อมูลภายใน 8051 ช่วง 00H ถึง 07F สามารถแบ่งออกได้เป็น 3 กลุ่มคือ

1. Register bank 0ถึง 3 อยู่ในตำแหน่งหน่วยความจำช่วงตำแหน่งที่ 00H ถึง 1FH หน่วยความจำนี้จะแบ่งออกเป็น 4 ชุด ชุดละ 8 ไบท์แต่ละชุดเรียกว่า BANK แต่ละไบต์ใน 1BANKจะมีชื่อของรีจิสเตอร์ ว่า R0,R1,R2,R3,R4,R5R6และ R7 รีจิสเตอร์เหล่านี้เหล่านี้สามารถเรียกใช้งานในระหว่างทำโปรแกรมได้อย่างสะดวก และรีจิสเตอร์เหล่านี้จะมีชื่อซ้ำกันในทุก BANK การใช้งานจึงต้องจึงต้องเรียกใช้งานทีละ BANK เท่านั้น โดยการกำหนดในรีจิสเตอร์ PSW เมื่อมีการ Reset การทำงานของ 8051 จะเริ่มการใช้งานของริจิสเตอร์ R0 ถึง R7 ที่ BANK 0 ซึ่งรีจิสเตอร์ RO ถึงR7 ในแต่ละ BANK ในการอ้างอิงในหน่วยความจำสำหรับข้อมูลภายใน 8051 ดังในตารางที่ 3.1

รีจิสเตอร์	ตำแหน่งหน่วยความจำ			
	BANK 0	BANK 1	BANK 2	BANK 3
R0	0	8	10	18
R1	1	9	11	19
R2	2	A	12	1A
R3	3	B	13	1B
R4	4	C	14	1C
R5	5	D	15	1D
R6	6	E	16	1E
R7	7	F	17	1F

ตารางที่ 3.1 แสดงหน่วยความจำของรีจิสเตอร์

2. Bit Address Area เป็นหน่วยความจำในช่วงตำแหน่ง 20H ถึง 2FH หน่วยความจำแต่ละบิตในช่วงของหน่วยความจำดังกล่าวจะสามารถตรวจสอบหรือตั้งค่าเป็น 1 หรือ 0 ได้โดยการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมภาษาเครื่องแต่ละบิตของข้อมูลในหน่วยความจำช่วงนี้จะมีค่าตำแหน่งดังใน Memory Map

3. Scratched pod Area เป็นช่วงของหน่วยความจำตำแหน่ง30 ถึง 7FH หน่วยความจำช่วงนี้จะใช้สำหรับเก็บข้อมูลทั่วไป ถ้ารีจิสเตอร์ Stack pointer ชี้มายังหน่วยความจำช่วงนี้จะต้องระวังไม่ให้เกิดการเขียนทับของข้อมูลอันจะทำให้การทำงานของโปรแกรมผิดพลาดได้

Byte	(MSB)							(LSB)	
7FH	Scratch Psd Area								
30H									
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	62	61	61	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	49	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	
1FH	R0 - R7								รีจิสเตอร์ แบงค์ 3
18H									
17H	R0 - R7								รีจิสเตอร์ แบงค์ 2
10H									
0FH	R0 - R7								รีจิสเตอร์ แบงค์ 1
08H									
07H	R0 - R7								รีจิสเตอร์ แบงค์ 0
00H									

รูปที่ 3.6 128 ไบต์ของ Ram ที่เข้าถึงข้อมูลแบบทางตรงและทางอ้อม

มีรีจิสเตอร์อีกกลุ่มหนึ่งที่เรียกว่า Special Function Register (SFR) เป็นรีจิสเตอร์ใช้เฉพาะงาน แต่ละตำแหน่งของ SFR อาจไม่ใช้เป็นหน่วยความจำแต่อาจใช้เป็น Counter Register,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42628

Shift register ซึ่งการอ้างอิงข้อมูลในตำแหน่งนั้น 8051 จะมองว่าเป็นหน่วยความจำตำแหน่งหนึ่ง จึงเรียกการมองข้อมูลแต่ละตำแหน่งว่า Memory Map I/O ดังแสดงตำแหน่งดังนี้

สัญลักษณ์	ชื่อ	ตำแหน่ง
*ACC	Accumulator	OEOH
*B	B Register	OFOH
*PSW	Program Status Word	ODOH
SP	Stack Pointer	81H
DPTR	Data Pointer	
DPL	Low Byte	82H
DPH	High Byte	83H
*PO	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	OAOH
*P3	Port 3	OBOH
*IP	Interrupt Priority Control	OB8H
*IE	Interrupt Enable	OA8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
THO	Timer/Counter 0 High Byte	8CH
TLO	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

ตารางที่ 3.2 แสดงรีจิสเตอร์ที่มีการใช้งานเฉพาะ

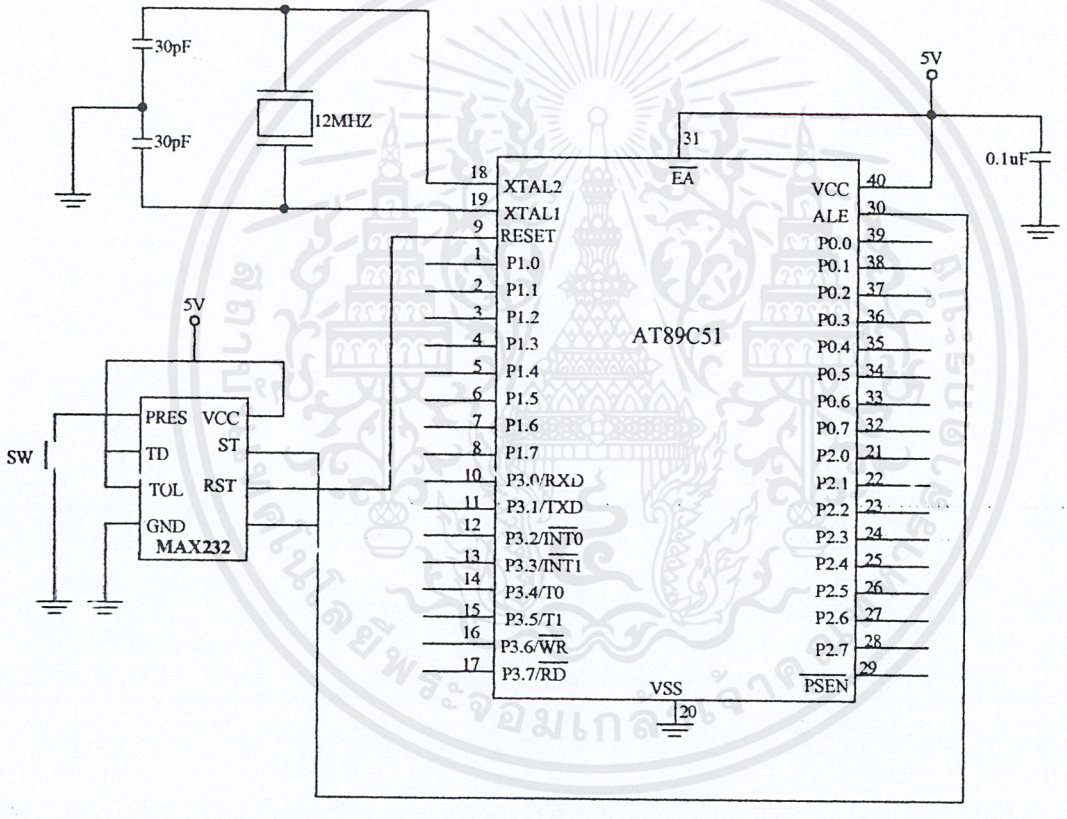
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกับอุปกรณ์ภายนอก

ในที่นี้เราจะใช้ไมโครคอนโทรลเลอร์ ซึ่งมีทั้งหมด 4 พอร์ตที่สามารถเป็นได้ทั้งอินพุตและเอาท์พุตภายในเครื่องล้างผักนี้เราจะใช้เฉพาะ หน่วยความจำโปรแกรมภายในเท่านั้นและการต่อ MCS-51 ไปใช้งานจะได้กล่าวต่อไป

การเอาไมโครคอนโทรลเลอร์ไปใช้งาน

การเอาไมโครคอนโทรลเลอร์ไปใช้เราจะต่อคริสตัลให้กับ MCS-51 เพื่อเป็นตัวผลิตความถี่ให้กับไมโครคอนโทรลเลอร์และต่อวงจรรีเซ็ทให้ไมโครคอนโทรลเลอร์เสียก่อนจึงสามารถเอาไปเขียนโปรแกรมเพื่อควบคุมต่อไปได้แสดงดังรูปที่ 3.7



รูปที่ 3.7 แสดงการประยุกต์ไมโครคอนโทรลเลอร์

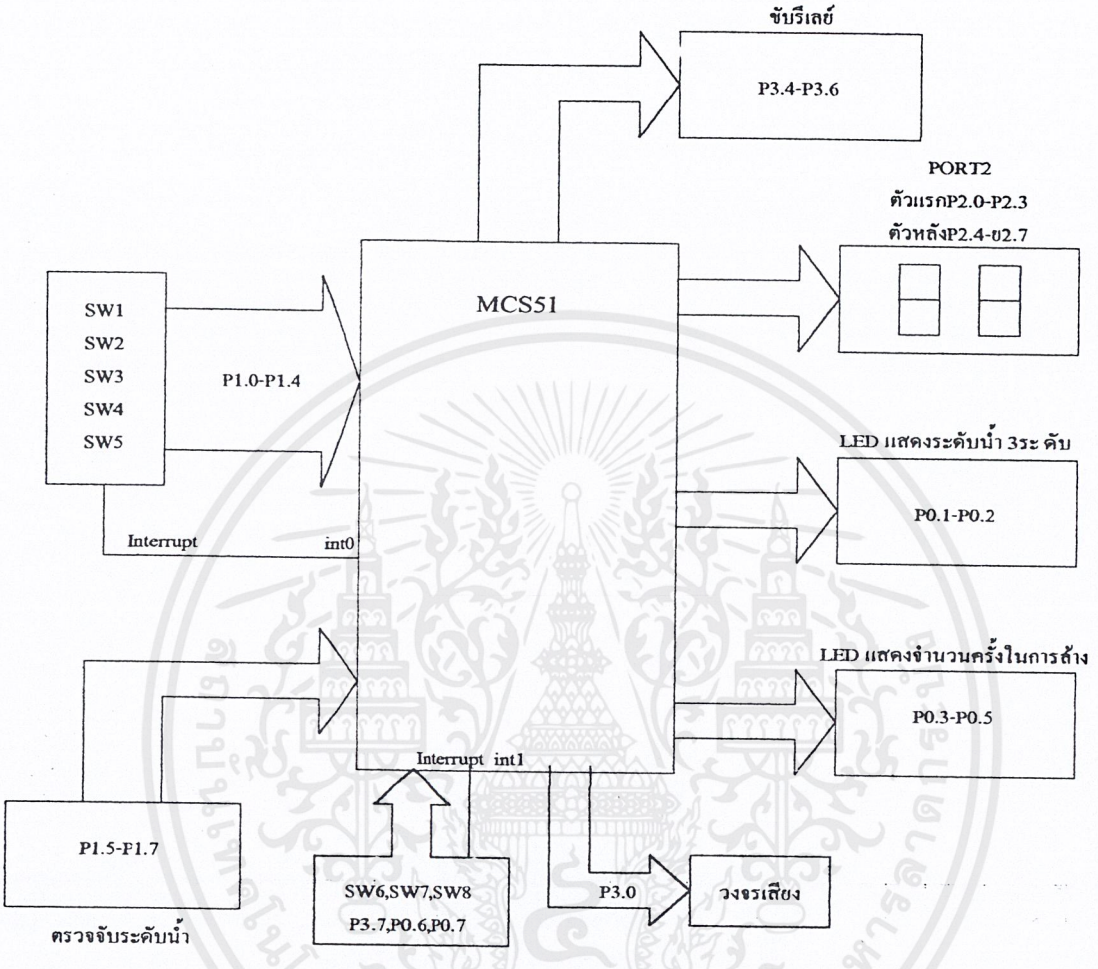
พอร์ต 1 ใช้เป็นอินพุตเพื่อรับการตรวจเช็คการกดสวิตซ์และการตรวจระดับน้ำภายในเครื่องล้างผัก

พอร์ต 2 ใช้เป็นเอาท์พุตในการขับ7-Segment 2 หลักต่อไปซึ่งเป็นการแสดงเวลาที่เราตั้งไว้

พอร์ต 3 ใช้ในการควบคุมการขับ Relay ต่างๆเพื่อไปควบคุมการ Control Pump และ วาล์วตามที่เราต้องโปรแกรมไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมการควบคุมเครื่องล้างผัก



รูปที่ 3.8 แสดงบล็อกไดอะแกรมควบคุมเครื่องล้างผัก

จากบล็อกไดอะแกรมเราสามารถอธิบายวงจรแต่ละบล็อกดังนี้

วงจรสวิตซ์ควบคุมต่างๆของเครื่อง

ในเครื่องล้างผักจะมีสวิตซ์ควบคุมการทำงาน 5 สวิตซ์ดังนี้

SW. ตัวที่ 1 เป็นสวิตซ์เลือกว่าเราต้องการล้างผักกี่น้ำซึ่งการกดสวิตซ์จะทำให้เกิดสัญญาณขึ้นแล้วไปเข้าขา P1.0 ของไมโครคอนโทรลเลอร์ เพื่อนำไปเขียนโปรแกรมควบคุมต่อไป

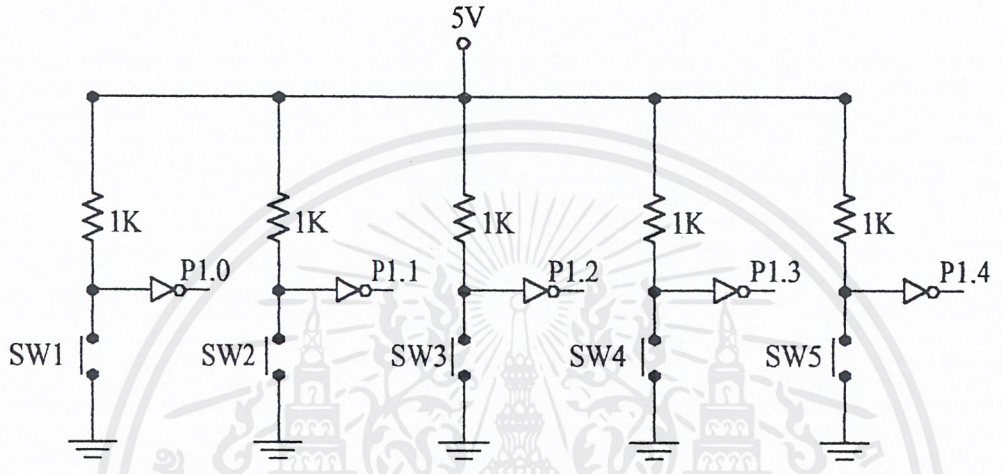
SW.ตัวที่ 2 เป็นตัวเลือกว่าต้องการน้ำระดับไหน

SW. ตัวที่ 3 และตัวที่ 4 เป็นสวิตซ์เลือกเวลาในการล้างผักว่าต้องการเวลาเท่าไรในการล้างผัก

SW. ตัวที่ 5 เป็นสวิตซ์สำหรับการเริ่มทำงานของเครื่องล้างผักและสำหรับการหยุดทำงานของเครื่องล้างผัก

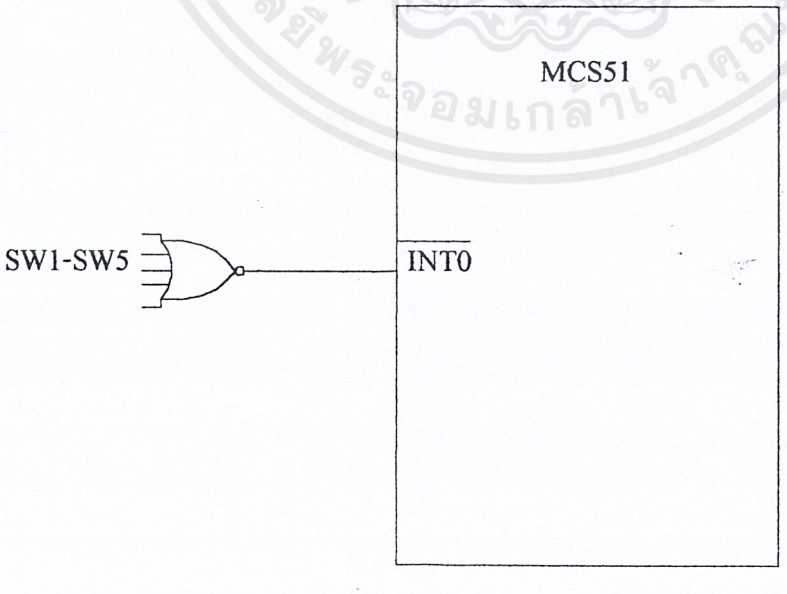
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของวงจรดังแสดงรูปที่ 3.9 คือเมื่อไม่มีการกดสวิตซ์ใดๆแรงดันจะไปตกคร่อมที่สวิตซ์ทำให้ Schmitt- Trigger Inverter ทางอินพุตจะเป็น High ทำให้เอาท์พุตเป็น Low แล้วไปเข้าไมโครคอนโทรลเลอร์ และถ้ามีการกดสวิตซ์จะทำให้แรงดันตกคร่อมสวิตซ์เป็นศูนย์ดังนั้นอินพุตของ Schmitt- Trigger Inverter เป็น Low ทำให้เอาท์พุตเป็น High แล้วเอาไปเข้าไมโครคอนโทรลเลอร์เพื่อนำไปเขียนโปรแกรมควบคุมต่อไป



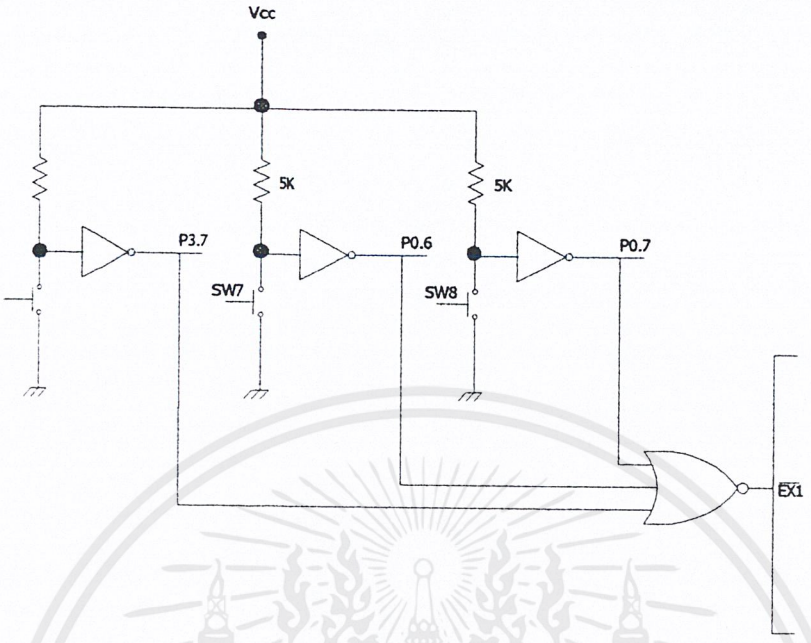
รูปที่ 3.9 แสดงวงจรสวิตซ์ 1-5

และ Output ของวงจรรูปที่3.10 จะเข้าไปที่ Nor Gate เพื่อนำสัญญาณเข้าไปยังขา $\overline{\text{INT 0}}$ เพื่อเป็นสัญญาณ Interrupt ให้กับ MCS-51 เพื่อความง่ายต่อการเขียนโปรแกรมควบคุม



รูปที่ 3.10 แสดงลักษณะการต่อของNor Gate เข้าขา Interrupt 0 ของ Mcs-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงวงจรสวิตซ์ 6-8

หลักการทำงานของวงจรคือเมื่อไม่ได้กดสวิตซ์ทำให้แรงดัน Vcc ไปตกคร่อมที่สวิตซ์ทำให้ NOT GATE มีอินพุตเป็น High และเป็นผลให้เอาท์พุตที่ NOT GATE เป็น Low และเอาท์พุตที่ได้นี้จะเอาไปเข้า P3.7, P0.6 และP0.7แต่ถ้ามีการกดสวิตซ์จะทำให้แรงดันไปตกคร่อม R 5k ทำให้อินพุตของ NOT GATE เป็น Low เป็นผลให้เอาท์พุตของ NOT GATE เป็น High และเอาท์พุตที่ได้จะนำไปเข้า P3.7, P0.6 และP0.7 ของ MCS-51 เพื่อเขียนโปรแกรมควบคุมต่อไป

หน้าที่ของสวิตซ์แต่ละตัว

SW6 จะทำหน้าที่เป็นปุ่มกดเพื่อเริ่มตั้งโปรแกรมล้างผักด้วยตัวเองและเมื่อล้างผักเสร็จก็ต้องกดปุ่มนี้เพื่อให้เครื่องเตือนการล้างหายไป

SW7 เป็นโปรแกรมที่1สำหรับล้างผักน้อยคือตั้งระดับน้ำไว้ที่ระดับ1 ใช้เวลา 10 นาที

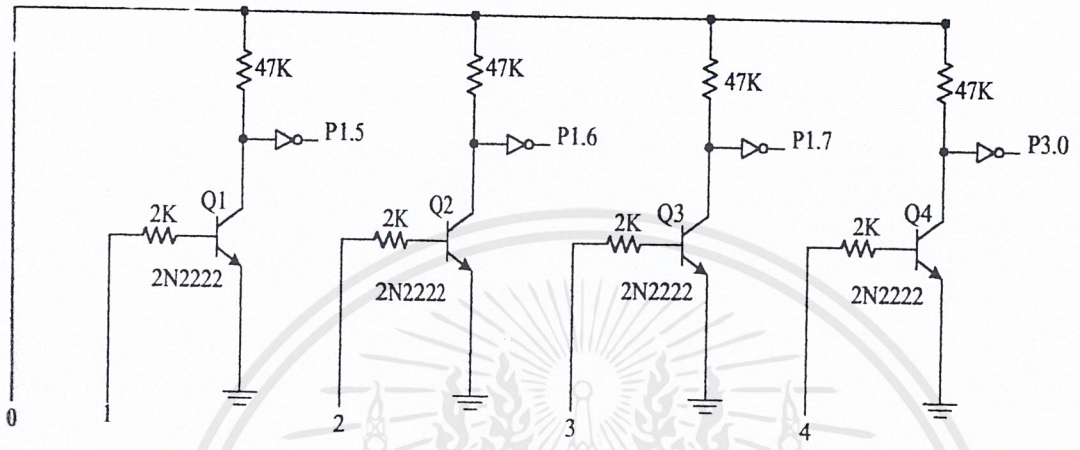
SW8 เป็นโปรแกรมที่2สำหรับล้างผักมากคือตั้งระดับน้ำไว้ที่ระดับ3 ใช้เวลา 20 นาที

และเอาท์พุตของวงจรนี้จะนำไปเข้าขา $\overline{EX1}$ ของ MCS51 เป็นสัญญาณ interrupt เพื่อให้ง่ายต่อการเขียนโปรแกรมคือถ้ามีการเขียนโปรแกรมก็จะมีการ interrupt เกิดขึ้น แต่ถ้าไม่มีการเขียนโปรแกรมก็จะไม่มีการ interrupt เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร Sensor ระดับน้ำ

เครื่องล้างผักจะมีการวัดระดับน้ำ 3 ระดับ ในการล้างผักโดยขึ้นอยู่กับปริมานของผักเราจึงเลือกระดับน้ำในการล้างผักให้เหมาะสมกับปริมาณผัก



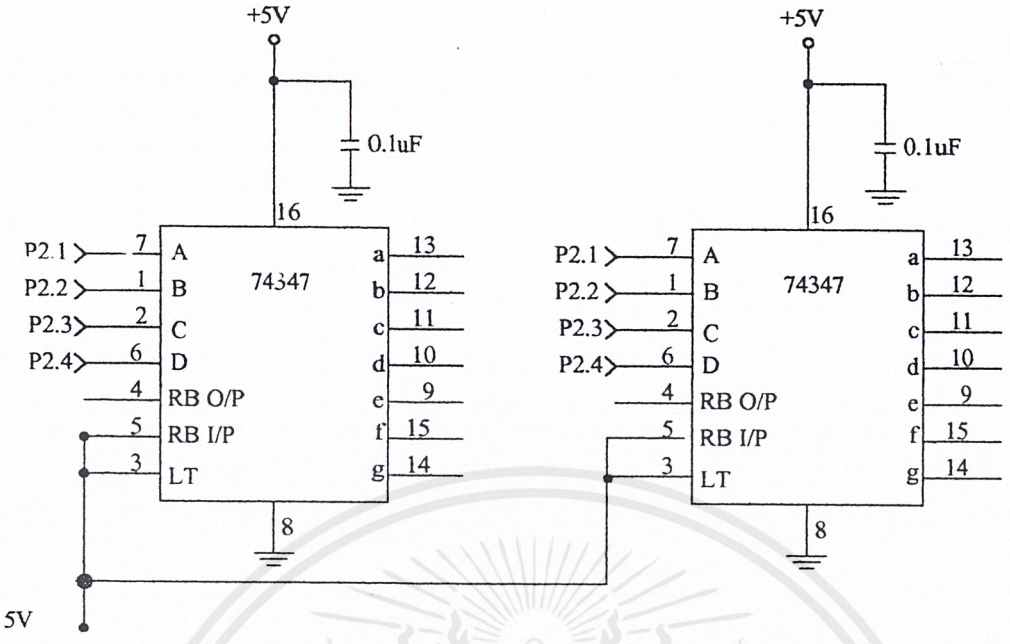
รูปที่ 3.12 แสดงวงจรการ Sensor ระดับน้ำของเครื่องล้างผัก

หลักการทำงานของวงจร Sensor ระดับน้ำเมื่อมีการเลือกระดับน้ำที่ SW.ที่2 เช่นเลือกระดับ 1 แล้วเมื่อน้ำเข้าไปในเครื่องล้างผักแล้วทำให้น้ำไปถึงขุด 0 และขุดที่ 1 ทำให้ขุด 0 และขุด 1 ต่อถึงกันทำให้มีกระแสผ่านเข้าไปขา Base ของ Q1 ทำให้ Q1 ทำงานทำให้ VCE ของ Q1 มีค่าประมาณ 0.2V ทำให้ Schmitt- Trigger Inverter ทางอินพุตเป็นLow ทำให้ได้เอาท์พุต Schmitt-Trigger Inverter เป็น High แล้วนำไปเข้าไมโครคอนโทรลเลอร์เพื่อทำงานตามโปรแกรมต่อไป ส่วนระดับน้ำที่ระดับขุด 2 และขุด 3 ก็ทำงานเช่นเดียวกับระดับหนึ่งแต่ถ้าระดับน้ำที่ระดับ 0ไปไม่ถึงขุด 1ก็จะไม่มีกระแส Base จึงทำให้ Transistor Q1 Off ทำให้เอาท์พุตของ Schmitt- Trigger Inverter เป็น Low แล้งจึงนำเอาท์พุตของ Schmitter - Trigger ไปเขียนโปรแกรมต่อไป

วงจรแสดงผลทาง 7-Segment 2 หลักของเครื่องล้างผัก

ในการแสดงผลทาง Segment เป็นการแสดงเวลาที่เราตั้งไว้ในการล้างผักและส่งออกมาทาง Port 2 เพื่อควบคุม Segment ในการแสดงที่ Segment เราจะใช้ IC เบอร์74247 ในการขับ Segment เพื่อความสะดวกของการเขียนโปรแกรมและประหยัด Port ซึ่งภายใน IC เบอร์74247 เป็นวงจรGate ที่เขาออกแบบไว้แล้วมันเป็น IC BCD - to - Seven - Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

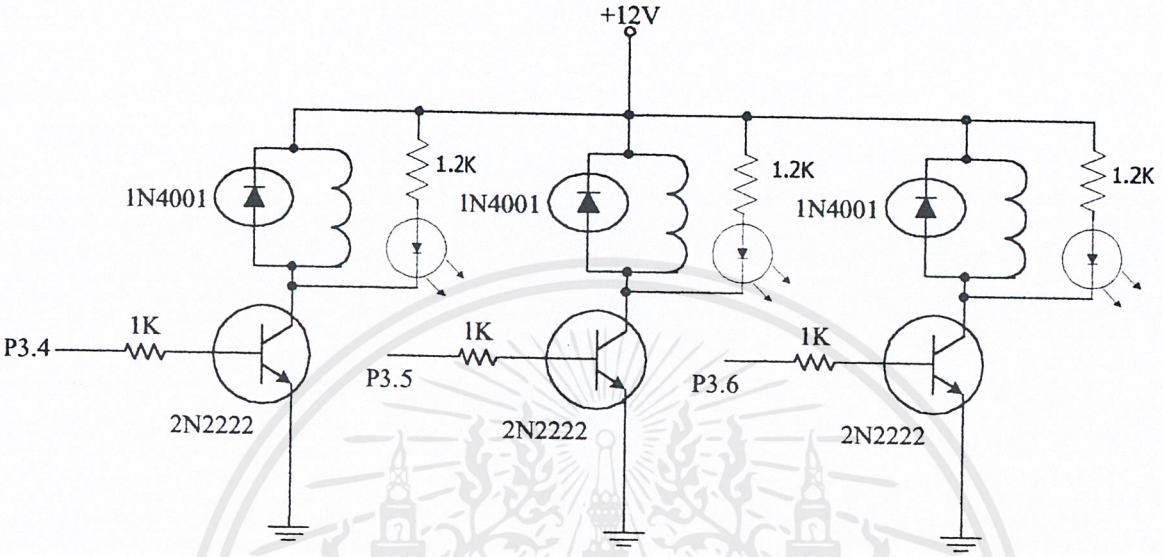


รูปที่ 3.13 แสดงวงจรการต่อวงจรแสดงการตั้งเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรขับ Pump และวาล์วน้ำ

เป็นวงจรที่ไปขับ Pumpและวาล์วเพราะเราจะเอาเอาท์พุตของไมโครคอนโทรลเลอร์ไปขับ Pump และวาล์วตรงๆไม่ได้ต้องมีวงจรขับดังรูปที่ 3.14



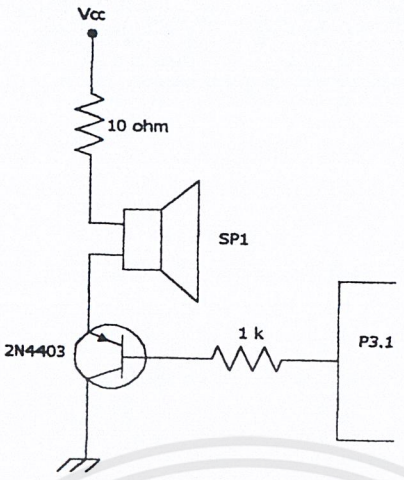
รูปที่ 3.14 แสดงการขับ Pump และ วาล์ว

หลักการทำงานของวงจรขับ Pump และวาล์วคือไมโครคอนโทรลเลอร์ส่งลอจิก 0 มาที่ Transistor ของวงจรขับแล้วTransistor ของวงจรขับไม่มีการนำกระแสทำให้ Relay ไม่ทำงานจึงเป็นผลให้ Pump และวาล์วที่ Relay ควบคุมไม่ทำงานแต่ถ้าไมโครคอนโทรลเลอร์ส่งลอจิก 1 มาที่วงจรขับของ Transistor แล้วTransistor ของวงจรขับนำกระแสทำให้ Relay ทำงานแล้วทำให้ Pump และวาล์วที่ Relay ไปควบคุมการทำงาน

จากที่กล่าวมาทั้งหมดในบทนี้เป็นการแสดงถึงวงจรควบคุมต่างๆของเครื่องล้างแต่จะให้เครื่องล้างผักทำงานสมบูรณ์ต้องมีการออกแบบโปรแกรมที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรเสียง (Sound circuit)



รูปที่ 3.15 แสดงวงจรเสียง

หลักการทำงานของวงจรเสียงคือจะทำงานแบบ Active Low คือถ้าMCS-51 ป้อน High จะทำให้ทรานซิสเตอร์ Off ก็จะไม่มีไฟคร่อมลำโพงจะทำให้ลำโพงไม่มีเสียง แต่ถ้าเราให้ MCS-51 ป้อน Low ที่ P3.0 จะทำให้ทรานซิสเตอร์ทำงานเป็นผลให้มีไฟไปคร่อมลำโพงทำให้ลำโพงเกิดเสียงและเราจะใช้การเขียนโปรแกรมเป็นตัวควบคุมให้เกิดเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ซอฟต์แวร์ควบคุมการทำงานของเครื่องล้างผัก

จากที่ได้กล่าวถึงวงจรควบคุมต่างๆของเครื่องล้างผักซึ่งเป็นฮาร์ดแวร์ที่ใช้ควบคุมการทำงานของเครื่องล้างผักทั้งหมด แต่มาถึงจุดนี้หลายคนอาจสงสัยว่าฮาร์ดแวร์เพียงเล็กน้อยเหล่านั้นจะทำให้เครื่องล้างผักมีความสมบรูณ์ในการใช้งานได้อย่างไร ซึ่งนั้นเป็นความคิดที่ถูกเพราะว่าความสมบูรณ์ ของระบบนั้นไม่เพียงแต่ฮาร์ดแวร์เท่านั้นแต่ต้องประกอบด้วยซอฟต์แวร์ที่เปรียบเสมือนมันสมองของเครื่องที่จะต้องคอยสั่งการให้ฮาร์ดแวร์ของเครื่องล้างผักทำงานตามลำดับขั้นตอนที่เราได้โปรแกรมไว้โปรแกรมของเครื่องล้างผักเป็นดังนี้

การกำหนด Port ต่าง ๆ เป็นอินพุทและเอาท์พุทของ MCS-51 ดังนี้

SW1 P1.0 จำนวนครั้ง

เก็บค่าจำนวนครั้งใน Num = 20H (1=1ครั้ง , 2=2ครั้ง , 3=3ครั้ง)

แสดงจำนวนครั้งในการล้าง

Led 1 = P0.0 จำนวน 1 ครั้ง

Led 2 = P0.1 จำนวน 2 ครั้ง

Led 3 = P0.2 จำนวน 3 ครั้ง

SW2 P1.1 ระดับน้ำ

เก็บค่าระดับน้ำใน LEVEL = 21H (1 = ระดับ1, 2= ระดับ2, 3=ระดับ3)

Led 4 = P0.3 ระดับ 1

Led 5 = P0.4 ระดับ 2

Led 6 = P0.5 ระดับ 3

SW3 ลดเวลา

SW4 เพิ่มเวลา เก็บค่าเวลาใน Time = 22H

SW5 P1.4 กด Start / Stop

SW6 เป็น Switch กดเพื่อตั้งโปรแกรมล้างผักเองและเพื่อให้เสียงหยุดเมื่อล้างผักเสร็จ

SW7 เป็น Switch โปรแกรม1 เพื่อล้างผักน้อย

SW8 เป็น Switch โปรแกรม2 เพื่อล้างผักมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น Flag (1 = Start , 0 = Stop) เมื่อกดปุ่ม Start / Stop ถ้า Flag = 0 Set Bit Flagเป็น 1 แล้วส่งสัญญาณขับมอเตอร์ แล้วไปยัง Run เพื่อตรวจเช็คระดับน้ำและนับเวลา ถ้า Flag = 1 CLR Bit Flag เป็น 0

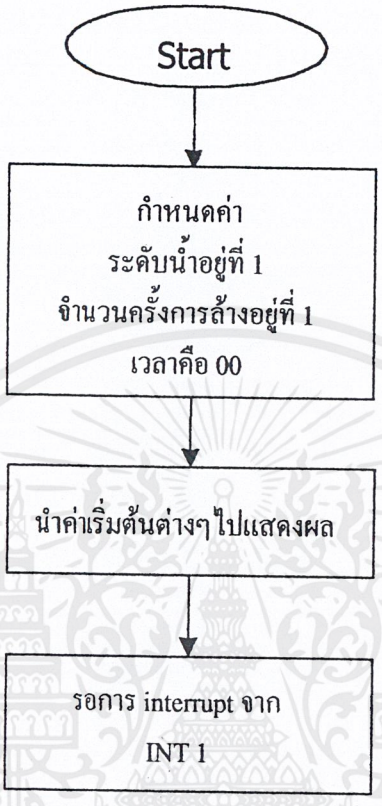
โฟล์วชาร์ทการทำงานของซอฟต์แวร์การควบคุมการทำงานของเครื่องล้างผัก

สามารถอธิบายได้ดังนี้โดยโปรแกรมจะเริ่มทำงานที่โปรแกรมหลัก1คือเป็นการกำหนดค่าเริ่มต้นให้เวลาเริ่มที่ 00 และระดับน้ำกับจำนวนครั้งอยู่ที่ระดับ 1 และต่อจากนั้นแสดงค่าเริ่มต้นออกมาและจะรอรับการ Interrupt จาก INT1 และเมื่อมีการกด SW6, SW7 และSW8 จะเกิดการ INT1 เกิดขึ้น โดยถ้ากด SW6 โปรแกรมจะไปที่โปรแกรมหลัก2 ถ้าเป็นการกด SW7จะไปที่โปรแกรม1 คือล้างผักน้อย ถ้ากดSW8 จะเป็นการล้างผักมาก

โปรแกรมหลัก1 จะเป็นการกำหนด INT0 และกำหนดค่าเริ่มต้นต่างๆและนำไปแสดงผลและรอการ Interrupt จาก INT0 และเมื่อมีการกดสวิตซ์ใดสวิตซ์หนึ่งคือ SW1-SW5 จะเกิดการ Interrupt จาก INT0 และเมื่อมีการกดสวิตซ์ใดสวิตซ์หนึ่งจะเกิดการInterrupt โปรแกรมจะไปทำงานที่โปรแกรมบริการ Interrupt ทันที่โดยโปรแกรมย่อยนี้จะทำการตรวจสอบว่าสวิตซ์ที่ 1 ถูกกดหรือไม่ถ้าถูกกดโปรแกรมจะไม่ทำการเปลี่ยนระดับน้ำและกลับสู่โปรแกรมหลักแต่ถ้าไม่ถูกกดจะไปตรวจสอบสวิตซ์ที่ 2 ว่าถูกกดหรือไม่ ถ้าถูกกดจะไปเปลี่ยนจำนวนครั้งในการล้างและกลับสู่โปรแกรมหลักแต่ถ้าไม้ถูกกด จะไปตรวจว่าสวิตซ์ที่ 3 ถูกกดหรือไม่ ถ้าถูกกดก็จะเพิ่มเวลาในการล้างและกลับสู่โปรแกรมหลักแต่ถ้าไม่ถูกกด จะไปตรวจสอบว่าสวิตซ์ 4 ถูกกดหรือไม่ถ้าถูกกดจะไปลดเวลาในการล้างและกลับสู่โปรแกรมหลักแต่ถ้าไม่ถูกกดจะไปตรวจว่า สวิตซ์ที่ 5 ถูกกดหรือไม่ถ้าถูกกดจะไปโปรแกรมเริ่มการล้างทันที่และพอทำการล้างเรียบร้อยก็จะกลับสู่โปรแกรมหลักเพื่อรอการ Interrupt ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

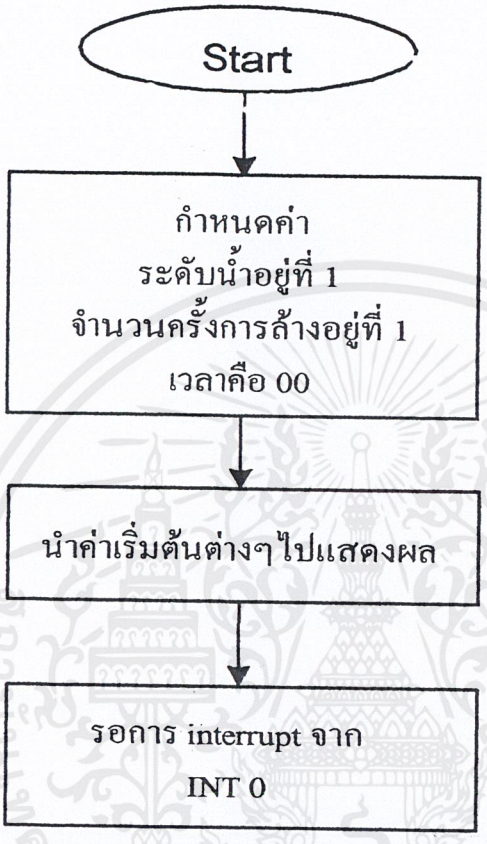
โฟล์วชาร์ทโปรแกรมหลัก 1



รูปที่ 4.1 แสดงโฟล์วชาร์ทโปรแกรมหลัก 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

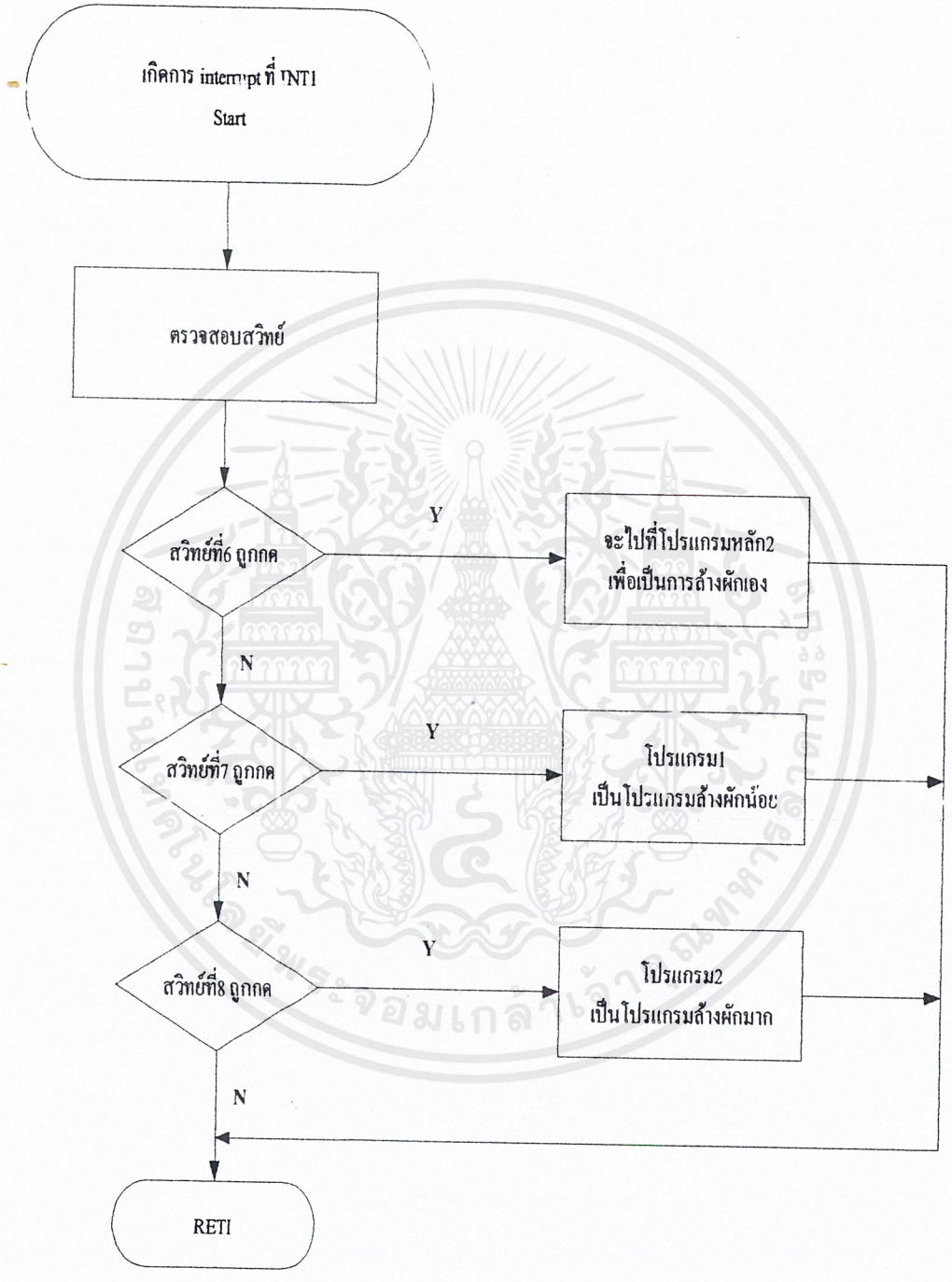
โฟล์วชาร์ทโปรแกรมหลัก2



รูปที่ 4.2 แสดงโฟล์วชาร์ทโปรแกรมหลัก 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

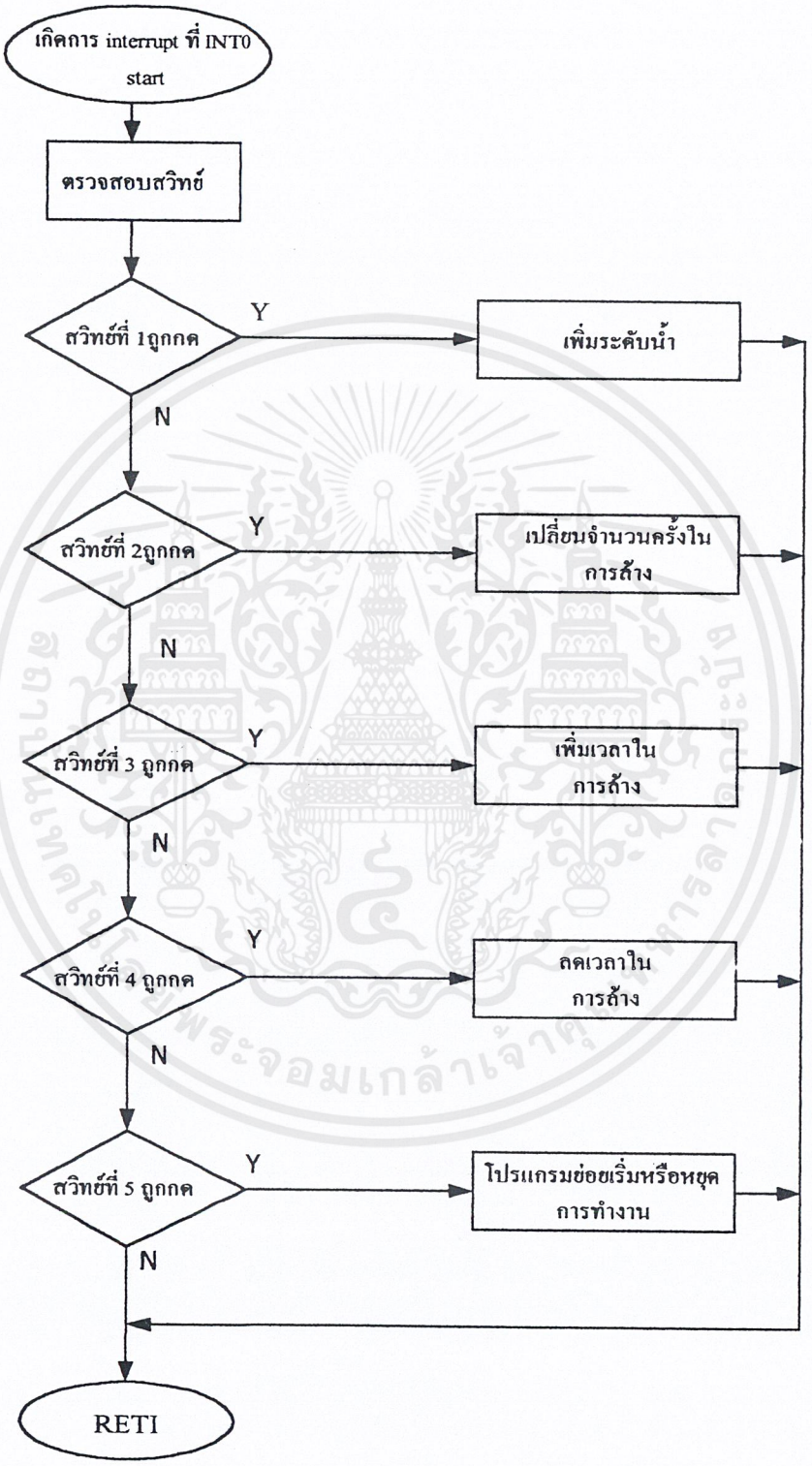
โฟล์วชาร์ทโปรแกรม Interrupt 1



รูปที่ 4.3 แสดงโฟล์วชาร์ท โปรแกรม Interrupt 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟล์วชาร์ทโปรแกรม Interrupt 0



รูปที่ 4.4 แสดงโฟล์วชาร์ทโปรแกรม Interrupt **0**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

จากผลการทดสอบเครื่องล้างผักพบว่าเครื่องสามารถล้างที่เป็นผล เช่น มะม่วง หัวมัน ส้ม ถั่วลิสง ฯลฯ ส่วนผักที่เป็นใบก็สามารถล้างได้ เช่น ผักคะน้า ผักกาดขาว ผักกาดหอม ฯลฯ

เครื่องล้างผักเป็นเครื่องที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์โดยจะมีโปรแกรมในการล้าง 3 โปรแกรมคือ

1. โปรแกรมสำหรับล้างผักจำนวนมาก (Automatic)
2. โปรแกรมสำหรับล้างผักจำนวนน้อย (Automatic)
3. โปรแกรมการล้างผักด้วยตนเอง (Manual)

ในขณะที่เครื่องกำลังทำงานอยู่เราสามารถที่จะให้เครื่องหยุดทำงานได้ เมื่อผู้ใช้ต้องการที่จะแก้ไขโปรแกรมที่ได้โปรแกรมเอาไว้ และเมื่อเครื่องทำงานเสร็จแล้ว จะมีเสียงเตือนเพื่อให้ผู้ใช้ทราบ

ลักษณะของเครื่อง

- ตัวเครื่องล้างผักมีขนาด 12x12x12 นิ้ว
- ใช้กำลังงานประมาณ 350 วัตต์
- ใช้ระบบน้ำไหลเวียนในการชำระล้างสิ่งสกปรกที่ติดมากับผักผลไม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. Ayala,K.J.,"The 8051 Microcoltroller Architecture,Programming and Applications",West Publishing Company,1991
2. ธีรวัฒน์ ประกอบผล, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์",กรุงเทพฯ : ประชาชน บริษัท,155 หน้า,2540
3. คู่มือไอซี CMOS 4000 Series, ซีเอ็ดยูเคชั่น บริษัท

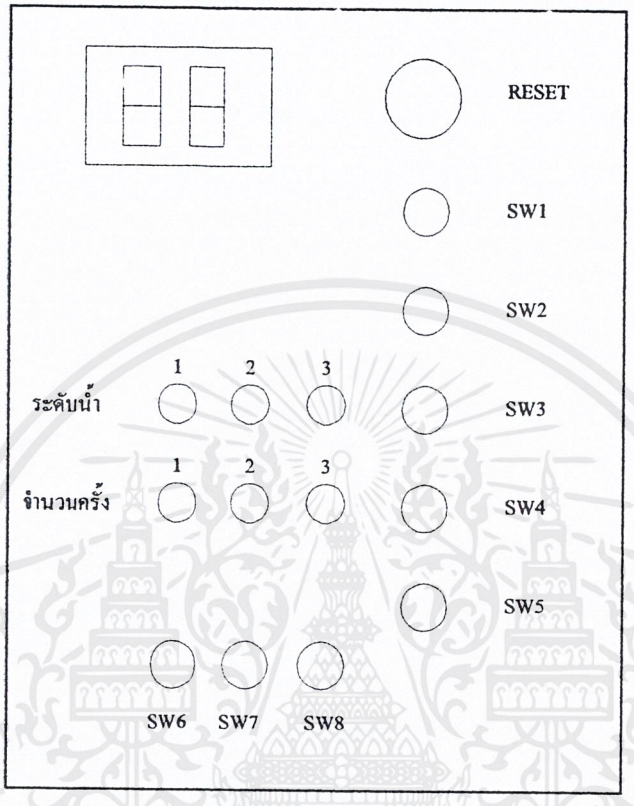
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คำแนะนำในการใช้เครื่อง



SW1 : เป็นสวิตซ์สำหรับเพิ่มเวลาในการล้าง

SW2 : เป็นสวิตซ์สำหรับลดเวลาในการล้าง

SW3 : เป็นสวิตซ์สำหรับเลือกระดับน้ำในการล้าง ซึ่งมีอยู่ 3 ระดับ คือ ระดับ 1, 2 และ 3

SW4 : เป็นสวิตซ์สำหรับเลือกจำนวนครั้งในการล้าง ซึ่งมีอยู่ 3 ครั้ง คือ 1, 2 และ 3

SW5 : เป็นสวิตซ์สำหรับกดเพื่อให้เครื่องเริ่มทำงาน(Start) และสำหรับหยุดเครื่อง (Stop) เมื่อผู้ใช้ต้องการเปลี่ยนแปลงหรือแก้ไขโปรแกรมการล้าง

SW6 : เป็นสวิตซ์สำหรับกดเพื่อตั้งโปรแกรมด้วยตัวผู้ใช้เอง(Manual) คือสามารถที่จะเลือกระดับน้ำ, จำนวนครั้ง และเวลาในการล้างผักด้วยตัวของผู้ใช้เอง

SW7 : เป็นสวิตซ์สำหรับการล้างผักวนน้อยแบบอัตโนมัติ(Automatic) คือตั้งเวลาไว้ที่ 10 นาที, ระดับน้ำอยู่ที่ระดับ 1 และล้างผักเป็นจำนวน 2 ครั้ง

SW8 : เป็นสวิตซ์สำหรับล้างผักจำนวนมากแบบอัตโนมัติ(Automatic) คือ ตั้งเวลาไว้ที่ 20 นาที ระดับน้ำอยู่ที่ระดับ 3 และล้างเป็นจำนวน 3 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อควรระวังในการใช้เครื่อง

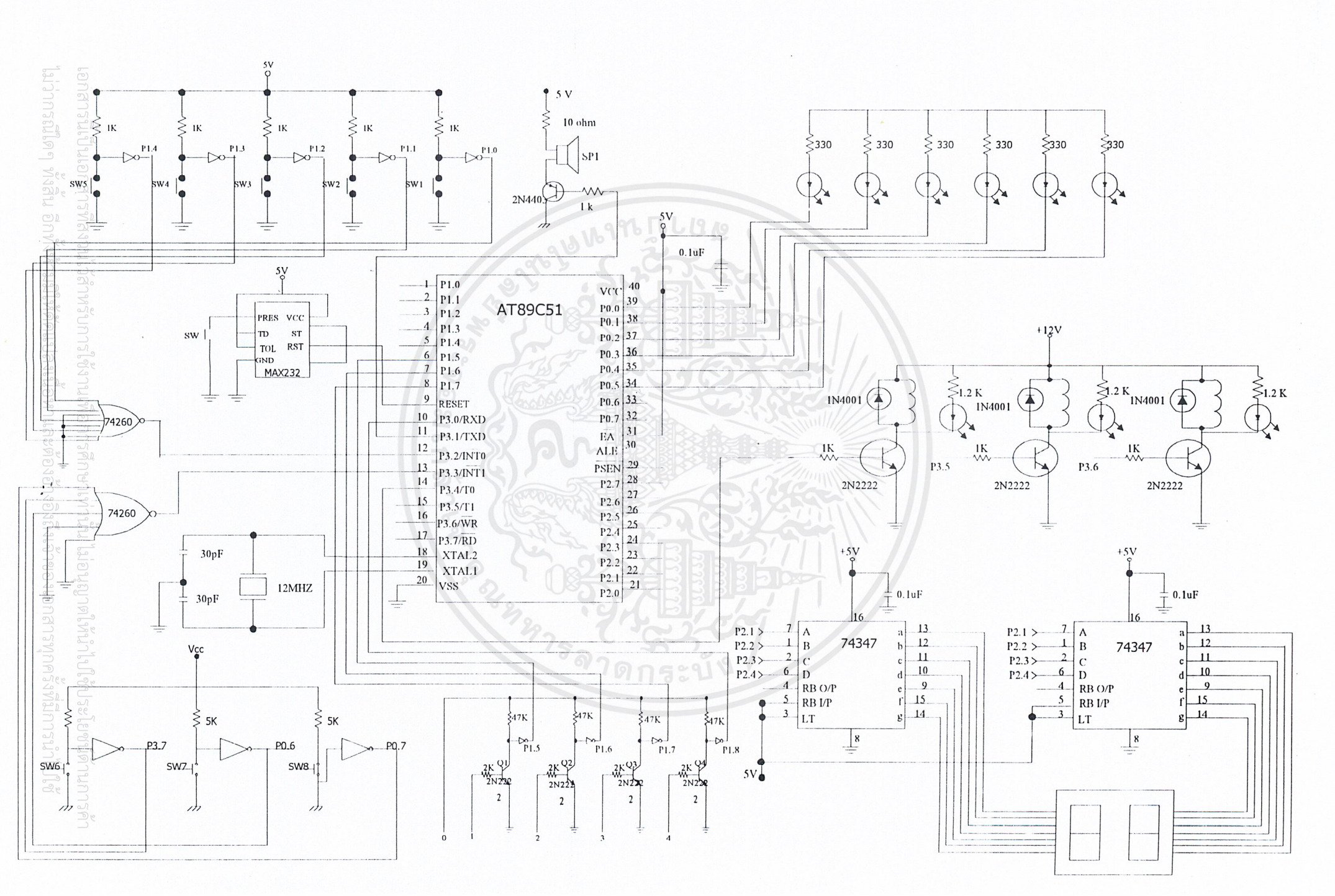
1. ควรอ่านคู่มือการใช้ก่อนการใช้งาน
2. ตรวจสอบสภาพของสายไฟให้อยู่ในสภาพเรียบร้อย ไม่มีการชำรุดเสียหาย
3. ไม่ควรเปิดฝาเครื่องควบคุมตัวเครื่อง
4. ไม่ควรแก้ไขหรือดัดแปลงวงจรภายในตัวเครื่อง
5. ตรวจให้แน่ใจว่าสวิตซ์และสายไฟอยู่ในตำแหน่งที่ถูกต้อง
6. ควรวางสายไฟให้ห่างจากบริเวณที่มีน้ำ
7. วางกล่องควบคุมให้อยู่ในตำแหน่งที่เหมาะสมเพื่อป้องกันการกระแทกและอันตรายที่อาจเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ผังวงจรรวมเครื่องล้างผัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


5V
1K
P1.4
P1.3
P1.2
P1.1
P1.0
SW5
SW4
SW3
SW2
SW1
5 V
10 ohm
SP1
2N4401
1 k
330
0.1uF
AT89C51
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
RESET
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
XTAL2
XTAL1
VSS
VCC
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
EA
ALE
PSEN
P2.7
P2.6
P2.5
P2.4
P2.3
P2.2
P2.1
P2.0
SW
PRES
TD
TOL
GND
VCC
ST
RST
MAX232
74260
30pF
12MHZ
+12V
1N4001
1.2 K
2N2222
P3.5
P3.6
+5V
74347
A
B
C
D
RB O/P
RB I/P
LT
Vcc
5K
SW6
SW7
SW8
P3.7
P0.6
P0.7
47K
2K
2N2222
Q1
Q2
Q3
Q4
P1.5
P1.6
P1.7
P1.8

ภาคผนวก ค

Source Code MCS-51

```
        NUM EQU 20H
        LEVEL EQU 21H
        TIME  EQU 22H
        TIME_HEX EQU 25H
        COUNT_TIMEH EQU 23H
        COUNT_TIMEL EQU 24H
        S_FLAG EQU 7FH
        TP_FLAG EQU 7EH

ORG 0H
        SJMP MAIN1
ORG 3H
        JMP SWITCH_V
ORG 0BH
        JMP TIMER0_INT
ORG 13H
        JMP SW_
ORG 40H

MAIN1:  MOV TIME,#0
        MOV TIME_HEX,#0
        MOV NUM,#1
        MOV LEVEL,#1
        MOV P2,TIME
        ACALL NUM_DISPLAY
        ACALL LEVEL_DISPLAY
        CLR TP_FLAG
        SETB EA
SETB EX1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SJMP $
          ;---------------------- SW PROGRAM ----------------------
SW_:    JB TP_FLAG,SW_STOP
          ;--------------------------------------------------------------------
        JNB P0.6,Z1
        CLR EX1
        SETB EX0
        ACALL SOUND
        SETB S_FLAG
        MOV TIME,#20H
        MOV TIME_HEX,#20H
        MOV NUM,#3
        MOV LEVEL,#3
        MOV P2,TIME
        ACALL NUM_DISPLAY
         ACALL LEVEL_DISPLAY
        POP ACC
        POP ACC
        MOV DPTR,#START_STOP
        PUSH DPL
        PUSH DPH
        RETI
Z1:     JNB P0.7,Z2
        CLR EX1
        SETB EX0
        ACALL SOUND
        SETB S_FLAG
        MOV TIME,#10H
        MOV TIME_HEX,#10H
        MOV NUM,#2
        MOV LEVEL,#1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV P2,TIME
        ACALL NUM_DISPLAY
        ACALL LEVEL_DISPLAY
        POP ACC
        POP ACC
        MOV DPTR,#START_STOP
        PUSH DPL
        PUSH DPH
        RETI
Z2:     JNB P3.7,Z3
        ACALL SOUND
        CLR EX1
        POP ACC
        POP ACC
        MOV DPTR,#MAIN
        PUSH DPL
        PUSH DPH
Z3:     RETI
        ;-------------------------------------------------------------------
SW_STOP: JNB P3.7,Z3
        ACALL SOUND
        ACALL SOUND
        ACALL SOUND
        POP ACC
        POP ACC
        MOV DPTR,#MAIN1
        PUSH DPL
        PUSH DPH
        RETI
        ;-------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAIN:       MOV TIME,#0
            MOV TIME_HEX,#0
            MOV NUM,#1
            MOV LEVEL,#1
MAIN_:      MOV P2,TIME
            ACALL NUM_DISPLAY
            ACALL LEVEL_DISPLAY
            CLR S_FLAG
            SETB EA
            SETB EX0
            SJMP $

;-----------------------------interrupt rutine-----------------------------

SWITCH_V:   JB S_FLAG,SW_START

;-----------------SWITCH VERIFY  WHEN STATUS STOP-----------------
            JNB P1.1,PP1
            ACALL NUMBER
            ACALL NUM_DISPLAY
            ACALL SOUND
            ACALL DELAY
            RETI
PP1:        JNB P1.0,PP2
            ACALL LEVEL_WATER
            ACALL SOUND
            ACALL LEVEL_DISPLAY
            ACALL DELAY
            RETI
PP2:        JNB P1.2,PP3
            ACALL DEC_TIME
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL SOUND
                ACALL TIME_DISPLAY
                ACALL DELAY
                RETI
PP3:            JNB P1.3,PP4
                ACALL INC_TIME
                ACALL TIME_DISPLAY
                ACALL SOUND
                ACALL DELAY
                RETI
PP4:            JNB P1.4,PP5
                JB P1.4,$
                ACALL SOUND
                ACALL DELAY
                SETB S_FLAG
                POP ACC
                POP ACC
                MOV DPTR,#START_STOP
                PUSH DPL
                PUSH DPH

PP5:            RETI
;----------------SWITCH VERIFY WHEN STATUS START----------------

SW_START:       JNB P1.4,AA1
                ACALL SOUND
                ACALL DELAY
                CLR ET0
                CLR TR0
                SETB P3.4
                SETB P3.5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB P3.6
                POP ACC
                POP ACC
                MOV DPTR,#MAIN_
                PUSH DPL
                PUSH DPH
AA1:            RETI

;---------------กำหนดจำนวนครั้งในการล้าง----------------------------
;NUM=เก็บจำนวนครั้งในการล้าง
;REG : A

NUMBER:         MOV A,NUM
                CJNE A,#3,NN
                MOV NUM,#1
                RET
NN:             INC NUM
                RET

;---------------กำหนด ระดับน้ำ----------------------------------
;LEVEL=WATER LEVEL
;REG : A

LEVEL_WATER:    MOV A,LEVEL
                CJNE A,#3,LL
                MOV LEVEL,#1
                RET
LL:             INC LEVEL
                RET
;---------------กำหนดเวลาในการล้าง----------------------------
;=========DEC_TIME==========
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;REG : A

DEC_TIME:   MOV A,TIME
            CJNE A,#0,DD
            MOV TIME,#99H
            MOV TIME_HEX,#99
            RET
DD:         DEC A
            DEC TIME_HEX
            PUSH ACC
            ANL A,#0FH
            CJNE A,#0FH,DD1
            POP ACC
            ANL ACC,#0F0H
            ORL ACC,#09H
            MOV TIME,A
            RET
DD1:        POP ACC
            MOV TIME,ACC
            RET

;==========INC_TIME==========

INC_TIME:   MOV A,TIME
            CJNE A,#99H,INN
            MOV TIME,#0
            MOV TIME_HEX,#0
            RET
INN:        ADD A,#1
            DA A
            MOV TIME,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LLL:        CLR P3.1
            ACALL DELAY2
            SETB P3.1
            ACALL DELAY2
            DJNZ R1,LLL
            MOV R2,#3
ABC:        ACALL DELAY2
            DJNZ R2,ABC
            JMP LLL1
            JMP MAIN1
            RET

;-------------------------น้ำทิ้ง-------------------------

WARTER_OUT:     CLR ET0
            SETB P3.4
            CLR P3.5
            CLR P3.6
K:          JB P1.5,$
            ACALL DELAY
            JB P1.5,K
            ACALL DELAY
            JB P1.5,K
            SETB P3.5
            SETB P3.6
            RET

;-------------------------น้ำเข้าถัง-------------------------

WARTER_IN_TANK:MOV A,LEVEL
            SETB P3.5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB P3.6
        CJNE A,#3,L2
L3:     CLR P3.4
        SETB P3.6
        ACALL DELAY
        ACALL DELAY
        ACALL DELAY
        CLR P3.6
        ACALL DELAY
        ACALL DELAY
        ACALL DELAY
U:      JNB P3.0,L3
        ACALL DELAY
        JNB P3.0,L3
        ACALL DELAY
        JNB P3.0,L3
        ACALL DELAY
        JNB P3.0,L3
        ACALL DELAY
        JNB P3.0,L3
        SETB P3.4
        RET
L2:     CJNE A,#2,L1
        CLR P3.4
        SETB P3.6
        ACALL DELAY
        ACALL DELAY
        ACALL DELAY
        CLR P3.6
        ACALL DELAY
        ACALL DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL DELAY
U1:     JNB P1.7,L2
        ACALL DELAY
        JNB P1.7,L2
        ACALL DELAY
        JNB P1.7,L2
        ACALL DELAY
        JNB P1.7,L2
        ACALL DELAY
        JNB P1.7,L2
        SETB P3.4
        RET

L1:     CLR P3.4
        SETB P3.6
        ACALL DELAY
        ACALL DELAY
        ACALL DELAY
        CLR P3.6
        ACALL DELAY
        ACALL DELAY
        ACALL DELAY
U2:     JNB P1.6,L1
        ACALL DELAY
        JNB P1.6,L1
        ACALL DELAY
        JNB P1.6,L1
        ACALL DELAY
        JNB P1.6,L1
        ACALL DELAY
        JNB P1.6,L1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB P3.4
                RET

;----------------DISPLAY----------------------------------------
;=========แสดงผลจำนวนครั้ง=============================

NUM_DISPLAY:    SETB P0.0
                SETB P0.1
                SETB P0.2
                MOV A,NUM
                CJNE A,#1,N_LED1
                CLR P0.0
                RET
N_LED1:         CJNE A,#2,N_LED2
                CLR P0.1
                RET
N_LED2:         CLR P0.2
                RET

;=========แสดงผลระดับน้ำ=============================

LEVEL_DISPLAY:  SETB P0.3
                SETB P0.4
                SETB P0.5
                MOV A,LEVEL
                CJNE A,#1,L_LED1
                CLR P0.3
                RET

L_LED1:         CJNE A,#2,L_LED2
                CLR P0.4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
L_LED2:         CLR P0.5
                RET

;==========TIME DISPLAY==========
;TIME= ตัวแปรส่งผ่านเท่ากับค่าที่ต้องการแสดงเป็น BCD

TIME_DISPLAY:       MOV P2,TIME
                RET

;----------SOUND----------
SOUND:          CLR P3.1
                ACALL DELAY1
                SETB P3.1
                RET

;-------------------------------------------------------------------------

;REG: R5,R6,R7
DELAY:          MOV R5,#3
DY2:            MOV R6,#0
DY1:            DJNZ R7,$
                DJNZ R6,DY1
                DJNZ R5,DY2
                RET
;-------------------------------------------------------------------------

DELAY2:         MOV R5,#10
F2:             MOV R6,#0AH
F1:             DJNZ R7,$
                DJNZ R6,F1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ R5,F2
                RET
;-------------------------------------------------------------------------------

DELAY1:         MOV R5,#0fh
jj:             MOVR6,#0
                DJNZR6,$
                DJNZ R5,jj
                RET
;-------------------------------------------------------------------------------

;interupt service rotine timer0
;REG R0,R1
TIMER0_INT: PUSH 0
                PUSH 1
                CLR TR0
                MOV TH0,#44H
                MOV TL0,#44H
                SETB TR0
                MOV R0,COUNT_TIMEL
                MOV R1,COUNT_TIMEH
                INC R0
                CJNE R0,#0,TIM1
                INC R1
TIM1:           CJNE R1,#4,TIM2
                CJNE R0,#0E0H,TIM2
                MOV COUNT_TIMEL,#0
                MOV COUNT_TIMEH,#0
                ACALL DEC_TIME
                ACALL TIME_DISPLAY
                SJMP INT_OUT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TIM2:       MOV COUNT_TIMEL,R0
            MOV COUNT_TIMEH,R1
INT_OUT:
            POP 1
            POP 0
            RETI

            END
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

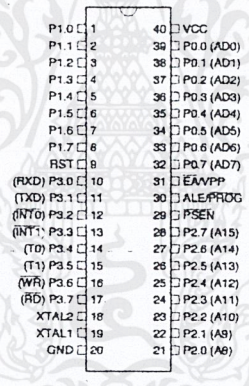
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt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

Description

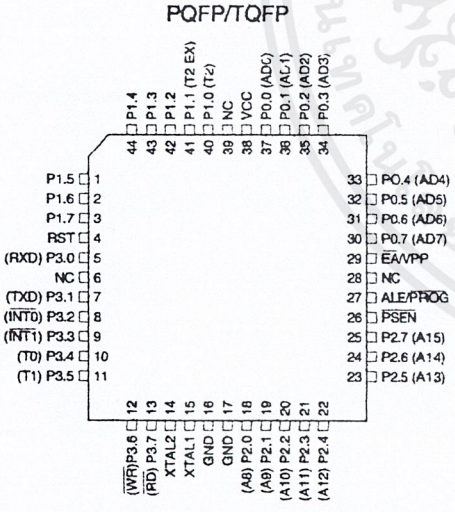
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and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

Pin Configurations

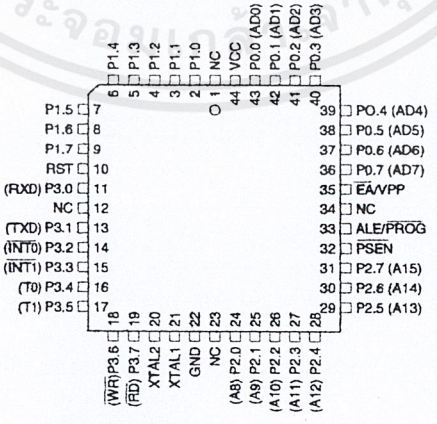
PDIP



PQFP/TQFP



PLCC



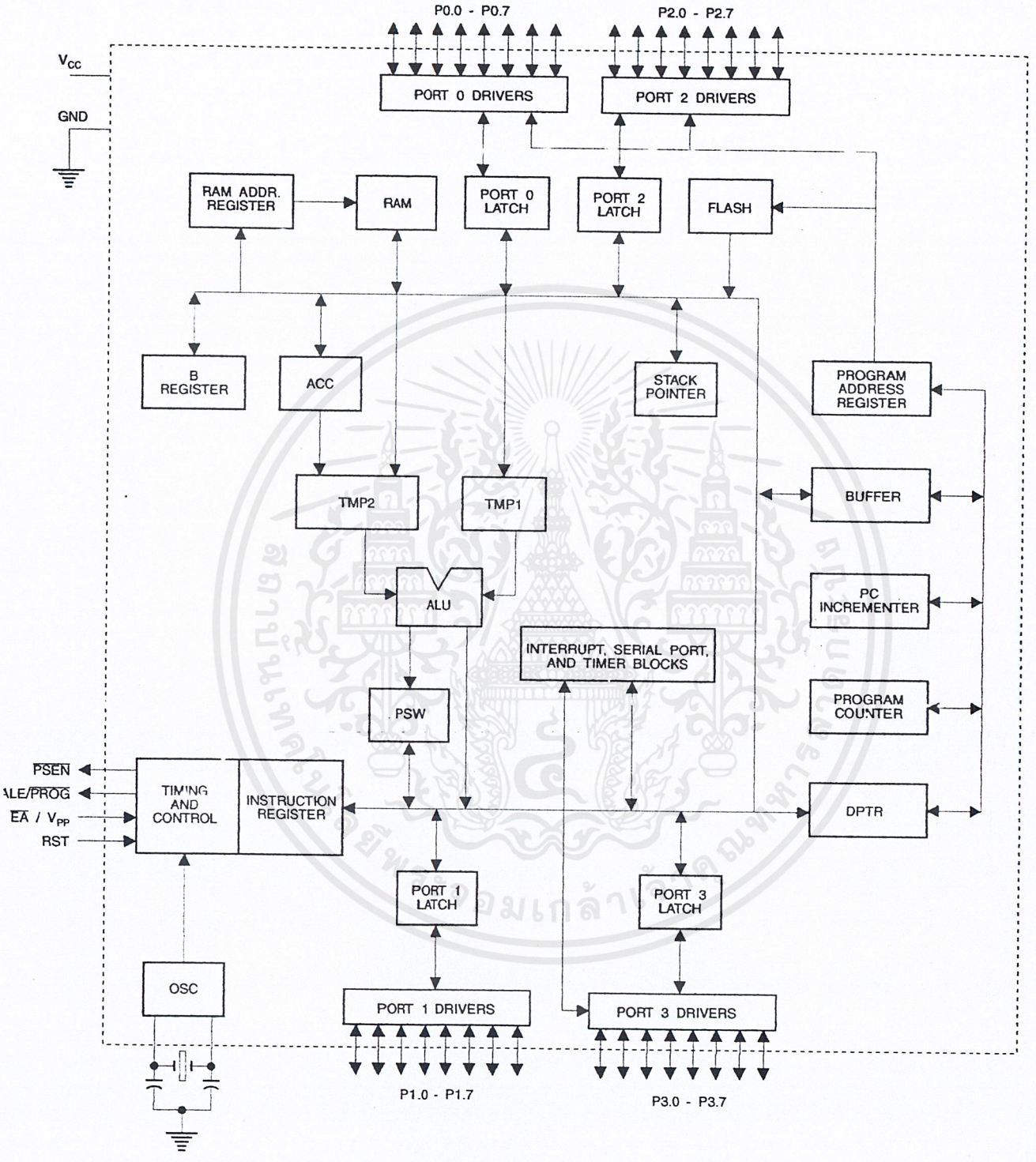
8-bit Microcontroller with 4K Bytes Flash

AT89C51

Rev. 0265G–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

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

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

Port 1

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

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

Port 2

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

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

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

Port 3

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

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

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

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

RST

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

ALE/$\overline{PROG}$

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pulse is skipped during each access to external Data Memory.

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

$\overline{\text{PSEN}}$

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

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

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

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

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

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

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

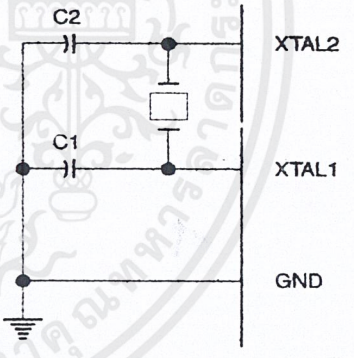
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

Idle Mode

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

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

Figure 1. Oscillator Connections



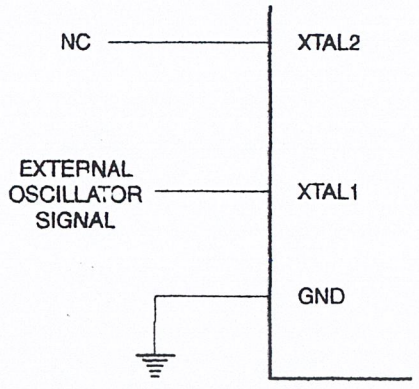
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function 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 V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

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

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

Lock Bit Protection Modes

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash

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

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

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

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

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

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

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

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

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

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

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

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

Programming Interface

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 3. Programming the Flash

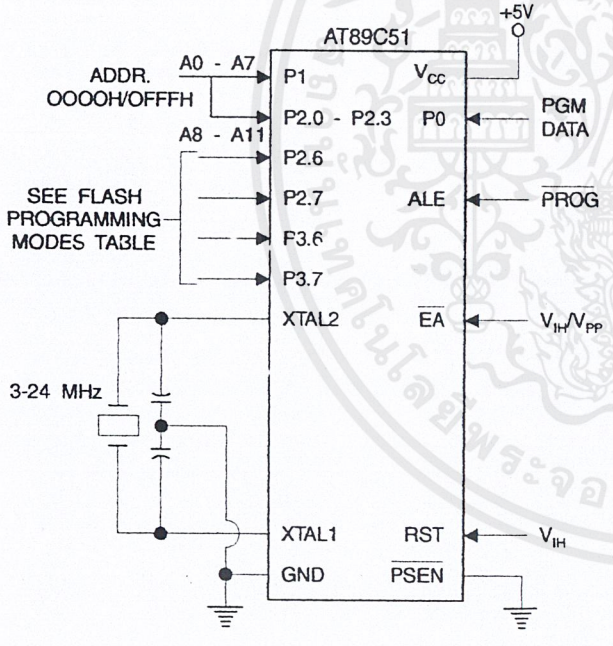
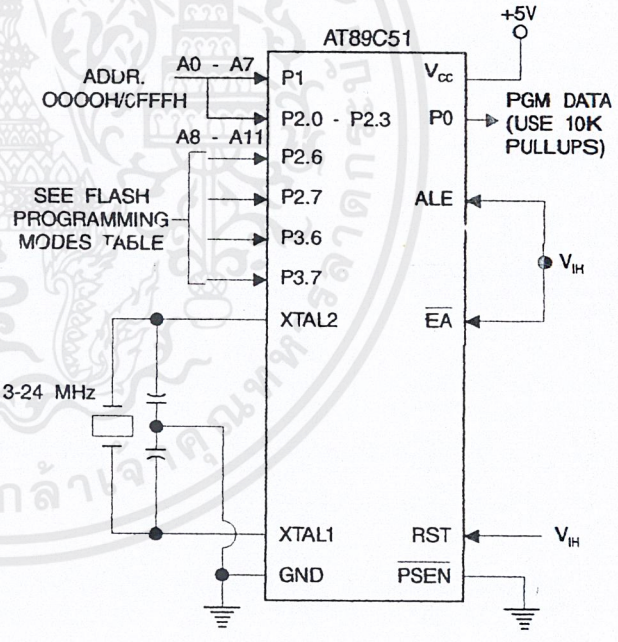
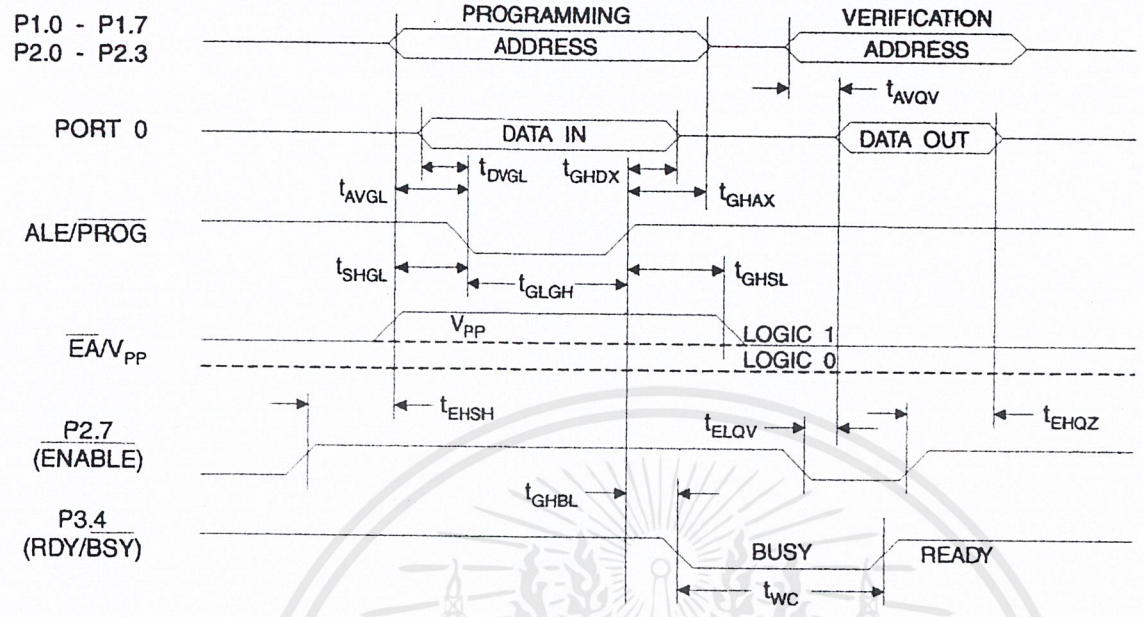


Figure 4. Verifying the Flash

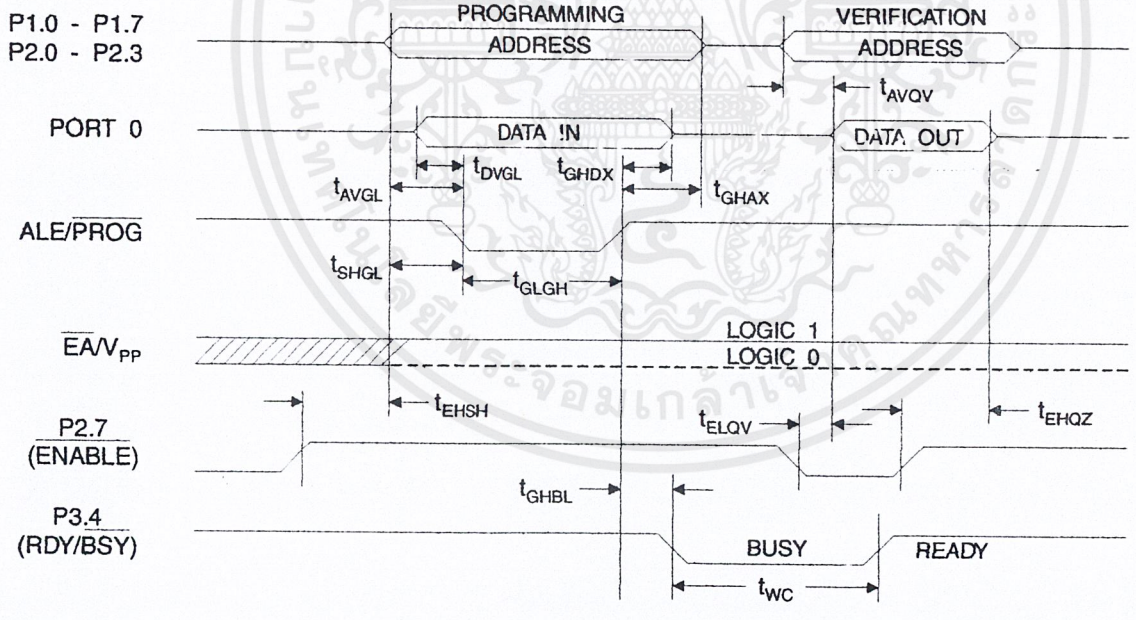


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

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

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

DC Characteristics

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

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

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port: Port 0: 26 mA
Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

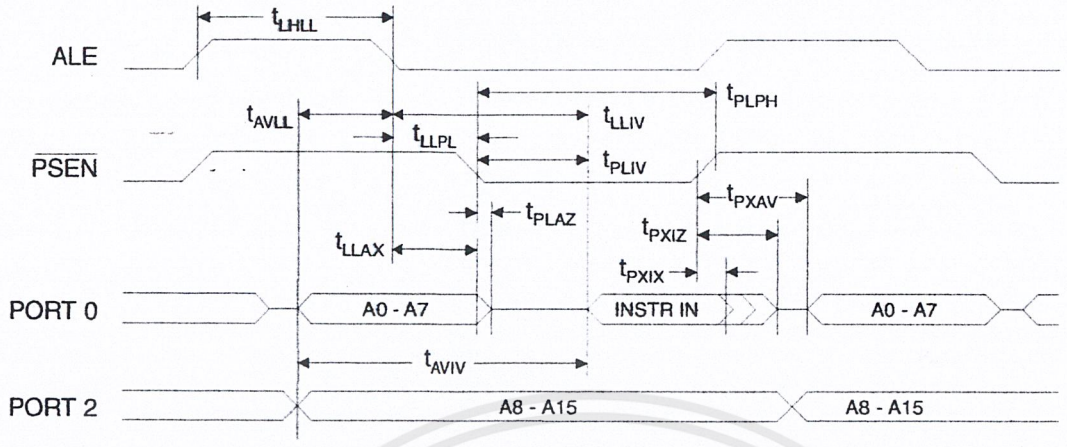
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

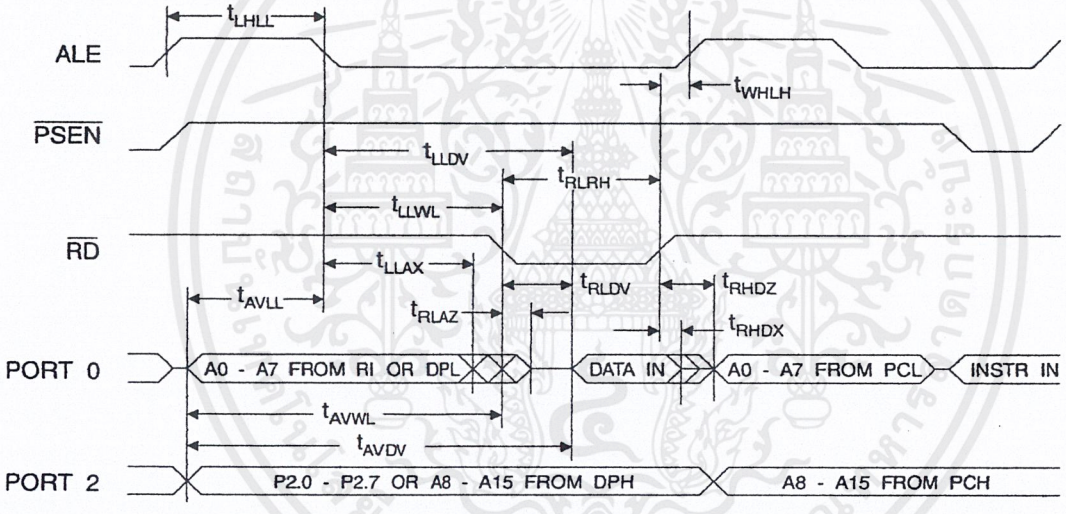
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

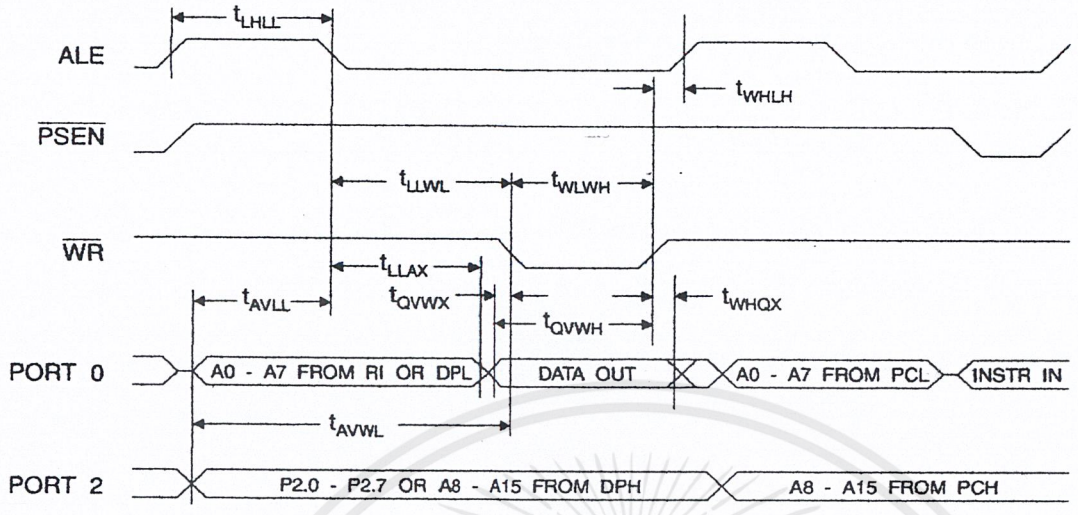


External Data Memory Read Cycle

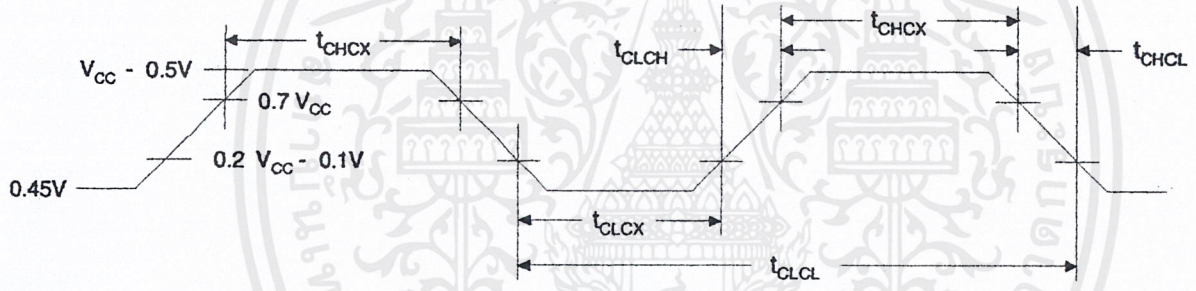


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

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

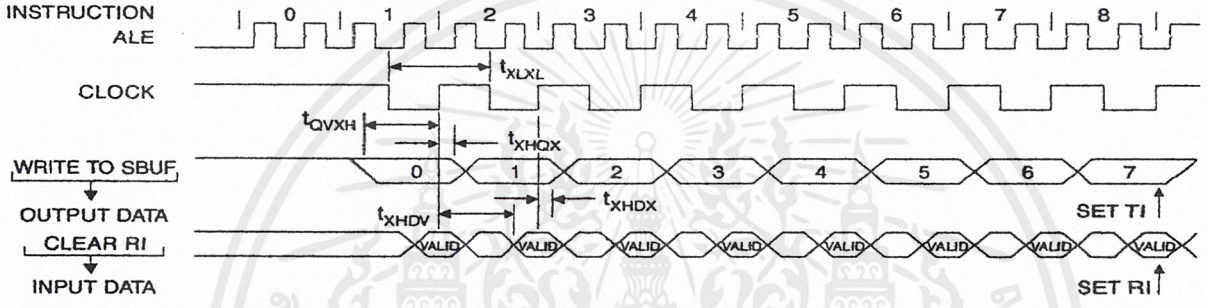
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

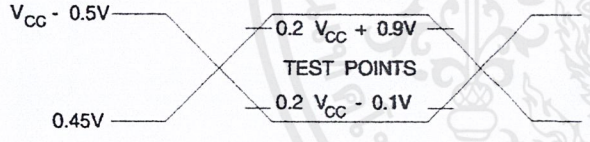
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

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

Shift Register Mode Timing Waveforms

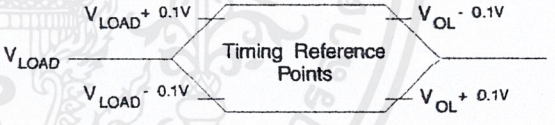


AC Testing Input/Output Waveforms[1]



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

Float Waveforms[1]



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

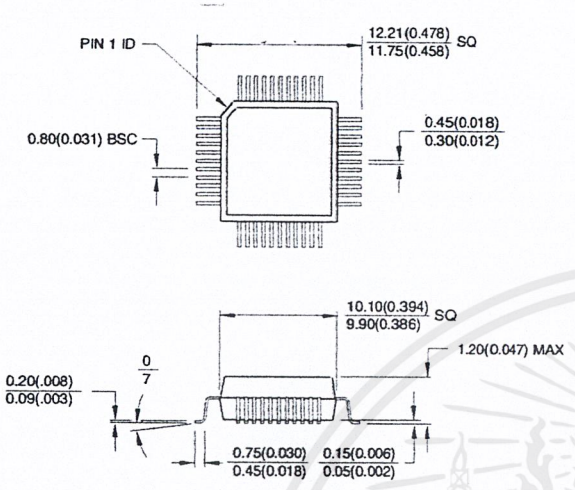
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial
		AT89C51-12JC	44J	(0°C to 70°C)
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial
		AT89C51-12JI	44J	(-40°C to 85°C)
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial
		AT89C51-16JC	44J	(0°C to 70°C)
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial
		AT89C51-16JI	44J	(-40°C to 85°C)
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial
		AT89C51-20JC	44J	(0°C to 70°C)
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial
		AT89C51-20JI	44J	(-40°C to 85°C)
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

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

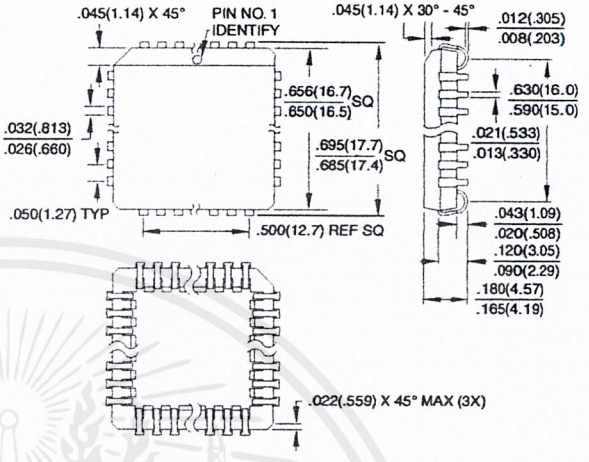
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

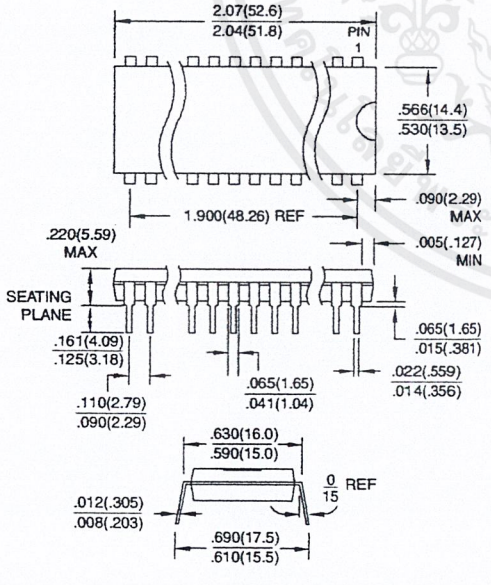


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC

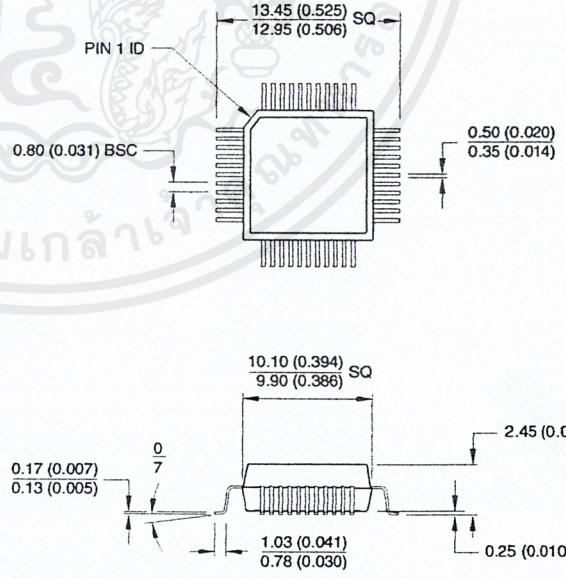


Controlling dimension: millimeters

40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 2000.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0265G-02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้