

ระบบน้ำหยด
WATER DRIP SYSTEM



โดย
นาย ปรีดา เทียบกว้าง
นาย อนุชิต เหมหอมวงษ์

เลขหมู่.....
เลขทะเบียน 42246
วัน, เดือน, ปี 16 พ.ศ. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบควบคุมน้ำหยด

โดย นาย ปรีดา เทียบกว้าง เลขประจำตัว 42015558
นาย อรุณจิต เหมหอมวงษ์ เลขประจำตัว 42015583

อาจารย์ที่ปรึกษา อาจารย์ บุณย์ชนะ ภูระหงษ์

ภาควิชา เทคนิคอุตสาหกรรม

สาขาวิชา เทคโนโลยีอิเล็กทรอนิกส์

ปีการศึกษา 2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้
ปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร
บัณฑิต

คณะกรรมการการสอบปริญญานิพนธ์

..... ประธานกรรมการ

(.....)

..... กรรมการ

(.....)

..... กรรมการ

(.....)

..... กรรมการ

(.....)

..... กรรมการ

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบควบคุมน้ำหยด

โดย นายปริดา เทียบกว้าง เลขประจำตัว 42015558
 นายอนุชิต เหมหอมวงษ์ เลขประจำตัว 42015583

อาจารย์ที่ปรึกษา อาจารย์ บุญยชนะ ภูระหงษ์

ภาควิชา เทคนิคอุตสาหกรรม

สาขาวิชา เทคโนโลยีอิเล็กทรอนิกส์

ปีการศึกษา 2544

บทคัดย่อ

ในปัจจุบันไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) มีใช้กันอย่างแพร่หลาย และได้มีการนำไปประยุกต์ใช้งานในรูปแบบที่ต่างกัน ในปริญญานิพนธ์ฉบับนี้ได้กล่าวถึง การนำไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 (MCS 51) มาประยุกต์ใช้งาน ในการควบคุมระบบน้ำหยด (WATER DROP SYSTEM) ตั้งแต่การให้น้ำแก่พืช, การควบคุม Solenoid Valve, ควบคุมปั้มน้ำ, แสดงผลออกที่จอ LCD เป็นต้น

PROJECT **WATER DRIP SYSTEM**
BY **Mr. Preeda TeabGrang**
 Mr. Anuchit HemHomWong

ADVISOR **BoonChana Phurahong**

DEPARTMENT **INDUSTIAL TECHNOLOGY**
PROGRAMIN **ELECTRONIC TECHNOLOGY**
ACDEMIC YEAR **2001**

Abstract

Recently ,Single Chip Microcontroller has been used commonly. It has been applied and use in different formats. This thesis is about applying the Single Chip Microcontroller (MCS-51) to control The Water Drop System from watering plants,control solenoid valve,control pump and display to LCD.

กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณบิดามารดาที่ให้เราเกิดมา ขอบพระคุณอาจารย์ที่ปรึกษาและ บริษัท โฟร์บรอส ซัพพลายส์ ที่ให้คำปรึกษาในการเริ่มต้นทำโครงการเล่มนี้ ตลอดจนข้อคิดเห็นและแนวทางในการแก้ไขปัญหาต่างๆ จนสามารถสำเร็จมาได้ด้วยดี และอาจารย์ภาควิชา เทคนิคอุตสาหกรรมที่ให้คำแนะนำในการปฏิบัติงานต่างๆ รวมถึงการเรียนการสอนในวิชา ไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ ในภาคเรียนที่ 2 ปีการศึกษา 2542 ที่ได้ความรู้ในเนื้อหาวิชาที่เกี่ยวข้องกับโครงการชิ้นนี้ โดยผู้จัดทำรายงานได้นำความรู้ที่เรียนผ่านมา นำมาประยุกต์ใช้และนำมาแก้ไขปัญหาต่างๆ ซึ่งเป็นประโยชน์อย่างมากทำให้สามารถแก้ไขปัญหาต่างๆสำเร็จได้ด้วยดี ซึ่งถ้าขาดบุคคลดังกล่าวมาแล้วผู้จัดทำรายงานก็คงไม่สามารถที่จะทำงานให้เสร็จลงได้

โดยผู้จัดทำรายงานได้นำความรู้ที่ได้รับมา นำมาอ้างอิงใช้เป็นเนื้อหาของรายงานฉบับนี้ โดยสุดท้ายนี้ผู้จัดทำรายงานใคร่ขอขอบพระคุณบุคคลที่ได้กล่าวถึงเป็นอย่างสูง ผู้จัดทำรายงานขอระลึกถึงความขอบพระคุณเป็นอย่างยิ่ง

นาย ปรีดา เทียบกว้าง
นาย อนุจิต เหมหอมวงษ์

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความหมายของการชลประทานแบบหยด	1
1.2 ข้อดีของการชลประทานแบบหยด	5
1.3 ข้อเสียและปัญหาของการชลประทานแบบหยด	6
บทที่ 2 ระบบการชลประทานแบบหยด	8
2.1 องค์ประกอบของระบบชลประทานแบบหยด(Drop System Components)	8
2.2 ชนิดของระบบชลประทานแบบหยด (Types of Drop Irrigation System)	10
2.3 คุณลักษณะและชนิดของหัวปล่อยน้ำ (Charateristic and Type of Emitter)	10
บทที่ 3 การออกแบบระบบน้ำหยดเพื่อให้น้ำแก่พืช	15
บทที่ 4 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิปเดี่ยวตระกูล 51	22
4.1 คุณสมบัติของไมโครคอนโทรลเลอร์ MCS 51	22
4.1.1 โครงสร้างภายในของ MCS 51	22
4.1.2 โครงสร้างภายนอกของ MCS 51	23
4.1.3 การจัดหน่วยความจำ	24
4.1.4 รีจิสเตอร์หน้าที่พิเศษ (SFR)	24
4.1.5 รีจิสเตอร์ใช้งานทั่วไป	26
4.1.6 หน่วยความจำโปรแกรม	26
4.1.7 โปรแกรมแปลภาษา Assembly (Assembler)	27
4.2 การใช้งานโปรแกรม SXA51 ASSEMBLER	27
4.2.1 รายละเอียดทั่วไปของ SXA51 ASSEMBLER	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.2.2 การใช้งาน SXA51 ASSEMBLER	27
4.2.3 ข้อกำหนดของ SOURCE FILE	29
4.2.4 การใช้งาน SXA51	30
4.3 GRAPHIC LCD	31
4.3.1 คำสั่งควบคุมของ LCD	33
4.3.2 INTERFACE LCD DV-12864	35
4.3.3 คีบอร์ดแบบเมตริก 4 x 4	36
4.3.4 ขั้นตอนการเขียนโปรแกรมตรวจสอบสวิตช์มีดังนี้	37
บทที่ 5 การสร้างและประกอบโครงงาน	39
5.1 คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์	39
5.1.1 ลักษณะทั่วไปของบอร์ดควบคุมไมโครคอนโทรลเลอร์	39
5.2 วงจรควบคุม SOLENOID VALE	41
5.2.1 ส่วนประกอบทาง HARDWARE	41
5.2.2 ส่วนประกอบทาง SOFTWARE	42
บทที่ 6 การทดลองโครงงาน	46
บทที่ 7 สรุปผลการทดลองและแนวทางการพัฒนา	51
บรรณานุกรม	52
ภาคผนวก ก ซอร์สโค้ดของโปรแกรม	53
ภาคผนวก ข Data Sheets	78

สารบัญตาราง

ตารางที่	หน้า
3.1 ค่าสัมประสิทธิ์การให้น้ำของพืช	16
3.2 ปริมาณการระเหยน้ำจากการถาดวัดการระเหยน้ำแบบ A	16
3.3 อัตราการซึมของน้ำผ่านดินชนิดต่างๆ	17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ

รูปที่	หน้า
1.1 องค์ประกอบอุปกรณ์ต่างๆของระบบชลประทานน้ำหยด	3
1.2 รูปการใช้งานจริง	4
2.1 หัวปล่อยน้ำแบบทางไหลยาวและคดเคี้ยวไปมา	12
2.2 หัวปล่อยน้ำแบบน้ำไหลเข้าด้านข้างแบบน้ำวน	13
2.3 หัวปล่อยน้ำแบบต่อออกหลายจุด	14
3.1 พื้นที่ กว้าง 50 ม. x ยาว 100 ม	15
3.1.1 ขนาดของพื้นที่เพื่อใช้ในการวางแผนปลูกส้ม	15
3.2 ลักษณะการแบ่งโซนสวนส้มออกเป็น 4 โซน	20
3.3 ลักษณะการแบ่งโซนสวนส้มออกเป็น 4 โซน	21
4.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์	22
4.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์	23
4.3 รีจิสเตอร์หน้าที่พิเศษ (SFR)	25
4.4 แสดงการแบ่งการควบคุม	31
4.5 การตั้งค่า Display start line	32
4.6 โครงสร้างภายในและขาควบคุม	35
4.7 การต่อแบบเมตริก (Matrix switch)	36
4.8 คีย์บอร์ดแบบเมตริก 4 x 4	37
4.9 บล็อกไดอะแกรม Scankey board	38
5.1 วงจรควบคุมระบบน้ำหยด	40
5.2 วงจรจรขับโซลินอยด์วาล์ว	41
5.3 ส่วนประกอบทาง Hard Ware	42
5.4 บล็อกไดอะแกรมของการเลือกโหมดการทำงาน	43
5.5 บล็อกไดอะแกรมโหมด MANUAL	44
5.6 บล็อกไดอะแกรม เลือกวิน	45
6.1 ลักษณะหน้าจอของ LCD ขณะเริ่มเปิด	46
6.2 การเลือกโหมดการทำงาน	46
6.3 การเลือกโหมดการ AUTOMATIC	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ (ต่อ)

รูปที่	หน้า
6.4 การเลือกโหมด MANUAL	47
6.5 ป้อนวัน เดือน ปี ที่ปลูกของพืช	47
6.6 ระบบควบคุมน้ำหยด	48
6.7 การควบคุมการปล่อยน้ำ	48
6.8 หัวปล่อยน้ำแบบ Spayer	49
6.9 หัวปล่อยน้ำแบบหยด	49
6.10 แสดงหัวปล่อยน้ำแบบ Springer	50



บทที่ 1

บทนำ

วิธีการให้น้ำแก่พืชนั้น ปกติปกติสามารถทำกันได้หลายวิธี นับตั้งแต่การปล่อยน้ำให้ไหลท่วมเป็นผืนยาว (Border) หรือแบบอ่าง (Basin) การให้น้ำแบบร่อง (Furrow) และระบบฉีดฝอย (Sprinkler) จนกระทั่งถึงการให้น้ำแบบหยด สำหรับการให้น้ำแต่ละวิธีนั้น ต่างก็มีข้อดีและข้อจำกัดแตกต่างกันออกไปขึ้นอยู่กับสภาพการปลูกพืชแต่ละท้องที่ แหล่งน้ำ ปัญหาทางเศรษฐกิจและวิชาการ แต่อย่างไรก็ตามการให้น้ำแบบหยด สามารถนำไปใช้กับการปลูกพืชได้เกือบทุกชนิด นับตั้งแต่พืชผัก พืชสวนผลไม้ขนาดใหญ่ พืชไร่เช่น อ้อย ข้าวโพด หน่อไม้ฝรั่ง เป็นต้น นอกจากนี้ยังใช้ได้กับเนื้อดินแทบทุกประเภท ตั้งแต่เนื้อดินหยาบเช่นพวกทราย จนกระทั่งพวกเนื้อละเอียดเช่นพวกดินเหนียว และใช้ได้ในพื้นที่ทุกลักษณะ ตั้งแต่ที่ราบเรียบจนกระทั่งถึงพื้นที่บนไหล่เขาที่ชันเข่น ทางภาคเหนือของประเทศไทย โดยเฉพาะที่มีการปลูกพืชเมืองหนาว ที่ปลูกตามเนินเขา ไหล่เขา หรือพื้นที่ลาดชันมากๆปกติไม่อาจให้น้ำได้โดยวิธีอื่นได้ นอกจากการให้น้ำแบบหยด แต่อย่างไรก็ตามการจะเลือกใช้ก็จะต้องคำนึงถึงความเหมาะสมหลายๆด้าน ว่าคุณมีค่าลงทุนหรือไม่ การจะพิจารณาได้นั้นต้องทราบรายละเอียดอีกมากมาย

1.1 ความหมายของการชลประทานแบบหยด

การชลประทานแบบหยด อาจจะมีคำอธิบายได้หลายอย่าง แต่ก็มีพื้นฐานของความหมายอันเดียวกันคือ เป็นการให้น้ำแก่พืชด้วยปริมาณน้อยๆอย่างช้าๆ แต่ให้น้ำบ่อยๆครั้งตามความเหมาะสมของพืชและดิน และให้น้ำเฉพาะบริเวณรากพืชเท่านั้น

จุดมุ่งหมายสำคัญของการให้น้ำแบบนี้ คือ จะรักษาระดับความชื้นของดินบริเวณเขตรากพืชให้อยู่ในระดับที่พืชสามารถดูดความชื้นไปได้ เพื่อสร้างความเจริญเติบโตได้อย่างสมบูรณ์เพียงพอเหมาะกับเหมาะสมกับความต้องการตลอดเวลา การจะรักษาระดับความชื้นให้เหมาะสมนั้น ระบบนี้ต้องมีการควบคุมเวลา และอัตราการให้น้ำในแต่ละจุด เพื่อที่จะไม่ทำให้ดินอมน้ำหรือแห้งจนเกินไป อุปกรณ์ที่สำคัญในการควบคุม และจ่ายน้ำให้แก่ต้นพืชก็คือ หัวปล่อยน้ำ ซึ่งจะติดอยู่บนท่อแขนงที่วางไปตามแถวต้นพืช และท่อแขนงก็ไปต่อกับท่อนำน้ำมายังพื้นที่ คือ ท่อแยกประธาน และท่อประธานเป็นต้น นอกจากนี้ยังมีอุปกรณ์ต่างๆอีกมากมายที่จะได้กล่าวถึงต่อไป

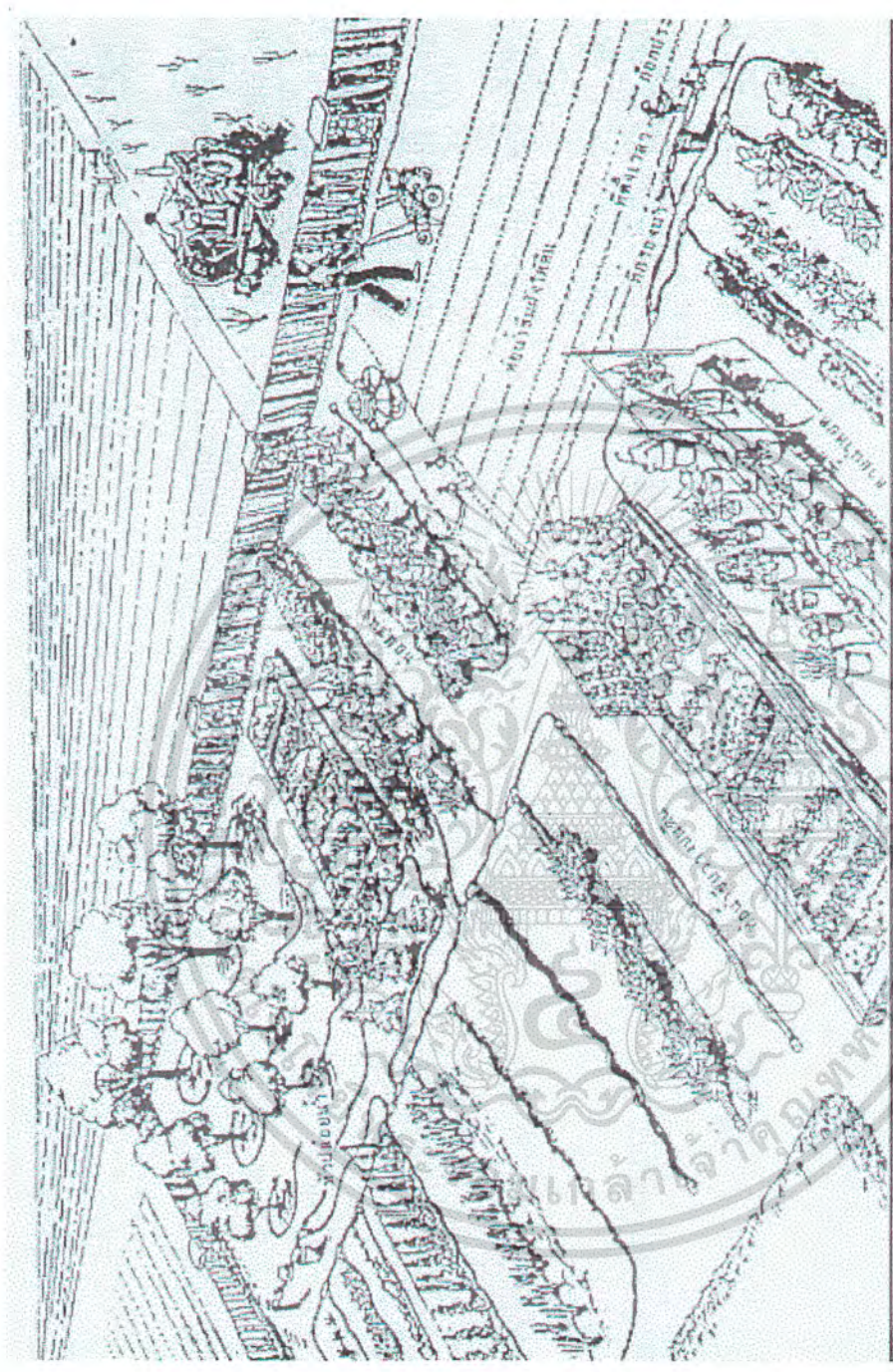
นอกจากนี้ยังมีความหมายไว้สำหรับอธิบายวิธีการให้น้ำแก่พืชโดยมีลักษณะที่สำคัญดังต่อไปนี้

1. เป็นวิธีการให้น้ำด้วยอัตราที่น้อย (น้อยกว่า 15 ลิตร/ชม. ต่อหัว)
2. เป็นวิธีการให้น้ำแต่ละครั้งใช้เวลานาน (นานมากกว่า 4 ชม. ติดต่อกัน)
3. เป็นวิธีการให้น้ำช่วงบ่อยครั้ง (ไม่เกินวันละ 3 ครั้ง)
4. เป็นวิธีการให้น้ำโดยตรงในบริเวณเขตรากพืช
5. เป็นวิธีการให้น้ำด้วยระบบท่อที่ใช้ความดันต่ำ (ความดันที่หัวปล่อยน้ำไม่เกิน 15 ปอนด์/ตารางนิ้ว)

ตารางนี้

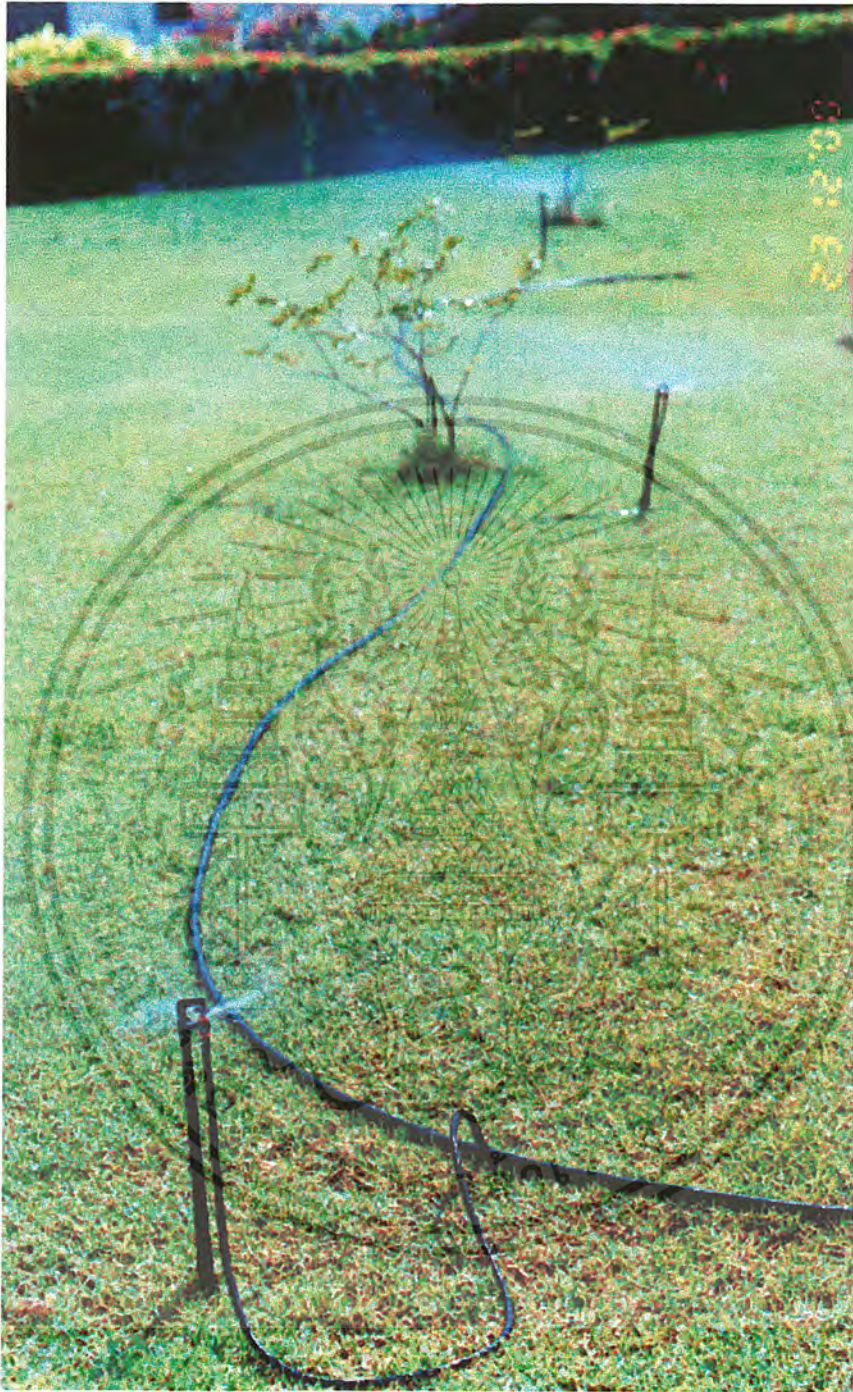
ปัจจุบันการให้น้ำแก่พืชวิธีนี้กำลังเป็นที่นิยมกันมากในหลายๆประเทศ เช่น สหรัฐอเมริกา อังกฤษ อิสราเอล นิวซีแลนด์ อิตาลี ญี่ปุ่น ออสเตรเลีย เยอรมันและฝรั่งเศส เป็นต้น สำหรับในประเทศไทยกำลังมีผู้สนใจพัฒนาการใช้กันมากขึ้นเรื่อยๆ ถึงแม้ว่าลงทุนจะยังนับว่าแพง แต่ในอนาคตอันใกล้ วิธีการให้น้ำแบบนี้ น่าจะมีความจำเป็นและคุ้มค่าการลงทุน ถ้าเกษตรกรมีความรู้ความเข้าใจในการใช้งานมากเพียงพอ และนำไปใช้กับพืชที่ให้ผลตอบแทนสูง เช่น ผลไม้และพืชเมืองหนาว เป็นต้น โดยเฉพาะน่าจะนำไปใช้ในภาคอีสาน เพราะวิธีนี้เหมาะกับดินที่ไม่ค่อยอุ้มน้ำ มีปัญหาดินเค็ม และแหล่งน้ำมีจำนวนจำกัด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 องค์ประกอบอุปกรณ์ ต่าง ๆ ของระบบชลประทานน้ำหยด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 การใช้งานระบบน้ำหยดจริงในงานนิทัศน์ ตลาดกระบี่ ปี 2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ข้อดีของการชลประทานแบบหยด

1. เพิ่มผลผลิต เนื่องจากการให้น้ำแบบหยดจะรักษาระดับความชื้นในดินอยู่ในเกณฑ์ที่พอเหมาะตลอดเวลา จะทำให้พืชงอกงามและได้ผลผลิตดีที่สุดในดินอยู่ ซึ่งผลผลิตจะสูงกว่าการให้น้ำแบบอื่นประมาณ 10-20 เปอร์เซ็นต์ โดยเฉพาะในดินทราย ดินที่มีความเค็มหรือคุณภาพของน้ำไม่ดี การให้น้ำแบบหยดจะให้ผลผลิตมากกว่าถึงสองเท่า
2. ประหยัดน้ำได้มาก เนื่องจากการให้น้ำแบบหยดเป็นการให้น้ำแก่รากพืชโดยตรง ฉะนั้นจึงถูกจำกัดให้ซึมลงไปเฉพาะในบริเวณรากพืชเท่านั้นพื้นดินระหว่างต้นพืชหรือระหว่างแถวจะไม่เปียกน้ำ จึงไม่มีการระเหยจากผิวดิน และน้ำก็ไม่สูญเสียไปเพราะวัชพืชเอาไว้ใช้
3. ใช้แรงงานน้อย ในการดำเนินงานเนื่องจากอุปกรณ์ต่างๆของระบบการให้น้ำแบบหยดนั้น ได้ติดตั้งไว้เป็นการค่อนข้างถาวร พร้อมทั้งให้น้ำได้ทุกเมื่อ ซึ่งจะเป็นผลให้มีค่าใช้จ่ายในการดำเนินงานต่ำ
4. ไม่เป็นอุปสรรคกีดขวางการดำเนินงานด้านอื่นในพื้นที่เพราะปลูก เช่น การพ่นปราบศัตรูพืช การตัดแต่งกิ่ง ตลอดจนการเก็บผลผลิต การทำงานเหล่านี้สามารถทำได้ในขณะที่ทำการให้น้ำ โดยเฉพาะสวนองุ่นและสวนไม้ผลเป็นต้น
5. ควบคุมปริมาณการให้น้ำได้ดี เพราะเป็นการให้น้ำครั้งละน้อยๆ ซึ่งสามารถควบคุมเวลาการให้น้ำได้ และปริมาณน้ำใกล้เคียงกับความต้องการได้มากกว่าวิธีการให้น้ำแบบอื่นๆ
6. บัญและยาปราบศัตรูพืชบางชนิดสามารถให้แก่พืชได้ โดยการละลายไปพร้อมกับน้ำที่ให้น้ำ ทำให้การให้บูนและยาปราบศัตรูพืชเป็นไปอย่างมีประสิทธิภาพและประหยัด
7. ควบคุมป้องกันโรคพืชและแมลงต่างๆ ที่จะทำอันตรายแก่พืชได้ เพราะการให้น้ำแบบหยดไม่ทำให้ใบของพืชเปียก ความชื้นบริเวณใบจึงต่ำ โรคพืชย่อมเกิดได้ยาก นอกจากนั้นการพ่นยาปราบศัตรูพืชต่างๆ ก็มีประสิทธิภาพสูง เนื่องจากยาซึ่งติดตามใบ กิ่ง ก้าน และลำต้นของพืชจะไม่ถูกชะล้างไป
8. การควบคุมวัชพืช เนื่องจากการให้น้ำเป็นจุดเฉพาะบริเวณโคนต้น ทำให้พื้นที่ที่เปียกน้ำเป็นส่วนน้อยของพื้นที่ทั้งหมด พื้นที่บริเวณอื่นจึงไม่มีน้ำ ทำให้การเจริญเติบโตของวัชพืชเป็นไปไม่ได้โดยยาก
9. ทำให้พืชงอกและเจริญเติบโตอย่างสม่ำเสมอ เพราะพืชที่ปลูกได้รับน้ำอย่างสม่ำเสมอและทั่วถึงกัน โดยเฉพาะในขณะที่กำลังงอกและยังเล็กอยู่ ทำให้เปอร์เซ็นต์การรอดตายมีมาก และอัตราการเจริญเติบโตสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. สามารถใช้ได้ดีกับดินที่มีคุณภาพต่ำ เนื่องจากวิธีการให้น้ำแบบหยด เป็นการให้น้ำและธาตุอาหารแก่รากพืชโดยตรง ครั่งละน้อยๆ อย่างสม่ำเสมอ ฉะนั้นแม้ในดินทรายก็สามารถปลูกพืชได้โดยการให้น้ำแบบหยดนี้

11. ลดปัญหาเรื่องการระบายน้ำ เพราะให้น้ำไม่มากเกินไปเกินความต้องการของพืช จึงไม่ก่อให้เกิดปัญหาการระบายน้ำ

12. ลดอันตรายที่พืชจะได้รับจากความเค็ม เนื่องจากการให้น้ำแบบหยดสามารถเพิ่มน้ำให้มากขึ้น เพื่อไล่เกลือออกไปเฉพาะบริเวณเขตรากพืช และความเค็มของเกลือลดลงเนื่องจากน้ำซึมลงดินเกือบตลอดเวลาและใบพืชไม่ไหม้ เนื่องจากไม่มีเกลือติดอยู่ตามใบพืช เหมือนเมื่อให้น้ำแบบฉีดฝอย

13. ไม่ทำให้หน้าดินแน่นเป็นแผ่น เพราะเป็นการให้น้ำอย่างช้าๆ ปริมาณน้อยๆ ดินจึงไม่เกิดการอัดตัว

14. สามารถติดตั้งเครื่องควบคุมน้ำชนิดอัตโนมัติ ให้ทำงานตามกำหนดเวลาแบบรอบเวรได้เองทั้งระบบ

15. ไม่มีปัญหาเรื่องลม การให้น้ำแบบอื่นๆ เมื่อมีลมแรงๆ จะทำให้การกระจายของน้ำไม่สม่ำเสมอ โดยเฉพาะการให้น้ำแบบฉีดฝอยขนาดใหญ่

1.3 ข้อเสียและปัญหาของการชลประทานแบบหยด

ถึงแม้การให้น้ำแบบหยดจะมีข้อดีมากมายหลายอย่าง แต่ก็มีข้อเสียและปัญหาบางประการที่ต้องนำมาพิจารณาประกอบการตัดสินใจ ในการเลือกระบบเทียบกับระบบอื่นๆ พอสรุปได้ดังนี้

1. การอุดตันที่หัวปล่อยน้ำ นับว่าเป็นปัญหาสำคัญที่สุดที่ทำให้ระบบการให้น้ำแบบหยดต้องล้มเหลว ถึงแม้ว่าการกรองน้ำจะเป็นวิธีที่ดีที่สุดในการลดปัญหาการอุดตัน แต่บางกรณีใช้วิธีการกรองอย่างเดียวไม่เพียงพอ จะต้องมีการใช้น้ำยาเคมีเข้ามาช่วย เนื่องจากการอุดตันอาจเกิดขึ้นได้จากหลายสาเหตุ เช่น ตะกอน ทราย โคลนตมพอกรองได้ แต่สำหรับการตกตะกอนของสารเคมีที่ละลายอยู่ในแหล่งน้ำ เช่น แคลเซียมและแมกนีเซียมเหล็ก หรือเกิดจากการเจริญเติบโตของสิ่งมีชีวิตในท่อ หรือหัวปล่อยน้ำ เช่น ตะไคร้ แบคทีเรีย กำมะถันหรือเหล็กต้องใช้น้ำยาเคมีเข้าช่วย เป็นต้น

2. ต้องมีการบำรุงรักษาสูง มีการตรวจสอบระบบการทำงานต่างๆ อย่างสม่ำเสมอ

3. มีการบำรุงรักษาสูง มีการตรวจสอบระบบการทำงานต่างๆ อย่างสม่ำเสมอ

4. อาจจำกัดการเจริญเติบโตของรากพืช ในกรณีที่ใช้ระบบการให้น้ำแบบหยดเป็นหลัก และพื้นที่ซึ่งมีฝนตกน้อย รากพืชจะเจริญเติบโตเฉพาะบริเวณที่เปียกน้ำเท่านั้น ถ้าบริเวณที่เปียกน้ำ มีบริเวณเล็กเกินไปหรือเปียกเฉพาะแถบเดียว และการแผ่ขยายของรากไม่เพียงพอ เวลาลมพัดแรงๆ พืชอาจจะโยกคลอนได้ โดยเฉพาะพืชที่ปลูกในดินเหนียว น้ำซึมด้านลึกได้น้อย อาจทำให้รากพืช หยั่งลงไม่ลึกเช่นกัน ดังนั้นการออกแบบติดตั้งหัวปล่อยน้ำจึงเป็นเรื่องสำคัญมากเกี่ยวกับการ กระจายความชื้น

5 ระบบท่อที่วางบนดินอาจจะได้รับความเสียหาย จากการทำงานของคนงานปรบวัชพืช หรือจากสัตว์ต่างๆ เช่น สุนัข หนูหรือกระรอก มากัดแทะเป็นต้น มดหรือแมลงสาปอาจเข้าไปในรู ของหัวปล่อยน้ำขณะที่หยุดส่งน้ำ เป็นต้น

6. บางทีอาจจะเกิดความเสียหายจากการสะสมของเกลือ บริเวณเขตรากถ้าพื้นที่นั้นมีเกลือ อยู่มากและให้ปริมาณน้ำไม่เพียงพอที่จะผลักดันให้บริเวณที่มีเกลือสะสมเข้มข้นออกพ้นจากเขต ราก ซึ่งความเข้มข้นของเกลือจะอยู่บริเวณขอบเปียกของน้ำที่ให้แบบหยดนั่นเอง

7. อย่างน้อยต้องมีระบบกรองน้ำที่เชื่อถือได้ เพราะรูหัวปล่อยน้ำมีขนาดเล็กมาก ง่ายต่อการอุดตัน ถึงแม้ว่าน้ำจะดูว่าใสสะอาดก็จำเป็นต้องมีเครื่องกรองเสมอ

8. ระบบรากเคยชินกับการได้รับน้ำอย่างสม่ำเสมอ ถ้าน้ำที่เคยให้ด้วยระบบนี้เกิดขัดข้อง พืชจะอยู่ในสภาพที่แยกว่าต้นพืชที่ไม่ได้น้ำด้วยระบบน้ำหยด ทั้งนี้เพราะพืชที่ให้น้ำด้วยวิธีอื่น จะมีรากที่แตกต่างกันและมีโครงสร้างของเซลล์ที่เหนียวแน่นกว่า ฉะนั้นการให้น้ำแบบหยด แม้จะ ประหยัดน้ำกว่าการให้น้ำด้วยวิธีอื่นก็ตาม แต่จำเป็นต้องออกแบบให้ไว้ใจได้จริงๆและดูแลให้อยู่ ในสภาพที่สามารถใช้งานได้ตลอดเวลา

9. ค่าลงทุนครั้งแรกค่อนข้างสูง เนื่องจากระบบต้องใช้ท่อแขนง ท่อแยกประธาน และท่อ ประธานเป็นจำนวนมาก และก็ต้องใช้หัวปล่อยเป็นจำนวนมากด้วย เฉพาะค่าหัวปล่อยน้ำชนิดที่พอ เชื่อถือได้ คิดเป็นเงินประมาณ 25 ถึง 35 เปอร์เซ็นต์ของเงินลงทุนทั้งระบบและต้องมีเครื่องกรองน้ำ และอุปกรณ์อื่นๆอีก ค่าลงทุนเฉลี่ยสำหรับพืชสวนตกไร่ละ 4000 - 8000บาท และสำหรับพืชไร่ หรือพืชผัก ตกไร่ละ 6000-10000บาท ฉะนั้นระบบนี้จึงเหมาะที่จะใช้กับพืชที่ให้ผลตอบแทนสูง น้ำ ที่ใช้จัดหามาได้ด้วยราคาแพง พื้นที่ลาดชันหรือสูงๆต่ำๆเป็นคลื่น แรงงานหายากและมีราคาแพง ไม่เหมาะที่จะให้น้ำด้วยวิธีอื่น

10. ความรู้สึกของผู้ใช้เอง ถ้าเจ้าของพื้นที่หรือผู้ดูแลมีความรู้สึกไม่ชอบที่จะใช้ระบบนี้แต่ แรกหรือคิดว่าคงไม่ได้ผล ก็ไม่ควรนำมาใช้ เพราะโอกาสเสียหายจะง่ายและเร็วกว่าของผู้ที่จะ พยายามอยากจะใช้ระบบนี้ให้ได้ผลจริงๆ โดยมีการดูแลเอาใจใส่อย่างทั่วถึง ปรับปรุงแก้ไขตลอด เวลา ไม่ใช่รอจนเสียหายมากๆแล้วค่อยแก้ไขมักจะไม่ค่อยทันการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ระบบการชลประทานแบบหยด

การชลประทานแบบหยด เป็นการให้น้ำแก่พืชที่จำเป็นต้องมีอุปกรณ์พิเศษหลายอย่าง เพื่อควบคุมการให้น้ำแก่พืช ให้มีประสิทธิภาพสูงที่สุดเมื่อเปรียบเทียบกับการให้น้ำแก่พืชด้วยวิธีแบบอื่นๆ ฉะนั้นจึงมีความจำเป็นในเรื่องดินที่จะต้องทำความเข้าใจถึงหน้าที่และความสำคัญของอุปกรณ์ต่างๆ ที่จำเป็นต้องใช้ในระบบ เพื่อจะนำไปใช้งานให้สอดคล้องกับสภาพภูมิประเทศและพื้นที่เพาะปลูก ตลอดจนเงินทุนและผลผลิตที่จะได้รับ ถ้าใช้ไม่ถูกต้องตามที่ควรจะเป็น นอกจากจะเสียเงินลงทุนเพิ่มขึ้น ยังอาจจะใช้ไม่ได้ผลเท่าที่ควรอีกด้วย

2.1 องค์ประกอบของระบบชลประทานแบบหยด (Drip System Component)

ระบบชลประทานแบบหยด ส่วนใหญ่จะต้องประกอบด้วยอุปกรณ์ที่สำคัญอย่างน้อยดังต่อไปนี้ หัวปล่อยน้ำ(Emitter) ท่อแขนง(Lateral Line) ท่อประธานย่อย(Sub-Main) ท่อประธาน(Main Line) ประตูน้ำ(Valve) เครื่องวัดความดัน(Pressure Gauge) ส่วนเครื่องปั้มน้ำบางทีอาจไม่จำเป็นต้องใช้ถ้าแหล่งน้ำมีความดันเพียงพอ เช่น ระบบที่ใช้น้ำประปา ที่มีความดันมากกว่า 10 ปอนด์ต่อตารางนิ้วหรือแหล่งน้ำจากที่สูงเกินกว่า 6 เมตร จากพื้นดินเป็นต้น และอุปกรณ์ที่นับว่าสำคัญมากของระบบนี้ คือ เครื่องกรองน้ำ(Filter) ดังรูปที่ 2.1 องค์ประกอบของระบบชลประทานแบบหยดอย่างง่าย ที่ใช้กันในบริเวณบ้นี้ ซึ่งต่อระบบจากแหล่งน้ำประปา โดยไม่ต้องอาศัยปั้มน้ำ ระบบนี้ใช้ได้กับพืชเกือบทุกชนิด ตั้งแต่พืชสวนครัว ไม้ประดับ ไม้ยืนต้น ตลอดจนกระทั่งสนามหญ้าและสำหรับในพื้นที่เพาะปลูกขนาดใหญ่ นอกจากมีอุปกรณ์ดังกล่าวมาแล้ว บางครั้งก็อาจจำเป็นต้องมีอุปกรณ์พิเศษเพิ่มขึ้นอีกตามความเหมาะสม โดยเฉพาะอุปกรณ์ที่ถือว่าเป็นเครื่องควบคุมการจ่ายน้ำต้นทาง(Control Head) อันได้แก่ เครื่องวัดปริมาณการไหลของน้ำ(Meter Water) เครื่องฉีดผสมปุ๋ยหรือสารเคมี(Fertilizer Tank or Chemical Injector) เครื่องควบคุมความดัน(Presure Regulator) ประตูป้องกันน้ำไหลกลับ(Non-Return Valve) ประตูระบายอากาศ(Air-Release Valve) ทางฉีดล้างระบายน้ำออก(Outlet For Flushing) และนอกจากนี้ยังสามารถติดตั้งเป็นระบบควบคุมการจ่ายน้ำอัตโนมัติได้อีกด้วย โดยติดตั้งประตูน้ำไฟฟ้า(Solenoid Valve)และระบบการควบคุมด้วยคอมพิวเตอร์(Computer Control Unit) เป็นต้น

ฉะนั้นเพื่อให้เข้าใจดีขึ้นจะขออธิบายหน้าที่ของอุปกรณ์ต่างๆพอสังเขปดังต่อไปนี้

1. หัวปล่อยน้ำ จะทำหน้าที่เป็นตัวควบคุม ปริมาณการไหลของน้ำจากท่อแขนงไปสู่ดิน ปกติหัวปล่อยน้ำนี้จะติดตั้งอยู่บนท่อแขนงที่วางไว้บนผิวดิน หรือมีท่อเล็กๆสีดำต่อแยกออกไปจากท่อแขนงอีกที โดยมากที่นิยมใช้มีอัตราการไหลประมาณ 1-10 ลิตร/ชั่วโมง และมีขนาดของรูจ่ายน้ำ 0.5-1.5 มม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท่อแขนงอีกที โดยมากที่นิยมใช้มีอัตราการไหลประมาณ 1-10 ลิตร/ชั่วโมง และมีขนาดของรูจ่ายน้ำ 0.5-1.5 มม.

2. ท่อแขนง เป็นท่อซึ่งต่อแยกออกจากท่อประธานย่อย หรือบางครั้งก็ต่อออกจากท่อประธานโดยตรงและเป็นท่อซึ่งติดตั้งหัวปล่องน้ำ วางชิดขนานไปกับแถวของต้นพืช อาจใช้ท่อแขนง 1 แถว สำหรับพืช 1-2 แถว หรือท่อแขนง 1-2 แถว สำหรับพืช 1 แถวก็ได้แล้วแต่ความเหมาะสม สำหรับขนาดของท่อ จะถูกกำหนดโดยจำนวนต้นพืชต่อแถวและจำนวนของหัวปล่องน้ำที่ไหลต่อต้น หรือโดยจำนวนของหัวและความยาวของท่อที่ใช้ โดยทั่วไปมีขนาดเส้นผ่านศูนย์กลางตั้งแต่ 6-25 มม. ที่ใช้กันมาก 12-20 มม. ความยาวของท่อไม่ควรเกิน 300 ม.

3. ท่อประธานย่อย เป็นท่อที่ต่อแยกจากท่อประธาน เพื่อแบ่งการควบคุมออกเป็นส่วนๆ ท่อนี้ถ้าระบบไม่ใหญ่มากนักก็ไม่จำเป็นต้องมี คือ มีเฉพาะท่อแขนงต่อโดยตรงกับท่อประธานก็ได้ โดยทั่วไปมีขนาดเส้นผ่านศูนย์กลางประมาณ 20-80 มม. ที่ใช้กันมากคือขนาด 30-50 มม. ท่อประธานย่อยวางทั้งชนิดบนดินและฝังใต้ดิน

4. ท่อประธาน เป็นท่อที่เชื่อมโยงท่อประธานย่อย หรือท่อแขนงในแต่ละสายให้ต่อไปยังแหล่งน้ำ และจากท่อประธานนี้ ท่อประธานย่อย หรือท่อแขนง อาจจะแยกออกไปด้านเดียวหรือทั้งสองข้างก็ได้ ปกติท่อประธานนิยมฝังไว้ใต้ผิวดิน ดังรูปที่ 2.3 แสดงให้เห็นท่อประธาน(Main Line) ท่อประธานย่อย(Sub Main) และท่อแขนงชนิดฝังใต้ดินและวางบนดิน

5. เครื่องกรองน้ำ เป็นอุปกรณ์ที่สำคัญ เพื่อขจัดปัญหาอุดตันซึ่งมักจะเกิดขึ้นเสมอ ตามปกติจะเป็นแบบตะแกรง(Screen Filter) และแบบกรวดทราย(Gravel-Sand Filter)

6. เครื่องควบคุมการจ่ายน้ำต้นทาง เป็นอุปกรณ์ต่างๆ ที่ติดตั้งไว้ใกล้กับแหล่งน้ำตรงจุดเริ่มต้นของระบบ เพื่อควบคุมความดัน ปริมาณน้ำและคุณภาพน้ำ เป็นต้น อาจจะประกอบด้วย

6.1 เครื่องวัดปริมาณการไหลของน้ำ เพื่อให้ทราบอัตราการใช้น้ำว่าเป็นไปตามที่กำหนด

6.2 เครื่องวัดความดัน เพื่อให้ทราบความดันขณะใช้งานเพราะจะสัมพันธ์กับปริมาณน้ำที่จ่ายออกจากหัวปล่องน้ำ

6.3 เครื่องควบคุมความดัน เพื่อให้ความดันไม่เปลี่ยนแปลงมากนัก อาจจะติดตั้งที่ต้นทางของท่อประธานย่อยหรือท่อแขนงก็ได้ จะช่วยรักษาความดันของน้ำในแต่ละท่อแขนงให้เท่ากันมากที่สุด จะทำให้ปริมาณน้ำที่ไหลออกจากหัวปล่องน้ำใกล้เคียงกันประตูป้องกันน้ำไหลกลับ เมื่อเวลาหยุดการให้น้ำอาจจะมิให้น้ำไหลกลับทางไปยังเครื่องสูบน้ำ ถ้าน้ำผสมปุ๋ยก็อาจเกิดปัญหาได้ ละ อาจทำให้หัวปล่องน้ำที่วางบนผิวดินดูดเอาตะกอนดินเข้าไปในหัวปล่องน้ำได้

6.4 ประตูป้องอากาศ ประตูนี้จะติดตั้ง ณ ตำแหน่งที่สูงสุด เพื่อใช้ระบายอากาศที่สะสมอยู่ในระบบท่อ ซึ่งจะทำให้น้ำไหลไม่เต็มท่อ เป็นการสูญเสียความดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ชนิดของระบบชลประทานแบบหยด (Types of Drip Irrigation System)

ระบบชลประทานแบบหยดโดยทั่วไปพอจะแบ่งได้เป็น 4 แบบดังนี้คือ

1. ระบบให้น้ำทางดิน (Surface Systems) เป็นระบบที่ท่อแขนงอยู่บนผิวดิน ซึ่งเป็นชนิดที่นิยมใช้กัน โดยทั่วไป และจะใช้กับพืชที่ปลูกระยะห่างๆ กัน
2. ระบบให้น้ำใต้ผิวดิน (Subsurface Systems)
3. ระบบให้น้ำแบบหยดที่เคลื่อนที่ได้ (Mechanical-move Drip Systems) ระบบนี้สามารถเคลื่อนย้ายที่สำหรับการให้น้ำแบบหยดกับพืชที่ปลูกเป็นแถว
4. ระบบให้น้ำแบบหยดที่จ่ายเป็นจังหวะ (Pulse Systems) ระบบนี้มีกรรมนำมาพิจารณาใช้กับหัวปล่อยน้ำที่มีอัตราการจ่ายน้ำสูง แต่มีการให้น้ำเป็นจังหวะเวลา

2.3 คุณลักษณะและชนิดของหัวปล่อยน้ำ (Types of Drip Irrigation Systems)

หัวปล่อยน้ำเป็นอุปกรณ์ที่เปรียบเสมือนหัวใจของระบบชลประทานแบบหยด ซึ่งสามารถนำไปติดตั้งกับท่อแขนงได้เป็น 3 ลักษณะดังต่อไปนี้

1. หัวปล่อยน้ำที่ติดตั้งบนท่อแขนง (One-line Emitter) คือวิธีการเจาะผนังท่อและเสียบหัวปล่อยน้ำ ติดตั้งไว้บนท่อด้านบน เป็นลักษณะที่นิยมใช้กันทั่วไปดังรูปที่ 2.4 ก.
2. หัวปล่อยน้ำที่ติดตั้งเป็นส่วนเดียวกับท่อแขนง (In-line Emitter) หรืออยู่ภายในท่อแขนง แบบนี้สะดวกในการติดตั้งและเก็บรักษา เพราะไม่มีส่วนยื่นออกมาเกาะก่อกวน นอกจากนี้ยังสามารถส่งติตหัวมาจากโรงงานตามระยะที่ต้องการได้ และนำมาวางในพื้นที่เพาะปลูกได้เลย ดังรูปที่ 2.4 ข.
3. หัวปล่อยน้ำที่ติดตั้งต่อยื่นออกมาจากท่อแขนง (Riser) แบบนี้นั้นว่าประหยัดและกระจายน้ำได้ดีกว่าแบบอื่นๆ ดังรูปที่ 2.4 ค. เพราะสามารถกระจายจุดการให้น้ำ ออกไปได้กว้างโดยรอบต้นจากท่อแขนงสายเดียว ปกติหัวปล่อยน้ำจะทำหน้าที่เป็นตัวลดความดันของน้ำจากท่อแขนง เพื่อให้อัตราการไหลที่ปล่อยออกมาน้อยตามความต้องการ

2.3.1 คุณลักษณะพื้นฐานของหัวปล่อยน้ำ

1. มีอัตราการไหลคงที่และสม่ำเสมอ โดยทั่วไปอัตราการไหลของน้ำอยู่ระหว่าง 1-10 ลิตร/ชม. ด้วยความดันที่ใช้ประมาณ 5-10 เมตร (เป็นเฮคของน้ำ) เป็นอย่างน้อย ทำให้รู้ที่ให้น้ำออกมาต้องเล็ก ผู้ผลิตหัวปล่อยน้ำต้องมีความประณีต ที่จะต้องทำให้ขนาดของรูสม่ำเสมอมากที่สุด เพราะจะมีผลทำให้ปริมาณการไหลเปลี่ยนแปลงไปมาก ถ้าขนาดของรูแตกต่างกัน

2. มีขนาดของรูใหญ่พอที่จะลดปัญหาการอุดตัน ปกติการลดความดันที่หัวปล่อยน้ำ เพื่อให้ได้อัตราการไหลที่น้อย ขนาดของรูควรจะอยู่ระหว่าง 0.3-1.0 มม. แต่รูเล็กเช่นนี้จะทำให้อุดตัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง่าย แต่ถ้าทำให้ขนาดของรูใหญ่ ก็ลดความดันได้น้อย น้ำก็ออกมาเยอะ ใ้ตามจุดประสงค์ที่ต้องการคือ อัตราการไหลที่ต่ำ โดยมีความดันที่สูงและขนาดของรูให้ใหญ่

3. มีราคาไม่แพงนักและสะดวกต่อการใช้งาน ระบบการให้น้ำแบบหยดโดยทั่วไปค่าลงทุนสำหรับหัวปล่อยน้ำคิดเป็นเงินประมาณ 20-30 เปอร์เซ็นต์ ของเงินที่ใช้ลงทุนทั้งระบบ

2.3.2 ชนิดของหัวปล่อยน้ำ

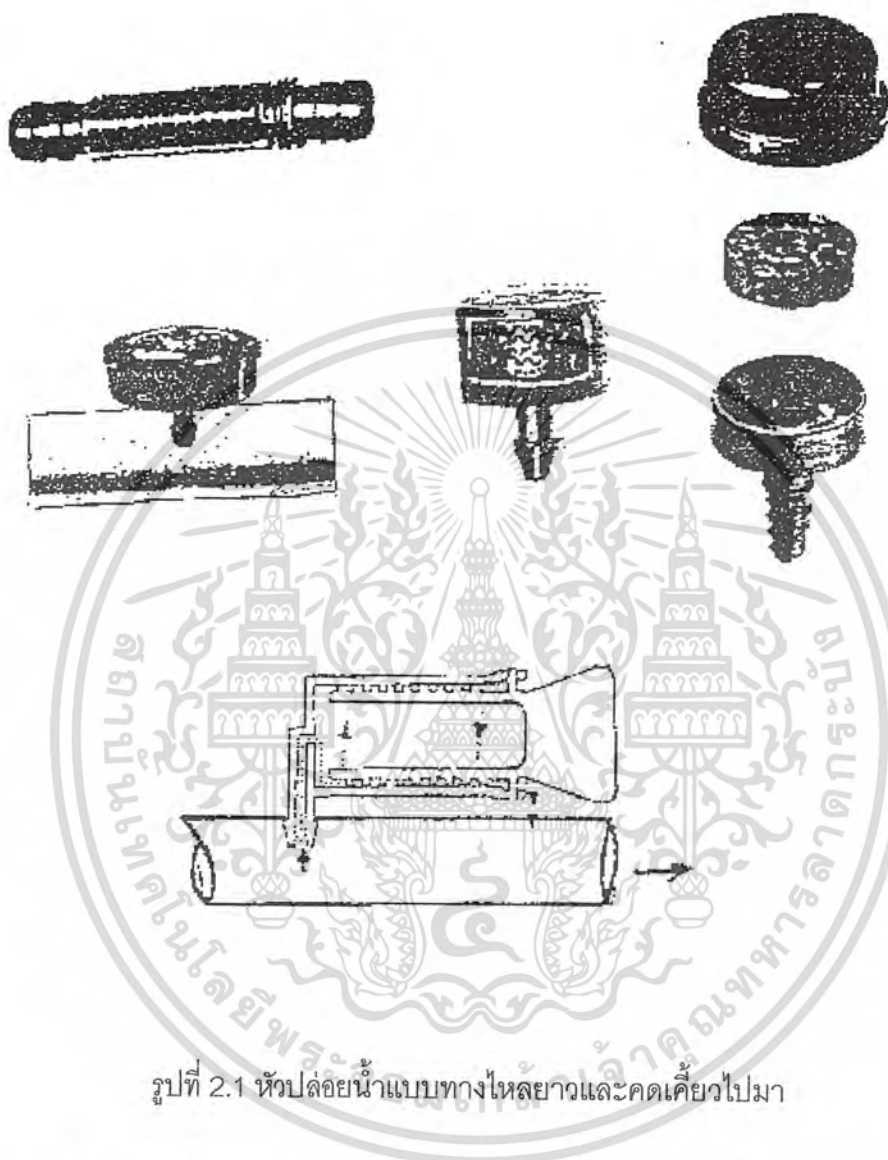
เพื่อความสะดวกและง่ายต่อการเข้าใจ ขอแบ่งชนิดของหัวปล่อยน้ำออกเป็น 3 ประเภทใหญ่ดังต่อไปนี้ คือ

1. ประเภทที่สามารถปรับปริมาณการไหลของน้ำได้ที่หัวโดยตรง ส่วนมากหัวประเภทนี้จะมีลักษณะการปรับด้วยการขันเกลียวเข้าออก เพื่อเปิด-ปิดรูให้ใหญ่หรือเล็กตามต้องการ ดังแสดงในรูปที่ 2.5 อัตราการไหลส่วนใหญ่สามารถปรับได้ตั้งแต่ 0-20 ลิตร/ชั่วโมง โดยความดันของน้ำเพียง 2-5 เมตร

2. ประเภทไม่สามารถปรับอัตราการไหลได้ที่หัว เป็นแบบที่จะกำหนดอัตราการไหลไว้ก่อนข้างที่จะแน่นอนด้วยความดันที่กำหนดต่างๆ โดยอาศัยการออกแบบ

วิธีลดความดันที่หัวปล่อยน้ำด้วยวิธีการต่างๆกัน ซึ่งสามารถแยกได้เป็น 4 ลักษณะที่แตกต่างกัน

(ก) หัวปล่อยน้ำแบบมีทางไหลของน้ำยาวคดเคี้ยวไปมา ความดันของน้ำจะลดลงไปเรื่อยๆเมื่อน้ำไหลผ่านช่องแคบๆ คดเคี้ยวด้วยระยะทางยาวๆ เพื่อให้ได้อัตราการไหลที่กำหนด เช่น 4-8 ลิตร/ชม. ดังตัวอย่างแสดงในรูปที่ 2.1 มีทั้งชนิดที่ติดตั้งบนท่อแขนง หัวติดอยู่ภายในท่อแขนง และหัวยื่นออกนอกท่อแขนง



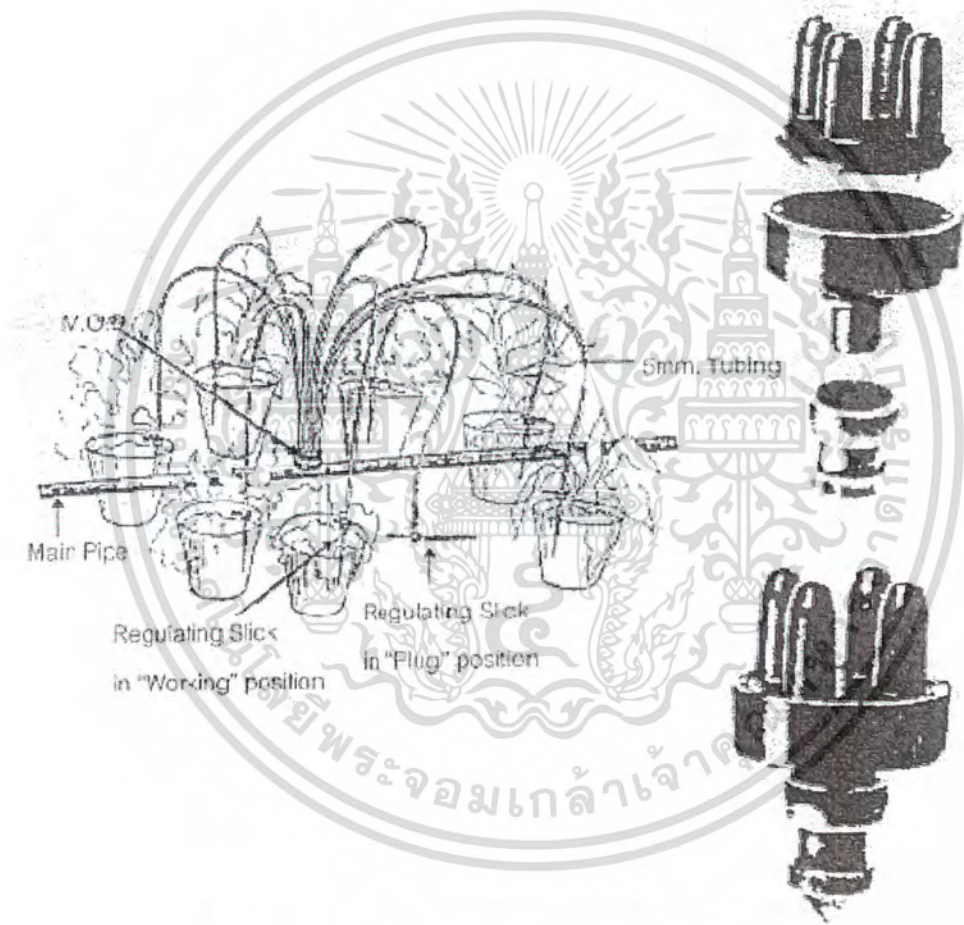
รูปที่ 2.1 หัวปลี่ยน้ำแบบทางไหลยาวและกดเคี้ยวไปมา

(ข) หัวปลี่ยน้ำแบบรูเล็กๆ ให้น้ำไหลเข้าทางด้านเข้า และเป็นแบบให้น้ำไหลวน ซึ่งเป็นการช่วยลดความดันน้ำลง ได้อย่างมากดังแสดงในรูป นอกจากนี้อาจจะมีการต่อท่อขนาดเล็ก จากหัวปลี่ยน้ำแบบรู เพื่อกระจายน้ำไปหลายจุด ก็ช่วยลดความดันลงไปได้อีก หัวปลี่ยน้ำประเภทนี้มักจะกำหนดอัตราการไหลไว้แน่นอนโดยมีความดันที่กำหนดให้ เช่น ที่ นิยมใช้กันมากจะมีอัตรา 2-4 ลิตร / ชั่วโมง โดยใช้ความดัน 15 ปอนด์ / ตารางนิ้ว เป็นต้น

(ค) หัวปลี่ยน้ำแบบปรับขีดเซกความดันได้เอง หัวชนิดนี้จะกำหนดให้มีอัตราการไหลที่คงที่ หรือเปลี่ยนแปลงก็ไม่มากนัก หัวชนิดนี้ถือชื่อว่าเป็นแบบที่มีประสิทธิภาพสูง โดยที่หัวปลี่ยน้ำชนิดนี้ภายใต้หัวจะมีแผ่นยางที่ยืดหยุ่นได้และมีรูให้น้ำผ่าน เมื่อที่หัวปลี่ยน้ำมีความดันเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูง แผ่นยางนี้จะโค้งตัวขึ้น ทำให้ขนาดของรูให้น้ำออกเล็กลง โดยมีความสัมพันธ์กับอัตราการไหลของน้ำที่กำหนด ดังในรูปแสดงตัวอย่างหัวปล่อยน้ำชนิดนี้ และแสดงความสัมพันธ์กับปริมาณน้ำที่เปลี่ยนแปลงกับอัตราการไหลของน้ำที่หัว

(ง) หัวปล่อยน้ำแบบท่อขนาดจิ๋ว (Microtube) เป็นแบบที่ถือว่าผลิตขึ้นมาใช้งานก่อนแบบอื่น เป็นหัวปล่อยน้ำที่จัดว่ามีราคาถูกและทำได้ง่าย



รูปที่ 2.2 หัวปล่อยน้ำแบบน้ำไหลเข้าด้านข้างแบบน้ำวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 หัวปล่อยน้ำแบบต้อออกหลายจุด

3 ประเภทน้ำซึมหรือไหลออกจากผนังท่อแขนง ไม่ต้องมีหัวปล่อยน้ำบังคับความดันของน้ำจะลดลงในขณะที่น้ำซึม หรือหยดออกมาตามรูที่เจาะเล็กๆ หรือรูพรุนตามผนังซึ่งทำจากวัสดุพิเศษที่ให้น้ำซึมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบระบบน้ำหยดเพื่อให้น้ำแก่พืช

ชาวสวนผู้หนึ่งต้องการทำสวนส้มที่อำเภอคลอง จังหวัดแพร่ โดยจะปลูกส้มจำนวน 200 ต้น ซึ่งชาวสวนผู้นี้มี

พื้นที่เพาะปลูก ประมาณ 3 ไร่คือกว้างประมาณ 50 เมตร และยาวประมาณ 100 เมตร

พืชที่ปลูก ส้ม จำนวน 200 ต้น

ระยะปลูก ระยะห่างระหว่างต้น 5 เมตร และระยะห่างระหว่างแถว 5 เมตร(5x5เมตร)

ลักษณะของดิน ดินร่วนปนทราย (อัตราการซึมซาบน้ำ 10-20 มม./ชม)

แหล่งน้ำ บ่อขุดมีน้ำสูบมาเติมได้ตลอดเวลา จำนวนชั่วโมงการให้น้ำไม่เกิน 8 ชั่วโมงต่อวัน หัวปล่อยน้ำเป็นแบบสปริงเกอร์



รูปที่ 3.1.1 ขนาดของพื้นที่เพื่อใช้ในการวางแผนปลูกส้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ค่าสัมประสิทธิ์การให้น้ำของพืช(Kc)ชนิดต่าง ๆ ของการพัฒนาการของไม้ผล

ช่วงพัฒนาการของพืช	ชนิดของไม้ผล				
	ทุเรียน	มังคุด	เงาะ	ส้ม	ไม้ผลอื่นๆ
1 การพัฒนาการทางด้านกิ่งก้านสาขา	0.75	0.75	0.75	0.75	0.75
2 การชั่งน้ำหนักยอดผล *	0	0	0.00 / 0.70*	0.00 / 75*	0.00 / 75*
3 การพัฒนาการของดอก	0.85	0.8	0.8	0.8	0.8
4 การติดผล	0.8	0.85	0.9	0.9	0.9
5 การพัฒนาการของผลอ่อน	0.95	0.95	0.95	0.95	0.95
6 การเจริญเติบโตของผล	1	1	1	1	1
7 การเริ่มสุกแก่	0.9	0.9	0.85	0.85	0.85
เฉลี่ย	0.76	0.75	0.75	0.75	0.75

หมายเหตุ * ในกรณีให้น้ำตลอดจนไม้ผลชนิดอื่น เช่น เงาะ และส้ม ในช่วงของการติดผลให้ออกดอก
 ล่วงหน้านั้นช่วงสั้นระยะเวลาหนึ่ง จากนั้นจึงเริ่มให้น้ำอย่างปกติและเพื่อการกระตุ้นการออกดอก

ตารางที่ 3.2 ปริมาณการระเหยน้ำจากถาดวัดการระเหยน้ำแบบเอ(Epan)
 เฉลี่ยรายเดือนสำหรับจังหวัดต่างๆ

(หน่วย: มม./ชม.)

จังหวัด	เดือน											
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ล.	พ.ย.	ธ.ค.
1 นครศรีธรรมราช	3.3	4.3	5.3	7.3	6.2	4.3	4.8	3.9	4.0	3.9	3.1	2.7
2 เชียงราย	3.1	4.7	5.5	5.5	4.7	3.7	3.1	2.7	3.0	3.1	2.9	2.7
3 พะเยา	3.5	4.4	5.7	6.7	5.5	4.6	4.1	4.1	3.7	3.3	2.9	2.8
4 เชียงใหม่	3.4	4.6	5.5	6.5	5.6	4.8	4.2	4.0	4.2	4.1	3.5	3.0
5 ลำปาง	2.9	4.0	5.1	6.2	5.9	4.5	4.1	3.9	3.7	3.2	2.9	2.9
6 ลำพูน	3.4	5.3	6.7	7.7	6.9	5.9	4.9	4.7	4.0	3.5	3.1	3.0
7 นพว	3.7	4.7	6.3	7.3	6.3	5.2	4.0	4.8	4.6	4.2	3.8	3.4
8 นนท	2.6	3.2	3.8	4.7	4.4	3.7	3.3	3.1	3.3	3.3	2.9	2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 อัตราการซึมของน้ำผ่านดินชนิดต่างๆ

ชนิดดิน	อัตราการซึมของน้ำ (มม./ชม.)
1. ดินทราย	สูงกว่า 20
2. ดินร่วนปนทราย	10 - 20
3. ดินร่วน	5 - 10
4. ดินเหนียว	1 - 5

วิธีการคำนวณและขั้นตอนการคำนวณ

ขั้นตอนที่ 1 หาปริมาณการใช้น้ำของส้ม

จากสูตรโดยพิจารณาจากตารางที่ 3.2 จังหวัดแพร่ มีอัตราการระเหยของน้ำสูงสุดในเดือนเมษายน วันละ 7.3 มม. คำนวณหาปริมาณการใช้น้ำของส้มจากสูตร

$$\begin{aligned}
 \text{เมื่อ } E_{Tc} &= K_p \times E_{pan} \times K_c \\
 E_{pan} &= 7.3 \text{ มม./วัน} \\
 K_p &= 0.85 \\
 K_c &= 0.75 \text{ (จากตารางที่ 3.1)} \\
 \therefore E_{Tc} &= 0.85 \times 7.3 \times 0.75 \\
 &= 4.65 \text{ มม./วัน}
 \end{aligned}
 \tag{1}$$

ดินร่วนปนทรายมีความชื้นที่พืชสามารถนำไปใช้ได้ 1.2 มม./ชม. ดินจากข้อมูลที่รากส้มลึก 120 ซม. คิดเฉพาะเขตรากส่วนบนสุด 25% ที่มีน้ำส่วนที่ส้มสามารถนำไปใช้ได้มากที่สุดที่ความลึก 1/4 ของความลึกรากส้ม มีปริมาณความชื้นที่ส้มสามารถนำไปใช้ได้

$$\begin{aligned}
 &= 120 \times 1/4 \times 1.2 \\
 &= 36 \text{ มม.}
 \end{aligned}$$

แต่ยอมให้ส้มใช้น้ำได้เพียง 40 % ของความชื้นที่ส้มสามารถใช้ได้ (ก่อนเริ่มให้น้ำ)

\therefore จะเป็นปริมาณความชื้น

$$\begin{aligned}
 &= 36 \times 40/100 \\
 &= 14.4 \text{ มม. เท่านั้น}
 \end{aligned}
 \tag{2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 คำนวณหารอบเวรของการให้น้ำ

$$\begin{aligned}
 &= \text{ปริมาณความชื้นที่ยอมให้พืชใช้ได้ (2)} \div \text{ปริมาณการใช้น้ำของพืช (1)} \\
 &= 14.4 \div 4.65 \\
 &= 3 \text{ วัน}
 \end{aligned}$$

$$\therefore \text{ใน 1 รอบเวรพืชใช้น้ำ (จากดิน) ไป} = 4.65 \times 3 = 13.95 \text{ มม.}$$

ขั้นตอนที่ 3 เลือกระบบให้น้ำที่จะใช้ และกำหนดประสิทธิภาพของระบบ

เลือกระบบให้น้ำที่จะใช้และกำหนดประสิทธิภาพของระบบ(เลือกเพื่อไว้เมื่อต้น โตเต็มที่) เช่น
เลือกระบบมินิสปริงเกอร์ กำหนดประสิทธิภาพของระบบ 75 %

$$\begin{aligned}
 \text{ดังนั้นในการให้น้ำคืนให้กับดินจะต้องให้น้ำ} &= 13.95 \times 100/75 \\
 &= 18.6 \text{ มม.}
 \end{aligned}$$

กำหนดให้พื้นที่วงเปียก จากมินิสปริงเกอร์ครอบคลุมเพียง 80 เปอร์เซ็นต์ ของเขตรากส้ม
พื้นที่เขตรากเท่ากับ $3.14 \times r^2$

$$\begin{aligned}
 \text{เมื่อ } r \text{ คือ รัศมีของทรงพุ่ม} &= 5/2 \\
 \text{เพราะฉะนั้น 80 เปอร์เซ็นต์ของราก} &= 22/7 \times 2.5 \times 2.5 \times 80/100 \\
 &= 15.71 \text{ ตรม.}
 \end{aligned}$$

$$\begin{aligned}
 \text{คิดเป็นปริมาณน้ำที่ต้องให้} &= 18.6 \text{ มม.} \times 15.71 \text{ ตรม.} \\
 &= 292 \text{ ลิตร}
 \end{aligned}$$

ขั้นตอนที่ 4 การเลือกสปริงเกอร์

$$\begin{aligned}
 \text{ถ้าเลือกสปริงเกอร์ที่มีอัตราการจ่ายน้ำ 150 ลิตร/ชม. ที่แรงดัน 1.5 บาร์ ดังนั้นจะต้องใช้เวลาในการ} \\
 \text{ให้น้ำ} &= 292/150 \\
 &= 2 \text{ ชม}
 \end{aligned}$$

ถ้าในหนึ่งวันใช้เวลาในการให้น้ำ 8 ชม. จะมีคาบการให้น้ำ

$$\begin{aligned}
 &= 8/2 \\
 &= 4 \text{ คาบ}
 \end{aligned}$$

ขั้นตอนที่ 5 การแบ่งโซนการให้น้ำ

ในการแบ่งโซน (Zone) ของการให้น้ำสามารถแบ่งได้มากที่สุด

$$= \text{จำนวนคาบ} \times \text{รอบเวรการให้น้ำ}$$

$$= 4 \times 3$$

$$= 12 \text{ โซน}$$

ขั้นตอนที่ 6 ตรวจสอบอัตราการจ่ายน้ำของสปริงเกอร์

ตรวจสอบอัตราการจ่ายน้ำของสปริงเกอร์ โดยเปรียบเทียบกับอัตราซึมน้ำของดินร่วนปนทราย เพื่อมั่นใจว่าไม่ให้น้ำเร็วเกินไปจนเกิดน้ำนอง อัตราการซึมของดินร่วนปนทราย (จากตารางที่ 3.3) ได้ 10-20 มม. / ชม. อัตราการจ่ายน้ำของสปริงเกอร์ 150 ลิตร / ชม. พื้นที่วงเปียก 15.71 ตร.ม.

แปลง 150 ลิตร / ชม. น้ำต่อพื้นที่วงเปียกเป็นความลึกของน้ำบนพื้นที่วงเปียกเท่ากัน

$$= 0.150 / 15.71$$

$$= 0.00955 \text{ เมตร / ชม.}$$

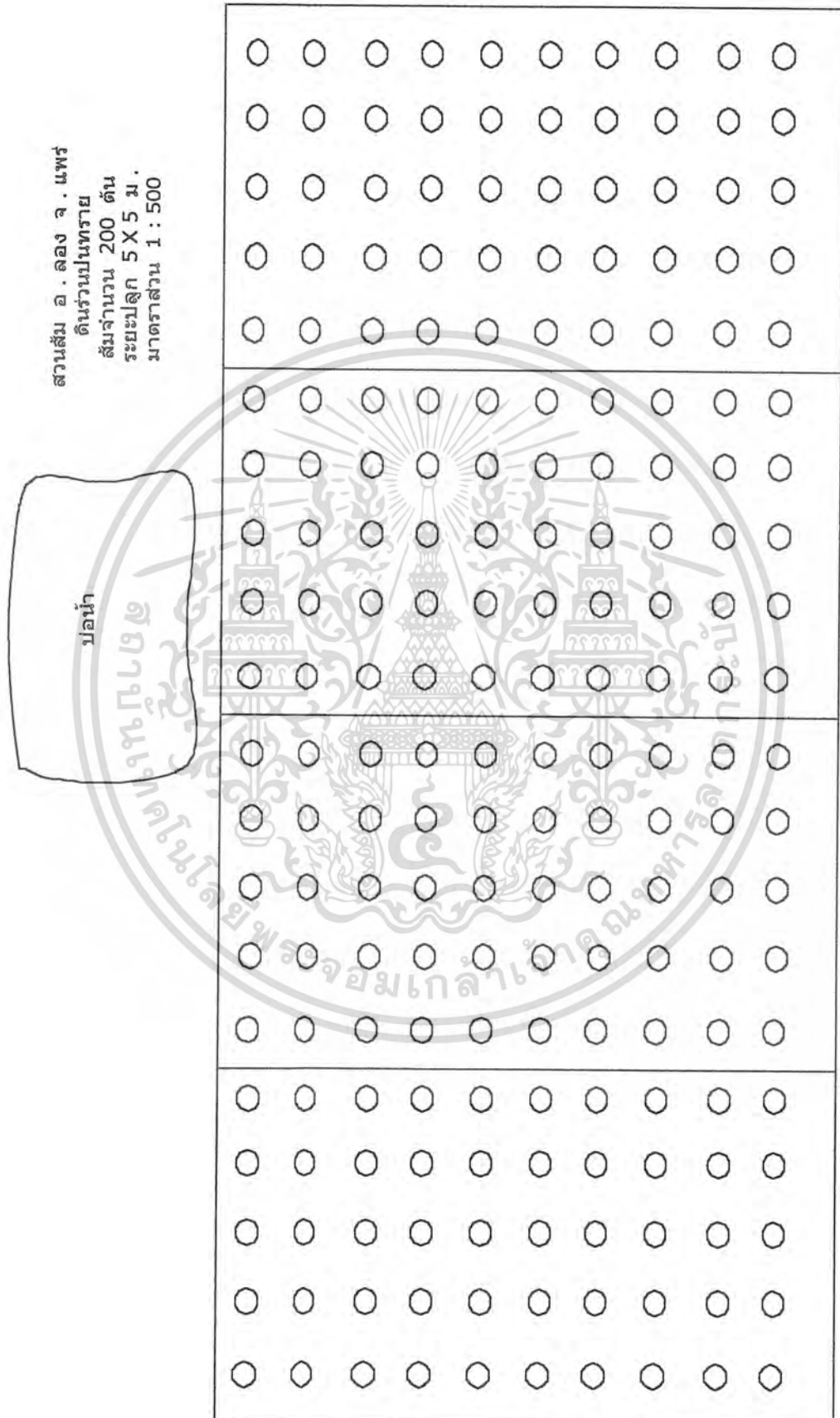
$$= 9.55 \text{ มม. / ชม. (ซึ่งต่ำกว่าอัตราซึมของดิน = 10-20 มม. / ชม.)}$$

แสดงว่าสปริงเกอร์ที่เลือกมาใช้ได้โดยจะไม่ก่อให้เกิดปัญหาการให้น้ำเร็วเกินไปจนซึมลงดินไม่ทัน

ขั้นตอนที่ 7 แนวคิดในการแบ่งโซน (Zone)

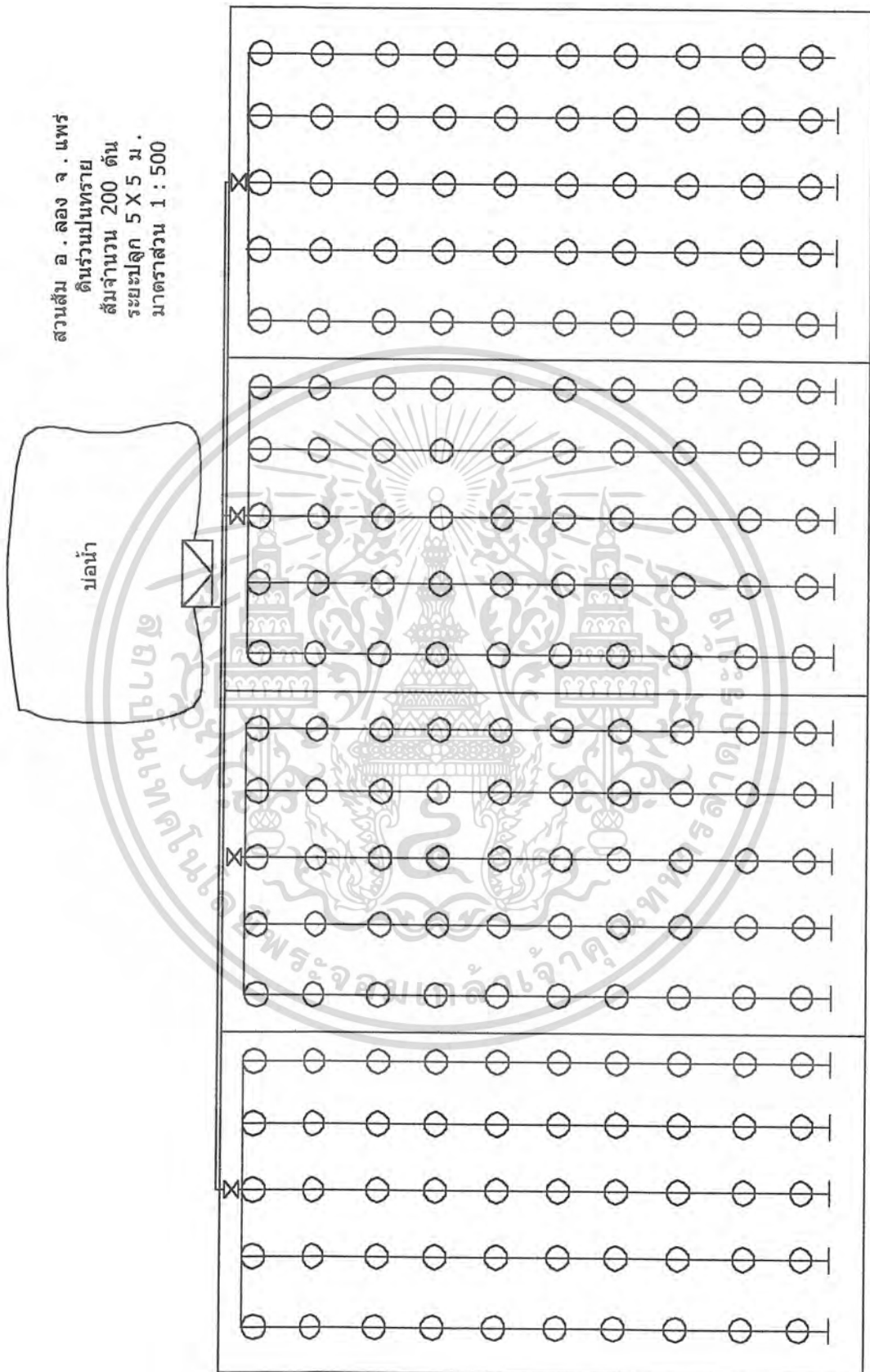
(7.1) หลักในการแบ่งโซน โดยทั่วไปแล้วจะพยายามแบ่ง โซนพื้นที่ออกมาเป็นจำนวนคู่ เพื่อความสะดวกในการเดินท่อและการใช้งาน นำแผนผังมาทำการแบ่งโซน จำนวนการแบ่งโซน สูงสุดได้เท่ากับ 12 โซน ในการแบ่งโซนนั้นไม่จำเป็นต้องแบ่งให้ได้ โซนมากเท่ากับจำนวนสูงสุดเสมอไป เพราะถึงแม้การแบ่งโซนมากจะได้อัตราการให้น้ำที่เล็กลง ขนาดท่อและเครื่องสูบน้ำก็เล็กตาม แต่จะต้องคำนึงถึงการปฏิบัติงานจะต้องคล่องตัวด้วย นอกจากนี้อุปกรณ์ที่ใช้จะมากกว่าการแบ่งโซนน้อยและเครื่องสูบน้ำที่เล็กมักจะมีประสิทธิภาพต่ำ

(7.2) แนวคิดในการพิจารณาหาจำนวนโซน ที่ดีที่สุด คือ การพิจารณาทดลองหาจำนวนโซน (และวางแนวท่อในการตัดสินใจ)หลายๆแบบแล้วเลือกเอาแบบที่ดีที่สุดมาใช้



รูปที่ 3.2 ลักษณะการแบ่งโซนสวนส้มออกเป็น 4 โซน

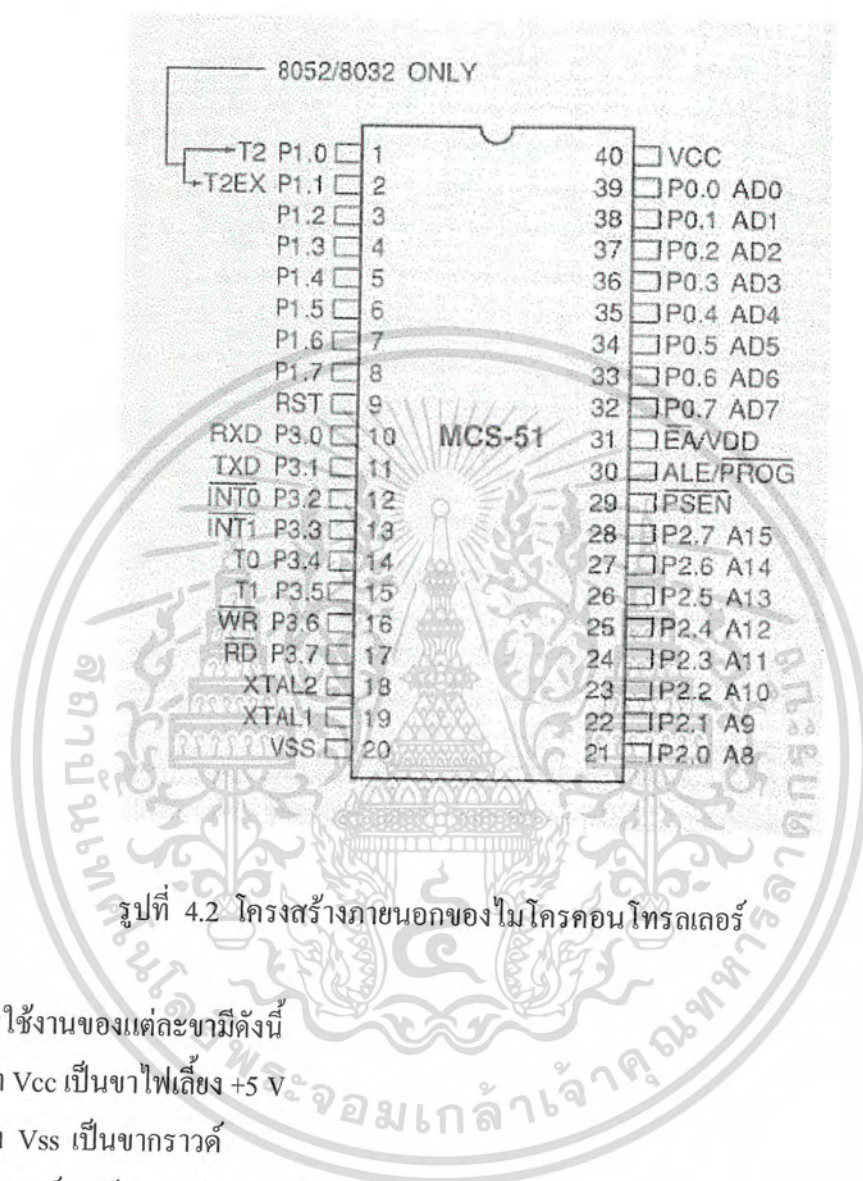
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ลักษณะการแบ่งโซนสวนส้มออกเป็น 4 โซน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 โครงสร้างภายนอกของ MCS 51



รูปที่ 4.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์

หน้าที่การใช้งานของแต่ละขามีดังนี้

ขา Vcc เป็นขาไฟเลี้ยง +5 V

ขา Vss เป็นขากาวด์

ขา พอร์ต 0 มี 8 ขา P0.0- P0.7

ขา พอร์ต 1 มี 8 ขา P1.0- P1.7

ขา พอร์ต 2 มี 8 ขา P2.0- P2.7

ขา พอร์ต 3 มี 8 ขา P3.0- P3.7

ขา รีเซ็ต RST ใช้สำหรับรีเซ็ตการทำงานของ CPU

ขา ALE/PORG เป็นขาควบคุมการแลตช์ (latch) ค่าแอดเดรสไบต์ต่ำเมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ยังทำหน้าที่เป็นรับพัลส์ในการโปรแกรมในส่วนของหน่วยความจำ EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา PSEN (PROGRAM STORE ENABLE) ทำหน้าที่เป็นสัญญาณสไตรบเพื่ออ่านคำสั่งจากหน่วยความจำ

ขา EA / VPP EXTERNAL ACCESS ENABLE /VPP เป็นขาสำหรับการเลือกใช้หน่วยความจำจากภายในหรือภายนอก สถานะ 0 คือให้ไมโครคอนโทรลเลอร์อ่านหน่วยความจำจากภายนอก

ขา XTAL1 และขา XTAL2 เป็นขาอินพุตและเอาต์ของวงจรถ่ายอินเทอร์ดีง

4.1.3 การจัดหน่วยความจำ

ในระบบ MCS 51 จะมีการแบ่งหน่วยความจำเหมือนกับ CPU ทั่วไปแบ่งเป็น 2 ชนิด

หน่วยความจำข้อมูล (data memory)

หน่วยความจำโปรแกรม (program memory)

หน่วยความจำข้อมูล (data memory)

หน่วยความจำข้อมูล หมายถึง หน่วยความจำส่วนที่เป็นแรม RAM ซึ่งเราสามารถอ่านหรือเขียนข้อมูลเปลี่ยนแปลง ได้ตลอดเวลาและถูกแบ่งออกเป็น 2 ส่วน เหมือนกันคือ หน่วยความจำข้อมูลภายในและภายนอก สำหรับหน่วยความจำภายในแบ่งเป็น 2 ส่วน คือ SFR กับส่วนที่ใช้เก็บข้อมูลทั่วไป

4.1.4 รีจิสเตอร์หน้าที่พิเศษ (SFR)

รีจิสเตอร์หน้าที่พิเศษมีบทบาทอย่างมากในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ และแสดงสถานะการทำงานในรีจิสเตอร์บางตัวยังสามารถเข้าถึงได้ระดับ บิต (bit addressable) ในรูปที่ 4.3

ตำแหน่ง แอดเดรส	(MSB)	บิตแอดเดรส							(LSB)	รีจิสเตอร์ หน้าที่พิเศษ
	WDT	T32	SERR	IZC	P3HZ	P2HZ	P1HZ	ALF		
0F8H	FF	FE	FD	FC	FB	FA	F9	F8	IOCON	
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B	
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC	
	CY	AC	F0	RS1	RS0	OV	F1	P		
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW	
0CDH	ไม่สามารถเข้าถึงได้ระดับบิต								TH2	
0CCH	ไม่สามารถเข้าถึงได้ระดับบิต								TL2	
0CBH	ไม่สามารถเข้าถึงได้ระดับบิต								RCAP2H	
0CAH	ไม่สามารถเข้าถึงได้ระดับบิต								RCAP2L	
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2		
0C8H	CF	CE	CD	CC	CB	CA	C9	C8	T2CON	
	PCT		PT2	PS	PT1	PX1	PT0	PX0		
0B8H	BF	—	BD	BC	BB	BA	B9	B8	IP	
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3	
	EA		ET2	ES	ET1	EX1	ET0	EX0		
0A8H	AF	—	AD	AC	AB	AA	A9	A8	IE	
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2	
99H	ไม่สามารถเข้าถึงได้ระดับบิต								SBUF	
	SM0	SM1	SM2	REN	TB8	RB8	T1	R1		
98H	9F	9E	9D	9C	9B	9A	99	98	SCON	
90H	97	96	95	94	93	92	91	90	P1	
8DH	ไม่สามารถเข้าถึงได้ระดับบิต								TH1	
8CH	ไม่สามารถเข้าถึงได้ระดับบิต								TH0	
8BH	ไม่สามารถเข้าถึงได้ระดับบิต								TL1	
8AH	ไม่สามารถเข้าถึงได้ระดับบิต								TL0	
89H	ไม่สามารถเข้าถึงได้ระดับบิต								TMOD	
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0		
88H	8F	8E	8D	8C	8B	8A	89	88	TCON	
87H	ไม่สามารถเข้าถึงได้ระดับบิต								PCON	
83H	ไม่สามารถเข้าถึงได้ระดับบิต								DPH	
82H	ไม่สามารถเข้าถึงได้ระดับบิต								DPL	
81H	ไม่สามารถเข้าถึงได้ระดับบิต								SP	
80H	87	86	85	84	83	82	81	80	P0	

รูปที่ 4.3 แสดงรีจิสเตอร์หน้าที่พิเศษ (SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 รีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์ใช้งานทั่วไปไม่มีไว้สำหรับให้ผู้เขียนโปรแกรมสามารถนำเอาข้อมูลไปเก็บไว้ได้ ซึ่งรีจิสเตอร์ใช้งานทั่วไปนี้มี 8 ตัว R0-R7 และมีให้ใช้ได้ถึง 4 แบนด์ BANK นั่นคือมีรีจิสเตอร์ใช้งานทั่วไปถึง 32 ตัว การเลือกใช้รีจิสเตอร์ในแบนด์ใดๆ กำหนดได้จากในรีจิสเตอร์หน้าที่พิเศษ PSW ทำให้ผู้เขียนโปรแกรมใช้งานรีจิสเตอร์ทั่วไปนี้ได้ถึง 32 ตัว อย่างเต็มที่และไม่ยุ่งยากในการเขียนโปรแกรม

ในส่วนของหน่วยความจำข้อมูลภายในที่อ้างอิงแบบ direct และ Indirect นั้น จะสามารถแบ่งออกได้ 3 ส่วน ดังแสดงในรูปที่ 7 โดยมีรายละเอียดดังนี้

ส่วนที่ 1 เรียกว่า REGISTER BANK 0-3 ซึ่งอยู่ที่ตำแหน่งหน่วยความจำข้อมูลภายในตั้งแต่ 00H ถึง 1FH จำนวน 32 ไบต์ โดยจะแบ่งออกเป็นชุด ชุดละ 8 ไบต์ จำนวน 4 ชุด ซึ่งแต่ละชุดจะมีชื่อเรียกเป็น R0-R7

ส่วนที่ 2 เรียกว่า Bit Addressable Area ซึ่งมีขนาด 16 ไบต์ ที่ตำแหน่งหน่วยความจำข้อมูล 20H ถึง 2FH ในส่วนนี้สามารถที่จะอ้างอิงได้เป็นระดับบิตถึง 128 บิต โดยการอ้างอิงตำแหน่งคดตรงในลักษณะบิต ตำแหน่ง 00H ถึง 0FH

ส่วนที่ 3 เรียกว่า scratch pad area จะอยู่ที่ตำแหน่ง 30H ถึง 7FH ซึ่งเป็นบริเวณหน่วยความจำข้อมูลภายในเอนกประสงค์ที่ผู้ใช้ได้โดยตรง นอกจากนี้ยังสามารถใช้เก็บข้อมูลแบบ STACK ได้ด้วย

ในส่วน ของบริเวณ นี้จะมีขนาด 128 ไบต์ แต่ในการใช้งานนั้นใช้ได้เฉพาะตำแหน่งเท่านั้น หากผู้ใช้อ้างอิงตำแหน่งที่นอกเหนือจากนี้นั้นจะได้ข้อมูลที่คาดเดาไม่ได้

4.1.6 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมสามารถแบ่งได้เป็น ส่วนคือ หน่วยความจำโปรแกรมภายในและภายนอกหน่วยความจำโปรแกรมภายในถูกเลือกใช้งานถ้า EA = 1 โดยจะถูกใช้ช่วงแอดเดรส 0-FFFFH ในกรณีตรงข้ามถ้า EA = 0 จะถูกใช้งานจากหน่วยจำจากภายนอก ในช่วงแอดเดรส 0-FFFFH โปรแกรมจะเขียนใน editor ซอร์สโค้ดที่ได้จะมีนามสกุล .ASM โปรแกรมแอสเซมบลียังไม่สามารถใช้งานได้ที่ทันที แต่ต้องนำแปลงเป็นภาษาเครื่องซึ่งมีนามสกุล .HEX

4.1.7 โปรแกรมแปลภาษา Assembly (Assembler)

ภาษา Assembly เป็นการเขียนโปรแกรมชนิดหนึ่งโดยการนำเอาคำสั่งของไมโครคอนโทรลเลอร์มาเขียนโปรแกรมสั่งการทำงานของไมโครคอนโทรลเลอร์ตามต้องการ ดังนั้นในการเขียนโปรแกรมจะต้องมีการแปลภาษา assembly เป็นภาษาเครื่องเสียก่อน machine code สำหรับ assembler สำหรับในโครงการนี้ใช้ SXS1 ซึ่งใช้แปลภาษา assembly เป็น machine code หรือ เลขฐาน 16 ซึ่งนามสกุล .HEX

คำสั่งเทียมของภาษา assembly

เพื่อให้การเขียนโปรแกรมง่ายขึ้นสะดวกขึ้นจึงต้องมีคำสั่งเทียมขึ้นมารูปแบบคำสั่งเทียมมีดังนี้

DB สำหรับกำหนดพื้นที่เก็บข้อมูลเป็นไบต์
 DS สำหรับกำหนดจองพื้นที่หน่วยความจำสำหรับเก็บข้อมูล
 EQU สำหรับกำหนดค่าตัวแปร
 ORG สำหรับกำหนดตำแหน่งเริ่มต้นของหน่วยความจำที่จะเขียน
 END สำหรับการสิ้นสุดของโปรแกรมที่จะเขียนซึ่งจำเป็นต้องมี

4.2 การใช้งานโปรแกรม SXAS1 ASSEMBLER

4.2.1 รายละเอียดทั่วไปของ SXAS1 ASSEMBLER

SXAS1 เป็นโปรแกรม 8051/51 ครอสแอสเซมเบลอร์ ใช้สำหรับทำการแปล SOURCE PROGRAM ซึ่งเขียนเป็นภาษาแอสเซมบลีของ MCS-51 (FILE.ASM) จะได้เอาท์พุทไฟล์เป็น INTEL HEX FILE (FILE.HEX) การเขียน SOURCE FILE ให้กระทำจากโปรแกรม EDITOR ใดๆ ก็ได้ จากนั้นจึงนำมาแปลด้วย SXAS1 ซึ่งทำได้โดยการเรียกใช้โปรแกรม SXAS1.EXE และตามด้วยชื่อ SOURCE PROGRAM ที่เขียนขึ้น การแปลของโปรแกรม SXAS1 จะกระทำในแบบ 2 PASS ASSEMBLER ซึ่งหากมีข้อผิดพลาดใด ๆ SXAS1 จะแสดงให้ทราบในขณะที่แปล ลักษณะของ SOURCE PROGRAM จะเป็นไปตามมาตรฐานของการเขียนโปรแกรม ASSEMBLY ทั่วไป

4.2.2 การใช้งาน SXAS1 ASSEMBLER

โปรแกรมแอสเซมเบลอร์ที่ใช้งานคือ SXAS1.EXE โดย ASSEMBLER จะทำการแปล SOURCE FILE ซึ่งถูกกำหนดไว้เป็น FILE.ASM เมื่อทำการแปลแล้ว จะได้ FILE OUTPUT เป็น INTEL-HEX FILE ซึ่งถูกกำหนดไว้เป็น FILE.HEX โดย FILE.HEX นี้เอง ที่ผู้ใช้สามารถนำไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหลดเข้า บอร์ดต่างๆ หรืออัดเข้าตัว EPROM เพื่อนำไปทดสอบโปรแกรมตามที่เขียนได้ นอกจากนี้ SXAS1 ยังสามารถสร้าง LISTING FILE ต่าง ๆ ได้ตามรูปแบบของ OPTION ดังนี้

A>SXAS1 <-OPTION> SOURCE FILE

โดย SOURCE FILE : คือชื่อ FILE ของ โปรแกรม กรณีที่ไม่ใส่ नामสกุล ของ FILE ตัว SXAS1 จะกำหนดให้เป็น .ASM โดยอัตโนมัติ

<-OPTION> : เป็นส่วนของการกำหนด OPTION ต่างๆ ประกอบด้วยเครื่องหมาย “-“ และตามด้วยตัวอักษรย่อเพียงตัวเดียว ซึ่งการกำหนด OPTION สามารถกำหนดได้มากกว่า 1 ตัวในการแปลแต่ละครั้ง รายละเอียด ของ OPTION จะมีดังนี้

- L หมายถึงให้สร้าง FILE แสดงรายละเอียด (LISTING FILE) โดย OUTPUT FILE ที่ได้คือ FILE.LST

- N หมายถึงให้ทำการคอมไพล์ โดยไม่สร้าง FILE OUTPUT ใดๆ ทั้งนี้ เพื่อการตรวจสอบความถูกต้องของ โปรแกรมอย่างรวดเร็ว

- C หมายถึงให้สร้าง FILE แสดงรายละเอียด เช่นเดียวกับ -L โดยเพิ่ม symbol cross-reference ไว้ที่ท้ายของ FILE ทั้งนี้เพื่อความสะดวกในการค้นหาตำแหน่งต่าง ๆ ตามชื่อ LABEL

- D หมายถึง ให้แสดง PROCESS ขณะที่กำลังทำการ ASSEMBLER โดยโปรแกรมจะแสดงเลขที่บรรทัดที่กำลังแปลอยู่
ตัวอย่างการใช้คำสั่ง

A>SXAS1 J31EX4

หมายถึง SOURCE : J31EX4.ASM

LISTING : ไม่สร้าง

HEX : สร้าง HEX FILE ชื่อ J31EX4.HEX

A>SXAS1 -L J31EX4.ASM

หมายถึง SOURCE : J31EX4.ASM

LISTING : สร้าง LISTING FILE ชื่อ J31EX4.LST

HEX : สร้าง HEX FILE ชื่อ J31EX4.HEX

A>SXAS1 -N J31EX4.ASM

หมายถึง SOURCE : J31EX4.ASM

LISTING : ไม่สร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ข้อกำหนดของ SOURCE FILE

1. ในแต่ละบรรทัดต้องมีคำสั่งในภาษาแอสเซมบลี หรือ ASSEMBLY DIRECTIVE เพียงหนึ่งคำสั่งเท่านั้น
2. LABEL DECLARATION ทุก LABEL ต้องปิดท้ายด้วย “:”
3. โปรแกรม SXAS1 จะมองตัวอักษรตัวเล็กหรือตัวใหญ่เหมือนกัน
4. บรรทัดว่างใน โปรแกรมจะยังคงปรากฏอยู่ใน FILE แสดงรายละเอียด (FILE.LST) แต่ ใน FILE แสดงรหัสภาษาเครื่อง (FILE.HEX) จะถูกตัดทิ้งไป
5. SOURCE FILE ในแต่ละบรรทัดต้องประกอบด้วยตัวอักษรไม่เกิน 128 ตัวอักษร ซึ่งในแต่ละบรรทัดต้องมีรูปแบบดังนี้

[LABEL:] OPERATION [OPERAND1],[OPERAND2] [;COMMENT]

LABEL เป็นกลุ่มของตัวอักษรและตัวเลขที่จะใช้เป็นตัวแปร หรือตำแหน่งหน่วยความจำสำหรับอ้างอิงในโปรแกรม LABEL ต้องขึ้นต้นด้วยตัวอักษร A-Z หรือ “_” “.” “?” และต้องปิดท้ายด้วย “:” เสมอ

OPERATION ส่วนนี้จะเป็นคำสั่งมาตรฐานของ MCS51 เช่นคำสั่ง MOV INC DEC ADD

OPERAND เป็นกลุ่มของตัวอักษรและตัวเลขซึ่งอาจจะอยู่ในรูปของการกระทำทางคณิตศาสตร์ หรือการกระทำทางด้าน LOGIC

COMMENT หมายถึงประโยคหรือข้อความที่ตามหลังเครื่องหมาย “:” โดยทั่วไปมักจะใช้อธิบายการทำงานของโปรแกรมสั้นๆ เพื่อให้ทำความเข้าใจได้ง่ายขึ้น เครื่องหมาย “:” จะอยู่ส่วนใดของโปรแกรมก็ได้ ซึ่งข้อความที่ตามหลังเครื่องหมาย “:” จะไม่มีผลต่อการแปลของ ASSEMBLER ASSEMBLER DIRECTIVE

DB expression [,expression...]

DB (DEFINE BYTE) คำสั่งนี้ใช้กำหนดค่าข้อมูลเป็นไบต์ลงในหน่วยความจำโปรแกรม สามารถใส่ข้อความรหัสแอสกี(ASCII STRING) ลงตำแหน่งในหน่วยความจำได้ โดยข้อความนั้นต้องอยู่ในเครื่องหมาย QUOTES “” และยังกำหนดค่าข้อมูลที่เป็นตัวเลขและตัวอักษรในบรรทัดเดียวกันได้โดยการคั่นด้วยเครื่องหมาย “,” เช่น

```
POINTER DB 'This is a string',0
```

DS expression

DS (DEFINE SPACE) ใช้สำหรับกำหนดที่ว่างเป็นจำนวน ไบต์สำหรับเก็บค่าตัวแปรหรือใช้เป็นส่วนของ BUFFER

DW expression [,expression...]

DW (DEFINE WORD) ใช้ในการเขียนข้อมูล 16 บิต ของหน่วยความจำโปรแกรมโดยการกำหนดค่าข้อมูลไบต์ต่อก่อนแล้วตามด้วยข้อมูลไบต์สูง โดยข้อมูลแต่ละ WORD ต้องคั่นด้วยเครื่องหมาย“,” เช่นเดียวกับคำสั่ง DB

ORG expression

ORG (ORIGIN) ใช้สำหรับกำหนด ADDRESS เริ่มต้นของโปรแกรม (ORIGIN PROGRAM LOCATION)

END

END คำสั่งนี้มีไว้เพื่อบอกให้ ASSEMBLER ทราบถึงจุดสิ้นสุดของโปรแกรมว่าอยู่ที่ตำแหน่งใดคำสั่งนี้จำเป็นต้องใส่ไว้ท้ายโปรแกรมด้วย

SYMBOL_NAME EQU expression

สำหรับการกำหนดค่าให้กับ SYMBOL_NAME (หรือที่เรียกกันว่า LABEL) โดยสามารถใช้คำสั่ง EQU หรือคำสั่ง BIT ก็ได้ ซึ่งจะทำงานเหมือนกันทุกประการ

SYMBOL_NAME SET expression

คำสั่ง SET จะทำงานเหมือนกับ EQU ทุกประการ แต่จะสามารถกำหนดให้ SYMBOL_NAME มีค่าได้หลาย ๆ จุดตามที่ต้องการ

4.2.4 การใช้งาน SXAS1

เนื่องจากโปรแกรม SXAS1 ทำงานภายใต้ DOS ต้องกำหนดให้เครื่องที่เป็น WINDOW เข้าสู่ MS DOS MODE ก่อนที่จะมีการใช้งาน SXAS1 มีรูปแบบดังนี้

SXAS1 [-lc] [source-file]

L สร้างไฟล์เอกสารนามสกุล .LST ขึ้นมา

Source file เป็น file ที่เก็บโปรแกรมภาษา assembly ในรูป text file จะเป็นไฟล์ชนิด

ASM

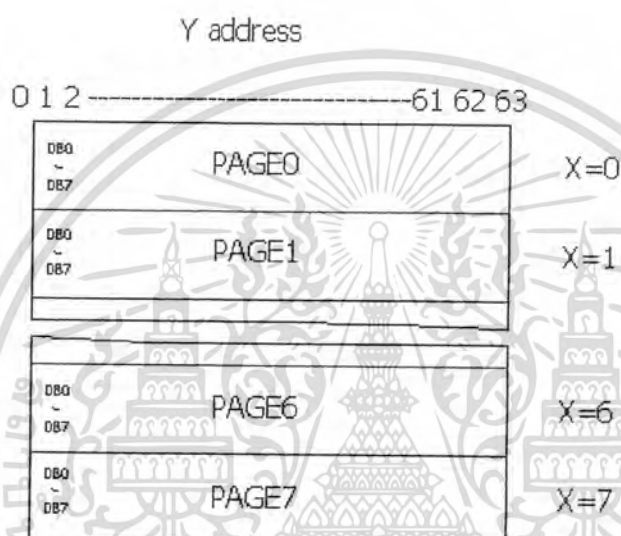
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 GRAPHIC LCD

LCD 12864 เป็น LCD Graphic ขนาด 128*64 DOT ซึ่งมี Controller (HD 61202, HD61203) โดยการทำงานของ controller จะมีลักษณะการแบ่งการควบคุมไว้ดังนี้

Line คือการอ้างถึงบรรทัดของข้อมูล ภายในจะแบ่งเป็น 64 แถว (com1 – com64)

PAGE X ADDRESS เป็นการอ้างถึงหน้าต่างของการแสดงผลภายในหนึ่ง page จะประกอบไปด้วย



รูปที่ 4.4 แสดงการแบ่งการควบคุม

8 line ซึ่งจะเป็นการอ้างถึงถึง DATA BUS โดยตรง ภายในของ LCD จะประกอบด้วย 8 line ซึ่งถูกชี้โดย X register โดยเมื่อต้องการให้ LCD แสดงผลที่หน้าต่างข้างใดของจอ เราต้องตั้งค่า x ให้กับ LCD เมื่อตั้งแล้วค่า x จะไม่เปลี่ยนแปลง

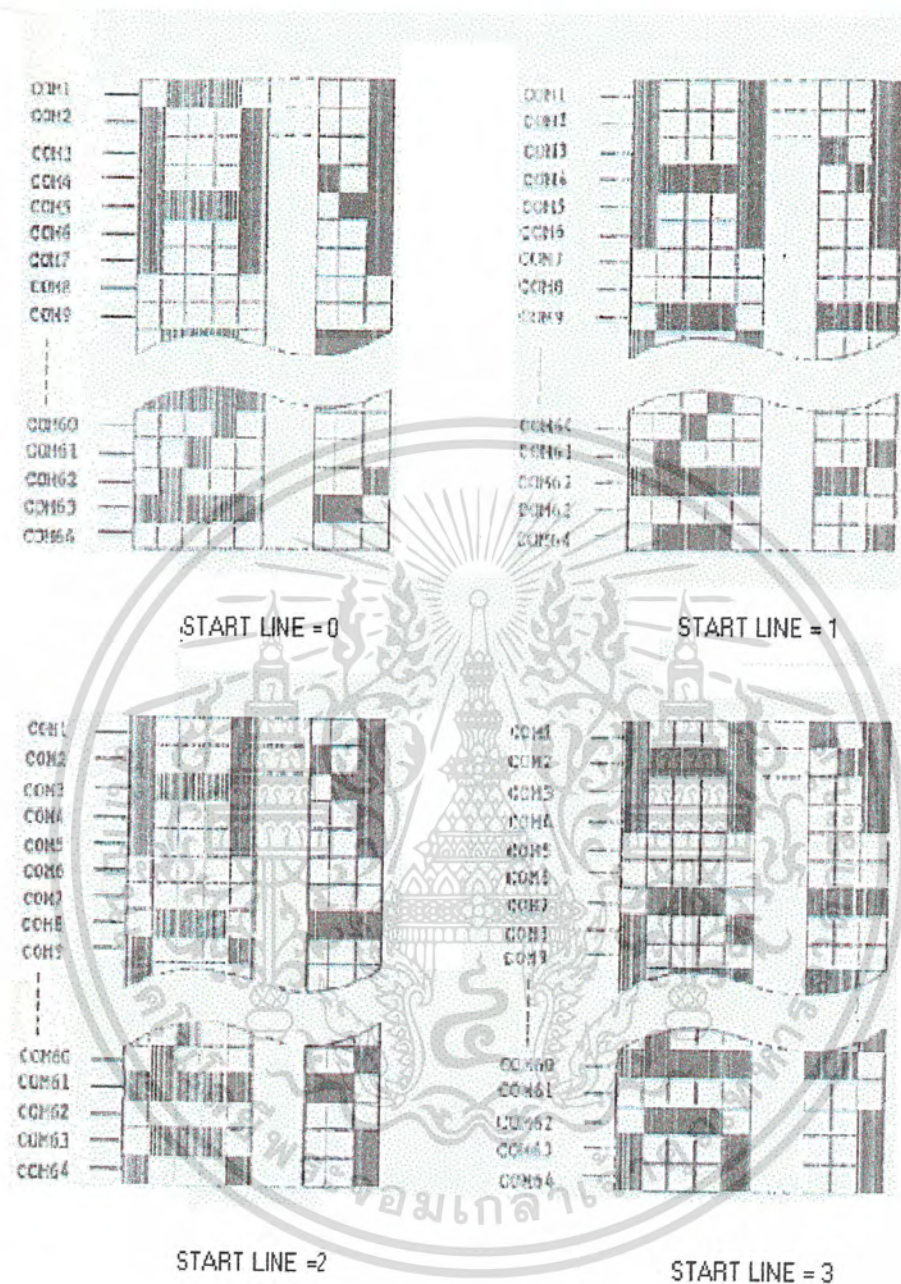
segment (Y address) เป็นค่าพอยท์เตอร์ในการชี้ที่อยู่ของข้อมูล ซึ่งภายใน LCD จะถูกควบคุมการชี้ของข้อมูล โดย HD61202 โดยในตัว HD61202 จะสามารถชี้ที่อยู่ของข้อมูลได้ 64 segment ซึ่ง HD61202 ทั้งสองตัวก็จะสามารถทำการค้าง segment ได้ถึง 128 segment

โดยการใช้งาน เมื่อทำการตั้งค่า Y แล้ว ค่าจะถูกเพิ่มค่าขึ้นเสมอ เมื่อมีการอ่านหรือเขียนข้อมูลบน LCD

**** ข้อควรระวัง ****

เมื่อค่า Y ถูกเพิ่มขึ้นมากกว่า 63 แล้ว ค่า Y จะยังไม่เป็นการอ้างถึงข้อมูล segment 64 (segment 0 ของ cs2) ดังนั้นตัวโปรแกรมจะต้องช่วยจัดการในส่วนนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

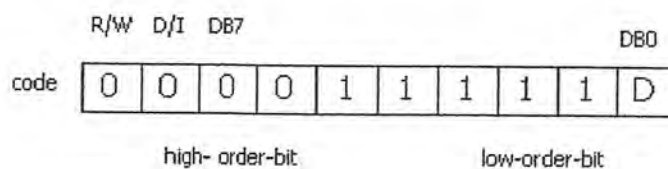


รูปที่ 4.5 การตั้งค่า Display start line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 คำสั่งควบคุมของ LCD

1 display ON/OFF



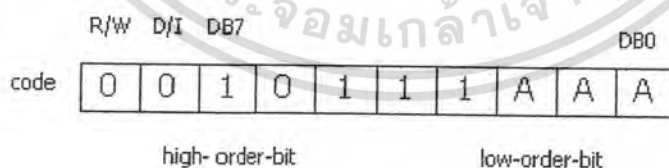
เป็นคำสั่งควบคุมการแสดงผล โดยการแสดงผลขึ้นอยู่กับค่า D (DB0) เมื่อค่า D เป็น 1 LCD จะเป็นการแสดงผล และ เมื่อค่า D เป็น 0 LCD จะไม่ทำการแสดงผล ข้อมูลภายใน LCD จะไม่มีการแสดงผล

2 Display start line



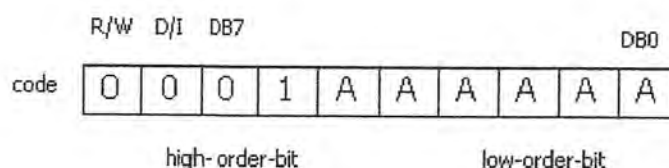
ค่า A จะเป็นค่าหมายเลขบรรทัด ที่จะทำให้ LCD แสดงผลเป็นบรรทัดแรกของจอภาพ ในรูปที่ 4.5 จะเป็นตัวอย่าง ของการเลือกค่า LINE จาก 0-3 ซึ่งจะทำให้การแสดงผลแตกต่างกันออกไป

3. SET PAGE (X-Address)



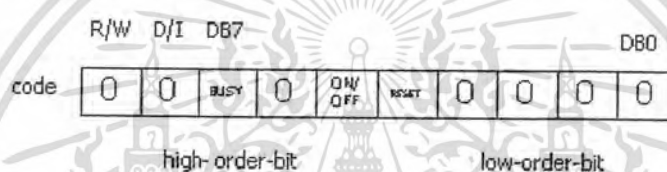
ค่า AAA ของคำสั่ง จะเป็นการตั้งค่า X Address ซึ่งหลังจากทำคำสั่งนี้แล้ว ข้อมูลจาก DB0-DB7 จะเป็นการติดต่อที่ page ตลอด จนกว่าจะมีการตั้งค่าใหม่ให้กับ LCD

4. SET Y Address



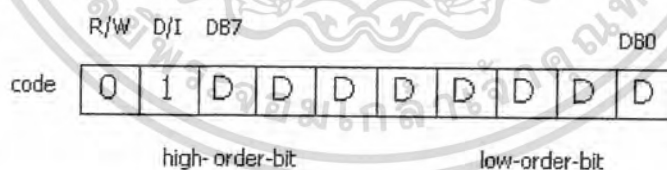
ค่า A จะเป็นการตั้งค่าของ Y Address ค่า Y จะมีค่าอยู่ระหว่าง 0-63 และ ค่า Y จะเพิ่มครั้งละ 1 เมื่อมีการเขียนหรืออ่านข้อมูล จาก CPU

5. STATUS Read



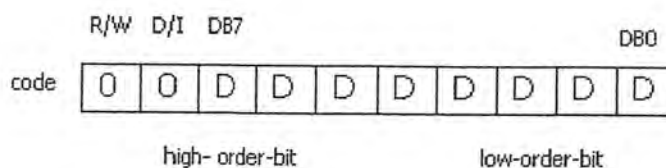
เป็นการอ่านค่าสถานะของ LCD โดยถ้า BUSY เป็น 1 LCD จะทำงานในส่วนภายใน ซึ่งจะทำให้ไม่สามารถควบคุม LCD ขณะนี้ได้ เพราะฉะนั้นเพื่อให้แน่ใจในการควบคุมครั้งต่อไป จะต้องตรวจค่า BUSY ให้ได้ค่าเป็น 0 เสียก่อน

6. WRITE DISPLAY DATA



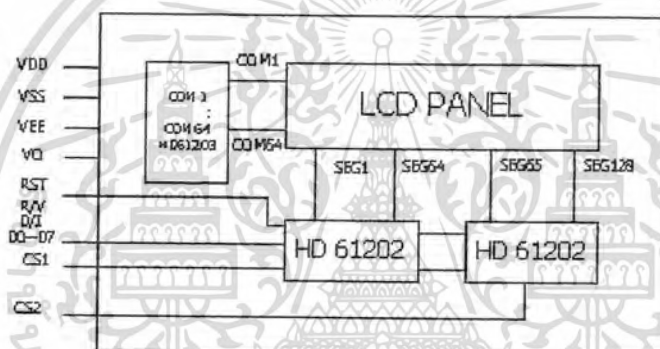
การเขียนข้อมูลเข้า LCD ซึ่งข้อมูล DDDDDDDD จะถูกเก็บใน LCD RAM และ ค่า Y จะถูกเพิ่มขึ้น 1

7.READ DISPLAY DAT



เป็นการอ่านข้อมูลที่แสดง โดย LCD จะให้ค่าของข้อมูลออกมาที่ DATA BUS ค่า Y จะถูกเพิ่มค่าขึ้น 1 เช่นเดียวกับการเขียนข้อมูล

4.3.2 INTERFACE LCD DV-12864



รูปที่ 4.6 โครงสร้างภายในและขาควบคุม

โครงสร้างภายในของ LCD จะประกอบด้วย ส่วนของ CONTROLLER โดย HD 61203 จะควบคุมการอ้าง page ของข้อมูล และ HD61202 จะควบคุมในการอ้างของ segment ซึ่งในการใช้งาน เราจะต้อง control ส่วนเหล่านี้ โดยการส่งรหัสควบคุมไปที่ขาของ LCD ดังนี้

ขา RST เป็นขาที่ใช้ RESET การทำงานของ LCD

ขา E เป็นขา ENABLE การรับส่งข้อมูล จะทำงานที่ LOGIC HIGH และขอบขาลง

ขา R/W เป็นขาที่ใช้กำหนด การอ่านหรือเขียนข้อมูล

ขา CS1 Chip Select ของ HD612202 ตัวแรก

ขา CS/ Chip Select ของ HD612202 ตัวสอง

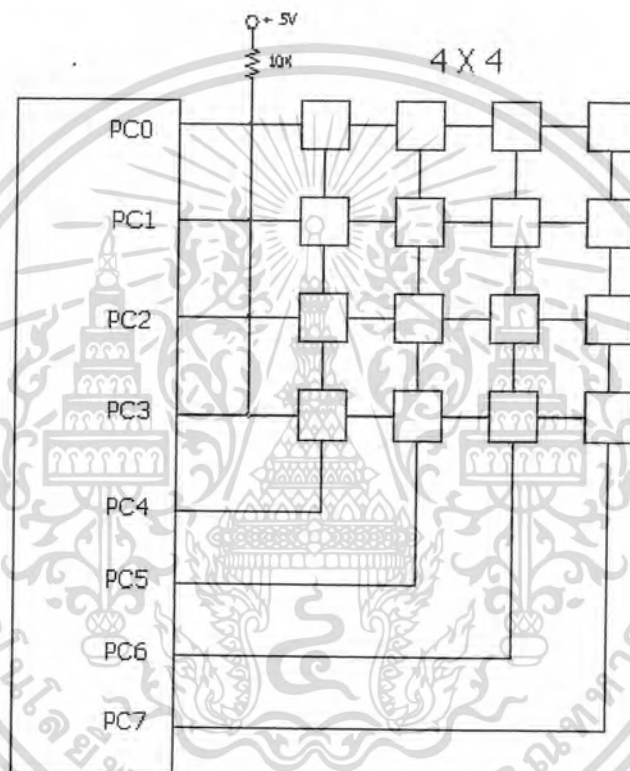
ขา DATA BUS เป็นขาใช้ส่งข้อมูล หรือ รหัสควบคุม

หมายเหตุ 1 เมื่อ CS1 เป็น High และ CS2 เป็น LOW จะเป็นการอ้างอิง segment ที่ 0-63 และเมื่อ CS1 เป็น LOW CS2 เป็น High จะเป็นการอ้างอิง segment ที่ 64-127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 คีบอร์ดแบบเมตริก 4 x 4

การค่าจากคีย์บอร์ดหรือรับค่าจากการกดสวิตช์โดยใช้ไมโครคอนโทรลเลอร์นั้นคือการอ่านค่าจากลอจิกที่เกิดขึ้นเมื่อมีการกดสวิตช์โดยการอ่านค่าที่พอร์ตของไมโครคอนโทรลซึ่งมีค่าลอจิกเป็น 0 หรือ 1 นั้นเองการต่อวงจรของสวิตช์จึงมีลักษณะใหญ่ 2 ประการ ต่อเข้ากับไฟเลี้ยง หรือ กราวด์โดยตรงซึ่งส่วนมากแล้วจะใช้ลอจิก 0 เป็นหลัก เนื่องจากตรวจได้ง่ายกว่านั้นเอง ในรูปที่ 4.7 แสดงวงจรการรับสวิตช์อย่างง่าย



รูปที่ 4.7 การต่อแบบเมตริก (Matrix switch)

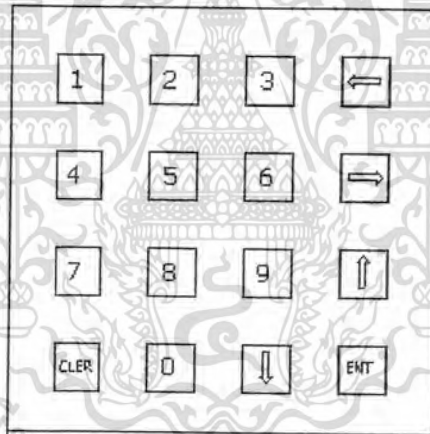
สำหรับในรูปที่ 4.7 ต้องการใช้สวิตช์จำนวนมากแล้วจะนิยมการต่อแบบเมตริก (Matrix switch) ดังในรูปที่ 4.7 ซึ่งสวิตช์จะถูกต่อกันในลักษณะแนวแกนตั้งและแกนนอน โดยจะเรียกแนวตั้งว่า คอลัมน์ column ในขณะที่แกนนอนเรียกว่า แถว ROW ดังนั้นค่าของสวิตช์จะต้องประกอบด้วย ตำแหน่งในแนวหลักและแถว ซึ่งกระบวนการที่จะอ่านค่าจากสวิตช์ได้นั้นจะใช้การทำงานของโปรแกรมและความเร็วของไมโครคอนโทรลเลอร์ เพื่อตรวจสอบสวิตช์ที่เรียกว่า Multiplex โดยวิธีการตรวจสอบทีละคอลัมน์จนครบวงจรสวิตช์แบบนี้สามารถรองรับกับสวิตช์ที่เพิ่มขึ้นได้ อย่างเพียงพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจร ในรูป 4.7 เป็นวงจรที่ใช้ในโครงการนี้โดยใช้พอร์ต C ของ 8255 ในบอร์ดที่
งานทำหน้าที่ตรวจสอบการกดสวิตซ์ขนาด 16 ตัว โดยให้ 4 บิต แรกเป็น บิตอินพุต และ 4 บิต
บนทำหน้าที่เป็นเอาต์พุต โดยในส่วนนี้ใช้ความต้านทานต่อกับ 5 โวลต์ ไว้เพื่อให้ขณะที่ไม่มีการ
กดสวิตซ์มี สถานะลอจิก เป็น 1 ซึ่งจุดตัดในแนวนอนและแนวตั้งไม่ได้ต่อดึงกันและจะต่อดึงกันก็
ต่อเมื่อกดสวิตซ์เท่านั้น

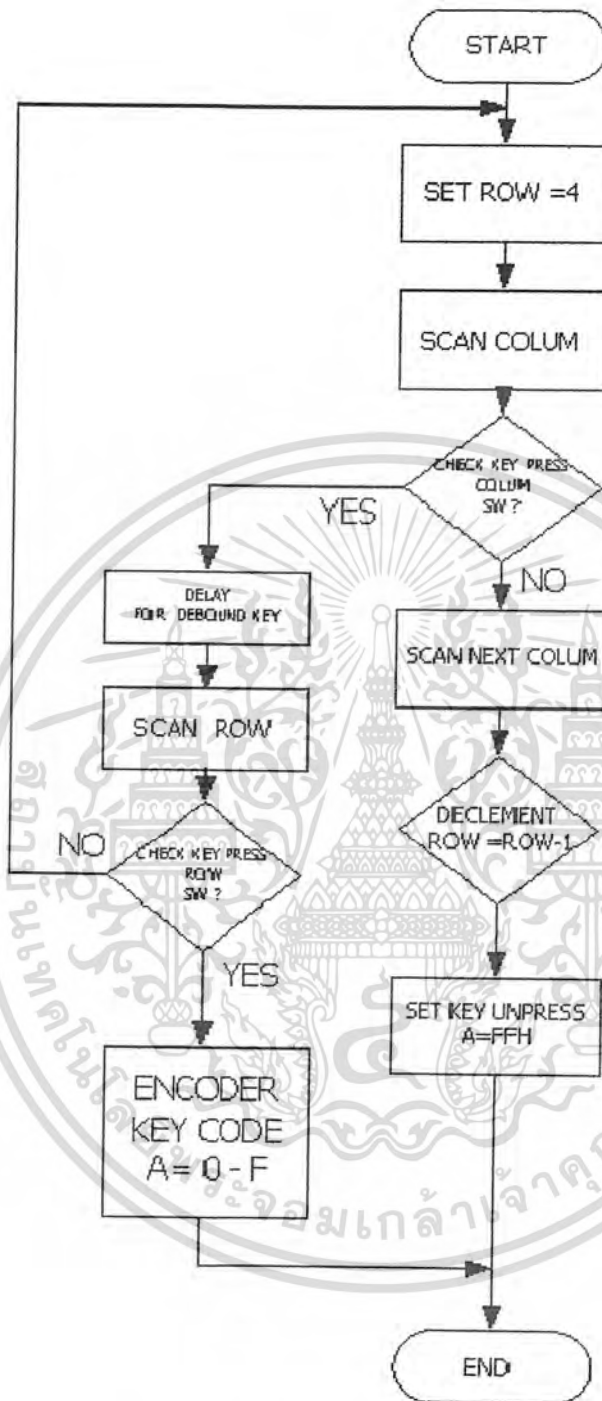
4.3.4 ขั้นตอนการเขียนโปรแกรมตรวจสอบสวิตซ์มีดังนี้

จะต้องให้บิตที่ควบคุมคอลัมน์แรกเท่านั้นที่เป็นลอจิก 0 หลังจากนั้นจะอ่านค่าในส่วน
ของแถวหรือโรว์เพื่อตรวจสอบค่าในบิตจนครบจำนวนหากตรวจสอบพบว่ามีสวิตซ์ตัวใดถูกกดจะ
เก็บค่าของตัวนับไว้ ทำอย่างนี้จนครบทุกคอลัมน์จบขั้นตอนการรับการกดสวิตซ์



รูปที่ 4.8 คีย์บอร์ดแบบเมตริก 4 x 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 บล็อกไดอะแกรม Scankey board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การสร้างและประกอบโครงงาน

ส่วนประกอบของโครงงานระบบควบคุมน้ำหยด ประกอบด้วยบอร์ดควบคุมไมโครคอนโทรลเลอร์บอร์ดไมโครเวร์ซึ่งทำหน้าที่ขับโซลินอยด์วาล์วให้ทำงาน และในส่วนของเพาเวอร์ซัพพลายเพื่อจ่ายไฟเลี้ยงให้กับวงจร

ต่าง ๆ ซึ่งออกแบบแยกเป็น 2 ส่วนใหญ่ ๆ คือ

ส่วนที่ 1 บอร์ดควบคุมไมโครคอนโทรลเลอร์

ส่วนที่ 2 บอร์ดไมโครเวร์

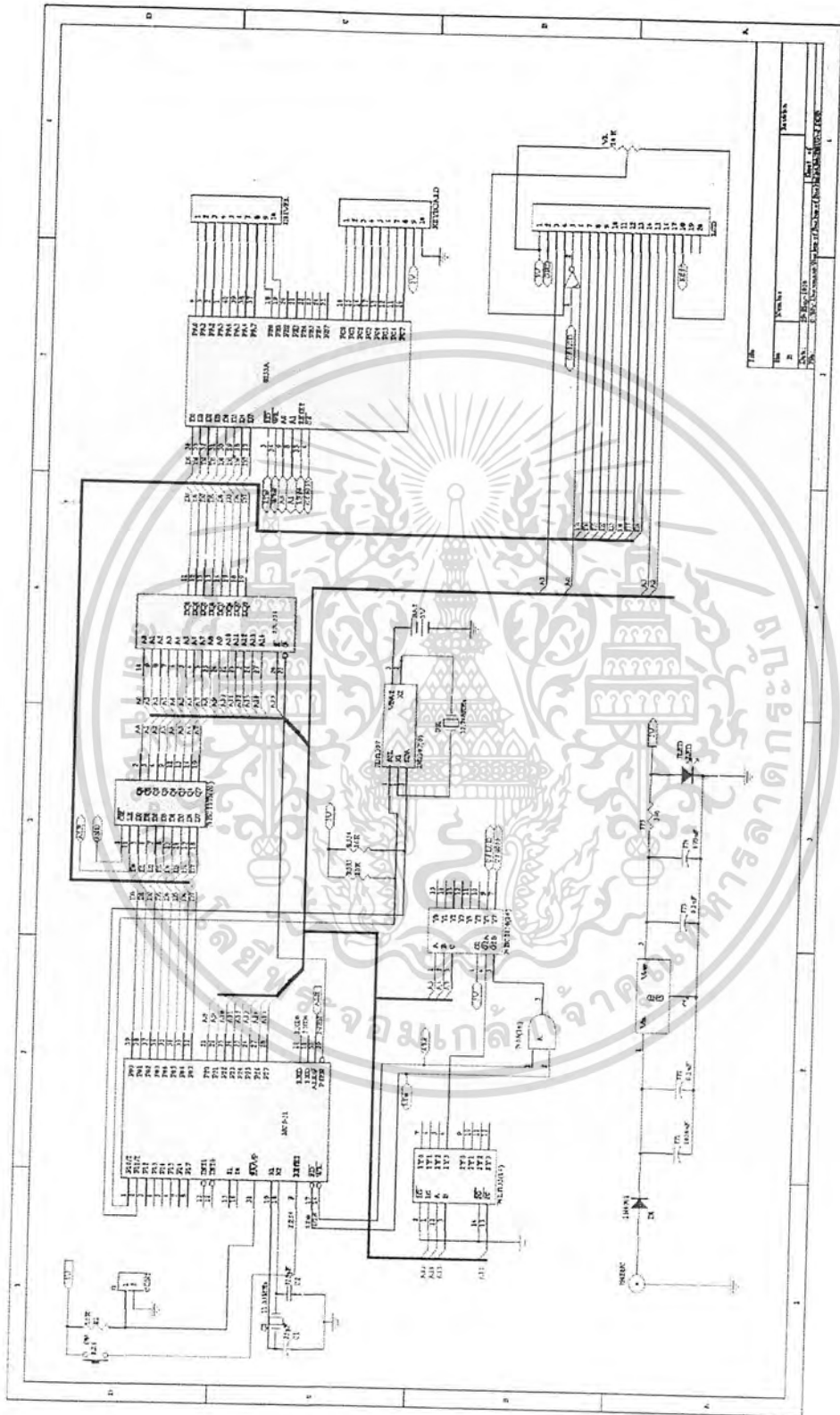
5.1 คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์

- CPU 89S8252 Run ความถี่ 11.059 Mhz
- 32 KB Eprom Monitor (27256)
- 8 KB / 32KB RAM พร้อมวงจร BACKUP (6264/65256) on Board 8KB (62256)
- 32 KB ROM /RAM Expantion (27256/62256)
- 40 PIN MCS 51 Bus compatible
- 20 pin Expantion CPU Port (P1 & P3) compatible with AT89C1051 /2051 CPU
- 24 Bit I / O 8255 Port 34 Pin
- keyboard Matrix 4x4
- Serial Port RS232 Channel
- Real Time Clock (RTC DS1307)
- 7805 Regulateor Power Supply onboard (+ 9 V DC Input

5.1.1 ลักษณะทั่วไปของบอร์ดควบคุมไมโครคอนโทรลเลอร์

บอร์ดควบคุมไมโครคอนโทรลเลอร์ เป็นบอร์ดไมโครคอนโทรลเลอร์ ในตระกูล MCS 51 ที่ถูกออกแบบและพัฒนาอย่างต่อเนื่อง โดยได้พัฒนาโครงสร้างของในส่วนที่มีข้อจำกัดบ้างประการ และเสริมประสิทธิภาพของวงจรใช้งานทั้งในด้านของหน่วยความจำและอุปกรณ์สนับสนุน อินพุทเอาต์พุต / เอาต์พุทแบบต่าง ๆ ให้สามารถให้สามารถเลือกใช้ได้หลากหลายมากขึ้นและยังคงขนาดของบอร์ดให้เล็กลงขนาดเหมาะสมต่อการใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

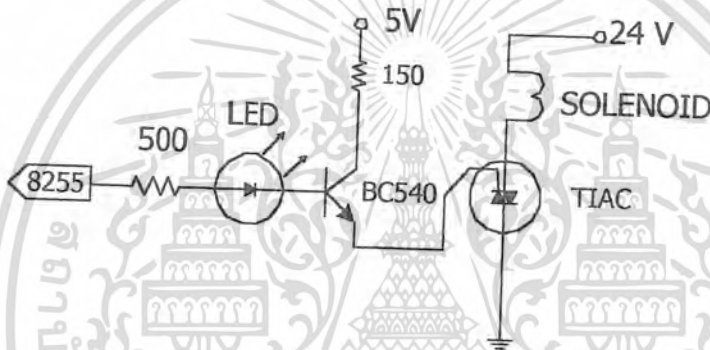


รูปที่ 5.1 แสดงวงจรควบคุมระบบนำหยุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 วงจรควบคุม SOLENOID VALE

วงจรควบคุม SOLENOID VALE แสดงในรูปที่ 5.2 สัญญาณอินพุตมาจากคอนเนกเตอร์ ซึ่งเชื่อมต่อเข้ากับบอร์ด Interface แล้วส่งไปให้ทรานซิสเตอร์ขยายกระแสและป้องกันความเสียหายที่จะเกิดขึ้นกับวงจรได้ เอาต์พุตจากทรานซิสเตอร์ จะส่งไปยังขาเกตของ ไตแอค TIAC เพื่อทริกให้ไตแอค ทำงานนำกระแส และที่ LOAD ซึ่งถูกต่อโดย โซลินอยด์วาล์วให้ทำงานซึ่ง โซลินอยด์วาล์วนี้จะไปเปิดให้น้ำ ตามการสั่งงานจากบอร์ดควบคุมไมโครคอนโทรลเลอร์ให้ทำงานตามที่เราระบุเวลาที่หน้าจอ LCD ซึ่งการทำงานของโซลินอยด์วาล์ว จะทำงานตามคำสั่งของไมโครคอนโทรลเลอร์



รูปที่ 5.2 วงจรควบคุม SOLENOID VALE

ส่วนประกอบของโครงการ

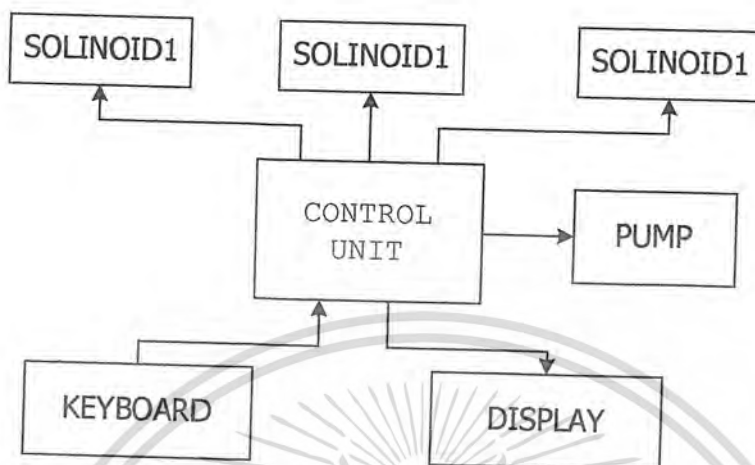
โครงการระบบควบคุมน้ำหยดประกอบด้วย 2 ส่วน คือ

1. HARDWARE
2. SOFTWARE

5.2.1 ส่วนประกอบทาง HARDWARE

HARDWARE จะประกอบไปด้วย 2 ส่วนหลัก คือ ส่วนของ CONTROL UNIT และส่วนของ MACHANICS HARD WARE ดังรูป 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

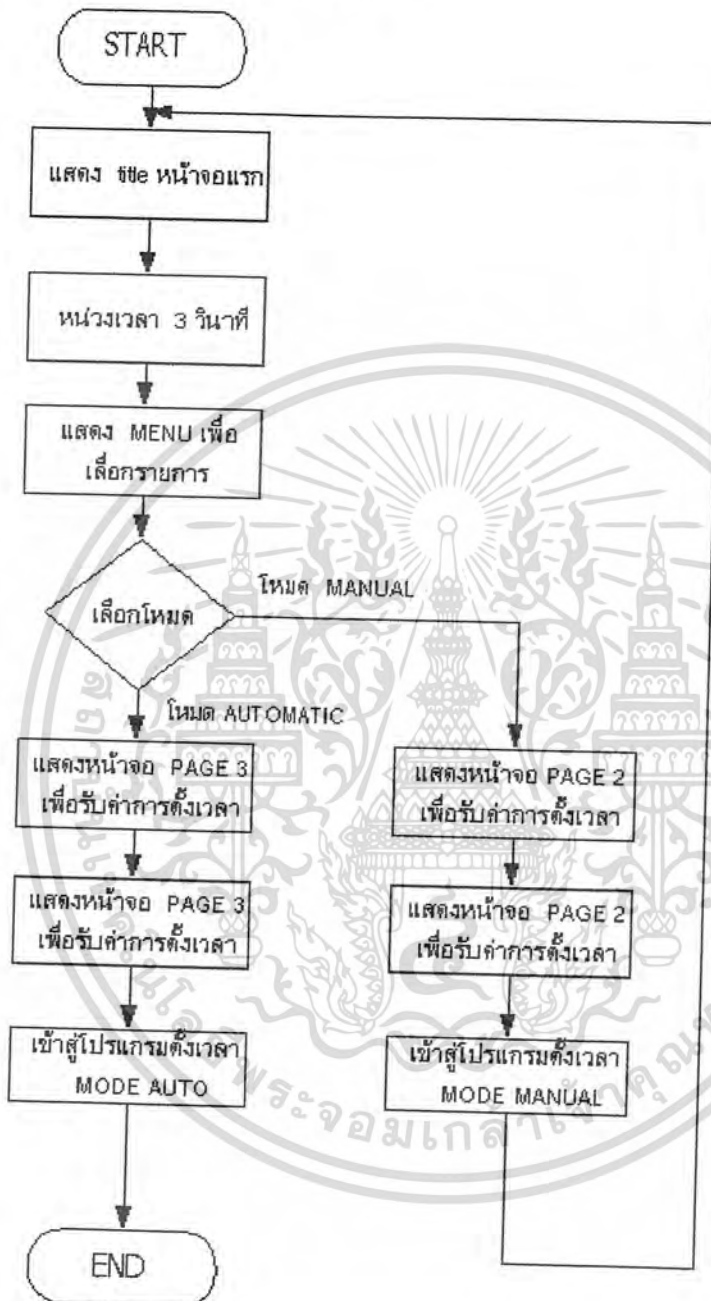


รูปที่ 5.3 ส่วนประกอบทาง HARDWARE

5.2.2 ส่วนประกอบทาง SOFTWARE

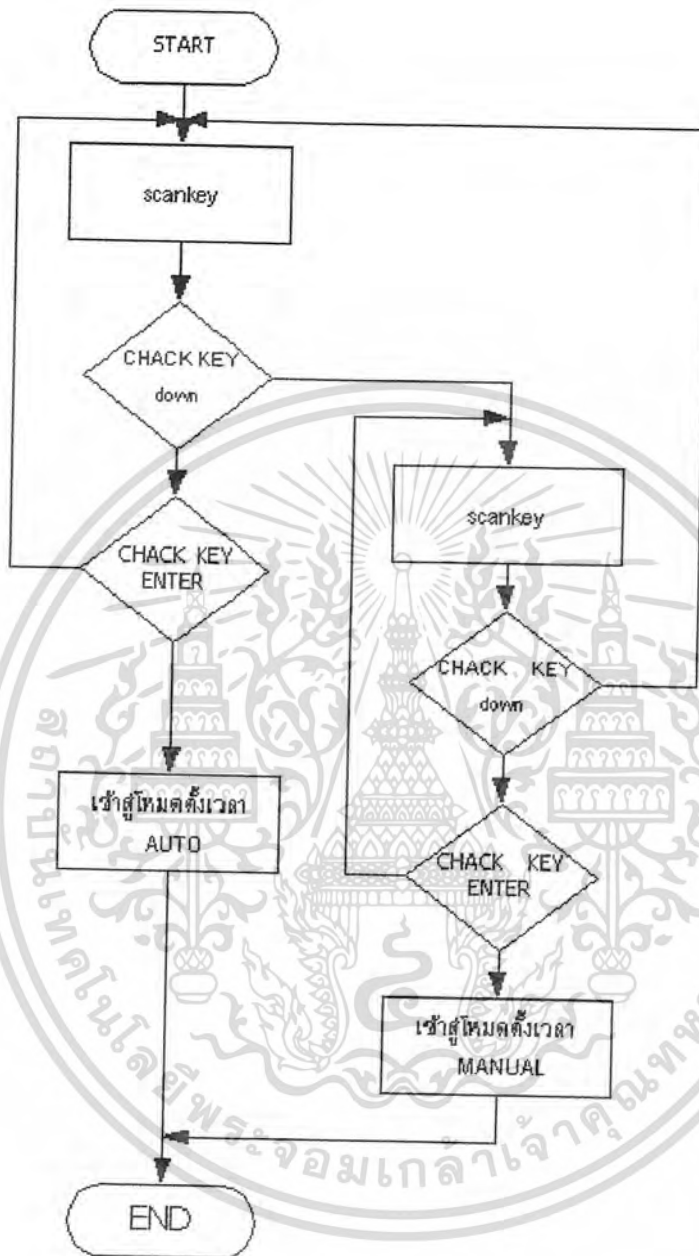
ส่วนประกอบทาง SOFTWARE คือ ส่วนของโปรแกรมควบคุมการทำงานของระบบควบคุมน้ำหยดซึ่งเขียนโปรแกรมควบคุมด้วย ไมโครคอนโทรลเลอร์ MCS 51 ด้วยภาษาแอสเซมบลี ซึ่งขั้นตอนการเขียนโปรแกรมต้องเริ่มจากการเขียน Flow Chart การทำงานของระบบก่อนเสมอ เพื่อการเรียงลำดับขั้นตอนการเขียน โปรแกรมควบคุมระบบ แยกแยะส่วนต่าง ๆ ของโปรแกรม และควรเขียนทีละ block ของ ฟอร์ดชาร์ตเป็น ฟังก์ชันไว้เรียกใช้งาน เพื่อความเป็นระเบียบเรียบร้อยของโปรแกรม

ขั้นตอนของการเขียนบล็อกไดอะแกรม มีดังนี้



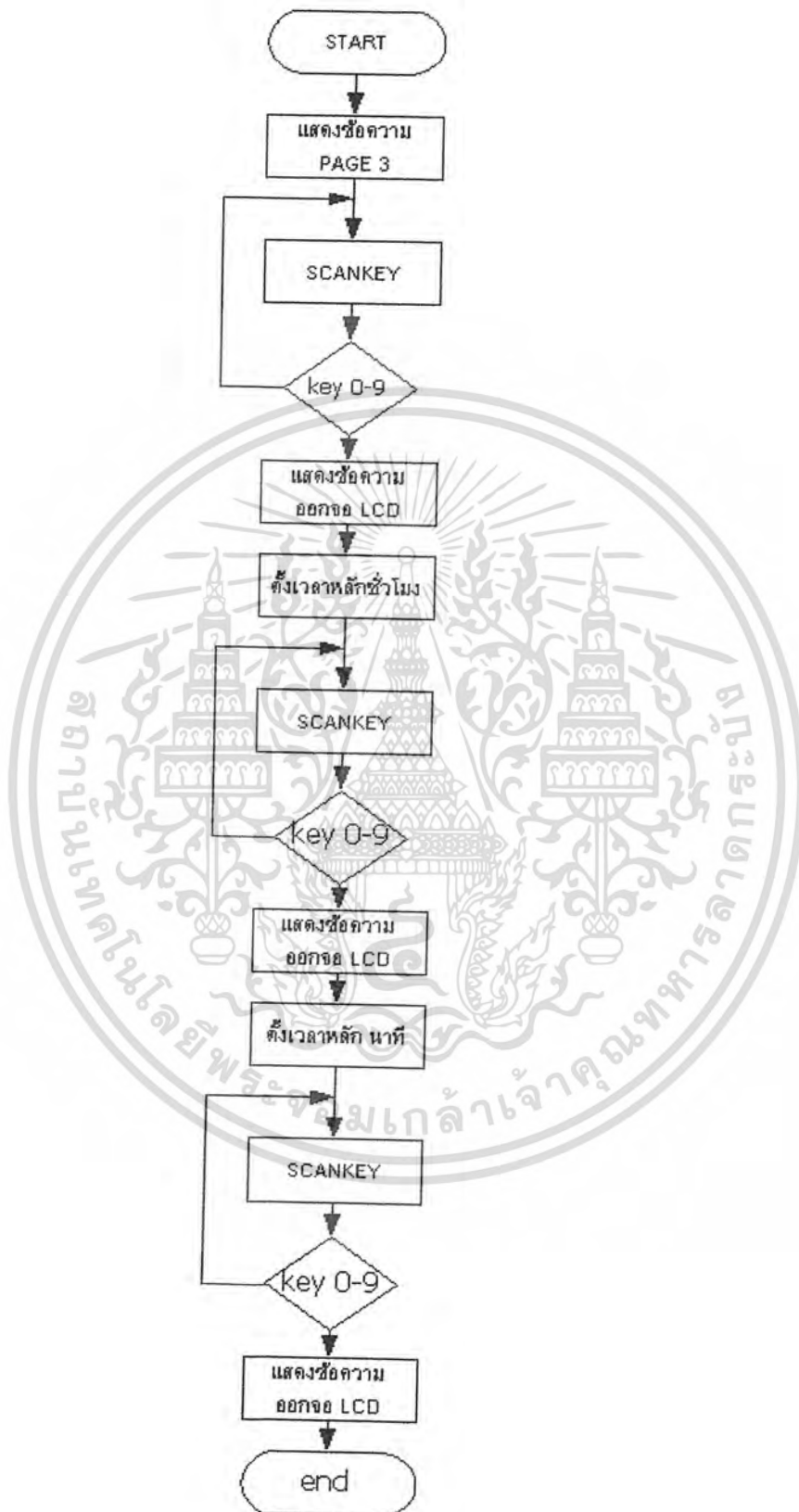
รูปที่ 5.4 บล็อกไดอะแกรมเลือกโหมดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 บล็อกไดอะแกรมโหมด MANUAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 บล็อกไดอะแกรม นาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองโครงงาน

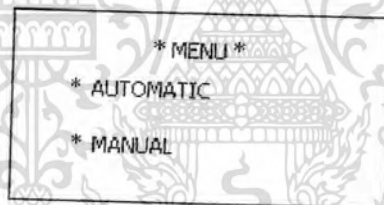
ในการทดลองโครงงานระบบน้ำหยด สรุปลงเป็นขั้นตอนการทำงานดังนี้

6.1 เมื่อทำการเปิดเครื่อง โมดูล LCD จะเป็นลักษณะดังรูปที่ 6.1



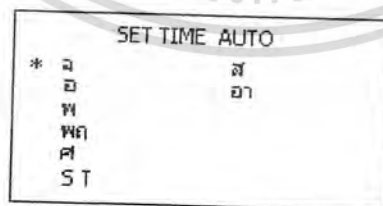
รูปที่ 6.1 ลักษณะหน้าจอ LCD ขณะเริ่มเปิด

6.2 หลังจากนั้น จะเป็นการเลือกโหมดการทำงาน



รูปที่ 6.2 การเลือกโหมดการทำงาน

6.3 ถ้าเลือกโหมด AUTOMATIC จะแสดงหน้าจอ ดังรูป



รูปที่ 6.3 การเลือกโหมดการ AUTOMATIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 ถ้าเลือกโหมด MANUAL จะแสดงหน้าจอตั้งรูป

SET TIME MANUAL	
ST	M : N
—	— : —

รูปที่ 6.4 การเลือกโหมด MANUAL

6.5 แสดงการตั้งเวลาโหมด AUTOMATIC ตามเวลาที่เรานำเลือกวัน ซึ่งการตั้งเวลาสามารถตั้งได้ 7 วัน ให้ระบบทำงานแบบอัตโนมัติ

SET TIME AUTO	
ST	M : N
* CH —	— : —
CH —	— : —
CH —	— : —
CH —	— : —
ST —	— : —
ET —	— : —

รูปที่ 6.5 ป้อนวัน เดือน ปี ที่ปลูกของพืช

หลังจากที่เซตค่าเสร็จระบบก็จะเริ่มทำงาน ถึงตอนนี้ระบบจะรู้ว่าพืชที่ใช้งานเป็น พืชชนิดใด มีอายุเท่าใด จากนั้นก็จะเริ่มนับเวลารอบเวรการให้น้ำ และรอบเวรการให้ปุ๋ย

ระบบจะทำการอ่านค่าข้อมูล ปริมาณน้ำที่ต้องการ ปริมาณปุ๋ยที่ต้องการ อัตราส่วนระหว่างปุ๋ยที่ต่างๆ ตามชนิดและอายุของพืชจาก Table จากนั้นก็ทำตาม Flow chart ไปเรื่อยๆและเมื่อครบ 1 รอบเวร ระบบก็จะนับรอบเวรต่อไปเรื่อยๆ เมื่อถึงเวลาที่ในรอบเวรต่อไป ระบบก็จะทำงานซ้ำอีกในแบบเดิม แต่ค่าข้อมูล ปริมาณปุ๋ยที่ต้องการ ปริมาณน้ำที่ต้องการ อัตราส่วนระหว่างปุ๋ย จะเปลี่ยนไปตามอายุของพืช เพราะฐานเวลาของระบบจะเดินไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลภาพผลการทดลองระบบควบคุมน้ำหยด ณงานนิทรรศน์พระจอมเกล้าลาดกระบัง

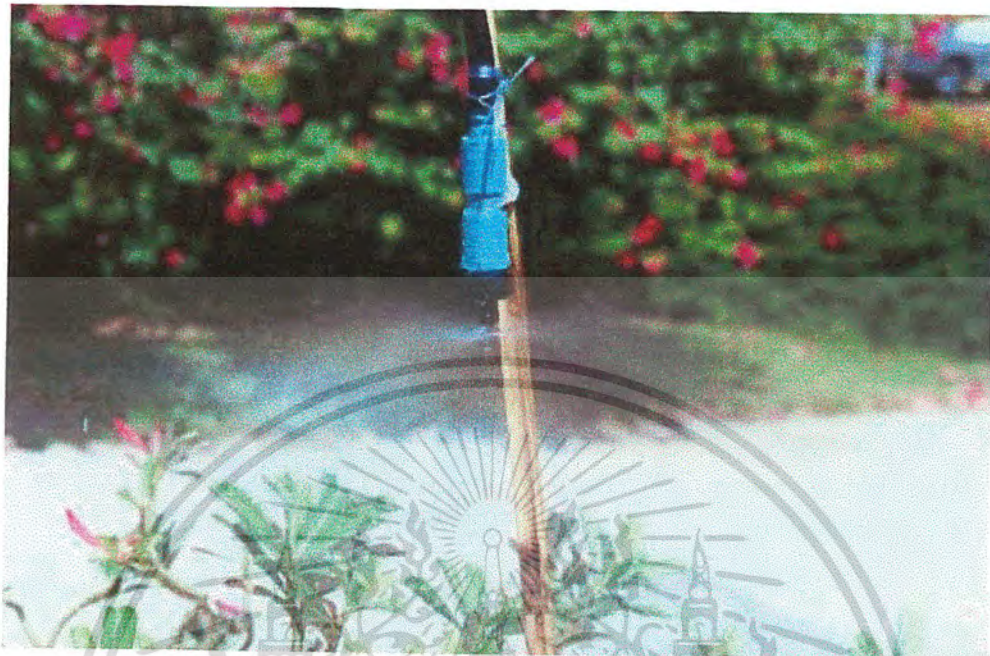


รูปที่ 6.6 ระบบควบคุมน้ำหยด

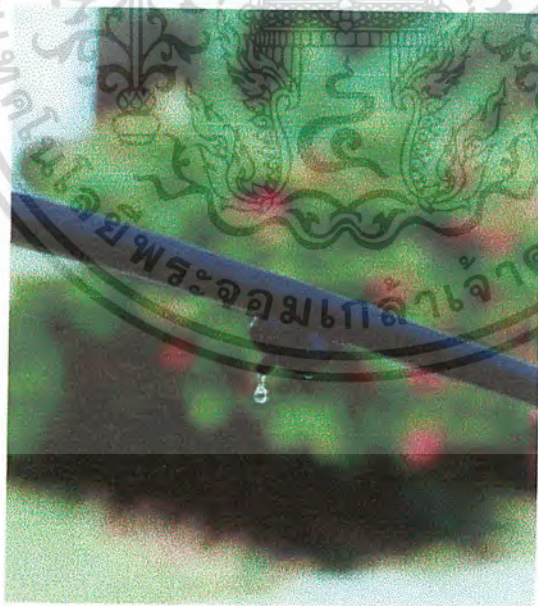


รูปที่ 6.7 การควบคุมการปล่อยน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

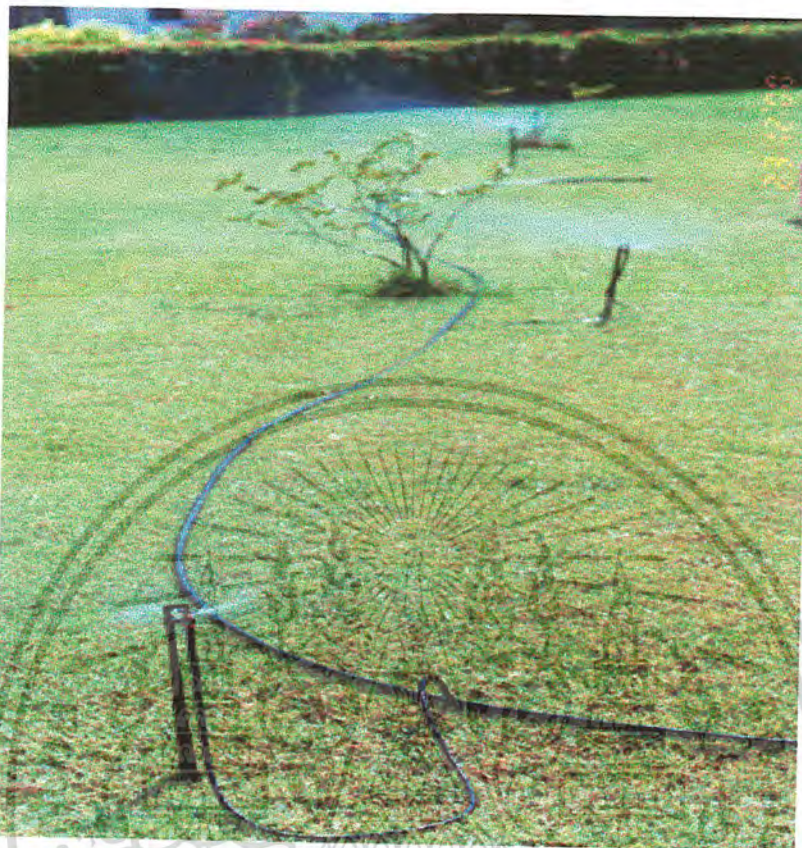


รูปที่ 6.8 หัวปล่อยน้ำแบบ Spayer



รูปที่ 6.9 หัวปล่อยน้ำแบบหยด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.10 แสดงหัวปล่อยน้ำแบบ Springer

จากผลการทดลอง การให้น้ำในสนามปรากฏว่ามีผลเป็นที่น่าในระดัหนึ่งพอใจ ระบบสามารถที่จะควบคุมการจ่ายน้ำได้ถูกต้องตามโปรแกรมที่ออกแบบไว้ แต่ระบบก็มีข้อเสยคือ ป้มน้ำที่ใช้คูน้ำไม่ค่อยได้ต้องคอยเช็คอยู่บ่อยๆ อาจเกิดจากท่อที่ต่อกวไม่ดี อาจทำให้ลมเข้าท่อทำให้แรงอัดน้ำไม่เพียงพอ และทำให้ลมย้อนอัดเข้าป้มน้ำด้วยจึงทำให้ป้มน้ำคูน้ำไม่ขึ้น ถ้าจะให้ดีระบบควรจะต้องมีเครื่องควบคุมความดันน้ำ และวาล์วสำหรับป้องกันน้ำไหลย้อนกลับ รวมทั้งที่กรองน้ำเพื่อป้องกันหัวปล่อยน้ำอุดตัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการทดลอง และแนวทางการพัฒนา

จากผลการทดลองสรุปได้ว่า เป็นไปตามเป้าหมายที่ตั้งไว้ แต่โครงการนี้ยังสามารถที่จะพัฒนาต่อไปได้อีกหลายด้าน และประยุกต์ใช้กับกระบวนการอื่นๆ เช่น ในส่วนของการผสมปุ๋ยอัดโนมิติ อาจจะนำไปประยุกต์เป็นการผสมอาหารสัตว์ เป็นต้น

ในส่วนของโครงการนี้ยังมีส่วนต่างๆ ที่จะเพิ่มเติมได้อีกเช่น เพิ่มในส่วนของการวัดความชื้นของดิน เพื่อเพิ่มประสิทธิภาพของระบบให้สูงขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- (1) สุเจตน์ จันทรัมย์ “ไมโครคอนโทรลเลอร์ซีพเดียว 8051”
โครงการตำรา วิทยาลัยมหานคร
- (2) คู่มือ ET-BOARD โดย บริษัท อีทีที จำกัด
- (3) ผศ. มนตรี คำชู “การชลประทานแบบหยด”
ภาควิชาวิศวกรรมชลประทาน คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก
ชอร์สโค้ดของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****/;
;/ PROGRAM MONITOR /;
;/ CONTROL PROGRAM /;
;/ DRAWING BY    /;
;/ ANUCHIT HAEMHOMWONG/;
;*****/;

WR_COMM1 EQU 0E0C4H
RD_COMM1 EQU 0E0C5H
WR_DATA1 EQU 0E0C6H
RD_DATA1 EQU 0E0C7H
WR_COMM2 EQU 0E0C8H
RD_COMM2 EQU 0E0C9H
WR_DATA2 EQU 0E0CAH
RD_DATA2 EQU 0E0CBH
PORTA EQU 0E0E0H ; Port A 8255
PORTB EQU 0E0E1H ; Port B 8255
PORTC EQU 0E0E2H ; Port C 8255
CONTRL EQU 0E0E3H ; Control Port 8255
CHMANUAL EQU 80H
TIMEHOUR EQU 81H
TIMEMINUTE EQU 82H
CHAU1 EQU 83H
CHAU2 EQU 84H
CHAU3 EQU 85H
CHAU4 EQU 86H
CHAU5 EQU 87H
MINUTE1 EQU 88H
MINUTE2 EQU 89H
MINUTE3 EQU 8AH
MINUTE4 EQU 8BH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MINUTE5 EQU 8CH
HOUR1 EQU 8DH
HOUR2 EQU 8FH
HOUR3 EQU 90H
HOUR4 EQU 91H
HOUR5 EQU 92H

ORG 20H

KEY_PRES: DS 1 ; FLAG_BUF.0
KEY_STAS: DS 1
X_ADDR_1: DS 1
X_ADDR_2: DS 1
Y_ADDR_1: DS 1
Y_ADDR_2: DS 1
LINE_CNT: DS 1
POST_CNT: DS 1
ROW: DS 1 ; CPU-RAM Buffer
COLUM: DS 1

ORG 0000H
MAIN: MOV R2,#02H ; Power-on Delay
PWR_DLY1: MOV R3,#00H
PWR_DLY2: MOV R4,#00H

DJNZ R4,$
DJNZ R3,PWR_DLY2
DJNZ R2,PWR_DLY1
;
MOV DPTR,#CONTRL ;INITIAL 8255
MOV A,#81H
MOVX @DPTR,A
LCALL INITGLCD ;INITIAL LCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL CLEARSC
DISPAGE1: MOV X_ADDR_1,#00H ;DISPLAY PAGE 1
MOV Y_ADDR_1,#18H
MOV DPTR,#YIN
LCALL DISP_1
MOV Y_ADDR_1,#28H
MOV DPTR,#YEI
LCALL DISP_1
MOV Y_ADDR_1,#30H
MOV DPTR,#YUO
LCALL DISP_1
MOV X_ADDR_2,#00H
MOV Y_ADDR_2,#08H
MOV DPTR,#YUB
LCALL DISP_2
MOV Y_ADDR_2,#18H
MOV DPTR,#YUOP
LCALL DISP_2
LCALL CLAEER_REG
MOV X_ADDR_1,#01H
MOV X_ADDR_2,#01H
MOV DPTR,#TABFONT
LCALL DISPL1
MOV X_ADDR_2,#02H
MOV Y_ADDR_2,#18H
MOV DPTR,#YUIO
LCALL DISP_2
MOV X_ADDR_1,#03H
MOV Y_ADDR_1,#38H
MOV DPTR,#YOIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DISP_1
LCALL CLAER_REG
MOV X_ADRR_1,#04H
MOV X_ADRR_2,#04H
MOV DPTR,#LINE4
LCALL DISPL1
MOV X_ADRR_1,#06H
MOV X_ADRR_2,#06H
MOV DPTR,#LINE6
LCALL DISPL1
MOV R2,#03H
DELPAGE1: LCALL DELAY1S
DJNZ R2,DELPAGE1
LCALL CLEARSC ;CLEAR PAGE 1
;
DISPAGE2: MOV X_ADRR_1,#01H
MOV X_ADRR_2,#01H
MOV DPTR,#MENUPAGE2 ;DISPLAY PAGE 2
LCALL DISPL1
MOV X_ADRR_1,#03H
MOV X_ADRR_2,#03H
MOV DPTR,#AUTOMATIC1
LCALL DISPL1
MOV X_ADRR_1,#05H
MOV X_ADRR_2,#05H
MOV DPTR,#MANUAL1
LCALL DISPL1
YO: LCALL SCANKEY
CJNE A,#0FH,YO1
LCALL CLEARSC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AJMP DISPAGE4
YO1:  CJNE A,#0AH,YO2
LCALL CLEARSC
AJMP DISPAGE1
YO2:  CJNE A,#0BH,YO
MOV DPTR,#NO_00
MOV X_ADDR_1,#03H
MOV Y_ADDR_1,#08H
LCALL DISP_1
MOV DPTR,#DOWN
MOV X_ADDR_1,#05H
LCALL DISP_1
YP:   LCALL SCANKEY
CJNE A,#0AH,YP1 ; UP TO PAGE 2
LCALL DELAY_200MS ; DELAY 200MSEC
LCALL CLAER_REG
SJMP DISPAGE2
YP1:  CJNE A,#0FH,YP
LCALL CLAER_REG
LCALL CLEARSC

DISPAGE3: MOV DPTR,#MODEMANUAL ; DISPLAY PAGE 3 ; SET MANUAL
LCALL DISPL1
MOV X_ADDR_1,#02H
MOV X_ADDR_2,#02H
MOV DPTR,#SETTIMER
LCALL DISPL1
MOV X_ADDR_1,#04H ; Y_ADDR_1=0
MOV X_ADDR_2,#04H ; Y_ADDR_2=0
MOV DPTR,#ST_HM
LCALL DISPL1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV X_ADDR_2,#06H
MOV Y_ADDR_2,#18H
MOV DPTR,#POINT
LCALL DISP_2
LCALL DELAY_200MS
SETMANUAL: LCALL SCANKEY      ; SET TIMER MANUAL
PUSH ACC
MOV X_ADDR_1,#06H
MOV Y_ADDR_1,#08H
LCALL RD_PAGE1
MOV DPTR,#NO_11
LCALL DISP_1
LCALL DELAY_100MS
LCALL WR_LCD1
LCALL DELAY_100MS
POP ACC
CJNE A,#0CH,CHMU1
LCALL DELAY_200MS
LCALL WR_LCD1
SJMP CHMUT
CHMU1:  CJNE A,#0AH,CHMU2
LCALL CLAER_REG
LCALL CLEARSC
AJMP DISPAGE2
CHMU2:  CJNE A,#0FFH,CHMU3
SJMP SETMANUAL
CHMU3:  LCALL SCANUMBER
MOV R1,A      ;R1=TEN
LCALL DISP_1
LCALL DELAY_200MS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHMUT:  LCALL SCANKEY
PUSH ACC      ;ADDRESS CURSOR
MOV Y_ADDR_1,#10H
LCALL RD_PAGE1
MOV DPTR,#NO_11
LCALL DISP_1
LCALL DELAY_100MS
LCALL WR_LCD1
LCALL DELAY_100MS
POP ACC
CJNE A,#0DH,CHMT1
LCALL DELAY_200MS
SJMP SETMANUAL
CHMT1:  CJNE A,#0CH,CHMT2
LCALL DELAY_200MS
SJMP HOMNEW
CHMT2:  CJNE A,#0FFH,CHMT3
SJMP CHMUT
CHMT3:  LCALL SCANUMBER
MOV R2,A
LCALL DELAY_100MS
MOV X_ADDR_1,#06H
LCALL DISP_1
LCALL DTOH
MOV R0,#CHMANUAL
MOV @R0,A
LCALL DELAY_200MS
LCALL DELAY_200MS
HOMNEW:  LCALL SCANKEY
PUSH ACC      ;ADDRESS CURSOR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV X_ADDR_2,#06H
MOV Y_ADDR_2,#10H
LCALL RD_PAGE2
MOV DPTR,#NO_11
LCALL DISP_2
LCALL DELAY_100MS
LCALL WR_LCD2
LCALL DELAY_100MS
POP ACC
CJNE A,#0DH,HOM1
LCALL DELAY_200MS
SJMP CHMUT
HOM1:  CJNE A,#0CH,HOM2
LCALL DELAY_200MS
SJMP MINUMNEW
HOM2:  CJNE A,#0FFH,HOM3
SJMP HOMNEW
HOM3:  MOV R2,A
LCALL SCANUMBER
LCALL DISP_2
MOV A,R2
MOV R0,#TIMEHOUR
MOV @R0,A
LCALL DELAY_200MS    ;SAVE IN TIMEHOUR
MINUMNEW: LCALL SCANKEY
PUSH ACC            ;ADDRESS CURSOR
MOV Y_ADDR_2,#20H
LCALL RD_PAGE2
MOV DPTR,#NO_11
LCALL DISP_2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DELAY_100MS
LCALL WR_LCD2
LCALL DELAY_100MS
POP ACC
CJNE A,#0DH,MINUM1
LCALL DELAY_200MS
SJMP HOMNEW
MINUM1: CJNE A,#0CH,MINUM2
LCALL DELAY_200MS
SJMP MINUMNEWT
MINUM2: CJNE A,#0FFH,MINUM3
SJMP MINUMNEW
MINUM3: MOV Y_ADDR_2,#20H
LCALL SCANUMBER
MOV R1,A ;COLUMN 10
LCALL DISP_2
LCALL DELAY_200MS
MINUMNEWT: LCALL SCANKEY
PUSH ACC ;ADDRESS CURSOR
MOV Y_ADDR_2,#28H
LCALL RD_PAGE2
MOV DPTR,#NO_11
LCALL DISP_2
LCALL DELAY_100MS
LCALL WR_LCD2
LCALL DELAY_100MS
POP ACC
CJNE A,#0DH,MINUMT2
LCALL DELAY_200MS
SJMP MINUMNEW

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MINUMT2:  CJNE A,#0FFH,MINUMT3
SJMP MINUMNEWT
MINUMT3:  LCALL SCANUMBER
MOV R2,A
MOV Y_ADDR_2,#28H
LCALL DISP_2
LCALL DTOH
MOV R0,#TIMEMINUTE
MOV @R0,A
LCALL DELAY_200MS
MINUMT4:  LCALL SCANKEY
CJNE A,#0DH,MINUMT5
LCALL DELAY_200MS
SJMP MINUMNEWT
MINUMT5:  CJNE A,#0FH,MINUMT4
LJMP PRO_MANUAL
DISPAGE4: MOV DPTR,#SET_AUTO ;DISPLAY PAGE SET AUTO
LCALL CLAER_REG
LCALL DISPL1
MOV X_ADDR_1,#01H
MOV Y_ADDR_1,#28H
MOV DPTR,#S
LCALL DISP_1
MOV Y_ADDR_1,#30H
MOV DPTR,#T
LCALL DISP_1
MOV X_ADDR_2,#01H
MOV Y_ADDR_2,#10H
MOV DPTR,#H
LCALL DISP_2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV Y_ADDR_2,#18H
MOV DPTR,#POINT
LCALL DISP_2
MOV Y_ADDR_2,#20H
MOV DPTR,#M
LCALL DISP_2
MOV X_ADDR_1,#02H
MOV Y_ADDR_1,#00H
MOV DPTR,#POINTER
LCALL DISP_1
MOV Y_ADDR_1,#08H
MOV DPTR,#NO_1
LCALL DISP_1
MOV A,#02H
ACALL CH
MOV X_ADDR_1,#03H
MOV DPTR,#NO_2
LCALL DISP_1
MOV A,#03H
ACALL CH
MOV X_ADDR_1,#04H
MOV DPTR,#NO_3
LCALL DISP_1
MOV A,#04H
ACALL CH
MOV X_ADDR_1,#05H
MOV DPTR,#NO_4
LCALL DISP_1
MOV A,#05H
ACALL CH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV X_ADDR_1,#06H
MOV DPTR,#NO_5
LCALL DISP_1
MOV A,#06H
ACALL CH
LJMP SETAUTO
;*****/;
; DISPLAY CH /;
;*****/;
CH: MOV X_ADDR_1,A
MOV Y_ADDR_1,#10H
MOV DPTR,#CF
PUSH ACC
LCALL DISP_1
POP ACC
MOV X_ADDR_1,A
MOV Y_ADDR_1,#18H
MOV DPTR,#HF
LCALL DISP_1
MOV Y_ADDR_1,#08H
RET
;/ CLAER REGIGTER /;
;*****/;
CLAER_REG: CLR A
MOV X_ADDR_1,A
MOV X_ADDR_2,A
MOV Y_ADDR_1,A
MOV Y_ADDR_2,A
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;**** PROGRAM SELECT DAY *****
SELECTDAY: MOV DATE,#01H
NEW_SEDAY: LCALL SCANKEY
            CJNE A,#0AH,NEXTDAY1 ; CHECK KEY UP
;***** CHANGE DATE *****
            LCALL DELAY_200MS
            DEC DATE
            LCALL SHIFTCUR      ; GOTU SHIFT CURSOR
            MOV A,DATE
            CJNE A,#00H,NEW_SEDAY
            LCALL CLEARSC
            LJMP DISPAGE2
;***** END CHANGE DATE *****
NEXTDAY1:  CJNE A,#0BH,NEXTDAY2 ; check key down
;***** CHANGE DATE *****
            LCALL DELAY_200MS
            INC DATE
            LCALL SHIFTCUR      ; GOTU SHIFT CURSOR
            MOV A,DATE
            CJNE A,#09H,NEXTDAY2
            MOV DATE,#01H
            LCALL SHIFTCUR      ; GOTU SHIFT CURSOR
            SJMP NEW_SEDAY
;***** END CHANGE DATE *****
NEXTDAY2:  CJNE A,#0FH,NEW_SEDAY ; CHECK KEY ENTER
            LCALL READEE      ;EDIT
            LCALL DELAY_200MS ; NO WORK
            LCALL CLEARSC
            LCALL DISPAGE5    ; DISPLAY DATA TO SCREEN MONITOR
            LJMP SETAUTO      ; STOP PROGRAM SELECT DAY
            LCALL WRITEE
            INC DATE          ; NEXT DAY
            AJMP NEW_SEDAY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****
;
;***  DISPLAY & SHIFT CURSOR  *****
;
;**   TO ADREESS DATE
;
*****
;

```

```
SHIFTCUR:  MOV A,DATE
```

```
          CJNE A,#01H,SHIFT1
```

```
          MOV X_ADRR_1,#04H
```

```
          MOV Y_ADRR_1,#08H
```

```
          MOV DPTR,#NO_00
```

```
          LCALL DISP_1
```

```
          MOV X_ADRR_1,#07H
```

```
          MOV Y_ADRR_1,#08H
```

```
          MOV DPTR,#NO_00
```

```
          LCALL DISP_1
```

```
          MOV X_ADRR_1,#03H
```

```
          MOV Y_ADRR_1,#08H
```

```
          MOV DPTR,#DOWN
```

```
          LCALL DISP_1
```

```
          AJMP STOPSHIFT
```

```
SHIFT1:  CJNE A,#02H,SHIFT2
```

```
          MOV R7,#02H
```

```
          MOV X_ADRR_1,#05H
```

```
CLRSC2:  MOV Y_ADRR_1,#08H
```

```
          MOV DPTR,#NO_00
```

```
          LCALL DISP_1
```

```
          MOV X_ADRR_1,#03H
```

```
          DJNZ R7,CLRSC2
```

```
          MOV X_ADRR_1,#04H
```

```
          MOV DPTR,#DOWN
```

```
          LCALL DISP_1
```

```
          AJMP STOPSHIFT
```

```
SHIFT2:  CJNE A,#03H,SHIFT3
```

```
          MOV R7,#02H
```

```
          MOV X_ADRR_1,#04H
```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLRSC3:  MOV Y_ADRR_1,#08H
```

```
        MOV DPTR,#NO_00
```

```
        LCALL DISP_1
```

```
        MOV X_ADRR_1,#06H
```

```
        DJNZ R7,CLRSC3
```

```
        MOV X_ADRR_1,#05H
```

```
        MOV DPTR,#DOWN
```

```
        LCALL DISP_1
```

```
        AJMP STOPSHIFT
```

```
SHIFT3:  CJNE A,#04H,SHIFT4
```

```
        MOV X_ADRR_1,#05H
```

```
        MOV Y_ADRR_1,#08H
```

```
        MOV DPTR,#NO_00
```

```
        LCALL DISP_1
```

```
        MOV X_ADRR_2,#03H
```

```
        MOV Y_ADRR_2,#10H
```

```
        MOV DPTR,#NO_00
```

```
        LCALL DISP_2
```

```
        MOV X_ADRR_1,#06H
```

```
        MOV DPTR,#DOWN
```

```
        LCALL DISP_1
```

```
        AJMP STOPSHIFT
```

```
SHIFT4:  CJNE A,#05H,SHIFT5
```

```
        MOV X_ADRR_1,#06H
```

```
        MOV Y_ADRR_1,#08H
```

```
        MOV DPTR,#NO_00
```

```
        LCALL DISP_1
```

```
        MOV DPTR,#NO_00
```

```
        MOV X_ADRR_2,#04H
```

```
        MOV Y_ADRR_2,#10H
```

```
        LCALL DISP_2
```

```
        MOV X_ADRR_2,#03H
```

```
        MOV DPTR,#DOWN
```

```
        LCALL DISP_2
```

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AJMP STOPSHIFT
SHIFT5:  CJNE A,#06H,SHIFT6
        MOV R7,#02H
        MOV X_ADRR_2,#05H
        MOV Y_ADRR_2,#10H
CLRSC5:  MOV DPTR,#NO_00
        LCALL DISP_2
        MOV X_ADRR_2,#03H
        DJNZ R7,CLRSC5
        MOV X_ADRR_2,#04H
        MOV DPTR,#DOWN
        LCALL DISP_2
        SJMP STOPSHIFT
SHIFT6:  CJNE A,#07H,SHIFT7
        MOV X_ADRR_1,#07H
        MOV Y_ADRR_1,#08H
        MOV DPTR,#NO_00
        LCALL DISP_1
        MOV X_ADRR_2,#04H
        MOV Y_ADRR_2,#10H
        MOV DPTR,#NO_00
        LCALL DISP_2
        MOV X_ADRR_2,#05H
        MOV DPTR,#DOWN
        LCALL DISP_2
        SJMP STOPSHIFT
SHIFT7:  CJNE A,#08H,STOPSHIFT
        MOV X_ADRR_2,#05H
        MOV Y_ADRR_2,#10H
        MOV DPTR,#NO_00
        LCALL DISP_2
        MOV X_ADRR_1,#07H
        MOV Y_ADRR_1,#08H
        MOV DPTR,#DOWN

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DISP_1
STOPSHIFT: RET
DISPAGE5: MOV DPTR,#SET_AUTO ;DISPLAY PAGE SET UINTAUTO
LCALL CLAER_REG
LCALL DISPL1
MOV X_ADRR_1,#01H
MOV Y_ADRR_1,#28H
MOV DPTR,#S
LCALL DISP_1
MOV Y_ADRR_1,#30H
MOV DPTR,#TW
LCALL DISP_1
MOV X_ADRR_2,#01H
MOV Y_ADRR_2,#10H
MOV DPTR,#H
LCALL DISP_2
MOV Y_ADRR_2,#18H
MOV DPTR,#POINT
LCALL DISP_2
MOV Y_ADRR_2,#20H
MOV DPTR,#M
LCALL DISP_2
MOV X_ADRR_1,#02H
MOV Y_ADRR_1,#00H
MOV DPTR,#POINTER
LCALL DISP_1
MOV Y_ADRR_1,#08H
MOV DPTR,#NO_1
LCALL DISP_1
MOV A,#02H
ACALL SC_CH
MOV X_ADRR_1,#03H
MOV DPTR,#NO_2
LCALL DISP_1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#03H
ACALL SC_CH
MOV X_ADRR_1,#04H
MOV DPTR,#NO_3
LCALL DISP_1
MOV A,#04H
ACALL SC_CH
MOV X_ADRR_1,#05H
MOV DPTR,#NO_4
LCALL DISP_1
MOV A,#05H
ACALL SC_CH
MOV X_ADRR_1,#06H
MOV DPTR,#NO_5
LCALL DISP_1
MOV A,#06H
ACALL SC_CH
MOV R3,#05H
MOV R4,#02H
NEWPOINT: MOV X_ADRR_2,R4
MOV Y_ADRR_2,#18H
MOV DPTR,#POINT
LCALL DISP_2
INC R4
DJNZ R3,NEWPOINT
MOV DPTR,#S
MOV X_ADRR_1,#07H
MOV Y_ADRR_1,#08H
LCALL DISP_1
MOV DPTR,#TW
MOV Y_ADRR_1,#10H
LCALL DISP_1
MOV DPTR,#POINT
MOV Y_ADRR_1,#30H

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL DISP_1
MOV DPTR,#E
MOV X_ADRR_2,#07H
MOV Y_ADRR_2,#08H
LCALL DISP_2
MOV DPTR,#TW
MOV Y_ADRR_2,#10H
LCALL DISP_2
MOV DPTR,#POINT
MOV Y_ADRR_2,#28H
LCALL DISP_2
SJMP SELECTCH
;*****/
; DISPLAY CH /;
;*****/
SC_CH: MOV X_ADRR_1,A
MOV Y_ADRR_1,#10H
MOV DPTR,#CF
PUSH ACC
LCALL DISP_1
POP ACC
MOV X_ADRR_1,A
MOV Y_ADRR_1,#18H
MOV DPTR,#HF
LCALL DISP_1
MOV Y_ADRR_1,#08H
RET

;*****/
;/ DELAY 1 MINUTE /;
;/
;*****/
DELAY_1M: MOV R4,#3CH
DELAY_1S: MOV R5,#64H

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DELAY_10MS:  MOV R7,#0AH
DELAY_10MS_1: MOV R6,#0E6H
DELAY_10MS_2: NOP
                NOP
                DJNZ R6,DELAY_10MS_2
                DJNZ R7,DELAY_10MS_1
                DJNZ R5,DELAY_10MS
                DJNZ R4,DELAY_1S
                RET

```

```

;*****;
;/ DELAY 1 SEC /;
;/           /;
;*****;

```

```

DELAY1S:  MOV R5,#64H
DELAY10MS:  MOV R7,#0AH
DELAY10MS_1: MOV R6,#0E6H
DELAY10MS_2: NOP
                NOP
                DJNZ R6,DELAY10MS_2
                DJNZ R7,DELAY10MS_1
                DJNZ R5,DELAY10MS
                RET

```

```

;*****;
;/ DELAY 200MS /;
;*****;

```

```

DELAY_200MS:  MOV R7,#0C8H
DELAY_200MS_1: MOV R6,#0E6H
DELAY_200MS_2: NOP
                NOP
                DJNZ R6,DELAY_200MS_2
                DJNZ R7,DELAY_200MS_1
                RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;/*****/;
```

```
;/ DELAY 100MS /;
```

```
;/*****/;
```

```
DELAY_100MS: MOV R7,#64H
```

```
DELAY_100MS_1: MOV R6,#0E6H
```

```
DELAY_100MS_2: NOP
```

```
    NOP
```

```
    DJNZ R6,DELAY_100MS_2
```

```
    DJNZ R7,DELAY_100MS_1
```

```
    RET
```

```
;/*****/;
```

```
;/* Table of Key Code */;
```

```
;/*****/;
```

```
;
```

```
KEY_TAB:  DB  0D7H,0EEH,0DEH,0BEH,0EDH,0DDH,0BDH,0EBH ; 01234567
```

```
          DB  0DBH,0BBH,07EH,07DH,07BH,0B7H,0E7H,077H ; 89UDRLC E
```

```
YIN:     DB  000H,030H,028H,024H,024H,028H,030H,000H
```

```
YEI:     DB  000H,030H,028H,024H,024H,028H,03EH,000H
```

```
YUO:     DB  000H,014H,01EH,014H,008H,004H,000H,000H
```

```
YUB:     DB  018H,038H,038H,020H,020H,010H,008H,004H
```

```
YUOP:    DB  000H,000H,000H,03CH,03CH,000H,000H,000H
```

```
TABFONT: DB  000H,000H,000H,000H,000H,000H,000H,000H ;LINE1
```

```
          DB  000H,000H,000H,000H,000H,000H,000H,000H
```

```
          DB  000H,000H,000H,000H,000H,000H,000H,000H
```

```
          DB  000H,06EH,095H,092H,080H,080H,0FFH,000H
```

```
          DB  002H,0F5H,08EH,040H,020H,0FFH,0A0H,040H
```

```
          DB  0FEH,081H,049H,035H,019H,001H,0FEH,000H
```

```
          DB  0FEH,081H,049H,036H,019H,001H,0FEH,000H
```

```
          DB  000H,072H,0A9H,091H,081H,081H,07EH,000H
```

```
          DB  002H,0F5H,08EH,040H,020H,0FEH,0A0H,040H
```

```
          DB  000H,002H,005H,005H,045H,0A5H,07AH,001H
```

```
          DB  002H,085H,0FEH,080H,080H,080H,0FEH,000H
```

```
          DB  0E4H,0D2H,00AH,00AH,012H,024H,0FAH,001H
```

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

YUIO: DB 000H,006H,01EH,020H,01EH,000H,000H,000H

YOIN: DB 000H,001H,027H,054H,054H,022H,001H,000H

LINE4: DB 000H,000H,000H,000H,000H,000H,000H,000H ;LINE4

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,002H,005H,005H,045H,0A5H,07AH,001H

DB 000H,066H,066H,044H,044H,044H,044H,022H

DB 002H,085H,0FFH,080H,080H,080H,0FFH,000H

DB 002H,085H,0FFH,080H,080H,080H,0FFH,000H

DB 002H,0F5H,08EH,040H,020H,0FFH,0A0H,040H

DB 000H,002H,001H,001H,001H,001H,0FEH,000H

DB 002H,0FDH,026H,010H,00AH,00DH,0FAH,000H

DB 000H,0FEH,095H,092H,080H,080H,0FFH,000H

DB 0FEH,081H,049H,035H,019H,001H,0FEH,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

DB 000H,000H,000H,000H,000H,000H,000H,000H

LINE6: DB 03FH,0C0H,020H,010H,020H,0C0H,03FH,000H ;W LINE 6

DB 0F8H,014H,012H,011H,012H,014H,0F8H,000H ;A

DB 003H,001H,081H,0FFH,081H,001H,003H,000H ;T

DB 081H,0FFH,089H,089H,09DH,081H,0E3H,000H ;E

DB 081H,0FFH,089H,019H,029H,0C9H,086H,080H ;R

DB 000H,000H,000H,000H,000H,000H,000H,000H ;

DB 081H,0FFH,081H,081H,081H,042H,03CH,000H ;D

DB 081H,0FFH,089H,019H,029H,0C9H,086H,080H ;R

DB 000H,07EH,081H,081H,081H,081H,07EH,000H ;O

DB 000H,081H,0FEH,091H,011H,011H,00EH,000H ;P

DB 000H,0E6H,049H,089H,091H,092H,067H,000H ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ST_HM: DB 000H,000H,000H,000H,000H,000H,000H,000H ;LINE5
 DB 000H,0E6H,049H,089H,091H,092H,067H,000H ;S
 DB 003H,001H,081H,0FFH,081H,001H,003H,000H ;T
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H ;H
 DB 008H,0FFH,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,06CH,06CH,000H,000H,000H ;:
 DB 000H,000H,000H,000H,000H,000H,000H,000H ;M
 DB 018H,004H,002H,0FFH,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 SET_AUTO: DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 000H,0E6H,049H,089H,091H,092H,067H,000H ;S
 DB 081H,0FFH,089H,089H,09DH,081H,0E3H,000H ;E
 DB 003H,001H,081H,0FFH,081H,001H,003H,000H ;T
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 003H,001H,081H,0FFH,081H,001H,003H,000H ;T
 DB 000H,000H,081H,0FFH,081H,000H,000H,000H ;I
 DB 0FFH,002H,004H,018H,004H,002H,0FFH,000H ;M
 DB 081H,0FFH,089H,089H,09DH,081H,0E3H,000H ;E
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 DB 0F8H,014H,012H,011H,012H,014H,0F8H,000H ;A
 DB 001H,07EH,081H,080H,080H,081H,07EH,001H ;U
 DB 003H,001H,081H,0FFH,081H,001H,003H,000H ;T
 DB 000H,07EH,081H,081H,081H,081H,07EH,000H ;O
 DB 000H,000H,000H,000H,000H,000H,000H,000H
 S: DB 000H,0E6H,049H,089H,091H,092H,067H,000H ;S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TW: DB 003H,001H,081H,0FFH,081H,001H,003H,000H ;T
 HF: DB 000H,000H,07EH,010H,010H,07EH,000H,000H ;H
 H: DB 000H,0FFH,008H,008H,008H,008H,0FFH,000H ;H
 M: DB 0FFH,002H,004H,018H,004H,002H,0FFH,000H ;M
 CF: DB 000H,03CH,042H,042H,042H,024H,000H,000H ;C
 E: DB 081H,0FFH,089H,089H,09DH,081H,0E3H,000H
 A_A: DB 0F8H,014H,012H,011H,012H,014H,0F8H,000H ;A
 SR: DB 000H,0E6H,049H,089H,091H,092H,067H,000H ;S
 U: DB 001H,07FH,081H,080H,080H,081H,07FH,001H
 I: DB 000H,082H,082H,0FEH,082H,082H,000H,000H ;I
 N: DB 081H,0FFH,083H,00CH,018H,061H,0FFH,001H ;N
 O: DB 000H,07EH,081H,081H,081H,081H,07EH,000H ;O
 D: DB 081H,0FFH,081H,081H,081H,042H,03CH,000H ;D
 HK: DB 000H,0FEH,010H,010H,010H,010H,0FEH,000H ;H
 C_C: DB 000H,03CH,042H,081H,081H,081H,042H,000H ;C
 OOANG: DB 000H,064H,09AH,09AH,082H,082H,07CH,000H ;OANG
 ARE: DB 000H,004H,002H,002H,002H,0FCH,000H,000H ;ARE
 JJAN: DB 000H,004H,01AH,01AH,022H,042H,0FCH,000H ;JJAN
 PPAN: DB 000H,006H,0FEH,040H,03CH,040H,0FEH,000H ;PPAN
 ROREA: DB 000H,040H,03CH,032H,002H,002H,0FCH,000H ;ROREA
 SOSARA: DB 038H,0C4H,022H,032H,032H,004H,0FEH,001H ;sala
 SMALL_O: DB 000H,070H,088H,088H,088H,070H,000H,000H ;SMALL O
 SMALL_N: DB 000H,080H,0F0H,008H,008H,0F0H,000H,000H ;SMALL N
 SMALL_F: DB 000H,020H,0F8H,024H,004H,008H,000H,000H ;SMALL F
 TORTOA: DB 07AH,0A5H,041H,001H,001H,001H,0FEH,000H
 END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

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

Description

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

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

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



**8-Bit
Microcontroller
with 8K Bytes
Flash**

AT89S8252

0401D-A-12/97

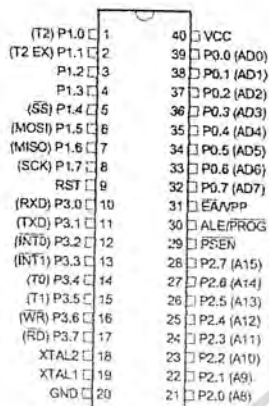


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

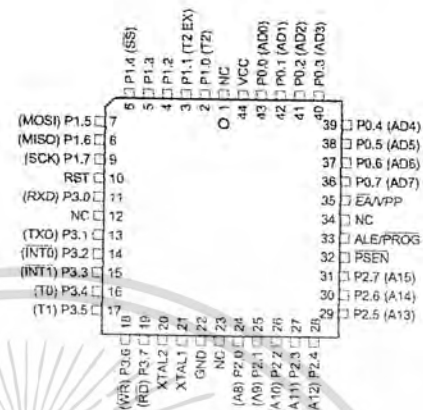


Pin Configurations

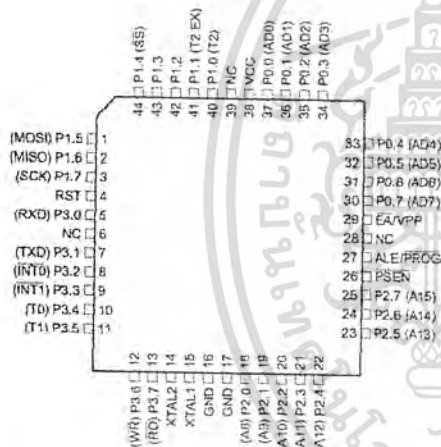
PDIP



PLCC



PQFP/TQFP



Pin Description

V_{CC}
Supply voltage.

GND
Ground.

Port 0

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

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

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

Port 1

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

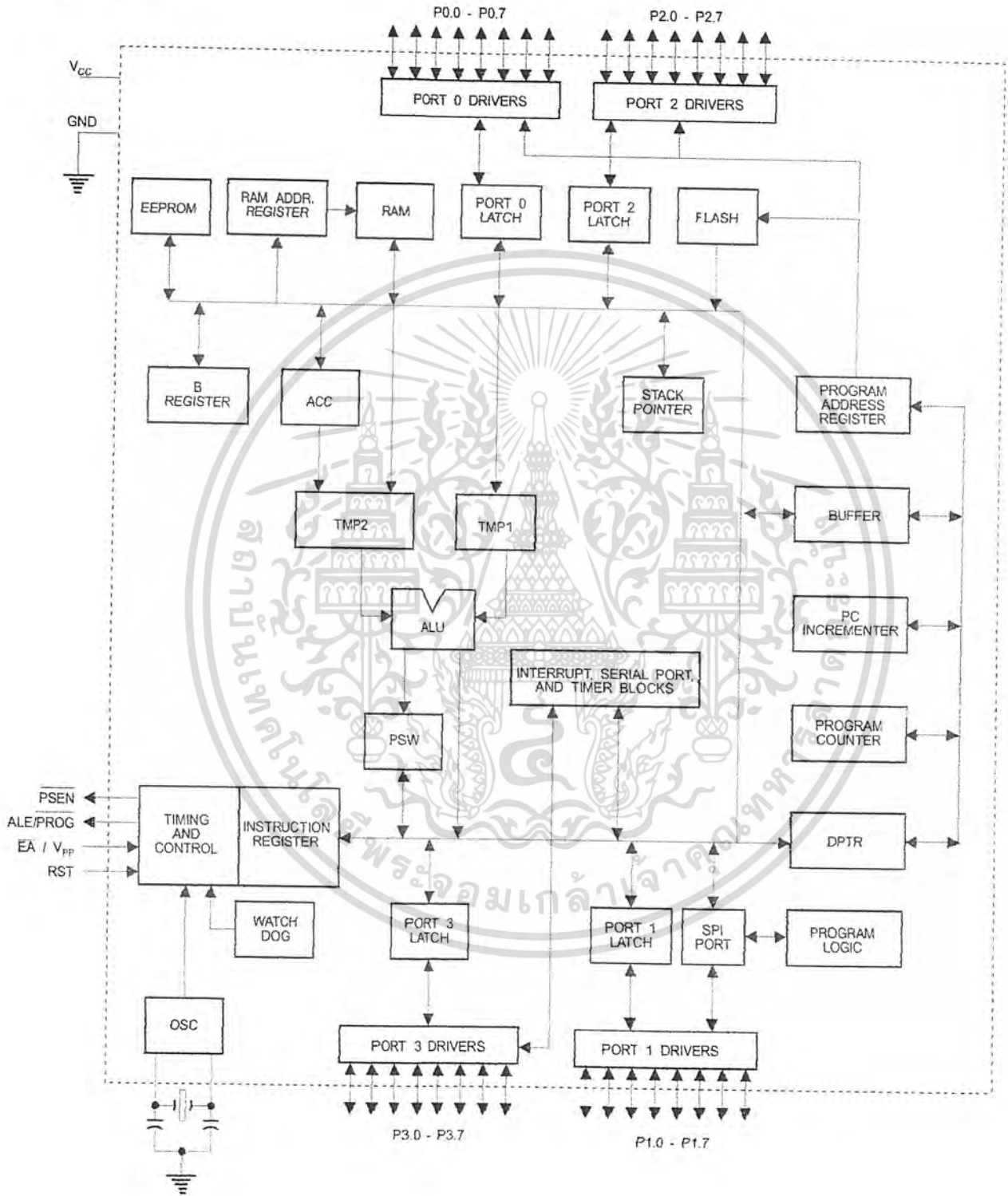
Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

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

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

Port 2

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

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

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

Port 3

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

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

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

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

RST

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

ALE/PROG

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (\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 pulse is skipped during each access to external data memory.

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

PSEN

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

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

EA/V_{PP}

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

\overline{EA} should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

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

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

Table 1. AT89S8252 SFR Map and Reset Values

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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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

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

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

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

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

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H		Reset Value = 0000 0000B						
Bit Addressable								
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T $\bar{2}$	CP/RL $\bar{2}$
Bit	7	6	5	4	3	2	1	0

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

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

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

appropriate value before accessing the respective Data Pointer Register.

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

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H									Reset Value = 0000 0010B
	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN	
Bit	7	6	5	4	3	2	1	0	
Symbol	Function								
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.								
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.								
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.								
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.								
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.								
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Table 4. SPCR—SPI Control Register

SPCR Address = D5H		Reset Value = 0000 01XXB							
Bit	7	6	5	4	3	2	1	0	
Symbol	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	
Function	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.	SPI Enable. SPI = 1 enables the SPI channel and connects SS, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.	Data Order. DORD = 1 selects LSB first data transmission, DORD = 0 selects MSB first data transmission.	Master/Slave Select. MSTR = 1 selects Master SPI mode, MSTR = 0 selects Slave SPI mode.	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.	SPI Clock Rate. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency F_{osc} is as follows: $SCK = F_{osc} \text{ divided by } \frac{2^{SPR1} + 2^{SPR0}}{4}$		

Table 5. SPSR—SPI Status Register

SPSR Address = 4AH		Reset Value = 00XX XXXXB					
Bit	7	6	5	3	2	1	0
Symbol	SPIF	WCOL					
Function	SPIF: SPI Interrupt Flag. When a slave transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.	WCOL: Write Collision Flag. This WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPSR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.					

AT91S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 6. SPDR—SPI Data Register

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

Data Memory—EEPROM and RAM

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

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

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

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

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

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

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

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

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

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

Programmable Watchdog Timer

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

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

Table 7. Watchdog Timer Period Selection

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



Timer 0 and 1

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

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit *C/T2* in the SFR *T2CON* (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in *T2CON*, as shown in Table 8. Timer 2 consists of two 8-bit registers, *TH2* and *TL2*. In the timer function, the *TL2* register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

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

Figure 1. Timer 2 in Capture Mode



Table 8. Timer 2 Operating Modes

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

Capture Mode

In the capture mode, two options are selected by bit *EXEN2* in *T2CON*. If *EXEN2* = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit *TF2* in *T2CON*. This bit can then be used to generate an interrupt. If *EXEN2* = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input *T2EX* also causes the current value in *TH2* and *TL2* to be captured into *KCAP2H* and *KCAP2L*, respectively. In addition, the transition at *T2EX* causes bit *EXF2* in *T2CON* to be set. The *EXF2* bit, like *TF2*, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-Reload (Up or Down Counter)

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

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

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

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

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

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

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

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/TF = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/32 the oscillator frequency). The baud rate formula is given below:

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

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

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

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

Figure 5. Timer 2 in Clock-Out Mode



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

The Slave Select input, $\overline{SS}/P1.4$, is set low to select an individual SPI device as a slave. When $\overline{SS}/P1.4$ is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

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

Figure 7. SPI Master-Slave Interconnection

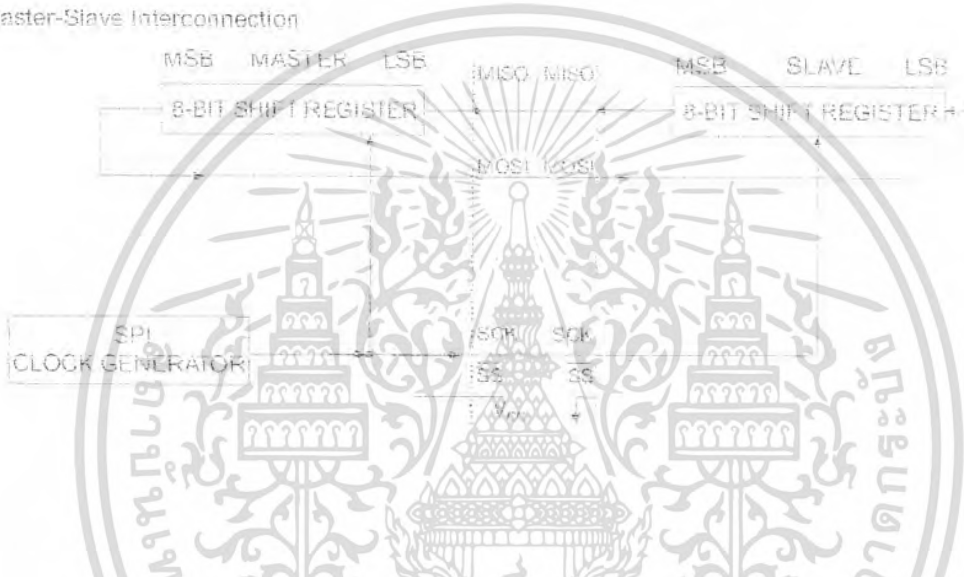


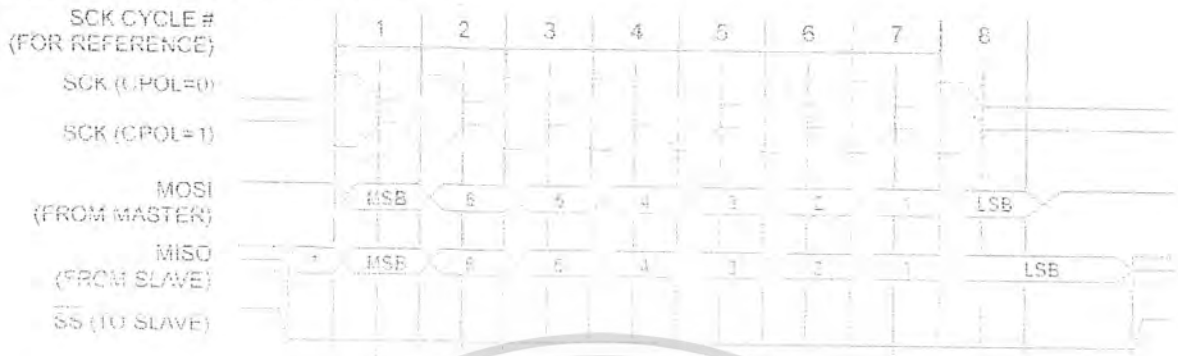
Figure 8. SPI transfer Format with CPHA = 0



*Not defined but normally MSB of character just received



Figure 9. SPI Transfer Format with CPHA = 1



*Not defined but normally LSB of previous transmitted character

Interrupts

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

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

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

Timer 2 interrupt is generated by the logical OR of bits TF2 and LYF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or LYF2 that generated the interrupt, and that bit will have to be cleared in software.

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

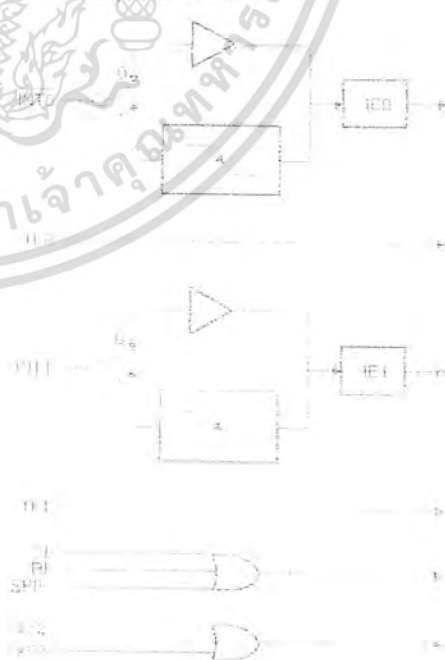
Table 10. Interrupt Enable (IE) Register

(MSB)	EA	ET2	ES	ET1	EX1	ET0	EX0	(LSB)
	1	0	1	0	1	0	1	0
	Enable Bit = 1 enables the interrupt							
	Enable Bit = 0 disables the interrupt							

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

*These bits were shown to never write 1s to unimplemented bits, because they may be used in future AT89 products.

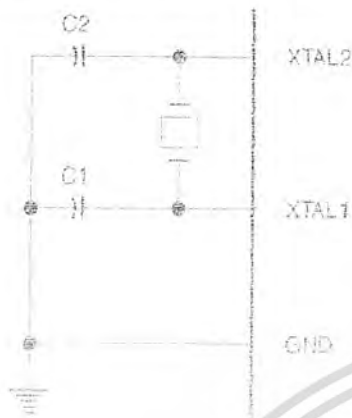
Figure 10. Interrupt Sources



AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 11. Oscillator Connections



Note: Note: C1, C2 = 30 pF = 1 µF for Crystals
= 10 pF = 10 nF for Ceramic Resonators

Oscillator Characteristics

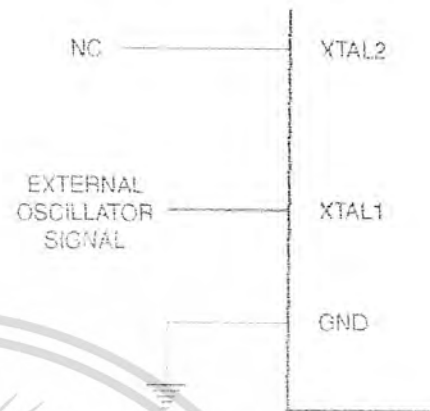
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

Idle Mode

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

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off up to two machine cycles before the

Figure 12. External Clock Drive Configuration



internal reset algorithm takes control. On-chip hardware permits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power Down Mode

In the power down mode, the oscillator is stopped and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. Exit from power down can be initiated either by a hardware reset or by an enabled external interrupt. Reset restores the SFRs but does not change the on-chip RAM.

The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power down via an interrupt, the external interrupt must be enabled as level sensitive before entering power down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

Status of External Pins During Idle and Power Down Modes

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Program Memory Lock Bits

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

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

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

Lock Bit Protection Modes⁽¹⁾⁽²⁾

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

Notes: 1 U = Unprogrammed
2 P = Programmed

Programming the Flash and EEPROM

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

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a high-voltage (12V) Parallel programming mode and a low-voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EEPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space, 0000H to 1FFFH for the Code array and 2000H to 2FFFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory located in the serial programming mode unless any of the lock bits have been programmed.

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

Parallel Programming Algorithm

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

1. Power-up sequence.
 - Apply power between V_{CC} and GND pins.
 - Set RST pin to "H".
 - Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Set PSEB pin to "L".
 - ATL pin to "H".
 - \overline{EA} pin to "H" and all other pins to "H".
3. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
 - Apply data to pins P0.0 to P0.7 for Write/Code operation.
5. Raise V_{AVDD} to 12V to enable Flash programming, erase or verification.
6. Pull up \overline{LE} pin once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte write cycle is self-timed and typically takes 1.5 μ s.
7. Invert the byte just programmed using pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางบริษัทฯ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence.
 - Set XTAL1 to "L".
 - Set RST and EA pins to "L".
 - Turn V_{CC} power off.

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

DATA Polling

The AT89S8252 features DATA Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. DATA Polling may begin any time after a write cycle has been initiated.

Ready/Busy

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

Program Verify

If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

Chip Erase

Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 10 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

Serial Programming Fuse

A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT89S8252 is shipped with the Serial Programming Mode enabled.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, 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) = 72H indicates 89S8252

Programming Interface

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

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

Serial Downloading

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

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

The Code and Data memory arrays have separate address spaces.

0000H to 1FFFFH for Code memory and 000H to 7FFFH for Data memory.

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



Serial Programming Algorithm

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

1. Power-up sequence

Apply power between V_{CC} and GND pins

Set RST pin to "H"

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

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

3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is

written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.

5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed).

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L"

Turn V_{CC} power off

Serial Programming Instruction

The instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high
Chip Erase	1010 1100	xxxx 0100	xxxx xxxx	Chip erase both 2K & 2K memory arrays.
Read Code Memory	xxxx 0011	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	xxxx 0110	low addr	xxxx xx	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	0010 0110	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	0010 0110	low addr	xxxx xx	Write data to Data memory location at selected address.
Write Lock Bits	0000 1100	0000 0111	xxxx 0xxx	Write lock bits. Set LE1, LE2 or LE3 = '0' to program to 1 bit.






- 1. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
- 2. "low addr" - high order address.
- 3. "xxxx" means

AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EAN _{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	H ⁽¹⁾	H ⁽¹⁾	X						
Chip Erase	H	L	 ⁽²⁾	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bit	H	L		12V	H	L	H	L	DIN	X
Bit 1									P0.7 = 0	X
Bit 2									P0.6 = 0	X
Bit 3									P0.5 = 0	X
Read Lock Bit	H	L		12V	H	L	L	L	DOUT	X
Bit 1									@P0.2	X
Bit 2									@P0.1	X
Bit 3									@P0.0	X
Read Athlet Code				12V				L	DOUT	30H
Read Device Code				12V				L	DOUT	31H
Serial Prog. Enable	H	L		12V	L	L	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L		12V	L	L	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L		12V	H	H	L	H	@P0.0	X

- Notes:
1. "H" = weakly pulled High internally.
 2. Chip Erase and Serial Programming Fuse require a 10-ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
 3. P0.0 is pulled Low during programming to indicate P0.0 busy.
 4. "X" = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

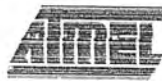


Figure 14. Programming the Flash/EEPROM Memory

Figure 15. Flash/EEPROM Serial Downloading

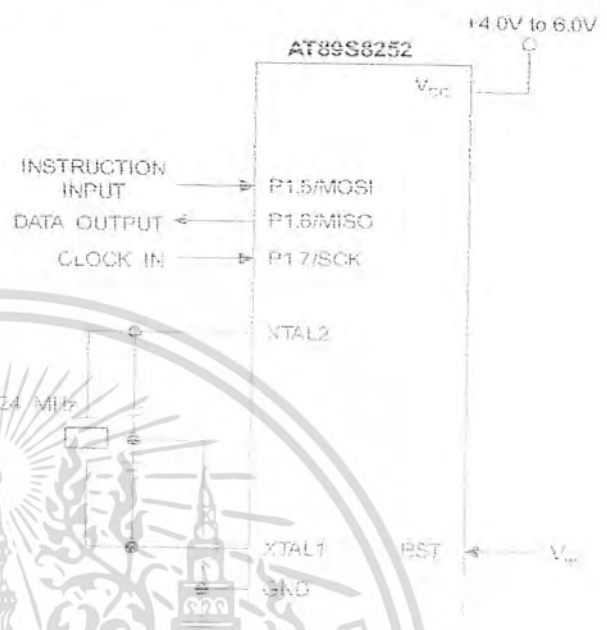
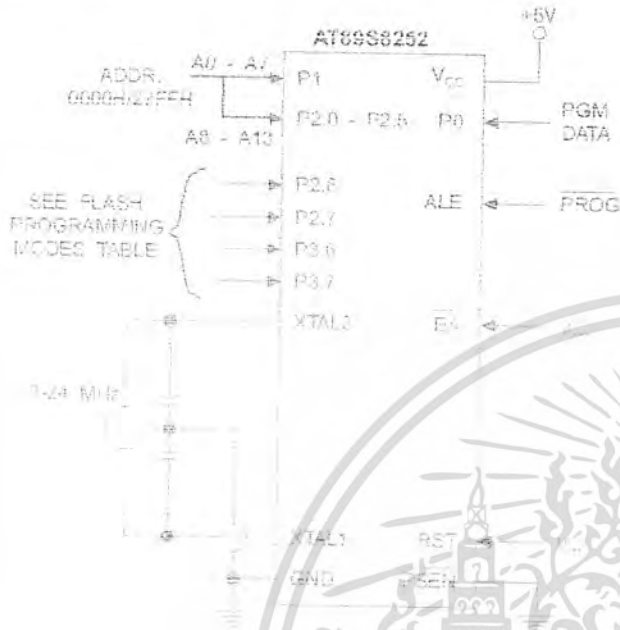
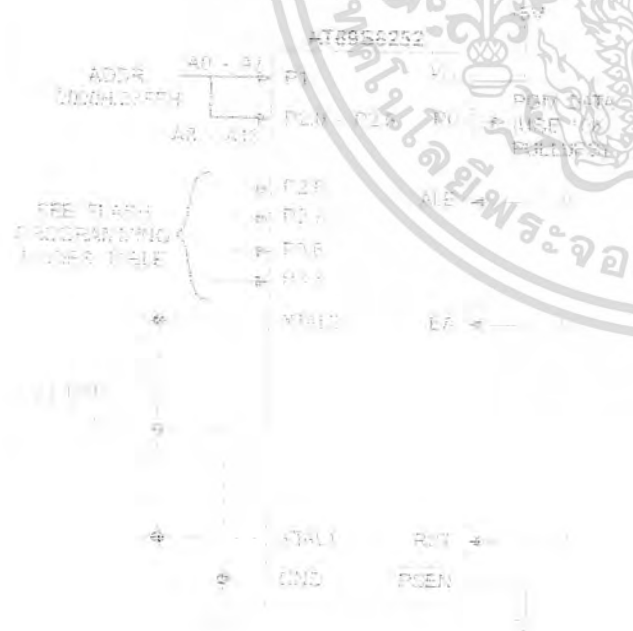


Figure 16. Verifying the Flash/EEPROM Memory



AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นาเบไซบระยชนตาดนการค
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

Flash Programming and Verification Characteristics-Parallel Mode

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$

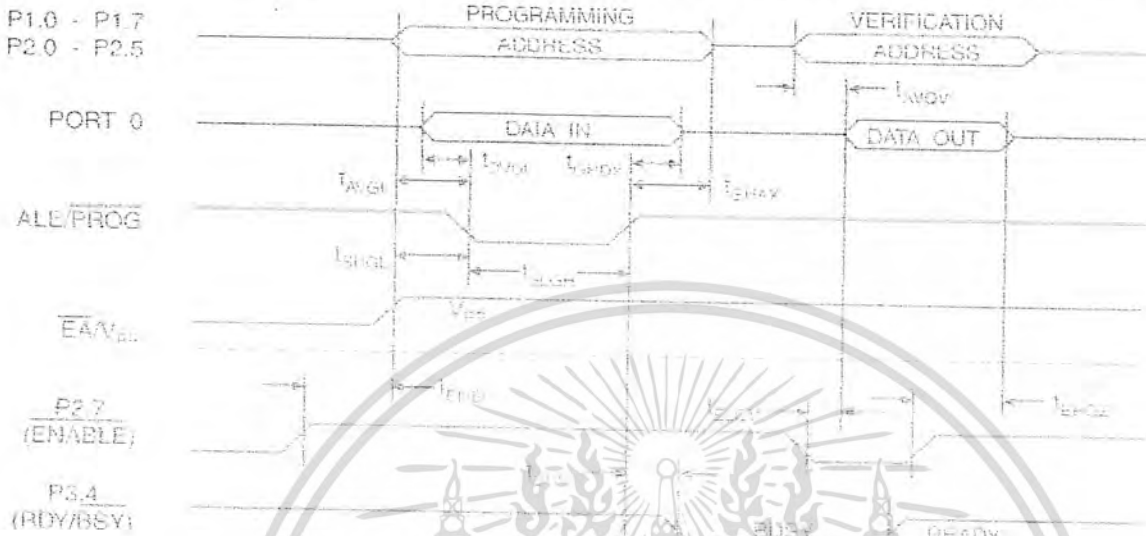
Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
f_{CLCK}	Oscillator Frequency	3	24	MHz
t_{AVDL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCK}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCK}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$46t_{CLCK}$		
t_{DHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCK}$		
t_{ENSH}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCK}$		
t_{SHDL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		ns
t_{GLCH}	$\overline{\text{PROG}}$ Width	1	110	ns
t_{AVDV}	Address to Data Valid		$48t_{CLCK}$	
t_{LDV}	ENABLE Low to Data Valid		$48t_{CLCK}$	
t_{HOZ}	Data Float After ENABLE	0	$48t_{CLCK}$	
t_{HDL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	ns
t_{WC}	Byte Write Cycle Time		2.0	ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





Flash/EEPROM Programming and Verification Waveforms - Parallel Mode



Serial Downloading Waveforms



AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางบริษัทฯ
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

Absolute Maximum Ratings*

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

***NOTICE**

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

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 5.0\text{V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL}	Input Low Voltage (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_{IH}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage (Ports 1, 2, 3)	$I_{OL} = 1.5\text{ mA}$		0.5	V
V_{OL}	Output Low Voltage (Port 0, ALE, PSEN)	$I_{OL} = 3.2\text{ mA}$		0.5	V
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, PSEN)	$I_{OH} = 100\ \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = 250\ \mu\text{A}$	$0.75 V_{CC}$		V
		$t_{rF} = 10\ \mu\text{s}$	$0.9 V_{CC}$		V
V_{OH}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = 800\ \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = 300\ \mu\text{A}$	$0.75 V_{CC}$		V
		$t_{rF} = 50\ \mu\text{s}$	$0.9 V_{CC}$		V
I_{L1}	Logical 0 Input Current (Ports 1, 2, 3)	$V_{IH} = 0.45\text{V}$		-50	μA
I_{L2}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	$V_{IH} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{L3}	Input Leakage Current (Port 0, EA)	$V_{IH} = 0.45\text{V}$		± 10	μA
RRST	Reset Pulldown Resistor		50	300	k Ω
C_{IC}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
I_{CC}	Power Down Mode ¹	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

Notes: 1. Under steady-state (non-transient) conditions, I_{OL} must be externally limited as follows:
 Maximum I_{OL} per port pin: 10 mA
 Maximum I_{OL} per 8-bit port:
 Port 0: 20 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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้