



ระบบจำลองการให้น้ำแก่พืชในโรงเพาะชำโดยใช้คอมพิวเตอร์

Simulation system of automatic sprinkle water to plant by PC.

ห้องสมุดคณะวิศวกรรมศาสตร์	
วันที่รับ..... 19/01/2544	14043583
เลขทะเบียน..... 0๗ 4400484	
เลขเรียกหนังสือ..... TC	
มหาวิทยาลัยนครสวรรค์	พ 8298 2543

นางสาวพุทธิชาติ ขันตันขง รหัส 40362378

นายภูษิต หินหาด รหัส 40362410

นางสาวสุวิมล วสุศิริกุล รหัส 40362634

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์

ปีการศึกษา 2543

หัวข้อโครงการ : ระบบจำลองการให้น้ำแก่พืชในโรงเพาะชำโดยใช้คอมพิวเตอร์
ผู้ดำเนินโครงการ : นางสาวพุทธชาติ ขันตันธง รหัส 40362378
นายภูษิต หินหาด รหัส 40362410
นางสาวสุวิมล วสุศิริกุล รหัส 40362634
อาจารย์ที่ปรึกษา : อ. สมชาย ไชคมาวิโรจน์
อาจารย์ที่ปรึกษาร่วม : อ. มุทิตา สงฆ์จันทร์
สาขา : วิศวกรรมไฟฟ้า
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา : 2543

บทคัดย่อ

โครงการวิศวกรรมนี้นำเสนอระบบจำลองการให้น้ำอัตโนมัติในโรงเพาะชำโดยใช้ไมโครคอนโทรลเลอร์ หลักการควบคุมใช้หลักการควบคุมการเปิด-ปิดวาล์วน้ำให้แก่พืช ซึ่งโครงการนี้ใช้ผักคะน้าเป็นกรณีศึกษาโดยมีความชื้นในดินเป็นเงื่อนไขในการควบคุม ระดับความชื้นในดินจะถูกควบคุมให้เหมาะสม คืออยู่ในช่วง 50 ถึง 70 เปอร์เซ็นต์ โดยใช้เซ็นเซอร์วัดความชื้นและวงจรอิเล็กทรอนิกส์เป็นตัววัดค่า เมื่อระดับความชื้นต่ำกว่าที่กำหนดไมโครคอนโทรลเลอร์จะสั่งงานให้โซลินอยด์วาล์วเปิดน้ำ จนถึงระดับความชื้น 70 เปอร์เซ็นต์ วาล์วน้ำก็จะปิด นอกจากนี้ยังมีส่วนของโปรแกรมคอมพิวเตอร์ใช้สำหรับติดต่อกับผู้ใช้ โดยสามารถแสดงตารางแนะนำการบำรุงรักษาพืชและผู้ใช้สามารถเก็บข้อมูลการเจริญเติบโตของพืชเพื่อใช้ในการเปรียบเทียบการเพาะปลูกในแต่ละรุ่นต่อไปได้ เพื่อประโยชน์ในการปรับปรุงการปลูกครั้งต่อไป

Project Title : Simulation System of Automatic Sprinkle Water to Plant by PC.
Name : Miss Puttachat Khuntontong ID. 40362378
Mr. Phusit Hinhard ID. 40362410
Miss Suwimon Wasusirikul ID. 40362634
Project Advisor : Mr.Somchai Chokmaviroj
Co-Project Advisor : Miss Mutita Songchun
Field of Study : Electrical Engineering
Department : Electrical and Computer Engineering
Academic Year : 2000

Abstract

This engineering project presents the simulation system of automatic control of sprinkle water using micro-controller. The state (ON-OFF) of solenoid valves are controlled for this purpose. In this project, the kale vegetable is used for case study and the relative humidity in soil are used as the condition for controller. The suitable relative humidity which is 50% to 70% is controlled. Using the humidity sensor and electronic circuit, the measurement of soil's humidity can be achieved. If the humidity is less than 50%, the solenoid valves will be turned ON, and then the humidity will be raised up until more than 70%, the solenoid valve will be turned OFF. In addition to, the PC program is used for user interface. The schedule and data of interested plant can be shown and recommended to user by this program. Not only that, the history data records of plant can be kept into computer for next decision of next plan.

ลำดับสัญลักษณ์

สัญลักษณ์	ความหมาย	หน่วย
A	กระแสไฟฟ้า	Ampere
Av	อัตราขยาย	-
f	ความถี่	Hz
Rf	ตัวต้านทานอ้างอิง	Ohms
Ri	ตัวต้านทานอินพุท	Ohms
T	เวลา	เมทซ์ิน ไซเคิล
V	ความต่างศักย์ไฟฟ้า	volt



กิตติกรรมประกาศ

โครงการนี้สามารถสำเร็จลุล่วงได้ด้วยดีเพราะได้รับข้อเสนอแนะจากอาจารย์สมชาย โชคมาวิโรจน์ ในเรื่องรูปแบบและขอบข่ายของโครงการ รวมทั้งได้รับคำปรึกษาเกี่ยวกับระบบไมโครคอนโทรลเลอร์จาก อาจารย์ปัญญา เหล่าอนันต์ธนา ขอขอบคุณท่านอาจารย์ทั้งสองเป็นอย่างยิ่ง โดยมีอาจารย์มูซิดา สงฆ์จันทร์ เป็นอาจารย์ที่ปรึกษาร่วม และขอขอบคุณความอนุเคราะห์จาก พี่ชาติ พี่น้อย ที่อำนวยความสะดวก เครื่องมืออุปกรณ์ สุกทำยก็เพื่อนๆ ที่ภาควิชาไฟฟ้าและคอมพิวเตอร์ โดยเฉพาะอย่างยิ่ง พิทักษ์ และ ก้อย ที่ให้ความรู้ ข้อเสนอแนะ และคำแนะนำต่าง ๆ อย่างดี

ข้าพเจ้าคณะผู้จัดทำโครงการขอขอบพระคุณทุกท่านเป็นอย่างยิ่ง

นางสาวพุทธชาติ	ขันตันรง
นายภูมิต	หินหาด
นางสาวสุวิมล	วสุศิริกุล



สารบัญ

	หน้า
ใบรับรองโครงการวิจัย	ก
บทคัดย่อ	ข
Abstract	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูปภาพ	ซ
สารบัญกราฟ	ฅ
ลำดับสัญลักษณ์	ฉ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบข่ายของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	3
1.6 งบประมาณที่ใช้	3
บทที่ 2 หลักการและทฤษฎี	4
2.1 ไมโครคอนโทรลเลอร์	4
2.2 บอร์ดไมโครคอนโทรลเลอร์ 8032	13
2.3 ออปแอมป์	17
2.4 วาล์ว	22
2.5 ขดลวดโซลินอยด์	31
2.6 เทคนิคการให้น้ำแก่พืช	41
2.7 มาตรฐาน RS-232-C	44

สารบัญ (ต่อ)

	หน้า
บทที่ 3 ออกแบบการทดลอง	47
3.1 การทำงานของบอร์ดควบคุม	47
3.2 วงจรตรวจวัดความชื้น	49
3.3 วงจรวัดอุณหภูมิ	50
3.4 การทดลองระบบทั้งหมด	51
บทที่ 4 ผลการทดลองและวิเคราะห์	53
4.1 ค่าแรงดันเอาต์พุตจากวงจรวัดความชื้น	53
4.2 วิเคราะห์ผลการทดลองจากวงจรวัดความชื้น	54
4.3 ค่าแรงดันเอาต์พุตที่ได้จากวงจรวัดอุณหภูมิ	54
4.4 วิเคราะห์ผลการทดลองจากวงจรวัดอุณหภูมิ	55
4.5 ค่าแรงดันเมื่อผ่านการแปลงจากวงจร A/D	55
4.6 ผลการทำงานทั้งระบบ	56
4.7 วิเคราะห์ผลการทดลองการทำงานทั้งระบบ	57
4.8 โปรแกรมที่ใช้ในการติดต่อกับผู้ใช้	57
บทที่ 5 บทสรุป	62
5.1 สรุปผลการทดลอง	62
5.2 ปัญหาที่พบในการทำโครงงาน	63
5.2 ประเมินผลและข้อเสนอแนะ	63
5.3 แนวทางการพัฒนาระบบต่อ	64
บรรณานุกรม	
ภาคผนวก	
ภาคผนวก ก	ก-1
ภาคผนวก ข	ข-1
ภาคผนวก ค	ค-1
ภาคผนวก ง	ง-1
ประวัติผู้เขียน	

สารบัญตาราง

	หน้า
ตารางที่ 1.1 ขั้นตอนการดำเนินงาน	2
ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัทอินเทล	4
ตารางที่ 2.2 จำนวนชนิดทั่วไปของออปแอมป์	19
ตารางที่ 2.3 ลักษณะและความรู้สึกรู้สีกจากการสัมผัสเนื้อดินที่ระดับความชื้นต่างๆ	43
ตารางที่ 2.4 การทำงานของ RS-232 ทั้งสองแบบ	46
ตารางที่ 4.1 ผลการทดลองวัดค่าจากเซนเซอร์วัดความชื้น	53
ตารางที่ 4.2 สรุปผลแรงดันจากเซนเซอร์วัดความชื้น	54
ตารางที่ 4.3 ผลการทดลองวัดค่าเอาต์พุตจากวงจรวัดอุณหภูมิ	54
ตารางที่ 4.4 ค่าแรงดันเมื่อแปลงเป็นค่าดิจิทัล	55
ตารางที่ 4.5 ผลการทดลองการทำงานทั้งระบบเมื่อเริ่มที่ดินมีความชื้นน้อย	56
ตารางที่ 4.6 ผลการทดลองการทำงานทั้งระบบเมื่อเริ่มที่ดินมีความชื้นมาก	57
ตารางที่ ง-1 วัดการเจริญเติบโตของคะน้ำ	ง-3
ตารางที่ ง-2 ชนิดพันธุ์คะน้ำในประเทศไทย	ง-8
ตารางที่ ง-3 พันธุ์คะน้ำเงินของต่างประเทศ	ง-9

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แผนภาพบล็อกแสดงหน่วยงานพื้นฐานของ MCS-51	6
รูปที่ 2.2 การกำหนดหน้าที่ขาสัญญาณของ ไอซี 8051	6
รูปที่ 2.3 การใช้คริสตอลภายนอกต่อเข้ากับวงจรรอสซิลเลเตอร์ภายใน	7
รูปที่ 2.4 แผนภาพเวลาพื้นฐานของ 8051 และลำดับของช่วงเวลา state ในการทำคำสั่งหนึ่งไบต์	8
รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูลภายใน	12
รูปที่ 2.6 การเชื่อมโยงติดต่อระหว่างหน่วยความจำโปรแกรมข้อมูลกับ ไมโครคอนโทรลเลอร์	13
รูปที่ 2.7 ผังเวลาของสัญญาณ ALE และ PSEN	14
รูปที่ 2.8 การเขียนข้อมูลลงในหน่วยความจำข้อมูลภายนอก	16
รูปที่ 2.9 การอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก	16
รูปที่ 2.10 ส่วนประกอบภายในของออปแอมป์	17
รูปที่ 2.11 ลักษณะทั่วไปของออปแอมป์	17
รูปที่ 2.12 ลักษณะออปแอมป์แต่ละชนิด	18
รูปที่ 2.13 ความสัมพันธ์ระหว่างขั้วออปแอมป์	19
รูปที่ 2.14 วงจรขยายกลับเฟสแบบมาตรฐาน	20
รูปที่ 2.15 การต่อออปแอมป์เป็นแหล่งจ่ายกระแส	21
รูปที่ 2.16 ลักษณะของลิ้นวาล์ว	23
รูปที่ 2.17 ประดูแบบ Full bore through conduit pipe line gate valve	23
รูปที่ 2.18 ลักษณะของการประกอบของก้านวาล์วในแบบต่าง ๆ	23
รูปที่ 2.19 บอนเน็ตแบบต่าง ๆ	24
รูปที่ 2.20 รูปบอลล์วาล์ว แสดงการเจาะรูผ่านบูกบอลล์	26
รูปที่ 2.21 วาล์วที่ใช้กลไกของแควมทำให้ก้านวาล์วเคลื่อนที่	27
รูปที่ 2.22 บอลล์วาล์วทรันชันเม้าต์ (Trunnionmounted)	27
รูปที่ 2.23 โครงสร้างแบบต่าง ๆ ของลิ้น	27
รูปที่ 2.24 ส่วนประกอบของโกลบวาล์ว	28

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 2.25 ลักษณะของดิน	28
รูปที่ 2.26 วาล์วผีเสื้อขนาดเล็ก	29
รูปที่ 2.27 การประกบของแบบเวเฟอร์	29
รูปที่ 2.28 ไดอะแฟรมวาล์ว	30
รูปที่ 2.29 เชื้อควาล์วแบบสวิงและลิฟต์เชื้อควาล์ว	31
รูปที่ 2.30 สภาวะปกติของโซลีนอยด์วาล์ว	31
รูปที่ 2.31 สภาวะทำงานของโซลีนอยด์วาล์ว	31
รูปที่ 2.32 โครงสร้างโซลีนอยด์แบบแผ่นกลม	32
รูปที่ 2.33 โครงสร้างโซลีนอยด์แบบตุ๊กบอล	32
รูปที่ 2.34 โครงสร้างโซลีนอยด์แบบนั่งบ่า	33
รูปที่ 2.35 โซลีนอยด์วาล์วแบบลูกสูบเลื่อน	33
รูปที่ 2.36 ขดลวดไฟฟ้า	33
รูปที่ 2.37 หม้อแปลง	35
รูปที่ 2.38 ขั้นตอนการแปลงกระแสไฟฟ้าสลับเป็นกระแสไฟฟ้าตรง	35
รูปที่ 2.39 โซลีนอยด์	36
รูปที่ 2.40 โซลีนอยด์แบบเปียก	36
รูปที่ 2.41 โครงสร้างของโซลีนอยด์	38
รูปที่ 2.42 วงจรโซลีนอยด์แบบพรอพเพอร์ชันนัล	38
รูปที่ 2.43 สัญลักษณ์ของทรานซิสเตอร์	39
รูปที่ 2.44 ผลที่เกิดขึ้นทางอาร์คฟุต	40
รูปที่ 2.45 โครงสร้างภายในของโซลีนอยด์วาล์ว	40
รูปที่ 2.46 การเกิดแรงของสนามแม่เหล็ก	41
รูปที่ 2.47 การปรับเปลี่ยนค่ากระแสไฟฟ้าที่เข้าไปในขดลวด	41
รูปที่ 3.1 แผนผังการทำงานของโปรแกรม	48
รูปที่ 3.2 วงจรวัดความชื้น	49

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.3 วงจรวัดอุณหภูมิ	51
รูปที่ 3.4 แผนภาพการต่อระบบรวม	52
รูปที่ 4.1 หน้าจอควบคุมการเปิด/ปิดวาล์ว	58
รูปที่ 4.2 หน้าจอนำเสนอวิธีการเพาะปลูกคะน้า	59
รูปที่ 4.3 แบบฟอร์มการบันทึกข้อมูลการเจริญเติบโตของคะน้า	60
รูปที่ 4.4 แบบฟอร์มนำเสนอวิธีการแก้ไขปัญหาโรคพืชและแมลงของคะน้า	61
รูปที่ ข-1 วงจรบอร์ดไมโครคอนโทรลเลอร์ 8032	ข-2
รูปที่ ง-1 ลักษณะแปลงปลูกสัปดาห์ที่ 1	ง-3
รูปที่ ง-2 สัปดาห์ที่ 2	ง-4
รูปที่ ง-3 การเก็บข้อมูลความชื้นและความสูงในสัปดาห์ที่ 3	ง-4
รูปที่ ง-4 การวัดความสูงในสัปดาห์ที่ 4	ง-5
รูปที่ ง-5 การเจริญเติบโตในสัปดาห์ที่ 5	ง-5
รูปที่ ง-6 ลักษณะของคะน้าในสัปดาห์ที่ 6	ง-5
รูปที่ ง-7 ลักษณะของการจัดการแปลงปลูก	ง-6
รูปที่ ง-8 ลักษณะของเมล็ดพันธุ์คะน้า	ง-7

สารบัญกราฟ

	หน้า
กราฟที่ 2.1 เวลาในการกินกระแสไฟฟ้า	34
กราฟที่ 2.2 เปรียบเทียบวงจรแบบอินดักแตนซ์และ แบบคาปาซิแตนซ์ (Inductance and Capacitance)	34
กราฟที่ 2.3 การใช้กระแสไฟฟ้าของคอยล์แบบกระแสตรง	37
กราฟที่ 2.4 การใช้กระแสไฟฟ้าของคอยล์แบบกระแสสลับ	37
กราฟที่ 2.5 ฮีตเตอร์รีซิส	39



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ประเทศไทยเป็นประเทศที่มีผลผลิตทางการเกษตรออกสู่ตลาดทั้งในและนอกประเทศเป็นจำนวนมาก แต่ผลผลิตที่ได้ออกมานั้นยังคงด้อยคุณภาพ ทั้งนี้อาจเป็นเพราะเกษตรกรไม่มีความรู้ในการนำเอาเทคโนโลยีมาประยุกต์ในการเพิ่มคุณภาพของผลผลิต

น้ำเป็นปัจจัยหนึ่งที่สำคัญต่อการเจริญเติบโตของพืช ซึ่งถ้าพืชได้รับปริมาณน้ำในระดับที่เหมาะสมแล้ว พืชก็จะมี การเจริญเติบโตที่ดี ทั้งยังส่งผลให้ผลผลิตมีคุณภาพดีอีกด้วย ซึ่งจะทำให้เป็นที่ยอมรับในระดับต่างประเทศได้

โครงการนี้เป็นต้นแบบและแนวทางในการนำเอาเทคโนโลยีมาประยุกต์ใช้ในการเกษตร โดยใช้คอมพิวเตอร์มาควบคุมการส่งจ่ายน้ำในปริมาณที่เหมาะสมและเพียงพอต่อความต้องการของพืช ซึ่งนอกจากพืชจะได้รับปริมาณน้ำในระดับที่พอเหมาะแล้ว ยังเป็นการประหยัดทรัพยากรน้ำ และค่าใช้จ่ายได้อีกทางหนึ่งด้วย ทั้งนี้รูปแบบของโครงการนี้จะใช้ได้กับการเพาะปลูกพืชได้กับพื้นที่หลายไร่ และยังสามารถขยายต่อไปได้อีก

1.2 วัตถุประสงค์

- 1) เพื่อควบคุมการให้น้ำแก่พืชให้เหมาะสมกับความต้องการของพืชทำให้ผลผลิตที่ได้รับมีปริมาณและคุณภาพตามที่ตลาดต้องการได้
- 2) เพื่อช่วยเหลือเกษตรกรในการเก็บข้อมูลการเจริญเติบโต ผลผลิตและข้อเสนอแนะต่าง ๆ ในการดูแลรักษาพืชโดยใช้โปรแกรมคอมพิวเตอร์

1.3 ขอบข่ายของโครงการ

การปลูกพืชไม่ว่าจะเป็นพืชสวน พืชไร่หรือพืชผัก ก็ย่อมต้องใช้น้ำในการทำให้น้ำแก่พืชเจริญเติบโต แต่พืชแต่ละชนิดก็ต้องการปริมาณน้ำที่ไม่เท่ากัน ดังนั้นการที่จะให้น้ำแก่พืชจะต้องมีการควบคุมระดับค่าความชื้นเพื่อให้ตรงกับความต้องการของพืช อีกปัจจัยหนึ่งที่สำคัญคือ ช่วงเวลาที่จะให้น้ำแก่พืชก็ควรจะคำนึงถึงด้วยเพราะในบางเวลาพืชก็ไม่ต้องการน้ำ

เช่น ในตอนกลางวันอากาศร้อน หากให้น้ำแก่พืชก็จะทำให้พืชเหี่ยวหรืออาจตายได้ ในโครงการนี้ เราเลือกที่จะศึกษาพืชในกลุ่มพืชผัก คือ ผักคะน้า ด้วยเหตุผลที่ว่า คะน้าสามารถปลูกง่ายทุกพื้นที่ ทุกฤดูกาล และเป็นพืชที่มีอายุสั้นจึงง่ายต่อการเก็บข้อมูล

ระบบจำลองการควบคุมน้ำในเรือนเพาะชำด้วยคอมพิวเตอร์นี้ได้ใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการเปิด-ปิด โซลีนอยด์วาล์ว ที่จะต่อกับหัวฉีดน้ำที่ให้น้ำกับพืช เงื่อนไขในการเปิด-ปิดน้ำ จะใช้ช่วงเวลาและระดับความชื้นในดินเป็นแปรที่ใช้กำหนด ซึ่งจะกำหนดโดยผู้ใช้งานทางไมโครคอมพิวเตอร์ ตามชนิดและความต้องการของพืชรวมไปถึงลักษณะของดินด้วย

เมื่อไมโครคอนโทรลเลอร์รับค่าระดับความชื้นจากไมโครคอมพิวเตอร์แล้วก็ทำการเปรียบเทียบกับค่าที่ได้จากเซนเซอร์วัดระดับความชื้นในดิน ถ้าตรงตามเงื่อนไขที่กำหนดคือ ถึงช่วงเวลาที่ต้องการให้เปิดน้ำและระดับความชื้นในดินมีค่าน้อยกว่าที่พืชต้องการแล้วไมโครคอนโทรลเลอร์ก็จะสั่งให้ชุดขับโซลีนอยด์วาล์วทำงานแล้วน้ำก็จะถูกฉีดให้กับพืช ในทางกลับกันวาล์วจะปิดเมื่อค่าระดับความชื้นถึงค่าที่กำหนดไว้และจะทำงานอีกครั้งเมื่อเงื่อนไขเป็นคั้งที่กล่าวมา

สำหรับโปรแกรมที่ติดต่อกับผู้ใช้และเก็บข้อมูลเกี่ยวกับการเจริญเติบโตจะใช้งานบนไมโครคอมพิวเตอร์ เพื่อให้ผู้ใช้ได้กำหนดช่วงเวลารวมไปถึงเก็บรวบรวมข้อมูลเพื่อนำไปวิเคราะห์ต่อไป

1.4 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

กิจกรรม	เดือน-ปี						
	มี.ค. 43	เม.ย. 43	พ.ค. 43	มิ.ย. 43	ก.ค. 43	ส.ค. 43	ก.ย. 43
1. ศึกษาพืช,ศึกษาหลักการทำงานของคอนโทรลเลอร์, วาล์ว, เซนเซอร์ เพื่อควบคุมการให้น้ำแก่พืช	←→						
2. อุปกรณ์อินเตอร์เฟสกับcom, software ในการควบคุมคอนโทรลเลอร์		←→					

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน (ต่อ)

กิจกรรม	เดือน-ปี						
	มี.ค. 43	เม.ย. 43	พ.ค. 43	มิ.ย. 43	ก.ค. 43	ส.ค. 43	ก.ย. 43
3. วางระบบ เริ่มฮาร์ดแวร์			←————→				
4. เขียนแผนงาน					←————→		
5. สอบpaper							←————→

1.5 ผลที่คาดว่าจะได้รับ

ในการนำเอาระบบคอมพิวเตอร์มาใช้ในการควบคุมการเปิด-ปิดวาล์วน้ำนั้นจะมีผลคือ

1.5.1) โครงการนี้เป็นการพัฒนาการเกษตร โดยการนำเอาเทคโนโลยีมาประยุกต์ใช้งานเพื่อการเกษตร ทำให้การเกษตรกรรมมีประสิทธิภาพทางผลผลิตที่สูงขึ้น

1.5.2) การควบคุมระบบเป็นไปอย่างมีประสิทธิภาพ สามารถตอบสนองความต้องการของผู้ใช้ได้ตามต้องการ

1.5.3) เป็นระบบที่สามารถให้คำแนะนำเบื้องต้นแก่เกษตรกรได้ โดยอาศัยโปรแกรมที่มีข้อมูลเบื้องต้นของพืช

1.6 งบประมาณที่ใช้

1.) อุปกรณ์อิเล็กทรอนิกส์	1500	บาท
2.) โซลินอยด์วาล์ว	1500	บาท
รวม	<u>3000</u>	บาท

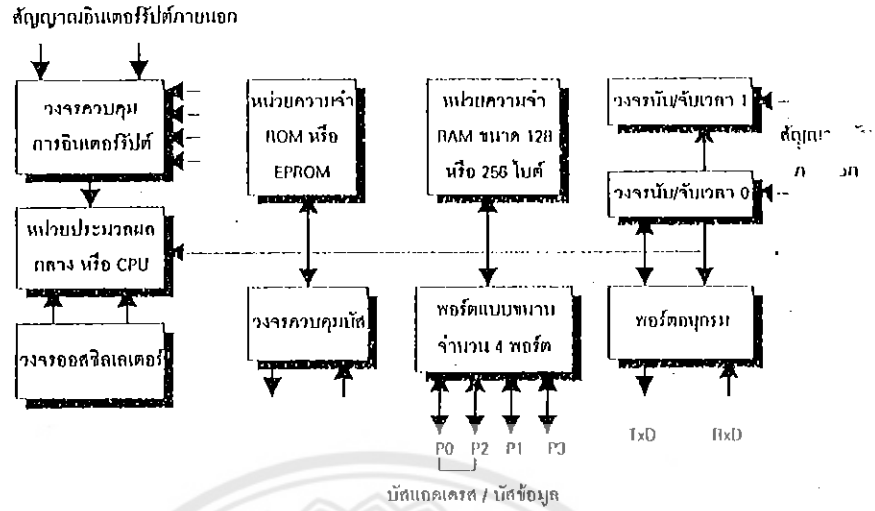
ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัทอินเทล(ต่อ)

EMBEDDED CONTROLLER										
Feature	8051AH	8031AH	8751H	80C51BH	80C31BH	87C51	8052AH	8032AH	8752	8044H
Max Clock Frequency (MHz)	12	12	12	16	16	16	16	12	12	12
Typical instruction Time (us)	1	1	1	0.75	0.75	0.75	1	1	1	1
16-Bit Timer/Counters	2	2	2	2	2	2	3	3	3	2
Serial Communication	Synchronous Mode, Asynchronous Mode, 9 or 10 -Bit Programmable									HDLC/SDLC
No. of I/O Line	32	16	32	32	16	32	32	16	32	32
Interrupt Sources (Two Priority Levels) Power Requirements 125	5	5	5	5	5	5	6	6	6	5
(ICC Max. mA) Programmable Power Modes Idle Power Down	-	-	-	4.0 mA 50 uA	4.0 mA 51 uA	4.0 mA 52 uA	-	-	-	- 30 mA

2.1.1 คุณสมบัติพื้นฐานของ 8051

จากแผนภาพในรูปที่ 2.1 แสดงให้เห็นถึงหน่วยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ที่จัดอยู่ในตระกูล MCS-51 นี้ ประกอบด้วย

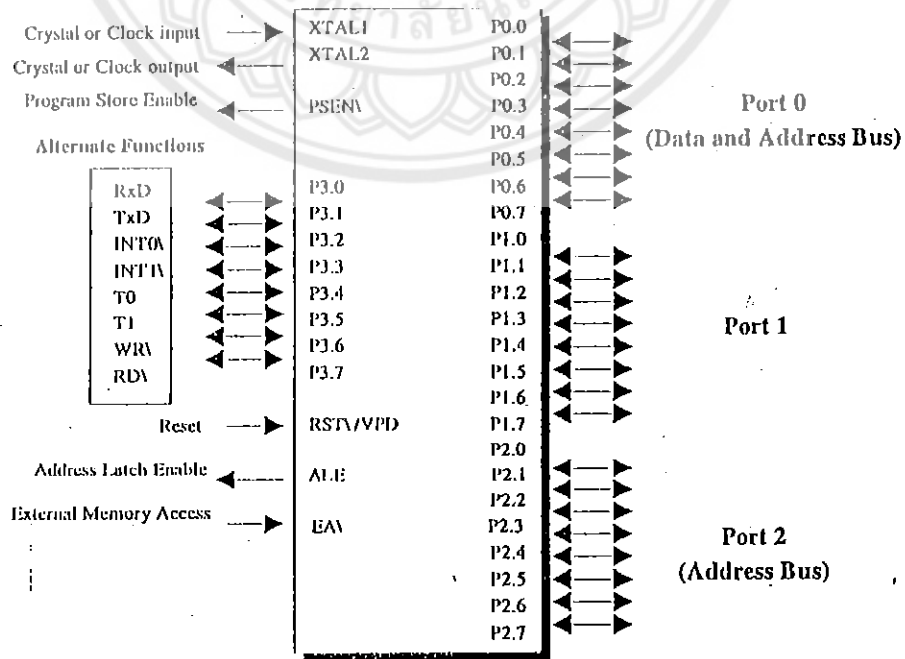
- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ EPROM (เบอร์ 8751) หรือแบบรอม (ROM) (เบอร์ 8051)
- หน่วยความจำแบบแรม (RAM) ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)



รูปที่ 2.1 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของ MCS-51

- วงจรรวมการอินเทอร์พรีตได้จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้สองระดับ
- วงจรรอซิงโครไนซ์ภายใน

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา ดังแผนภาพในรูปที่ 2.2



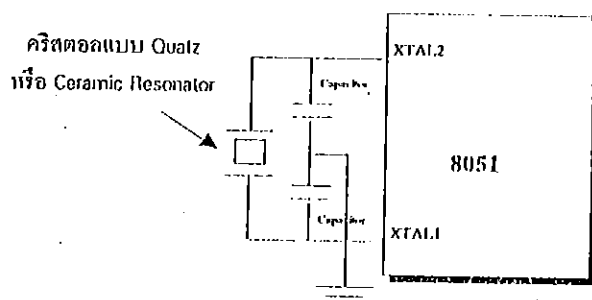
รูปที่ 2.2 การกำหนดหน้าที่ขาสัญญาณของ ไอซี 8051

ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ที่อยู่ย่อที่กำกับในแต่ละขา อย่างไรก็ตาม จะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า Alternate Functions ในรูปที่ 2.2) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่น ขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุต/อินพุตตามปกติ หรืออาจทำหน้าที่เป็นขาสัญญาณอินพุตของข้อมูลสื่อสารแบบอนุกรม (RxD) ให้กับวงจรสื่อสารแบบอนุกรมของ 8051 ได้ ซึ่งการกำหนดว่าจะทำงานในลักษณะใดก็ขึ้นอยู่กับ การเชื่อมต่อวงจรเข้ากับขาสัญญาณและโปรแกรมควบคุมของระบบนั้น

2.1.2 ฐานเวลาในการทำงานของซีพียูภายใน 8051

8051 มีวงจรออสซิลเลเตอร์อยู่ภายใน สำหรับการสร้างพัลส์ของสัญญาณนาฬิกา ซึ่งจะนำไปเป็นฐานเวลา หรือการกำหนดจังหวะการทำงานของหน่วยการทำงานทั้งหมดให้สอดคล้องกัน (Synchronization) โดยปกติแล้วก็มักจะทำโดยการใช้คริสตัลเชื่อมต่อเข้ากับขาสัญญาณ XTAL1 และ XTAL2 พร้อมกับตัวเก็บประจุตั้งลักษณะในรูปที่ 2.3 หรืออาจจะเป็นสัญญาณนาฬิกาจากภายนอกก็ได้

พัลส์ความถี่ของสัญญาณนาฬิกาจะเรียกว่าพัลส์ (ใช้สัญลักษณ์เป็นตัวอักษร P) และคาบของสัญญาณนาฬิกา นี้ เรียกว่า คาบเวลาออสซิลเลเตอร์ (Oscillator Period) คาบเวลาออสซิลเลเตอร์ จำนวนสองคาบ เรียกว่า สเตท (State) (ใช้สัญลักษณ์เป็นตัวอักษร S) ซึ่งจะนำไปใช้เป็นช่วงเวลาพื้นฐานการทำงานย่อยของไมโครคอนโทรลเลอร์ เช่น การนำคำสั่ง (Fetch) การถอดความหมาย (Decode) การประมวลผล (Execute) และการเขียนข้อมูล (Write) เป็นต้น ดังแสดงเป็นแผนภาพในรูปที่ 2.4 ช่วงเวลาจำนวน 6 ครั้ง จะเรียกว่า แมชชีน ไซเคิล (Machine Cycle) ดังนั้นค่าหนึ่งแมชชีน ไซเคิลจะใช้เวลา 12 คาบเวลาออสซิลเลเตอร์ ค่าของแมชชีน ไซเคิลนี้จัดว่าเป็นช่วงเวลาที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ซึ่งหากว่าเป็นคำสั่งที่ซับซ้อนมากก็จะต้องใช้เวลานานสองถึงสามแมชชีน ไซเคิล



รูปที่ 2.3 การใช้คริสตัลภายนอกต่อเข้ากับวงจรออสซิลเลเตอร์ภายใน

การคำนวณหาว่าเวลาที่ใช้ในการทำคำสั่งใดจนเสร็จสิ้น จะต้องดูว่าคำสั่งนั้นใช้จำนวนเมกซ์ซีนไซเคลเป็นเท่าไรในการประมวลผลเวลาที่ใช้จะคำนวณตามสูตร

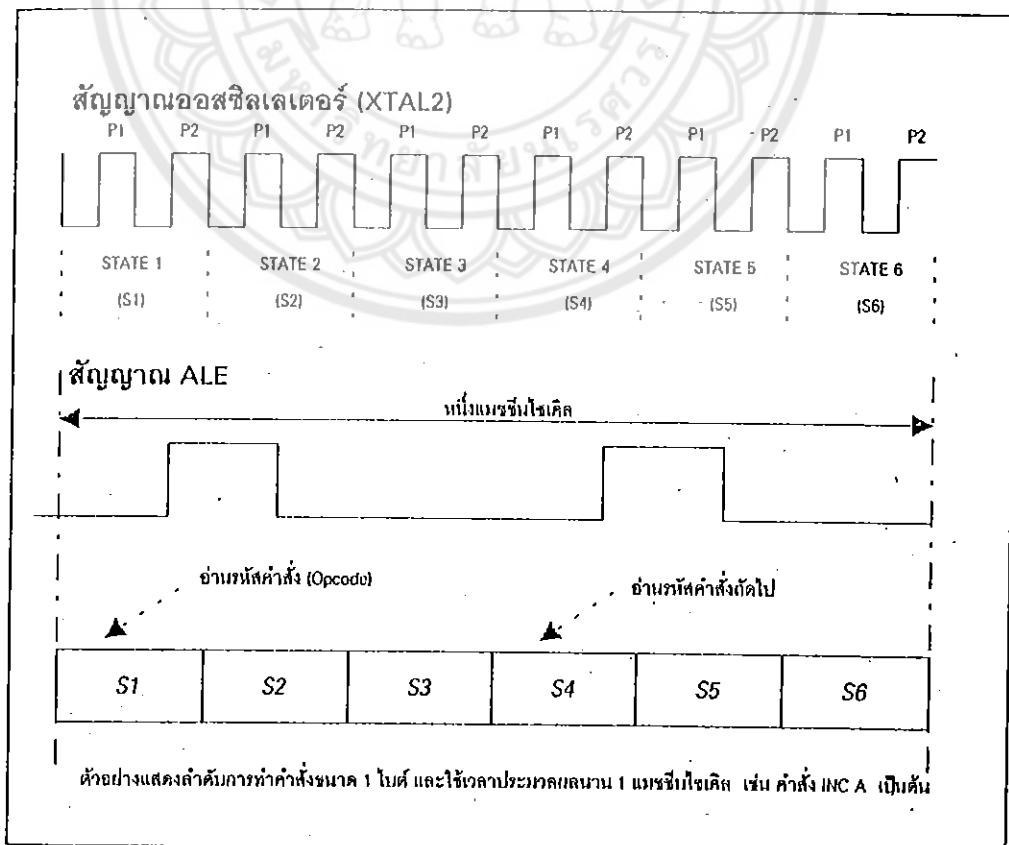
$$T = \frac{C \times 12}{Crystal \quad Frequency}$$

โดย C เป็นค่าจำนวนเมกซ์ซีนไซเคลของคำสั่ง
Crystal Frequency เป็นค่าความถี่ของคริสตอลที่ใช้กับ 8051

ตัวอย่างเช่น

เวลาในการทำคำสั่ง ADD A, R1 ซึ่งต้องการ 3 เมกซ์ซีนไซเคล
เมื่อใช้คริสตอล 16 เมกะเฮิร์ต จะเป็นเวลานาน 0.75 ไมโครวินาที และ
เมื่อใช้คริสตอล 12 เมกะเฮิร์ต จะเป็นเวลานาน 1 ไมโครวินาที เป็นต้น

อย่างไรก็ตาม ในบางครั้งอาจจะพบเห็นการใช้ค่าของคริสตอลเป็น 11.059 เมกะเฮิร์ต ทั้งนี้โดยมีเหตุผลเนื่องจาก สามารถนำค่าความถี่ที่ได้ไปใช้ในการเป็นฐานเวลาสำหรับการสร้างความถี่ในการรับ/ส่งข้อมูลอนุกรมซึ่งเป็นหน่วยการทำงานหนึ่งภายใน 8051 เอง โดยจะทำให้ได้ค่าที่ใกล้เคียงกับค่ามาตรฐานคือ 19200, 9600, 4800, 2400, 1200 และ 300 บิต/วินาที



รูปที่ 2.4 แผนภาพเวลาพื้นฐานของ 8051 และลำดับของช่วงเวลา State ในการทำคำสั่งหนึ่ง ไบต์

2.1.3 หน่วยความจำของคอนโทรลเลอร์ตระกูล 8051

ส่วนสำคัญที่สุดส่วนหนึ่งของคอนโทรลเลอร์ก็คือ หน่วยความจำแรมภายใน ซึ่งจำเป็นในการเขียนและอ่านเพื่อรักษาข้อมูลที่จำเป็นในขณะที่โปรแกรมทำงานอยู่และเรียกกลับมาใช้งานอีกครั้ง เมื่อจำเป็น

ในหน่วยความจำจะแบ่งพื้นที่ที่ใช้เป็นฟังก์ชันพิเศษ คือ

1) รีจิสเตอร์ฟังก์ชันพิเศษ

พื้นที่ที่สำคัญหนึ่งในหน่วยความจำแรมก็คือ รีจิสเตอร์ฟังก์ชันพิเศษ ซึ่งเราจะเรียกย่อ ๆ ว่า SFR ดังเช่นชื่อของมัน รีจิสเตอร์แบบนี้จะรับผิดชอบควบคุมฟังก์ชันพิเศษ ซึ่งจำนวนของรีจิสเตอร์แบบนี้จะแตกต่างกันในแต่ละเวอร์ชันของคอนโทรลเลอร์ในตระกูลของ 8051 เพราะในเวอร์ชันหลัง ๆ ของคอนโทรลเลอร์จะมี On-Chip-Peripherie เพิ่มขึ้นเรื่อย ๆ ซึ่งทำให้ SFR มีจำนวนเพิ่มขึ้น ส่วนชุดคำสั่งของคอนโทรลเลอร์จะไม่มีผลกระทบเพิ่มขึ้นตามมา ตรงนี้จะอธิบายถึง SFR แบบสั้น ๆ เกี่ยวกับฟังก์ชันการทำงานของมัน

1.1) แอคคิวมูลเตเตอร์ (A)

แอกคิวมูลเตเตอร์จะทำหน้าที่เกี่ยวกับการคำนวณทางคณิตศาสตร์ทั้งหมดหรือหน้าที่รองลงมา เราสามารถนำมาใช้เหมือนรีจิสเตอร์ใช้งานทั่วไป เช่น ใช้เก็บรักษาข้อมูลทั่วไป แล้วเราจะได้เห็นว่ามีบางคำสั่งที่จะต้องอ้างผ่านแอกคิวมูลเตเตอร์อย่างเดียว ในกรณีนี้ก็คือ ข้อมูลที่ต้องนำมาทำกับคำสั่งนี้จะต้องถูกถ่ายเทเข้ามาที่แอกคิวมูลเตเตอร์ก่อน

1.2) รีจิสเตอร์ B (B-Register)

มันถูกเรียกว่าเป็นผู้ช่วยแอกคิวมูลเตเตอร์ในคำสั่งการคูณและการหาร (MULAB และ DIVAB) และเช่นเดียวกัน สามารถนำมาใช้เป็นรีจิสเตอร์ใช้งานทั่วไปได้

1.3) ตัวชี้สแต็ก (Stackpointer หรือ SP)

ใน Stackpointer จะเก็บตำแหน่งอ้างอิงหน่วยความจำ รีจิสเตอร์นี้จะมีการเปลี่ยนแปลงทุกครั้งที่มีการเรียกโปรแกรมย่อยใช้งาน เพื่อดูแลการกระโดดกลับเข้าสู่ตำแหน่งอ้างอิงเดิมในโปรแกรมหลัก

1.4) ตัวชี้ข้อมูล (Datapointer หรือ DPTR)

ตัวชี้ข้อมูลเป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งไบต์สูง (DPH) และไบต์ต่ำ (DPL) สามารถถูกใช้งานแยกกันได้ ตัวชี้ข้อมูลนี้จะถูกใช้ในการอ้างอิงตำแหน่งของข้อมูล ซึ่งอยู่ในหน่วยความจำข้อมูล เช่นเดียวกับการทำงานกับอุปกรณ์ภายนอก

1.5) พอร์ตรีจิสเตอร์ (Port Register หรือ Px)

พอร์ตรีจิสเตอร์ P0, P1, P2 และ P3 ควบคุมการติดต่อกับอุปกรณ์อินพุท/เอาต์พุทพอร์ต โดยที่ P2 ของคอนโทรลเลอร์จะใช้ในการติดต่อกับหน่วยความจำโปรแกรมภายนอกบนคอมพิวเตอร์บอร์ดของเราเท่านั้น คอนโทรลเลอร์จะส่งตำแหน่งอ้างอิงหน่วยความจำนี้ออกจากพอร์ตทั้งสองเพื่อติดต่อกับอีพีรอมและแรม

1.6) Interrupt-Priority-Register (IP)

IP จะเก็บข้อมูลของการแบ่งระดับความสำคัญของการอินเทอร์รัปต์ จากแหล่งอินเทอร์รัปต์จากที่ต่าง ๆ

1.7) Interrupt-Enable-Register (IE)

ด้วยการเช็ทบิตที่เกี่ยวข้องใน Interrupt-Enable-Register (IE) เป็นการอนุญาตให้เกิดสัญญาณอินเทอร์รัปต์จากภายนอกได้

1.8) Timer-Mode-Register (TMOD)

Timer-Mode-Register เป็นตัวกำหนดโหมดการทำงานของ Timer 0 และ Timer 1

1.9) Timer-Control-Register (TCON)

Timer-Control-Register จะบรรจุบิตควบคุมของ Timer0 และ Timer1 ด้วย T2CON จะทำงานกับ Timer2 รีจิสเตอร์ตัวนี้จะอยู่ในเวอร์ชันตั้งแต่ 8032 ขึ้นไป แต่จะไม่มีใน 8031 และ แรม (RAM) เวอร์ชัน 8051

1.10) Timer 0 และ Timer 1

ตัวนับของ Timer 0 และ Timer 1 คือ รีจิสเตอร์ TH0, TL และ TH1, TL1

1.11) Timer 2

รีจิสเตอร์ที่ใช้คือ TH2, TL2 นอกจากนั้นยังมี Capture-Register (RCAP2H,RCAP2L) อีก 2 ตัว

1.12) Serial-Control-Register (SCON)

Serial-Control-Register เป็นตัวควบคุมพอร์ตอนุกรมของคอนโทรลเลอร์

1.13) Serial-Data-Buffer (SBUF)

Serial-Data-Buffer จะเก็บค่าที่จะส่งออกไปที่พอร์ตอนุกรม หรืออ่านข้อมูลมาจากพอร์ตอนุกรม

1.14) Processor-Control-Register (PCON)

ในรีจิสเตอร์นี้กับ 8032 จะต้องอ้างอิงแบบบิตแอดเดรส มันจะกำหนดความเร็วในการรับ/ส่งข้อมูลของพอร์ตอนุกรม บิตนี้เราจะเรียกว่า SMOD และจะไปวางไว้ที่ PCON.7

2) การอ้างอิงตำแหน่งแบบโดยตรงในพื้นที่แรม (Direct addressable RAM-area)

การอ้างอิงตำแหน่งโดยตรงในหน่วยความจำของคอนโทรลเลอร์ตระกูล 8051 จะอยู่ในพื้นที่ 00-7FH ในพื้นที่นี้ เราสามารถเขียนและอ่านคำสั่งโดยตรงบนตัวแรม

2.1) พื้นที่ที่สามารถอ้างอิงตำแหน่งแบบบิตแอดเดรส

คำสั่งต่าง ๆ ของคอนโทรลเลอร์ตระกูล 8051 สามารถอ้างอิงถึงบิตแต่ละบิตได้ บิตเหล่านี้จะอยู่ในตำแหน่งที่ 20H ถึง 2FH ของหน่วยความจำแรม นั่นหมายความว่า 16 ไบต์ของแรม ซึ่งสามารถอ้างอิงแบบบิตแอดเดรสได้ นั่นคือเราสามารถอ้างอิงตำแหน่งของบิตได้ทั้งหมด $16 \times 8 = 128$ บิต ซึ่งจะเริ่มจาก 0 ถึง 127 (00 ถึง 7FH) บิตแอดเดรส 0 จะมีค่าของบิตต่ำสุดที่ตำแหน่งหน่วยความจำแรม 20H และบิตแอดเดรส 07 จะมีค่าของบิตสูงสุดในตำแหน่งหน่วยความจำแรมเดียวกัน และจะเรียงลำดับไปตามนี้จนกระทั่งถึงบิตแอดเดรส 7FH ซึ่งเป็นบิตค่าสูงสุดของตำแหน่งหน่วยความจำแรม 2FH

2.2) พื้นที่ แรม (RAM) ที่อ้างอิงตำแหน่งบิตแอดเดรสทางอ้อม

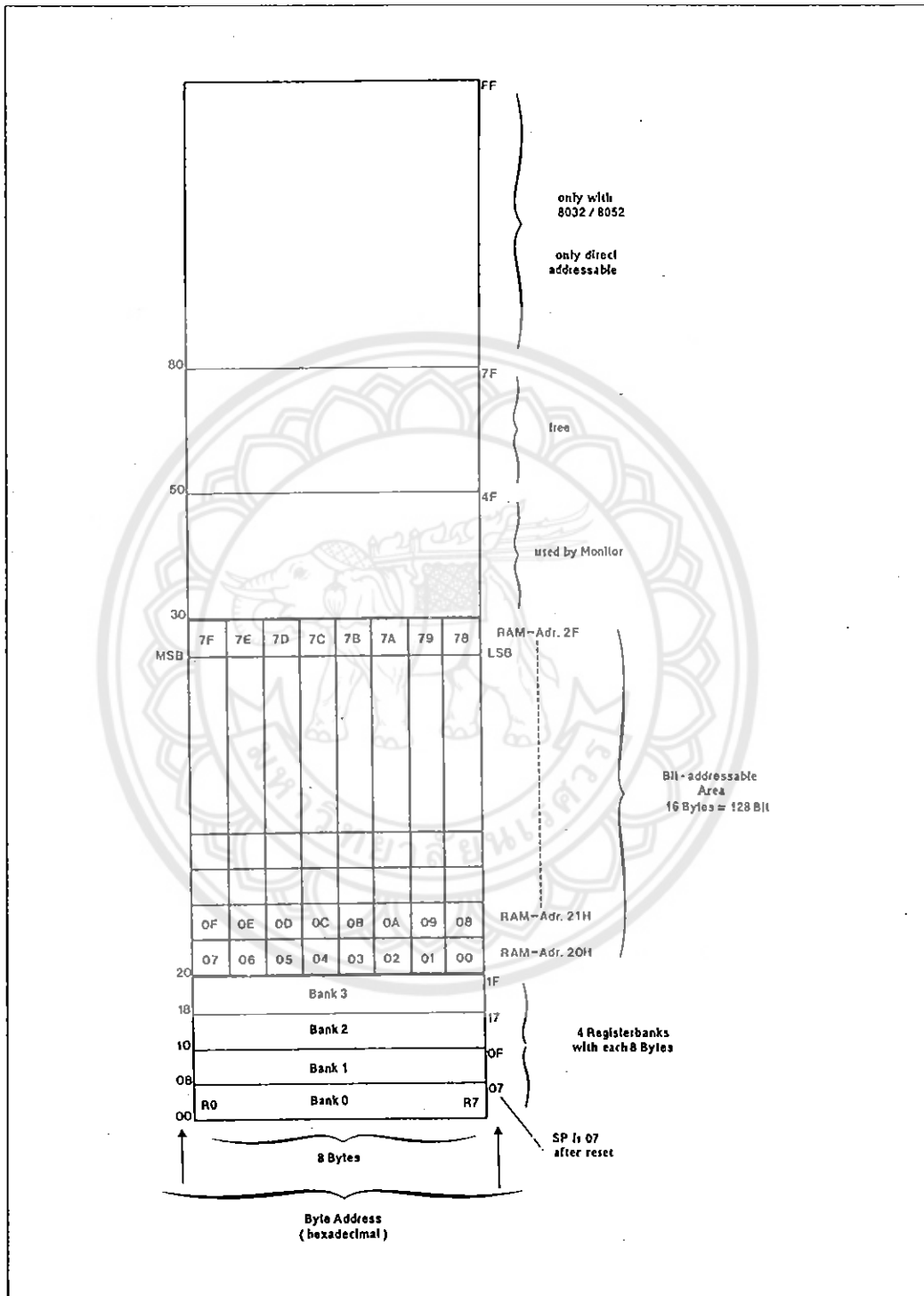
สำหรับเวอร์ชันของ 8032 ขึ้นมาจะมีพื้นที่แรมในตำแหน่งจาก 80H ถึง 0FFH พื้นที่ตรงนี้จะมีการอ้างอิงถึงแบบเดียวกับ SFRs แตกต่างกันก็เพียงตำแหน่งของพื้นที่แรม SFRs เราจะต้องอ้างอิงถึงแบบอ้างอิงแอดเดรสโดยตรง แต่แรมในพื้นที่นี้ (80H-0FFH) การอ้างอิงถึงจะต้องผ่านทางพอยน์เตอร์ (ตัวชี้ตำแหน่ง) นั่นคือเราต้องอ้างอิงแบบทางอ้อม (indirect)

2.3) หน่วยความจำสแต็ค

ตามหลักการเขียนโปรแกรม โปรแกรมจะต้องมีโปรแกรมย่อย ซึ่งสามารถจะถูกเรียกได้จากทุกที่ในโปรแกรมหลัก เพื่อที่หลังจากสิ้นสุดโปรแกรมย่อยจะกลับเข้าสู่การทำงานในโปรแกรมหลักในตำแหน่งที่ถูกคองนั้น จะต้องมีการเก็บค่าตำแหน่งอ้างอิงก่อนกระโดดออกไปจากโปรแกรมหลัก สำหรับจุดมุ่งหมายนี้ พื้นที่ส่วนหนึ่งในหน่วยความจำแรม ข้อมูลจะถูกเก็บลงในหน่วยความจำในพื้นที่ที่ว่างอยู่สุดท้ายของตำแหน่งหน่วยความจำใน นั่นคือการซ้อนทับกัน

2.4) การแบ่งจัดสรรพื้นที่ภายใน

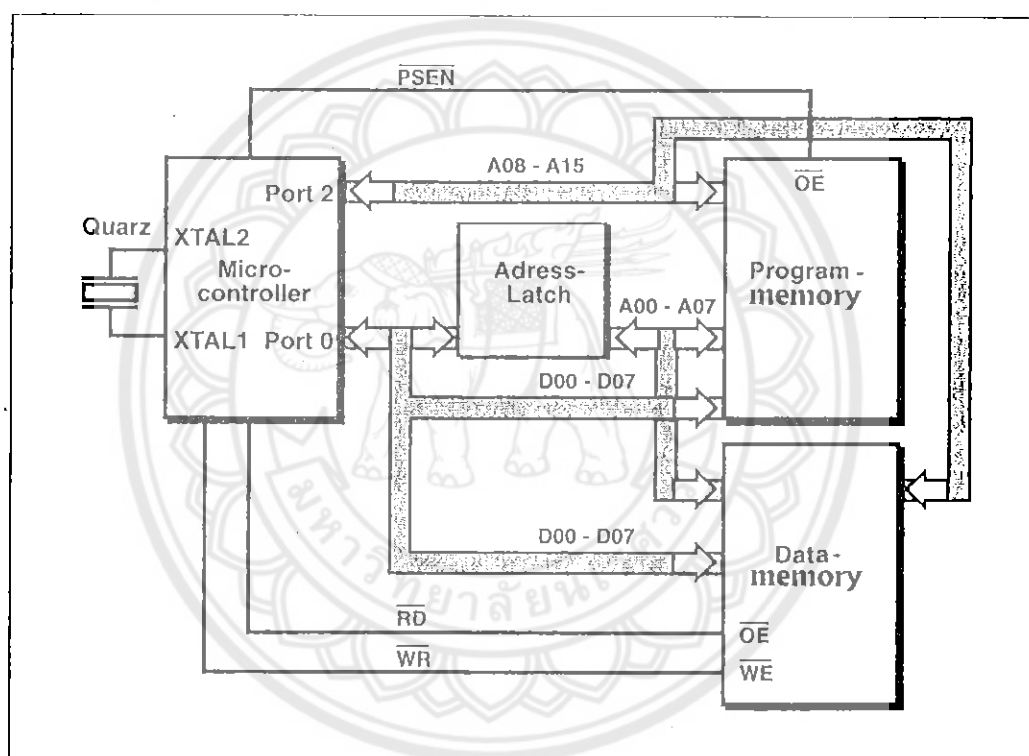
พื้นที่หน่วยความจำภายใน แรม (RAM) ของคอนโทรลเลอร์ 8051 จะเห็นจากรูปที่ 2.5 ซึ่งเราจะเห็นได้ว่าพื้นที่บางส่วนถูกมอนิเตอร์โปรแกรมของเราไว้ และไม่อนุญาตให้โปรแกรมใช้งานอื่น ๆ ใช้ได้



รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูลภายใน

2.2 บอร์ดไมโครคอนโทรลเลอร์ 8032

จากรูปลายวงจรของคอมพิวเตอร์บอร์ด (ภาคผนวก ก.) จะเห็นว่ามีอุปกรณ์ชิปมากมายที่เชื่อมต่ออยู่กับแอดเดรสบัสและบัสข้อมูลของไมโครคอนโทรลเลอร์เพื่อที่จะควบคุมการส่งผ่านข้อมูลระหว่างชิป ส่วนควบคุมจะต้องสร้างสัญญาณควบคุมขึ้นในระหว่างที่โปรแกรมโค้ดถูกทำงานด้วยไมโครคอนโทรลเลอร์ สัญญาณควบคุมต่าง ๆ จะถูกสร้างโดยอัตโนมัติ รูปแบบการทำงานกับหน่วยความจำจะมีอยู่ 2 ลักษณะ คือ การอ่านโปรแกรมโค้ดออกจากหน่วยความจำโปรแกรม และการอ่านและเขียนข้อมูลลงในหน่วยความจำข้อมูล



รูปที่ 2.6 การเชื่อมโยงติดต่อระหว่างหน่วยความจำโปรแกรมข้อมูลกับไมโครคอนโทรลเลอร์

จากรูป 2.6 แสดงให้เห็นถึงการเชื่อมโยงของหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลกับไมโครคอนโทรลเลอร์ สมมติว่า ถ้าเราใช้อีพีรอมเป็นหน่วยความจำโปรแกรมและแรมแทนที่หน่วยความจำข้อมูล เราสามารถอ่านข้อมูลมาได้เช่นเดียวกับอีพีรอมแต่สามารถจะเก็บข้อมูล (เขียนข้อมูล) ได้อีกด้วย สำหรับขั้นตอนในการอ่านและเขียน สัญญาณควบคุมต่าง ๆ ที่เกี่ยวข้องจะถูกผลิตออกมา

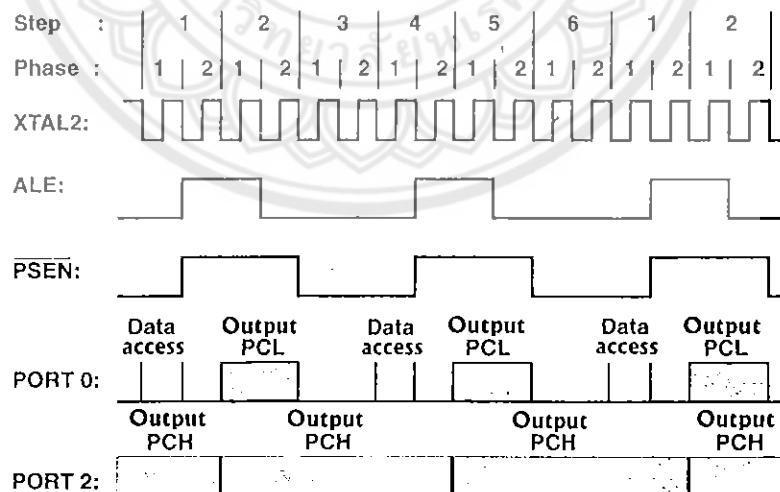
ส่วนควบคุมถูกควบคุมและให้จังหวะการทำงานด้วย X-TAL สัญญาณนาฬิกาที่สามารถถูกสร้างได้โดยแร่คริสตอลซึ่งเป็นแหล่งกำเนิดคลื่น ใน 1 วงรอบคำสั่งจะกินเวลายาวนาน 12 คาบเวลา

ของสัญญาณนาฬิกาในการประมวลผลส่วนใหญ่ของคำสั่งจะใช้เวลาประมาณนี้ ถ้าเราใช้คริสตอกลความถี่ 12 ใน 1 วงรอบคำสั่งจะใช้เวลา 1 μ S

เพื่อที่เราจะสามารถแสดงให้เห็นว่าช่วงไหนของเวลาที่ (หน่วยประมวลผลกลาง) ทำงานอะไร เราจะแบ่ง 1 วงรอบคำสั่งเป็น 6 สเต็ป แต่ละสเต็ปกินเวลา 2 คาบเวลาของสัญญาณนาฬิกา ทุก ๆ สเต็ปจะถูกแบ่งออกเป็น 2 เฟสอีกครั้ง นั่นหมายความว่า ความยาวของเฟสหนึ่ง ๆ จะเท่ากับ 1 คาบเวลาของสัญญาณนาฬิกา เราจะให้ลำดับโดยสเต็ปและเฟส นั่นคือเราจะสังเกตการณ์ตั้งแต่ S1P1 ถึง S6P2

2.2.1 สัญญาณ ALE

พอร์ต 0 ส่งไบต์ค่าเพื่ออ้างอิงตำแหน่งของหน่วยความจำภายนอกพอร์ต นอกจากนี้ยังทำหน้าที่อ่านข้อมูล ดังนั้น การใช้พอร์ต 0 นี้เพื่ออ้างอิงตำแหน่งหน่วยความจำ และรับส่งข้อมูลจึงต้องทำตามลำดับกัน (จะทำพร้อมกันไม่ได้) ตำแหน่งอ้างอิงจึงต้องถูกบันทึกไว้ก่อน ซึ่งเราเรียกว่า การแลตช์ข้อมูล จะกระทำที่ขอบขาขึ้นของสัญญาณ ALE (Address-Latch-Enable) นั่นคือ ตำแหน่งอ้างอิงหน่วยความจำโปรแกรมและข้อมูลทางไบต์ค่าได้ถูกวางไว้แล้วบนแอดเดรสบัสโดยอุปกรณ์ (ชิป) Latch ส่วนตำแหน่งอ้างอิงไบต์สูงจะถูกส่งออกมาที่พอร์ต 2 ตลอดเวลา หลังจากนั้นก็สามารถที่จะอ่านหรือเขียนข้อมูลลงบนหน่วยความจำที่อ้างอิงได้



รูปที่ 2.7 ช่วงเวลาของสัญญาณ ALE และ PSEN

สัญญาณ ALE จะถูกสร้างขึ้น 2 ครั้งใน 1 วงรอบคำสั่ง มีกรณีขกเว้น เมื่อทำคำสั่ง ซึ่งเป็นคำสั่งที่เกี่ยวข้องกับหน่วยความจำข้อมูลในกรณีนี้ จะหายไป 1 ครั้ง

ความถี่ของสัญญาณ ที่ไม่ได้ทำงานกับหน่วยความจำข้อมูลจึงเท่ากับ

$$f_{ALE} = \frac{f_{osc}}{2}$$

โดยที่ f_{osc} หมายถึง ความถี่สัญญาณนาฬิกา สัญญาณความถี่ของ f_{ALE} นี้ เรายังสามารถนำมาใช้ประโยชน์อย่างอื่นได้อีก

2.2.2 สัญญาณ PSEN

สัญญาณ PSEN (Program-Store-Enable) จะแอกทีฟที่ขอบขาลงในระหว่างที่ติดต่อกับหน่วยความจำโปรแกรม ถ้าดูจากผังเวลาแล้วจะเห็นได้ว่า สัญญาณ PSEN จะแอกทีฟหลังจากที่ได้แลตซ์ตำแหน่งอ้างอิงหน่วยความจำสิ้นสุดลงแล้ว PSEN จะเชื่อมต่ออยู่กับขา OE (เอาต์พุต-Enable) ในระหว่างที่ PSEN (Low) แอกทีฟหน่วยความจำโปรแกรมจะวางข้อมูลที่ถูกอ้างอิงถึงลงบนบัสข้อมูล ส่วนควบคุมจะรับข้อมูลผ่านมาจาก Port 0

สัญญาณ PSEN จะมีพฤติกรรมคล้ายกับสัญญาณ ALE คือ จะหายไประหว่างที่ได้คำสั่งเป็น ในกรณีที่ติดต่อกับหน่วยความจำโปรแกรมภายในของ 8051 สัญญาณ PSEN นี้จะไม่ถูกสร้างขึ้น

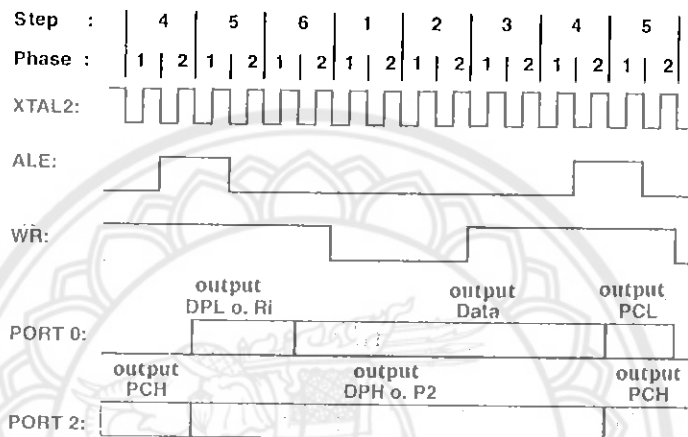
2.2.3 สัญญาณ WR

ส่วนควบคุมจะส่งสัญญาณ WR ในระหว่างที่มีการเขียนข้อมูลลงในหน่วยความจำข้อมูลภายนอก ตำแหน่งอ้างอิงของหน่วยความจำข้อมูลจะเหมือนกับการอ้างอิงหน่วยความจำโปรแกรมบนแอดเดรสบัส ในระหว่างที่ขอบขาลงของสัญญาณ ALE ตำแหน่งอ้างอิงจากไบต์ค่าที่พอร์ต 0 จะถูกเก็บไว้ในชิปแลตซ์ในไซเกิลถัดไปส่วนควบคุมจะให้สัญญาณ WR สู่ออกว่าต่ำ ซึ่งทำให้ขา WR ของหน่วยความจำข้อมูลภายนอกแอกทีฟและจะเขียนข้อมูลเข้าไปในตำแหน่งอ้างอิงที่ถูกอ้างอิงอยู่ไบต์สูงของแอดเดรส จะส่งออกมาทางพอร์ต 0 ตลอดช่วงเวลาของการเขียน

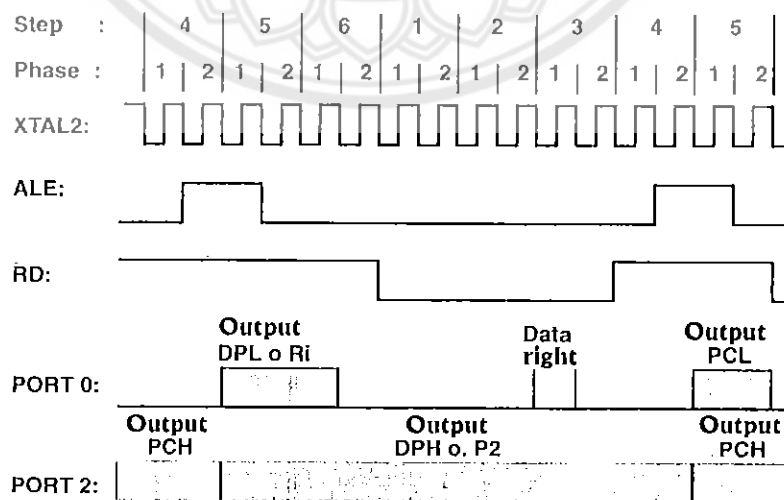
2.2.4 สัญญาณ RD

โปรเซสเซอร์จะสร้างสัญญาณ RD (Read Enable) ในขณะที่อ่านข้อมูลออกจากหน่วยความจำข้อมูลภายนอก หลังจากที่ไบต์ค่าอ้างอิงตำแหน่งได้ถูกแลตซ์แล้ว โปรเซสเซอร์จะทำให้ RD สู่ออกว่าต่ำ (Low) ดังนั้นที่ขา OE จึงแอกทีฟ หน่วยความจำข้อมูลก็จะวางข้อมูลในตำแหน่งที่ถูกอ้างอิงบนบัสข้อมูล และส่วนควบคุมก็จะอ่านข้อมูลจากบัสข้อมูลซึ่งผ่านมาจากพอร์ต 0 จากบัสล็อกของไคอะแกรมเวลาจะเห็นว่าสัญญาณ ALE แอกทีฟเพียงครั้งเดียวระหว่างทำคำสั่ง MOVX ตรงที่

S2P1 ที่ซึ่งให้ขอบขาสัญญาณลงของ ALE สัญญาณ RD หรือ WR จะแอกทีฟ ตรงจุดเวลานี้เองที่จะ
 ต้องไม่มีการเปลี่ยนแปลงของตำแหน่งอ้างอิงแอกเครสที่หน่วยความจำข้อมูล สัญญาณที่เวลานี้ต้อง
 หายไป เพื่อที่ในขณะที่คำสั่งจะมีเพียงหน่วยความจำข้อมูลเท่านั้นที่วางข้อมูลลงบนบัสข้อมูลไม่ใช่
 อีพีรอม



รูปที่ 2.8 การเขียนข้อมูลลงในหน่วยความจำข้อมูลภายนอก



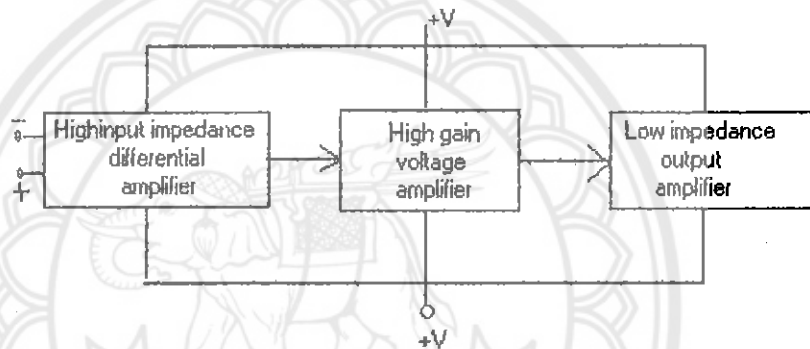
รูปที่ 2.9 การอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

2.3 ออปแอมป์

ออปแอมป์ คือ อุปกรณ์ชนิดหนึ่งออกแบบให้สามารถทำงานได้หลายรูปแบบ และยังมีความสะดวกในการใช้งานอีกด้วย โดยสามารถประกอบเป็นวงจรต่อร่วมกับอุปกรณ์ภายนอกเพียงไม่กี่ตัวเท่านั้น

2.3.1 คุณสมบัติทั่วไป

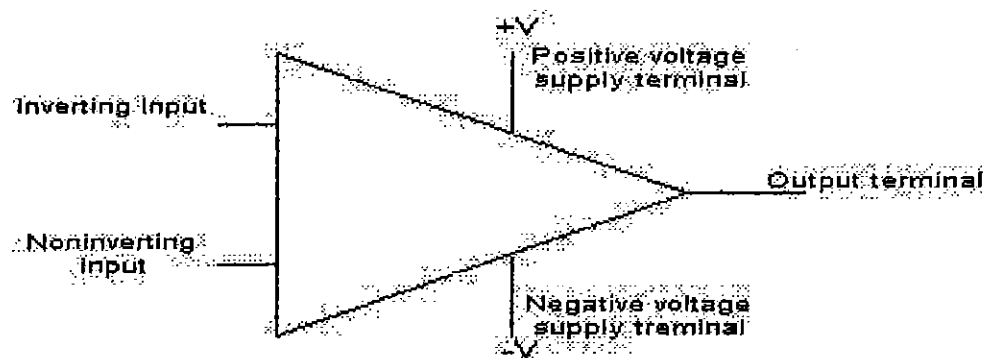
โดยทั่วไปแล้วออปแอมป์สามารถตรวจวัดได้ทั้งระดับไฟตรง ไฟสลับ และสามารถนำไปใช้ขยายสัญญาณ ออปแอมป์พื้นฐานประกอบไปด้วยวงจรภายในภาคต่างๆ ดังรูปที่ 2.10



รูปที่ 2.10 ส่วนประกอบภายในของออปแอมป์

2.3.2 คุณสมบัติที่สำคัญบางประการของออปแอมป์อุดมคติ

1. เนื่องจากอินพุตอิมพีแดนซ์ของออปแอมป์มีค่าสูงเป็นอนันต์ กระแสเข้าที่อินพุตจะต่ำจนเกือบเท่าศูนย์ หรือ ไม่มีกระแสอินพุตเข้าเลย
2. อัตราขยายขณะเปิดแอมป์ (ขณะที่ยังไม่มีการป้อนกลับ) จะมีค่าสูงมาก ซึ่งหมายความว่าแรงดันที่ขั้วอินพุตควรมีค่าใกล้เคียงศูนย์ เอาท์พุทอิมพีแดนซ์มีค่าต่ำมากจนไม่ทำตัวเป็นโหลดต่อภาคเอาท์พุทของวงจรขยาย



รูปที่ 2.11 แสดงลักษณะทั่วไปของออปแอมป์

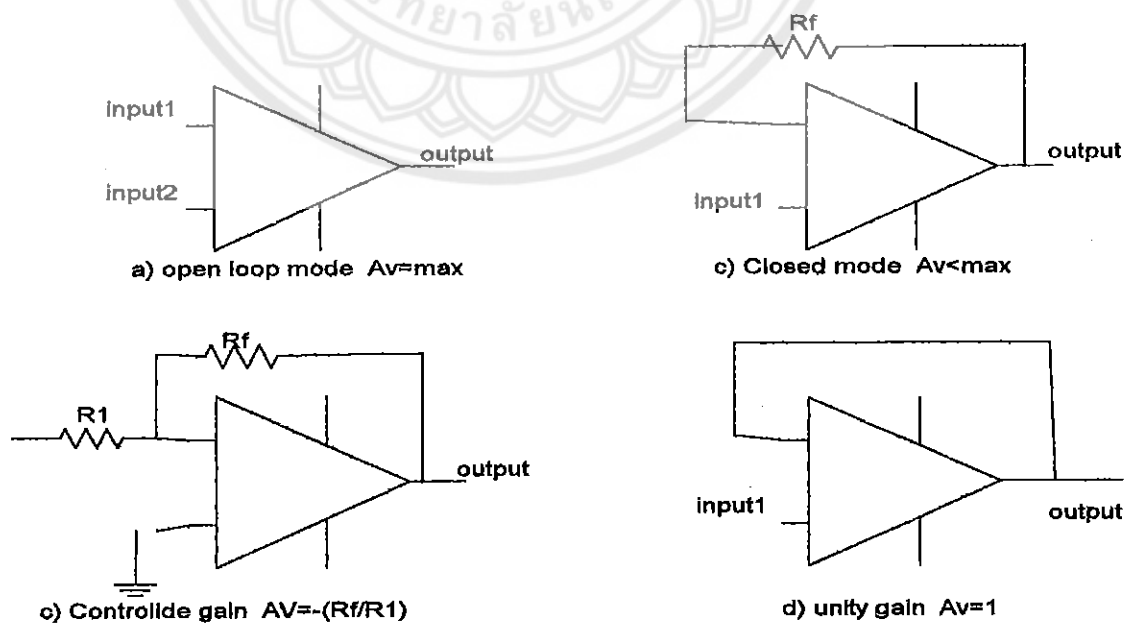
ขั้วอินพุตทั้งสองข้างของออปแอมป์มีลักษณะต่างกันดังนี้

- สำหรับขั้วลบ เมื่อป้อนไฟตรง หรือ ไฟสลับ เข้าไปในขณะที่ขั้วบวกต่อกับจุดอ้างอิงจุดหนึ่ง สัญญาณที่ออกมาที่เอาต์พุตจะกลับเฟสกับอินพุต 180 องศา
- ส่วนการป้อนสัญญาณที่ขั้วบวก เอาต์พุตจะมีเฟสตรงกับอินพุต ดังนั้นจึงกล่าวได้ว่า เครื่องหมายที่อินพุต คือการแสดงเฟสของเอาต์พุต

2.3.3 ลักษณะการทำงาน

ออปแอมป์ในอุดมคติจะมีอัตราขยายเป็นอนันต์ แต่ในทางปฏิบัติ อัตราขยายอาจมีค่าสูงสุดเพียง 10,000 ถึง 1,000,000 เท่า ซึ่งเรียกว่า อัตราขยายขณะลูปเปิด (A_v) ในขณะที่เกิดความแตกต่างของแรงดันเพียงเล็กน้อยระหว่างขั้วอินพุตทั้งสอง เอาต์พุตจะสามารถให้สัญญาณสูงขึ้นหลายเท่า (ตามค่าอัตราขยาย A_v) หากแต่จะถูกจำกัดด้วยขนาดของแหล่งจ่ายไฟเลี้ยงที่เราป้อนให้แก่ออปแอมป์ด้วย แต่ถึงอย่างไรก็ตาม เอาต์พุตก็จะไม่สามารถมีค่าสูงสุดเท่ากับแรงดันจากแหล่งจ่ายไฟเลี้ยงได้จริง โดยเอาต์พุตที่ได้สูงสุดประมาณ 90% ของแรงดันจากแหล่งจ่ายไฟเลี้ยงเท่านั้น

จากคุณสมบัติข้างต้น เราสามารถนำออปแอมป์ลูปเปิดไปใช้งานเป็นคอมพาราเตอร์ หรือ วงจรเปรียบเทียบแรงดันได้ โดยเอาต์พุตจะเปลี่ยนทันทีเมื่อมีความแตกต่างของแรงดันขึ้นระหว่างขั้วอินพุตของออปแอมป์

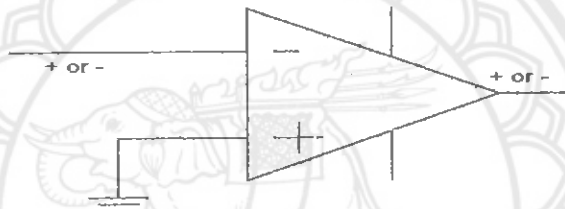


รูปที่ 2.12 ลักษณะออปแอมป์แต่ละชนิด

นอกจากนี้ จะพบว่าการใช้โอปแอมป์ในลักษณะของรูปปิด (มีการป้อนกลับ) จะทำให้โอปแอมป์มีประโยชน์สูงมากขึ้น การป้อนกลับในรูปตัวต้านทาน RF เพียงตัวเดียว ซึ่งมีผลทำให้วงจรมีค่าเสถียรภาพสูงขึ้นและมีสัญญาณน้อยลง ในขณะที่เดียวกัน อัตราขยายแรงดันจะ “ลดลง”

การใช้โอปแอมป์โดยมีการป้อนกลับสัญญาณเอาต์พุตกลับมายังอินพุตในวงจรรูป c นอกจากนี้ยังสามารถควบคุมอัตราขยายแรงดัน (ในขณะรูปปิด) ได้ โดยอาศัยตัวต้านทาน 2 ตัวเท่านั้น ทำให้อัตราขยายแรงดัน A_v มีค่าคงสมการ โดยที่เครื่องหมายแสดงถึงการกลับเฟสของเอาต์พุตเทียบกับอินพุต ส่วนวงจร d แสดงการป้อนกลับในกรณีที่ $A_v = 1$ ความสัมพันธ์ของขั้วต่างๆ ระหว่างอินพุตของโอปแอมป์ โดยแสดงได้ดังนี้

$$A_v = -\frac{R_f}{R_i}$$



รูปที่ 2.13 ความสัมพันธ์ระหว่างขั้วโอปแอมป์

จากรูป 2.13 ได้ข้อสรุปดังนี้

1. ถ้าขั้ว - มีศักดาสูงกว่าขั้ว + จะได้อาต์พุตเป็น -
2. ถ้าขั้ว - มีศักดาสูงกว่าขั้ว + จะได้อาต์พุตเป็น +

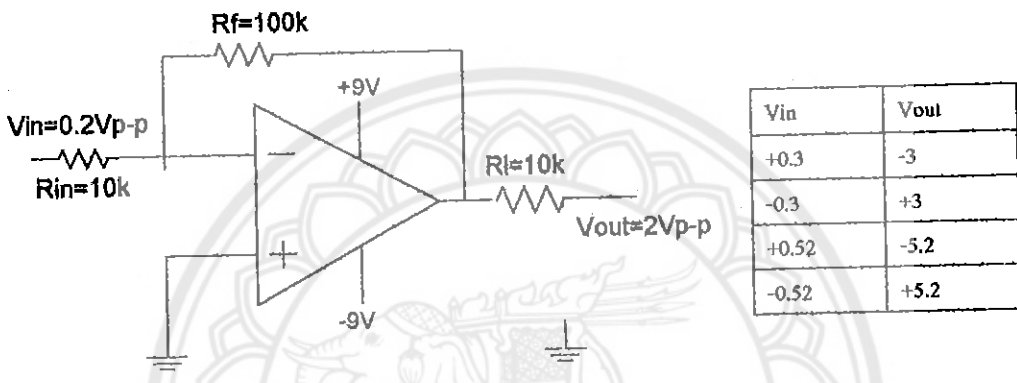
2.3.4 ชนิดของโอปแอมป์

ตาราง 2.2 จำนวนชนิดทั่วไปของโอปแอมป์

กลุ่มที่	ลักษณะการใช้งาน	คุณสมบัติพิเศษ
1	ใช้กับงานทั่วไป	ใช้งานได้ในช่วงความถี่ DC
2	ใช้กับสัญญาณ DC ได้ดี	มีค่าอินพุตอิมพีแดนซ์สูงมาก ทำให้กระแสไบแอสมีค่าต่ำ
3	ใช้กับสัญญาณ AC ได้ดี	มีแบนด์วิดท์กว้างมาก และอัตราสลับมีค่าสูง
4	ใช้กับแรงดันสูงและวงจรที่ค่อนข้างกำลังมาก	สามารถใช้บั๊ฟโวลต์ได้โดยตรง
5	ชนิดที่มีคุณสมบัติเฉพาะตัว	เช่นโอปแอมป์ชนิดที่ใช้กันโดยทั่วไป

2.3.5 วงจรขยายกลับเฟส

วงจรขยายจะหมายความว่าความถี่ของสัญญาณอินพุตที่น้อย ๆ มาเป็นสัญญาณเอาต์พุตที่มีขนาดสูงขึ้นกว่าเดิม ซึ่งสัญญาณที่ว่าอาจเป็นไปได้ทั้งแรงดันหรือกระแส วงจรขยายออปแอมป์มีอัตราขยายที่สูงมาก และลักษณะการขยายคงที่เสมอ (ในอุดมคติ) นอกจากนี้ยังสามารถควบคุมอัตราขยายได้



รูปที่ 2.14 วงจรขยายกลับเฟสแบบมาตรฐาน

จากรูปที่ 2.14 เป็นวงจรขยายกลับเฟสแบบลูปปิด เรียกว่าการป้อนกลับแบบลบ โดยนำเอาเอาต์พุตซึ่งกลับเฟสกับอินพุตมาป้อนที่จุดอินพุตอีกที ผลก็คือวงจรภายในออปแอมป์จะพยายามปรับเอาต์พุตจนกระทั่งความต่างศักย์ระหว่างอินพุตมีค่าเข้าใกล้ศูนย์

จากสมการของอัตราขยายแรงดัน Av

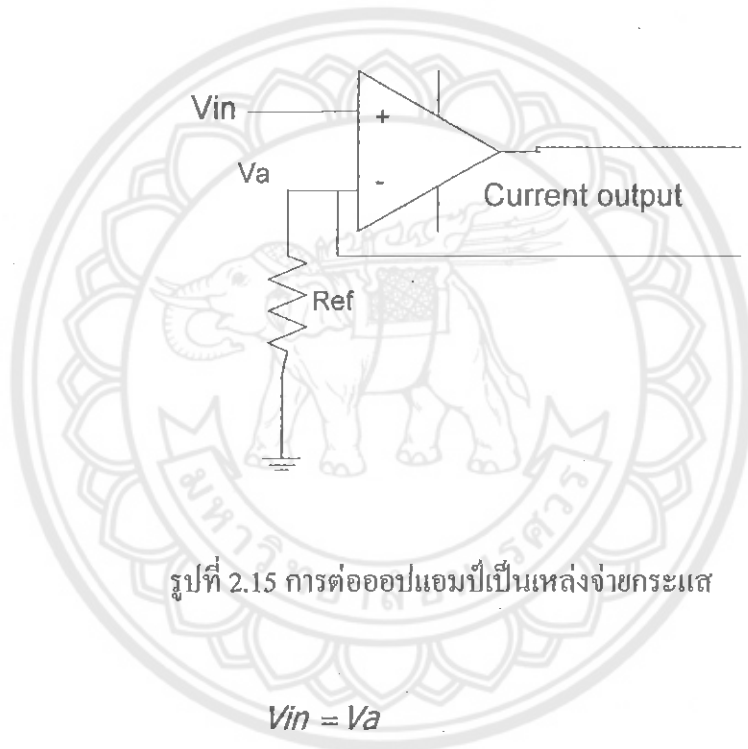
$$A_v = \frac{V_{out}}{V_{in}}$$

เราจะพิสูจน์ให้เห็นว่า ค่าขยายแรงดัน Av ของออปแอมป์เมื่อใช้ในโหมดของลูปจะ สามารถหาได้ด้วยสมการ

$$A_v = -\frac{R_f}{R_{in}}$$

2.3.6 วงจรแหล่งจ่ายกระแส

จากคุณสมบัติของออปแอมป์ที่แรงดันที่ขั้วบวกและลบมีแรงดันเท่ากันเพราะภายในออปแอมป์จะมีค่าอิมพีแดนซ์สูงมาก จากรูปที่ 2.15 จะเห็นได้ว่าเมื่อมีแรงดันเข้ามาเท่ากับ V_{in} จะทำให้ V_a มีค่าเท่ากับ V_{in} ด้วย และไม่มีกระแสไหลไปทางด้านเอาต์พุตเนื่องจากออปแอมป์มีอิมพีแดนซ์สูง เมื่อมีแรงดันตกคร่อม R_f จึงเกิดกระแสขึ้นตามกฎของโอห์ม ค่ากระแสที่เกิดขึ้นจะมีค่าเท่ากับ อัตราส่วนระหว่าง แรงดัน V_a กับค่าความต้านทาน R_f และตามกฎการไหลของกระแสของ เครฟฮอร์ฟจะเห็นได้ว่าที่จุด V_a จะมีกระแสไหลเข้าหนึ่งและไหลออกหนึ่งทาง ดังนั้นกระแสเอาต์พุตจึงมีค่าเท่ากับกระแสที่เกิดจากตัวต้านทาน ดังแสดงในสมการข้างล่างนี้

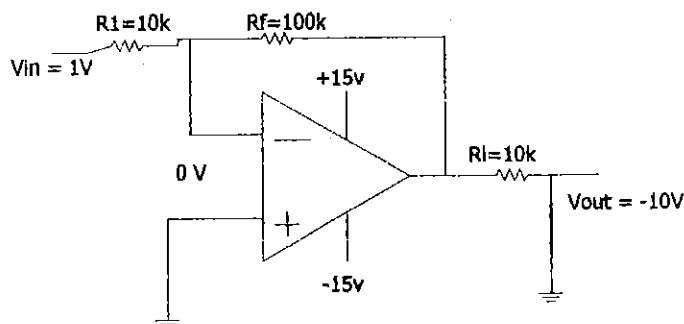


และ

$$\text{Current output} = \frac{V_a}{R_f}$$

วงจรที่กล่าวมานำมาใช้ประโยชน์ในการส่งสัญญาณได้ในระยะที่ไม่จำกัดและไม่ต้องกังวลเรื่องแรงดันตกในสายเนื่องจากการส่งแบบกระแส ดังนั้นที่ปลายทางไม่ว่าสายส่งสัญญาณจะยาวเท่าใดกระแสก็จะมีค่าเท่าเดิม

แต่วงจรก็ยังมีข้อเสียคือต้องมีการสร้างวงจรภาครับด้วยเนื่องจากเป็นค่ากระแสจึงต้องทำการเปลี่ยนเป็นค่าแรงดันก่อนที่จะนำไปใช้งาน โดยการใช้ความต้านทานคงที่ค่าหนึ่งมาให้กระแสไหลผ่านก็จะได้เป็นแรงดันนำไปใช้งานได้



รูปที่ 2.15 ความสัมพันธ์ระหว่างขั้ว

พิจารณาวงจรในรูป 2.15 จะได้ข้อสรุปคือ

1. ความต่างศักย์ระหว่างขั้วอินพุตเท่ากับศูนย์
2. กระแสเข้าขั้วอินพุตใดๆ เท่ากับศูนย์

โดยใช้หลักการข้อที่ 1 ซึ่งทำให้ขั้วอินพุตลบมีค่าศักดาเทียบเท่ากับกราวด์ เราสามารถเขียนได้ว่า กระแส I_{in} มีค่า ดังสมการ

$$I_{in} = \frac{V_{in}}{R_{in}} = \frac{1v}{10k} = 0.1mA$$

จากข้อสรุปที่ 2 เราจะพบว่า กระแสที่ผ่าน R_f ก็จะต้องเท่ากับกระแส I_{in} ดังนั้นเราจึงสามารถหาแรงดันที่เอาต์พุตเทียบกับกราวด์ได้ โดยที่กระแส I_f จะมีค่า

$$I_f = -\frac{V_{out}}{R_f}$$

และเนื่องจาก $I_{in} = I_f$

$$\frac{V_{in}}{R_{in}} = -\frac{V_{out}}{R_f}$$

$$A_v = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}}$$

ดังนั้นในการหา V_{out} ของวงจรออปแอมป์ เราจะได้

$$V_{out} = A_v V_{in} = -\frac{R_f}{R_{in}} V_{in}$$

โดยที่เครื่องหมายลบแสดงการกลับเฟสของเอาต์พุตเทียบกับอินพุต

2.4 วาล์ว

2.4.1 หน้าที่ของวาล์ว

1. ทำหน้าที่ปิด-เปิด
 - ประตุน้ำ
 - ปลั๊กวาล์ว
 - บอลล์วาล์ว
2. ทำหน้าที่ควบคุมอัตราการไหล
 - โกลบวาล์ว
 - วาล์วผีเสื้อ
 - ไคอะเฟรมวาล์ว
3. ทำหน้าที่กั้นการไหลกลับ
 - เช็ควาล์ว
4. อื่น ๆ
 - คอนโทรลวาล์ว
 - โซลินอยด์วาล์ว

ต่อไปนี้จะพูดถึงวาล์วแต่ละแบบว่ามีโครงสร้างอย่างไร ทำงานอย่างไร ระบบกันซึมและคุณสมบัติทั่ว ๆ ไปเป็นอย่างไร

2.4.2 ประตุน้ำ

ประตุน้ำเป็นที่นิยมใช้กันอย่างแพร่หลายที่สุดในกลุ่มวาล์วที่ทำหน้าที่ประเภทเดียวกัน (ปลั๊กและบอลล์วาล์ว) ประตุน้ำใช้ทำหน้าที่ปิดเปิดเท่านั้น ไม่ควรที่จะเปิดครึ่ง ๆ กลาง ๆ จะต้องเปิดให้กว้างสุด ทั้งนี้เพราะโครงสร้างของประตุน้ำก่อให้เกิดการขัดสีที่ขอบของประตุน้ำได้ (ถ้าเปิดครึ่ง ๆ ไว้) ถึงแม้จะเป็นที่นิยมมากก็ตาม แต่ก็ยังมีข้อจำกัดในการใช้

- ชิ้นส่วนที่ทำหน้าที่เป็นลิ้น

โดยทั่ว ๆ ไป ลิ้นจะเป็นลิ้นคั่น พร้อมทั้งมีร่องลิ้น (หรือร่องประตุน้ำ) เอียง (Incline seat) แบบนี้เมื่อใช้ไปนาน ๆ จะสึกได้ ทำให้ปิดไม่สนิท ถ้าใช้ลิ้นที่มีความยืดหยุ่นได้ (flexible wedge) โดยทำเป็นรูปโครงสร้างแบบตัว “H” หรือ ตัว “C” (ดูรูปที่ 2.16) ก็สามารถลดอัตราการสึกหรอไปได้และไม่ต้องคำนึงถึงว่าลิ้นจะสอคบิดตรงแนวหรือไม่ ลิ้นที่เป็นแบบลิ้นยังมีอีกแบบคือ แบบแยกลิ้น (ดูรูปที่ 2.16) ชิ้นหนึ่งจะเป็นครึ่งทรงกลมยื่นออกสวมอยู่ในลิ้นอีกชิ้นหนึ่งที่

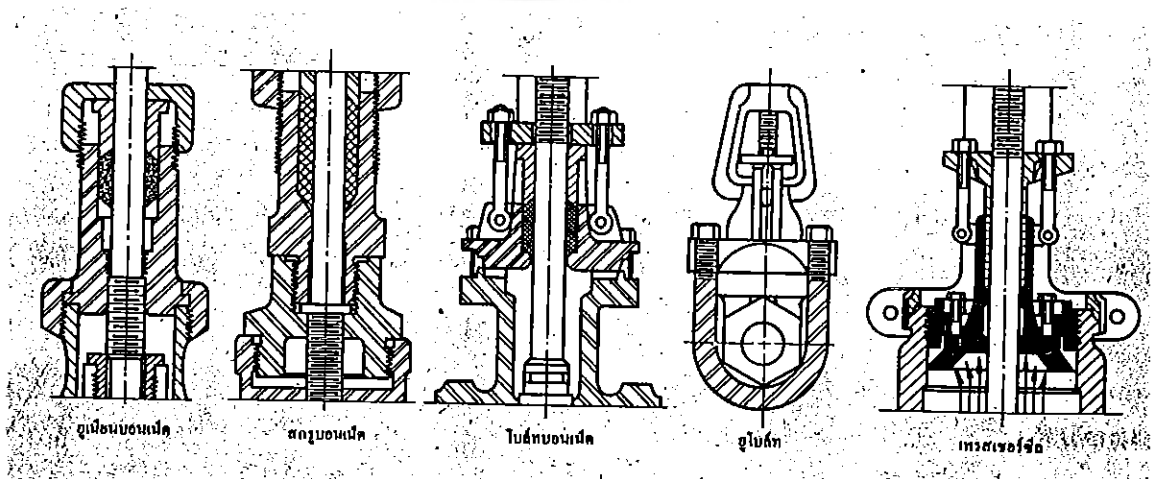
-ชิ้นส่วนที่ไปเปิดปิดลิ้น

ชิ้นส่วนที่ไปเปิดปิดลิ้นคือ ก้านวาล์ว (Stem) นั่นเอง ปลายก้านวาล์วข้างหนึ่งจะจับกับลิ้น อีกข้างหนึ่งยื่นออกมาติดกับส่วนที่ใช้หมุน (hand wheel) รูปแบบของเม็คแคนิซึมก็มีหลายแบบด้วยกัน โดยทั่วไปมี 4 แบบ เท่าที่เห็นในรูปที่ 2.18 ทางซ้ายสุดเป็นแบบสกรูของก้านวาล์วอยู่ภายใน (inside screw) ที่หมุน (hand wheel) จะขันแน่นติดกับก้านวาล์ว ดังนั้นเวลาเปิดปิดวาล์ว ก้านวาล์ว และที่หมุนจะเคลื่อนที่ไปด้วยกัน ถัดมาเป็นแบบสกรูของก้านวาล์วอยู่ภายนอก ที่หมุนขันติดกับตัววาล์ว เวลาเปิดปิดก้านวาล์วจะเคลื่อนที่ขึ้นลง โดยที่ที่หมุนอยู่กับที่ ถัดมาเป็นแบบสกรูของก้านวาล์วอยู่ภายในอีกเช่นกัน แต่สกรูจะไปขันกับลิ้น ก้านวาล์วอยู่กับที่ที่ที่หมุนขันติดกับก้านวาล์ว เวลาเปิดปิดจึงมีแต่ลิ้นเท่านั้นที่เคลื่อนที่ขึ้นลง แบบนี้ไม่ค่อยนิยมใช้ รูปสุดท้ายไม่ใช่เกลียว แต่ใช้คันโยกชักเข้าชักออก แบบนี้จะเปิดหรือปิดได้รวดเร็วกว่า 3 แบบแรก การที่จะเลือกใช้แบบไหนนั้นขึ้นกับสภาพรอบ ๆ ถ้าสภาพรอบ ๆ มีความกัดกร่อนมาก เช่น ใช้ในที่ใกล้ทะเล เป็นต้น จะใช้แบบสกรูภายในเพื่อป้องกันการกัดกร่อนที่สกรูของก้านวาล์ว ส่วนแบบสกรูภายนอกก็มีประโยชน์ตรงที่สามารถหล่อลิ้นได้สะดวก

-ระบบกันซึม (Sealing Methods)

ประตูน้ำจะต้องมีการป้องกันการซึม 4 จุดด้วยกัน โดย 3 จุดแรก จะกันของเหลวออกนอกวาล์วคือ ที่ข้อต่อระหว่างตัววาล์วกับบอนเน็ต (Bonnet ก็คือ ตัวที่ขีดยก้านวาล์ว) 2 จุด และอีกจุดคือ ที่ซึ่งออกจากก้านวาล์ว จุดที่เหลือคือ ที่ลิ้นของวาล์ว

ลักษณะของบอนเน็ตมีหลายแบบด้วยกัน (รูปที่ 2.19) แบบยูนิเยนและสกรูบอนเน็ตมักใช้กับวาล์วตัวเล็ก แบบโบลท์บอนเน็ตใช้กับวาล์วตัวใหญ่ แบบเพรสเชอร์ซีลใช้กับความดันและอุณหภูมิสูง



รูปที่ 2.19 บอนเน็ตแบบต่าง ๆ

การกันรั่วที่ลิ้นมี 2 แบบด้วยกัน ระหว่างโลหะต่อโลหะ และระหว่างโลหะกับวัสดุหุ่่นแบบระหว่างโลหะต่อโลหะจะมีความแข็งแรง แต่เมื่อใช้ไปสักพักผิวสัมผัสระหว่างหน้าโลหะจะเสียด ใช้การไม่ได้ แต่ถ้ามีการใช้โลหะที่มีความแข็งต่างกัน ก็ลดการสึกหรอได้บ้าง สำหรับแบบโลหะต่อวัสดุหุ่่นนั้นใช้กับอุณหภูมิสูงมาก ๆ ไม่ได้

-ลักษณะอื่น ๆ

ประตูน้ำมีข้อเสียตรงที่มีน้ำหนักมากและกินที่มากจึงต้องการที่รองรับน้ำหนักด้วย (ถ้าจำเป็น) การติดตั้งและซ่อมบำรุงยากลำบาก

2.4.3 ปลั๊กวาล์ว (Plug Valve)

-ชิ้นส่วนที่ทำหน้าที่เป็นลิ้น

ปลั๊กวาล์วทำหน้าที่เช่นเดียวกับประตูน้ำ คือ ปิดและเปิดเท่านั้น แต่ปลั๊กวาล์วยังใช้กำหนดเส้นทางการไหลได้ โครงสร้างของตัวปลั๊กมีทั้งแบบกรวยและทรงกระบอก รูที่เจาะผ่านกันมีทั้งวงกลมและสี่เหลี่ยม บางทีก็ทำให้คอคอดกึ่งเพื่อลดขนาดของตัววาล์ว

-ชิ้นส่วนที่ใช้หมุนปลั๊ก

วาล์วขนาดเล็กจะใช้ก้านหมุน (หมุนที่กึ่งกลางแล้วแต่ชนิดของรูเจาะผ่าน) แรงบิด (Torque) ที่ใช้หมุนจะมากขึ้นเมื่อเป็นวาล์วตัวใหญ่และมีความดันสูง บางที่อาจต้องใช้เกียร์ทดแรง

-ระบบกันซึม

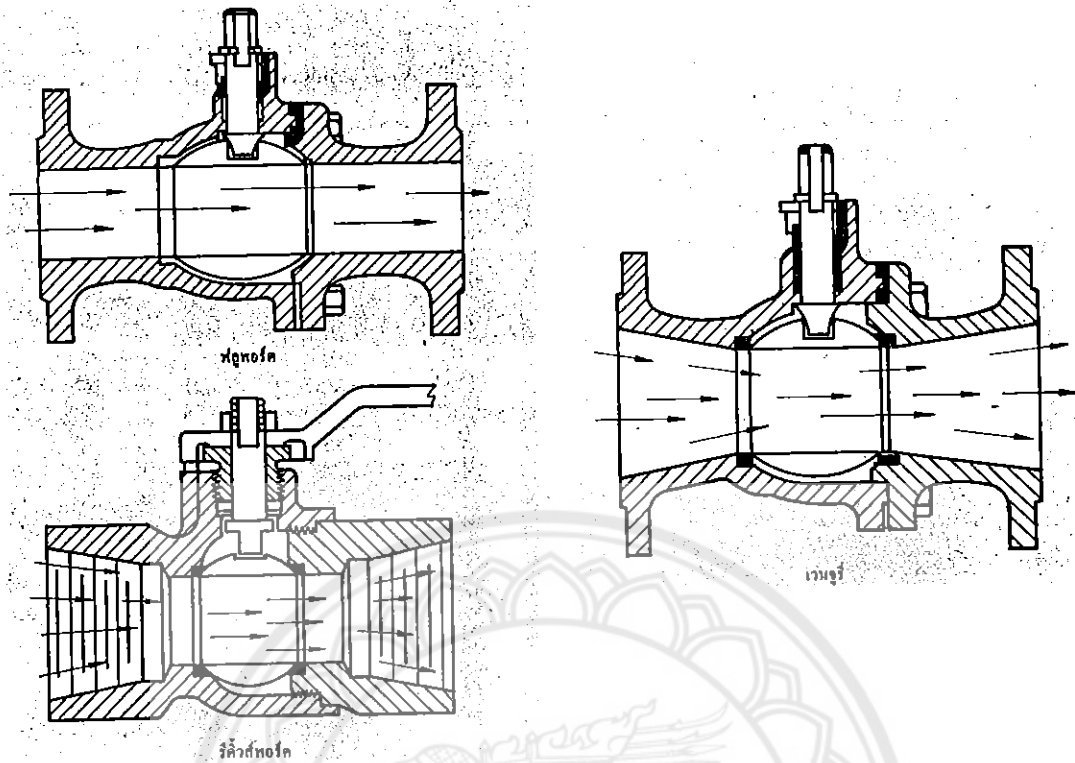
ระบบกันซึมก็เหมือนกับของประตูน้ำ ที่ก้านวาล์วจะใช้โอ-ริง (O-ring) แทนตัวอัดแพคกิ้ง (grand pack) ได้ ซึ่งสะดวกมาก การกันรั่วที่ปลั๊กมี 2 แบบ คือ แบบหล่อลิ้นและแบบปลอก (Sleeve) น้ำมันหล่อลิ้นที่อัดเข้าไปตามร่องของปลั๊กจะทำหน้าที่หล่อลิ้นและกันรั่วไปในตัว แบบปลอกนั้นจะอาศัยความยืดหยุ่นของตัวปลอกทำหน้าที่กันซึม วัสดุที่ใช้ทำปลอก เช่น ฟลูออโรคาร์บอน TFE ซึ่งไม่ทำปฏิกิริยากับของเหลว (มีบางชนิดเท่านั้น) TFE ยังมีค่าสัมประสิทธิ์ของความเสียดทานน้อยมาก จึงเท่ากับทำหน้าที่หล่อลิ้นแบบแห้งภายในตัว

-คุณสมบัติอื่น ๆ ปลั๊กมีขนาดเล็กกระทัดรัด น้ำหนักเบา ใช้ที่น้อย

2.4.4 บอลวาล์ว (Ball Valve)

-ชิ้นส่วนที่ทำหน้าที่เป็นลิ้น

ถึงแม้โอกาสที่จะใช้บอลวาล์วนั้นมีน้อยก็ตาม แต่บอลวาล์วก็มีใช้กันอย่างแพร่หลายนั้นเป็นเพราะการวิวัฒนาการด้านวัสดุพวก อีลาสโตเมอร์ และพลาสติกประกอบกับการค้นพบวิธีการผลิตลูกโลหะกลมด้วยกรรมวิธีที่ง่าย และราคาถูกพอสมควร บอลวาล์วนั้นปรับปรุงมาจาก



รูปที่ 2.20 รูปบอลล์วาล์ว แสดงการเจาะรูผ่านลูกบอลล์

ปลั๊กวาล์ว คือแทนที่จะเป็นปลั๊กก็ใช้ลูกบอลโลหะแทน โดยเจาะรูตลอดแกนหนึ่ง (รูปที่ 2.20) บอลล์วาล์วก็มีชนิดหลายทางด้วยเหมือนกัน

การเจาะรูผ่านลูกบอลมี 3 ลักษณะ คือ แบบฟูลพอร์ท (fullport) รีดิวซ์พอร์ท (reduced port) และแบบเวนจัวร์ แบบฟูลพอร์ท หมายความว่า เจาะรูให้เส้นผ่านศูนย์กลางเท่ากับเส้นผ่านศูนย์กลางของท่อ ส่วนแบบรีดิวซ์และเวนจัวร์นั้น จะเจาะรูให้เส้นผ่านศูนย์กลางเล็กกว่าของท่อเพื่อลดขนาดของตัววาล์ว ค่าความดันตกทั้ง 3 แบบ แตกต่างกันไป

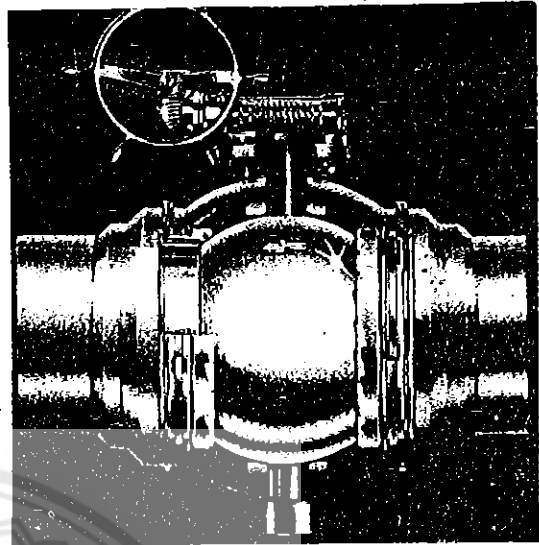
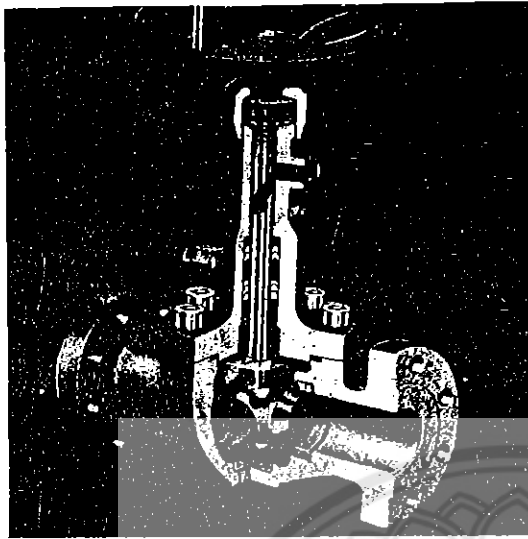
-ส่วนที่ใช้หมุนบอลล์

ก้านวาล์วของบอลล์วาล์วก็เหมือนกับของปลั๊กวาล์ว วาล์วตัวเล็กใช้ก้านหมุน ตัวใหญ่ก็ใช้เกียร์ทด

ในรูปที่ 2.21 เป็นแบบหนึ่งของวิธีการหมุนลูกบอลล์โดยอาศัยกลไกของแคม (cam mechanism) ในการเปลี่ยนการเคลื่อนที่จากขึ้นลงมาเป็นการเคลื่อนที่แบบหมุน จากรูปจะเห็นว่า ก้านวาล์วจะเคลื่อนที่ขึ้นลงจากการหมุนเกลียว (เหมือนของประตุน้ำ) ที่ก้านวาล์วนั้นจะมีร่องแคม ซึ่งจะทำให้ก้านวาล์วนั้นหมุนในขณะที่เคลื่อนที่ขึ้นลงด้วย

-ระบบกันซึม

ระบบกันซึมที่ก้านวาล์วมีตั้งแต่ใช้โอ-ริง (O-ring) จนถึงการใช้แพคกิ้งแบบธรรมดา บอลล์วาล์วมีทั้งแบบบอลล์ลอย (floating ball) และ ทรันชัน เม้าท์เต็ด (trunnion mounted) แบบบอลล์ลอย



รูปที่ 2.21 วาล์วที่ใช้กลไกของแควมทำให้ก้านวาล์วเคลื่อนที่

รูปที่ 2.22 บอลล์วาล์วทรันชันเม้าต์ (Trunnion mounted)

นั่นถูกบอลล์จะถูกความดันดันให้ติดขอบด้านทางออก ซึ่งความดันมากก็ยิ่งปิดสนิท ดังนั้นเมื่อความดันด้านทางเข้าและออกต่างกันน้อยเกินไปจะปิดไม่สนิท ในกรณีนี้จะใช้แบบปาวาล์วที่มีความยืดหยุ่นมากขึ้นและให้บอลล์ลอยค้ำขึ้น

แบบ ทรันชัน เม้าท์เด็คนั้น ถูกบอลล์จะถูกยึดทั้งด้านบนและล่าง ดังรูปที่ 2.22 ปาวาล์วทางด้านเข้าและออกเป็นคนละชั้น ไม่ติดกัน บางครั้งก็ใส่สปริง ถ้าใช้กับความดันต่ำ

บอลล์วาล์วนั้นใช้กับอุณหภูมิต่ำ ๆ ไม่ได้ เพราะวัสดุที่ใช้เป็นระบบกันซึมจะเสีย บอลล์วาล์วไม่ต้องมีการหล่อลื่น

-คุณสมบัติอื่น ๆ

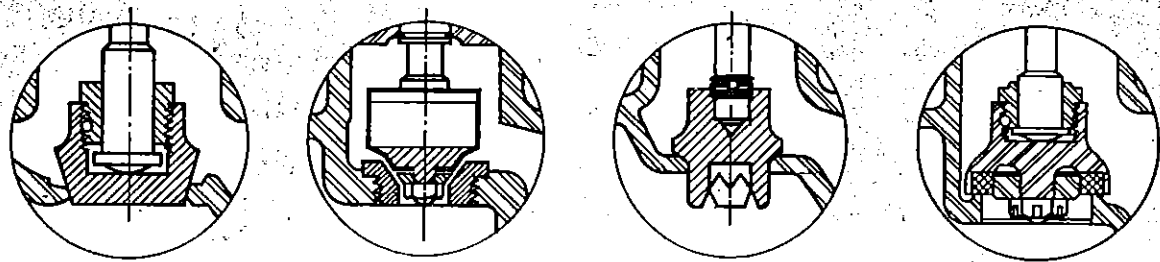
บอลล์วาล์วเล็กกระทัดรัด เมา ใช้ที่น้อยกว่าประตูน้ำมาก

2.4.5 โกลบวาล์ว (Globe Valve)

-ชิ้นส่วนที่ทำหน้าที่เป็นลิ้น

โกลบวาล์วนั้นสามารถทำหน้าที่ควบคุมอัตราการไหลได้ โครงสร้างของลิ้นนั้นมีหลายแบบด้วยกัน จากรูปที่ 2.23

ของเหลวจะไหลจากล่างขึ้นบน (โดยปกติจะมีถูกสรรจิศทางการไหลติดอยู่ที่ตัววาล์วเสมอ เพื่อป้องกันการใส่ผิข้าง) พื้นที่หน้าตัดของช่องผ่านจะเปลี่ยนตามที่เราขันก้านวาล์ว จึงสามารถควบคุมการไหลได้ "ความดันลด" (ผ่านโกลบวาล์วมี่ค่ามากพอๆกัน จึงไม่ควรใช้โกลบวาล์วสำหรับ



ลิ้นแบบปลั๊ก

ลิ้นแบบวอร์ค

ลิ้นเวทเพื่อคู่ปากลิ้นได้

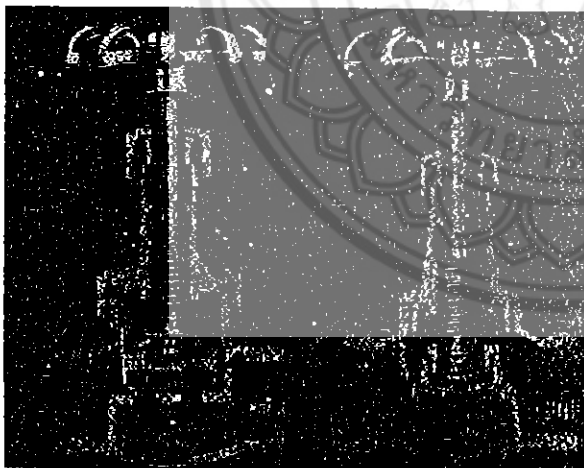
ลิ้นประกอบ

รูปที่ 2.23 โครงสร้างแบบต่าง ๆ ของลิ้น

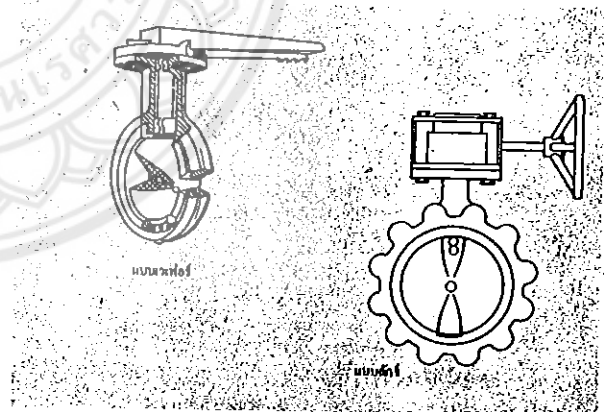
ทำหน้าที่ปิดเปิดอย่างเฉียบพลันเหมือนประตุน้ำ (ประตุน้ำมีความดันตลน้อยกว่า แต่ไม่สามารถควบคุมอัตราการไหลได้ โกลบวาล์วควบคุมการไหลได้ แต่ค่าความดันตลมากจึงไม่ควรให้ทำหน้าที่ปิดเปิดเท่านั้น)

-การดึงการปิดเปิดของลิ้น

วิธีการปิดเปิดก็เหมือนของประตุน้ำ ดูรูปที่ 2.24



รูปที่ 2.24 ส่วนประกอบของโกลบวาล์ว



รูปที่ 2.25 ลักษณะของลิ้น

-ระบบกันซึม

ระบบกันซึมของโกลบวาล์วก็เหมือนกับของประตุน้ำทุกประการ นอกจากที่ลิ้นเท่านั้น ที่ผิวสัมผัสของลิ้นกับร่องลิ้น (valve and valve seat) มีทั้งเป็นโลหะต่อกับโลหะ โลหะต่อกับวัสดุหุ่ยนต์ต่อโลหะ แบบหลังเหมาะกับความดันต่ำและของเหลวมีสารแขวนลอยปะปนมาด้วย ร่องลิ้น

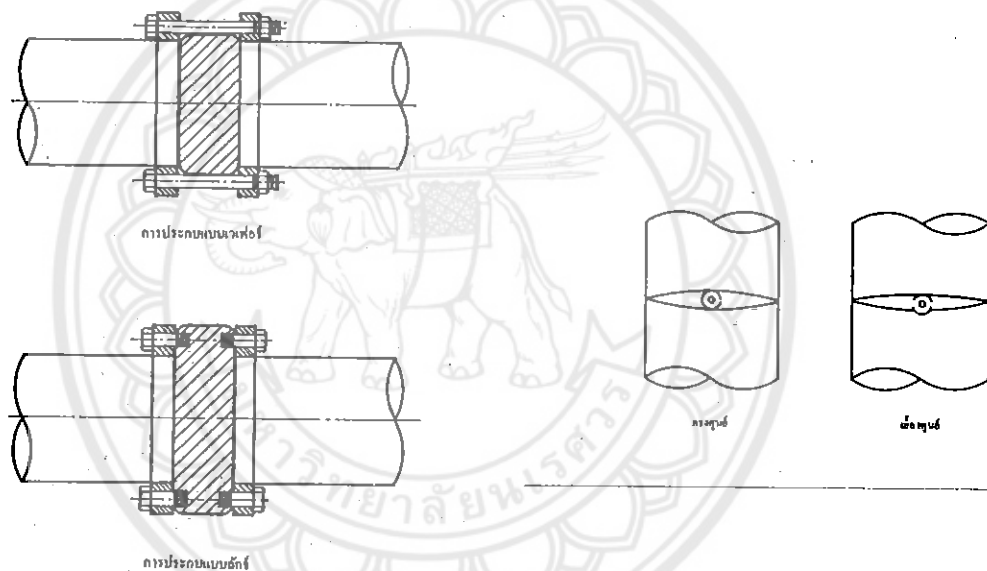
นั้นบางทีก็ได้รับการออกแบบให้สามารถเปลี่ยนได้ด้วย เช่น V-port type plug disk ในรูปที่ 2.24

-คุณสมบัติอื่น ๆ

โกลบวาล์วมีน้ำหนักมาก และใช้ที่มากพอ ๆ กับประตูน้ำ

2.4.6 วาล์วผีเสื้อ (Butterfly Valve)

แผ่นงานหรือปีกผีเสื้อที่ใช้เป็นลิ้นปิดเปิดนั้น มีแกนหมุนต่อกับก้านหมุน ในบางแบบแกนหมุนจะเอียงศูนย์กลางเล็กน้อยเพื่อให้ปิดได้สนิท ดังรูปที่ 2.26



รูปที่ 2.26 วาล์วผีเสื้อขนาดเล็ก

รูปที่ 2.27 การประกอบของแบบเวเฟอร์

ข้อต่อจะเป็นแบบเกลียวตัวใหญ่จะใช้หน้าแปลนประกบ มี 2 แบบด้วยกัน คือ แบบเวเฟอร์ (Wafer) และแบบตั๊ก (lug)

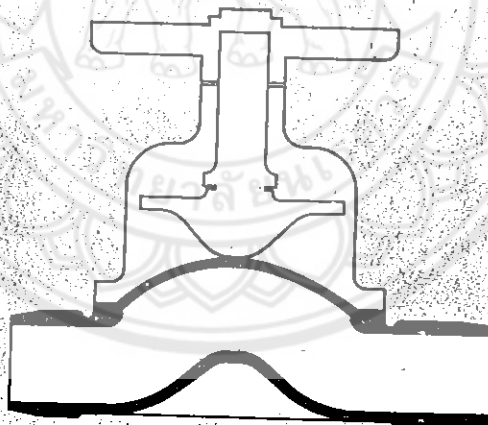
แบบเวเฟอร์นั้น เมื่อเวลาจะซ่อมท่อหรือส่วนที่อยู่ถัดจากตัววาล์วไป ซึ่งถ้าจะต้องถอดออกแล้วของเหลวที่อยู่ด้านหลังของวาล์วจะต้องปล่อยทิ้งได้ (หมายความว่าต้องถอดตัววาล์ว) แต่ในกรณีที่ต้องเก็บของเหลวทางด้านหลังของวาล์ว (หมายความว่าถอดวาล์วไม่ได้เพราะต้องใช้วาล์วนั้นเป็นตัวปิดของเหลวด้านหลัง) จะต้องใช้แบบตั๊ก เพราะแบบตั๊กนั้น ใช้โบลต์ขันติดกับหน้าแปลนข้างหนึ่งข้างใดก็ได้

วาล์วผีเสื้อมีข้อได้เปรียบกว่าวาล์วอื่น ๆ หลายประการด้วยกัน คือ น้ำหนักเบา ติดตั้งและซ่อมแซมสะดวกเหมาะกับระบบควบคุมอัตโนมัติ เพราะเพียงหมุน 90 องศา ก็ปิดหรือเปิดเต็มที่ แต่ในระหว่าง 0-90 องศา ก็สามารถควบคุมอัตราการไหลได้ (ในขณะที่โกลบวาล์วจะต้องหมุนหลายรอบทีเดียว) “ความดันลด” ผ่านวาล์วก็น้อยกว่าโกลบวาล์วมาก สามารถใช้กับของเหลวข้น ๆ ก็ได้ เพราะไม่มีร่องวาล์ว แต่เป็นบ่าวาล์ว สิ่งสกปรกจึงไม่จับเกาะ

วาล์วผีเสื้อมีข้อเสียตรงที่ใช้กับอุณหภูมิสูง ๆ ไม่ได้ เพราะบ่าวาล์วไม่สามารถทนความร้อนสูง ๆ ได้ แต่ก็มีความทนทานเหมือนกัน แต่ไม่สามารถปิดสนิทได้ จึงใช้ทำหน้าที่ควบคุมอัตราการไหลเท่านั้น

2.4.7 ไดอะแฟรมวาล์ว (Diaphragm Valve)

ไดอะแฟรมวาล์วประกอบด้วยแผ่นไดอะแฟรมซึ่งสามารถบิดได้ ทำหน้าที่เป็นลิ้นปิดเปิด ไดอะแฟรมวาล์วไม่ต้องมีระบบกันซึมที่ก้านวาล์ว ระบบกันซึมของบอนเน็ต (bonnet) ก็ใช้แผ่นไดอะแฟรมนั่นเอง

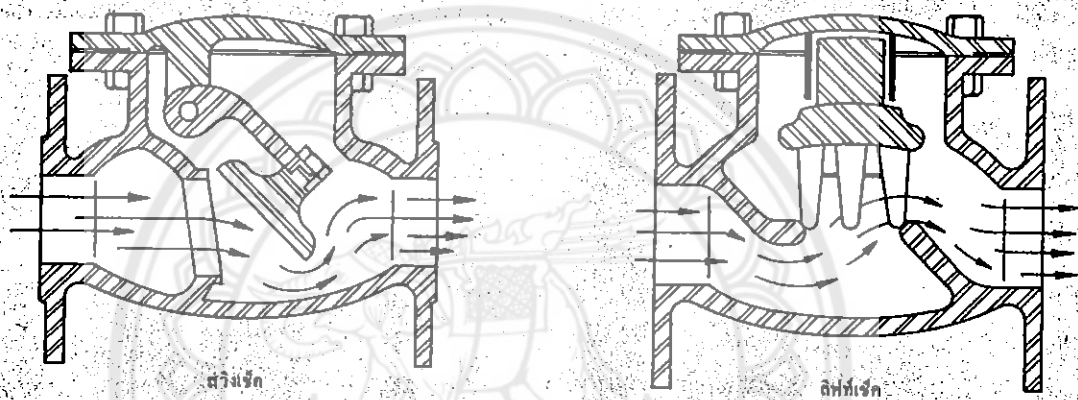


รูปที่ 2.28 ไดอะแฟรมวาล์ว

ไดอะแฟรมวาล์วเหมาะกับของเหลวที่มีความหนืดสูง มีสารแขวนลอยและกักร่อน ในกรณีที่ใช้กับของเหลวที่กร่อน แผ่นไดอะแฟรมจะทำจากเทฟลอน และตัววาล์วทำด้วยอีลาสโตเมอร์ พลาสติก แก้ว และเหล็กสแตนเลส

2.4.8 เช็ควาล์ว (Check Valve)

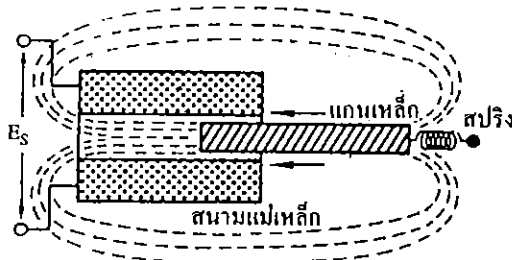
เช็ควาล์วทำหน้าที่ป้องกันการไหลกลับ มีโครงสร้างดังรูปที่ 2.29 ซึ่งเป็นเพียง 2 แบบ ในหลาย ๆ แบบ แบบสวิงที่เห็นในรูปนั้นประกอบด้วยลิ้น ซึ่งปิดได้ข้างเดียว (อีกด้านหนึ่งติดบ่าลิ้น) เมื่อของเหลวหยุดไหล ลิ้นจะตกมาปิดโดยอาศัยน้ำหนักของลิ้น และเมื่อมีของเหลวไหลกลับทิศ ก็จะมีแรงดันปิดลิ้นไว้ และแบบลิฟต์เช็ควาล์วก็อาศัยหลักในการทำงานเดียวกัน



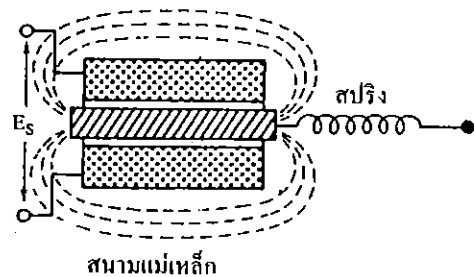
รูปที่ 2.29 เช็ควาล์วแบบสวิงและลิฟต์เช็ควาล์ว

2.5 ขดลวดโซลินอยด์ (Solenoid)

2.5.1 หลักการทำงาน



รูปที่ 2.30 สภาวะปกติ



รูปที่ 2.31 สภาวะทำงาน

จากรูปขดลวดโซลินอยด์ (Solenoid) จะประกอบด้วยขดลวดและแกนเหล็กซึ่งเคลื่อนที่ได้ สอดอยู่ในรูตรงกลางของขดลวด เมื่อต่อแรงดันไฟฟ้าให้กับขดลวดโซลินอยด์ กระแสไฟฟ้าจะไหล ผ่านในขดลวดทำให้เกิดเส้นแรงแม่เหล็กขึ้นรอบ ๆ ขดลวด เส้นแรงแม่เหล็กจะพยายามวางแนวเส้น ทางเดินระหว่างขั้วแม่เหล็กที่สั้นสุดเท่าที่จะเป็นไปได้ เส้นแรงแม่เหล็กที่เกิดขึ้นนี้จะทำให้เกิดแรง ดึงบนแกนเหล็กที่เคลื่อนที่ได้ และจะเอาชนะแรงสปริงดึงแกนเหล็กไปอยู่ตรงกลางของขดลวด ดัง รูปที่ 2.31 (สภาวะทำงาน) เมื่อตัดแรงดันไฟฟ้าออกจากวงจรก็จะมีกระแสไหลในขดลวด โซลินอยด์ โซลินอยด์ก็จะหมดสภาพเป็นแม่เหล็ก แรงดึงเนื่องจากอำนาจแม่เหล็กไฟฟ้าจะหมดไป แรงสปริงจะดึงแกนเหล็กให้กลับมากอยู่ในตำแหน่งเดิมดังรูปที่ 2.30 (สภาวะปกติ) จากคุณสมบัติ ของขดลวดโซลินอยด์นี้เอง เราจึงสามารถนำโซลินอยด์ไปเคลื่อนที่ของวาล์วในระบบนิวแมติกส์ได้

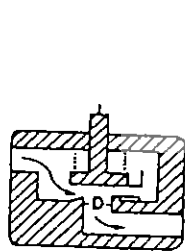
2.5.2 โครงสร้างและหลักการทำงานของโซลินอยด์วาล์ว

โครงสร้างของวาล์วควบคุมทั่วไปจะแบ่งออกเป็น 2 ชนิดใหญ่ ๆ คือวาล์วแบบนั่งบ่า (poppet valve หรือ seat valve) และวาล์วแบบลูกสูบเลื่อน (piston slide valve) วาล์วแต่ละอย่างจะมีข้อดีและข้อเสียต่างกัน

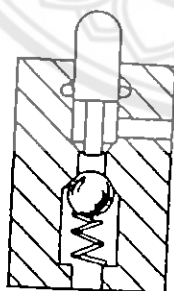
-วาล์วแบบนั่งบ่า (แบบลูกบอด) ตามรูปที่ 2.32

ข้อดี ปัญหาเรื่องซีดซำรุคมีน้อย

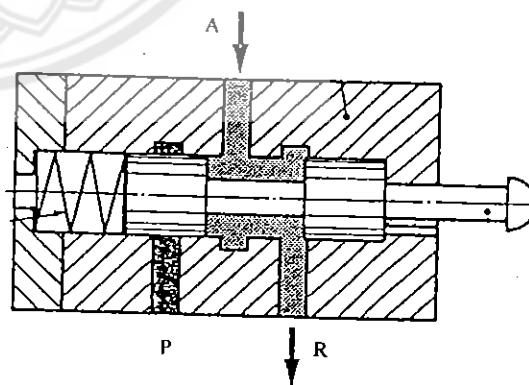
ข้อเสีย แรงที่ใช้ในการเลื่อนวาล์วมัก เนื่องจากต้องเอาชนะแรงสปริงและแรงดันลม



แบบแผ่นกลม

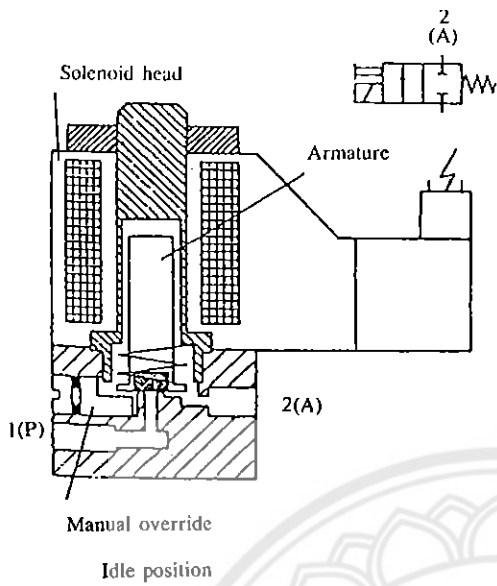


แบบลูกบอด

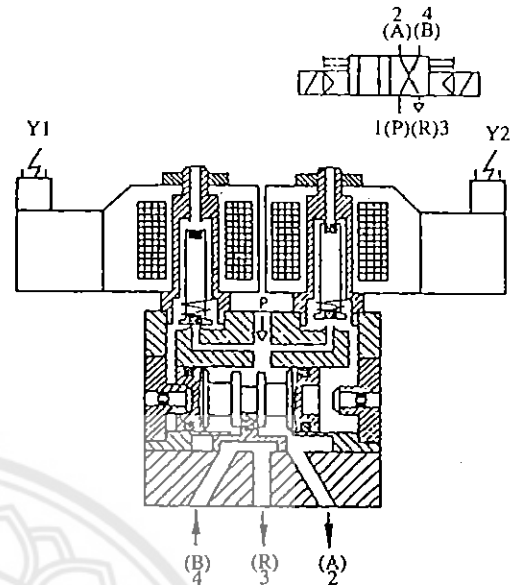


รูปที่ 2.32 แบบแผ่นกลม

รูปที่ 2.33 แบบลูกบอด



รูปที่ 2.34 โซลินอยด์แบบนั้งบ่า



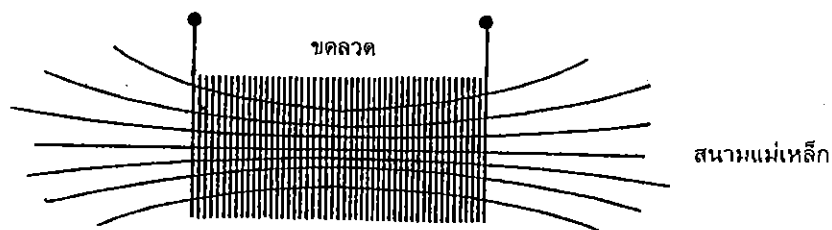
รูปที่ 2.35 โซลินอยด์วาล์วแบบลูกสูบเลื่อน

-วาล์วแบบลูกสูบเลื่อน ตามรูปที่ 2.35

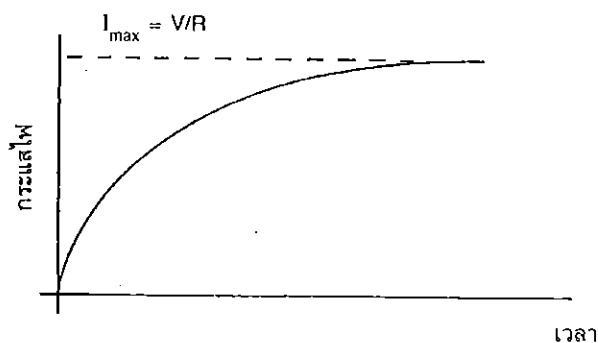
จะมีข้อดี คือ แรงที่ใช้ในการเลื่อนวาล์วน้อยกว่าวาล์วแบบนั้งบ่า เพราะแรงที่ใช้เพียงเอาชนะแรงสปริงเท่านั้น ส่วนข้อเสีย คือ ซีลของวาล์วจะเสียดสีกับ โครงตลอดเวลาที่มีการเลื่อนวาล์ว ทำให้อายุการใช้งานของซีลสั้นลง

2.5.3 ระบบควบคุมโดยไฟฟ้า

เนื่องจากระบบไฟฟ้าหรือระบบอิเล็กทรอนิกส์ โดยทั่วไปที่ใช้ในระบบไฮดรอลิกนั้น จะใช้ขดลวดไฟฟ้าหรือที่เรียกว่า โซลินอยด์ (Solenoid) คือใช้อำนาจของแม่เหล็กจากสนามแม่เหล็กที่เกิดขึ้นในเวลาที่มีไฟฟ้าไปในขดลวด เพื่อไปเลื่อนแกนเหล็กให้ดันแกนของลิ้นทลอค (Spool) อีกทีหนึ่งและสนามแม่เหล็กที่เกิดขึ้นนั้นดูได้ที่รูป 2.36

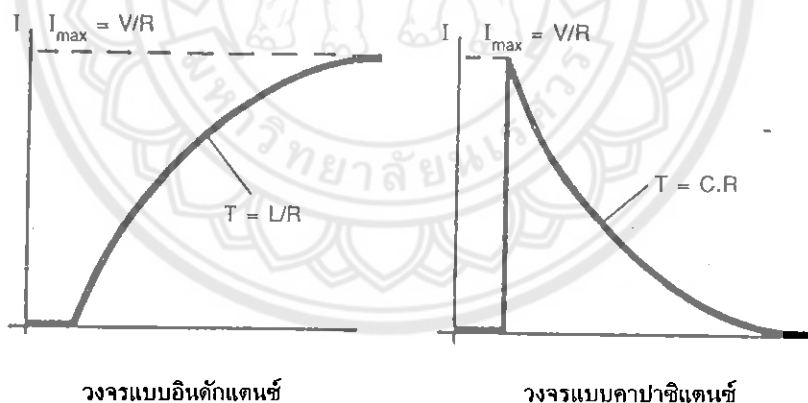


รูปที่ 2.36 ขดลวดไฟฟ้า



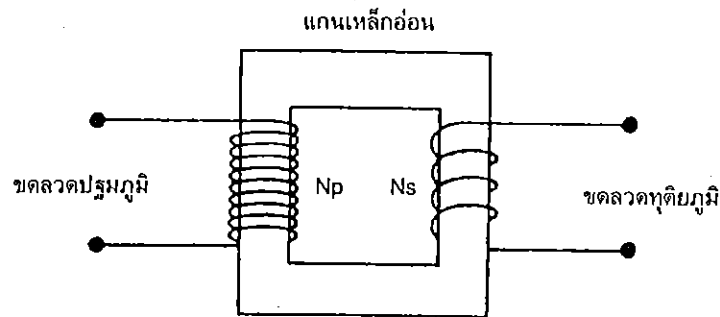
กราฟที่ 2.1 เวลาในการกินกระแสไฟฟ้า

จากกราฟที่ 2.1 ที่แสดงนี้ ถ้าป้อนกระแสไฟฟ้าเข้าไปในขดลวดในระยะเริ่มแรกนั้น การใช้กระแสของไฟฟ้าจะเริ่มคืบจากศูนย์แล้วจะเพิ่มขึ้นเรื่อย ๆ จนถึงระดับที่สูงสุดของกระแสไฟฟ้าในขดลวดซึ่งสามารถที่จะทราบเวลาว่าจะใช้เวลาเท่าไรนับจากเริ่มการใช้กระแสไฟฟ้าจากศูนย์จนถึงใช้ไปมากที่สุด



กราฟที่ 2.2 เปรียบเทียบวงจรแบบอินดักแตนซ์และแบบคาปาซิแตนซ์ (Inductance and Capacitance)

จากกราฟที่ 2.2 เป็นการแสดงการใช้กระแสไฟฟ้าของวงจรที่มีการต่อค่าความต้านทานอย่างเดียวกับการต่อค่าความต้านทานกับตัวคาปาซิแตนซ์ จะเห็นได้ว่าแบบธรรมดานั้น เวลาเริ่มป้อนกระแสไฟฟ้าเข้าไปในวงจร การใช้กระแสไฟฟ้าจะค่อย ๆ ใช้ไปถึงค่าที่สูงสุด จากนั้นก็จะใช้กระแสไฟฟ้าทำนั้นไปตลอดจนกว่าจะหยุดการป้อนกระแสไฟฟ้า แต่ตรงกันข้ามกับวงจรที่มีการต่อคาปาซิแตนซ์เข้าไป ปรากฏว่าเริ่มแรกที่ป้อนกระแสไฟฟ้าเข้าไปในวงจรการใช้กระแสไฟฟ้าจะขึ้นสูงสุดทันทีจากนั้นกระแสไฟฟ้าจะค่อย ๆ ลดลงเรื่อย ๆ จนถึงศูนย์แล้วจะใช้กระแสไฟฟ้าทำนั้นไปตลอด



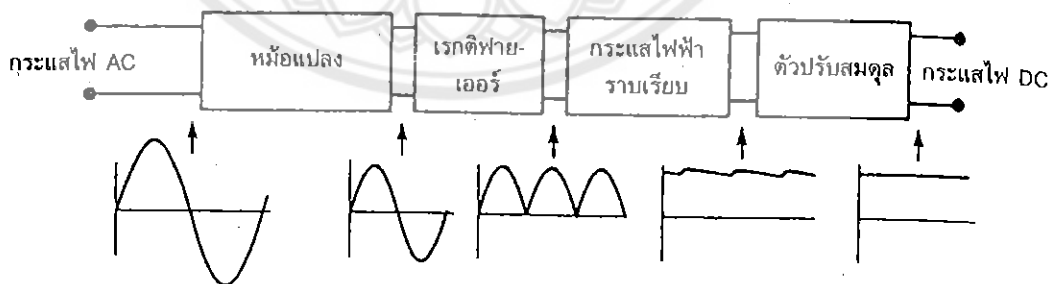
รูปที่ 2.37 หม้อแปลง

จากรูปที่ 2.37 แสดงโครงสร้างภายในของหม้อแปลง (Transformer) ภายในหม้อแปลงนี้จะมีขดลวดหลักอยู่สองขดและแกนเหล็กอ่อนอีกชุดหนึ่ง โดยมีขดลวดพันอยู่รอบแกนของเหล็กอ่อน เมื่อป้อนกระแสไฟฟ้าเข้าไปในขดลวดข้างใดข้างหนึ่ง กระแสไฟฟ้าที่ออกมาจากขดลวดอีกข้างหนึ่งจะได้กระแสไฟฟ้าที่แตกต่างจากค่าที่ป้อนเข้าไปในครั้งแรก พอสรุปได้ดังต่อไปนี้

เมื่อค่า $N_s < N_p$ จะได้ค่ากระแสไฟฟ้าออกมาน้อยกว่าค่าที่ป้อนเข้าไป

เมื่อค่า $N_s > N_p$ จะได้ค่ากระแสไฟฟ้าออกมามากกว่าค่าที่ป้อนเข้าไป

เมื่อต้องการที่จะแปลงไฟฟ้าจากมากไปหาน้อย จะต้องใช้กระแสไฟฟ้าป้อนเข้าที่ด้านขดลวดค่ามากแล้วออกทางขดลวดค่าน้อย แต่เมื่อต้องการเพิ่มกระแสไฟฟ้าก็ให้ป้อนกระแสไฟฟ้าเข้าที่ขดลวดค่าน้อยแล้วออกในค่านขดลวดค่ามาก



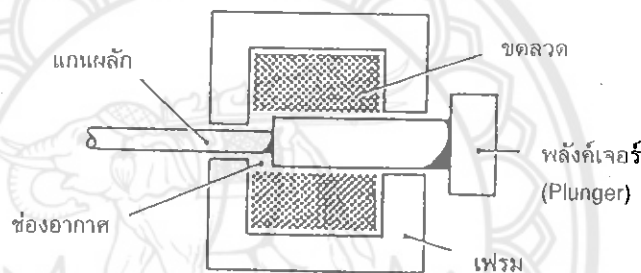
รูปที่ 2.38 ขั้นตอนการแปลงกระแสไฟฟ้าสลับเป็นกระแสไฟฟ้าตรง

ขั้นตอนการแปลงกระแสไฟฟ้าสลับเป็นกระแสตรงเริ่มจากกระแสไฟฟ้าสลับเข้าในหม้อแปลง เมื่อผ่านเรกติฟายเออร์ (Rectifier) กระแสไฟฟ้าจะเปลี่ยนเป็นกระแสตรง จากนั้นจะผ่านเข้าไปในตัวคาปาซิเตอร์ (Capacitor) เพื่อให้กระแสไฟฟาราบเรียบมากขึ้น (Smoothing Capacitor) แล้ว

ผ่านเข้าไปในตัวปรับสมดุล (Stabilizer) เพื่อให้ได้กระแสไฟฟ้าตรงที่ราบเรียบมากที่สุด

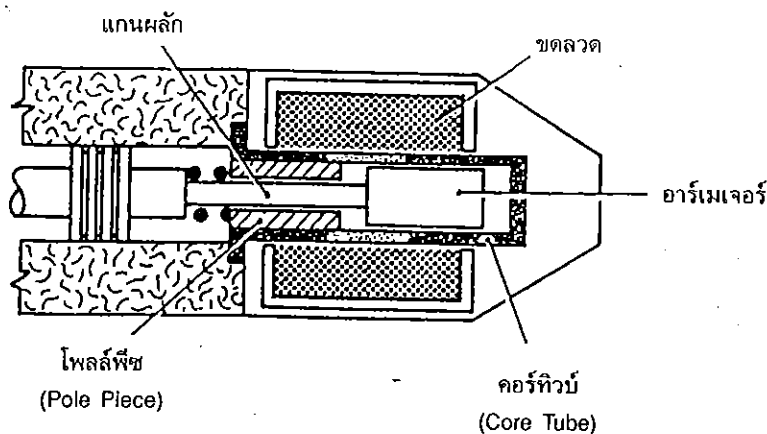
โซลินอยด์อีกชนิดหนึ่งเรียกว่า โซลินอยด์แบบแห้ง (Air - gap Solenoid) จะมีขดลวดและแกนเหล็กและมีช่องว่างของอากาศ ดังนั้นเมื่อป้อนกระแสเข้าไป สนามแม่เหล็กที่เกิดขึ้นจะมีอำนาจที่แรงมาก และอำนาจแม่เหล็กนี้ก็ดูดให้แกนของโซลินอยด์เลื่อนเข้าด้วยความเร็วมากและแรง เมื่อหยุดการจ่ายกระแสไฟฟ้า แรงของสปริงภายในจะดันให้แกนของโซลินอยด์เลื่อนกลับเองโดยอัตโนมัติ

เมื่อป้อนกระแสไฟฟ้าเข้าไปในขดลวดและทำให้เกิดอำนาจแม่เหล็กดูดให้แกนของโซลินอยด์เลื่อนแล้วทำให้สนามแม่เหล็กครบวงจร เมื่อสนามแม่เหล็กครบวงจรแล้วกระแสไฟฟ้าในวงจรจะใช้น้อยลงกว่าเดิม แต่ถ้าไม่ครบวงจร กระแสไฟฟ้าจะต้องใช้มาก ซึ่งทำให้เกิดความร้อนที่ขดลวด ซึ่งทำให้ขดลวดไหม้ได้



รูปที่ 2.39 โซลินอยด์

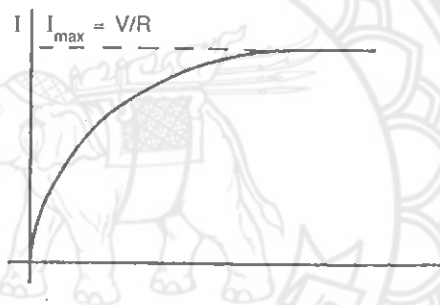
จากรูปที่ 2.40 เป็นโซลินอยด์อีกประเภทหนึ่งเรียกว่า โซลินอยด์แบบเปียก (Wet-armature Solenoid) ภายในช่องว่างระหว่างขดลวดกับแกนเหล็กอ่อนนั้นจะมีน้ำมันอยู่แทนที่อากาศ ซึ่งโซลินอยด์แบบเปียกนี้ใช้น้ำมันที่ขังตัวอยู่นี้ทำหน้าที่ระบายความร้อนในขดลวด ส่วนหลักการทำงานนั้นจะเหมือนกับโซลินอยด์แบบแห้งทุกประการ



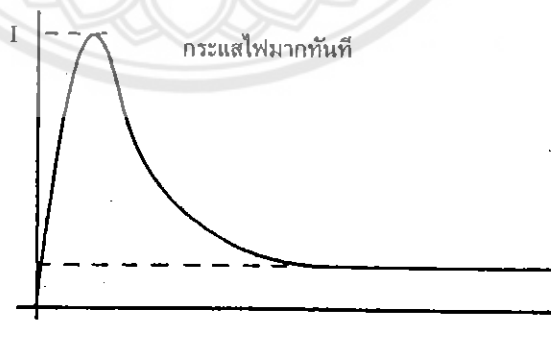
รูปที่ 2.40 โซลินอยด์แบบเปียก

การใช้กระแสไฟฟ้าของคอยล์แบบกระแสตรงมีหลักการทำงานดังต่อไปนี้ เมื่อจ่ายกระแสไฟฟ้าเข้าไปในขดลวดแล้ว สนามแม่เหล็กจะค่อย ๆ เกิดขึ้นจนสามารถดูดหรือดันให้แกนเหล็กเคลื่อนไปได้ ดังนั้นกระแสไฟฟ้าในระบบจึงค่อย ๆ เพิ่มขึ้นสูงสุด จากนั้นก็จะคงที่ตลอดไป เมื่อเป็นเช่นนี้ การเคลื่อนของแกนเหล็กก็จะราบเรียบไม่กระชาก หรือเมื่อแกนเหล็กเคลื่อนออกจนสุดระยะแล้ว การใช้กระแสไฟฟ้างี้ก็จะคงที่ตลอดไป แต่ถ้าการเคลื่อนของแกนเหล็กออกไปไม่สุดระยะ กระแสไฟฟ้าก็ขึ้นไปไม่สุด ทำให้คอยล์ไม่ร้อน

การใช้โซลินอยด์แบบกระแสตรงนี้จะใช้กระแสไฟฟ้ามากตลอดเวลาที่มีการทำงาน แต่มีข้อดีที่ว่า การเคลื่อนของแกนเหล็กที่จะไปดันแกนของวาล์วนั้นเคลื่อนออกไปอย่างนุ่มนวล จึงทำให้การจ่ายน้ำมันออกไปจากวาล์วได้อย่างไม่รุนแรง ทำให้กระบอกสูบเคลื่อนที่ได้อย่างนุ่มนวล



กราฟที่ 2.3 การใช้กระแสไฟฟ้าของคอยล์แบบกระแสตรง



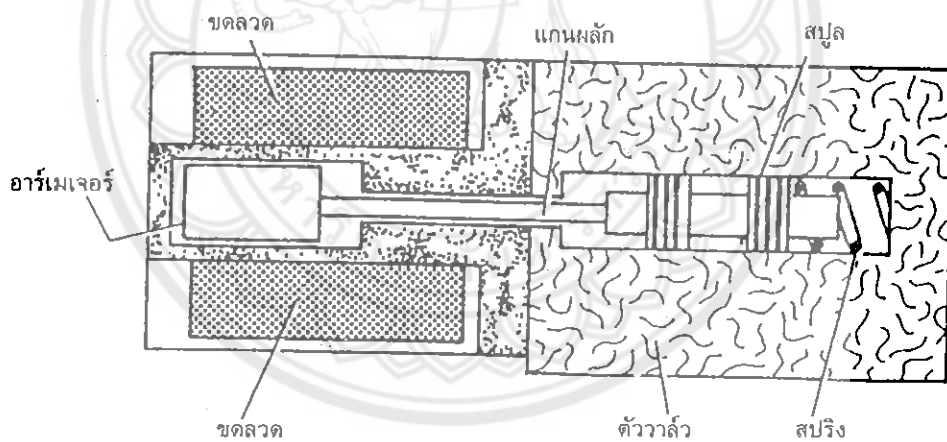
กราฟที่ 2.4 การใช้กระแสไฟฟ้าของคอยล์แบบกระแสสลับ

จากกราฟที่ 2.4 นี้ เมื่อจ่ายไฟฟ้า 220 โวลต์ เข้าไปในขดลวด สนามแม่เหล็กจะเกิดขึ้นทันที เพราะว่กระแสไฟฟ้าที่ขดลวดใช้ในช่วงแรกจะกระชากขึ้นสูงมาก จากนั้นก็จะค่อย ๆ ลดลงจนต่ำ

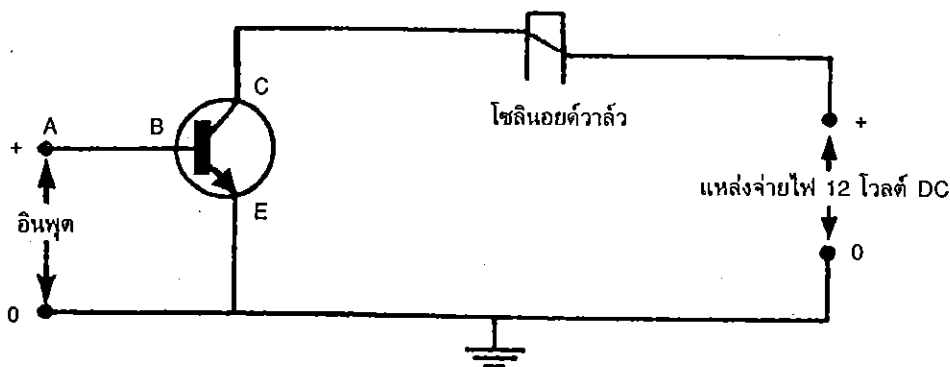
ลงมาที่ค่าหนึ่ง แล้วจะคงที่ตลอดไปจนกระทั่งเกิดการป้อนกระแสไฟฟ้า ด้วยเหตุนี้จึงเป็นผลทำให้ การเลื่อนของแกนเหล็กออกไปอย่างรุนแรงมาก เป็นผลให้กระบอกสูบเลื่อนออกไปอย่างรุนแรง มากกว่าการใช้โซลินอยด์แบบกระแสไฟตรง

สรุปได้ว่าการใช้โซลินอยด์แบบกระแสสลับนั้นจะใช้กระแสไฟฟ้าน้อยกว่าแบบไฟกระแสตรง แต่การเลื่อนของแกนเหล็กหรือแกนของวาล์วจะรุนแรงมากกว่าแบบกระแสตรง และเมื่อมีการ เลื่อนออกไปของแกนเหล็กแล้วเกิดการติดค้างขึ้น กระแสก็จะค้างอยู่ที่สูงสุดอย่างนั้นตลอดไป จึง เป็นสาเหตุทำให้ขดลวดร้อนและไหม้ได้ในที่สุด แต่จะประหยัดค่ากระแสไฟฟ้ามากกว่าแบบ กระแสตรง

จากรูปที่ 2.41 เป็นการแสดงโครงสร้างภายในของโซลินอยด์วาล์วที่มีแกนเลื่อนหลักและ สปูล (Push Pin and Spool) โดยลักษณะทั่ว ๆ ไปแล้วก็เหมือนกับโซลินอยด์วาล์วที่มีอยู่โดยทั่ว ๆ ไป แต่ที่เรียกว่าโซลินอยด์แบบพรอพอร์ชันนัล (Proportional Solenoid) นั้นจะแตกต่างกับโซลินอยด์ แบบอื่น ๆ คือ จะใช้กระแสไฟตรงและสามารถควบคุมการป้อนกระแสไฟฟ้าที่เข้าไปในขดลวดได้ ตามที่ต้องการ

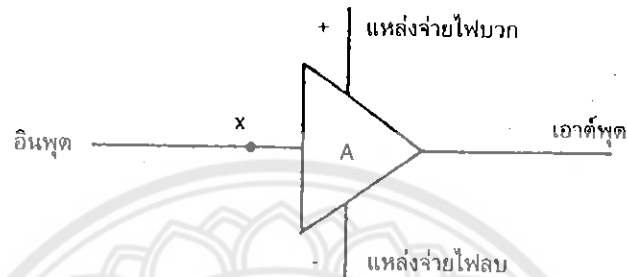


รูปที่ 2.41 โครงสร้างของโซลินอยด์

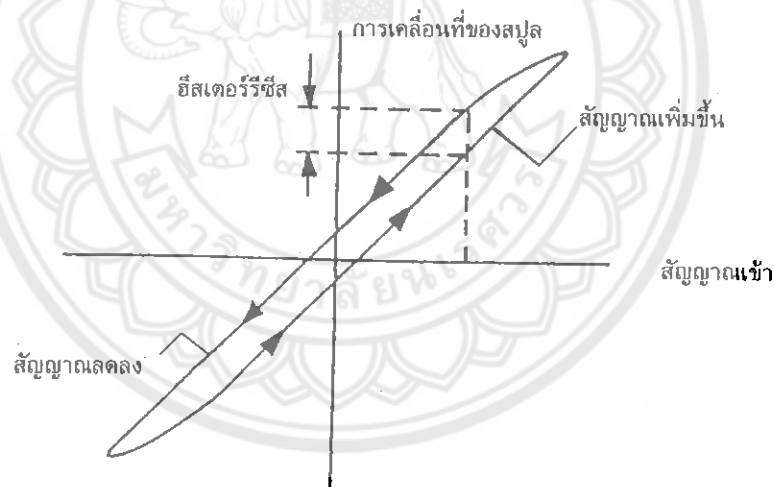


รูปที่ 2.42 วงจร โซลินอยด์แบบพรอพอร์ชันนัล

จากรูปที่ 2.42 เป็นการแสดงวงจรที่ใช้อุปกรณ์ทางอิเล็กทรอนิกส์มาเกี่ยวข้องกับ โซลีนอยด์ สามารถทำงานได้ตามปกติ แต่การป้อนไฟฟ้าทางด้านอินพุตสามารถที่จะป้อนกระแสไฟฟ้าค่าที่ต่ำกว่าด้านจ่ายไฟขนาด 12 V.DC. ส่วนในรูปที่ 2.41 ก็เช่นเดียวกัน เป็นการแสดงรูปสัญลักษณ์ที่ใช้แทนทรานซิสเตอร์ในวงจรของรูปที่ 2.43



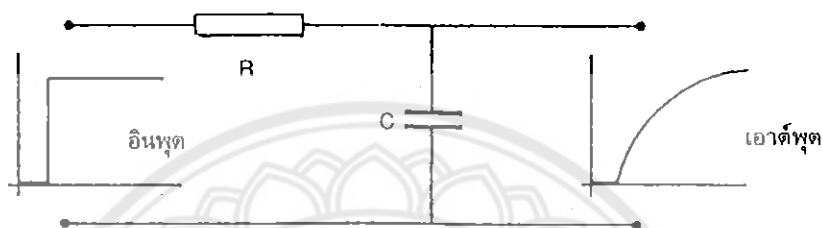
รูปที่ 2.43 สัญลักษณ์ของทรานซิสเตอร์



กราฟที่ 2.5 ฮิสเตอร์รีซิส

จากกราฟที่ 2.5 เป็นกราฟแสดงค่าที่ส่งเข้าไปเพื่อสั่งให้แกนวาล์วเคลื่อนได้ตามความต้องการ เมื่อสั่งงานให้สปูลวาล์วเคลื่อนที่ ค่าที่ได้จากการเดินทางไปกลับนั้นไม่เท่ากัน คือ เมื่อสั่งให้แกนวาล์วเคลื่อนออกโดยการตั้งด้านอินพุตให้เคลื่อนออกไปนั้น จะมีค่าอินพุตที่ค่อย ๆ เพิ่มขึ้นจนถึงค่าที่ได้ตั้งเอาไว้ ซึ่งเมื่อค่าอินพุตเพิ่มขึ้นนั้นการเคลื่อนของแกนวาล์วจะค่อย ๆ เคลื่อนไปด้วยจนถึงจุด ๆ หนึ่งแล้วจากนั้นแกนวาล์วก็จะค่อย ๆ เคลื่อนกลับโดยที่ค่าของกระแสไฟในด้านอินพุตนั้นก็ค่อย ๆ ลดลงด้วย แต่ในการเคลื่อนตัวกลับของแกนวาล์ว ระยะเวลาที่เคลื่อนไปกับค่าของกระแสไฟฟ้าที่

ป้อนเข้าไปให้มันไม่เท่ากับที่ใช้ในครั้งแรก จึงเกิดค่าผิดพลาดขึ้นที่ระยะการเลื่อนของแกนวาล์ว หรืออาจจะเกิดขึ้นจากการที่เกิดค่าความเสียดทานระหว่างแกนวาล์วกับตัววาล์ว จึงทำให้เกิดการผิดพลาดขึ้นในการสั่งงานให้เลื่อนออกไปแล้วให้เลื่อนกลับในทิศทางเดียวกัน อาการนี้เรียกว่าฮิสเตอร์รีซิส (Hysteresis) ซึ่งพิจารณาได้จากกราฟที่ 2.5 ในการแก้ข้อผิดพลาดนี้นั้นสามารถที่จะแก้ไขได้จากการปรับค่าความต้านทานในแผ่นของการ์ดแอมพลิฟายเออร์ (Amplifier Card)

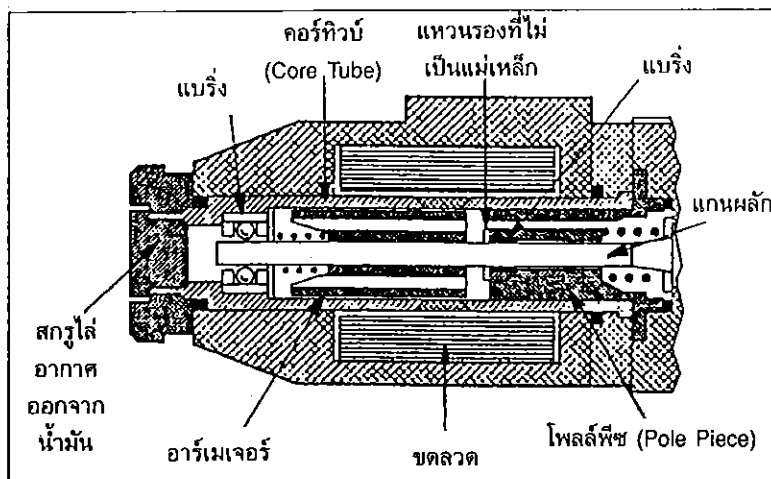


รูปที่ 2.44 ผลที่เกิดขึ้นทางเอาต์พุต

จากรูปที่ 2.44 เป็นการแสดงผลที่เกิดขึ้นในด้านเอาต์พุตว่ามีรูปแบบออกมาเป็นแบบใด เมื่อเราป้อนค่าทางด้านอินพุตเข้าไปแบบหนึ่ง แล้วค่าที่ออกมาเป็นอีกแบบหนึ่ง จากรูปที่เห็นนี่เป็นการแสดงกราฟระหว่างแรงดันและเวลา แสดงให้เห็นว่าเมื่อป้อนเข้าไปแบบหนึ่งแล้วจะได้ผลออกมาอีกแบบหนึ่ง ความชันของเส้นกราฟขึ้นอยู่กับค่าความต้านทานที่มีในด้านอินพุต ในกรณีนี้สามารถที่จะเปลี่ยนค่าความต้านทานได้

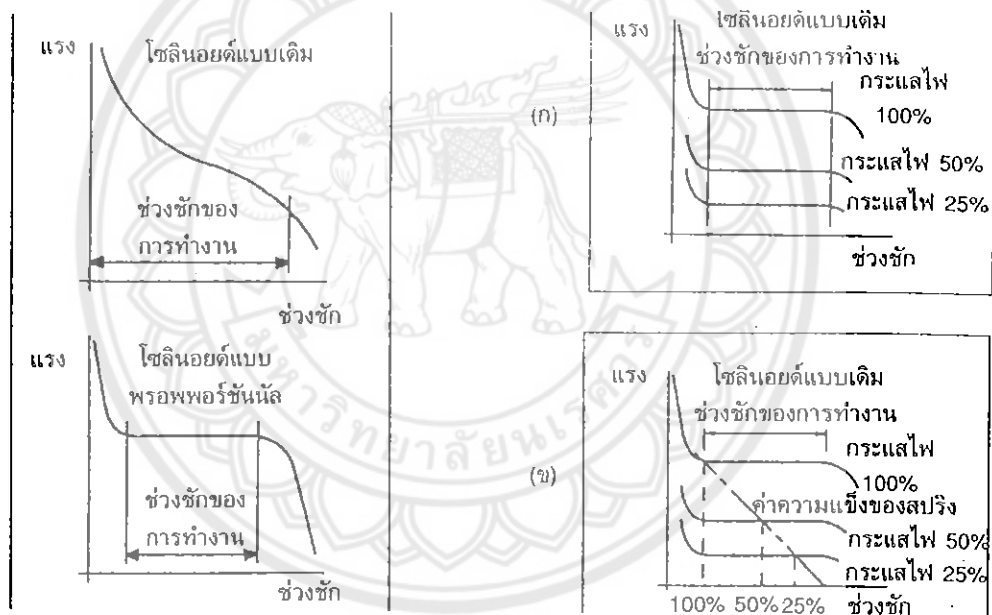
2.5.4 ระบบการทำงานของวาล์วชนิดต่าง ๆ

โซลินอยด์วาล์วและ โครงสร้างภายใน



รูปที่ 2.45 โครงสร้างภายในของโซลินอยด์วาล์ว

จากรูปที่ 2.45 เป็นโครงสร้างของโซลินอยด์วาล์ว ซึ่งไม่ว่าจะเป็นแบบธรรมดาหรือแบบพรอพเพอร์ชันนัลวาล์ว แรงกับระยะเลื่อนของแกนวาล์วของแต่ละแบบจะแตกต่างกัน จากรูปที่แสดงนี้จะเห็นความแตกต่างระหว่างแบบธรรมดากับแบบพรอพเพอร์ชันนัล ว่าเมื่อเริ่มทำงานแล้วแรงที่เกิดขึ้นกับสนามแม่เหล็กเพื่อนำมาใช้ค้ำแกนเหล็กนั้นมีค่าคงที่หรือไม่เมื่อเลื่อนออกไปแล้ว โดยดูได้จากรูปที่ 2.45 และในทำนองเดียวกันสำหรับโซลินอยด์แบบพิเศษนี้สามารถที่จะปรับเปลี่ยนค่ากระแสไฟฟ้าที่ป้อนให้กับขดลวดไฟฟ้าโดยดูได้จากรูปที่ 2.46 จะเห็นว่าเมื่อเราต้องการให้กระแสไฟฟ้าที่เข้าไปมีค่าประมาณ 100%, 50% หรือ 25% ก็สามารถปรับค่าได้โดยระยะของช่วงชักการทำงานยังคงเท่าเดิมอยู่ แต่ในรูปที่ 2.46 (ข) นั้น บอกถึงค่าความแข็งของสปริงที่ใช้ หรือค่าความแข็งเมื่อเราป้อนค่ากระแสไฟฟ้าเข้าไปในขนาดที่ไม่เท่ากัน



รูปที่ 2.46 การเกิดแรงของสนามแม่เหล็ก

รูปที่ 2.47 การปรับเปลี่ยนค่ากระแสไฟฟ้าที่เข้าไปในขดลวด

2.6 เทคนิคการให้น้ำแก่พืช

น้ำเป็นปัจจัยพื้นฐานในการเจริญเติบโตของพืช ฉะนั้นพืชจึงต้องรับน้ำอย่างเพียงพอ

2.6.1 ปัจจัยที่มีผลต่อการให้น้ำ

1. สภาพการระเหยของน้ำในอากาศ
2. ความชื้นของดิน ปัจจัยที่กำหนดปริมาณน้ำที่ดินสามารถอุ้มน้ำไว้ได้คือ ลักษณะการ

กระจายตัว ของช่องว่างในดิน และปริมาณอินทรีย์วัตถุ โดยที่สภาพความชื้นในดินมี 3 ระดับ

- ระดับจุดอิ่มน้ำเต็มที่ คือสภาพที่ช่องว่างในดินเต็มไปด้วยน้ำทั้งหมด ระบบรากพืชผักจะขาดออกซิเจน และตายในที่สุด

- ระดับความจุความชื้นสนาม คือจากสภาพที่หลังจากให้น้ำบนดินแล้ว น้ำจะไหลออกจากช่องว่างขนาดใหญ่ด้วยแรงดึงดูดของโลก และอากาศจะเข้าแทนที่จนถึงจุดหนึ่ง ซึ่งแรงที่พยายามดึงนั้นจะเท่ากับแรงที่น้ำเกาะอยู่กับเม็ดดิน โดยทั่วไปดินส่วนใหญ่ที่ใช้ปลูกผัก จะใช้ระยะเวลาประมาณ 2 วัน

- ระดับจุดเหี่ยวถาวร ได้แก่ ระดับความชื้นในดินที่ต่ำลงมากจนทำให้พืชผักเหี่ยวและไม่สามารถกลับคืนสภาพได้อีกในสภาพอากาศชื้น

3. ชนิดของพืชผัก ปัจจัยต่างๆ เช่น ความลึกและการแพร่กระจายของรากมีผลต่อความต้องการน้ำของพืช

2.6.2 ยุทธศาสตร์การให้น้ำ

ให้น้ำมากตลอดเวลา ให้น้ำน้อยมากจนแน่ใจว่าไม่มีทางที่ผักจะขาดน้ำได้ จะใช้วิธีนี้เมื่อมีแหล่งจ่ายน้ำมาก ทำให้ดินพืชมีการเจริญเติบโตสูงสุด แต่วิธีนี้มีข้อเสียหลายประการ คือ

- เกิดการสูญเสียธาตุอาหารพืชไปให้กับน้ำมาก คือเกิดการชะล้างของน้ำ
- ดินแฉะมาก ก่อให้เกิดโรครากเน่าต่างๆ ได้ง่าย
- สูญเสียน้ำมาก
- ส่งผลความเปลี่ยนแปลงของสภาพอากาศ
- มีความรู้เกี่ยวกับผักชนิดนั้นๆ ว่าสามารถทนทานต่อการขาดน้ำได้มากเพียงใด
- เก็บตัวอย่างไปตรวจสอบความชื้น
- ทดลองให้น้ำจนพบสูตรที่พอเหมาะ

2.6.3 ใช้เครื่องมือวัดความชื้น ก่อนที่จะเริ่มให้น้ำ

- วิธีการให้น้ำ

1. การให้น้ำแบบปล่อยไหลบนหน้าดิน
2. การให้น้ำใต้ระดับผิวดิน
3. การให้น้ำเหนือระดับดิน เป็นวิธีง่ายที่สุดตามธรรมชาติ
4. ฉีดพ่นฝอยออกตามสาย(ท่อ) ส่วนใหญ่ใช้กับพืชผักที่ข้าง่าย และราคาแพง เช่นผักสลัด ขนาดที่ฉีดพ่นขึ้นกับ ขนาดหัวพ่นและระยะห่างแต่ละรู

ตารางที่ 2.3 ลักษณะและความรู้สึกจากการล้มผลเนื้อดินที่ระดับความชื้นต่างๆ
(ที่มา: หลักการปลูกผักสวนผัก 1 เมืองทอง ทวนทวี)

% ความชื้นที่เป็นประโยชน์ได้ในดิน	ดินเนื้อหยาบมาก (ดินทราย)	ดินเนื้อหยาบ (ดินร่วนทราย)	ดินเนื้อหยาบปานกลาง (ดินร่วน)	ดินเนื้อละเอียด (ดินเหนียว, ดินเหนียวร่วน)
น้อยกว่า 50%	แห้ง หลวม และเป็นเม็ดเล็กๆ ผ่านนิ้วมือ แห้งและไม่เป็นลูกกลมเมื่อบีบปั้น	แห้ง หลวม ไหลผ่านนิ้วมือ แข็ง ไม่เป็นลูกกลม	ผง บางครั้งเป็นก้อนเล็กๆ ที่แตกเป็นผงได้ง่าย เป็นก้อนบ้างและจะปั้นเป็นลูกกลมได้	แข็ง เปราะ มีรอยร้าวและเศษผงที่ผิว อ่อนหยุ่น และปั้นเป็นลูกได้
50-75%	ให้น้ำ แห้ง ไม่เป็นลูกกลมเมื่อบีบปั้น	ให้น้ำ จะปั้นเป็นลูกบอลซึ่งไม่คงรูปร่างได้	ให้น้ำ เป็นก้อนคล้ายพลาสติก	ให้น้ำ ปั้นเป็นก้อนกลมได้
75-100%	ชื้นเกาะตัวกันเล็กน้อยเป็นก้อนกลม หลวมมากสีคล้ำขึ้น	เป็นก้อนกลม หลวมๆซึ่งแตกได้ง่ายสีคล้ำ	เป็นก้อนกลมชื้น หยุ่นมาก เป็นมันสีคล้ำ	รู้สึกเหนียวมาก
100%	เมื่อบีบปั้นจะมีน้ำ	ส่วนเกินไหลออก	จะมีเศษดินเกาะมือ	สีคล้ำ

สปริงเกอร์ที่หมุนเป็นวงนิยมนำมาใช้กับแปลงผัก การคำนวณขนาดใหญ่ ใช้สปริงเกอร์สำหรับความดันปานกลาง สามารถฉีดพ่นได้กว้างถึง 33 เมตร และอัตราการน้ำที่ให้ประมาณ 6 มม./ชม.

การใช้เรือฉีดพ่นเครื่องน้ำ

การลากท่อสายยางฉีดบัวรดน้ำ

การใช้แครงสาธและกระบวยน้ำรด

ข้อเสนอแนะในการให้น้ำทั่วไป

ให้น้ำเพียงพอแก่ความต้องการของพืช ในระหว่างการเจริญเติบโตแต่ละช่วง

การให้น้ำมากเกินไปในแต่ละครั้งทำให้ผลผลิตลดลง เนื่องจากการทำให้การเจริญเติบโตทางใบ กิ่ง ก้าน มากขึ้นโดยที่ไม่สร้างส่วนที่กินได้เพิ่มขึ้น

การให้น้ำบ่อยๆ ที่ละน้อย แต่ควรให้มากเต็มทีจนน้ำซึมทั่วตลอดหน้าตัดของดิน ไม่ให้น้ำในตอนเย็นเพราะพืชจะเปียกทั้งคืน ทำให้เกิดโรคราบาดได้ง่าย ควรให้น้ำตอนเช้า

2.7 มาตรฐาน RS-232-C

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันกว้างขวางมากที่สุดคือ RS-232-C ซึ่งถูกประกาศในปี ค.ศ. 1969 โดย Electronic Industries Association โดยมาตรฐาน RS-232-C ได้ระบุคุณลักษณะทางไฟฟ้าของวงจรระหว่างอุปกรณ์ 2 ตัว และการกำหนดชื่อและหมายเลขแก่สายที่จำเป็นสำหรับการเชื่อมต่อวงจร

ชื่อตามมาตรฐาน RS-232-C จำได้ยาก ในทางปฏิบัติจึงใช้ชื่อย่อแทน ตัวอย่างเช่น สายเส้นที่ 2 มีชื่อว่า BA แต่ที่ใช้กันทั่วไปคือ TXD (Transmitted Data) ตามมาตรฐาน RS-232-C สายเส้นที่ 2 นำข้อมูลจากเทอร์มินัลเข้าสู่โมเด็ม เพื่อให้การทำงานถูกต้อง เทอร์มินัลต้องส่งเอาท์พุทออกที่สายสัญญาณเส้นที่ 2 และโมเด็มสามารถเป็นสายรับข้อมูลสำหรับอุปกรณ์บางอย่าง การเชื่อมต่อโดยตรงจากสายเส้นที่ 2 บนอุปกรณ์หนึ่งเข้ากับสายเส้นที่สองของอุปกรณ์อีกตัวหนึ่ง สามารถทำได้ก็ต่อเมื่ออุปกรณ์หนึ่งส่งข้อมูลบนสายเส้นที่ 2 และอีกตัวหนึ่งรับข้อมูลบนสายเส้นที่ 2

เพื่อป้องกันไม่ให้อุปกรณ์ส่งข้อมูลบนสายเส้นเดียวกัน อุปกรณ์จึงถูกแบ่งเป็น 2 ชนิด อุปกรณ์บางอย่างเช่น เทอร์มินัล ซึ่งใช้สายเส้นที่ 2 สำหรับเอาท์พุท เรียกว่า DTB (Data Terminal Equipment) อุปกรณ์อย่างเช่น โมเด็มซึ่งใช้สายเส้นที่ 2 สำหรับอินพุท เรียกว่า DCE (Data Communication Equipment)

2.7.1 สัญญาณทางไฟฟ้า

มาตรฐาน RS-232-C กำหนดคุณลักษณะของสัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่ออนุกรมโดยตรงมีเพียง 2 ลักษณะคือ SPACE แสดงถึง ไบนารี 0 หรือแรงดันไฟฟ้าบวก และ MARK แสดงถึง ไบนารี 1 หรือแรงดันไฟฟ้าลบ

บนสายข้อมูล (เช่นสาย 2 และ 3) แรงดันไฟฟ้าบวกแสดงถึงค่าลอจิก 0 และแรงดันไฟฟ้าลบแสดงถึงค่าลอจิก 1 บนสาย Handshaking (เช่น DTR และ DSR) แรงดันไฟฟ้าบวกแสดงว่าส่งข้อมูลได้ ส่วนแรงดันไฟฟ้าลบหมายถึงหยุดส่งข้อมูล

แรงดันไฟฟ้าบวกสถานะ Space อยู่ระหว่าง +5 ถึง +15 โวลต์ สำหรับเอาท์พุท และระหว่าง +3 ถึง +15 โวลต์สำหรับอินพุท ความแตกต่างมีไว้เพื่อกรณีแรงดันไฟฟ้าสูญหายเนื่องจากความยาวของสายสัญญาณ ในทำนองเดียวกัน แรงดันไฟฟ้าลบ สถานะ MARK กำหนดระหว่าง -5 ถึง -15 โวลต์สำหรับเอาท์พุท และ -3 ถึง -15 โวลต์สำหรับอินพุท

สังเกตว่า ถ้าสายสัญญาณยาวเกินไป ระดับแรงดันไฟฟ้าจะตกลงเกินขอบเขตที่ยอมรับได้นอกจากนี้ความจุไฟฟ้าที่เกิดขึ้นจะมีผลกับคุณภาพของสัญญาณ โดยทำให้การเปลี่ยนแปลงจากแรงดันไฟฟ้าบวกเป็นไฟลบไม่ชัดเจน RS-232-C ไม่ได้มุ่งหวังให้นำไปใช้สำหรับระยะทางไกล ๆ และ

โดยทั่วไประยะทางที่ไกลที่สุดในการใช้สายสัญญาณปกติที่อัตราการส่งข้อมูลปกติคือ 50 ฟุต ถ้าอุปกรณ์อยู่ห่างกันมาก อาจจำเป็นต้องใช้โมเด็ม หรืออุปกรณ์อื่น

ตัวอย่างข้อตกลงหนึ่งของพอร์ต RS-232 คือข้อกำหนดส่วนอินเตอร์เฟซทางกายภาพ ตัวอย่างเช่น ขั้วต่อต่าง ๆ ต่างก็ทำข้อตกลงร่วมกันไว้ว่าการกำหนดให้เป็นลักษณะใดถึงทำการเชื่อมต่อกันได้

ตัวอย่างการเชื่อมต่อทางกายภาพ เช่น ลักษณะของปลั๊กไฟฟ้าไปใช้งาน แต่หากเป็นเต้าเสียบหรือปลั๊กตัวเมีย ซึ่งมีลักษณะเป็นรู ก็เป็นการบ่งชี้ว่าเป็นจุดต่อร่วมให้เครื่องใช้ไฟฟ้าอื่นต่อไปใช้งาน เป็นต้น จากตัวอย่างดังกล่าวคงทำให้มองเห็นภาพลักษณะของการอินเตอร์เฟซทางกายภาพได้ชัดเจนขึ้นมาก

การอินเตอร์เฟซ RS-232 มีอยู่ 2 แบบด้วยกันคือโดยพื้นฐานแล้วหากเป็นส่วนที่ใช้ต่อเข้ากับอุปกรณ์ประมวลผลเรียกว่า DTE (Data Terminal Equipment) และส่วนที่ติดต่อกับอุปกรณ์สื่อสารเรียกว่า DCE (Data Circuitterminating Equipment) โดยทั่วไปจะใช้ขั้วต่อ DB25 ตัวผู้สำหรับต่อกับ DTE ส่วน DCE ใช้ขั้วต่อ DB25 ตัวเมีย

โดยทั่วไปแล้วในเครื่องพีซีและคอมพิวเตอร์ส่วนใหญ่จะมีการอินเตอร์เฟซแบบ DTE คือจะเป็น DB25 ตัวผู้ ส่วนอุปกรณ์ต่อพ่วงอย่างเช่น โมเด็มและเมาส์ การอินเตอร์เฟซจะเป็นแบบ DCE หรือขั้วต่อ DB25 ตัวเมียนั่นเอง พึงระลึกเสมอว่าที่พอร์ต RS-232 ซึ่งเป็นขั้วต่อสื่อสารเฉพาะกับการอินเตอร์เฟซ DCE เท่านั้น

2.7.2 การทำงานของ RS-232

ส่วนการอินเตอร์เฟซทางลอจิกของพอร์ต RS-232 นั้นถูกออกแบบสำหรับการสื่อสารที่มีความยาวไม่เกิน 50 ฟุตที่ความเร็ว 20,000 บิตต่อวินาที (สามารถใช้งานที่ความยาวสายสื่อสารมากกว่านี้ หรือที่ความเร็วสูงกว่านี้ หรือทั้งสองอย่าง แต่ตามมาตรฐานแล้วกำหนดไว้ที่ความยาว 50 ฟุต) การสื่อสารจะติดต่อโดยผ่านสายไฟ 25 เส้นซึ่งทำหน้าที่แตกต่างกันออกไป พอร์ต RS-232 สามารถรองรับการสื่อสารได้ทั้งแบบซิงโครนัสและแบบอะซิงโครนัส ถ้าหากใช้ในการสื่อสารแบบอะซิงโครนัสแล้วเส้นควบคุมบางเส้นใน 25 เส้นนี้ไม่จำเป็นต้องใช้ตามที่กล่าวไปแล้ว โดยจะเทียบให้ระดับแรงดัน +3 โวลต์หรือมากกว่าเป็นสภาวะทำ และระดับแรงดัน -3 โวลต์หรือน้อยกว่าเป็นสภาวะไม่ทำ ถ้าหากเป็นระดับแรงดันระหว่าง -3 โวลต์ ถึง +3 โวลต์ก็จะไม่เทียบเป็นระดับใดทั้งสิ้น ส่วนสำคัญของพอร์ต RS-232 คือ โฟลคอนโทรล (Flow Control) หรือส่วนควบคุมการทำงาน

สายที่สำคัญในการสื่อสารแบบอะซิงโครนัสของพอร์ต RS-232 มีอยู่ 10 เส้น ดังแสดงในตารางที่ 2.4 สายต่าง ๆ แต่ละเส้นมีความสำคัญซึ่งควรทำความเข้าใจ เช่น สายเส้นที่ 2 เป็นสายส่งข้อมูล ถ้าหากเราต่อสายเส้นทั้งสองนี้เข้ากับสายเส้นที่ 2 ของปลายทาง ด้านปลายทางจะไม่สามารถรับข้อมูลได้ เป็นต้น แต่ละเส้นทำหน้าที่ในการควบคุมอุปกรณ์ปลายทาง (ยกเว้นสายสัญญาณกราวด์)

ตารางที่ 2.4 การทำงานของ RS-232 ทั้งสองแบบ

(ที่มา: โครงการงานฟาร์มเลี้ยงไก่ไขควบคุมด้วยระบบคอมพิวเตอร์ ทวีเดช รุ่งเรืองอนันต์)

คำอธิบาย	หมายเลข ขาแบบ25 ขา	หมายเลขขา แบบ 9 ขา	จาก	ตัวย่อ
เส้นข้อมูล				
ข้อมูลส่ง (Transmit Data)	2	3	DTE	TD
ข้อมูลรับ (Receive Data)	3	2	DCE	RD
เส้นแสดงว่ามีพลังงาน				
ข้อมูลพร้อมส่ง (Data Set Ready)	6	6	DCE	DSR
ข้อมูลเทอร์มินัลพร้อม (Data Terminal Ready)	20	4	DTE	DTR
สัญญาณที่รับจากภายนอก				
พบพาหะข้อมูล (Data Carrier Detect)	8	1	DCE	CD
สัญญาณกริ่ง (Ring Indicator)	22	9	DCE	RI
เส้นสัมพันธ์การทำงานพร้อมส่งหรือรับ				
ร้องขอส่ง (Request To Send)	4	7	DTE	RTS
หมดให้ส่ง Clear To Send)	5	8	DCE	CTS
เส้นกราวด์				
สัญญาณกราวด์ (Signal Ground)	7	5		SG
กราวด์ป้องกัน (Protective Ground)	1			FG

ตัวอย่างการโต้ตอบระหว่าง RS-232 มีขั้นตอนดังนี้ (แบบ 25 ขา)

- อุปกรณ์ทั้งสองได้รับไฟเลี้ยงและอยู่ในสภาวะพร้อมทำงาน
- เมื่อโมเด็มทั้งสองเชื่อมต่อกันจะทำการแลกเปลี่ยนพาหะกัน แล้วโมเด็ม (DCE) จะแจ้งกลับไปยังเทอร์มินัล (DTE) โดยผ่านเส้นที่ 8 (เส้น Data Carrier Detect) เพื่อแจ้งให้เทอร์มินัลปลายทางรับทราบว่ารับพาหะของข้อมูลแล้ว
- หากเทอร์มินัล (DTE) พร้อม จะสอบถามไปยังโมเด็ม (DCE) โดยผ่านสัญญาณเส้นที่ 4 ชื่อ RTS (Request to Send) เพื่อแจ้งว่าเทอร์มินัลต้องการจะส่งข้อมูล

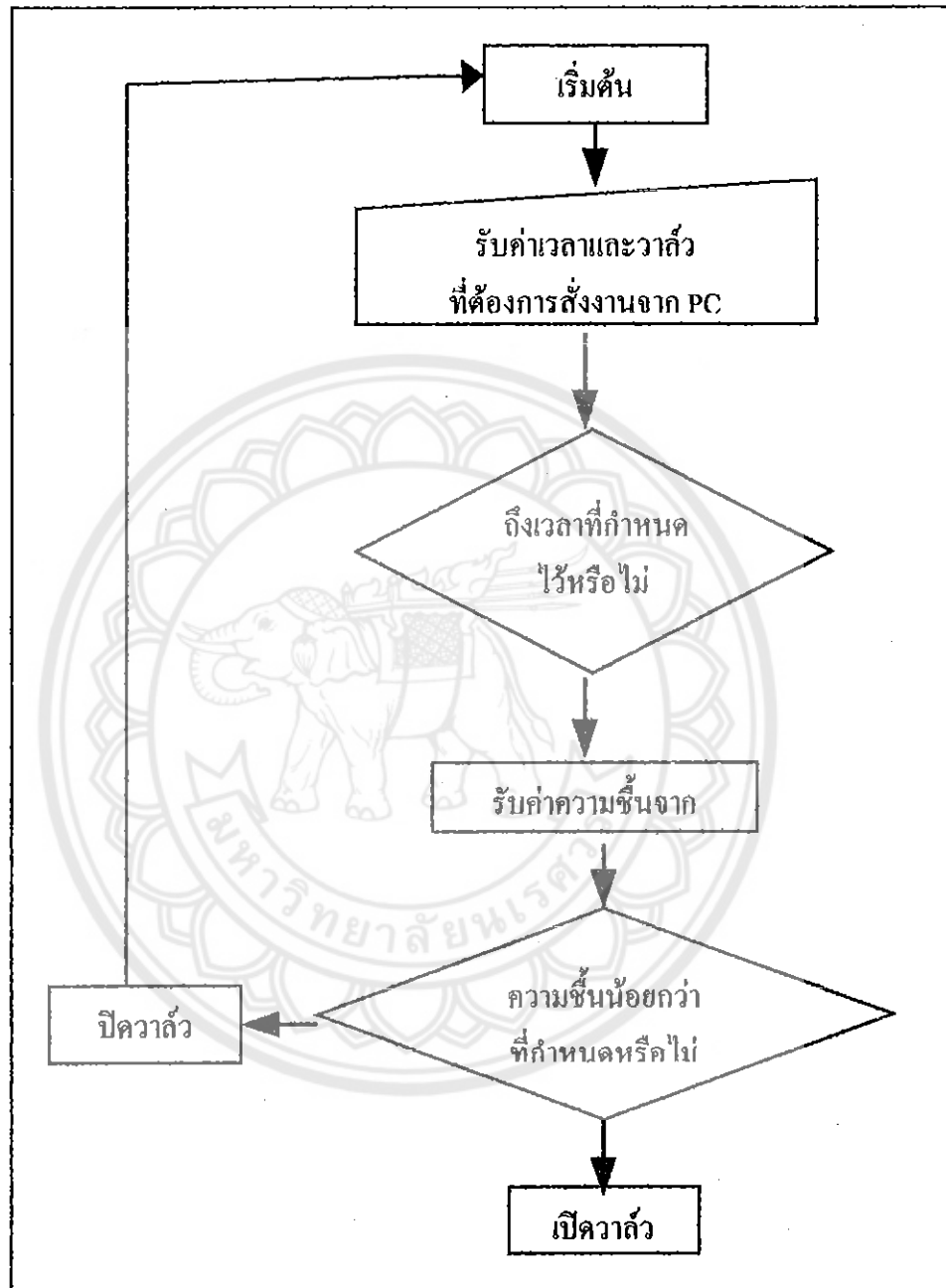
บทที่ 3

ออกแบบการทดลอง

จากทฤษฎีที่เราได้กล่าวไว้ในบทที่ 2 ซึ่งจะสามารถนำมาประยุกต์ใช้ในการออกแบบบอร์ดควบคุมโดยจะนำการทำงานของไมโครคอนโทรลเลอร์มาเป็นตัวทำงานในด้านการประมวลผลและสั่งงาน ส่วนอุปกรณ์ทางด้านอิเล็กทรอนิกส์ เช่น เซนเซอร์ โซลีนอยด์วาล์ว จะเป็นตัวทำงานทางด้านภาค อินพุท เอาท์พุทของระบบ โดยการออกแบบจะใช้การสร้างในแต่ละส่วนจนสามารถทำงานได้ตามที่เรากำหนดไว้แล้วจึงทำการเอาวงจรแต่ละส่วนมาต่อเชื่อมกัน แล้วทำการทดสอบการทำงานให้มีความสัมพันธ์กันอีกครั้ง ระบบของเราจะประกอบไปด้วยส่วนต่าง ๆ ประกอบด้วยการทำงานของบอร์ดควบคุม วงจรวัดความชื้น และ วงจรวัดอุณหภูมิ โดยจะแยกเป็นหัวข้อได้ดังนี้

3.1 การทำงานของบอร์ด ควบคุม

บอร์ดควบคุมที่ใช้ในการสั่งงานเปิด-ปิดวาล์ว จะมีแผนผังการทำงานของโปรแกรมดังรูปที่ 3.1 ซึ่งจะเริ่มทำงานโดยจะควบคุมให้ไ้ระดับความชื้นอยู่ในช่วง 50%- 70% ระบบสามารถควบคุมวาล์วได้ทั้งหมด 6 ตัว แต่ละตัวจะเป็นอิสระจากกัน มีการแสดงสถานะการทำงานที่จอ LCD ระบบจะเริ่มทำงานเมื่อรับคำสั่งมาจากคอมพิวเตอร์ โดยผ่านพอร์ต RS-232 ผู้ใช้กำหนดการทำงานผ่านโปรแกรมโดยการตั้งค่าเวลาและเลือกวาล์วที่จะเปิด เมื่อถึงเวลาที่กำหนดบอร์ดควบคุมจะรับคำสั่งเริ่มทำงานโดยการรับค่าจากเซนเซอร์ผ่าน ADC แล้วทำการประมวลผลสัญญาณที่ได้ ถ้าอยู่ในช่วงความชื้นที่น้อยกว่าที่กำหนดระบบจะทำการเปิดวาล์ว แต่ถ้าความชื้นอยู่ในช่วงหรือมากกว่าที่กำหนดระบบจะทำการปิดวาล์ว โดยการรับค่าจากเซนเซอร์ จะทำการวัดและประมวลผลทุก ๆ 10 นาที ถ้าวาล์วยังคงทำงานอยู่จนถึง 1 ชั่วโมง บอร์ดควบคุมจะสั่งให้ปิดวาล์วเองเพื่อเป็นการป้องกันในกรณีระบบเกิดปัญหา การสั่งงานจะสามารถทำได้ การควบคุมการทำงานของวาล์วจะสั่งงานผ่านชุดไมโครเวร์ โดยใช้ รีเลย์ 12 Vdc เพื่อขับโซลีนอยด์วาล์วเปิดน้ำ

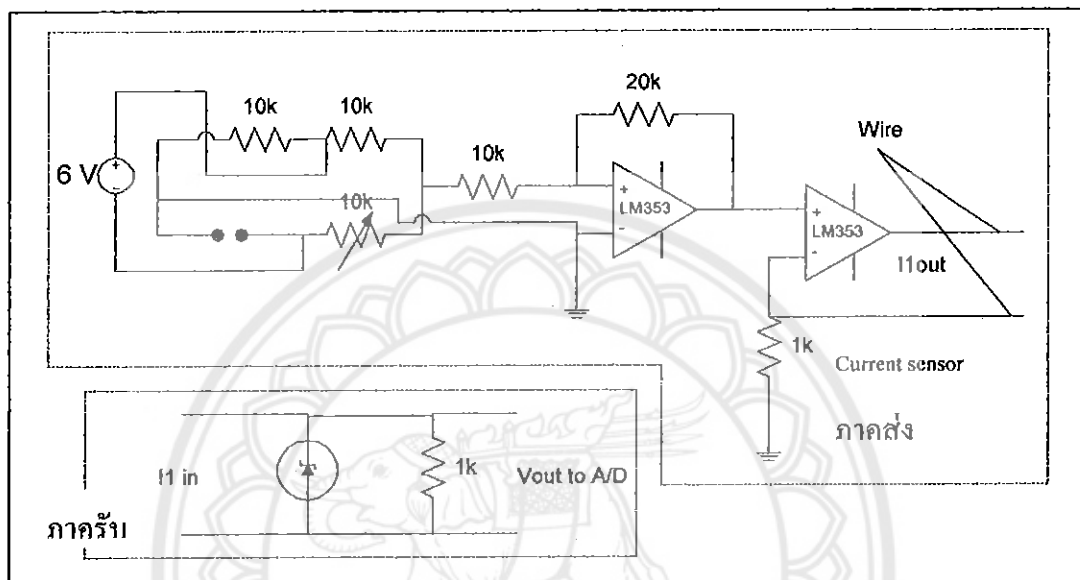


รูปที่ 3.1 แผนผังการทำงานของโปรแกรม

3.2 วงจรตรวจวัดความชื้น

3.2.1 หลักการทำงาน

วงจรความต้านทานบรिจด์ จะมีค่าแรงดันเอาท์พุท เปลี่ยนแปลงค่าตามความต้านทานในดิน เนื่องจากความชื้น โดยการเปลี่ยนแปลงเป็นดังนี้



รูปที่ 3.2 วงจรวัดความชื้น

ลักษณะการทำงาน

ในดินมีค่าความต้านทานซึ่งจะเปลี่ยนแปลงตามความชื้น เมื่อเรานำเอาวงจรบรिจด์มาต่อ โดยให้แขนของบริจด์ด้านหนึ่งเสียบลงดิน เมื่อความต้านทานของดินเปลี่ยนแปลงค่าแรงดันที่ผ่านวงจรออกมาจะมีค่าเปลี่ยนแปลง จากนั้นจะนำค่าแรงดันที่ได้มาทำการขยายสัญญาณแบบอินเวอร์ต แล้วได้ระดับแรงดันที่ต้องการมาเข้าที่ออปแอมป์ตัวที่ 2 ซึ่งจะทำหน้าที่ส่งสัญญาณเป็นแบบแหล่งจ่ายกระแสที่ภาคส่งเพื่อแก้ไขปัญหาแรงดันตกในสาย แล้วจะมีวงจรภาครับเพื่อแปลงจากกระแสให้กลับเป็นแรงดันอีกครั้งแล้วก็ส่งเข้าสู่วงจรแปลงอนาลอกเป็นดิจิทัล

การปรับตั้งค่าสัญญาณแรงดันเอาต์พุท

- 1) นำแท่งอิเล็กโทรดเสียบลงในดินที่มีความแห้งสูง โดยเสียบที่ความลึกประมาณ 6-10 เซนติเมตร วัดค่าแรงดันที่ได้แล้วหมุนความต้านทานปรับค่าได้ให้ค่าแรงดันมีค่าใกล้เคียง 0 โวลต์
- 2) นำแท่งอิเล็กโทรดเสียบลงในดินที่เปียกชื้น โดยเสียบที่ความลึกประมาณ 6-10 เซนติเมตร วัดค่าแรงดันที่ได้แล้วหมุนความต้านทานปรับค่าได้ให้ค่าแรงดันมีค่าใกล้เคียง 5 โวลต์

วิธีทดลอง

- 1) ทดลองที่ดินมีความชื้นน้อยที่สุดก่อน โดยใช้เซนเซอร์วัดความชื้นและเครื่องวัดความชื้นมาตรฐาน เสียบลงดินที่ความลึก 6-10 เซนติเมตร แล้วอ่านค่าความชื้น และค่าแรงดันจากเซนเซอร์ บันทึกค่าที่ได้ลงตาราง
- 2) เพิ่มค่าความชื้น โดยการเติมน้ำลงในดินที่ทดลองจากข้อ 1 อ่านค่าความชื้นและค่าแรงดัน บันทึกลงตาราง ค่อยๆเพิ่มระดับความชื้น โดยการรดน้ำไปอย่างช้าๆ

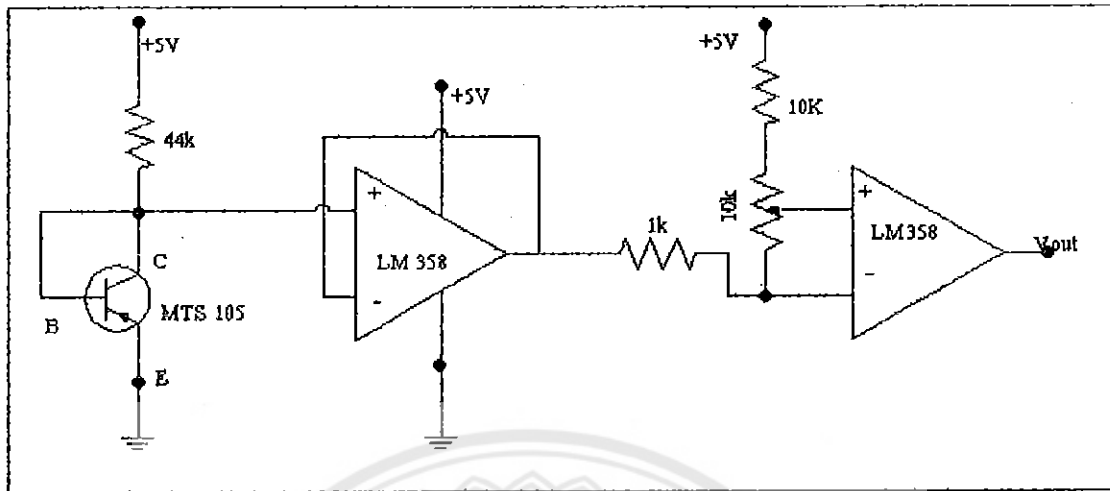
3.3 วงจรวัดอุณหภูมิ

3.3.1 หลักการทำงาน

เมื่ออุณหภูมิมีค่าเปลี่ยนแปลงจะทำให้แรงดันที่ V_{bb} ผ่าน MTO 105 มีค่าเปลี่ยนแปลง โดยมีการเปลี่ยนแปลง $-2mV/1$ องศาเซลเซียส

ลักษณะการทำงาน

เมื่ออุณหภูมิเปลี่ยนแปลง ค่าแรงดันที่ได้จะผ่าน LM 358 ตัวแรกจะมีค่าอัตราขยาย 1 เท่า เพื่อใช้กลับเฟส ส่งค่าไปยัง LM 358 ตัวถัดไปโดยมีอัตราขยาย 20 เท่า ส่วนการเลือกค่าความต้านทานปรับค่าได้จะทำการปรับเพื่อให้ได้ระดับที่เหมาะสมกับวงจรแปลงสัญญาณดิจิตอล



รูปที่ 3.3 วงจรวัดอุณหภูมิ

วิธีทดลอง

- 1) นำเซนเซอร์วัดอุณหภูมิ วัดค่าแรงดันเอาท์พุทเทียบกับเทอร์โมมิเตอร์ แล้วบันทึกค่าอุณหภูมิและค่าแรงดันลงในตารางผลการทดลอง
- 2) ทำการวัดที่อุณหภูมิต่าง ๆ กันหลาย ๆ ครั้ง

3.4 การทดลองระบบทั้งหมด

การต่อระบบทั้งหมดเข้าด้วยกันจะต่อคอมพิวเตอร์เข้ากับบอร์ดควบคุมโดยต่อเข้าที่ com1 หรือ com2 ของคอมพิวเตอร์แล้วมาต่อกับบอร์ดควบคุมที่ชุด ไมโครคอนโทรลเลอร์ ที่ไมโครคอนโทรลเลอร์จะเชื่อมต่อกับวงจรขับวาล์วโดยจะส่งออกที่ port P1.0-P1.7 ของไมโครคอนโทรลเลอร์ สัญญาณดิจิทัลที่ไมโครคอนโทรลเลอร์จะรับเข้าจะได้จากชุด A/D ซึ่งจะรับสัญญาณอนาลอกมาจาก 2 ส่วน คือ จากวงจรวัดความชื้น และ วัดอุณหภูมิ ดังรูปที่ 3.4

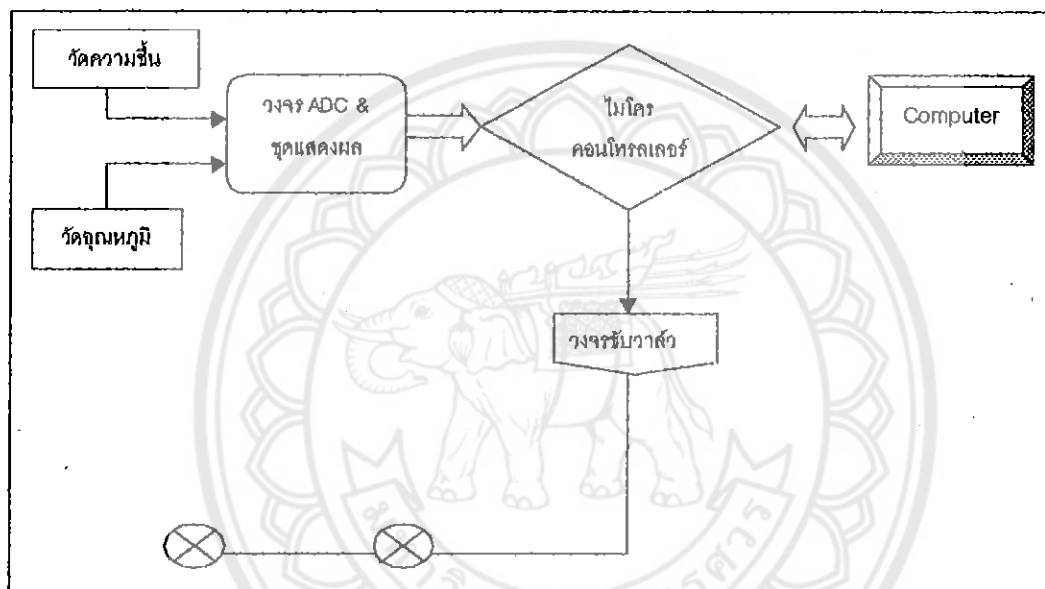
ลักษณะการทำงาน

เมื่อต่อระบบโดยรวมเข้ากันแล้วเริ่มจากรับค่าการสั่งงานมาจากโปรแกรมบนคอมพิวเตอร์ให้ชุดควบคุมเริ่มทำการประมวลผลค่าความชื้น เพื่อเข้าเงื่อนไขการตัดสินใจ และทำการวนตรวจเช็คค่าความชื้นทุก ๆ 10 นาทีจนผ่านไปเป็นเวลา 1 ชั่วโมงจึงหยุดทำงาน

วิธีทดลอง

- 1) นำเซนเซอร์วัดความชื้นเสียบแปลงทดลองที่ต้องการ โดยเสียบที่ความลึก 6-10 เซนติเมตร

- 2) เปิดแหล่งจ่ายไฟให้กับอุปกรณ์ทุกชิ้น
- 3) เชื่อมต่อระหว่างบอร์ดชุดควบคุมกับคอมพิวเตอร์ด้วย RS232
- 4) ทำการกำหนดค่า เวลาและวาล์ว ที่ต้องการเปิด-ปิด จากโปรแกรมบนเครื่อง คอมพิวเตอร์
- 5) สังเกตผลการทำงานของวาล์ว เมื่อเวลาผ่านไปแล้วบันทึกผล
- 6) เปลี่ยนตำแหน่งแท่งอิเล็กทรอนิกส์ให้มีความชื้นที่ต่างจากครั้งแรก และสังเกตการทำงานของวาล์ว



รูปที่ 3.4 แผนภาพการต่อระบบรวม

บทที่ 4

ผลการทดลอง และวิเคราะห์

จากการประกอบชุดอุปกรณ์ต่าง ๆ แล้วเมื่อนำมาทดลองกับค่าจริงโดยทำการวัดความชื้นจากดินที่มีสภาพความชื้นที่แตกต่างกัน ส่วนของการทดลองค่าอุณหภูมิจากเซนเซอร์จะวัดที่อุณหภูมิห้องในแต่ละสถานที่ ที่มีอุณหภูมิแตกต่างกัน โดยทำตามการทดลองในบทที่ 3 จะได้ค่าต่าง ๆ ดังแสดงในตารางข้างล่าง ดังนี้

4.1 ค่าแรงดันเอาต์พุตจากวงจรวัดความชื้น

ค่าแรงดันที่ได้จาก เซนเซอร์ วัดความชื้นจากการทดลองในบทที่ 3 โดยทำการวัดเทียบกับเครื่องวัดความชื้นมาตรฐาน

ตารางที่ 4.1 ผลการทดลองวัดค่าจากเซนเซอร์ วัดความชื้น

ระดับแรงดันจากเซนเซอร์ (โวลต์)	ค่าความชื้น (เปอร์เซ็นต์)
0.8	30
1.1	40
1.68	50
2.53	60
3.18	70
4.23	80
4.58	90

เนื่องจากในกรณีที่ทำการศึกษาใช้พืชกระถาง ซึ่งมีความต้องการความชื้นในช่วง 50-70 % จึงทำการสรุปตารางค่าความชื้นให้อยู่ในช่วงที่ต้องการได้ดังตารางที่ 4.2

ตารางที่ 4.2 สรุปผลแรงดันจากเซนเซอร์ วัดความชื้น

ระดับแรงดันจากเซนเซอร์ (โวลต์)	ค่าความชื้น (เปอร์เซ็นต์)
0-1.25	น้อยกว่า 50%
1.25-3.50	50-70%
3.50-5.00	มากกว่า 70%

4.2 วิเคราะห์ผลการทดลองจากวงจรวัดความชื้น

ค่าที่ได้จากเซนเซอร์ออกมาเป็นค่าแรงดันเมื่อทำการเปรียบเทียบกับเครื่องวัดมาตรฐานแล้วปรากฏว่าค่าความชื้นน้อยกว่า 25 % จะมีแรงดันอยู่ในช่วง 0 – 1.25 โวลต์ ถ้าค่าความชื้นอยู่ในช่วง 50 – 70 % จะมีค่าแรงดัน 1.25 – 3.5 โวลต์ และถ้าความชื้นมากกว่า 70 % จะมีค่าแรงดัน 3.5 – 5.0 โวลต์ จะเห็นได้ว่าที่ระดับความชื้นที่แตกต่างกันจะให้ค่าแรงดันจากเซนเซอร์แตกต่างกันด้วยโดยมีแนวโน้ม คือ ค่าความชื้นมากค่าแรงดันก็จะมากด้วย

4.3 ค่าแรงดันเอาต์พุตที่ได้จากวงจรวัดอุณหภูมิ

จากการวัดค่าแรงดันจากเซนเซอร์ วัดอุณหภูมิเทียบกับเทอร์โมมิเตอร์

ตารางที่ 4.3 ผลการทดลองวัดค่าเอาต์พุตจากวงจรวัดอุณหภูมิ

ระดับแรงดัน (โวลต์)	ค่าอุณหภูมิ (องศาเซลเซียส)
1.65	20
1.64	22
1.63	24
1.62	26
1.61	28
1.60	30
1.59	32
1.58	34

ตารางที่ 4.3 ผลการทดลองวัดค่าเอาต์พุตจากวงจรวัดอุณหภูมิ (ต่อ)

ระดับแรงดัน (โวลต์)	ค่าอุณหภูมิ (องศาเซลเซียส)
1.57	36
1.56	38
1.55	40
1.54	42
1.53	44
1.52	46
1.51	48
1.50	50

4.4 วิเคราะห์ผลการทดลองจากวงจรวัดอุณหภูมิ

อุณหภูมิที่ต่างกันทำให้ค่าแรงดันเอาต์พุตที่ได้จากวงจรมีค่าแตกต่างกัน ซึ่งที่อุณหภูมิมีค่าสูงจะได้ค่าแรงดันจากเครื่องวัดมีค่าน้อย เป็นไปตามคุณลักษณะของทรานซิสเตอร์ คือมีระดับแรงดันเปลี่ยนแปลงที่ $-2\text{mV}/1^\circ\text{C}$ ผลต่างของระดับแรงดันที่ได้สามารถนำไปเปรียบเทียบกับวงจร A/D ได้

4.5 ค่าแรงดันเมื่อผ่านการแปลงจากวงจร A/D

ตารางที่ 4.4 ค่าแรงดันเมื่อแปลงเป็นค่าดิจิทัล

ค่าแรงดัน (V)	สัญญาณดิจิทัล 8 Bit								ดิจิทัล HEX
	8	7	6	5	4	3	2	1	
0.00	0	0	0	0	0	0	0	0	00
0.25	0	0	0	0	1	1	0	1	0D
0.50	0	0	0	1	1	0	1	0	1A
0.75	0	0	1	0	0	1	1	1	29
1.00	0	0	1	1	0	0	1	1	33
1.25	0	1	0	0	0	1	0	1	41
1.50	0	1	0	0	1	0	0	1	4D
1.75	0	1	0	1	1	0	1	0	5A

ตารางที่ 4.4 ค่าแรงดันเมื่อแปลงเป็นค่าดิจิทัล (ต่อ)

ค่าแรงดัน (V)	สัญญาณดิจิทัล 8 Bit								ดิจิทัล HEX
	8	7	6	5	4	3	2	1	
2.00	0	1	1	0	0	1	1	1	67
2.25	0	1	1	1	0	0	0	0	74
2.50	1	0	0	0	0	1	0	0	80
2.75	1	0	0	0	1	1	0	1	8D
3.00	1	0	0	1	1	0	1	0	9A
3.25	1	0	1	0	0	1	1	1	A7
3.50	1	0	1	1	0	0	0	0	B4
3.75	1	1	0	0	0	1	0	0	C0
4.00	1	1	0	0	1	1	0	1	CD
4.25	1	1	0	1	1	0	1	0	DA
4.50	1	1	1	0	0	1	1	1	E7
4.75	1	1	1	1	0	1	0	0	F4
5.00	1	1	1	1	1	1	1	1	FF

4.6 ผลการทำงานทั้งระบบ

เมื่อนำระบบมาประกอบเข้าทำงานร่วมกัน ดังรูปที่ 3.4 และทำการทดลองโดยเปลี่ยนแปลงค่าความชื้นและสังเกตผลการทำงานของวาล์ว จากตารางที่ 4.5 กำหนดค่าเริ่มต้นของความชื้นในดินที่มีค่าต่ำกว่า 50 % เพื่อทำการทดสอบระบบว่าสามารถทำการเปิดวาล์วได้หรือไม่ และเมื่อระบบทำการเปิดวาล์วแล้ว ค่าความชื้นมีค่ามากขึ้นจนถึงช่วง 50 – 70 % ระบบสามารถปิดวาล์วได้

ตารางที่ 4.5 ผลการทดลองการทำงานทั้งระบบเมื่อเริ่มที่ดินมีความชื้นน้อย

เวลา	ค่าความชื้น	ลักษณะการทำงานของวาล์ว
8:00	30	ON
8:10	30	ON
8:20	40	ON
8:30	60	OFF
8:40	60	OFF
8:50	60	OFF
9:00	60	OFF

จากตารางที่ 4.6 กำหนดค่าเริ่มต้นของความชื้นในดินให้มีค่ามากกว่า 70 % เพื่อทำการทดสอบระบบว่ายังสามารถวัดระดับความชื้นที่มีค่ามากแล้วจะไม่สั่งงานเปิดวาล์วและเมื่อระดับความชื้นลดลงแล้วระบบสามารถเปิดวาล์วได้หรือไม่

ตารางที่ 4.6 ผลการทดลองการทำงานของระบบเมื่อเริ่มที่ดินมีความชื้นมาก

เวลา	ค่าความชื้น	ลักษณะการทำงานของวาล์ว
9:00	70	OFF
9:10	70	OFF
9:20	60	OFF
9:30	50	OFF
9:40	40	ON
9:50	40	ON
10:00	50	OFF

4.7 วิเคราะห์ผลการทดลองการทำงานของระบบ

จากการทดลองจะเห็นได้ว่าระบบสามารถทำงานได้ตามวัตถุประสงค์ คือ เมื่อค่าความชื้นต่ำกว่า 50 % ระบบสั่งงานให้วาล์วเปิดได้ และเมื่อความชื้นอยู่ในช่วง 50 – 70 % ระบบสั่งให้วาล์วหยุดทำงานได้

4.8 โปรแกรมที่ใช้ในการติดต่อกับผู้ใช้

โปรแกรมที่ใช้ในการเขียนคือ โปรแกรม Delphi ซึ่งเป็นโปรแกรมที่ใช้ในการสร้างแอปพลิเคชันเพื่อใช้งานระบบปฏิบัติการ Windows (9x) ซึ่งมีประสิทธิภาพการทำงานที่เยี่ยมยอด

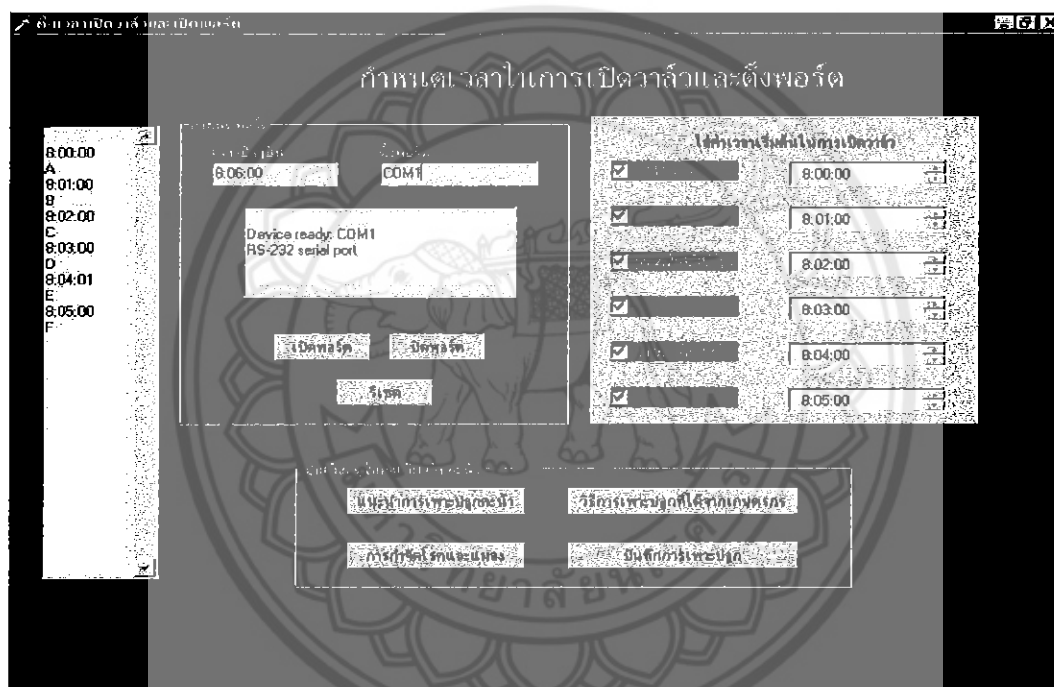
Delphi ผลิตโดยบริษัท Inprise ซึ่งเป็นเครื่องมือพัฒนาแอปพลิเคชันแบบ Visual Programming ทำให้เราสามารถเห็นผลลัพธ์การทำงานไปพร้อม ๆ กับการลงมือสร้างแอปพลิเคชัน

4.8.1 ขั้นตอนการใช้โปรแกรมในส่วนที่ติดต่อกับผู้ใช้

1. หน้าแรกเป็นส่วนที่ให้ผู้ใช้งานกำหนดค่าเวลาในการเริ่มทำงานของวาล์ว โดยการทำตามขั้นตอนกำหนดส่วนติดต่อส่งข้อมูลให้กับคอนโทรลเลอร์ โดยกำหนดพอร์ตว่าเป็นพอร์ต 1 หรือพอร์ต 2 ถ้าเป็นพอร์ต 1 ให้ใส่ com1 ลงไปในช่องดังรูป ถ้าเป็นพอร์ต 2 ก็ให้ใส่ com2 ลงไปในช่องดังรูป

- 1.1) จากนั้นกดปุ่ม OpenPort
- 1.2) ใส่ค่าเวลาเริ่มต้นในการทำงานของวาล์วของแต่ละตัว
- 1.3) กดปุ่ม Checkbox ของวาล์วตัวที่กำหนดเวลาเริ่มต้นในการทำงานแล้ว

เมื่อทำตามขั้นตอนดังกล่าวแล้ว โปรแกรมจะทำการส่งค่าไปให้คอนโทรลเลอร์ เพื่อทำการตรวจสอบความชื้นว่ามีค่าน้อยหรือมาก ถ้าถึงเวลาที่กำหนดแล้ว และมีค่าความชื้นต่ำ วาล์วก็จะเปิด และจะปิดโดยอัตโนมัติ เมื่อมีค่าความชื้นสูง โดยวาล์วตัวที่ไม่มีการกำหนดค่าเวลาเริ่มต้นในการทำงาน ก็จะไม่มีการทำงาน ดังรูป



รูปที่ 4.1 หน้าจอควบคุมการเปิด/ปิดวาล์ว

2. หน้าที่สองเป็นส่วนที่อธิบายขั้นตอนการเพาะปลูกกะน้ำ โดยใช้ข้อมูลดิบที่เก็บมาจากผู้ผลิต มาแสดงเพื่อเป็นแนวทางในการเพาะปลูก

วิธีการเพาะปลูก

วิธีการเพาะปลูกและข้อมูลดิบของอัตราการเจริญเติบโตของกะน้ำในแต่ละสัปดาห์

1. ขั้นตอนการเตรียมก่อนการปลูก

- 1) โถ
- 2) แพลงกว้าง 3 ฟุต ทว่านเมล็ด 7 กระป๋องต่อ 1 โถ

2. ขั้นตอนการดูแลรักษาหลังการทว่านเมล็ด

- 1) ใส่ปุ๋ยธาตุอาหารทุกครั้งที่ (15-15)
- 2) รดน้ำ 2 ครั้ง กลางวัน-เย็น
- 3) 45-50 วัน ตัดขายได้ (ตัดแยกแปลง)
- 4) ได้ผลผลิตประมาณ 2500 กิโลกรัมต่อทว่านโถ

สัปดาห์ที่	ความยาวรอบลำต้น (cm)	ความสูงจากโคนถึงยอด(cm)	จำนวนใบ (ใบ)
1	0.58	1	2
2	0.83	2.5	3
3	2.3	12.6	4
4	3.9	21.4	5
5	4.3	25	6
6	5.8	27.2	7

กลับไปที่เมนูหลัก

รูปที่ 4.2 หน้าจอแนะนำเสนอวิธีการเพาะปลูกกะน้ำ

3. เป็นส่วนที่ใช้ในการบันทึกอัตราการเจริญเติบโตของค่น้ำของผู้ปลูก ทำให้สามารถเก็บข้อมูลไว้ในการประเมินผลการเพาะปลูกได้ โดยสามารถกรอกข้อมูลลงไปในตารางได้ดังรูป

บันทึกการเพาะปลูก

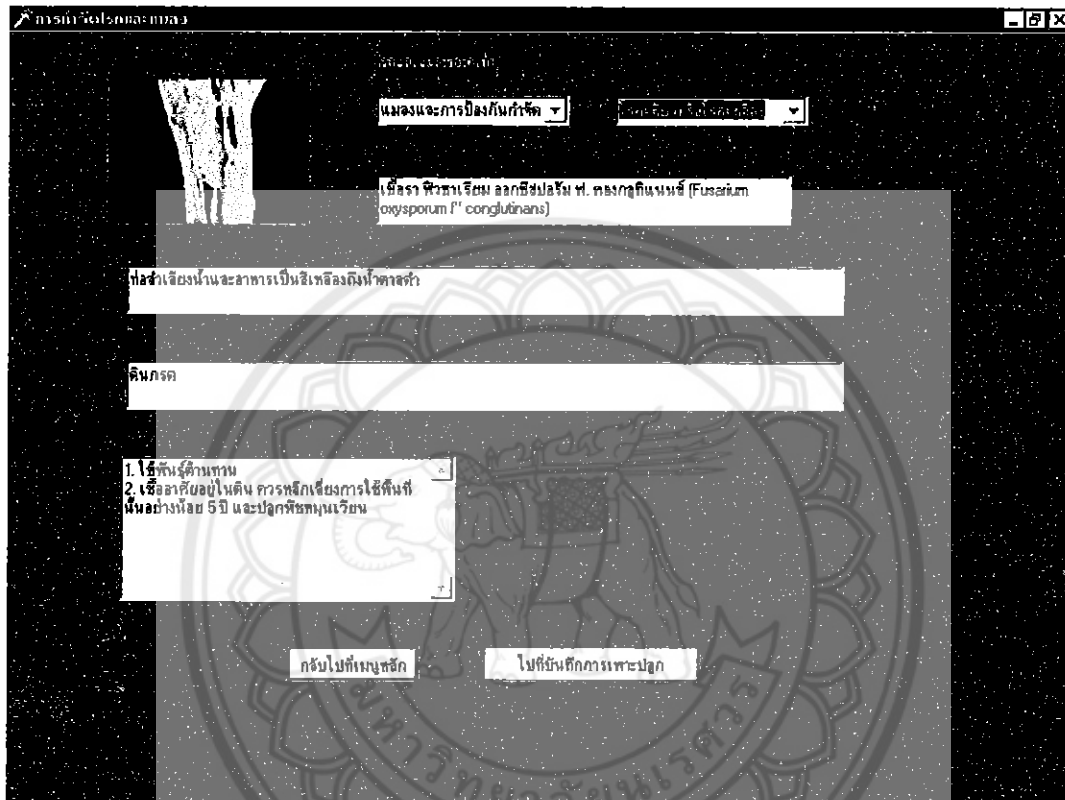
ตารางสำหรับผู้ปลูกบันทึกอัตราการเจริญเติบโตของค่น้ำและแสดงอุณหภูมิ ระดับความชื้นในดิน

วันที่	ความยาวรอบต้น (cm)	ความสูงจากโคนถึงยอด(cm)	จำนวนใบ (ใบ)	อุณหภูมิเฉลี่ย (celsius)
1				
2				
3				
4				
5				
6				

กลับไปที่เมนูหลัก

รูปที่ 4.3 แบบฟอร์มการบันทึกข้อมูลการเจริญเติบโตของค่น้ำ

4. เป็นส่วนที่แสดงข้อมูลในการแนะนำการกำจัดและป้องกันโรคและแมลงของคะน้า โดยถ้าผู้ใช้ต้องการทราบถึงแนวทางในการแก้ไขปัญหาของโรคและแมลงของคะน้าก็สามารถทำได้โดยการคลิกเลือกที่ช่องเลือกด้านบน เลือกโรคและแมลงต่างๆ ข้อมูลก็จะปรากฏออกมา



รูปที่ 4.4 แบบฟอร์มนำเสนอวิธีการแก้ไขปัญหาโรคพืชและแมลงของคะน้า

บทที่ 5

บทสรุป

5.1 สรุปผลการทดลอง

5.1.1 วงจรวัดความชื้น

วงจรวัดความชื้นจะต้องมีเซนเซอร์เป็นตัวตรวจจับระดับความชื้นในดิน ในการติดตั้งจะต้องเสียบโพรบให้เข้าลึกไปในดินประมาณ 15 cm และเอียงทำมุมประมาณ 45°C จึงจะทำให้ได้ค่าที่ถูกต้องเมื่อเปรียบเทียบกับค่าที่วัดได้จาก เทนซิโอมิเตอร์ และสายนำสัญญาณ ไม่ต้องคำนึงถึงระดับสัญญาณจะอ่อนเพราะเราใช้การส่งสัญญาณแบบแหล่งจ่ายกระแส แล้วจึงใช้วงจรแปลงเป็นระดับแรงดัน ในช่วง 0-5 V โดยระดับแรงดันจะขึ้นอยู่กับค่าระดับความชื้น ซึ่งจะมีความสัมพันธ์กันได้ว่า ถ้าค่าความชื้นน้อยค่าแรงดันจะมีค่าน้อย แต่ถ้าค่าความชื้นมากก็จะได้แรงดันมีค่ามากแล้วส่งค่าไปที่วงจรแปลงเป็นดิจิทัล 8 บิต และส่งให้กับไมโครคอนโทรลเลอร์ ผลที่ได้จากการทดลองเราสามารถสรุปได้ว่า ถ้าค่าความชื้นน้อยกว่า 50 % ระบบจะทำการเปิดน้ำ แต่ถ้าค่าความชื้นอยู่ในช่วง 50 – 70 % ระบบก็จะทำการปิดน้ำซึ่งเป็นไปตามที่ได้ออกแบบ

5.1.2 วงจรวัดอุณหภูมิ

วงจรวัดอุณหภูมิเราได้ทำการทดลองที่อุณหภูมิในช่วง $20 - 50^{\circ}\text{C}$ วงจรจะไม่สามารถเปลี่ยนแปลงได้ในสภาวะทันทีทันใด เพราะเป็นการวัดอุณหภูมิในอากาศ จึงไม่จำเป็นที่ต้องเปลี่ยนแปลงรวดเร็วและเราไม่ได้คำนึงถึงความแม่นยำมากนัก เราเพียงใช้เป็นข้อมูลให้กับผู้ใช้งานเท่านั้น

5.1.3 การเปิด-ปิดวาล์ว

การควบคุมการเปิด-ปิดวาล์ว ผู้ใช้จะกำหนดเงื่อนไขเป็นช่วงเวลา และเลือกที่จะเปิดวาล์วในครั้งที่ต้องการ โดยก่อนที่จะสั่งงานจะต้องทำการเปิดซีเรียลพอร์ต (serial port) ก่อน ไม่ว่าจะ เป็น com1 หรือ com2 ก็ได้ตามที่เราได้ต่อสายไว้แล้ว ใช้โปรแกรมที่เราเขียนไว้ในการติดต่อก่อนเวลาที่นั้นเป็นเวลาที่พื้นฐานของเครื่องคอมพิวเตอร์โดยเราสามารถดึงได้จากคอมพิวเตอร์เครื่องนั้นและระบบจะทำการเปรียบเทียบกับเวลาที่เราร้องการแล้วจึงสั่งการเปิดวาล์วโดยอัตโนมัติ

5.1.4 โปรแกรมการดูแลรักษาพีซี

โปรแกรมการดูแลรักษาพีซีที่เราศึกษาจะเป็นคำแนะนำ เป็นพีซีที่มีอายุประมาณ 45 วัน เราก็ได้มีคำแนะนำบอกไว้ ทั้งหัวข้อแนะนำการเพาะปลูก การดูแล รวมไปถึงโรคพืชที่เกิดขึ้นก็มีภาพตัวอย่างประกอบด้วย จึงทำให้ง่ายต่อการทำความเข้าใจ สำหรับผู้ที่ไม่เคยปลูกมาก่อนสามารถที่จะอ่านคำแนะนำและไปปฏิบัติได้ เราได้มีการทำแบบฟอร์มให้กับผู้ใช้ได้เก็บข้อมูลการเจริญเติบโตเพื่อช่วยให้ผู้ใช้ได้ทราบว่าขณะนั้นพีซีที่เราปลูกเจริญเติบโตเพียงใดแล้วเปรียบเทียบกับที่เราแนะนำ

5.2 ปัญหาที่พบในการทำโครงงาน

1. ความแม่นยำของอุปกรณ์วัดความชื้นและอุณหภูมิ เนื่องจากเป็นวงจรที่ได้ออกแบบเองจึงอาจมีข้อผิดพลาดได้เราจึงแก้ไขปัญหาโดยการทำการรับค่าความชื้นมาถึง 5 ครั้ง แล้วทำการหาค่าเฉลี่ย ก่อนที่จะมีการประมวลผล ในวงจรวัดอุณหภูมิเราไม่ได้คำนึงถึงความแม่นยำและความรวดเร็วมากนัก แต่เราได้มีการปรับเทียบกับเทอร์โมมิเตอร์แล้ว ได้ค่าที่ไม่ผิดพลาดมากนัก
2. การเชื่อมต่อของคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ จะเชื่อมต่อโดยใช้มาตรฐานของ IEEE คือ RS-232 เข้ากับคอมพิวเตอร์ที่พอร์ต com1 หรือ com2 และต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ ปัญหาที่พบ คือ การส่งรับค่าระหว่างอุปกรณ์ทั้งสองไม่สามารถทำได้เนื่องจากการเขียนโปรแกรมไม่ถูกต้อง จึงต้องทำการแก้ไขโปรแกรม
3. การทำงานร่วมกันของระบบ ในช่วงแรกที่ทำกรออกแบบระบบ จะทำงานกันเป็นส่วน ๆ แล้วจึงนำแต่ละส่วนมาเชื่อมต่อกันทำให้เกิดความผิดพลาดมากขึ้น จึงต้องทำการปรับเทียบในแต่ละส่วนใหม่ทั้งหมด เพื่อลดให้ระบบมีความผิดพลาดให้น้อยที่สุด
4. การกำหนดโครงสร้างของโปรแกรม เนื่องจากเป็นระบบที่ยังไม่เคยมีผู้ศึกษาจึงเป็นการยากที่จะเขียนโปรแกรมให้ทำงานได้ถูกต้อง เมื่อสร้างระบบได้แล้วก็เหลือในส่วนของโปรแกรมควบคุมและจัดการ ในการเปิด - ปิดควาล์ว การเขียนโปรแกรมจะแยกออกเป็นส่วน ๆ เพื่อง่ายในการเขียนและทดสอบการทำงานของแต่ละส่วนก่อน เมื่อทุกส่วนสามารถทำงานได้แล้วเราจึงนำทั้งหมดมารวมกันแล้วทดสอบอีกครั้ง

5.3 ประเมินผลและข้อเสนอแนะ

จากผลโครงงานวิจัย เมื่อเปรียบเทียบกับวัตถุประสงค์ที่เราตั้งไว้ ได้ผลดังนี้

1. การควบคุมการเปิด-ปิดน้ำ จะใช้วาล์วที่ควบคุมโดยไมโครคอนโทรลเลอร์โดยมีตัวแปรที่เป็นตัวควบคุมคือ ช่วงเวลา และระดับความชื้น การทำงานจะต้องให้ผู้ใช้เป็นผู้ตั้งงานเพียงแต่ไมโครคอนโทรลเลอร์มีหน้าที่ในการเปรียบเทียบระดับความชื้นแล้ว เมื่อความชื้นอยู่ในระดับที่พืช

ต้องการแล้ว ไมโครคอนโทรลเลอร์จะทำการปิดวาล์ว ในการเปิด-ปิดวาล์วจะเป็นอิสระจากกัน แต่ผู้ใช้ไม่สามารถที่จะให้เปิดวาล์วในเวลาเดียวกันได้ จะต้องสั่งเปิดในเวลาใกล้เคียงกันใน ระดับเป็นนาที่ เพราะในการตรวจวัดระดับความชื้นต้องใช้เวลาประมาณ 1 นาที เพื่อความถูกต้องแม่นยำ

2. โปรแกรมที่ช่วยในการดูแลจัดการพืชนั้นเราได้ทำเป็นกรณีศึกษาคือ พืชกระน้ำ โดยจะได้ข้อมูลมาจาก 2 แหล่งคือ จากการไปเก็บข้อมูลจากผู้ประกอบอาชีพปลูกผักอยู่ และจากหนังสือรวมหัตถ์บทความต่าง ๆ นำมารวบรวมและเรียบเรียงให้ผู้ใช้ได้ศึกษาแล้วนำไปปฏิบัติได้การเก็บข้อมูลการเจริญเติบโต ผู้ใช้จะต้องไปเก็บข้อมูลแล้วนำมากรอกในแบบฟอร์มที่มีให้เพื่อติดตามผลการเจริญเติบโตของกระน้ำได้
3. เมื่อระบบทำงานผิดพลาดผู้ใช้สามารถให้ระบบกลับสู่สภาวะปกติได้โดยกดปุ่มรีเซต (reset) ที่บอร์ดไมโครคอนโทรลเลอร์และรันโปรแกรมที่เครื่องคอมพิวเตอร์ใหม่อีกครั้ง แล้วระบบจะกลับสู่สภาวะเดิมได้ ระบบใช้ไฟ DC ทั้งหมด ในระดับแรงดัน 5 V, 12 V และ +/- 15 V ส่วนคอมพิวเตอร์จะใช้ไฟจากบ้านธรรมดา 220 V.

5.4 แนวทางการพัฒนาระบบต่อ

เนื่องจากโครงการนี้เป็นโครงการที่ริเริ่มขึ้นมาจากแนวความคิดของสมาชิกเอง ฉะนั้นขอบข่ายของโครงการอาจจะยังไม่เป็นที่สมบูรณ์ เนื่องด้วยเวลา และเงินทุนที่จำกัด ดังนั้นทางคณะผู้จัดทำโครงการจึงขอฝากแนวความคิดที่จะนำโครงการชิ้นนี้ไปพัฒนาต่อ เพื่อให้เกิดประโยชน์สูงสุดต่อไป

แนวทางการพัฒนาอาจแบ่งได้เป็นข้อ ๆ ดังนี้

1. ในภาคอุตสาหกรรม นำไปประยุกต์ใช้กับระบบควบคุม แทนการใช้ระบบ PLC เนื่องจากการใช้ระบบไมโครคอนโทรลเลอร์ จะสามารถขยายการควบคุมได้ง่ายและลดต้นทุนได้ดีกว่า
2. ในภาคการเกษตร อาจเพิ่มการควบคุม เช่นระบบการฉีดพ่นยาในแปลงพืช โดยนำตัวยาผ่านไปกับท่อให้น้ำ เพื่อไม่เป็นอันตรายต่อเกษตรกรในการสัมผัสกับสารเคมีโดยตรง
3. เป็นที่ทราบกันดีว่า การเกษตรนั้นจำเป็นต้องพึ่งสภาพอากาศเป็นสำคัญ การนำระบบเชื่อมต่อกับ ระบบอินเทอร์เน็ตติดต่อกับกรมอุตุนิยมวิทยา เพื่อทราบข้อมูล ดิน ฟ้า อากาศ ช่วยในการตัดสินใจของเกษตรกร หรือ ทราบข้อมูลการระบาดของศัตรูพืชในฤดูกาลต่าง ๆ สามารถเพิ่มประสิทธิภาพในการตัดสินใจ
4. พัฒนาโปรแกรมติดต่อกับผู้ใช้ให้สามารถคำนวณค่าใช้จ่ายในการลงทุน เพื่อนำไปคิดเป็นต้นทุนการผลิตได้ และสามารถเก็บสถิติของพืชแต่ละรุ่นเพื่อใช้เป็นพื้นฐานการพัฒนาในรุ่นต่อไป

บรรณานุกรม

- กนก กุศลมาลัยนุกูล และ ไกรวุฒิ มั่นเสถียรสิน. “คู่มือการเขียนโปรแกรมด้วย Delphi 4”
กรุงเทพฯ : ไซคเซส มีเดีย จำกัด.
- ดวงแก้ว สวามิภักดิ์, คร., “การโปรแกรมภาษา C” กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2539.
- ทวีเดช รุ่งเรืองอนันต์ และ สราวุธ กระจุกทิพย์. “ฟาร์มเลี้ยงไก่ไข่ควบคุมด้วยระบบ
คอมพิวเตอร์” หน้า 46-49.
- ทศิมพร จรัสอารมณ. “เครื่องวัดอุณหภูมิรุ่นตัวเสกขนาดมือถือ” Hobby Electronics. (63) :
กันยายน 2539. หน้า 36-40.
- รัชพงษ์ นิลอุบล, ร.อ., “ไมโครคอนโทรลเลอร์ เล่ม 1” กรุงเทพฯ : เอ็มแอนดีอี, 2541.
- นิวัฒน์ ปัญญาเจริญ. “เครื่องวัดความชื้นในดิน” Hobby Electronics. (68)
: กุมภาพันธ์ 2540. หน้า 97-98
- นุกูล กระจาย. “การเขียนโปรแกรมแบบวิซวลด้วย Delphi 4” กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2542.
- ชิน ภู่วรรณ. “อิเล็กทรอนิกส์อุตสาหกรรม” กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2536.
- วิโรจน์ อัสวรงค์, ชัชวาล เต็มฤทธิวงศ์และกรชูลี ใสสัตต. “การใช้งานออปแอมป์และลิเนียร์
ไอซี” กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2536.
- สุนทร วิบูลย์พงษ์. “การใช้งานไมโครคอนโทรลเลอร์ ตระกูล 8051”
- อำนาจ คล่องกิจกุล และ คณะ. “ระบบไฟฟ้าพลังงานทดแทนแบบผสม” กรุงเทพฯ :
มหาวิทยาลัยเกษตรศาสตร์, 2541-2542.
- เมืองทอง ทวนทวี และ สุธีรัตน์ ปัญญาโคณะ. “หลักการปลูกผัก สวนผัก 1” พิมพ์ครั้งที่ 2.
กรุงเทพฯ : โรงพิมพ์ทั้งฮั่วชิน, 2532.
- แคมป์เบล, โจ. “ไขปัญหา RS-232” แปลจาก The RS-232 Solution. โดย ทวีชัย ภูริทิพย์.
กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2538.
- โยธิน เปรมปราณีรัชต์, รศ. ดร. ชม กัมปาน, รศ และ คณะ. “เครื่องมือวัดทางอิเล็กทรอนิกส์และ
เทคนิคการวัด” กรุงเทพฯ : โรงพิมพ์นำอักษรการพิมพ์.
- ไกรวุฒิ ไรจน์ประเสริฐสุด. “ไมโครโปรเซสเซอร์ 2”. กรุงเทพฯ : ซีเอ็ดดูเคชั่น, 2539.





ภาคผนวก ก.

โปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์

////!!!!!!!### Simulation System Automatic Control Sprinkle By PC. ###!!!!!!!

```
#include <reg51.h>
```

```
#include <absacc.h>
```

```
#include <math.h>
```

```
#include <string.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
sbit P1_0 = 0x90;
```

```
sbit P1_1 = 0x91;
```

```
sbit P1_2 = 0x92;
```

```
sbit P1_3 = 0x93;
```

```
sbit P1_4 = 0x94;
```

```
sbit P1_5 = 0x95;
```

```
sbit P1_6 = 0x96;
```

```
sbit P1_7 = 0x97;
```

```
sbit P3_7 = 0xB7;
```

```
#define Pa XBYTE[0xE0E0]
```

```
#define Pb XBYTE[0xE0E1]
```

```
#define Pc XBYTE[0xE0E2]
```

```
#define Pcon XBYTE[0xE0E3]
```



```
#define LCD_CONTROL XBYTE[0xE0C0]
#define LCD_BUSY_FLAG XBYTE[0xE0C1]
#define LCD_WRITEDATA XBYTE[0xE0C2]
#define LCD_READDATA XBYTE[0xE0C3]
```

```
#define ON=1
#define OFF=0
```

```
xdata unsigned int i;
```

```
xdata char zz[16],j;
```

```
//data unsigned char clock0;
```

```
//data unsigned char clock1;
```

```
xdata char float_string[16];
```

```
void delay(unsigned int time)
```

```
{
    data unsigned int i;
    for(i= 0;i < time;i++)
        {
        }
}
```

```
void wait_for_lcd_ready (void)
```

```
{
    while ((LCD_BUSY_FLAG & 0x80) != 0)
        {
```

```
    }  
    //wait util buy flag = 0 (lcd ready)  
}  
  
void specified_lcd(void)  
{  
    //for lcd more than 2 line  
    //and 5x7 dot per cursor font  
    LCD_CONTROL=0x38; //function set bit5=1  
    wait_for_lcd_ready ();  
}  
  
void clear_lcd(void)  
{  
    LCD_CONTROL=0x01;  
    //bit0=1 clear lcd  
    //and cursor goto top left  
    wait_for_lcd_ready ();  
}  
  
void cursor_on(void)  
{  
    LCD_CONTROL = 0x0f; //bit2=1 led monitor on  
    wait_for_lcd_ready ();  
}  
  
void cursor_off(void)  
{  
    LCD_CONTROL = 0x0c;
```



```
wait_for_lcd_ready ();  
}  
  
void entry_mode_set(void)  
{  
    LCD_CONTROL = 0x02;  
    //bit0=0 after put data shift cursor to right  
    //bit1=1 after read/write dd ram address increase1
```

```
wait_for_lcd_ready ();  
}
```

```
void init_lcd(void)  
{  
    specified_lcd();  
    cursor_on();  
    clear_lcd();  
    entry_mode_set();  
}
```

```
void WChar(char ch)  
{  
    LCD_WRITEDATA = ch;  
    wait_for_lcd_ready();  
}
```

```
void gotoxy(unsigned char row, unsigned char col)  
{
```

```
if(row == 1) LCD_CONTROL = 0x80+col-1; else
if(row == 2) LCD_CONTROL = 0xc0+col-1; else
if(row == 3) LCD_CONTROL = 0x90+col-1; else
if(row == 4) LCD_CONTROL = 0xd0+col-1;
wait_for_lcd_ready ();
}

void WCharxy(unsigned char row,unsigned char col,unsigned char ch)
{
    gotoxy(row,col);
    WChar(ch);
}

void string_to_lcd (unsigned char row,unsigned char col,char *st)
{
    data unsigned char mcol,colfinish,len;

    len = strlen(st);
    colfinish=col+len-1;
    if (colfinish > 16) colfinish=16;

    for (mcol=col; mcol<=colfinish; mcol++)
        {
            WCharxy(row,mcol,st[mcol-col]);
        }
}

void string_to_lcd32 (char *st)
{
```

```
data unsigned char mcol,colfinish,len,i;
xdata char display_lcd[32];

for (i=0;i<=31;i++)
    {
        display_lcd[i]=' ';
    }
len = strlen(st);

if (len>32) len=32;
for (i=0;i<=len-1;i++)
    {
        display_lcd[i]=st[i];
    }

for (mcol=1; mcol<=16; mcol++)
    {
        WCharxy(1,mcol,display_lcd[mcol-1]);
    }

for (mcol=1; mcol<=16; mcol++)
    {
        WCharxy(2,mcol,display_lcd[(mcol+16)-1]);
    }
}

void string_to_lcd32RotateLeft (char *st)
{
    data unsigned char mcol,colfinish,len,i,temp;
    xdata char display_lcd[32];
```

```
for (i=0;i<=31;i++)
    {
        display_lcd[i]=' ';
    }
len = strlen(st);

if (len>32) len=32;
for (i=0;i<=len-1;i++)
    {
        display_lcd[i]=st[i];
    }

while (1)
    {
        string_to_lcd32(display_lcd);

        temp=display_lcd[0];
        for (i=1;i<=31;i++)
            {
                display_lcd[i-1]=display_lcd[i];
            }
        display_lcd[31]=temp;

        delay(10000);
    }
}

cal_temp()
```

```
{  
  
    xdata unsigned char aaa;  
    xdata float y,y1,x,x1,a,b,xx;  
    xdata char scr[15],s[3],c[3],r[3];  
    xdata unsigned int l;  
  
    x=170.0;           //Level ref  
    a=0.1;            //slope  
    b=26.0;           //Temp ref  
  
    Pa=0x18;  
    delay(90000);  
    Pa=0x10;  
    delay(90000);  
    delay(90000);  
  
    aaa=Pb;  
    l=aaa;  
    delay(90000);  
    sprintf(scr,"%d",l);  
    scr[3]='\0';  
    xx=atof(&scr);  
  
    sprintf(scr,"%f",xx);  
    // string_to_lcd(1,1,scr);  
    delay(90000);  
    delay(90000);  
    Pc=xx;  
    if(xx==x)
```

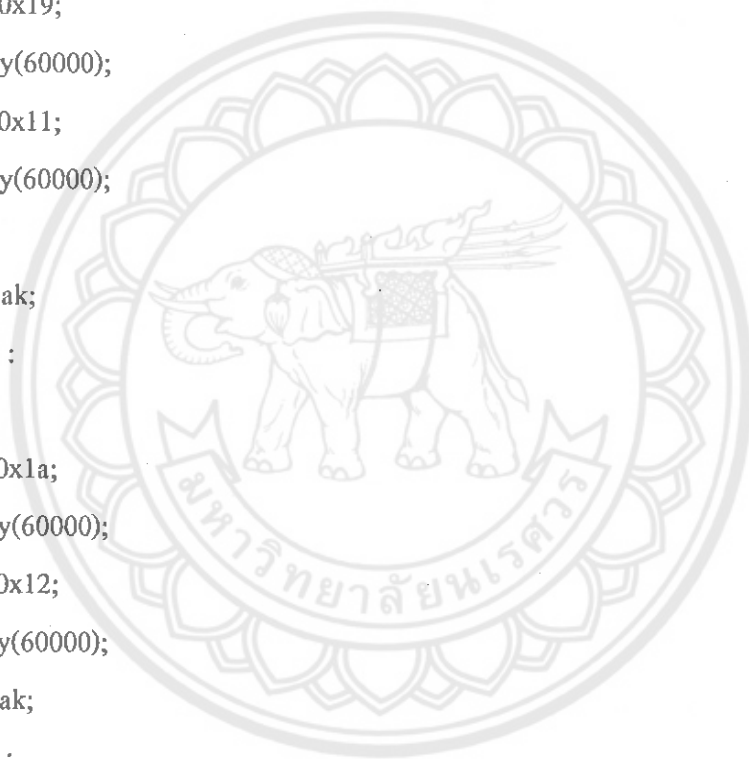
```
    {  
        y=b;  
        string_to_lcd(1,1,"Temp ="); //7  
        sprintf(s,"%3.2f",y);  
        string_to_lcd(1,8,s);  
        delay(3000);  
    }  
    if(xx!=x)  
    {  
        x1=x-xx;  
        y=(a*x1)+b;  
        string_to_lcd(1,1,"Temp ="); //7  
        sprintf(c,"%3.2f",y);  
        string_to_lcd(1,8,c);  
        delay(3000);  
    }  
}
```

```
void init_serial(void)  
{  
    SCON = 0x52;  
    PCON=0x00;  
    TMOD = 0x20;  
    TH1 = 0xfd;  
    TR1 = 1;  
    // TI = 1;  
    // RI = 0;  
}
```

```
void start(int i)
{
    switch(i)
    {
        case 1 :
            {
                Pa=0x19;
                delay(60000);
                Pa=0x11;
                delay(60000);

            }break;
        case 2 :
            {
                Pa=0x1a;
                delay(60000);
                Pa=0x12;
                delay(60000);
            }break;
        case 3 :
            {
                Pa=0x1b;
                delay(60000);
                Pa=0x13;
                delay(60000);

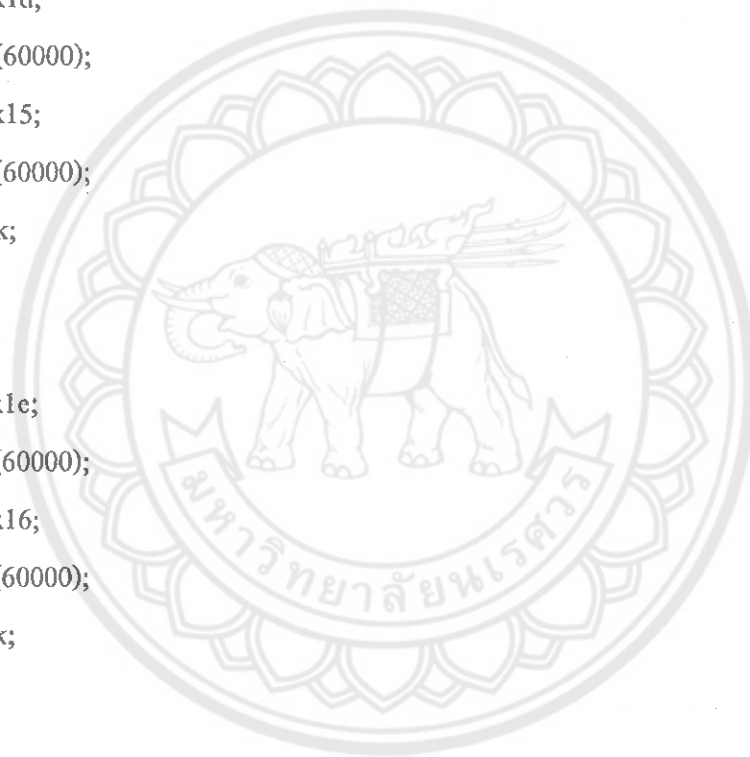
            }break;
        case 4 :
            {
```



```
    Pa=0x1c;
    delay(60000);
    Pa=0x14;
    delay(60000);
    }break;
case 5 :
    {
    Pa=0x1d;
    delay(60000);
    Pa=0x15;
    delay(60000);
    }break;
case 6 :
    {
    Pa=0x1e;
    delay(60000);
    Pa=0x16;
    delay(60000);
    }break;
    }
}

float get_value(float M)
{
    xdata unsigned int j;
    xdata float A,B,C,D,E;

    for (j=0;j<=4;j++)
    {
```




```
if(j==0)
{
    start(i);
    A=Pb;
    delay(6000);
    string_to_lcd(1,15,"1");
    string_to_lcd(1,14,"#");
    delay(6000);
}
if(j==1)
{
    start(i);
    B=Pb;
    delay(6000);
    string_to_lcd(1,14,"#");
    string_to_lcd(1,15,"2");
    delay(6000);
}
if(j==2)
{
    start(i);
    C=Pb;
    delay(6000);
    string_to_lcd(1,14,"#");
    string_to_lcd(1,15,"3");
    delay(6000);
}
if(j==3)
{
```

```
start(i);
D=Pb;
delay(6000);
string_to_lcd(1,14,"#");
string_to_lcd(1,15,"4");
delay(6000);
}
if(j==4)
{
start(i);
E=Pb;
delay(6000);
string_to_lcd(1,14,"#");
string_to_lcd(1,15,"5");
delay(6000);
}
}
M=(A+B+C+D+E)/0x05;
Pc=M;
delay(60000);
delay(60000);
delay(60000);
delay(60000);
Pc=0x00;

return(M);
}
```

```
int delivitive(float mmm)
```

```
{
```

```
xdata float mean;  
mean=mmm;  
Pc=mean;
```

```
    if(mean<0x41)  
    {  
        return(0);  
    }  
    if( mean>0x41 && mean<0xc0 )  
    {  
        return(0);  
    }  
    if(mean>0xc0)  
    {  
        return(1);  
    }  
}
```

```
void out_valve (int i,int x)  
{
```

```
    switch(i)  
    {  
        case(1):  
        {  
            if(x==ON)  
            {
```

```
P1_0 = 0; ##### valve1 "ON"
Pc=0x01;
// printf("valve1 ON");
init_lcd();
    string_to_lcd(2,1,"Lower50% Val ON" );
delay(100);
delay(10000);
delay(10000);
delay(10000);
delay(10000);
cal_temp();
delay(10000);
delay(10000);
delay(10000);
delay(10000);
delay(10000);
}
else
{
P1_0 = 1; ##### valve1 "OFF"
Pc=0x00;
init_lcd();
    string_to_lcd(2,1," valve1 Off" );
delay(10000);
delay(10000);
delay(10000);
delay(10000);
cal_temp();
delay(10000);
delay(10000);
```

```
        delay(10000);
        delay(10000);
    }
}break;
case(2):
{
    if(x==ON)
    {
        P1_1 = 0; ##### valve2 "ON"
        Pc=0x02;
//        printf("valve2 ON");
        string_to_lcd(2,1," Lower50%");
        delay(100);
        delay(10000);
        cal_temp();
        delay(10000);
    }
else
    {
        P1_1 = 1; ##### valve "OFF"
        Pc=0x00;
        string_to_lcd(2,1," Over70%");
        delay(10000);
        cal_temp();
        delay(10000);
    }
}break;
```

```
case(3):
{
if(x==ON)
{
P1_2 = 0; ##### valve "ON"
Pc=0x04;
// printf("valve3 ON");
string_to_lcd(2,1," Lower50%");
delay(10000);
cal_temp();
delay(10000);
}
else
{
P1_2 = 1; ##### valve "OFF"
Pc=0x00;
string_to_lcd(2,1," Over70%");
delay(10000);
cal_temp();
delay(10000);
}
}break;
case(4):
{
if(x==ON)
{
P1_3 = 0; ##### valve "ON"
Pc=0x08;
```

```
// printf("valve4 ON");
    string_to_lcd(2,1," Lower50%");
    delay(10000);
    cal_temp();
    delay(10000);

}

else
{
    P1_3 = 1; ##### valve "OFF"
    Pc=0x00;
    string_to_lcd(2,1," Over70%");
    delay(10000);
    cal_temp();
    delay(10000);
}
}break;
case(5):
{
    if(x==ON)
    {
        P1_4 = 0; ##### valve "ON"
        Pc=0x10;
// printf("valve5 ON");
        string_to_lcd(2,1," Lower50%");
        delay(10000);
        cal_temp();
        delay(10000);
```

```
    }  
else  
    {  
        P1_4 = 1; ##### valve "OFF"  
        Pc=0x00;  
        string_to_lcd(2,1," Over70%");  
        delay(10000);  
        cal_temp();  
        delay(10000);  
    }  
}break;  
case(6):  
    {  
        if(x==ON)  
        {  
            P1_5 = 0; ##### valve "ON"  
            Pc=0x20;  
            // printf("valve6 ON");  
            string_to_lcd(2,1," Lower50%");  
            delay(10000);  
            cal_temp();  
            delay(10000);  
        }  
    }  
else  
    {  
        P1_5 = 1; ##### valve "OFF"  
        Pc=0x00;  
        string_to_lcd(2,1," Over70%");
```



```
    delay(10000);  
    cal_temp();  
    delay(10000);  
  }  
  }break;  
}
```



```
main(void)  
{  
  xdata float M,mean;  
  xdata int status;  
  unsigned char mBUF,m;  
  unsigned char mess[10];  
  
  delay(20000);  
  
  delay(100);  
  Pcon=0x82;  
  delay(100);
```

```
P1_0 =1;
P1_1 =1;
P1_2 =1;
P1_3 =1;
P1_4 =1;
P1_5 =1;
P1_6 =1;
P1_7 =1;

Pc=0xff;
init_lcd();
    delay(10000);
    delay(10000);
init_serial();
    cal_temp();
    delay(10000);
    delay(10000);

m=0;
while (1)
{
    while (!RI)
    {

    }

    RI = 0;
    mBUF=SBUF;
    mess[0]=mBUF;
    Pa=mBUF;
        delay (10000);
```

```
string_to_lcd(2,1,"Open port");  
delay(10000);  
delay(10000);  
delay(10000);  
string_to_lcd(2,15,mess);  
printf(mess);  
delay(3000);
```

```
if(mess[0]=='A')  
{  
    string_to_lcd(2,2,"Check Valve1");  
    delay(1000);  
    delay(1000);  
    delay(1000);  
    delay(1000);  
    i=1;  
    mean=get_value(M);  
    Pc=mean;  
    delay(10000);  
    delay(10000);  
    delay(10000);  
    delay(10000);  
    delay(10000);  
    delay(10000);  
    delay(10000);  
    status=delivative(mean);  
    out_valve(i,status);  
}
```

```
if(mess[0]=='B')
{
    string_to_lcd(2,1,"Check Valve2");
    delay(1000);
    delay(1000);
    delay(1000);
    delay(1000);
    i=2;

    mean=get_value(M);
    Pc=mean;
    delay(10000);
    delay(10000);
    delay(10000);
    delay(10000);
    delay(10000);
    delay(10000);
    delay(10000);
    delay(10000);
    status=delivative(mean);
    out_valve(i,status);
}

if(mess[0]=='C')
{
    string_to_lcd(2,1,"Check Valve3");
    delay(1000);
    delay(1000);
    delay(1000);
    delay(1000);
    i=3;
```



```
status=delivitive(mean);
out_valve(i,status);

}

if(mess[0]=='E')
{
string_to_lcd(2,1,"Check Valve5");
delay(1000);
delay(1000);
delay(1000);
delay(1000);
i=5;
mean=get_value(M);
Pc=mean;
delay(10000);
delay(10000);
delay(10000);
delay(10000);
delay(10000);
delay(10000);
delay(10000);
delay(10000);
status=delivitive(mean);
out_valve(i,status);

}

if(mess[0]=='F')
{
string_to_lcd(2,1,"Check Valve6");
delay(1000);
delay(1000);
```

```
delay(1000);  
delay(1000);  
i=6;  
mean=get_value(M);  
Pc=mean;  
delay(10000);  
delay(10000);  
delay(10000);  
delay(10000);  
delay(10000);  
delay(10000);  
delay(10000);  
status=delivative(mean);  
out_valve(i,status);  
}  
}  
}
```

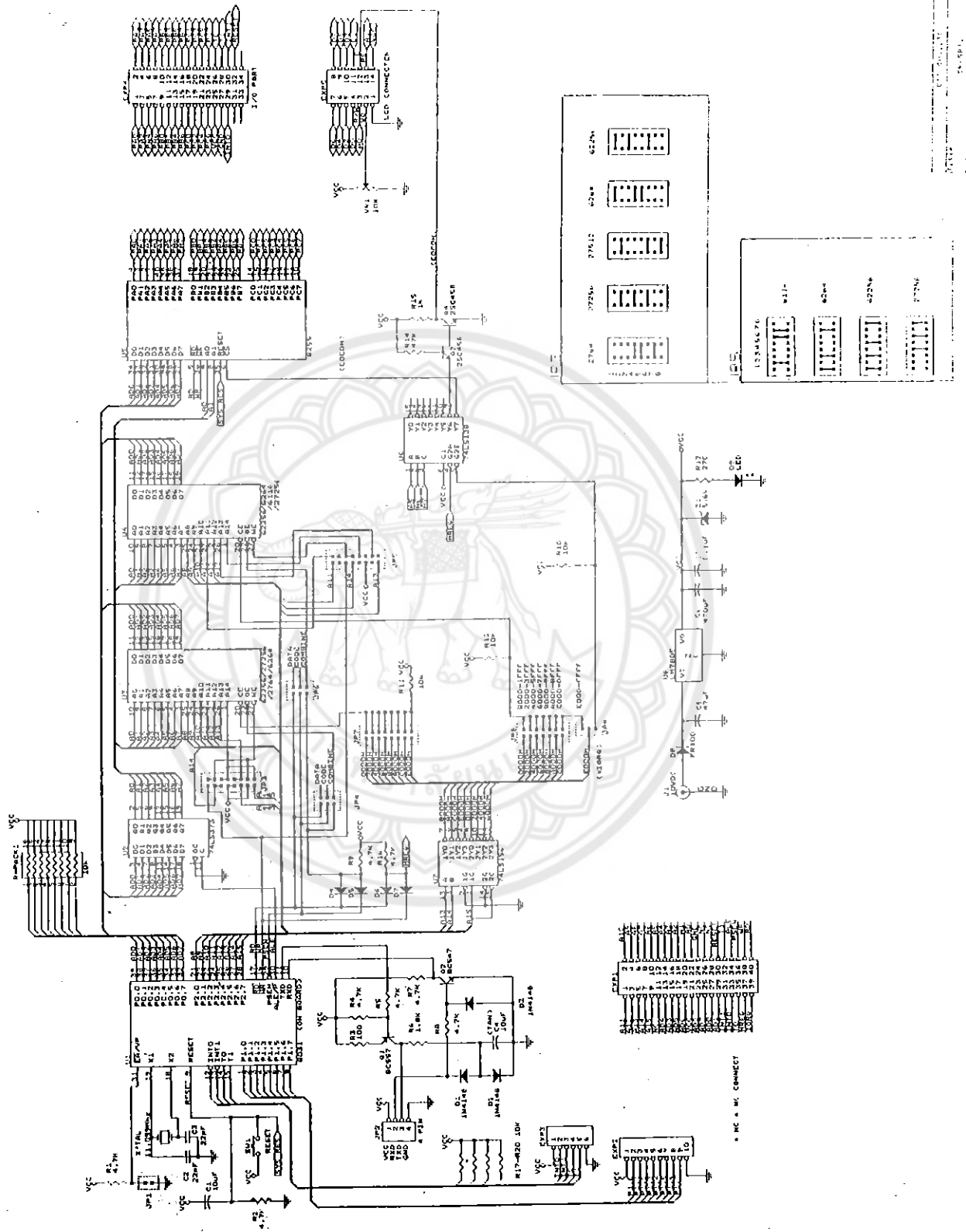




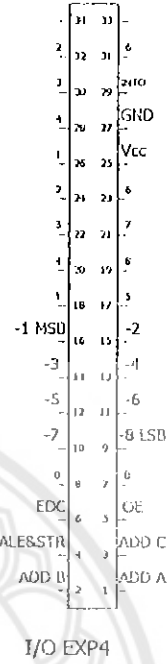
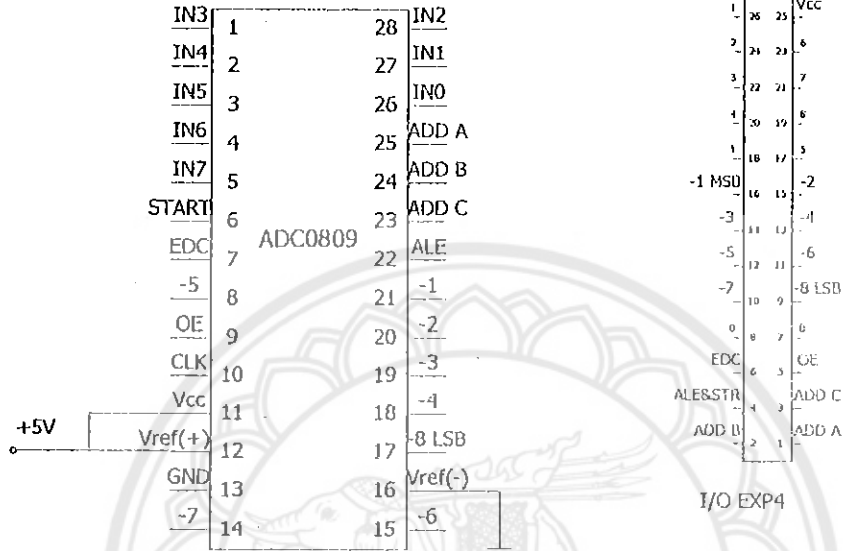
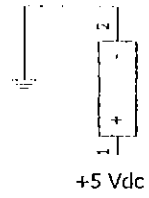
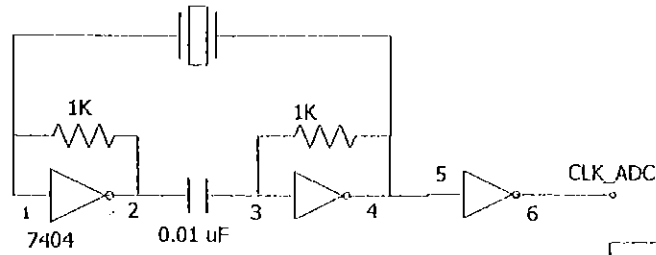
ภาคผนวก ข.

รูปวงจรของไมโครคอนโทรลเลอร์บอร์ด 8032

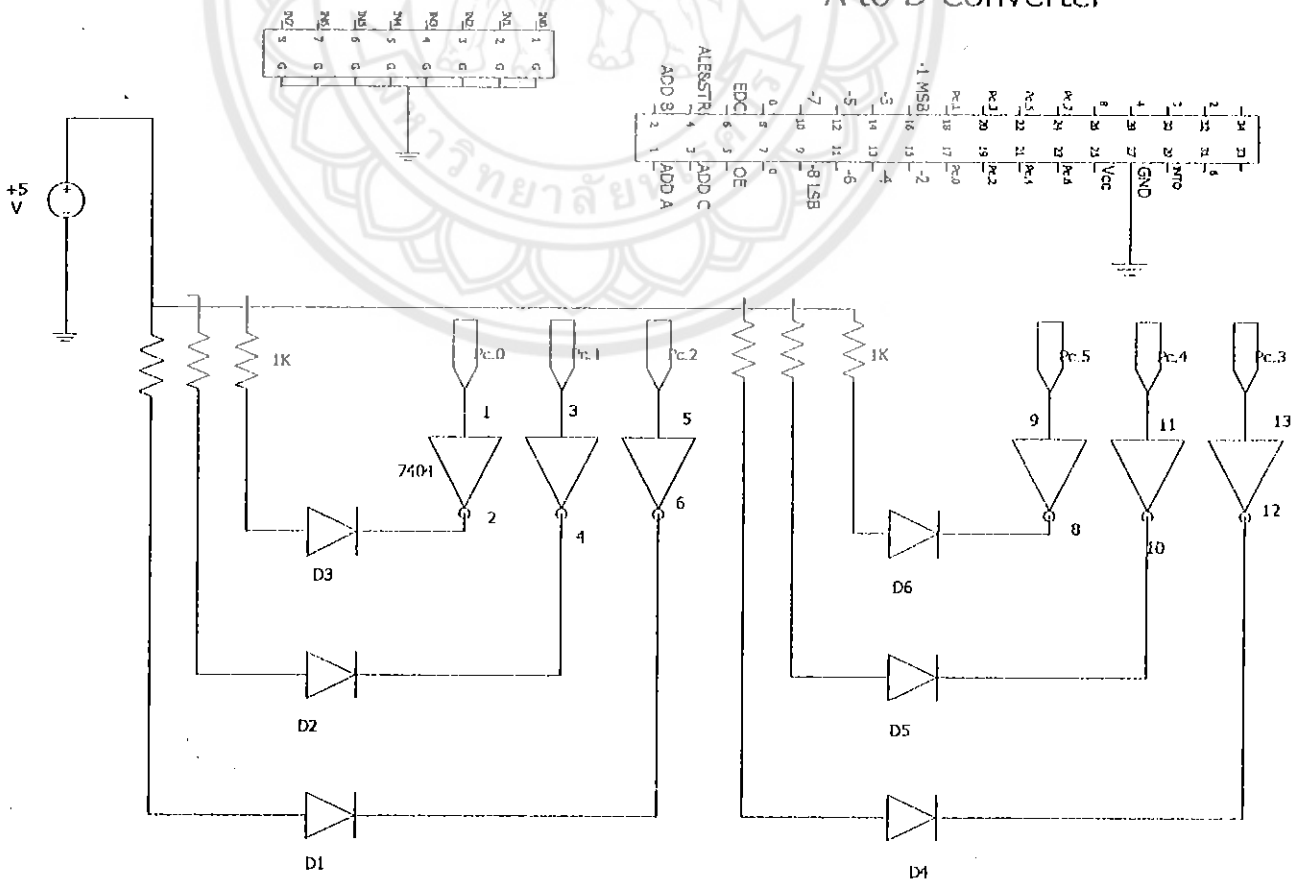
รูปวงจรส่วนแปลงขนาดอกเป็นดิจิตอล และ ภาคแสดงผล



รูปที่ ข-1 วงจรบอร์ดไมโครคอนโทรลเลอร์ 8032



A to D Converter



Display panel

ภาคผนวก ก.

Data sheet

ADC 0809



ADC0808/ADC0809

8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

Features

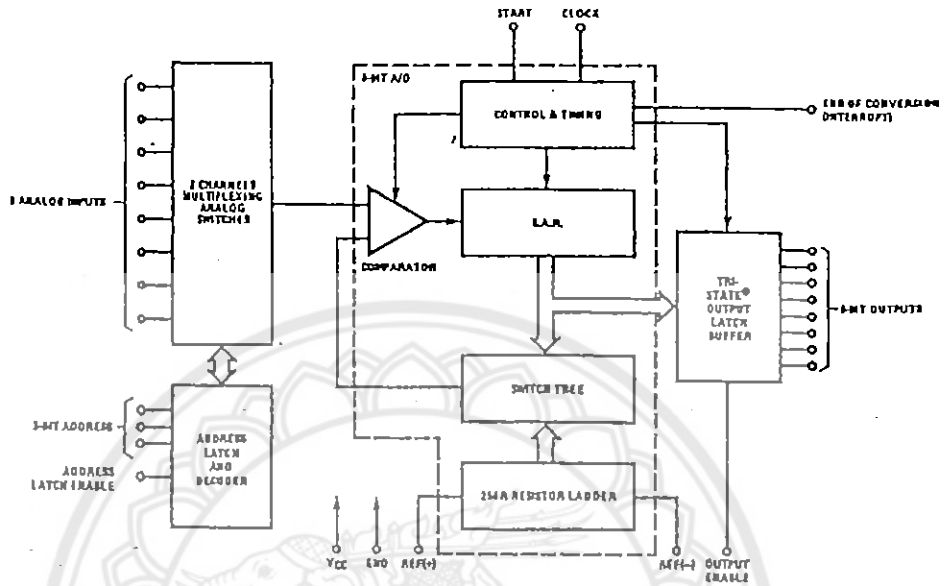
- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{OC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

- Resolution: 8 Bits
- Total Unadjusted Error: $\pm 1/2$ LSB and ± 1 LSB
- Single Supply: 5 V_{OC}
- Low Power: 15 mW
- Conversion Time: 100 μ s

TRI-STATE® is a registered trademark of National Semiconductor Corp.

Block Diagram



05005472-1

See Ordering Information

Absolute Maximum Ratings (Notes 1, 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V _{CC}) (Note 3)	6.5V
Voltage at Any Pin Except Control Inputs	-0.3V to (V _{CC} +0.3V)
Voltage at Control Inputs (START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	-0.3V to +15V
Storage Temperature Range	-65°C to +150°C
Package Dissipation at T _A =25°C	875 mW
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C

Dual-In-Line Package (ceramic)	300°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 8)	400V

Operating Conditions (Notes 1, 2)

Temperature Range (Note 1)	T _{MIN} ≤ T _A ≤ T _{MAX}
ADC0808CJ	-55°C ≤ T _A ≤ +125°C
ADC0808CCJ, ADC0808CCN, ADC0809CCN	-40°C ≤ T _A ≤ +85°C
ADC0808CCV, ADC0809CCV	-40°C ≤ T _A ≤ +85°C
Range of V _{CC} (Note 1)	4.5 V _{DC} to 6.0 V _{DC}

Electrical Characteristics

Converter Specifications: V_{CC}=5 V_{DC}=V_{REF+}, V_{REF(-)}=GND, T_{MIN} ≤ T_A ≤ T_{MAX} and I_{CLK}=640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808					
	Total Unadjusted Error (Note 5)	25°C T _{MIN} to T _{MAX}			±½ ±¾	LSB LSB
	ADC0809					
	Total Unadjusted Error (Note 5)	0°C to 70°C T _{MIN} to T _{MAX}			±1 ±1¼	LSB LSB
	Input Resistance	From Ref(+) to Ref(-)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND-0.10		V _{CC} +0.10	V _{CC}
V _{REF(+)}	Voltage, Top of Ladder	Measured at Ref(+)		V _{CC}	V _{CC} +0.1	V
$\frac{V_{REF(+)} + V_{REF(-)}}{2}$	Voltage, Center of Ladder		V _{CC} /2-0.1	V _{CC} /2	V _{CC} /2+0.1	V
V _{REF(-)}	Voltage, Bottom of Ladder	Measured at Ref(-)	-0.1	0		V
I _{IN}	Comparator Input Current	I _C =640 kHz, (Note 6)	-2	±0.5	2	µA

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CJ 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_A ≤ +125°C unless otherwise noted
ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75V ≤ V_{CC} ≤ 5.25V, -40°C ≤ T_A ≤ +85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
ANALOG MULTIPLEXER						
I _{OFF(+)}	OFF Channel Leakage Current	V _{CC} =5V, V _{IN} =5V, T _A =25°C T _{MIN} to T _{MAX}		10	200 1.0	nA µA
I _{OFF(-)}	OFF Channel Leakage Current	V _{CC} =5V, V _{IN} =0, T _A =25°C T _{MIN} to T _{MAX}	-200 -1.0	-10		nA µA
CONTROL INPUTS						
V _{IN(1)}	Logical "1" Input Voltage		V _{CC} -1.5			V
V _{IN(0)}	Logical "0" Input Voltage				1.5	V
I _{IN(1)}	Logical "1" Input Current (The Control Inputs)	V _{IN} =15V			1.0	µA
I _{IN(0)}	Logical "0" Input Current (The Control Inputs)	V _{IN} =0	-1.0			µA
I _{CC}	Supply Current	I _{CLK} =640 kHz		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CJ 4.5V $V_{CC} \leq 5.5V$, $-55^{\circ}C \leq T_A \leq 125^{\circ}C$ unless otherwise noted
 ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75V $V_{CC} \leq 5.25V$, $-40^{\circ}C \leq T_A \leq 85^{\circ}C$ unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = -360 \mu A$	$V_{CC} - 0.4$			V
$V_{OUT(0)}$	Logical "0" Output Voltage	$I_O = 1.6 mA$			0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	$I_O = 1.2 mA$			0.45	V
I_{OUT}	TRI-STATE Output Current	$V_O = 5V$ $V_O = 0$	-3		3	μA μA

Electrical Characteristics

Timing Specifications $V_{CC} = V_{REF(+)} = 5V$, $V_{REF(-)} = GND$, $t_r = t_f = 20 ns$ and $T_A = 25^{\circ}C$ unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t_s	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t_H	Minimum Address Hold Time	(Figure 5)		25	50	ns
t_D	Analog MUX Delay Time From ALE	$R_B = 0\Omega$ (Figure 5)		1	2.5	μS
t_{HI}, t_{LO}	OE Control to Q Logic State	$C_L = 50 pF$, $R_L = 10k$ (Figure 6)		125	250	ns
t_{HI}, t_{OH}	OE Control to HI-Z	$C_L = 10 pF$, $R_L = 10k$ (Figure 6)		125	250	ns
t_c	Conversion Time	$f_c = 640 kHz$, (Figure 5) (Note 7)	90	100	116	μS^{-1}
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(Figure 5)	0		8+2	μS Clock Periods
C_{IN}	Input Capacitance	All Control Inputs		10	15	pF
C_{OUT}	TRI-STATE Output Capacitance	All TRI-STATE Outputs		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{CC} .

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0V_{DG} to 5V_{DG} input voltage range will therefore require a minimum supply voltage of 4.900 V_{CC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See Figure 2. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example; 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See Figure 13.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (Figure "NO TGT: Fig NS0592"). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. *Table 1* shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (*Figure 1*) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached $+1/2$ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n -iterations are required for an n -bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

Functional Description (Continued)

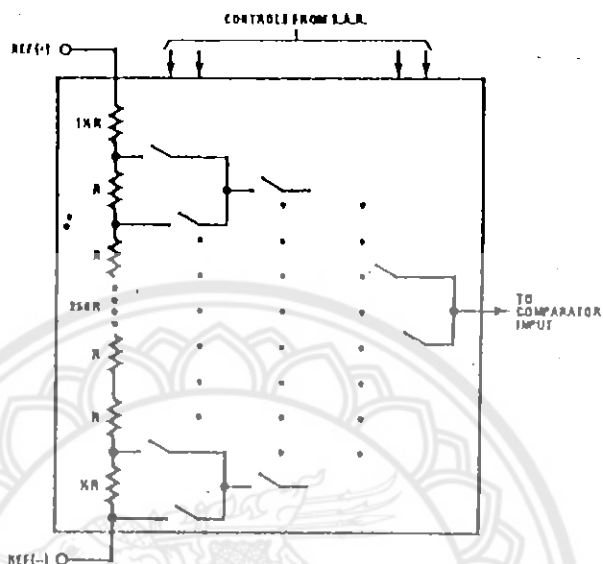


FIGURE 1. Resistor Ladder and Switch Tree.

D5006472-8

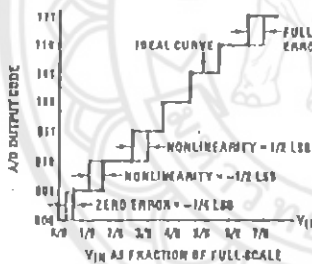


FIGURE 2. 3-Bit A/D Transfer Curve

D5006472-13

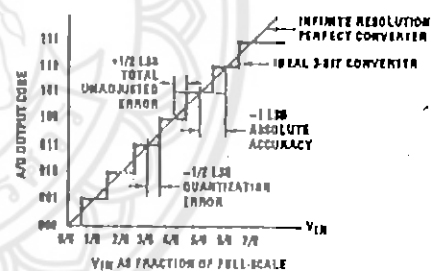


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

D5006472-14

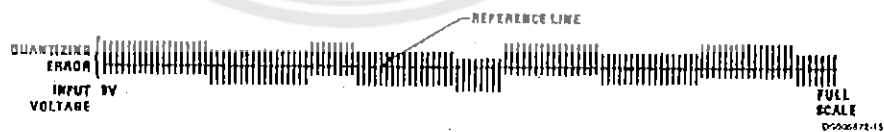
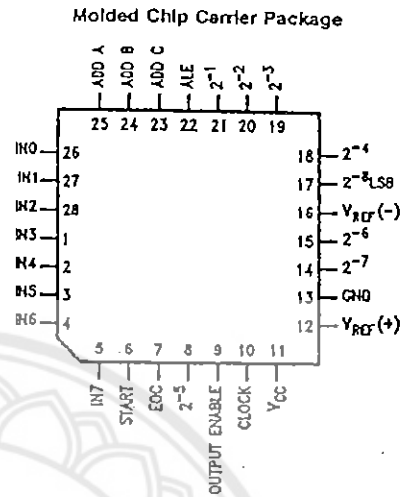
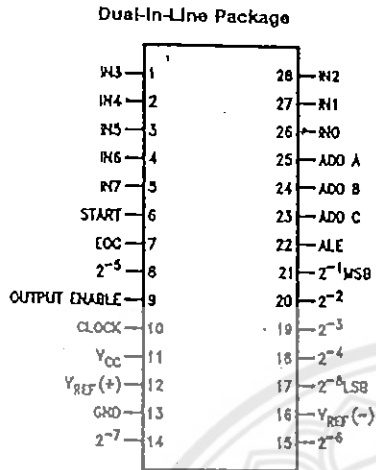


FIGURE 4. Typical Error Curve

D5006472-15

Connection Diagrams



Order Number ADC0808CCN, ADC0809CCN,
ADC0808CCJ or ADC0808CJ
See NS Package J28A or N28A

Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Timing Diagram

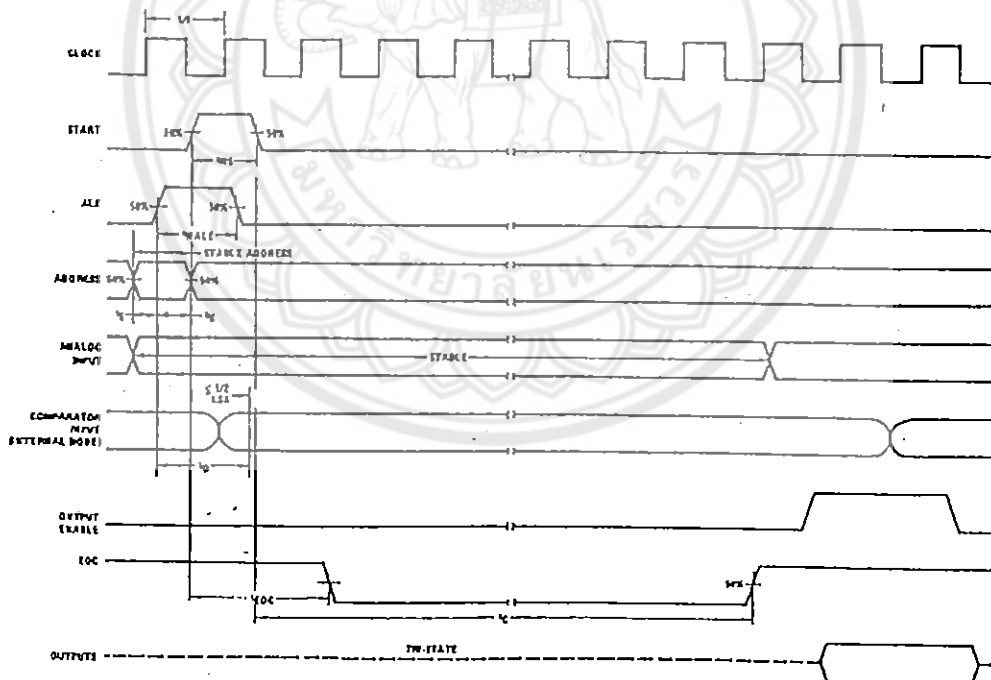


FIGURE 5.

DS005472-4

ภาคผนวก ง.

ข้อมูลที่ใช้ในการทำโปรแกรมควบคุมวาล์วและดูแลรักษาพืช



ภาคผนวก ง.

การเก็บข้อมูลดิบของการเจริญเติบโตของกะน้ำ

จากการเก็บข้อมูลซึ่งจะเก็บตามสถานที่ต่าง ๆ 3 ที่ เพื่อหาข้อมูลดิบทั้งขั้นตอนการปลูก
กะน้ำและอัตราการเจริญเติบโตได้ดังนี้

1. สถานที่ ไร่พี่ระพร จงจิตตานนท์

ขั้นตอนการปลูกกะน้ำ

1. การเตรียมดินและแปลงปลูก

1.1) เตรียมดิน 1 อาทิตย์

1.2) ใช้รถไถใหญ่ ตามด้วยรถไถเล็กเกรด, ถีดยา (100 c.c./20 ลิตร), จากนั้นเอาฟางคลุมและ
รดน้ำ

1.3) ถอนหญ้า (ไม่ถีดยาหลังจากปลูก)

2. วิธีปลูก

2.1) หว่านเมล็ด 4 กระป๋องต่อไร่

2.2) รดน้ำเข้าเย็น (ตั้งเกิดดิน)

3. การปฏิบัติดูแล

3.1) ก่อนแยกพืชจะหว่านปุ๋ย (15-15-15) ทุก 15 วัน โดยจะใช้ในอัตราครั้งละ 1 ลูกครึ่งต่อไร่
(1 ลูก = 50 ก.ก. และราคา 580 บาทต่อลูก)

3.2) ใบที่ 3 จะถีดยาฆ่าหมีด

3.3) ถูฝุ่น ถ้ามีโรคเชื้อรา ก็ทำดังนี้

3.3.1) 15-20 วัน แยกตอนออก (10-12 ชม.)

3.3.2) จากนั้นก็รดน้ำก่อนถีดยา โดยถีด 5-7 วันหนึ่งครั้ง

3.3.3) ก่อนตัดขายหว่านปุ๋ย 7 วัน

4. การเก็บเกี่ยว

4.1) ระยะที่ตัด ประมาณ 2 นิ้วจากพื้น (บน 2 ใบ)

4.2) ผลผลิต มากที่สุด 25,000 ก.ก.ต่อไร่ หน้าแล้ง 4,000 ก.ก.ต่อไร่

5. รายได้ต่อหนึ่งไร่

ขายได้ประมาณ 7000 บาทต่อไร่

6. ข้อสำคัญที่ต้องคำนึงถึง

หน้าหนาวจะมีแมลงรบกวนน้อย และปลายฝนต้นหนาวจะน้ำจะงามที่สุด

2. สถานที่ ไร่หลวง 225 หมู่ 4 ต.บึงพระ อ.เมือง จ.พิษณุโลก

ขั้นตอนการปลูกคะน้า

1. การเตรียมดินและแปลงปลูก

1.1) ไถคราดดินด้วยรถไถ โดยใช้รถไถใหญ่ไถหนึ่งรอบ แล้วตามด้วยรถไถเล็กคราด

1.2) ฝึดยา 400 c.c. ต่อไร่

1.3) ใช้ฟางคลุม

2. วิธีปลูก

2.1) หว่านด้วยเมล็ดกระป๋อง ถ้าหน้าฝนหว่าน 5 กระป๋อง หน้าแล้งหว่าน 4 กระป๋อง (ไม่ต้องไถกลบ)

2.2) 3 คืบแตกเมล็ด

3. การปฏิบัติดูแล

หว่านปุ๋ยประมาณ 7 ครั้งต่อการปลูกแต่ละรอบ (ตอนเย็น) 10 วันต่อครั้ง

3. สถานที่ ไร่สมพร ต่างจันทร์ 150/7 หมู่ 5 ต.บึงพระ อ.เมือง จ.พิษณุโลก

จำนวน 2 ไร่

ขั้นตอนการปลูกคะน้า

1. การเตรียมดินและแปลงปลูก

1.1) ไถ

1.2) กว้าง 3 สอก (แปลง)

2. วิธีปลูก

หว่านเมล็ด 7 กระป๋องต่อไร่

3. การปฏิบัติดูแล

3.1) 2 สัปดาห์ ใส่ปุ๋ย ฝึดยา หลังจากนั้น ฝึดยา 1 สัปดาห์ต่อครั้ง

3.2) ใส่ปุ๋ยอาหารคัณฑ์หนึ่งครั้ง ใช้สูตร 15-15

3.3) รดน้ำ 2 ครั้ง ตอนกลางวันและเย็น

3.4) ถ้าหากมีแมลง ให้ทำดังนี้ ใช้น้ำผสมยาแอมแพค, นีโต้ ฝึดยา ไร่ละ 3 แกลลอน

4. การเก็บเกี่ยว

ตัดขายได้เมื่อมีอายุครบ 45-50 วัน

5. ผลผลิตและรายได้เฉลี่ย

ได้ผลผลิตประมาณ 2,500 ก.ก. ต่อไร่ และตัดยกแปลง ขายได้กิโลกรัมละ 5 บาท (มากที่สุด 12 บาท)

6. ข้อสำคัญที่ต้องคำนึงถึง

หน้าแล้งแมลงเยอะ หน้าแล้งน้ำหนักดี

จากการเก็บข้อมูลสามารถรวบรวมและสรุปเป็นตารางวัดการเจริญเติบโตได้ดังนี้

ตารางที่ ง-1 วัดการเจริญเติบโตของคะน้า

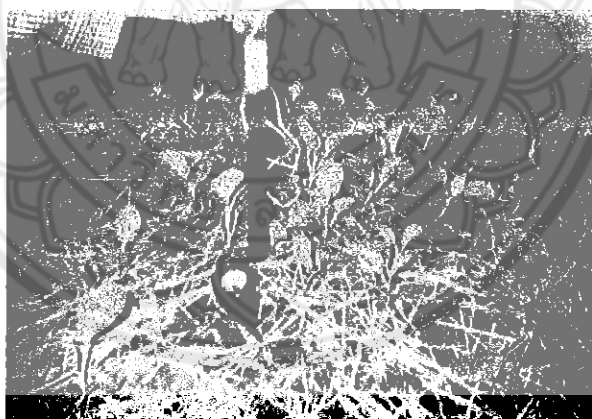
สัปดาห์ ที่	ความยาวรอบลำต้น (ซ.ม.)					ความสูงจากโคนถึงยอด (ซ.ม.)					จำนวนใบ (ใบ)				
	ต้นที่					ต้นที่					ต้นที่				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	ทุกต้นลำต้นเล็ก					1.0	1.5	1.0	0.5	1.0	2	2	2	2	2
2	ทุกต้นลำต้นเล็ก					2.0	2.5	3.0	2.0	3.0	2.5	2	3	2	3
3	2.0	2.5	2.5	2.5	2.0	14	12	10	14	13	6	8	7	6	7
4	4.5	3.5	3.0	4.5	4.0	23	20	19	25	20	5	5	5	6	5
5	5.8	4.7	5.1	4.9	5.5	24	27	25	23	22	6	7	5	7	6
6	7	5.5	5.0	5.5	5.5	30	25	25	27	29	7	5	6	6	7



รูปที่ ง-1 ลักษณะแปลงปลูกสัปดาห์ที่ 1



รูปที่ ๙-2 สัปดาห์ที่ 2



รูปที่ ๙-3 การเก็บข้อมูล ความชื้นและความสูงในสัปดาห์ที่ 3



รูปที่ ง-4 การวัดความสูงในสัปดาห์ที่ 4



รูปที่ ง-5 การเจริญเติบโตในสัปดาห์ที่ 5



รูปที่ ง-6 ลักษณะของคะน้ำในสัปดาห์ที่ 6

ข้อมูลจากหนังสือ

คะน้าจีน Chinese Kale

ถิ่นกำเนิด เอเชียไมเนอร์

ประเภทผัก อายุหลายปี แต่ปลูกเป็นผักอายุปีเดียว

อายุปลูก ตั้งแต่หว่านหรือหยอดเมล็ดจนถึงเก็บเกี่ยว 45-55 วัน

ขนาด ต้นสูงประมาณ 35-50 ซม.

ผลผลิต บ้านเราเกษตรกรปลูกได้ประมาณ 950-2,000 กก./ไร่ เฉลี่ย 1,150 กก./ไร่

ฤดูปลูก ปลูกได้ตลอดทั้งปี แต่ปลูกได้ผลดีที่สุด ในช่วงเดือน ต.ค.-เม.ย.

คะน้าจีนเป็นผักที่เราปลูกเพื่อบริโภคส่วนของใบและลำต้น เป็นผักที่มีความสำคัญและนิยมบริโภคมากในบ้านเราและหลายประเทศในทวีปเอเชีย เช่น จีน ไต้หวันฮ่องกง มาเลเซีย เป็นต้น

สภาพแวดล้อมที่ต้องการ

ประเภทดิน แทะทุกชนิดที่มีความอุดมสมบูรณ์สูง

พีเอชดิน ช่วงที่พอเหมาะ 5.5-6.8

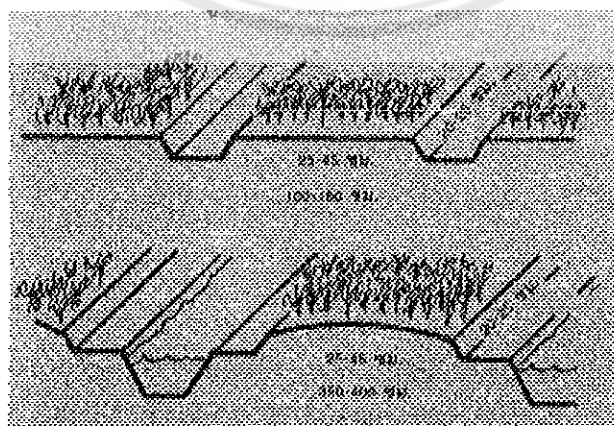
ความชื้น สูงและสม่ำเสมอ

แสง แสงแดดเต็มที่

อุณหภูมิ ช่วงที่เหมาะสม 20-25 เซลเซียส

การเตรียมดินและแปลงปลูก

จะสามารถเตรียมได้ดังรูปที่ ง-7



รูปที่ ง-7 ลักษณะของการจัดการแปลงปลูก

ระยะปลูก

ต้นเดียวต่อหลุมหรือจุด ระหว่างต้น 20-25 ซม. ระหว่างแถว 24-25 ซม.

จำนวนต้นอยู่ระหว่าง 17,000-25,000 ต้น/ไร่ ดังรูป

การขยายพันธุ์



เมล็ด: สี น้ำตาล

ลักษณะ กลม ผิวเรียบ

จำนวน 1 กรัม มี 200-300 เมล็ด

รูปที่ จ-8 ลักษณะของเมล็ดพันธุ์คะน้ำ

จำนวนเมล็ดที่ใช้หว่านโดยตรง ในแปลงปลูก 1 ไร่ ประมาณ 1-2 กก./ไร่ หรือ 1.2-2.5 ลิตร/ไร่ ขึ้นกับฝีมือการหว่าน และสำหรับแปลงเพาะกล้าใช้ประมาณ 300-320 กรัม จะได้ต้นกล้าเพียงพอสำหรับพื้นที่ปลูก 1 ไร่ ในพื้นที่ประมาณ 5-7 ตารางเมตร และใช้หยอดหลุมหรือโรยเป็นแถวปลูกโดยตรงในแปลงประมาณ 350-800 กรัม/ไร่

วิธีปลูก

นิยมปลูกแบบหว่านกระจายเมล็ดโดยตรงในแปลงมากกว่าย้ายกล้า เพราะต้นกล้าหรือต้นคะน้ำขนาดเล็กที่ถอนแยกออกสามารถจำหน่ายได้เป็นคะน้ำยอด และเป็นที่ต้องการของตลาดสูง.

การปฏิบัติดูแล

การให้ปุ๋ย ปุ๋ยรองพื้น (กก./ไร่) N= 6.4-12, P= 6.4-12, K=6.4-12 และปุ๋ยในโตรเจนเสริมโรยหน้าหรือโรยข้าง 2 ครั้ง ๆ ละ 4-5 กก./ไร่ เมื่อต้นอายุ 15-20 วัน และ 30 วัน หรือหลังการถอนแยกต้นอ่อนแต่ละครั้ง

การให้น้ำ

คะน้ำควรได้รับน้ำอย่างเพียงพอและสม่ำเสมอ เนื่องจากการเจริญเติบโตอย่างรวดเร็ว ปล่อยให้ต้นขาดน้ำ จะทำให้ชะงักการเจริญเติบโตและคุณภาพไม่ดี

การพรวนดินกำจัดวัชพืช ควรปฏิบัติในระยะแรกพร้อม ๆ กับการถอนแยก

โรคสำคัญ โรคโคนน้ำ โรคราน้ำค้าง โรคเหี่ยว

แมลงสำคัญ หนอนใยผัก ค้างหมัดกระโดด หนอนคืบกะหล่ำ เพลี้ยอ่อน

การเก็บเกี่ยว

อายุการเก็บเกี่ยวของคะน้าพันธุ์ที่นิยมปลูกในบ้านเรา ประมาณ 45-55 วัน ซึ่งเป็นคะน้าที่โตเต็มที่ บางครั้งขณะต้นเริ่มมีดอก สำหรับคะน้าต้นอ่อน หรือที่เรียกว่า ยอดคะน้า มักเก็บเกี่ยวขณะถอนแยกหรือที่อายุประมาณ 25-30 วัน

ใช้มีดตัดต้นคะน้าบริเวณโคนต้น รวบรวมนำมาตัดแต่งเอาส่วนใบแก่ที่เป็นโรคหรือถูกแมลงทำลายออก แล้วจึงบรรจุส่งตลาดต่อไป

พันธุ์

พันธุ์ที่นิยมปลูกในบ้านเรา แบ่งเป็น 3 กลุ่มใหญ่ ๆ คือ

1. พันธุ์ใบกลม ลักษณะใบกว้าง ปล้องสั้น ปลายใบมน ผิวใบเป็นคลื่นเล็กน้อย มีทั้งดอกขาว ดอกเหลือง เช่น พันธุ์ฝางเบอร์ 1
2. พันธุ์ใบแหลม ลักษณะใบแคบกว่า ปลายใบแหลม ขั้วห่าง ใบผิวเรียบ ดอกขาว เช่น พันธุ์พีแอล 20
3. พันธุ์ก้าน ลักษณะใบเหมือนคะน้าแหลม จำนวนใบต่อต้นน้อยกว่า ปล้องยาวกว่า มีดอกขาว เช่น พันธุ์แม่โจ้ 1

ตารางที่ ง-2 ชนิดพันธุ์คะน้าในประเทศไทย

ชื่อพันธุ์	ลักษณะประจำพันธุ์	ที่ปลูก	ฤดูปลูก	ผู้จำหน่าย
1. คะน้าใบ	พันธุ์ผสมเปิด ลำต้นใหญ่ ขั้วสั้น ปล้องถี่ ก้านใบเล็ก ใบกลมหนา กรอบอร่อย ทนทาน ต่อสภาพดินฟ้าอากาศดี อายุเก็บเกี่ยว ประมาณ 45-50 วัน	ทุกภาค	ทุกฤดู	เจียใต้
2. คะน้ายอด	พันธุ์ผสมเปิด ลำต้นใหญ่อวบ ก้านใบยาว ปลายใบแหลม ผิวใบเรียบ ทนทานต่อโรค ความร้อนและความชื้น อายุเก็บเกี่ยว ประมาณ 30-55 วัน	ทุกภาค	ทุกฤดู	เจียใต้
3. บางบัวทอง 30	พันธุ์ผสมเปิด เป็นคะน้ายอด ลำต้นอวบ แข็งแรง ขั้วถี่พอเหมาะ ใบแหลมหนา น้ำหนักดี ทนทานต่อโรคใบด่าง ใบกรอบ	ทุกภาค	ทุกฤดู	อีแสเวสต์-ซิคส์

พันธุ์กะน้าจีนของต่างประเทศที่น่าสนใจ

ตารางที่ ง-3 พันธุ์กะน้าจีนของต่างประเทศ

ชื่อพันธุ์	ลักษณะประจำพันธุ์	ผู้ผลิต	ประเทศ
1. ฟลุไวท์ (Full White)	ผสมเปิด มีปล้องสั้น ใบหนา ต้นสูง ประมาณ 15-20 ซม. เมื่อเริ่มมีดอก ทนทานความร้อนได้ดี	Known You	ไต้หวัน
2. พอยท์ลีฟแอนด์ไวท์ฟลาวเวอร์ (Pointed leaf & White Flower)	ผสมเปิด ลำต้นหนาอ่อนนุ่ม ปล้องยาว ทนความร้อนได้ดี	Evergrow	ไต้หวัน
3. ราวลีฟแอนด์เยลโลฟลาวเวอร์ (Round leaf & Yellow Flower)	ผสมเปิด ใบกว้างใหญ่ ทนร้อนได้ดี	Evergrow	ไต้หวัน

แมลงและการป้องกันกำจัดของตระกูลครุซิเฟอรัส Cruciferae

1. หนอนคืบกะหล่ำ Cabbage Looper

พันธุ์: ไครโคพลูเซีย ใน Trichoplusia ni

ลักษณะ: ตัวหนอนระยะเล็กจะใสไม่มีสี ต่อมาค่อย ๆ มีสีเขียวเข้มขึ้น พอโตเต็มที่จะสีเขียวซีด โดยมีเส้นสีขาวพาดตามยาว โดเต็มทีขนาดยาวประมาณ 4 ซม. เวลาเคลื่อนไหว จะงอตัวและคืบตัวระหว่างขาปล้องสุดท้ายกับส่วนกลางลำตัว

ชีวจักร: ตัวผีเสื้อจะวางไข่ลักษณะกลม สีขาวนวล เป็นพองเดี่ยว ๆ นับร้อยพองบริเวณใต้ใบ ระยะไข่ประมาณ 3 วัน จะฟักเป็นตัวหนอนซึ่งมีขนาดโตเต็มที่ประมาณ 4 ซม. อายุหนอนประมาณ 14 วัน จะเข้าดักแด้ สังเกตได้จากเส้นใยขาวคลุมอยู่บาง ๆ ใต้ใบ ระยะนี้ประมาณ 7 วัน ตัวแก่เป็นผีเสื้อขนาดกลางสีเทาดำ ตรงกลางปีกคู่หน้ามีจุดสีขาวข้างละจุด อายุตัวแก่ประมาณ 7 วัน จะเริ่มวางไข่

การทำลาย: กัดกินใบ จะกินเนื้อใบขาดและมักเหลือเส้นใบไว้ ระบาดมากในฤดูหนาว

การป้องกันกำจัด:

- 1) ใช้เชื้อแบคทีเรีย แบคซิลลัส ทูรินจิงซิส 15-30 กรัมค่อน้ำ 20 ลิตร พ่นทุก 5-7 วัน ควรผสมสารจับใบเพื่อเพิ่มประสิทธิภาพ ควรพ่นในเวลาเย็นขณะที่หนอนกัดกินใบ เพื่อให้หนอนกินเชื้อและเป็นโรคตาย การใช้เชื้อแบคทีเรียมีความปลอดภัยสูงมากต่อคน สัตว์ และสิ่งแวดล้อม

แต่มีอันตรายต่อหนอนหม่อนไหม และผึ้ง จึงควรใช้ด้วยความระมัดระวังมาก ถ้ามีกิจกรรมเพาะเลี้ยงในบริเวณดังกล่าว

2) ใช้สารเคมี

- ไซเปอร์เมทริน 25% (10-20 ซซ./20 ล.) ฉีดพ่นทุก 5-7 วัน
- เพอร์เมทริน 25 % (10-20 ซซ./20 ล.) ฉีดพ่นทุก 5-7 วัน
- เคลตามาทริน 3 % (12-22 ซซ./20 ล.)

2. หนอนใยฝัก Diamond Back Moth

พันธุ์: *Plutella xylostella* (Plutellidae)

ลักษณะ: มีขนาดเล็กที่สุดในพวกหนอนศัตรูฝัก หัวท้ายแหลม และมีปุ่มยื่นออกมา 2 แฉกที่ส่วนท้าย สีเปลี่ยนตามวัยตั้งแต่สีเขียวอ่อน เทาอ่อน และเขียวปนเหลือง เมื่อถูกตัวจะคืบอย่างแรง และทิ้งตัวลงดินโดยสร้างเส้นใย

ชีพจักร: ผีเสื้อจะวางไข่ในบรื้อยฟอง เป็นกลุ่ม ๆ กลุ่มละ 4-5 ฟอง หรือวางไข่เดี่ยว ๆ ขนาดเล็ก สีเหลือง กลมแบน ระยะไข่ประมาณ 3 วัน ฝักเป็นตัวหนอนขนาดเล็กมองเห็นได้ยาก อายุหนอนประมาณ 7 วัน มีขนาดโตเต็มที่ประมาณ 1 ซม. เข้าคักแค้หุ้มด้วยเส้นใยบาง ๆ คีคอยู่ใค้ใบ อายุคักแค้ประมาณ 3-4 วัน ตัวแก่เป็นผีเสื้อสีเทา ส่วนหลังมีแถบสีเหลืองส้มลักษณะรูปหลายเหลี่ยมคล้ายเพชร อายุตัวแก่ไม่เกิน 7 วัน จะเริ่มวางไข่

การทำลาย: กัดกินผิวด้านล่างของใบจนเกิดเป็นรูพรุน ส่วนใหญ่จะเป็นใบส่วนยอดที่กำลังเจริญ ระบาดตั้งแต่ฤดูหนาว และรุนแรงที่สุดช่วงฤดูหนาวต่อฤดูแล้ง

การป้องกันกำจัด:

- 1) อย่าปล่อยให้ต้นฝักที่เก็บเกี่ยวแล้วทิ้งไว้ในแปลงเป็นแหล่งขยายพันธุ์ของแมลง ทำความสะอาดแปลง กำจัดวัชพืชที่เป็นแหล่งอาศัยให้หมด
- 2) ใช้เชื้อแบคทีเรีย แบคซิลลัส ฐรินจินซิส อัตรา 20-30 กรัมต่อน้ำ 20 ลิตร ฉีดพ่นทุก 5-7 วัน และอาจใช้ผสมกับเพอร์เมทริน 25 % เพื่อให้ยามีประสิทธิภาพดีขึ้น และปลอดภัยต่อผู้บริโภคสูงกว่าการใช้สารเคมีอื่นในปัจจุบัน
- 3) ใช้สารเคมี
 - เพอร์เมทริน 25 % (20 ซซ./20 ล.) ฉีดพ่นทุก 5-7 วัน
 - โพรไซ โอฟอส 50 % (30 ซซ./20 ล.) ทุก 5-7 วัน เมื่อระบาด.
 - ควินอิลฟอส 25% (40 ซซ./20 ล.) ทุก 5-7 วัน
 - โพรเฟโนฟอส 50% (30 ซซ./20 ล.) ทุก 5-7 วัน

3. ค้างหมัดผัก Vegetable Flea Beetle

พันธุ์: ชนิดลาย (Phyllotreta sinuata)

ชนิดสีน้ำเงิน (P.Chontalica)

ลักษณะ: ที่พบมี 2 ชนิด ชนิดแรกมีแถบสีน้ำตาลอ่อนพาด 2 แถว มีประมาณ 80% และอีกชนิดสีน้ำเงินเข้มอ่อนมีขนาดเล็กสีขาวใส โคนเต็มทีขนาดยาว 0.5 ซม. ตัวแก่ซึ่งเป็นแมลงปีกแข็งขนาดเล็กมีลำตัวยาวประมาณ 1.5 ซม. เวลาถูกกระทบกระเทือน จะกระโดดโดยอาศัยโคนขาหลัง ซึ่งมีขนาดใหญ่ ดัดตัวไปได้ไกล

ชีวจักร: ตัวคั้งวางไข่ในดินใกล้ต้นพืช ตัวอ่อนมีขนาดเล็กสีขาวใส เมื่อโตเต็มที่เข้าคักคักในดินแล้วออกเป็นตัวแก่ เป็นแมลงปีกแข็งขนาดเล็ก

การทำลาย: ตัวแก่จะกัดกินใบจนพรวน ตัวอ่อนที่เป็นตัวหนอนชอบกัดกินรากพืช โดยปกติชอบอาหารพวกผักประเภทที่มีกลิ่นฉุน มีกระบาดในพื้นที่ปลูกผักพวกนี้ซ้ำเดิม ระบาดทุกฤดูกาล โดยเฉพาะฤดูฝน

การป้องกันกำจัด:

- 1) ไถตากดินในฤดูแล้ง จะช่วยทำลายตัวอ่อนและคักคักที่อยู่ในดิน
- 2) กำจัดวัชพืชในบริเวณแปลงผักเพื่อตัดวงจรอาหารของตัวหนอน
- 3) ใช้สารเคมี
 - ไคโคร โดฟอส 50% (20-30 ชช./20 ล.)
 - คาร์บาริล (20-30 กรัม/20 ล.) ฉีดพ่นเป็นครั้งคราว
 - ไคอะซีนอน (30 ชช./20 ล.) ฉีดพ่นทุก 5 วัน

โรคและการป้องกันรักษาของตระกูลครุซีเฟอ์ Cruciferae

1. โรคเหี่ยวหรือโรคเหลือง Wilt or Yellows

สาเหตุ: เชื้อรา ฟิวซาเรียม ออกซีสปอรัม ฟ. กองกลูทิแนนส์ (Fusarium oxysporum f. conglutinans)

อาการ: ใบเปลี่ยนเป็นสีเหลือง คายและร่วงหลุดไปในที่สุด ใบแก่ที่อยู่ล่างสุดจะแสดงอาการเช่นนี้ก่อนใบอื่น ๆ เมื่อผ่าดูจะเห็นท่อลำเลียงน้ำและอาหารเป็นสีเหลืองถึงน้ำตาลดำ

สภาพที่ส่งเสริมการระบาด: ดินกรด

การป้องกันรักษา:

- 1) ใช้พันธุ์ต้านทาน
- 2) เชื้ออาศัยอยู่ในดิน ควรหลีกเลี่ยงการใช้พื้นที่นั้นอย่างน้อย 5 ปี และปลูกพืชหมุนเวียน

2. โรคราน้ำค้าง Downy Mildew

สาเหตุ: เชื้อรา เพอโรสปอรา พาราสิทิกา (*Peronospora parasitica*)

อาการ: เกิดจุดสีออกม่วง หรือสีเหลืองน้ำตาลบนใบ เมื่อพลิกดูใต้ใบจะเห็นเส้นใยราเหมือนผงขาวอยู่เต็ม ไปหมด เข้าทำลายในระยะต้นกล้าและต้นโตแล้ว ถ้าระบาดมากจะแห้งทั้งใบและตายไป

สถานที่ส่งเสริมการระบาด: ความชื้นสูง

การป้องกันรักษา:

- 1) ใช้พันธุ์ต้านทาน
- 2) คลุกเมล็ดพันธุ์ด้วยแมนโคเซบ+เมทาแลกซิล (72%) หรือเมทาแลกซิล 35% (5-10 ก./เมล็ด 1 กก.)
- 3) ใช้สารเคมี
 - แมนโคเซบ+เมทาแลกซิล 72% (20 ก./20 ล.) เริ่มฉีดพ่นหลังจากปลูก 20 วัน
 - ซิเน็บ (30 ก./20 ล.) ฉีดพ่นทุก 3-5 วัน
 - แคลบแทน 50% (30-45 ก./20 ล.) ฉีดพ่นทุก 3-5 วัน

3. โรคเน่าคอดิน Damping-off

สาเหตุ: เชื้อรา ไฟทอปทอรา (*Phytophthora sp.*), เชื้อรา พิเทียม (*Pythium sp.*), เชื้อราไรซอกโทเนีย (*Rhizoctonia sp.*)

อาการ: ต้นกล้าเกิดแผลเน่าแห้งสีน้ำตาลที่ลำต้นบริเวณระดับผิวดิน ต้นหักพับและเหี่ยวตายในเวลาไล่เลี่ยกัน บริเวณเป็นโรคจะขยายออกเป็นบริเวณกว้างเป็นวงกลม ถ้าหากเชื้อทำลายเมื่อต้นโตแล้ว อาจเข้าทำลายใบ ก้านใบ และส่วนอื่น ๆ จะทำให้ต้นชะงักการเจริญเติบโต แคระแกร็น ต้นเหี่ยวเนื่องจากรากแขนงถูกทำลาย และต้นตายในที่สุด

สภาพที่ส่งเสริมการระบาด: สภาพดินชื้นแฉะ อุณหภูมิสูง ดินเปียกกันแน่น อากาศถ่ายเทไม่ดี

การป้องกันรักษา:

- 1) ควรคลุกเมล็ดพันธุ์ด้วยสารป้องกันกำจัดรา เช่น ซิรีแซน (2%) เป็นต้น
- 2) ดินเพาะกล้า ควรรดด้วยสารแคลพแตน (0.2%) ก่อนการหยอดเมล็ดพันธุ์
- 3) ฉีดพ่นต้นด้วยสารป้องกันกำจัดรา
 - แมนโคเซบ 50% (20-30 ก./20 ล.) ฉีดพ่นให้ถูกต้นและลงดินด้วย

ประวัติผู้เขียน

ชื่อผู้เขียน นางสาวพุทธชาติ ชันตันธง
วันเดือนปีเกิด 30 ธันวาคม พ.ศ. 2521
สถานที่เกิด จังหวัดสุโขทัย
วุฒิการศึกษาระดับมัธยมศึกษา ประกาศนียบัตรระดับมัธยมศึกษาตอนปลาย
สถานการศึกษา โรงเรียนเฉลิมขวัญสตรี
ปีที่สำเร็จการศึกษา 2540

ชื่อผู้เขียน นายภูษิต หินหาด
วันเดือนปีเกิด 20 มกราคม พ.ศ. 2522
สถานที่เกิด จังหวัดเพชรบูรณ์
วุฒิการศึกษาระดับมัธยมศึกษา ประกาศนียบัตรระดับมัธยมศึกษาตอนปลาย
สถานการศึกษา โรงเรียนตะพานหิน
ปีที่สำเร็จการศึกษา 2540

ชื่อผู้เขียน นางสาวสุวิมล วสุศิริกุล
วันเดือนปีเกิด 26 สิงหาคม พ.ศ. 2516
สถานที่เกิด จังหวัดพิษณุโลก
วุฒิการศึกษาระดับมัธยมศึกษา ประกาศนียบัตรระดับมัธยมศึกษาตอนปลาย
สถานการศึกษา โรงเรียนเฉลิมขวัญสตรี
ปีที่สำเร็จการศึกษา 2534