ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ เครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

Automatic Package Scale Machine

ชื่อนักศึกษา

1. นายชลอศักดิ์	มหาประทุม	รหัสประจำตัว	44035399
2. ว่าที่ร้อยตรีธานินทร์	คุณุทัย	รหัสประจำตัว	44035405
3. นายพฤหัส	ราชาเดช	รหัสประจำตัว	44035408
4. นายภูวดล	ทับสีรัก	รหัสประจำตัว	44035412

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** เทคโนโลยีการวัดคุมทางอุตสาหกรรม

อาจารย์ที่ปรึกษา อาจารย์วรวิทย์ สมหา

อาจารย์ที่ปรึกษาร่วม อาจารย์ปิยะ ศุภวราสุวัฒน์

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์วรวิทย์	สมหา	.....
2. อาจารย์ปิยะ	ศุภวราสุวัฒน์	.....
3. อาจารย์ปิยะ	จิตธรรมมาภิรมย์	.....
4. อาจารย์อำพล	ทองระอา	.....
5. อาจารย์อมรชัย	ชัยชนะ	.....

วัน/เดือน/ปีที่สอบ วันศุกร์ที่ 8 พฤศจิกายน พ.ศ. 2545 เวลา 15.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 30 เดือน พ.ย. พ.ศ. 45

<BT4503042>

เครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

เครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

AUTOMATIC PACKAGE SCALE MACHINE

T048318

นายชลอศักดิ์ มหาประทุม
ว่าที่ร้อยตรีธานินทร์ คุณุทัย
นายพฤหัส ราชาเดช
นายภูวดล ทับสีรัก

เลขหมู่.....
เลขทะเบียน 48313
วัน, เดือน, ปี 10 ต.ค. 2546

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องชั่งอัตโนมัติ

Automatic Package Scale Machine

วัตถุประสงค์

1) เพื่อศึกษาการทำงานของ MCS-51 แมคคานิคส์ นิวแมติกส์และโหลดเซลล์
2) เพื่อออกแบบเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ
3) เพื่อสร้างและทดลองเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ
4) เพื่อนำเครื่องชั่งบรรจุภัณฑ์อัตโนมัติไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1) รู้หลักการทำงานของ MCS-51 แมคคานิคส์ นิวแมติกส์ และโหลดเซลล์
2) ได้แบบหรือผังการทำงานของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ
3) ได้เครื่องต้นแบบของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ
4) ได้ผลการทดลองการทำงานของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ
5) ได้นำเครื่องชั่งบรรจุภัณฑ์อัตโนมัติไปใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เครื่องชั่งบรรจุภัณฑ์อัตโนมัติ
นักศึกษา	นายชลอศักดิ์ มหาประทุม
	ว่าที่ร้อยตรีธานินทร์ คุณุทัย
	นายพฤหัส ราชาเดช
	นายภูวดล ทับสีรัก
อาจารย์ที่ปรึกษา	อาจารย์วรวิทย์ สมหา
อาจารย์ที่ปรึกษาร่วม	อาจารย์ปิยะ ศุภวราสุวัฒน์
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	เทคโนโลยีการวัดคุมทางอุตสาหกรรม
ปีการศึกษา	2545

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอ การนำเอาอุปกรณ์ทางอิเล็กทรอนิกส์,ไฟฟ้าและทางด้านเครื่องกลมาสร้างเป็นระบบควบคุมอัตโนมัติ ในกระบวนการชั่งผลิตภัณฑ์ทางการเกษตร โดยใช้อุปกรณ์หลักๆอันได้แก่อุปกรณ์ทางด้านนิวแมติกส์อุตสาหกรรม , ไมโครคอนโทรลเลอร์ ฯลฯ จุดประสงค์เพื่อที่จะศึกษาถึงการนำเอาระบบไมโครคอนโทรลเลอร์มาประยุกต์ เข้ากับระบบทางไฟฟ้าและทางด้านเครื่องกล เพื่อนำมาสร้างเป็นระบบการควบคุมอัตโนมัติทางอุตสาหกรรม โดยอุปกรณ์หลักที่สำคัญของระบบนี้คือไมโครคอนโทรลเลอร์ตระกูล MCS-51 มาใช้เป็นตัวควบคุมระบบดังกล่าว โดยที่ขีดความสามารถของโครงงานนี้สามารถที่จะชั่งได้ตั้งแต่ 1-25 กิโลกรัม มีความคลาดเคลื่อนประมาณ ±20 กรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Automatic Packing Scale
Students	Mr. Shalorsak Mahapratoom
	No ActingLT.Tanin Kunuthai
	Mr. Paruhut Rachadet
	Mr. Phuvadol Tabseerag
Advisor	Mr.Worawit Somha
Co-Advisor	Mr.Piya Supavarasuwat
Education Level	Bachelor of Science in Industrial Education
Program in	Industrial Instrument Technology
Academic Year	2002

ABSTRACT

This thesis presents a provide device of Electronics, Electrical and Mechanical take to build automatic control system. In presents of Product of agriculture scale by using main devide such as Pneumatic device, Microcontroller ect. The objective for studied to addition Microcontroller system to apply in Electronic system and Mechanical system for the industrial automatic system. The main purpose to used Microcontroller MCS-51 family for control system.This project is performance for to scale 1-25 kilograms.It have to error ± 20 grams.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดีเพราะความร่วมมือ ร่วมใจของสมาชิกภายในกลุ่มซึ่งทุ่มเททั้งกำลังกายกำลังใจ กำลังสมองและกำลังทรัพย์เป็นเวลานานกว่าจะถึงวันนี้ รวมทั้งได้รับความช่วยเหลือจากท่านอาจารย์วรวิทย์ สมหา อาจารย์ปิยะ ศุภวราสุวัฒน์ และอาจารย์ในภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ได้อบรมสั่งสอนให้คำปรึกษา แนะนำพร้อมทั้งอำนวยความสะดวกในการทำปริญญานิพนธ์อย่างเต็มที่และที่สำคัญหาที่เปรียบมิได้คือ บุพการีที่ให้กำเนิดอบรมเลี้ยงดูให้โอกาสทางการศึกษาและผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่คอยช่วยเหลือและให้กำลังใจในการทำปริญญานิพนธ์จนเสร็จสมบูรณ์ไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.3 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.4 ความรู้เบื้องต้นเกี่ยวกับ I^2C	5
2.4.1 คุณสมบัติโดยทั่วไปของบัส I^2C	5
2.4.2 หลักการของบัส I^2C	6
2.4.3 สภาวะที่เกิดขึ้นบนบัส I^2C	6
2.4.4 การทำงานบนบัส I^2C	7
2.4.5 อุปกรณ์ที่ใช้ในการเชื่อมต่อแบบบัส I^2C	8
2.5 การออกแบบวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล	8
2.6 ระบบควบคุมนิวแมติกส์	8
2.7 กระบอกสูบ (Cylinder)	10
2.7.1 กระบอกสูบทางเดียว (Single Acting Cylinder)	10
2.7.2 กระบอกสูบสองทาง (Double Acting Cylinder)	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.8โซลินอยด์วาล์ว	12
2.8.1โครงสร้างและหลักการทำงานของโซลินอยด์วาล์ว	12
2.9 วงจรนิวแมติกส์พื้นฐาน	15
2.9.1 วงจรควบคุมการทำงานของกระบอกสูบ	15
2.9.2 ลักษณะของการควบคุม	17
2.10 อุปกรณ์ทรานส์ดิวเซอร์ (โหลดเซลล์)	18
2.11 เครื่องชั่งอิเล็กทรอนิกส์ (Electronic Scale)	21
2.11.1 โหลดเซลล์ (Load cell)	21
2.11.2 ชุดควบคุม (Control unit)	22
2.11.3 หน่วยแสดงผล	22
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	23
3.1 กล่าวนำ	23
3.2 ขั้นตอนการทำงาน	23
3.2.1 การทำงานของแผนผังโครงงาน	23
3.2.2 การทำงานของวงจรควบคุม	25
3.3 การออกแบบเครื่องชั่งอัตโนมัติ	26
3.3.1 ส่วนควบคุมการโรยของวัตถุดิบ	26
3.3.2 ส่วนควบคุมการชั่งตวง	29
3.3.3 ส่วนบรรจุวัตถุดิบ	30
3.3.4 ส่วนโครงเหล็ก	31
3.4 การออกแบบวงจรต่างๆและการทำงานของวงจร	36
3.4.1 วงจรแหล่งจ่ายไฟ	36
3.4.2 วงจรรวมสัญญาณจากโหลดเซลล์ (Summing)	37
3.4.3 วงจรควบคุม	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 การทดลอง และผลการทดลอง	39
4.1 การทดลองหาค่าอัตราการขยายของโหลดเซลล์	39
4.2 การทดลองโหลดเซลล์	40
4.3 การทดลองการควบคุมกระบอกสูบ	41
4.4 การทดลองวงจรแสดงผล	42
4.5 การทดลองวงจรขับรีเลย์	43
4.6 การทดลองกระบอกสูบ	44
4.7 รูปแสดงเครื่องชั่งอัตโนมัติเมื่อเสร็จสมบูรณ์	45
บทที่ 5 บทสรุป ปัญหา แนวทางการแก้ไข และการพัฒนา	46
5.1 บทสรุป	46
5.2 ปัญหาและแนวทางแก้ไข	46
5.2.1 ปัญหาในการสร้างโครงเครื่องชั่งอัตโนมัติ	46
5.2.2 ปัญหาทางด้านการควบคุมการทำงาน	47
5.3 แนวทางการพัฒนา	48
ภาคผนวก ก เครื่องต้นแบบ	49
ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์	52
ภาคผนวก ค ผังงาน และโปรแกรมควบคุมการทำงาน	54
ภาคผนวก ง คู่มือการใช้งาน	88
ภาคผนวก จ รายการอุปกรณ์	95
ภาคผนวก ฉ รายละเอียด และคุณสมบัติของอุปกรณ์	98
บรรณานุกรม	133
ประวัติผู้แต่ง	134

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท Intel และ Atmel	4
ตารางที่ 4.1 การทดลองโหลดเซลล์ตัวที่หนึ่งที่น้ำหนัก 0-25 กิโลกรัม	39
ตารางที่ 4.2 การทดลองโหลดเซลล์ตัวที่สองที่น้ำหนัก 0-25 กิโลกรัม	40
ตารางที่ ง.1 ค่าความต้านทานของโหลดเซลล์แต่ละตัว	92
ตารางที่ จ.1 คาปาซิเตอร์	96
ตารางที่ จ.2 ความต้านทาน	96
ตารางที่ จ.3 อุปกรณ์อื่นๆ	97
ตารางที่ จ.4 อุปกรณ์นิวแมติกส์	97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 สภาวะปกติปิดของกระบอกสูบทางเดียว	10
รูปที่ 2.2 สภาวะการทำงานของกระบอกสูบทางเดียว	10
รูปที่ 2.3 สภาวะปกติปิดของกระบอกสูบสองทาง	11
รูปที่ 2.4 สภาวะการทำงานของกระบอกสูบสองทาง	11
รูปที่ 2.5 สภาวะปกติปิดของวาล์ว 3/2	12
รูปที่ 2.6 สภาวะการทำงานของวาล์ว 3/2	13
รูปที่ 2.7 สภาวะปกติปิดของวาล์ว 5/2	13
รูปที่ 2.8 สภาวะการทำงานของวาล์ว 5/2	14
รูปที่ 2.9 สภาวะการทำงานของวาล์ว 5/2 เมื่อมีกระแสไหลผ่านทางซ้ายมือ	14
รูปที่ 2.10 สภาวะการทำงานของวาล์ว 5/2 เมื่อมีกระแสไหลผ่านทางขวามือ	15
รูปที่ 2.11 วงจรควบคุมกระบอกสูบทางเดียว	16
รูปที่ 2.12 วงจรควบคุมการทำงานของกระบอกสูบสองทาง	16
รูปที่ 2.13 วงจรการควบคุมโดยตรง	17
รูปที่ 2.14 วงจรควบคุมแบบทางอ้อม	18
รูปที่ 2.15 โหลดเซลล์แบบคาน	19
รูปที่ 3.1 ผังการทำงานของเครื่องชั่งอัตโนมัติ	24
รูปที่ 3.2 แผนผังการทำงานของวงจรควบคุม	25
รูปที่ 3.3 แบบส่วนควบคุมการ โรยวัตถุดิบ (ด้านข้าง) มาตราส่วน 1 : 10	27
รูปที่ 3.4 แบบส่วนควบคุมการ โรยวัตถุดิบ (ด้านบน) มาตราส่วน 1 : 10	28
รูปที่ 3.5 แบบส่วนควบคุมการ โรยวัตถุดิบ (โดยรวม) มาตราส่วน 1 : 10	29
รูปที่ 3.6 แบบส่วนควบคุมการชั่งตวง (โดยรวม) มาตราส่วน 1 : 10	30
รูปที่ 3.7 แบบส่วนบรรจุวัตถุดิบ (โดยรวม) มาตราส่วน 1 : 10	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.8 แบบเหล็กแผ่นส่วนควบคุมการโรยวัตถุดิบ,ส่วนควบคุมการชั่งตวง ส่วนควบคุมการบรรจุ (ด้านบน) มาตราส่วน 1 : 10	32
รูปที่ 3.9 แบบเครื่องชั่งอัตโนมัติส่วนโครงเหล็ก (ด้านข้าง) มาตราส่วน 1 : 10	33
รูปที่ 3.10 แบบเครื่องชั่งส่วนโครงเหล็กที่ติดตั้งทั้งสามส่วน (ด้านข้าง) มาตราส่วน 1:10	34
รูปที่ 3.11 แบบโดยรวมของเครื่องชั่งอัตโนมัติ	35
รูปที่ 3.12 วงจรแหล่งจ่ายไฟ	36
รูปที่ 3.13 วงจรแหล่งจ่ายไฟให้กับโหลดเซลล์	36
รูปที่ 3.14 รูปวงจรรวมสัญญาณจากโหลดเซลล์ (Suming)	37
รูปที่ 3.15 วงจรควบคุม	38
รูปที่ 4.1 การทดลองโหลดเซลล์	40
รูปที่ 4.2 การควบคุมกระบอกสูบ	41
รูปที่ 4.3 วงจรแสดงผล	42
รูปที่ 4.4 วงจรขับรีเลย์	43
รูปที่ 4.5 กระบอกสูบส่วนควบคุมการโรยวัตถุดิบ	44
รูปที่ 4.6 กระบอกสูบส่วนควบคุมการชั่งตวง	44
รูปที่ 4.7 เครื่องชั่งอัตโนมัติเมื่อเสร็จสมบูรณ์	45
รูปที่ ก.1 ภาพรวมของแผงวงจรควบคุม	50
รูปที่ ก.2 รูปสำเร็จของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ	51
รูปที่ ข.1 วงจรควบคุม วงจรขับโซลินอยด์วาล์ว วงจรภาคแสดงผล วงจรขับรีเลย์	53
รูปที่ ค.1 แผนผังควบคุมการทำงาน	55
รูปที่ ง.1 วงจรการต่อขั้ว J2	91
รูปที่ ง.2 วงจรการต่อขั้ว J5	92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันตามโรงสีขนาดใหญ่ บริษัท และห้างร้านที่รับซื้อ-ขายข้าวเปลือกและผลผลิตทางการเกษตรจากชาวนาหรือเกษตรกร เมื่อรับซื้อมาแล้วส่วนหนึ่งก็จะส่งออกขายต่อไป อีกส่วนหนึ่งก็จะนำไปแปรรูปแล้วบรรจุเป็นผลิตภัณฑ์ เพื่อส่งขายปลีกให้กับลูกค้าและผู้ที่ต้องการซื้อในจำนวนที่ไม่มากหรือซื้อย่อย แต่ในการตวงชั่งเพื่อบรรจุนั้นส่วนมากยังใช้แรงงานคนในการตวงแล้วนำไปชั่งจนกว่าจะได้น้ำหนักที่ต้องการจึงบรรจุลงถุง ซึ่งจะทำให้เกิดความล่าช้า ไม่สะดวก และยังอาจทำให้เกิดความผิดพลาดสูง ดังนั้นการสร้างเครื่องชั่งที่สามารถปรับค่าน้ำหนักที่ต้องการได้ตั้งแต่ 1-25 กิโลกรัม มีกระบวนการชั่งที่ต่อเนื่อง จะสามารถช่วยลดแรงงานคน สะดวก รวดเร็ว และมีความเที่ยงตรงมากยิ่งขึ้น

1.2 ขีดความสามารถของโครงงาน

1) สามารถชั่งน้ำหนักได้สูงสุด 25 กิโลกรัม
2) ปรับช่วงของน้ำหนักในการชั่งได้ 1-25 กิโลกรัม
3) ชั่งน้ำหนักได้ละเอียดมีความคลาดเคลื่อน ± 20 กรัม
4) แสดงผลด้วยระบบดิจิตอล มีจุดทศนิยม 2 หลัก
5) มีกระบวนการชั่งและทำงานอย่างต่อเนื่อง

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษาและทำความเข้าใจในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 1 บทนำ ประกอบด้วย ความเป็นมาและความสำคัญของปริญญานิพนธ์ ขีดความสามารถของโครงงาน และเนื้อหาโดยสังเขป

บทที่ 2 ทฤษฎี และหลักการ ประกอบด้วยเนื้อหาทางทฤษฎีต่างๆ ที่เกี่ยวข้องในการสร้างโครงงานนี้อันได้แก่ ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์ MCS-51 ระบบนิวแมติกส์ กระบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูบลม โซลินอยด์วาล์ว อุปกรณ์โหลดเซลล์ ตัวแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล และการเชื่อมต่อแบบ I^2C

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึง การออกแบบฮาร์ดแวร์ซึ่งได้แก่ โครงสร้างและส่วนประกอบต่างๆ ของเครื่องชั่ง ภาคแสดงผลด้วยระบบดิจิตอล ภาคแสดงผลแบบ LCD ตัวแปลงสัญญาณแอนะลอกเป็นดิจิตอล และวงจรควบคุมระบบนิวแมติกส์

บทที่ 4 การทดลอง และผลการทดลอง กล่าวถึง การทดลองโหลดเซลล์ วงจรรวมสัญญาณ วงจรแสดงผล และวงจรควบคุมการทำงานของเครื่องชั่ง

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา กล่าวถึงการสรุปผลการปฏิบัติงาน ปัญหาที่เกิดขึ้นในขณะทำโครงงาน แนวทางการแก้ไขปัญหา และแนวทางในการพัฒนาโครงงานให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์

ภาคผนวก ค ผังงาน และโปรแกรมควบคุมการทำงาน

ภาคผนวก ง คู่มือการใช้งาน

ภาคผนวก จ รายการอุปกรณ์

ภาคผนวก ฉ รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

เนื้อหาของปริญญานิพนธ์ในบทนี้เป็นทฤษฎี และหลักการที่นำมาใช้ประกอบการสร้างโครงงาน โดยประกอบในเรื่องของ ไมโครคอนโทรลเลอร์ตระกูล MCS-51, กระบอกสูบลม, โซลินอยด์วาล์ว, โหลดเซลล์ ตัวแสดงผล LCD และหลักการชั่งตวง ซึ่งจะกล่าวโดยละเอียดดังต่อไปนี้

2.2 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ขึ้นอยู่กับโครงสร้างภายในของมัน บางเบอร์จะมีหน่วยความจำภายในเป็น ROM บางเบอร์เป็นแบบ EPROM บางเบอร์มี RAM ภายใน 128 ไบต์ ซึ่งแตกต่างกันออกไปแล้วแต่บริษัทผู้ผลิตเพื่อต้องการให้ผู้ใช้สามารถเลือกใช้ได้ตามความเหมาะสมของงาน อย่างไรก็ตามถึงแม้ว่าไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะถูกผลิตออกมามากมายหลายบริษัทแต่ในส่วนของสถาปัตยกรรมภายใน โครงสร้าง หน้าที่หลัก และการจัดขาต่างๆยังคงเป็นมาตรฐานเดียวกัน จะต่างกันเพียงขนาดของหน่วยความจำภายใน และหน่วยทำงานภายในเท่านั้น

2.3 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

1) เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต

2) ภายในมีหน่วยความจำเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง

3) หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบ RAM ในบางเบอร์จะมีหน่วยความจำแบบ EPROM เพิ่มเติม

4) ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต

5) มีไทเมอร์/เคาเตอร์ขนาด 16 บิต 2 ตัว

6) สามารถอินเตอร์รัพท์ได้ 5 แหล่ง

7) มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิพ

8) มีพอร์ตอนุกรมที่สามารถรับส่งข้อมูลแบบฟูลดูเพลกซ์ความเร็วสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) สามารถประมวลผลทีละบิตได้

10) อ้างหน่วยความจำโปรแกรมภายนอกได้ 64 K

11) อ้างหน่วยความจำข้อมูลภายนอกได้ 64 K

12) สามารถอ้างหน่วยความจำแบบบิตได้ 210 บิต

ในตารางที่ 2.1 แสดงถึงคุณสมบัติบางส่วนของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท Intel และบริษัท Atmel รุ่นต่างๆที่นิยมใช้กัน

ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท Intel และบริษัท Atmel

เบอร์	หน่วยความจำ โปรแกรมภายใน	หน่วยความจำ ข้อมูลภายใน	วงจรนับ/จับเวลา 16 บิต
8031	—	128 ไบต์ (RAM)	2
8032	—	256 ไบต์ (RAM)	3
8051	4 กิโลไบต์ (ROM)	128 ไบต์ (RAM)	2
8052	8 กิโลไบต์ (ROM)	256 ไบต์ (RAM)	3
8751	4 กิโลไบต์ (EPROM)	128 ไบต์ (RAM)	2
8752	8 กิโลไบต์ (EPROM)	256 ไบต์ (RAM)	3
AT89C1051	1 กิโลไบต์ (FLASH)	64 ไบต์ (RAM)	1
AT89C2051	2 กิโลไบต์ (FLASH)	128 ไบต์ (RAM)	2
AT89C51	4 กิโลไบต์ (FLASH)	128 ไบต์ (RAM)	2
AT89C52	8 กิโลไบต์ (FLASH)	256 ไบต์ (RAM)	3
AT89C55	20 กิโลไบต์ (FLASH)	256 ไบต์ (RAM)	3
AT89S8252	8 กิโลไบต์ (FLASH)	256 ไบต์ (RAM) 2 กิโลไบต์ (EEPROM)	3
AT89S53	12 กิโลไบต์ (FLASH)	256 ไบต์ (RAM)	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ความรู้เบื้องต้นเกี่ยวกับ I^2C

I^2C ย่อมาจากคำว่า Inter-IC Communication หมายถึงการติดต่อสื่อสารระหว่างไอซี โดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูล อีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงานการต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือต่อพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่แอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรมหรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock Line) ในการอธิบายต่อไปนี้จะเรียกสายสัญญาณทั้งสองว่า สาย SDA และ SCL

2.4.1 คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (Bi-Diretional Line) ต้องมีการต่อตัวต้านทานพูลอัปกับแรงดัน +5 V ไว้ตลอดเวลาเพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งานทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (Open-Drain) หรือคอเลคเตอร์เปิด (Open-Collector)

อัตราการทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต (7 – Bit Addressing) หรือ 10 บิต (10-Bit Addressing)

ข้อเด่น อีกประการหนึ่งของบัส I^2C คือสามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้โดยอุปกรณ์บนบัส I^2C ตัวหนึ่งอาจใช้ไฟเลี่ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และสองต่อตัวต้านทานพูลอัป (Rp) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 หลักการของบัส I^2C

บัส I^2C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โปรโตคอล (Protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (Transmitter) อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (Receiver) ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (Master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (Slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อว่างเท่านั้น

2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

2.4.3 สภาวะที่เกิดขึ้นบนบัส I^2C

มีด้วยกัน 5 สภาวะ ดังนี้

1) บัสว่าง (Bus Not Busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

2) เริ่มต้นการถ่ายทอดข้อมูล (Start Data Transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (Start)

3) หยุดการถ่ายทอดข้อมูล (Stop Data Transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (Stop)

4) ข้อมูลดำรงอยู่บนบัส (Data Valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น "0" หรือ "1" ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอกจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

5) รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตรับรู้ (Acknowledge Bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่อลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่อยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

2.4.4 การทำงานบนบัส I^2C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่างๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I^2C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิต ก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวให้เรียบร้อยแล้วก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้นหัวใจสำคัญอันดับแรกของการทำงานบนบัส I^2C คือการอ้างถึงอุปกรณ์แต่ละตัว ซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ

1) การอ้างถึงแบบ 7 บิต (7-Bit Addressing) ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อหรือข้อมูลกำหนดแอดเดรส หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

2) การอ้างถึงแบบ 10 บิต ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกับแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิต บนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อเช่นเดียวกับการอ้างอิงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 อุปกรณ์ที่ใช้ในการเชื่อมต่อแบบบัส I^2C

ในปัจจุบัน บัส I^2C ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ด้วยข้อดีที่ชัดเจนคือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้น และการขยายระบบไมโครคอนโทรลเลอร์ ที่มีจำนวนอินพุตเอาต์พุตและหน่วยความจำจำกัดสามารถทำได้ง่ายขึ้นด้วยระบบบัส I^2C เมื่อเป็นเช่นนี้จึงมีอุปกรณ์เพอริเฟอรัลที่ใช้ในงานกัน ดังมีตัวอย่างต่อไปนี้

1) ไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander) PCF8574, PCF8582, PCF8584

2) ไอซีหน่วยความจำอีอีพรอมอนุกรม (Serial EEPROM) : 24Cxx, PCF8570, PCF72/73, PCF8582

3) ไอซี ADC/DAC : PCF 8591

4) ไอซีรับไทม์คล็อก (Real-Time Clock : RTC) PCF8583, PCF8593, PCF8598, 41T56C

5) ไอซีขับ LCD โมดูล (LCD Driver) : PCF8466, PCF8576, PCF8577/78, PCF8579, SAA1064

6) ไอซีกำเนิดสัญญาณ DTMF (DTMF Generator) : PCD3311/12

2.5 การออกแบบวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

คุณสมบัติของวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล จะมีลักษณะเหมือนกับวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก และจะมีคุณลักษณะพิเศษอีก 1 อย่างคือ การเปลี่ยนแปลงเวลา หมายถึง ช่วงเวลาที่วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล ใช้ในการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลได้ 1 ค่า

การออกแบบวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลนี้ จะใช้ตัวอินดิเคเตอร์ (Indicator) เป็นตัวรับค่าน้ำหนักจากโหลดเซลล์แล้วส่งผ่านทางพอร์ตอนุกรม RS232 ไปยังบอร์ดควบคุม ซึ่งภายในตัวอินดิเคเตอร์จะประกอบไปด้วยวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

2.6 ระบบควบคุมนิวแมติกส์

ความหมายของคำว่า " นิวแมติกส์ " ที่คนส่วนมากเข้าใจก็คือ การนำเอาอากาศมาเป็นวัสดุใช้งานในงานอุตสาหกรรม โดยเฉพาะอย่างยิ่ง ด้านการขับเคลื่อนหรือควบคุมเครื่องจักรหรือกระบวนการผลิตต่างๆ เพื่อให้ชัดเจนยิ่งขึ้นผู้เกี่ยวข้องทางด้านนี้ได้ให้ความหมายของคำว่า " นิวแมติกส์ " ไว้ว่าหมายถึง ระบบการส่งกำลังจากต้นทางไปยังปลายทาง โดยอาศัยลมเป็นตัวกลางใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งกำลังและ การควบคุมการทำงาน ส่วนคำว่า "นิวแมติกส์ไฟฟ้า" หมายถึง ระบบการส่งกำลังจากต้นทางไปยังปลายทาง โดยอาศัยลมเป็นตัวกลางในการส่งกำลัง และควบคุมการทำงานด้วยระบบลมผสมไฟฟ้า การควบคุมในระบบนิวแมติกส์ส่วนใหญ่แล้ว จะเป็นการควบคุมแบบ OPEN-LOOP หรือการควบคุมแบบเปิด-ปิด

2.6.1 ข้อดีและข้อเสียของระบบนิวแมติกส์

ในปัจจุบันได้มีการนำเอาระบบนิวแมติกส์มาใช้แทนระบบการควบคุมด้วยมอเตอร์ไฟฟ้ากันอย่างแพร่หลาย เนื่องจากการควบคุมด้วยระบบนิวแมติกส์ มีข้อดีหลายอย่างเมื่อเปรียบเทียบกับระบบไฟฟ้า

1) ข้อดีของระบบนิวแมติกส์

1.1) การเคลื่อนที่ในแนวเส้นตรงสามารถทำได้ง่ายและสะดวกกว่า เพราะมีอุปกรณ์ที่ทำงานในแนวเส้นตรง

1.2) การหยุดหรือเบรกระบบนิวแมติกส์สามารถกระทำได้ง่าย และสะดวกกว่า เช่นถ้าต้องการให้การทำงานของเครื่องจักรหยุดตำแหน่งใด ก็สามารถเลือกกระบอกสูบที่มีระยะชักตามตำแหน่งที่ต้องการ

1.3) การปรับความเร็วก็สามารถกระทำได้โดยง่าย กล่าวคือจะใช้อุปกรณ์ควบคุมความเร็วต่อเข้ากับระบบ

1.4) มีความปลอดภัยสูง เพราะอุปกรณ์ลมไม่เกิดการเสียหาย ถึงแม้จะใช้งานเกินกำลังเมื่อเกิดข้อบกพร่องในวงจรก็ไม่เกิดอันตรายต่อผู้ใช้เครื่องจักรนั้น

1.5) การบำรุงรักษา และการซ่อมบำรุงสามารถกระทำได้โดยง่าย

2) ข้อเสียของระบบนิวแมติกส์

2.1) มีเสียงดังในขณะที่ระบบทำงาน เพราะอุปกรณ์ทำงานต่างๆของระบบจะต้องระบายลมออกทางด้านวาล์วควบคุม แม้ว่าที่วาล์วควบคุมจะมีตัวเก็บเสียงอยู่ก็ตาม

2.2) ความดันของลมเปลี่ยนแปลงเมื่ออุณหภูมิเปลี่ยนแปลง เป็นผลทำให้การควบคุมในระบบอาจเกิดการเปลี่ยนแปลงได้

2.3) เนื่องจากลมสามารถอัดตัวได้ จะทำให้การเคลื่อนที่ของอุปกรณ์ทำงานในขณะที่มีโหลดมีโอกาสเคลื่อนที่ไม่เสมอ

2.4) ลมอัดมีความชื้น เมื่อลมอัดถูกทำให้เย็นลงหลังจากการอัดเข้าไปในถังเก็บลม จะทำให้เกิดการกลั่นตัวเป็นหยดน้ำ และเมื่อความชื้นที่ปนเข้ามากับลมอัด เข้าไปในระบบควบคุมจะทำให้อุปกรณ์ทำงานและวาล์วควบคุมต่างๆ เกิดสนิมเป็นเหตุให้เกิดการทำงานผิดพลาดไป และอายุการใช้งานของอุปกรณ์นั้นสั้นลง แต่ทั้งนี้ก็มีอุปกรณ์ที่สามารถกำจัดความชื้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

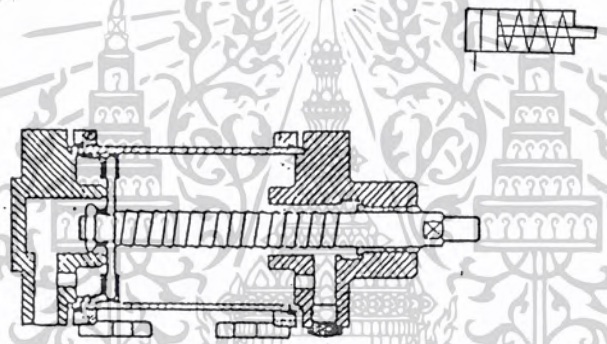
2.7 กระบอกสูบ (Cylinder)

กระบอกสูบเป็นอุปกรณ์ที่ทำงานในลักษณะแนวเส้นตรง ซึ่งแบ่งได้ดังนี้

2.7.1 กระบอกสูบทางเดียว (Single Acting Cylinder)

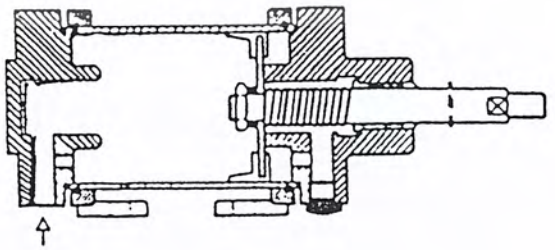
กระบอกสูบทางเดียวใช้แรงดันลมอัดกระทำก้านสูบให้เคลื่อนที่เพียงด้านเดียว ส่วนการเคลื่อนที่กลับจะอาศัยแรงสปริง กระบอกสูบแบบนี้จะใช้กับงานที่ต้องการแรงกระทำไม่มากนัก เนื่องจากแรงที่กระทำกับโหลดจะถูกต้านด้วยแรงสปริงขนาดของกระบอกสูบประเภทนี้ที่นิยมผลิตกันจะมีขนาดไม่โตกว่า 10 เซนติเมตร กระบอกสูบเป็นอุปกรณ์ที่ทำงานในแนวลักษณะแนวเส้นตรง ซึ่งแบ่งได้ดังนี้

สัญลักษณ์และหลักการทำงาน :



รูปที่ 2.1 สภาวะปกติของกระบอกสูบทางเดียว

จากรูปที่ 2.1 แสดงภาวะปกติของกระบอกสูบ เมื่อยังไม่มีแรงดันลมเข้ากระบอกสูบสปริงจะดันลูกสูบให้อยู่ในตำแหน่งเคลื่อนที่เข้าสุด



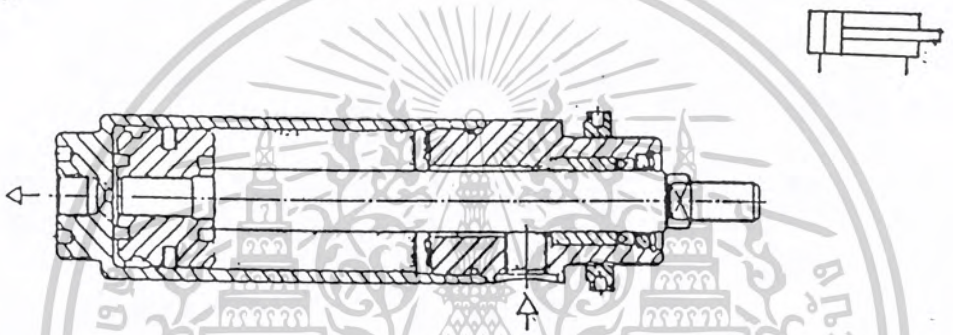
รูปที่ 2.2 สภาวะการทำงานของกระบอกสูบทางเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 แสดงสภาวะการทำงานของกระบอกสูบทางเดียว เมื่อมีแรงดันลมเข้าที่ด้านท้ายของกระบอกสูบแรงดันลมจะเอาชนะแรงสปริงและทำให้ลูกสูบเคลื่อนที่ออกสุด

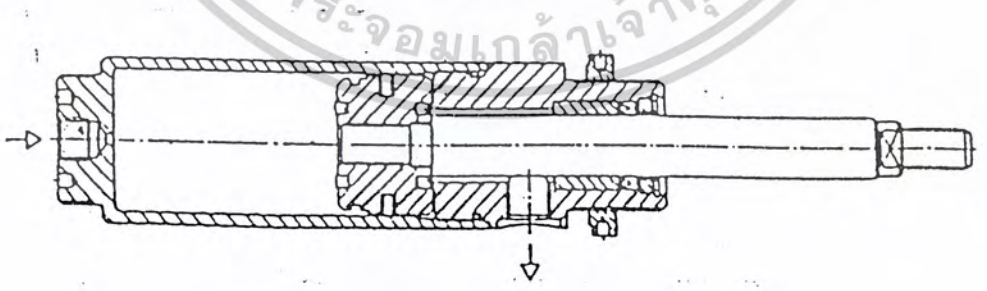
2.7.2 กระบอกสูบสองทาง (Double Acting Cylinder)

กระบอกสูบแบบสองทางจะใช้แรงดันลมกระทำให้ก้านสูบเคลื่อนที่เข้า และออกทั้งสองทาง แรงกระทำที่ได้จากกระบอกสูบชนิดนี้จะมากกว่ากระบอกสูบแบบทางเดียว เพราะแรงสปริงเป็นตัวต้าน จึงเหมาะสำหรับงานแทบทุกประเภทที่ต้องการการเคลื่อนที่ในลักษณะที่เป็นแนวเส้นตรง



รูปที่ 2.3 สภาวะปกติปิดของกระบอกสูบแบบสองทาง

จากรูปที่ 2.3 แสดงสภาวะการทำงานของกระบอกสูบ เมื่อมีแรงดันลมเข้าด้านหัวของกระบอกสูบ แรงดันลมจะดันให้ลูกสูบเคลื่อนที่เข้าสุด



รูปที่ 2.4 สภาวะการทำงานของกระบอกสูบสองทาง

จากรูปที่ 2.4 แสดงการทำงานของกระบอกสูบ เมื่อมีแรงดันลมเข้าที่ด้านท้ายของกระบอกสูบแรงดันลมจะดันให้ลูกสูบเคลื่อนที่ออกสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 โซลินอยด์วาล์ว

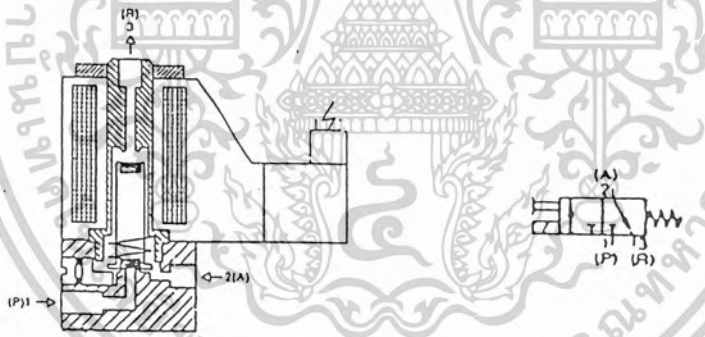
ในการควบคุมอุปกรณ์ทำงานในระบบนิวแมติกส์ให้ทำงานตามความต้องการนั้น อุปกรณ์ที่ทำให้อุปกรณ์ทำงานเปลี่ยนตำแหน่งได้ก็คือ **วาล์ว** ซึ่งในการเลื่อนวาล์วควบคุมนั้นสามารถทำได้หลายวิธีด้วยกัน เช่น การเลื่อนวาล์วโดยใช้กล้ามเนื้อ การเลื่อนโดยใช้กลไก การเลื่อนโดยใช้ลม การเลื่อนโดยใช้ไฟฟ้าควบคุม หรือการเลื่อนโดยใช้วิธีใดวิธีหนึ่งร่วมกัน ซึ่งในที่นี้จะขอกล่าวถึงวาล์วที่ใช้ไฟฟ้าเป็นตัวควบคุมในการเปลี่ยนตำแหน่ง หรือเรียกอีกอย่างหนึ่งว่า "โซลินอยด์วาล์ว"

2.8.1 โครงสร้างและหลักการทำงานของโซลินอยด์วาล์ว

โครงสร้างของวาล์วควบคุมโดยทั่วไปในที่นี้จะหมายรวมถึง โซลินอยด์วาล์วด้วย ซึ่งสามารถแบ่งออกเป็น 2 ชนิดใหญ่ด้วยกัน คือ วาล์วแบบนั่งบ่า และวาล์วแบบลูกสูบเลื่อน โดยที่วาล์วแต่ละประเภทจะมีคุณสมบัติที่แตกต่างกัน ดังที่จะกล่าวถึงต่อไปนี้

1) วาล์ว 3/2 ปกติปิดเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับสู่สภาพเดิมโดยแรงสปริง

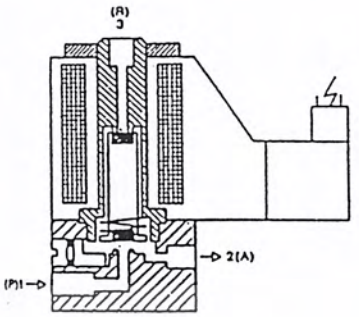
สภาวะปกติ : เมื่อยังไม่มีกระแสไหลเข้าขดลวดโซลินอยด์ โซลินอยด์จึงยังไม่มีอำนาจแม่เหล็ก แรงสปริงจะดันให้ลิ้นของวาล์วปิดทางลมจาก P ไม่ให้ผ่านไปยัง A ดังรูปที่ 2.5



รูปที่ 2.5 สภาวะปกติของวาล์ว 3/2

สภาวะการทำงาน : เมื่อมีกระแสไฟไหลผ่านขดลวดโซลินอยด์ เป็นผลทำให้โซลินอยด์เกิดแรงดึงดูดจากอำนาจแม่เหล็กเลื่อนลิ้นของวาล์วให้เคลื่อนที่ขึ้นเปิดทางลม ทำให้แรงดันลมจาก P ต่อถึง A ส่วนรู R จะถูกปิด ดังแสดงในรูปที่ 2.6

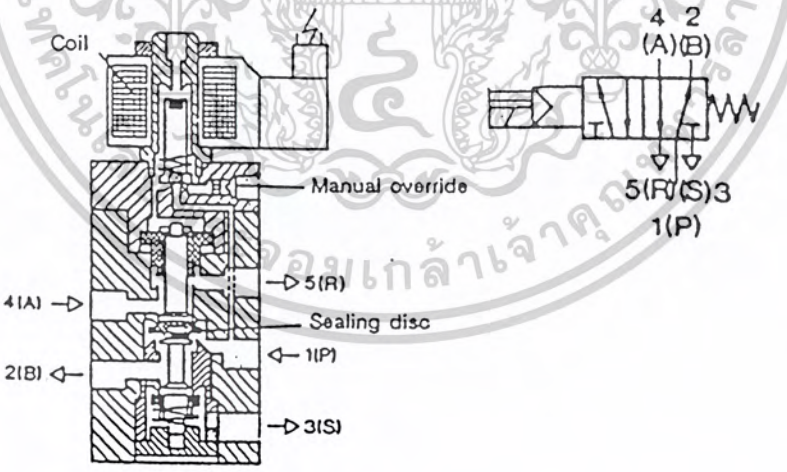
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 สภาวะการทำงานของวาล์ว 3/2

2) วาล์ว 5/2 ใช้โซลินอยด์และลมเป็นตัวเลื่อนวาล์ว กลับสู่สภาพเดิมโดยสปริง

สภาวะปกติ : เมื่อยังไม่มีกระแสไฟไหลผ่านขดลวดโซลินอยด์โซลินอยด์จะยังไม่มีอำนาจแม่เหล็ก แรงสปริงจะดันลิ้นเล็กด้านบนปิดทางลมจาก P ลิ้นใหญ่จะไม่ถูกเลื่อน เป็นผลทำให้รู P ต่อถึงรู B รู A จะต่อถึงรู R ส่วนรู S จะถูกปิด ดังรูปที่ 2.8

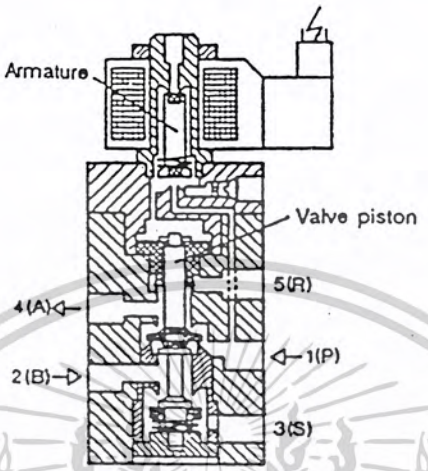


รูปที่ 2.7 สภาวะการปกติของวาล์ว 5/2

สภาวะทำงาน : เมื่อมีกระแสไฟไหลผ่านขดลวดโซลินอยด์ จะทำให้โซลินอยด์เกิดอำนาจแม่เหล็กเลื่อนลิ้นเล็กด้านบนให้เปิดทางลม ดังนั้นลมจากรู P จะผ่านลิ้นเล็กด้านบนไปเลื่อน

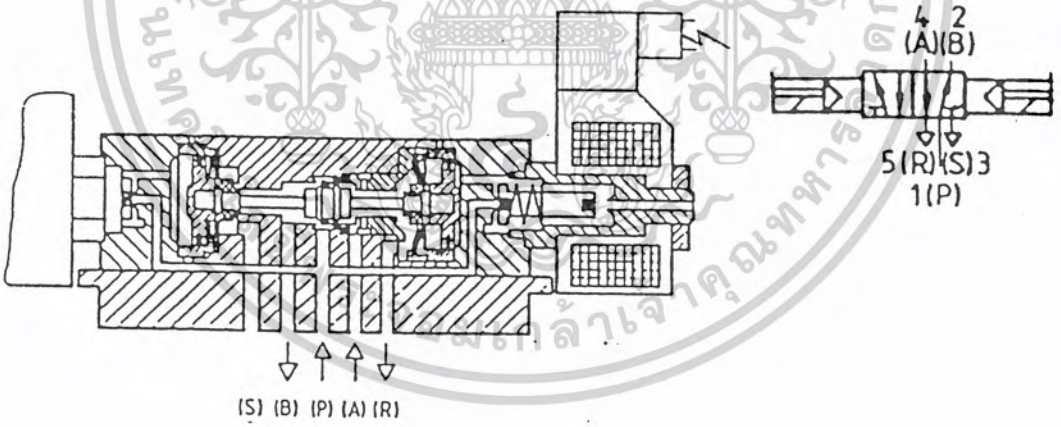
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิ้นใหญ่ ทำให้แรงดันลมจาก P เปลี่ยนทิศทาง คือรู P จะต่อถึงรู A รู B จะต่อถึงรู S ส่วนรู R จะถูกปิด ดังรูปที่ 2.8



รูปที่ 2.8 สภาวะการทำงานของวาล์ว 5/2

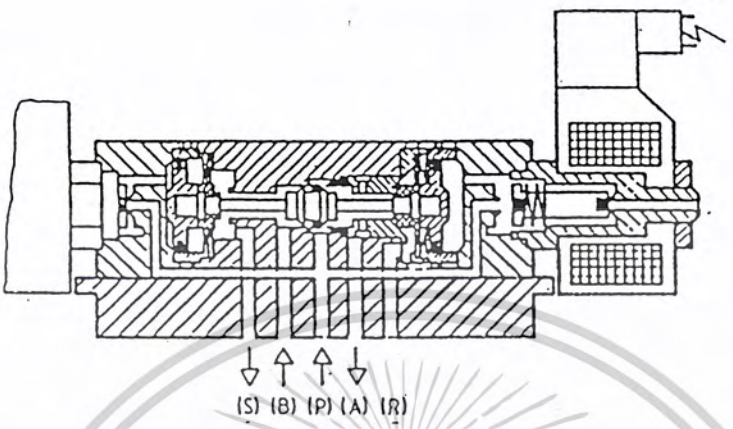
3) วาล์ว 5/2 โซลินอยด์เปิดทางลมและลมเป็นตัวเลื่อนวาล์วทั้งสองด้านแสดงดังรูป



รูปที่ 2.9 สภาวะการทำงานของวาล์ว 5/2 เมื่อมีกระแสไหลผ่านทางด้านซ้ายมือ

สภาวะทำงาน : จากรูปที่ 2.10 เมื่อมีกระแสไฟไหลผ่านขดลวดโซลินอยด์ด้านซ้ายมือ จะทำให้โซลินอยด์เกิดแรงดึงดูดจากอำนาจแม่เหล็กเลื่อนลิ้นเล็กด้านซ้ายมือให้เปิดทางลม ลมจาก P จะไหลผ่านลิ้นเล็กไปเลื่อนลิ้นของเมนวาล์วให้เลื่อนไปทางขวามือเป็นผลทำให้เกิดแรงดันลมจาก P ต่อถึง B และ A จะต่อถึงรู R ส่วนรู S จะถูกปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 สภาวะการทำงานของวาล์ว 5/2 เมื่อมีกระแสไฟไหลผ่านทางขวามือ

สภาวะทำงาน : จากรูปที่ 2.11 เมื่อมีกระแสไฟไหลผ่านขดลวดโซลินอยด์ด้านขวามือ จะทำให้โซลินอยด์เกิดแรงดึงดูดจากอำนาจแม่เหล็กเลื่อนลิ้นเล็กด้านขวามือให้เปิดทางลม ลมจาก P จะไหลผ่านลิ้นเล็กไปเลื่อนลิ้นของเมนวาล์วไปทางซ้ายมือ เป็นผลทำให้แรงดันลมจาก P เปลี่ยนทิศทางการไหลคือ รู P ต่อถึงรู A รู B ต่อถึงรู A รู B ต่อถึงรู S ส่วนรู R ถูกปิด

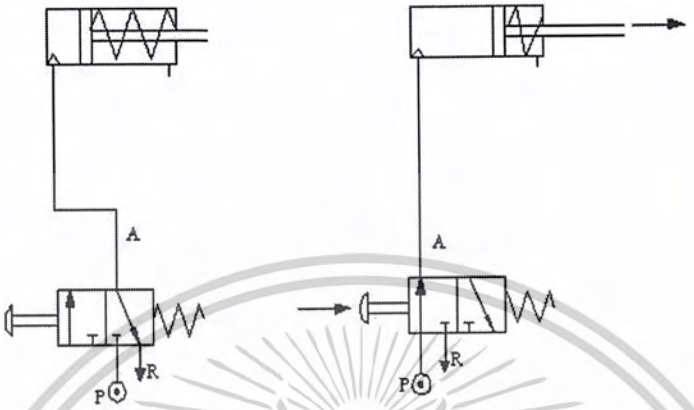
2.9 วงจรนิวแมติกส์พื้นฐาน

การออกแบบการทำงานเครื่องจักรกลหรือระบบกระบวนการต่างๆ ให้มีลักษณะการทำงานตามที่ต้องการนั้นมีความจำเป็นในงานอุตสาหกรรมทุกสาขา ในระบบนิวแมติกส์ก็เช่นกัน การนำความรู้เกี่ยวกับอุปกรณ์ทำงาน และวาล์ควบคุมแบบต่างๆที่ได้กล่าวมาในตอนต้นมาใช้ในการออกแบบวงจรนิวแมติกส์ในรูปแบบต่างๆ เพื่อให้ได้ผลผลิตตามที่ต้องการ รวมทั้งทำให้เกิดการประหยัด และใช้อุปกรณ์อย่างมีประสิทธิภาพสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 วงจรควบคุมการทำงานของกระบอกสูบ

1) วงจรควบคุมกระบอกสูบทำงานทางเดียว

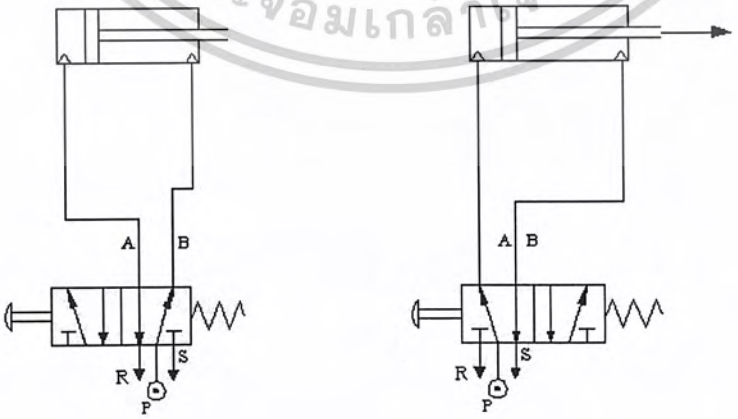


รูปที่ 2.11 วงจรควบคุมกระบอกสูบทางเดียว

สภาวะปกติ : เมื่อยังไม่กดวาล์ว 3/2 ลมจากรู P จะถูกกั้นไว้ ที่รู A และ R ต่อถึงกัน ก้านสูบจะเคลื่อนที่เข้าสุดด้วยแรงสปริงภายในกระบอกสูบ

สภาวะสั่งงาน : เมื่อกดวาล์ว 3/2 แรงดันลมจะไหลผ่านรู P ไปรู A แล้วเข้าไปดันให้กระบอกสูบเคลื่อนที่ออก ส่วนรู R จะถูกปิดกั้นไว้ และเมื่อปล่อยมือสปริงจะเลื่อนลิ้นของวาล์วให้สู่สภาพเดิม ทำให้แรงดันลมจากรู P จะถูกปิดกั้นการไหล แรงดันลมภายในห้องสูบจะระบายออกสู่บรรยากาศที่รู R ก้านสูบจะเคลื่อนที่กลับโดยแรงสปริงภายในกระบอกสูบ

2) วงจรควบคุมการทำงานของกระบอกสูบ 2 ทาง



รูปที่ 2.12 วงจรควบคุมการทำงานของกระบอกสูบ 2 ทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

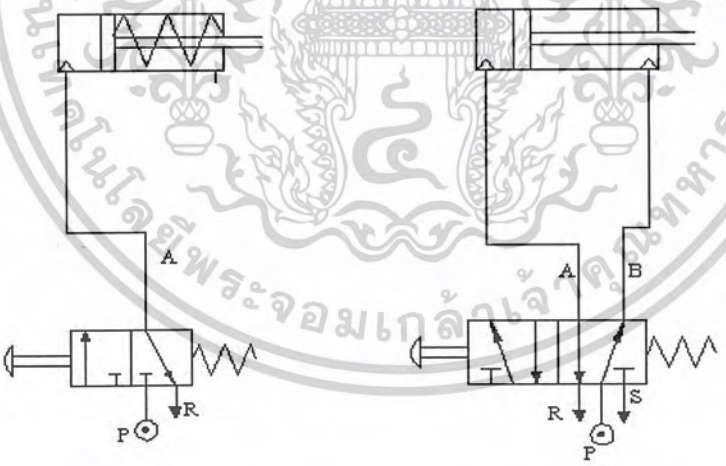
สภาวะปกติ : แรงดันลมจากรู P จะไหลผ่านรู B เข้าไปยังห้องสูบส่วนหัวหรือด้านขวามือ ดันให้ลูกสูบเคลื่อนที่เข้า ส่วนลมจากห้องสูบส่วนท้ายหรือด้านซ้ายมือ จะระบายออกสู่บรรยากาศผ่านรู A ไปยังรู R

สภาวะสั่งงาน : เมื่อกดวาล์ว 5/2 แรงดันลมจากรู P จะเปลี่ยนทิศทางการไหลคือ แรงดันลมจากรู P ไหลผ่านรู A ให้กระบอกสูบเคลื่อนที่ออก ส่วนลมจากห้องสูบด้านขวามือจะระบายออกสู่บรรยากาศผ่านรู A ไปยัง S และเมื่อปล่อยมือสปริงจะเลื่อนให้วาล์ว 5/2 กลับสู่สภาวะปกติเป็นผลทำให้ลูกสูบเคลื่อนที่เข้า

2.9.2 ลักษณะของการควบคุม

การควบคุมในระบบนิวแมติกส์ สามารถแบ่งลักษณะของการควบคุมออกเป็น 2 ประเภทด้วยกันคือ การควบคุมโดยตรงและการควบคุมโดยอ้อม

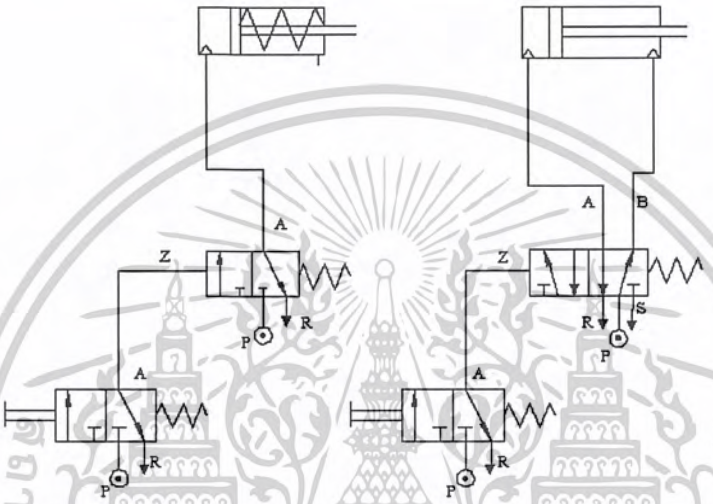
1) **การควบคุมโดยตรง** คือ แรงดันลมจากแหล่งจ่ายลมจะผ่านอุปกรณ์ควบคุมไปยังอุปกรณ์ทำงานหรือกระบอกสูบโดยตรง การควบคุมลักษณะนี้จะใช้ในกรณีที่ตำแหน่งการควบคุมอยู่ไม่ไกลจากอุปกรณ์มากนัก ถ้าอยู่ไกลกันมากจะทำให้เกิดความดันตกคร่อมที่ท่อส่ง ทำให้แรงดันลมที่ไหลเข้ากระบอกสูบต่ำลงเป็นผลให้แรงกระทำชิ้นงานลดลงด้วย



รูปที่ 2.13 วงจรการควบคุมโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
48318

2) การควบคุมแบบทางอ้อม คือ แรงดันลมที่ผ่านวาล์วควบคุมจะไม่ไหลเข้าไปยังกระบอกสูบโดยตรง แต่จะไหลเข้าไปยังเมนวาล์วจ่ายลมให้กระบอกสูบ การควบคุมแบบนี้จะแก้ปัญหาการเกิดความดันตกคร่อมในท่อส่ง หรือลดปัญหาในเรื่องแรงกระทำของกระบอกสูบลดลง และจะใช้ในกรณีที่จุดควบคุมอยู่ห่างจากอุปกรณ์ทำงานมากๆ ดังที่แสดงในรูปที่ 2.15



รูปที่ 2.14 วงจรควบคุมแบบทางอ้อม

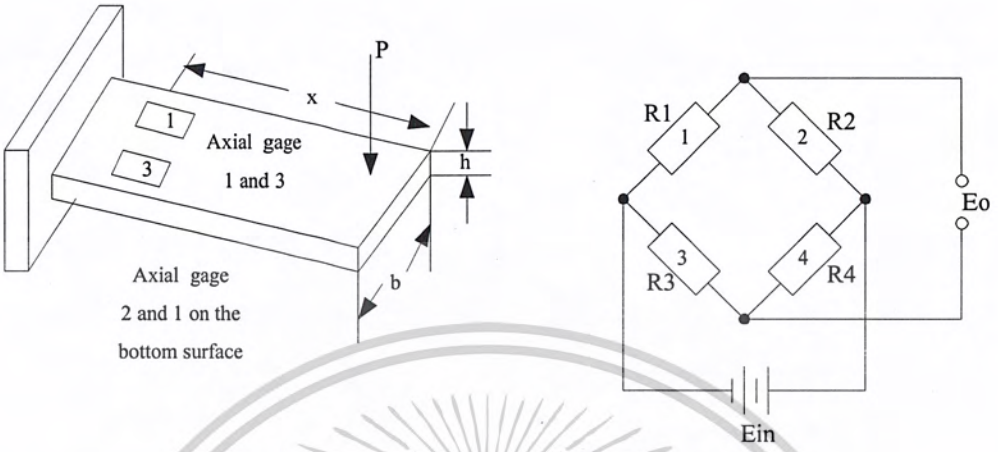
2.10 อุปกรณ์ทรานส์ดิวเซอร์ (โหลดเซลล์)

ทรานส์ดิวเซอร์ ซึ่งใช้วัดแรงทอร์คหรือความดันโดยทั่วไป ประกอบด้วยส่วนยืดหยุ่นหรือการเปลี่ยนแปลงเป็นระยะทางหรือความเครียด และการตรวจจับโดยปกติจะใช้เกจความเครียดในการวัดสัญญาณ อย่างไรก็ตามในบางครั้งจะใช้ LDVT หรือ โพเทนทิโอมิเตอร์สำหรับการวัดแบบสถิตย์ (Static) หรือควอไซสแตติกส์ (Quasi Static)

2.10.1 ทรานส์ดิวเซอร์แบบหลักการของแรง

1) โหลดเซลล์แบบคาน (Beam-Type Load Cell) โหลดเซลล์แบบคานที่นิยมใช้วัดภาระในกรณีที่ใช้โหลดเซลล์แบบลิงค์ไม่ได้ รูปที่ (ก) เป็นคานยื่นซึ่งมีเกจความเครียด 2 อัน อยู่ที่ผิวด้านล่าง(ทั้งหมดติดอยู่ในแนวขนานกับแนวของคาน) ซึ่งทำหน้าที่เป็นชิ้นส่วนยืดหยุ่นและเซนเซอร์ของเซลล์ชนิดนี้ คือ เกจซึ่งต่อกันเป็นวงจรวีสโตนบริดจ์ ดังแสดงในรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) โครงสร้างของโหลดเซลล์แบบคาน (ข) วงจรสมมูลย์

รูปที่ 2.15 โหลดเซลล์แบบคาน

ภาระ P ทำให้เกิดโมเมนต์ $M = P_X$ ที่ตำแหน่ง X ทำให้ได้ผลลัพธ์

$$\varepsilon_3 = -\varepsilon_2 = \varepsilon_3 = \varepsilon_1 = \frac{\sigma M}{Ebh^2} = \frac{6P_x}{Ebj^2} \qquad (2.1)$$

เมื่อ b คือ ความกว้างของหน้าตัดของคาน

h คือ ความหนาของหน้าตัดคาน

การตอบสนองของเกจความเครียดหาได้จากสมการ $\frac{R}{R} = S_g$ และสมการ (2.1) ดังนั้น

$$\Delta\frac{R_1}{R_1} = \Delta\frac{R_2}{R_2} = \Delta\frac{R_3}{R_3} = \Delta\frac{R_1}{R_1} = \Delta\frac{R_4}{R_4} = \sigma\frac{S_x P_x}{Ebh^2} \qquad (2.2)$$

แรงดันออก V_o จากวีทสโทนบริดจ์ ซึ่งเป็นผลจากการกระทำของภาระ P หาได้โดยแทนสมการ (2.3) ลงในสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้

$$V_0 = \left[\frac{r}{(1+r)^2} \left(\frac{\Delta R_1}{R_1} \frac{\Delta R_2}{R_2} + \frac{\Delta R_3}{R_3} \frac{\Delta R_4}{R_4} \right) \right] V_g \qquad (2.3)$$

และสมมติว่าเกจความเครียดทั้ง 4 เหมือนกันทุกประการจะได้

$$V_0 = \sigma \frac{S_g P V_x}{Ebh^2} \qquad (2.4)$$

หรือ

$$P = \left[\frac{Ebh^2}{\sigma S_g X V_x} \right] V_0 = C V_0 \qquad (2.5)$$

จากสมการ (2.5) แสดงว่าภาระ P เป็นสัดส่วนกับแรงดัน V_o และค่าคงตัวสัดส่วนหรือค่าคงตัวปรับเทียบ (Calibration Constant) C คือ

$$C = \frac{Ebh^2}{\sigma S_g X V_x} \qquad (2.6)$$

ความไวของการรวมกันระหว่างโหลดเซลล์กับวีสโตนบริดจ์ กำหนดโดยสมการ $S = V_0 / P$ ดังนั้น

$$S = \frac{V_0}{P} = \frac{1}{C} = \sigma \frac{S_g P V_x}{Ebh^2} \qquad (2.7)$$

จากสมการ (2.7) ข้างต้นแสดงว่าความไวของโหลเซลล์แบบคานขึ้นอยู่กับรูปร่างหน้าตัดของคาน (b, h) โมดูลัสของความยืดหยุ่นของวัสดุที่ใช้ทำคาน (E) ตำแหน่งของภาระเทียบกับเกจ (X) เกจแฟกเตอร์ (S_g) และแรงดันที่ป้อนให้กับวงจรบริดจ์ (V_s) ช่วงของโหลดที่ใช้กับโหลดเซลล์แบบคานขึ้นอยู่กับ รูปร่างหน้าตัดของคานตำแหน่งที่ภาระกระทำและความหล้าของวัสดุที่ใช้ทำคาน ถ้าสมมติว่าเกจติดอยู่ที่ตำแหน่งใกล้ตัวรองรับ (Beam Support) จะได้

$$M_{gage} = M_{max} \qquad (2.8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

$$P_{max} = \frac{S_f bh^2}{\sigma X} \quad (2.9)$$

อัตราส่วนแรงดันที่ภาวะสูงสุด $(\frac{V_0}{V_s})_{max}$ หาได้จากการแทนสมการ P_{max} ลงในสมการ (2.7) ดังนั้น

$$(\frac{V_0}{V_s})_{max} = \frac{S_g S_f}{E} \quad (2.10)$$

2.11 เครื่องชั่งอิเล็กทรอนิกส์ (Electronic Scale)

ในปัจจุบันการซื้อขายสินค้าแทบทุกประเภท จะซื้อขายกันในรูปแบบของน้ำหนัก ดังนั้น การชั่งจึงมีบทบาทในด้านอุตสาหกรรมและการพาณิชย์ เครื่องชั่งที่ไม่มีคุณภาพจะทำให้เกิดการสูญเสียในเชิงพาณิชย์อย่างมาก เครื่องชั่งจึงเป็นเครื่องมือที่สำคัญอย่างหนึ่งที่สามารถควบคุมกำไรหรือขาดทุนได้ เครื่องชั่งที่ดีจะต้องมีคุณสมบัติดังนี้

1) ความเชื่อถือได้ ซึ่งหมายถึงความแม่นยำ (Accuracy) เมื่อเทียบกับน้ำหนักมาตรฐาน

2) ความมีเสถียรภาพ (Stability or Consistency) ซึ่งหมายถึงการชั่งน้ำหนักชิ้นเดียวกันหลายๆ ครั้ง จะต้องอ่านค่าได้ใกล้เคียงกันมาก

3) ความไวต่อการรับรู้ (Sensitivity) ซึ่งขึ้นอยู่กับขนาดของเครื่องชั่ง เครื่องชั่งที่มีขนาดเล็กจะต้องวัดค่าความแตกต่างของน้ำหนักที่น้อยๆ ได้ เช่น วัดเป็นมิลลิกรัม แต่การที่จะวัดว่าต้องละเอียดมากแค่ไหนนั้นขึ้นอยู่กับจุดประสงค์ของการใช้งาน เพราะเครื่องชั่งที่มีขนาดใหญ่ก็จะสามารถวัดค่าความละเอียดได้น้อย

องค์ประกอบของเครื่องชั่งอิเล็กทรอนิกส์ มีดังนี้

2.11.1 โหลดเซลล์ (Load Cell)

เป็นอุปกรณ์ที่รับรู้น้ำหนักของวัตถุ แล้วทำหน้าที่เปลี่ยนจากน้ำหนักมาเป็นสัญญาณไฟฟ้า หลักการทำงานของโหลดเซลล์มีหลายชนิด ในที่นี้จะขอกล่าวถึงชนิดที่นิยมใช้กันได้แก่

1) สเตรนเกจโหลดเซลล์ (Strian Guage Load Cell) ซึ่งภายในประกอบด้วยเส้นลวดตัวนำ (Wire Strain Guage or Resistance Strain Guage) เส้นผ่านศูนย์กลางระหว่าง 0.0005 นิ้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึง 0.001 นิ้ว ซึ่งอาจทำด้วยทองแดง 45% และนิกเกิล 55% หรือ นิกเกิล 80% โครเมียม 20% ซึ่งสามารถให้ Strain Sensivity ได้สูง "Strain Sensitivity" ก็คือ ความไวต่อการเปลี่ยนแปลงความตึง (Strain) ในเส้นลวดนั้น ซึ่งจะแปลเป็นค่าสัญญาณไฟฟ้า กล่าวคือ เพียงแต่ได้รับแรงเพียงเล็กน้อยก็สามารถให้สัญญาณไฟฟ้าได้นั่นเอง เส้นลวดหรือแถบลวดนี้จะติดอยู่กับกระดาษหรือแผ่นฟิล์มพลาสติกที่มีความหนาประมาณ 0.003 นิ้ว ด้วยไนโตรเซลลูโลสหรือเรซิ่น(Nitro-Cellulose or Resin) หลายๆแผ่นรวมกัน เมื่อเส้นลวดนี้มีน้ำหนักมากดจะทำให้ทั้งความยาวและพื้นที่หน้าตัดของเส้นลวดเปลี่ยนแปลงไป เป็นผลทำให้ความต้านทานไฟฟ้า (Resistance) ของเส้นลวดเปลี่ยนแปลงไป เมื่อนำเส้นลวดนี้มาเป็นแขนหนึ่งของวงจรบริดจ์ (Bridge Circuit) ทางด้านไฟฟ้า ขณะที่แขนทั้งสามของวงจรเป็นตัวต้านทานที่รู้ค่าและมีค่าเท่ากัน เมื่อเกิดการเปลี่ยนแปลงความต้านทานบนเส้นลวดตัวนำจะทำให้วงจรบริดจ์เกิดการไม่สมดุลย์กันขึ้น และสามารถวัดออกมาเป็นรูปของแรงดันไฟฟ้าได้โดยที่ถ้ามีน้ำหนักกดลงไปที่โหลดเซลล์มากเท่าไร ก็จะทำให้ความต้านทานในเส้นลวดตัวนำเปลี่ยนไปมากขึ้นด้วยเป็นสัดส่วนกัน อย่างไรก็ตามค่าแรงดันไฟฟ้าที่วัดได้ยังน้อยมากจะต้องนำมาขยายสัญญาณอีกทีหนึ่งในวงจรภาคขยาย

2.11.2 ชุดควบคุม (Control Unit)

มีหน้าที่แปลงสัญญาณจากโหลดเซลล์ให้มาอยู่ในรูปแบบที่เราสามารถอ่านได้ เช่น เป็นตัวเลข (Digital) โดยชุดควบคุมนั้นยังจะทำหน้าที่เป็นเครื่องคำนวณ มีระบบหน่วยความจำสามารถเก็บค่าน้ำหนักในการชั่งแต่ละครั้ง

2.11.3 หน่วยแสดงผล

หน้าที่ของหน่วยแสดงผลคือจะแสดงผลของค่าน้ำหนัก โดยจะแสดงออกมาออกมาในรูปที่เข้าใจได้ง่าย และสามารถรู้ค่าที่อ่านได้ทันที เช่น เป็นตัวเลข (Digital) สามารถจะแสดงออกทางจอภาพหรือพิมพ์ออกทางเครื่องพิมพ์ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

ในการออกแบบและการสร้างเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ จะประกอบด้วยส่วนของโครงสร้าง วงจรต่างๆ ได้แก่ วงจรแหล่งจ่ายไฟ วงจรรวมสัญญาณจากโหลดเซลล์ วงจรภาคแสดงผลแบบดิจิตอล วงจรขับโซลินอยด์วาล์ว และวงจรควบคุม

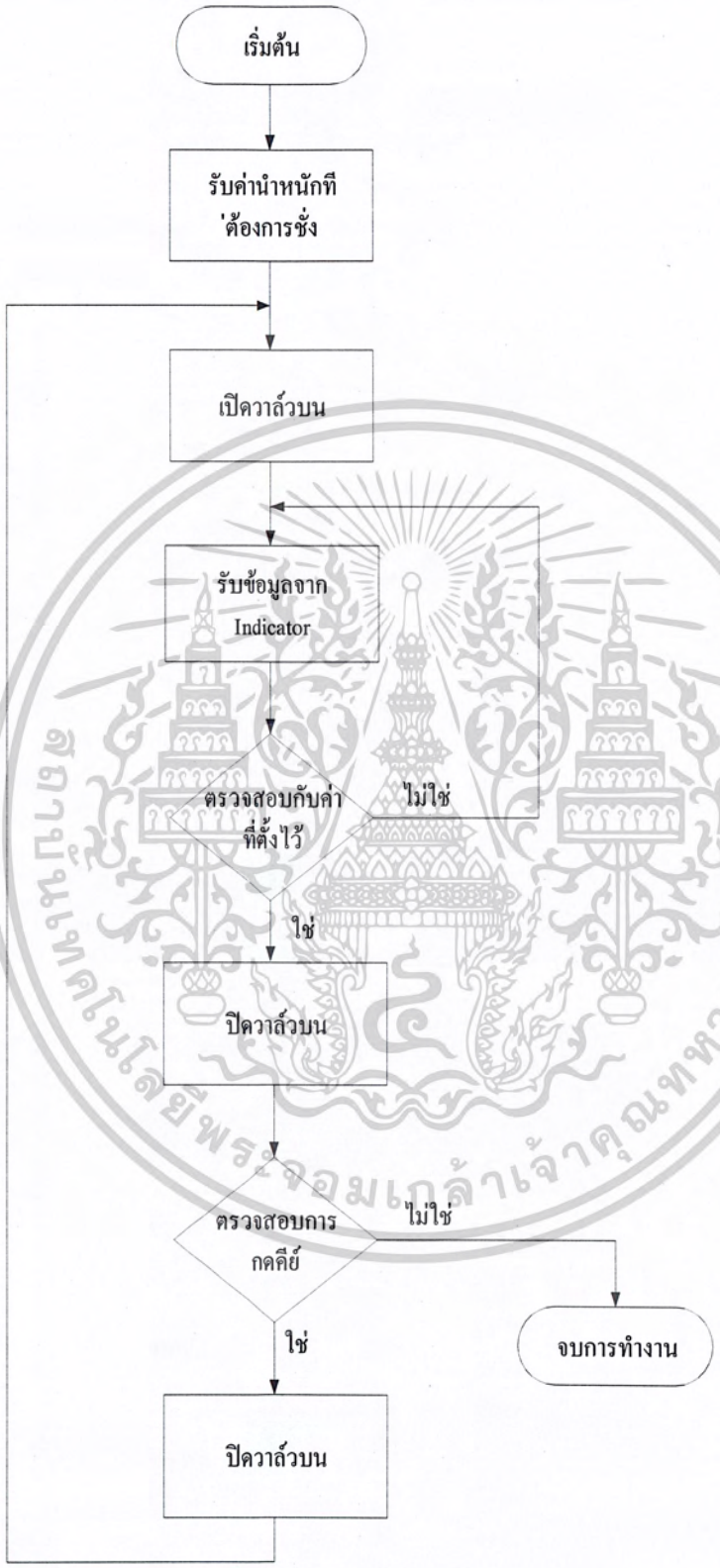
3.2 ขั้นตอนการทำงาน

ขั้นตอนการทำงานของโครงงานแสดงด้วยแผนผังการทำงาน ดังแสดงในรูปที่ 3.1 และขั้นตอนการทำงานของชุดควบคุมแสดงด้วยบล็อคไดอะแกรมการทำงานดังในรูปที่ 3.2

3.2.1 การทำงานของแผนผังโครงงาน

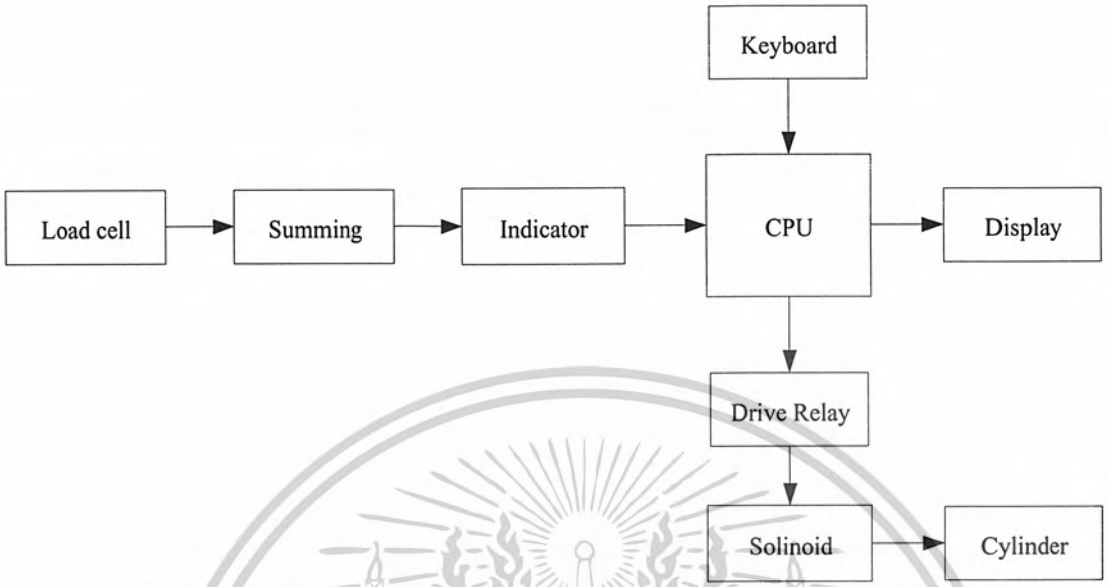
การทำงานของผังโครงงานเริ่มจากการรับค่าน้ำหนักที่ต้องการจะชั่ง จากนั้นจะทำการเปิดวาล์วบนเพื่อให้ข้าวไหลลงมาแล้วรับข้อมูลจาก Indicator แล้วทำการตรวจสอบกับค่าที่ตั้งไว้ ถ้าได้ตามค่าที่ตั้งไว้ให้ทำการปิดวาล์วบน ถ้าค่าที่ได้ยังไม่ตรงกับค่าที่ตั้งไว้ให้กลับไปรับค่าจาก Indicator เมื่อวาล์วตัวบนถูกปิดจะรอรับค่าจากการกดคีย์ ถ้ามีการกดคีย์จะสั่งให้เปิดวาล์วตัวล่าง เพื่อให้ข้าวที่ได้รับการชั่งลงสู่ถุง แล้วจะเริ่มต้นการทำงานใหม่โดยกลับไปเปิดวาล์วตัวบนอีกครั้ง แต่ถ้าไม่มีการกดคีย์จะเป็นการสิ้นสุดการทำงาน การทำงานของเครื่องชั่งจะเป็นเช่นนี้ไปตลอดแล้วแต่ผู้ที่ใช้งานนั้นจะสั่งให้ชั่งน้ำหนักเท่าใดแต่ไม่ควรเกิน 25 กิโลกรัมตามพิกัดของเครื่องชั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.1 แผนผังการทำงานของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนผังการทำงานของวงจรควบคุม

3.2.2 การทำงานของวงจรควบคุม

เมื่อจ่ายไฟเข้าไปที่โหลดเซลล์แรงดันของโหลดเซลล์ที่วัดได้จะอยู่ในรูปของแรงดันไฟฟ้ากระแสตรง ซึ่งมีค่าน้อยมากมีหน่วยเป็นมิลลิโวลต์ แรงดันที่วัดได้จากโหลดเซลล์ทั้งสองตัวจะมีค่าไม่เท่ากัน ดังนั้นจึงต้องผ่านวงจรรวมสัญญาณเพื่อปรับค่าแรงดันให้เท่ากัน แล้วจึงเข้าไปที่วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล (A/D) เพื่อที่ไมโครคอนโทรเลอร์ซึ่งเป็นตัวประมวลผลการทำงาน ไมโครคอนโทรเลอร์จะประมวลผลผ่านมาทางคีย์บอร์ดแล้วจึงแสดงผลมาทางตัวแสดงผล เช่นเดียวกับเซนเซอร์ที่จะส่งสัญญาณสั่งงานมาให้กับไมโครคอนโทรเลอร์ประมวลผลและสั่งงานแสดงผล ซึ่งไมโครคอนโทรเลอร์จะเป็นตัวสั่งงานให้โซลินอยด์ทำงานควบคุมการปิด/เปิดของกระบอกสูบ สรุปคือเราจะเห็นได้ว่าเราได้ใช้ไมโครคอนโทรเลอร์เป็นตัวรับและควบคุมการทำงานทั้งหมดของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

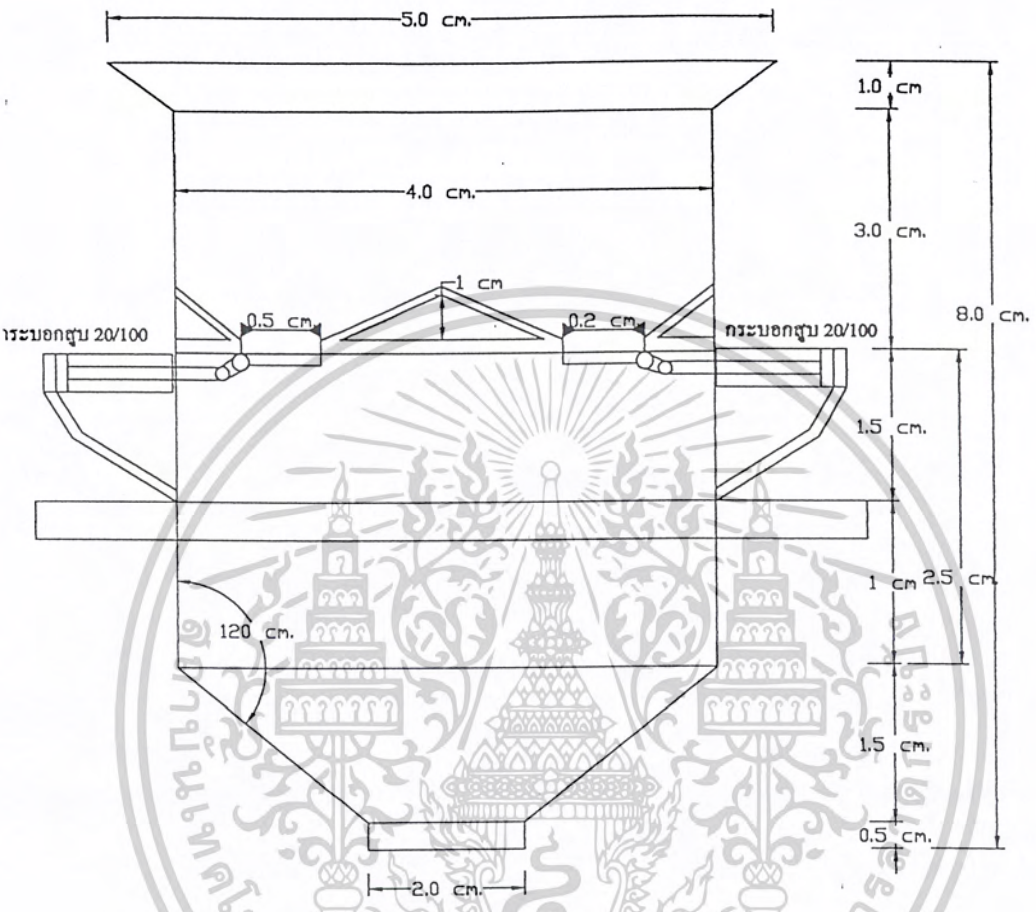
3.3 การออกแบบเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

เครื่องชั่งอัตโนมัตินี้ประกอบด้วยทั้งหมด 4 ส่วนด้วยกันดังที่กล่าวมานั้น ซึ่งความสำคัญทั้ง 3 ส่วนโดยถังใบที่ 1 คือส่วนควบคุมการโรยของวัตถุดิบ ซึ่งจะปล่อยวัตถุดิบลงสู่ถังที่ 2 คือส่วนควบคุมการชั่งตวง ซึ่งเป็นส่วนที่มีความสำคัญซึ่งจะทำหน้าที่ชั่งวัตถุดิบที่เราได้ปล่อยลงมาจากส่วนควบคุมการชั่งตวง ซึ่งเป็นส่วนที่มีความสำคัญซึ่งจะทำหน้าที่ชั่งวัตถุดิบที่เราได้ปล่อยลงมาจากถังที่ 1 แล้วปล่อยลงสู่ถังที่ 3 ซึ่งเสมือนเส้นทางลำเลียงเพื่อที่จะบรรจุหีบห่อ และถังทั้ง 3 ใบนี้จะถูกยึดติดกับตัวโครง มีแผ่นเหล็กเพื่อยึดติดกับแต่ละส่วนไว้สามารถแยกถอดประกอบได้ ซึ่งจะแสดงการออกแบบตามลำดับ

3.3.1 ส่วนควบคุมการโรยของวัตถุดิบ

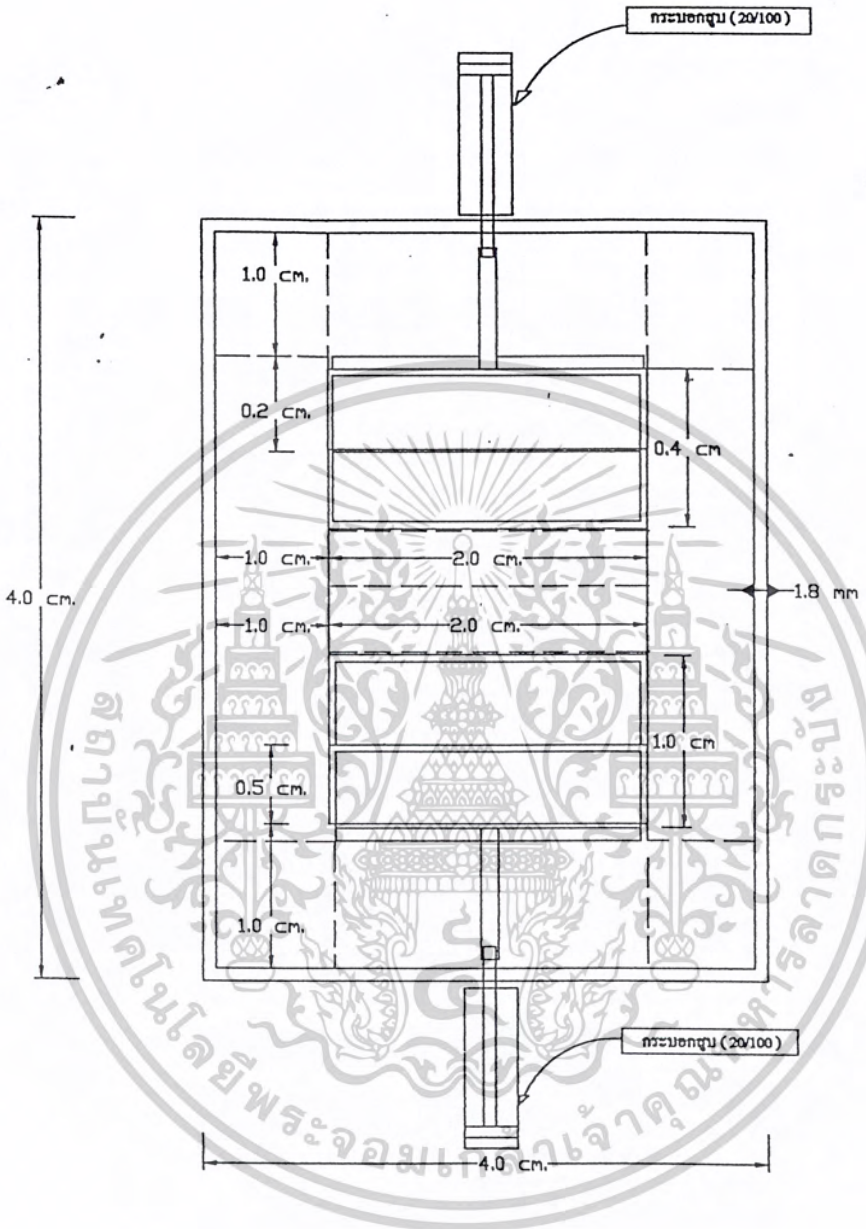
การทำงานของส่วนควบคุมการโรยของวัตถุดิบมีหน้าที่คือ ปิดและเปิดให้ข้าวสารไหลลงตามปริมาณที่เราได้ตั้งไว้ตามปริมาณของน้ำหนัก โดยจะทำการบังคับการปิดเปิดโดยกระบอกสูบทั้ง 2 ตัวที่อยู่ด้านใน ซึ่งมีขนาด 100/20 mm. ซึ่งจะทำการปิดเปิด 2 ช่องซึ่งมีขนาด 5 X 20 cm และ 2 X 20 cm ดังแสดงในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



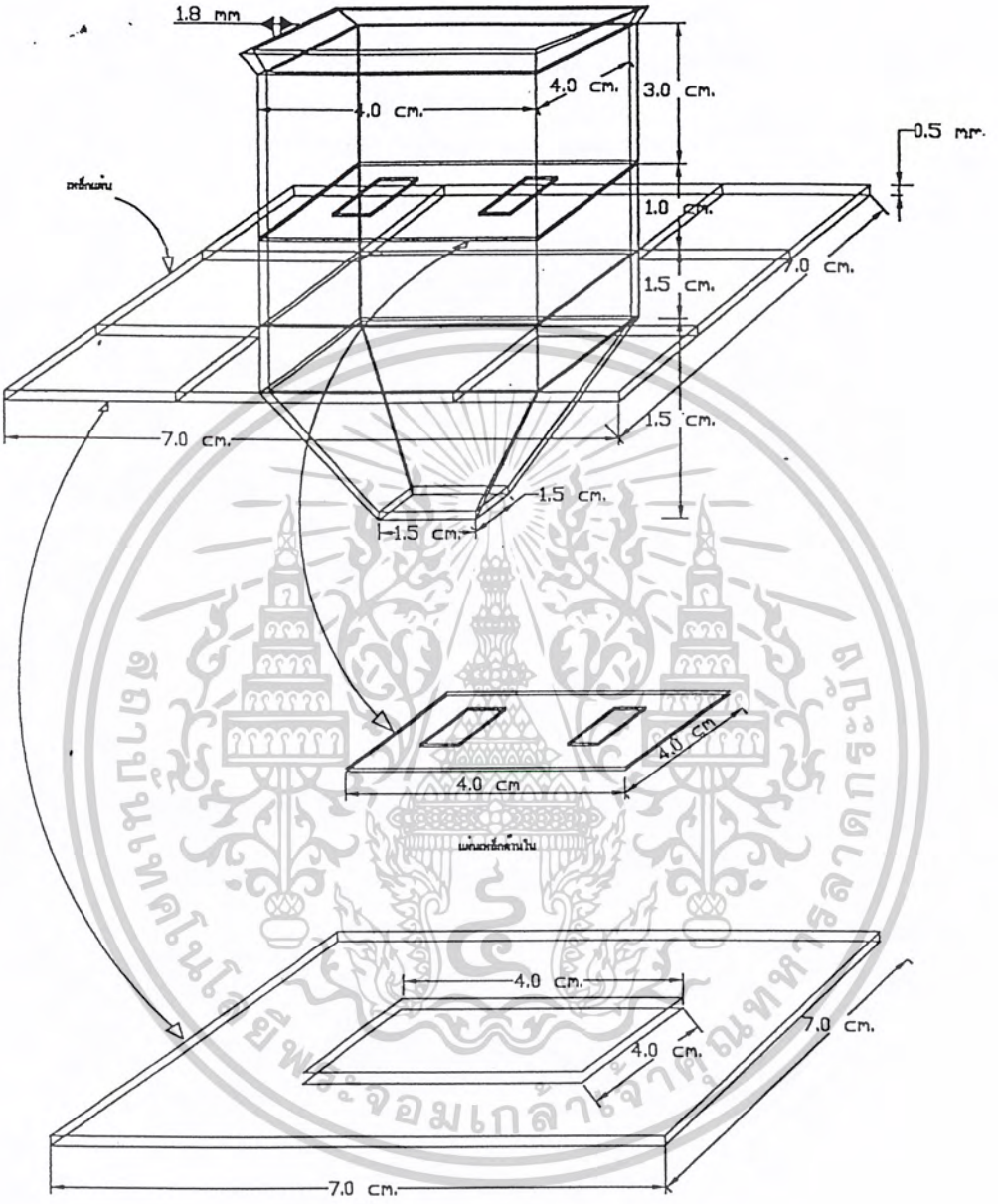
รูปที่ 3.3 แบบส่วนควบคุมการโรยวัตถุดิบ(ด้านข้าง) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แบบส่วนควบคุมการโรยวัตถุดิบ (ด้านบน) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



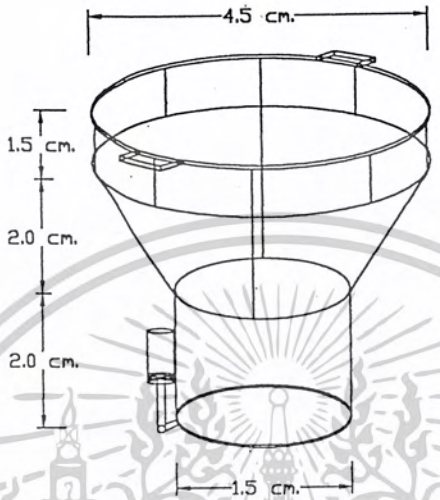
รูปที่ 3.5 แบบส่วนควบคุมการโรยวัตถุดิบ (โดยรวม) มาตราส่วน 1:10

3.3.2 ส่วนควบคุมการชั่งตวง

การทำงานของส่วนควบคุมการชั่งตวง คือเป็นส่วนที่ใช้ในการชั่งน้ำหนักของข้าวสารโดยอาศัยตัวโหลดเซลล์ ที่ติดตั้งอยู่บริเวณขอบของถังทั้ง 2 ด้านเป็นตัวปรับเทียบน้ำหนักทั้ง 2 ค่าให้เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

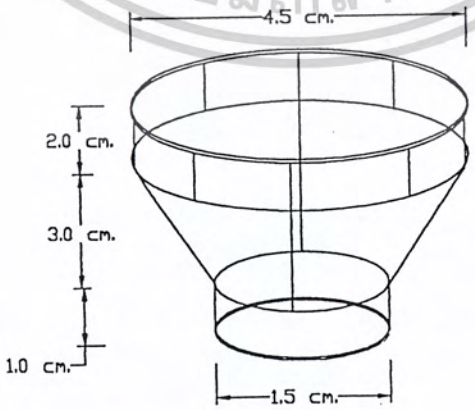
กันโดยอาศัยการเปรียบเทียบน้ำหนักของทั้ง 2 ด้านเพื่อให้ได้ค่าของน้ำหนักที่ถูกต้องและแม่นยำซึ่งมีขนาดและการออกแบบดังรูปที่ 3.6



รูปที่ 3.6 แบบส่วนควบคุมการชั่งตวง (โดยรวม) มาตราส่วน 1:10

3.3.3 ส่วนบรรจุวัตถุดิบ

การทำงานของส่วนบรรจุวัตถุดิบนั้นเป็นเพียงส่วนที่ใช้เป็นทางลำเลียงของข้าวที่ชั่งจากถังที่ 2 เสร็จเรียบร้อยแล้วเพื่อที่จะนำไปบรรจุหีบห่อเมื่อได้น้ำหนักที่ต้องการตามที่เราตั้งค่าไว้แล้วนั้น ซึ่งแสดงแบบของส่วนบรรจุวัตถุดิบดังรูปที่ 3.7

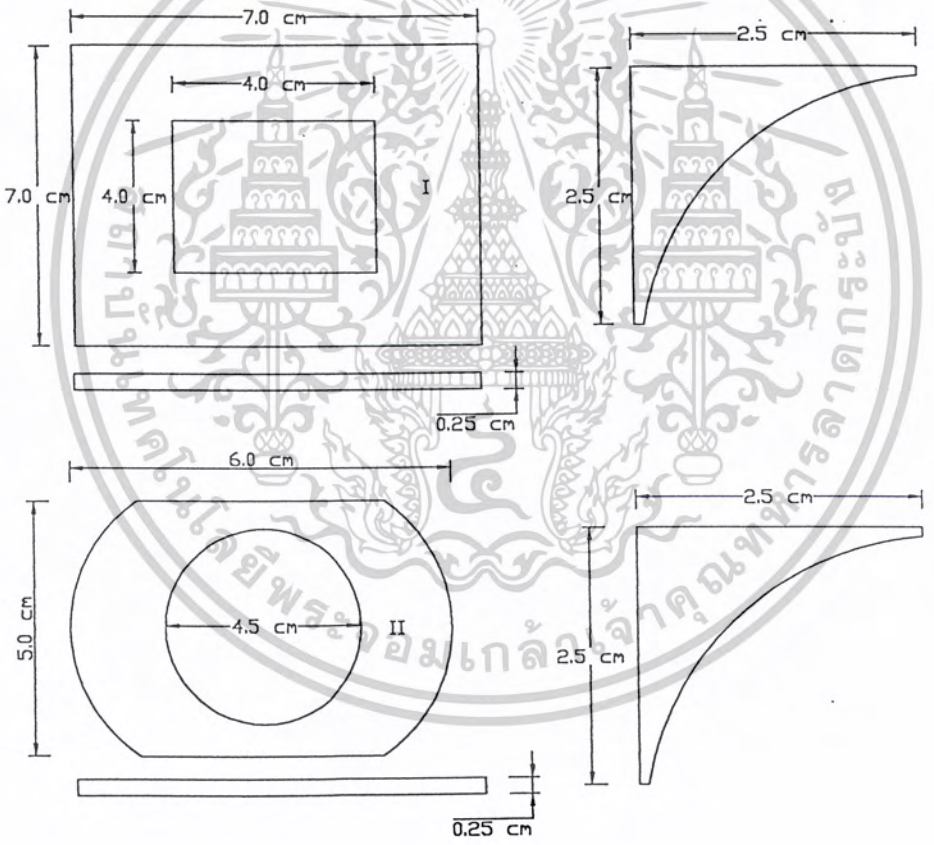


รูปที่ 3.7 แบบส่วนบรรจุวัตถุดิบ(โดยรวม) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

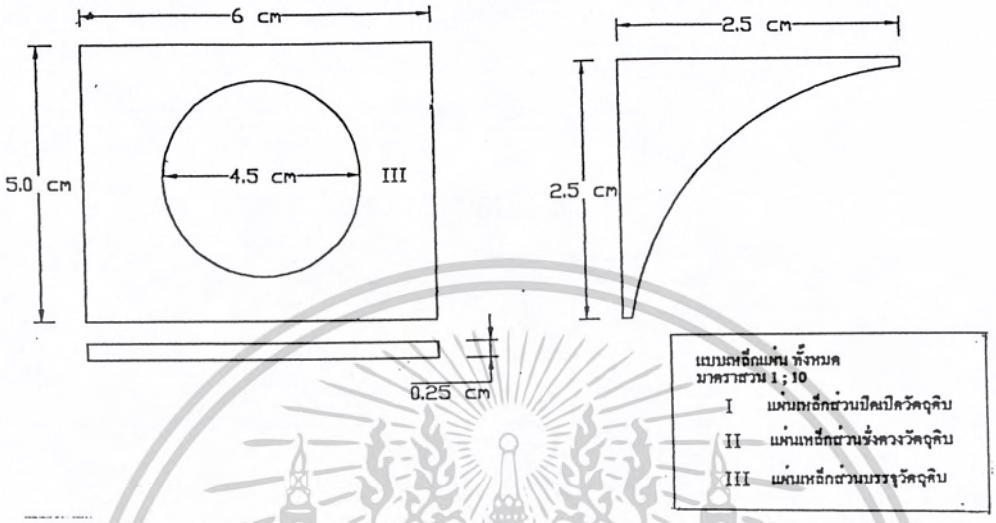
3.3.4 ส่วนโครงเหล็ก

ความสำคัญของส่วนนี้คือเป็นที่ยึดของทั้ง 3 ส่วนซึ่งได้แก่ส่วนควบคุมการโรยของวัตถุดิบ, ส่วนควบคุมการชั่งตวง, ส่ วนบรรจุวัตถุดิบ ซึ่งติดตั้งอยู่ในตำแหน่งดังกล่าวตามลำดับโดยมีแผ่นเหล็ก 3 แผ่นเป็นตัวกั้นเป็นเสมือนกับเป็นตัวแบ่งส่วนแต่ละส่วนออกจากกันแต่จะทำงานสัมพันธ์กันซึ่งแบบของแผ่นเหล็กทั้งหมดแสดงดังรูปที่ 3.8



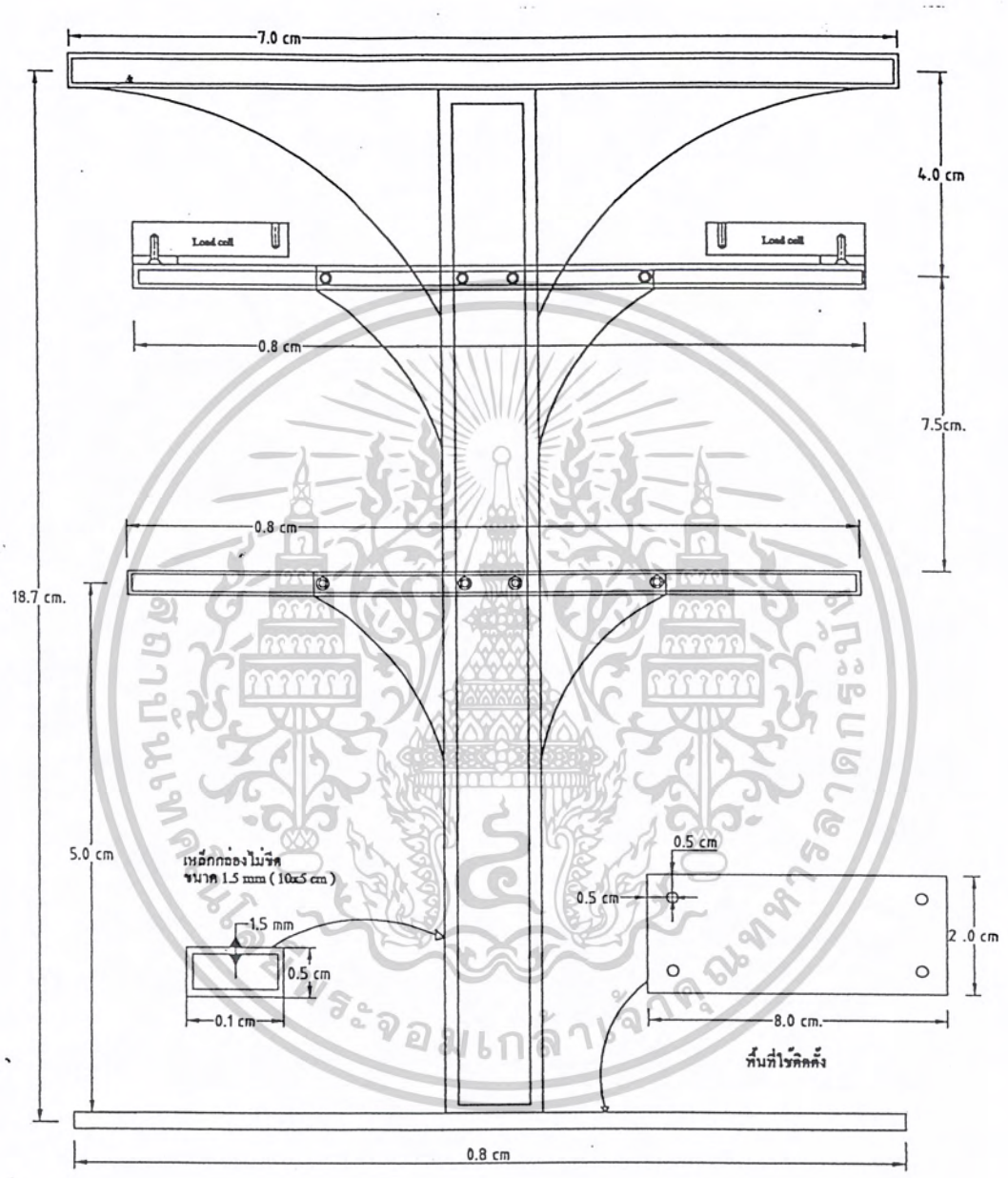
รูปที่ 3.8 แบบเหล็กแผ่นส่วนควบคุมการโรยวัตถุดิบ, ส่วนควบคุมการชั่งตวง, ส่วนควบคุมการบรรจุ (ด้านบน) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



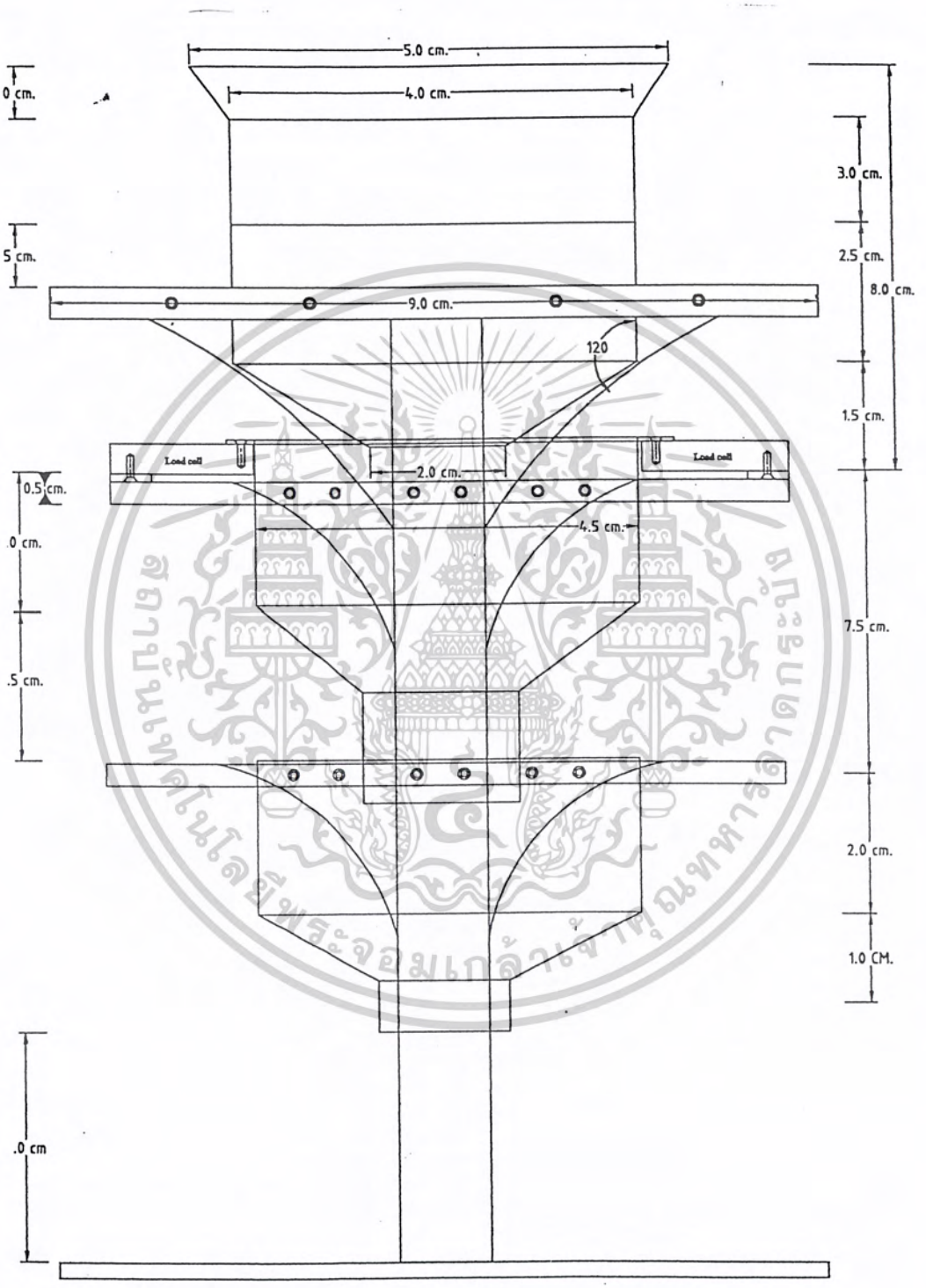
รูปที่ 3.8 (ต่อ) แบบเหล็กแผ่นส่วนควบคุมการโรยวัตถุดิบ, ส่วนควบคุมการชั่งตวง, ส่วนควบคุมการบรรจุ (ด้านบน) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



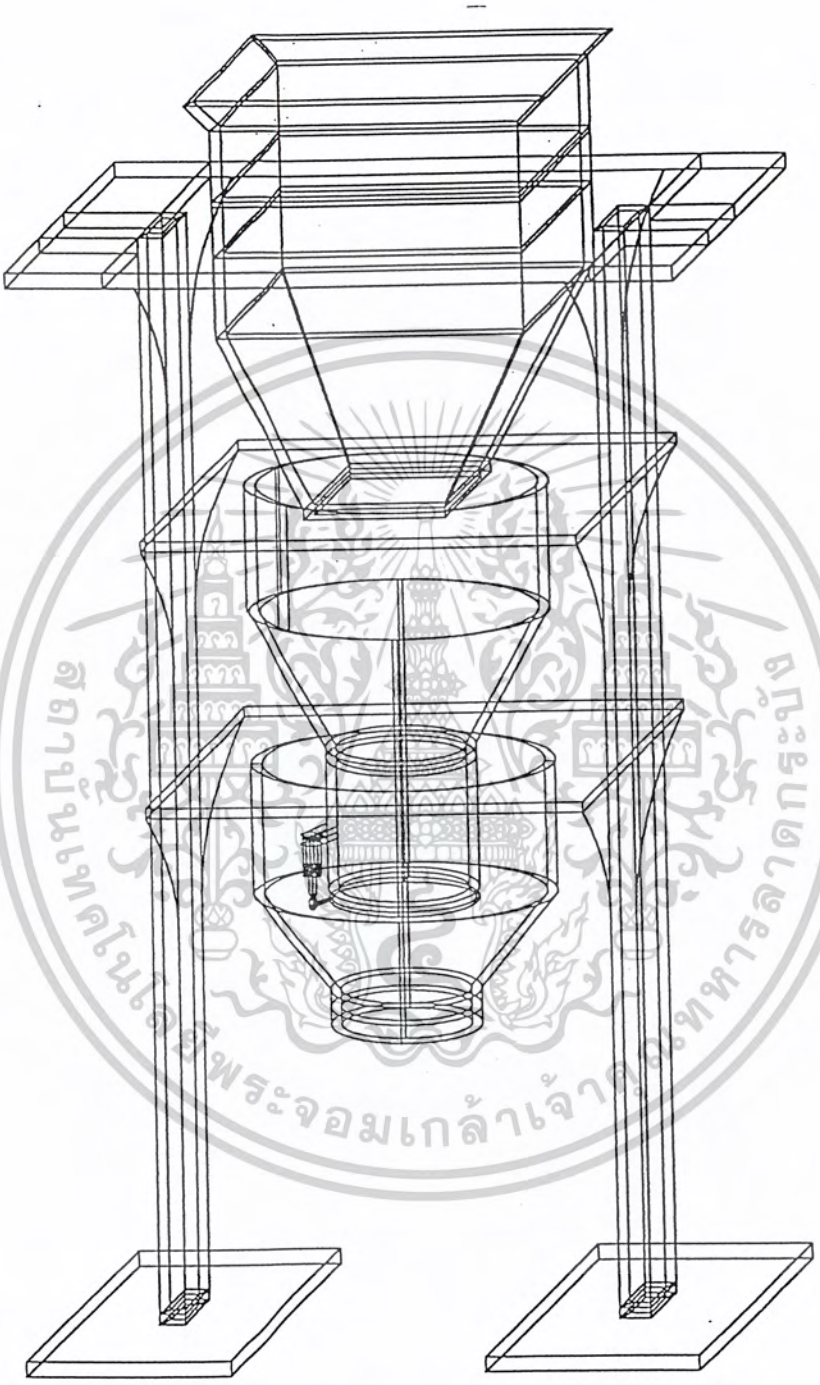
รูปที่ 3.9 แบบเครื่องชั่งอัตโนมัติส่วนโครงเหล็ก (ด้านข้าง) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แบบเครื่องชั่งส่วนโครงเหล็กที่ติดตั้งทั้ง 3 ส่วน (ด้านข้าง) มาตราส่วน 1:10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



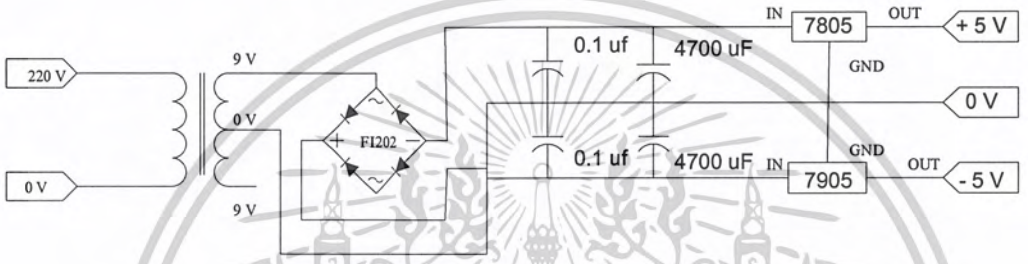
รูปที่ 3.11 แบบโดยรวมของเครื่องชั่งอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบวงจรต่างๆ และการทำงานของวงจร

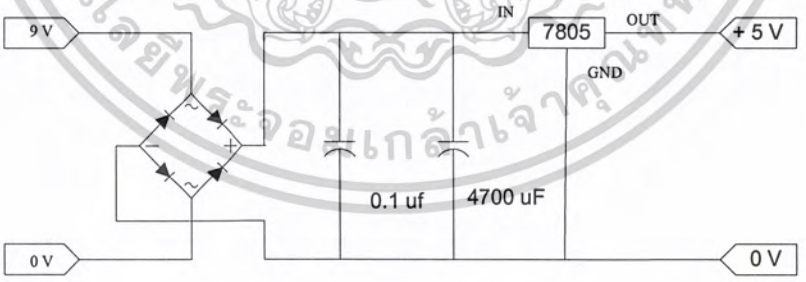
ในเครื่องชั่งอัตโนมัตินั้นประกอบด้วยวงจรต่างๆซึ่งสัมพันธ์กันและสำคัญอยู่ทั้งหมด 4 ภาคด้วยกันคือ วงจรแหล่งจ่ายไฟ วงจรรวมสัญญาณจากโหลดเซลล์ วงจรขับโซลินอยด์วาล์ว วงจรควบคุม ซึ่งจะทำการอธิบายการทำงานของวงจรต่างๆดังนี้

3.4.1 วงจรแหล่งจ่ายไฟ



รูปที่ 3.12 วงจรแหล่งจ่ายไฟ

จากรูปที่ 3.12 การทำงานของวงจรคือเป็นวงจรบริดจ์เรคติไฟเออร์ ซึ่งการทำงานคือจะทำการแปลงไฟฟ้ากระแสสลับจากหม้อแปลงให้เป็นให้เป็นไฟฟ้ากระแสตรง 5 V โดยใช้ IC 7805 และ 7905 เป็นตัวพิกัดแรงดันที่ 5 V

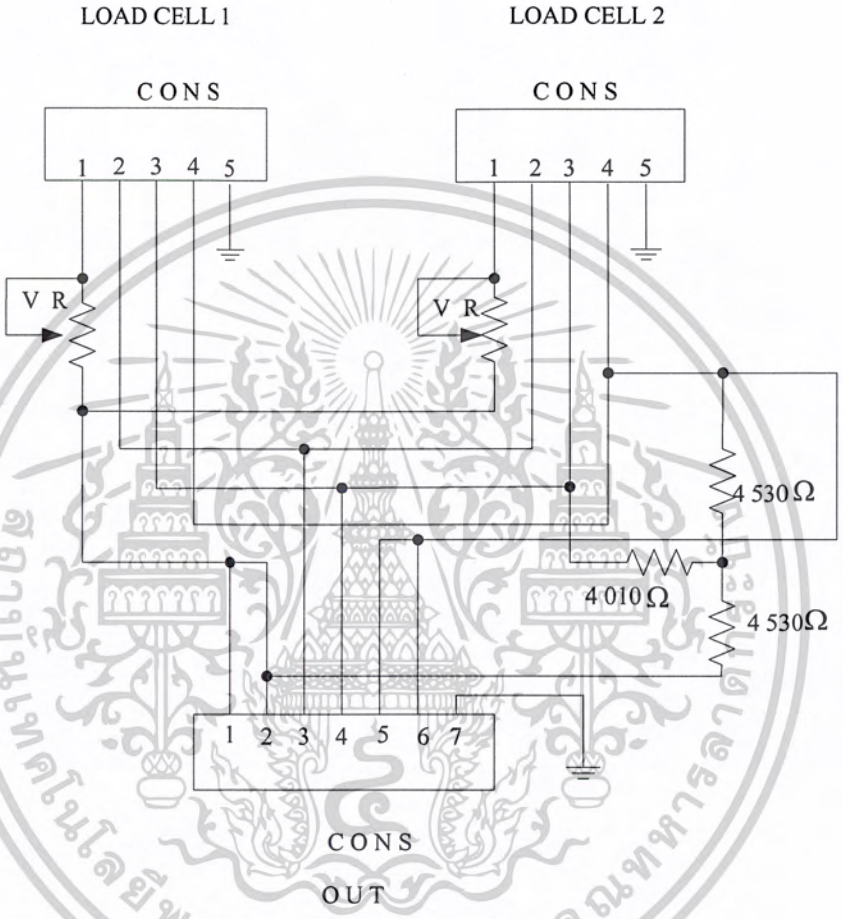


รูปที่ 3.13 วงจรแหล่งจ่ายไฟให้แก่โหลดเซลล์

จากรูปที่3.13 เป็นวงจรแหล่งจ่ายไฟให้แก่ภาคควบคุมโดยจะได้เอาต์พุตออกมาเป็นไฟ +5V ใช้ IC 7805 เป็นตัวควบคุมแรงดันไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 วงจรรวมสัญญาณจากโหลดเซลล์ (Summing)

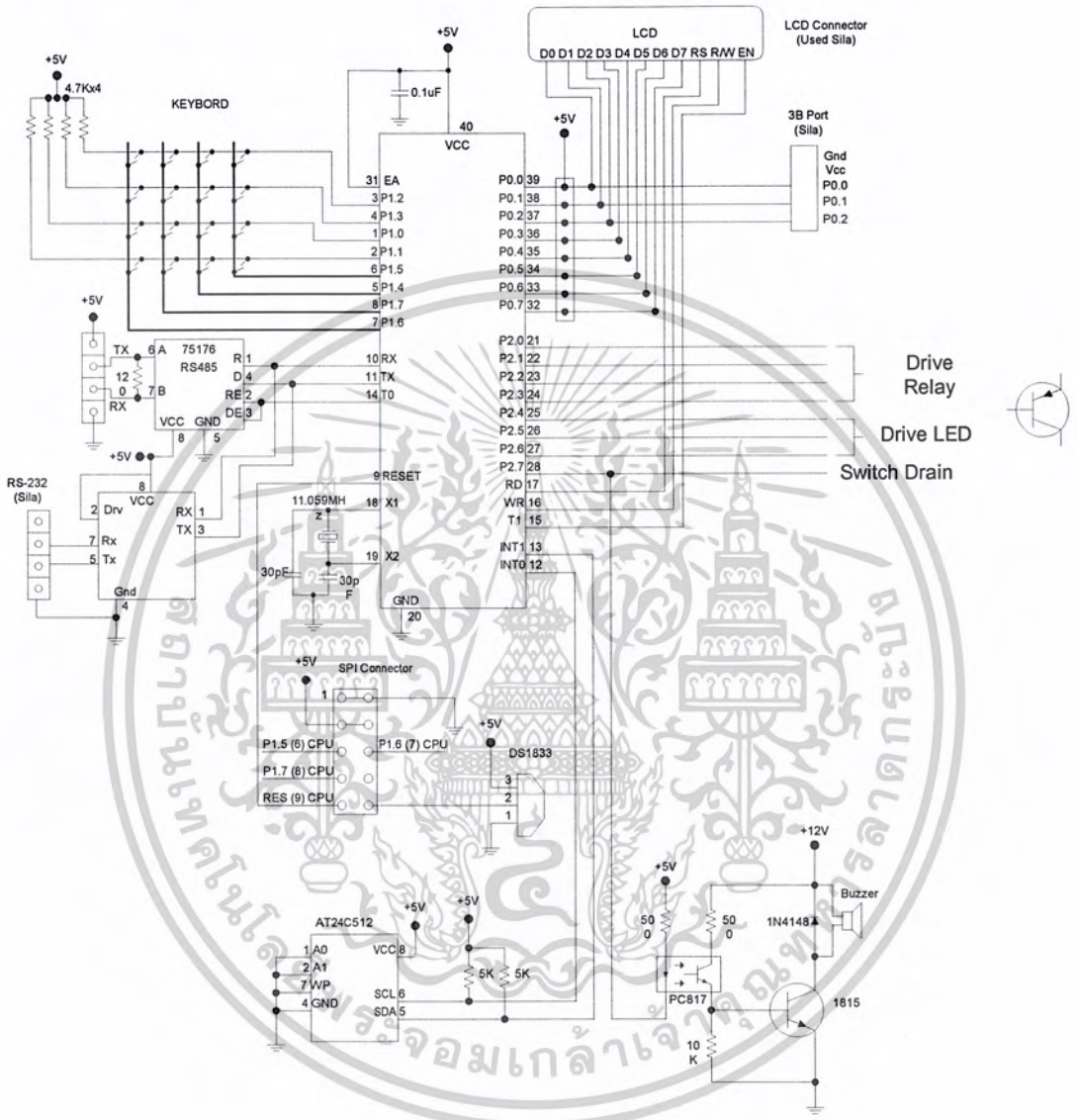


รูปที่ 3.14 วงจรรวมสัญญาณจากโหลดเซลล์

จากรูป การทำงานของวงจรจะทำหน้าที่ในการรวมสัญญาณให้ไหลเข้าสู่ Indicatorโดย อาศัยการปรับค่าของ VR ทั้ง 2 ตัวเพื่อให้ค่าแรงดันนั้นเท่ากัน ซึ่งวงจรนี้ใช้กับโหลดเซลล์ได้ 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 วงจรควบคุม



รูปที่ 3.15 วงจรควบคุม

การทำงานของวงจรควบคุมในที่นี่มี หน้าที่ในการควบคุมการทำงานของกระบอกสูบ ทั้ง 3 ตัว ซึ่งก็คือ ส่วนควบคุมการจ่ายวัตถุดิบ, ส่วนควบคุมการชั่งตวง โดยจะสั่งงานผ่านโซลินอยด์วาล์ว 3/2 ทั้ง 2 ตัว และวาล์ว 5/2 อีก 1 ตัว, วงจรขับรีเลย์ และรวมไปถึงการเชื่อมต่อกับชุดแสดงผลคือ LCD และ 7 SEGMENT เพื่อแสดงค่าน้ำหนักที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

โครงงานนี้คือ เครื่องชั่งบรรจุภัณฑ์อัตโนมัติซึ่งมีวงจรดังนี้ วงจรแหล่งจ่ายไฟ, วงจรขับโซลินอยด์วาลว์, วงจรรวมสัญญาณ (Summing), วงจรแสดงผล และวงจรควบคุมวงจรแต่ละส่วนทำงานสัมพันธ์กันเป็นระบบได้โดยใช้ตัวควบคุมซึ่งใช้ไมโครคอนโทรเลอร์ในการควบคุมในการทดลองได้ทำการจำแนกการทดลองและนำวงจรมาร่วมกันเป็นระบบได้ซึ่งมีผลการทดลองดังนี้

4.1 การทดลองหาค่าอัตราขยายของโหลดเซลล์

น้ำหนักที่วัดได้จากตัวโหลดเซลล์จะอยู่ในรูปของแรงดันไฟฟ้ากระแสตรงซึ่งมีค่าน้อยมากมีหน่วยเป็นมิลลิโวลต์ ซึ่งไม่พอที่จะเข้าวงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลจึงต้องผ่านวงจรขยายแรงดัน(Instrument Amplifier) เพื่อทำหน้าที่ขยายแรงดันให้ได้ 5V เพื่อที่จะเข้าวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล

4.1.1 การทดลองโหลดเซลล์

เริ่มต้นที่ไม่มีน้ำหนักพบว่า แรงดันเริ่มต้นที่โหลดเซลล์ตัวที่หนึ่งมีค่าเท่ากับ 1.9 มิลลิโวลท์ ที่โหลดเซลล์ตัวที่สองแรงดันเริ่มต้นมีค่าเท่ากับ 1.1 มิลลิโวลท์ เมื่อโหลดเซลล์ทั้งสองตัวให้ค่าแรงดันเริ่มต้นไม่เท่ากัน ทำให้การเขียนโปรแกรมควบคุมวาล์วเปิดและปิดของกระบอกสูบทั้งสองตัวที่ถังชั่งมีลักษณะที่ไม่เหมือนกัน ซึ่งโหลดเซลล์แต่ละตัวให้ค่าของน้ำหนักที่ถังชั่งดังนี้

ตารางที่ 4.1 การทดลองโหลดเซลล์ตัวที่หนึ่งที่น้ำหนัก 0-25 กิโลกรัม

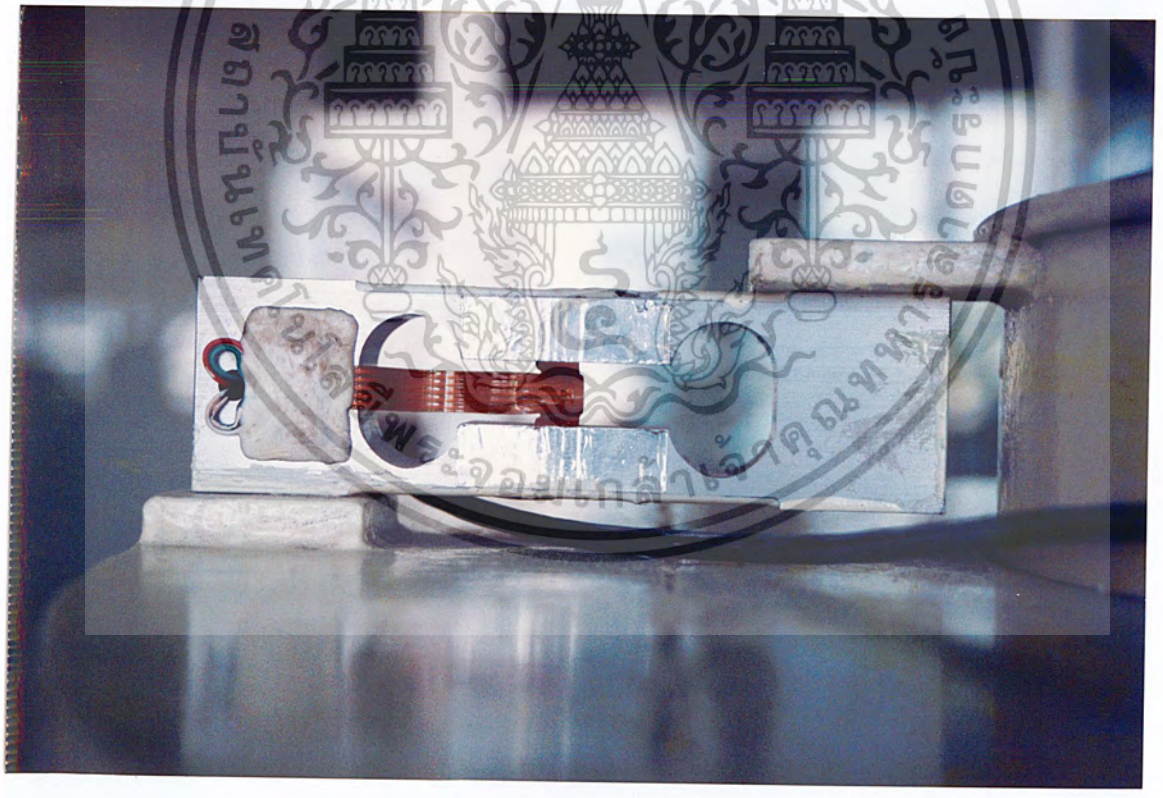
น้ำหนัก (กิโลกรัม)	แรงดัน (มิลลิโวลท์)
0	1.9
5	2.9
10	3.8
15	4.7
20	5.6
25	6.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การทดลองโหลดเซลล์ตัวที่สองที่น้ำหนัก 0-25 กิโลกรัม

น้ำหนัก (กิโลกรัม)	แรงดัน (มิลลิโวลท์)
0	1.1
5	2.2
10	3.4
15	4.5
20	5.6
25	6.7

4.2 การทดลองโหลดเซลล์



รูปที่ 4.1 การทดลองโหลดเซลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการควบคุมกระบอกสูบ

รูปที่ 4.2 การควบคุมกระบอกสูบ

ผลการทดลอง จากการทดลองกระบอกสูบ เมื่อส่งค่าลอจิก 1 ไปยังวงจรขับโซลินอยด์วาล์ว จะทำให้โซลินอยด์วาล์วยังไม่ทำงาน ซึ่งจะส่งผลให้กระบอกสูบยังไม่ทำงาน และเมื่อส่งค่าลอจิก 0 ไปยังวงจรขับโซลินอยด์วาล์ว จะทำให้โซลินอยด์วาล์วทำงาน ซึ่งส่งผลให้กระบอกสูบทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

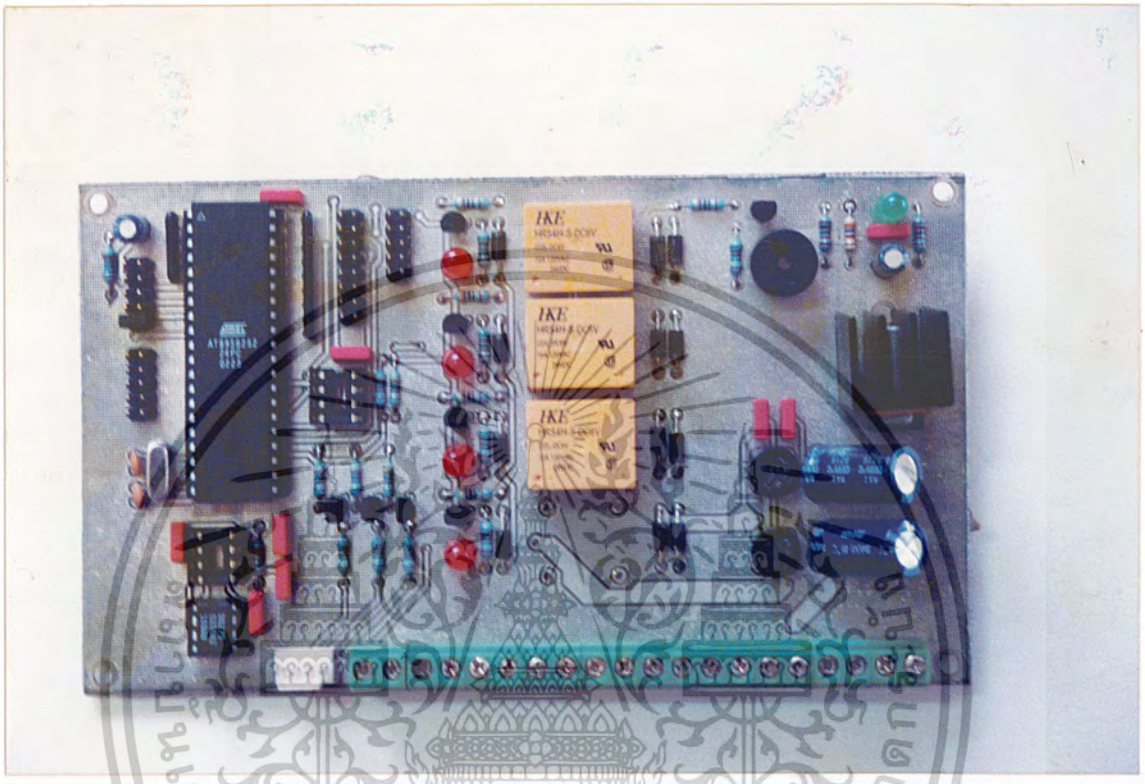
4.4 การทดลองวงจรแสดงผล

รูปที่ 4.3 วงจรแสดงผล

ผลการทดลอง หลังจากการทดลองใช้โปรแกรมรับค่าจากสวิตซ์และสั่งให้แสดงผลมีการทำงานได้ดีแสดงผลออกมาเป็นที่ชัดเจนเป็นที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองวงจรขับรีเลย์

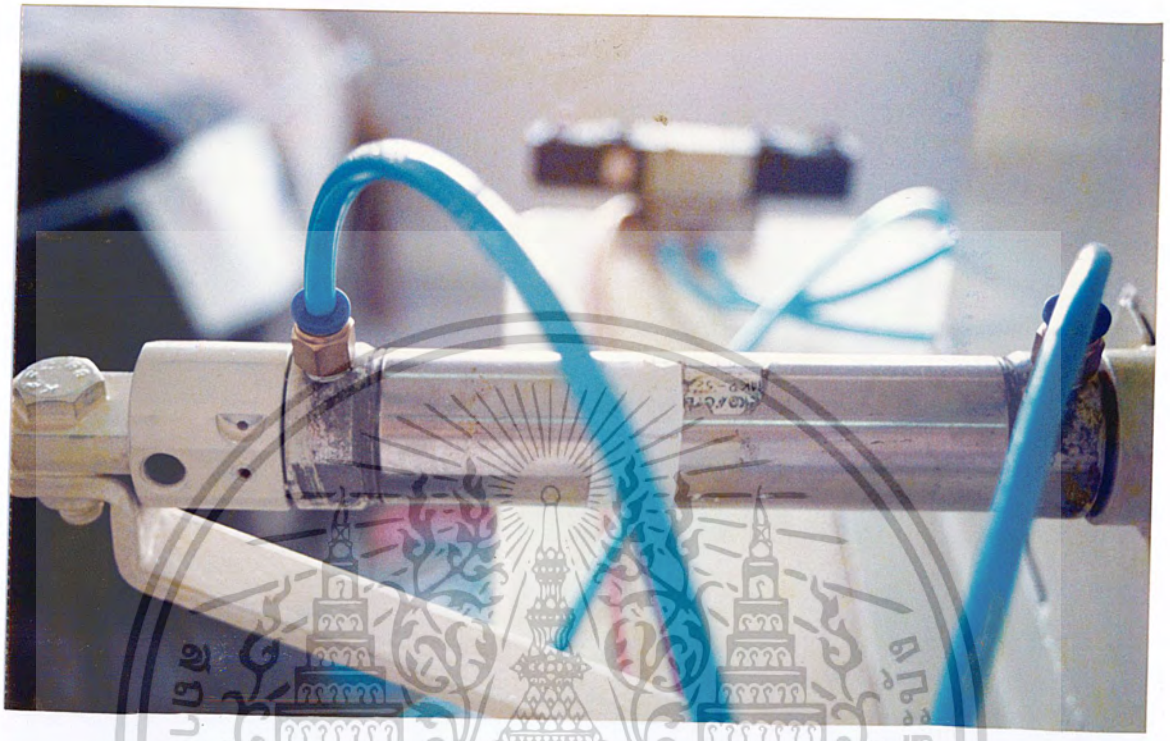


รูปที่ 4.4 วงจรขับรีเลย์

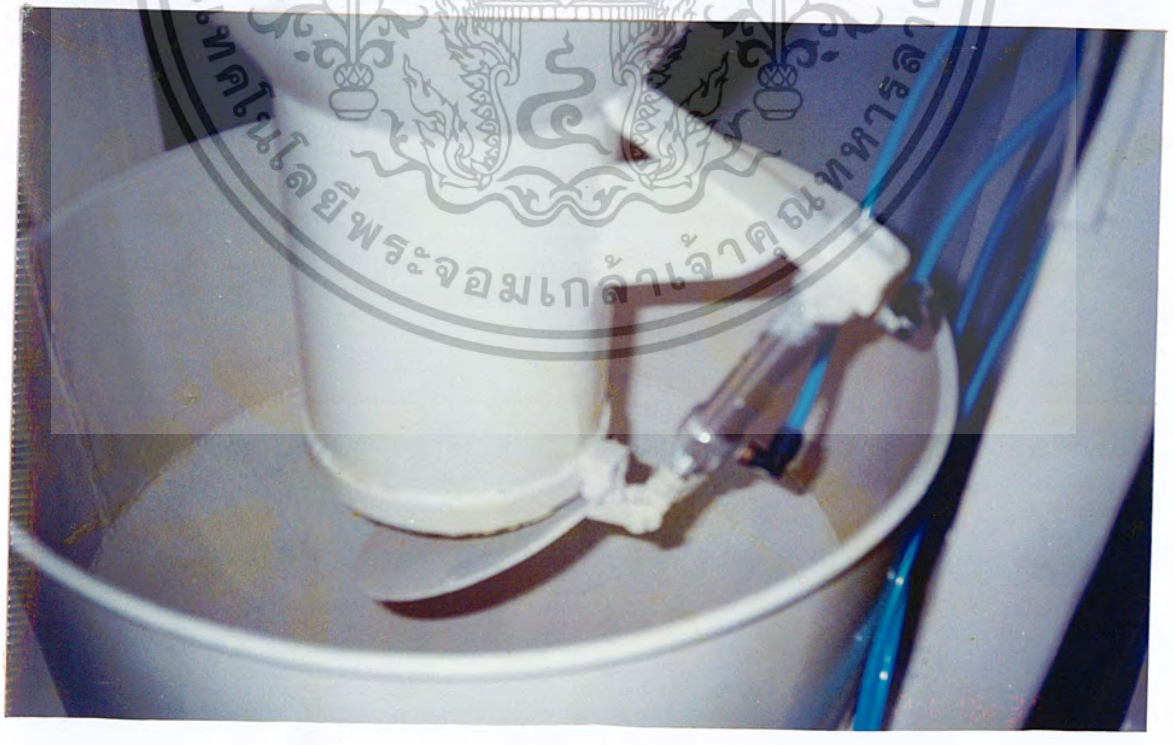
ผลการทดลอง จากการทดลองวงจรขับโซลินอยวาล์ว ซึ่งจะรวมอยู่ในชุดวงจรควบคุม ดังในรูปที่ 4.4 ผลการทดลองปรากฏว่า เมื่อจ่ายแรงดันไฟให้กับโซลินอยด์แต่ละตัวก็สามารถทำงานได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การทดลองกระบอกสูบ



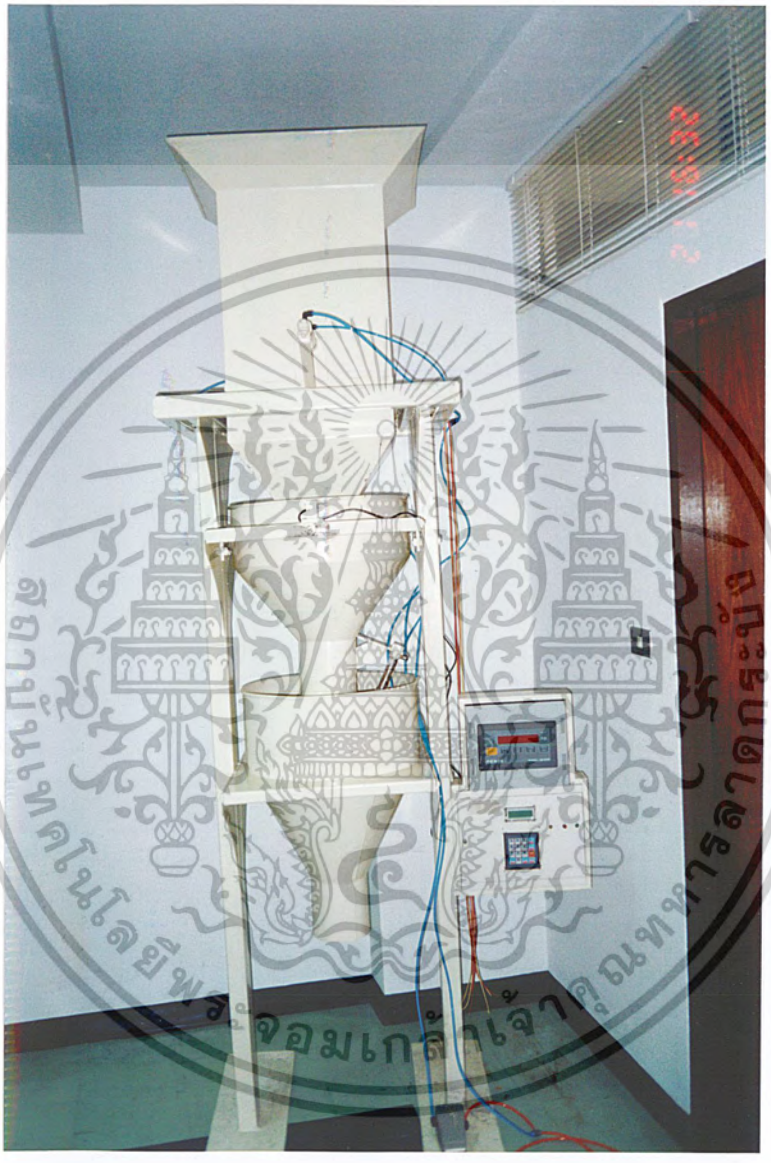
รูปที่ 4.5 กระบอกสูบส่วนควบคุมการโรยวัตถุดิบ



รูปที่ 4.6 กระบอกสูบส่วนควบคุมการชั่งตวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 รูปแสดงเครื่องชั่งอัตโนมัติเมื่อเสร็จสมบูรณ์



รูปที่ 4.7 เครื่องชั่งบรรจุภัณฑ์อัตโนมัติเมื่อเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

5.1 บทสรุป

ในการศึกษาวิธีการออกแบบและสร้างชุดเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ เพื่อสร้างเครื่องชั่งที่สามารถใช้งานได้จริง มีประสิทธิภาพและมีความเที่ยงตรงในการวัด จะสามารถช่วยให้ประหยัดกำลังงานคน และมีความสะดวกสบายมากยิ่งขึ้น

จากการศึกษาและได้ลงมือทำโครงงานนี้ทำให้รู้และเข้าใจรายละเอียด คุณสมบัติและการทำงานของวงจรขับโซลินอยด์วาล์ว วงจรนิวเมติกส์ หลักการของตัวแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิตอล อุปกรณ์โหลดเซลล์ วงจรแหล่งจ่ายไฟ วงจรรวมสัญญาณหรือ Summing และอุปกรณ์ต่างๆที่นำมาใช้ในการทำโครงงาน ในการทำงานนั้นก็จะมีทั้งที่สำเร็จได้ดีและมีทั้งปัญหาต่างๆเกิดขึ้นซึ่งเมื่อเจอปัญหาก็จะต้องหาวิธีแก้ไข ทำให้มีประสบการณ์และความรู้จากการแก้ปัญหาและที่สำคัญคือได้รู้จักถึงความสามัคคีของเพื่อนภายในกลุ่ม การร่วมแรงร่วมใจในการทำงานของสมาชิกในกลุ่ม การทำงานครั้งนี้ได้สำเร็จลุล่วงตามวัตถุประสงค์ที่ตั้งเอาไว้ คือสามารถนำข้อมูลเกี่ยวกับวงจรต่างๆมาทำการคำนวณ ออกแบบและจัดหาซื้ออุปกรณ์ เพื่อนำมาสร้างเป็นวงจรในโครงงานนี้ ขีดความสามารถของโครงงานนี้สามารถตั้งย่านการชั่งน้ำหนักได้ตั้งแต่ 0 – 25 กิโลกรัม และมีการทำงานอย่างต่อเนื่อง สามารถแสดงผลค่าน้ำหนักของวัตถุดิบที่กำลังชั่งได้โดย 7 – Segment และสามารถควบคุมการทำงานโดยอัตโนมัติ

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ปัญหาในการสร้างโครงเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

1) ปัญหาในการจัดหาซื้อกระบอกสูบและโซลินอยด์วาล์ว เนื่องจากกระบอกสูบและโซลินอยด์วาล์วที่ต้องการนั้นมีราคาสูงมาก จึงต้องหาซื้อตามร้านที่ขายอุปกรณ์มือสองซึ่งไม่มีขนาดตามที่ต้องการ

แนวทางแก้ไข จัดหาซื้อกระบอกสูบและโซลินอยด์วาล์วตามร้านขายอุปกรณ์มือสองที่ราคาถูกและใช้งานได้เพื่อเป็นการประหยัดงบประมาณ แล้วนำมาประยุกต์ใช้งานและติดตั้งแบบใหม่ซึ่งสามารถใช้งานได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ปัญหาการติดตั้งกระบอกสูบ เนื่องจากกระบอกสูบที่จัดหาซื้อมาได้มีขนาดใหญ่ แต่พื้นที่การติดตั้งมีจำกัดไม่สามารถติดตั้งตามที่ออกแบบไว้ได้

แนวทางแก้ไข ต้องออกแบบการติดตั้งกระบอกสูบใหม่ เพื่อให้ใช้งานได้ตามต้องการ

3) ปัญหาการติดตั้งโหลดเซลล์ เนื่องจากโหลดเซลล์เป็นอุปกรณ์ที่ไวต่อการสั่นสะเทือนของโครง ทำให้โหลดเซลล์เกิดค่าผิดพลาด ซึ่งจะมีผลต่อการชั่งน้ำหนัก

แนวทางแก้ไข ในการติดตั้งโหลดเซลล์จะทำการศึกษาและปรึกษาจากผู้ที่มีประสบการณ์เกี่ยวกับการใช้งานโหลดเซลล์ จึงทำให้ทราบว่าจะต้องยึดโหลดเซลล์ให้แน่นที่สุด เพื่อไม่ให้โหลดเซลล์เกิดการบิดตัว ทำให้สามารถลดค่าผิดพลาดของโหลดเซลล์ลงได้

4) ปัญหาการทดลองโครงงานทำได้ไม่สะดวก เนื่องจากนี้มีขนาดใหญ่และหนักทำให้การเคลื่อนย้ายลำบาก

แนวทางแก้ไข พยายามทำการทดลองโครงงานในส่วนที่ไม่ต้องเคลื่อนย้ายให้เสร็จก่อน และพยายามเคลื่อนย้ายโครงงานให้น้อยที่สุด

5.2.2 ปัญหาทางด้านการควบคุมการทำงาน

1) ปัญหาข้อมูลที่ใช้ศึกษาในการทำโครงงานหาได้ยาก เนื่องจากข้อมูลที่ใช้นั้นเป็นความรู้ใหม่ แหล่งข้อมูลมีน้อย

แนวทางแก้ไข ศึกษาจากผู้ที่มีความรู้และประสบการณ์ รวมทั้งค้นคว้าหาข้อมูลจากระบบอินเตอร์เน็ต

2) ปัญหาในการชั่งน้ำหนัก เนื่องจากเมื่อโหลดเซลล์ชั่งได้น้ำหนักตามที่กำหนดแล้วจะสั่งให้กระบอกสูบปิด จะมีช่วงที่วัตถุดิบลอยอยู่บนอากาศ(Free Fall)ทำให้เกิดค่าผิดพลาดได้

แนวทางแก้ไข ต้องทำระยะห่างระหว่างถังตัวบนกับตัวล่างที่รองรับ ให้มีระยะห่างกันน้อยลง และพยายามทำให้ให้วัตถุดิบค่อยๆโรยลงทีละน้อยในช่วงที่กำลังจะได้พิกัดน้ำหนักที่ต้องการ เพื่อลดค่าผิดพลาดในการชั่งให้น้อยลง

3) ปัญหาความก้าวหน้าของโครงงานมีน้อย เพราะปัญหาที่ได้กล่าวมาในข้างต้น รวมทั้งมีความรู้ ความชำนาญในการทำโครงงานน้อย ตลอดจนเครื่องมือที่ใช้ทำโครงงานก็มีน้อย จึงทำให้การทำงานล่าช้า

แนวทางแก้ไข พยายามศึกษา ค้นคว้าหาข้อมูลให้มากขึ้น รวมทั้งเข้าปรึกษาอาจารย์ที่ปรึกษาโครงงาน และอาจารย์ท่านอื่นอีกหลายๆท่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนา

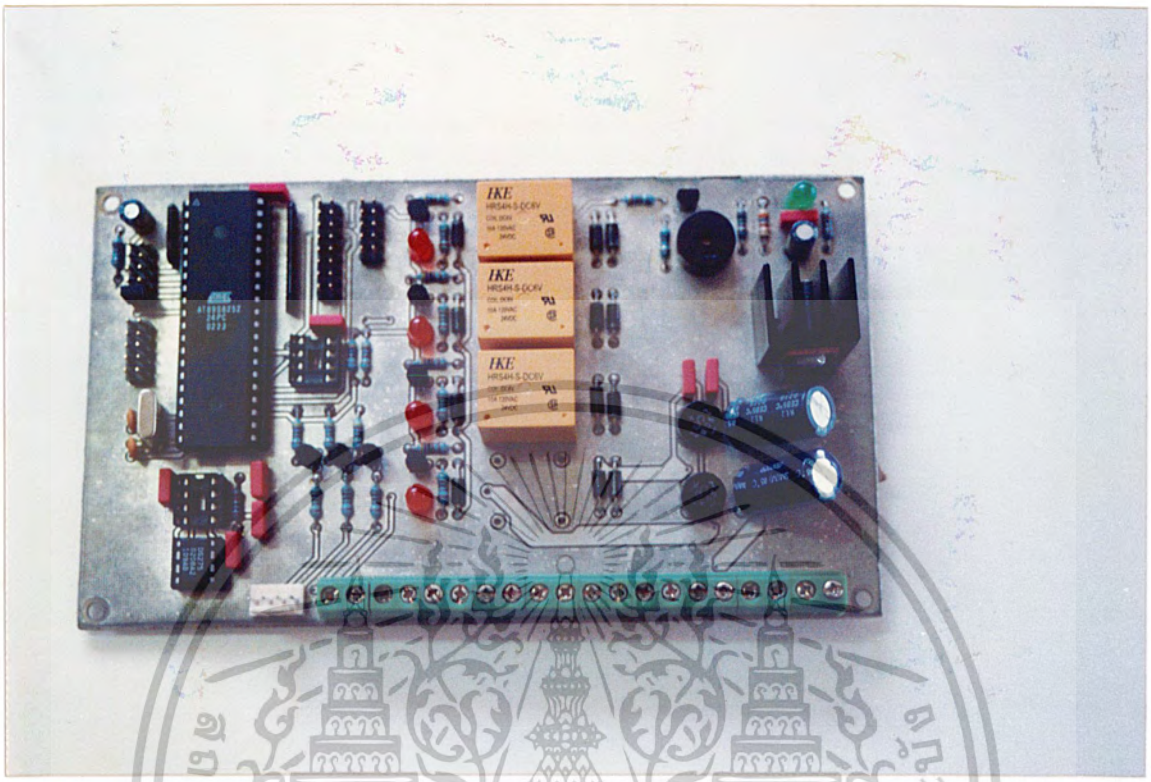
ในการพัฒนาเครื่องชั่งอัตโนมัติ ให้มีความสามารถในการทำงานได้ดีขึ้นกว่าเก่า สามารถพัฒนาโครงงานได้ ดังนี้

1) สามารถพัฒนาโครงงานนี้ให้สามารถชั่งน้ำหนักได้มากกว่า 25 กิโลกรัม
2) สามารถพัฒนาโครงงานนี้ให้ชั่งน้ำหนักของวัตถุดิบได้หลายชนิดกว่านี้
3) สามารถพัฒนาโครงงานนี้ให้สามารถลำเลียงวัตถุดิบที่บรรจุแล้วไปเก็บไว้ที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

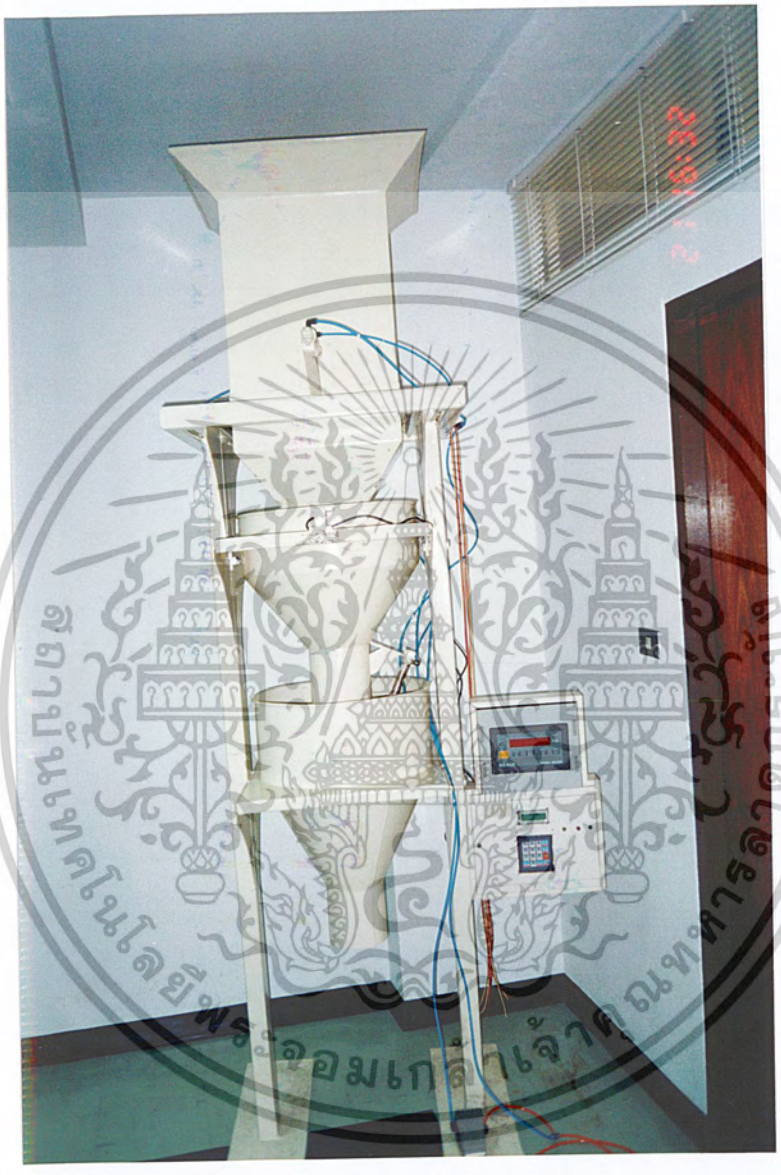
ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 ภาพรวมของแผงวงจรควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



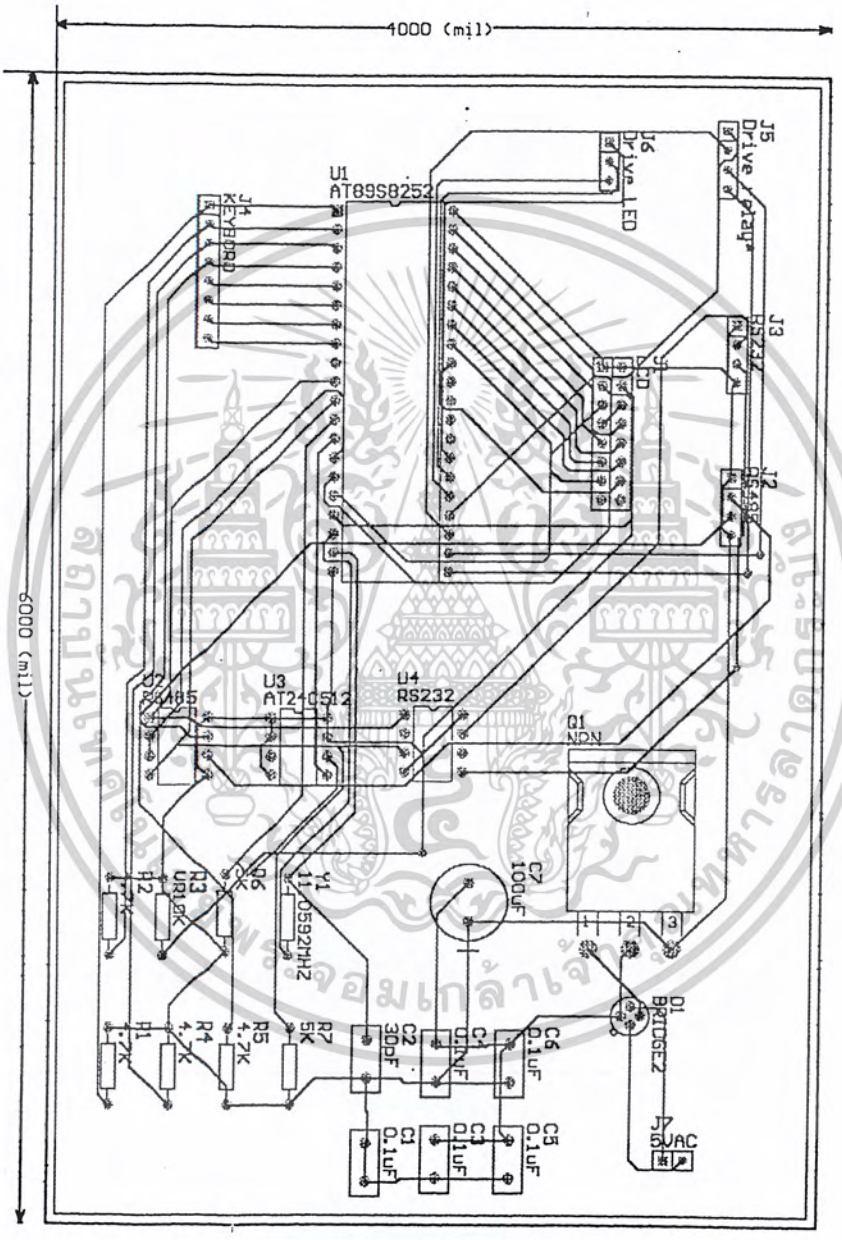
รูปที่ ก.2 รูปสำเร็จของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

วงจร และแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 วงจรควบคุม วงจรขับโซลินอยด์ วงจรภาคแสดงผล วงจรขับรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ผังงาน และโปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.1 แผนผังการทำงานของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************/
/* Weighing Control */
/*********************************/

#include<at89x52.h>   /* Necessary Include file */
#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#include<intrins.h>
#include<math.h>
#include<string.h>

#define TRUE      1    /* Bit definition */
#define FALSE     0

#define k_esc     15   /* Keyboard definition */
#define k_up      14
#define k_down    13
#define k_fnc     12
#define k_ent     11
#define k_dot     10

#define PA        0    /* 8255 port definition */
#define PB        1
#define PC        2
#define PD        3

#define CR        13   /* Line & return */
#define LF        10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sfr     WMCON  =  0x96;     /* WMCON 0f 89s8252 */

sbit   SOL1     =  P2^0;   /* Control solinoil */
sbit   SOL2     =  P2^1;
sbit   SOL3     =  P2^2;
sbit   DRAIN    =  P2^3;

sbit   LAMP1    =  P2^4;   /* Control lamp */
sbit   LAMP2    =  P2^5;
sbit   LAMP3    =  P2^6;

sbit   beep     =  P2^7;     /* Beep sound */
sbit   SW       =  P2^7;     /* Drain switch */

sbit   lcdrs    =  P3^7;     /* LCD COntrol bit */
sbit   lcdrw    =  P3^6;
sbit   lcden    =  P3^5;

sbit   rs       =  P3^4;     /* RS-485 Send/Recive bit */

char   data   ASCBUF[20];    /* General perpose buffer */
char   data   a, b;
float  data   ACTUAL, FINAL, FREEFALL, DRIBBLE;
char   xdata  *p;
unsigned   data   counter, timeout, sec10;
char   data   DTIME, VALVE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*******************/
/* Timer interrupt */
/*******************/

static void timer0_isr (void) interrupt 1 using 1
{
   sec10++;
   TH0 = 0xc4;                  /* high byte */
   TL0 = 0;                     /* reload timer 0 value low byte  */
   TR0 = 1;                     /* start timer 0                  */
}

/*****************************/
/* Delay function        */
/*  input : int loop counter */
/*****************************/

void delay (unsigned int count)
{
   for (; count; count--);
}

/*********************/
/* Write Instruction */
/*********************/

void lcdwi (char cmd)

{
   P0=cmd;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    lcdrs=0;
    lcdrw=0;
    lcden=0;
    lcden=1;
    _nop_ ();
    lcden=0;
    delay (128);
}

/**************/
/* Write Data */
/**************/

void lcdwd (char dta)

{
   P0=dta;
   lcdrs=1;
   lcdrw=0;
   lcden=0;
   _nop_ ();
   lcden=1;
   lcden=0;
   delay (128);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/***************/
/* Initial LCD */
/***************/

void initlcd (void)

{
   lcdwi (0x38);
   lcdwi (0x0c);
   lcdwi (1);
   lcdwi (0x80);
}

/******************/
/* Outtext to LCD */
/******************/

void text (char *msg)

{
   while ((*msg!=0L) && (*msg!=CR))
   {
      lcdwd (*msg);
      msg++;
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**************/
/* Beep right */
/**************/

void rbeep (char count)

{
   for (; count; count--)
   {
      beep=0;
      delay (5000);
      beep=1;
   }
}

/*****************************/
/* Keyboard hit ? function */
/*****************************/

bit kbhit (void)

{
   P1=0x0f;
   if ((P1&0x0f)!=0x0f)
      return (TRUE);
   else
      return (FALSE);
}

/*****************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Read keyboard */
/*****************/

char readkbd (bit n)

{
char   code   scankey[]={0x7f, 0xbf, 0xdf, 0xef};
//char   code   keycode[]={1, 2, 3, 14, 4, 5, 6, 13, 7, 8, 9, 12, 15, 0, 10, 11};
//char   code   keycode[]={10, 0, 11, 12, 1, 2, 3, 13, 4, 5, 6, 14, 7, 8, 9, 15};
char   code   keycode[]={11, 10, 0, 15, 12, 9, 8, 7, 13, 6, 5, 4, 14, 3, 2, 1};
char   data   k_a, k_b;


  while (1)
  {
    for (k_a=0; k_a<4; k_a++)
    {
      P1=scankey[k_a];
      k_b=P1&0x0f;
      if (k_b==0x0f)
        continue;
      else
      {
        delay (100);
        P1=scankey[k_a];
        k_b=P1&0x0f;
        if (k_b==0x0f)
          continue;

        if (k_b==0x07)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            k_b=0;
          if (k_b==0x0b)
            k_b=1;
          if (k_b==0x0d)
            k_b=2;
          if (k_b==0x0e)
            k_b=3;

          k_a*=4;
          k_b+=k_a;
          k_b=keycode[k_b];
          if (n==0)
          {
            rbeep (1);
            while ((P1&0x0f)!=0x0f);   /* Non repeat key */
          }
          return (k_b);
        }
      }
    }
}

/************************************************/
/* Convert text to floating point function      */
/*  input :                                     */
/*    - char count (limit of char)              */
/*  output :                                    */
/*    - error code return float (-1)            */
/*    - successfull return floating point number */
/************************************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float textnum (char count, char add)
{
   char    a, b;

   a=0;
   ASCBUF[0]=0;
   lcdwi (add);

   while (1)
   {
      b=readkbd (0);          /* Waiting for key press */

      if  (b==k_esc)          /* Esc key press */
         break;               /* Breake the loop and return (-1) */

      if  (b==k_ent)          /* Enter key press */
      {

         if  (a!=0)           /* If the key was pressed */
         return (atof (ASCBUF));   /* Return the keyboard value */
        else
         return ((float)-1);       /* The other return error (2) */

         }

      if  (b==k_up)           /* Clear key press */
      {
        if  (a==0)            /* Clear of key  press first */
        {                     /* Put the 0 to buffer */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ASCBUF[0]=0;
            continue;
          }
          a--;
          ASCBUF[a]=0;
          lcdwi (add+a);
          text (" ");
          lcdwi (add+a);
          lcdwi (add);
          text (ASCBUF);
        }

        if (a==count)
          continue;

        if (b<=10)          /* The numeric and dot key press */
        {
          if (b==k_dot)
            ASCBUF[a]='.';          /* put numeric value */
          else
            ASCBUF[a]=b+0x30;          /* put numeric value */
          a++;                  /* and convert to ASCII */
          ASCBUF[a]=0;                  /* both to display to 7 segment */
          lcdwi (add);
          text (ASCBUF);
        }
      }
      return ((float)-1);       /* Not successfull return (-1) */
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*******************************/
/* Get string from serial port */
/* with non-echo               */
/*******************************/

char getstr (char *dbuf)
{
   char   data   ct;

   ct=0;
   timeout=0;
   while (1)              /* Loop for get string from serial port */
   {
      if (RI==1)
      {
         *dbuf=_getkey ();     /* Use getkey function with not echo */
         *dbuf&=0x7f;
         ct++;
         if (*dbuf==LF)        /* End of string by \n */
            break;          /* Break loop */
         dbuf++;            /* Save string to buffer */
         timeout=0;
      }
      timeout++;
      if (timeout==0)
         return (FALSE);
   }
   return (ct);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
   Show indicator error
*/

void error (void)

{
   P2=0xff;       /* Turn off all relay */
   initlcd ();
   text ("Indicator Error."); /* Show indicator error */
   lcdwi (0xc0);
   text (" Press any key. ");
   a=readkbd (0);
}

/*
   Save eeprom function
*/

void save (int add, float  dta)

{
   WMCON|=0x18;    /* Enable eeprom */
   p=add;
   sprintf (ASCBUF, "%0.02f", dta);
   for (a=0; a<15; a++,p++)
   {
      while ((WMCON&0x02)!=2);
      *p=ASCBUF[a];      /* get near point */
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    WMCON=0x02;    /* turn off eeprom */
}

/*
   Read eeprom function
*/

float read (int add)

{
   WMCON|=0x08;   /* Enable eeprom */
   p=add;
   for (a=0; a<15; a++,p++)
      ASCBUF[a]=*p;      /* get setpoint */
   return (atof (ASCBUF));
   WMCON=0x02;   /* turn off eeprom */
}

/*
   Setting mode
*/

void setting (char t)

{
   initlcd ();

   if (t==0)
      text ("- Final weight -"); /* display final */
   if (t==1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   text ("-- Free fall. --");  /* display set n-point */
if  (t==2)
   text ("-Timmer/Dribble-");  /* display Dribblw point */
if  (t==3)
   text (" - Valve open - ");  /* display Dribblw point */
if  (t==4)
   text ("-- Drain time --");  /* display set n-point */

lcdwi (0xc3);
text (": ");
lcdwi (0x0f);
ACTUAL=textnum (7, 0xc5);

if  (t==0)
{
   if  (ACTUAL>=0)       /* Keep set point */
      FINAL=ACTUAL;
   save (0x0000, FINAL);
   sprintf (ASCBUF, "%0.02f", FINAL);
}

if  (t==1)              /* Keep Free fall time */
{
   if  (ACTUAL>=0)
      FREEFALL=ACTUAL;
   save (0x0010, FREEFALL);
   sprintf (ASCBUF, "%0.02f", FREEFALL);
}

if  (t==2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      if (ACTUAL>=0)        /* Keep set n-point */
        DRIBBLE=ACTUAL;
      save (0x0020, DRIBBLE);
      sprintf (ASCBUF, "%0.02f", DRIBBLE);
    }

    if (t==3)               /* Keep Free fall time */
    {
      if (ACTUAL>=0 && (ACTUAL<=3))
        VALVE=(char)ACTUAL;
      save (0x0030, (float)VALVE);
      sprintf (ASCBUF, "%d", (int)VALVE);
    }

    if (t==4)               /* Keep drain time */
    {
      if ((ACTUAL>=0) && (ACTUAL<=30))
        DTIME=(char)ACTUAL;
      save (0x0040, (float)DTIME);
      sprintf (ASCBUF, "%d", (int)DTIME);
    }
    lcdwi (0xc5);
    text (ASCBUF);
    sec10=0;
    while (sec10<90);
  }

  /*
    Running mode
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*/

void run (void)

{
   initlcd ();        /* show function */
   lcdwi (0x0f);
   lcdwi (0x80);
   text (" Enter P-Number ");
   lcdwi (0xc0);
   text ("Select(0-20) ");
   lcdwi (0xf0);

   while (1)
   {
      a=(char)textnum (2, 0xcd);  /* Enter program number */

      if  (a<0)      /* Esc key presss return */
         return;

      if  ((a>=0) && (a<=20))    /* Program entry 1 up to 20 */
      {
         counter=0x50*(int)a;    /* Set index program */
         initlcd ();
         sprintf (ASCBUF, "P-Number : %d ", (int)a);
         text (ASCBUF);      /* Show program number to LCD */
         sec10=0;
         while (sec10<60);
         break;
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else
    {
      lcdwi (0xcd);  /* Incorrect put new program */
      text ("   ");
    }
  }

  if (a!=0)
  {
    FINAL=read (counter);        /* Read old program */
    FREEFALL=read (counter+0x0010);
    DRIBBLE=read (counter+0x0020);
    VALVE=(char)read (counter+0x0030);
    DTIME=(float)read (counter+0x0040);
  }

  counter=0;
  initlcd ();      /* Show text in running mode */
  sprintf (ASCBUF, "FINAL:%0.02fkg ", FINAL);
  text (ASCBUF);
  sprintf (ASCBUF, "COUNT:%d ", counter);
  lcdwi (0xc0);
  text (ASCBUF);

  b=0;          /* Initial ready lamp */
  LAMP1=0;

//  rs=0;           /* Recive the response */

  rs=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    sec10=0;
    while (sec10<20);  /* Set zero */
    rs=1;



    if (RI==1)       /* Clear SBUF */
      a=getstr (ASCBUF);

//  if (a==FALSE)    /* Weighing indicator not exist */
//  {
//    error ();
//    return;       /* Return to main */
//  }

   FINAL-=FREEFALL;

   while (1)
   {
     if (b==0)      /* First step ready to run */
     {
       if (SW==0)   /* Drain key press */
       {
         delay (1000);      /* debouce key */
         if (SW==1)
           continue;

         DRAIN=LAMP1=0;    /* Turn on drain and lmap1 */
         for (b=0; b<DTIME; b++)
         {
           sec10=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while (sec10<60);
}

DRAIN=1;  /* turn off drain */

sec10=0;
while (sec10<120);   /* Waite for drain gate close 1.5 sec */
rs=0;
sec10=0;
while (sec10<20);   /* Set zero */
rs=1;

initlcd ();
sprintf (ASCBUF, "POINT:%0.2fkg ", FINAL+FREEFALL);
text (ASCBUF);
sprintf (ASCBUF, "COUNT:%d ", counter);
lcdwi (0xc0);
text (ASCBUF);
b=1;        /* Set to step 1 */

P2=0xff;
if (VALVE==3)
    SOL1=SOL2=SOL3=LAMP3=0; /* Turn on sol1,sol2,sol3 & lamp3 */
if (VALVE==2)
    SOL2=SOL3=LAMP3=0; /* Turn on sol2,sol3 & lamp3 */
if (VALVE==1)
    SOL3=LAMP3=0; /* Turn on sol3 & lamp3 */

if (DRIBBLE>FINAL)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            timeout=(unsigned)DRIBBLE;
            sec10=0;
            while (sec10<=timeout);

            SOL1=SOL2=1;

//            P2=0xff;  /* Turn on sol1,sol2,sol3 & lamp3 */
            timeout=90;      /* Stable checking */
            sec10=0;
            while (sec10<=timeout);
          }

          if (RI==1)
            a=getstr (ASCBUF);
          ACTUAL=0;
        }
      }

      if (RI==1)
      {
        a=getstr (ASCBUF);
        if (a==FALSE)    /* Weighing indicator not exist */
        {
          error ();
          return;    /* Return to main */
        }

// for GENIX-2000

//        ACTUAL=atof (ASCBUF);    /* for GX-2000 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//          ACTUAL*=0.01;


// for A&D
        if (a==11)    /* for A&D Series */
        {
//          ASCBUF[14]=0;  /* Fill null to end */
          ACTUAL=atof(ASCBUF+1);        /* Convert to float */
        }

    }

    if (b==1)     /* Step 1 check full fall */
    {
      if ((DRIBBLE<FINAL) && (ACTUAL>=FINAL/2))
      {
        SOL1=1;
        continue;
      }

      if ((FINAL-ACTUAL)<=DRIBBLE)  /* Dribble flow */
      {
        P2=0xff;
        LAMP2=SOL3=0;     /* set output to low fall */
        b=2;          /* set to next step */
        if (RI==1)
          a=getstr (ASCBUF);
        continue;
      }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (b==2)     /* Step 2 check final fall */
        {
          if (ACTUAL>=FINAL)
          {
            P2=0xff;        /* ACTUAL=FINAL trun off all */
            LAMP1=0;        /* exept lamp1 */
            counter++;
            b=0;            /* back to ready for repeat step */
            if (RI==1)
              a=getstr (ASCBUF);
            continue;
          }
        }

        if (kbhit()==TRUE)    /* If ESC key press */
        {
          a=readkbd (1);
          if (a==k_esc)
          {
            P2=0xff;
            FINAL+=FREEFALL;
            a=readkbd (0);
            return;    /* Return to main */
          }
        }
      }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
   Program function
   can be programed up to 20
*/

void program (void)

{
   initlcd ();        /* show function */
   lcdwi (0x0f);
   lcdwi (0x80);
   text (" Enter P-Number ");
   lcdwi (0xc0);
   text ("Select(1-20) ");
   lcdwi (0xf0);

   while (1)
   {
      a=(char)textnum (2, 0xcd);  /* Enter program number */

      if (a<0)      /* Esc key presss return */
         return;

      if ((a>=1) && (a<=20))     /* Program entry 1 up to 20 */

      {
         counter=0x50*(int)a;   /* Set index program */
         initlcd ();
         sprintf (ASCBUF, "P-Number : %d ", (int)a);
         text (ASCBUF);     /* Show program number to LCD */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      break;
    }
    else
    {
      lcdwi (0xcd);  /* Incorrect put new program */
      text ("   ");
    }
}

FINAL=read (counter);          /* Read old program */
FREEFALL=read (counter+0x0010);
DRIBBLE=read (counter+0x0020);
VALVE=(char)read (counter+0x0030);
DTIME=(float)read (counter+0x0040);

sprintf (ASCBUF, "Final : %0.02f", FINAL);
lcdwi (0xc0);
text (ASCBUF);     /* Show the first line FINAL value */
lcdwi (0xc8);
lcdwi (0x0f);      /* Trun on cursor */

b=0;          /* Set sub index */
ACTUAL=-1;
while (1)
{
  if (kbhit ()==FALSE)  /* Checking keyboard hit */
    continue;

  a=readkbd (1);     /* Read key with non repeat mode */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (a==k_esc)     /* Esckey press return to main */
{
   a=readkbd (0);
   break;
}

if ((a==k_up) && (b!=0))  /* Up key press */
{
   a=readkbd (0);
   b--;
}

if ((a==k_down) && (b!=4))     /* Down key press */
{
   a=readkbd (0);
   b++;
}

if ((a>=0) && (a<=9))     /* Numeric key press */
{
   text ("       ");        /* Get value from USER */
   ACTUAL=textnum (7, 0xc8);
}

switch (b)
{
   case   0:        /* Show final value to LCD */
      if (ACTUAL>0)
         FINAL=ACTUAL;
      sprintf (ASCBUF, "Final : %0.02f   ", FINAL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

    case  1:        /* Show freefall value to LCD */
      if (ACTUAL>0)
        FREEFALL=ACTUAL;
      sprintf (ASCBUF, "Free. : %0.02f  ", FREEFALL);
      break;

    case  2:        /* Show dribble value to LCD */
      if (ACTUAL>0)
        DRIBBLE=ACTUAL;
      sprintf (ASCBUF, "Dribb : %0.02f  ", DRIBBLE);
      break;

    case  3:        /* Show VALVE open value to LCD */
      if (ACTUAL>0)
        VALVE=(char)ACTUAL;
      sprintf (ASCBUF, "Valve : %d      ", (int)VALVE);
      break;

    case  4:        /* Show Drain time value to LCD */
      if (ACTUAL>0)
        DTIME=(char)ACTUAL;
      sprintf (ASCBUF, "Dtime : %d      ", (int)DTIME);
      break;

  }
  lcdwi (0xc0);
  text (ASCBUF);
  lcdwi (0xc8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACTUAL=-1;
    }

    save (counter+0x0000, FINAL);      /* Save all value to EEPROM */
    save (counter+0x0010, FREEFALL);
    save (counter+0x0020, DRIBBLE);
    save (counter+0x0030, (float)VALVE);
    save (counter+0x0040, (float)DTIME);
}

/*
   IO Test function
*/

void iotest (void)

{
   initlcd ();
   text ("Weighing Control");
   lcdwi (0xc0);
   text ("--- I/O Test ---"); /* display set point */

   while (1)
   {
      P2=0xff;
      a=readkbd (0);
      switch (a)
      {
         case   k_esc:  /* Esc key press */
            return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case   1:    /* 1 key on solinoi 1 */
          SOL1=0;
          break;
        case   2:    /* 2 key on solinoi 2 */
          SOL2=0;
          break;
        case   3:    /* 3 key on solinoi 3 */
          SOL3=0;
          break;
        case   4:    /* 4 key on solinoi 4 */
          DRAIN=0;
          break;

        case   5:    /* 5 key on lamp 1  */
          LAMP1=0;
          break;
        case   6:    /* 6 key on lamp 2  */
          LAMP2=0;
          break;
        case   7:    /* 7 key on lamp 3  */
          LAMP3=0;
          break;

       defult:
          continue;
    }
    sec10=0;
    while (sec10<150);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

/*
   Function mode
*/

void func (void)

{
   initlcd ();        /* show function */
   lcdwi (0x0f);
   lcdwi (0xc0);
   text ("  Select (1-9)");
   lcdwi (0x80);
   text (" Function : ");

   while (1)
   {
      a=readkbd (0);
      switch (a)
      {
         case   1:    /* Running mode */
            run ();
            return;

         case   2:    /* Final weight setting */
            setting (0);
            return;

         case   3:    /* Free fall setting */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setting (1);
            return;

          case   4:    /* Dribble point setting */
            setting (2);
            return;

          case   5:    /* Middle point setting */
            setting (3);
            return;

          case   6:    /* Drain time setting */
            setting (4);
            return;

          case   7:    /* Show last time conter */
            program ();
            return;

//        case   8:    /* Test indicator */
//          showweight ();
//          return;

          case   9:    /* Input/Output test */
            iotest ();
            return;

          case   k_esc:    /* Esc key press return */
            return;
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}

/****************/
/* main program */
/****************/

void main (void)

{
   SCON=0x50;              /* Init serial port */
   TMOD=0x21;              /* 11.0592 MHz - 9600 buad */
   TH1=0xfd;
   TR1=1;
   TI=1;

   TH0=0xc4;               /* Timer interrupt 1/60 sec */
   TL0=0;
   TR0=1;
   ET0=1;
   EA=1;                   /* Enable all interrupt */

   lcdrs=lcdrw=lcden=0;    /* LCD set */
   delay (512);
   initlcd ();   /* LCD startup */

   FINAL=read (0x0000);
   FREEFALL=read (0x0010);
   DRIBBLE=read (0x0020);
   VALVE=(char)read (0x0030);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DTIME=(float)read (0x0040);

    while (1)
    {

/*      initlcd ();
      while (1)
      {
        a=readkbd (0);
        initlcd ();
        sprintf (ASCBUF, "%d", (int)a);
        text (ASCBUF);
      }  */


      P2=0xff;          /* Off all relay */
      lcdwi (0x80);
      text ("Weighing Control");
      lcdwi (0xc0);
      text ("  Version 2.0  ");

      if (SW==0)        /* Check drain switch */
      {
        delay (1000);
        if (SW==1)
          continue;

        DRAIN=0;          /* Turn on drain output */
        lcdwi (0xc0);
        text ("---- Drain -----"); /* Drain time  */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for (b=0; b<DTIME; b++)    /* Delay drain time */
        {
          sec10=0;
          while (sec10<60);
        }
        continue;
      }

      if (kbhit ()==FALSE)
        continue;

      a=readkbd (0);

      if (a!=k_fnc)
        continue;

      func ();      /* Function mode */
      initlcd ();
      text ("Weighing Control");
      lcdwi (0xc0);
      text ("  Version 2.0   ");
    }
}
```

รูปที่ ค.2 โปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน
เครื่องชั่งบรรจุภัณฑ์อัตโนมัติ

กล่าวนำ

คู่มือการใช้งานนี้จะกล่าวถึง การใช้งานของเครื่องชั่งบรรจุภัณฑ์อัตโนมัติซึ่งประกอบด้วย การเริ่มต้นใช้งาน การตั้งค่าน้ำหนัก และการใช้งานตัวอินดิเคเตอร์ ดังจะกล่าวต่อไปนี้

ขั้นตอนการใช้งานและการตั้งค่าน้ำหนัก

1. เสียบปลั๊กเพื่อจ่ายไฟเข้าสู่ระบบ (ไฟฟ้า 220 Volt AC)

2. กด **FUNC +7** เพื่อทำการเข้าสู่การตั้งค่านำหนักซึ่งสามารถกำหนดค่าน้ำหนักได้ 20 ค่า

3. แสดงขั้นตอนการเซตค่าน้ำหนักที่ 5 Kg ไว้ที่ช่องที่ 1

3.1 กด **FUNC** + 7 ตามด้วย 1 แล้วกด **ENTER**

3.2 ปรากฏข้อความ **FINAL** ให้เราทำการใส่ค่าน้ำหนักที่ต้องการสมมุติใส่ค่า 5 Kg แล้วกด **ENTER**

3.3 ปรากฏข้อความ **FREE FALL** เป็นการกำหนดค่าเพื่อชดเชยค่าของวัตถุดิบขณะลอยอยู่ในอากาศ ให้เราใส่ค่า 0.07 แล้วกด **ENTER**

3.4 ปรากฏข้อความ **DRIBB** ให้เราใส่ค่าตั้งระยะเวลาการทำงานของช่องจ่ายวัตถุดิบช่องใหญ่

3.5 ปรากฏข้อความ **VALVE** ถ้าใส่เลข 1 จะทำการเปิดช่องโรยคือช่องเล็กทางขวามือ ถ้าใส่เลข 2 จะเป็นการเปิดทั้ง 2 ช่อง คือทั้งช่องซ้ายและขวา

3.6 ปรากฏข้อความ **DTIME** ให้เราใส่ระยะเวลาการทำงานของเครื่องให้ใส่เลข 3 คือให้หน่วง 3 วินาที

4. เมื่อกระทำการตั้งค่าเสร็จแล้วให้กดปุ่ม **CLEAR**

5. เมื่อต้องการเริ่มทำงานในโปรแกรมที่ตั้งไว้ให้ทำการกด **FUNCTION 1** ตามด้วยช่องที่ตั้งไว้แล้วกด **ENTER**

6. ทำการกดสวิตซ์เท้าเหยียบเพื่อให้วัตถุดิบไหลลงสู่ส่วนบรรจุ และเครื่องจะทำงานอย่างต่อเนื่อง

7. ถ้าจะทำการชั่งค่าน้ำหนักในค่าอื่น ๆ ให้กระทำตามขั้นตอนที่กล่าวมาข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งและการใช้งานจอแสดงน้ำหนัก GENIX GX – 2000

ด้านหน้าของเครื่อง มีปุ่มกดอยู่ 5 ปุ่ม โดยแต่ละปุ่มทำงานดังนี้

ZERO	เซ็ตน้ำหนักเท่ากับ 0
TARE	หักน้ำหนักภาชนะ
G/N	กดเลือกดูน้ำหนักรวม (GROSS) หรือ น้ำหนักสุทธิ (NET)
PRINT	พิมพ์น้ำหนักไปยังเครื่องพิมพ์
MENU	กดเลือกใช้โหมด: น้ำหนักโค๊ด CODE ---- SP (setpoint) ---- น้ำหนัก

ไฟแสดง

----0----	แสดงน้ำหนักกึ่งกลางศูนย์
NET	แสดงน้ำหนักสุทธิ
GROSS	แสดงน้ำหนักรวม

ขั้วต่อ J1

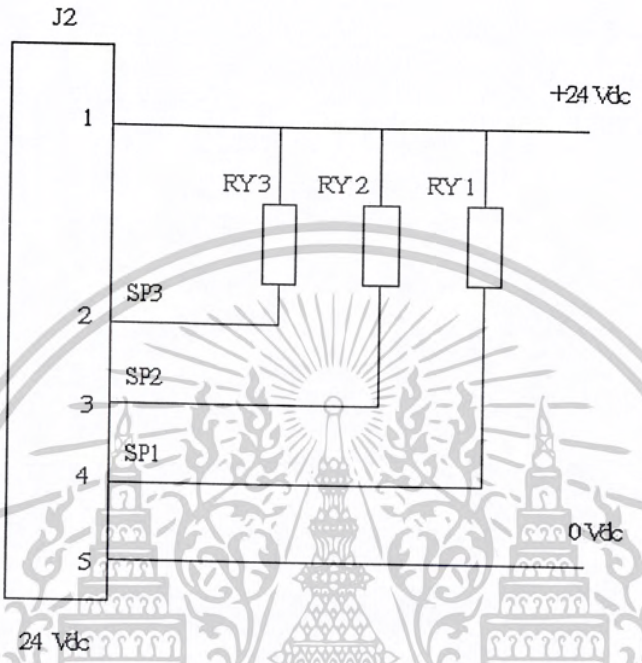
สำหรับต่อเข้ากับโหลดเซลล์โดยมีการจัดขั้วสัญญาณดังนี้

ขา 1	=	Shield
ขา 2	=	EX+
ขา 3	=	EX-
ขา 4	=	SIG+
ขา 5	=	SIG-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้วต่อ J2

Setpoint output



รูปที่ ง.1 วงจรการต่อขั้ว J2

ขั้วต่อ J3

Serial port (RS – 232)

ขา 1 = GND

ขา 2 = Tx (ส่ง)

ขั้วต่อ J4

AC Power supply

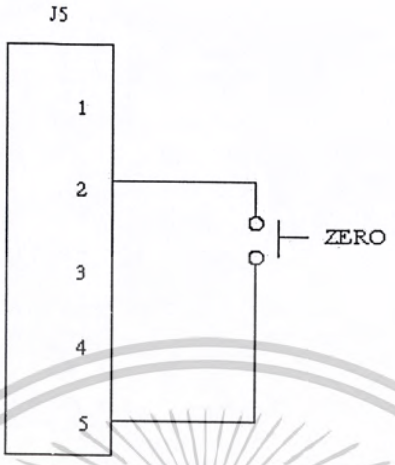
ขา 1 = 9 Vac

ขา 2 = 0 Vac

ขา 3 = 9 Vac

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้วต่อ J5



รูปที่ ง.2 วงจรการต่อขั้ว J5

EXT ZERO

ขา 1 = 5 Vdc
ขา 2 = ZERO
ขา 3 = NC
ขา 4 = NC
ขา 5 = GND

การติดตั้ง

การตั้งค่าขนาดพิกัดน้ำหนัก , จำนวนทศนิยม , ค่าอ่านละเอียด

1. ต่อขั้วโหลดเซลล์และเลือกค่า R3, R4 และ R10 ให้ถูกต้องตามจำนวนโหลดเซลล์

ตารางที่ ง-1 แสดงค่าความต้านทานของโหลดเซลล์แต่ละตัว

จำนวนโหลดเซลล์	R 3	R4	R10
1	40 k	40 k	40 k
2	20 k	20 k	20 k
3	12 k	12 k	12 k
4	10 k	10 k	10 k

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เปิดสวิตซ์จ่ายไฟเข้าเครื่อง เครื่องจะทดสอบหน้าจอโดยการนับ 0 – 9 แล้วแสดงน้ำหนัก 0

3. รอประมาณ 1 นาที

4. กดปุ่มเมนู ให้หน้าจอแสดง CODE

กดปุ่ม ENT หน้าจอแสดง 00000 ป้อนรหัสผ่าน 123 โดยใช้ปุ่ม ^ ในการเพิ่มตัวเลข และปุ่ม < ในการเลื่อนหลักตัวเลข

กดปุ่ม ENT หน้าจอแสดง CAL

กดปุ่ม MENU หน้าจอแสดง CONF

กดปุ่ม ENT หน้าจอแสดง CAP (พิกัดน้ำหนักสูงสุด)

กดปุ่ม ENT หน้าจอแสดงค่าพิกัดน้ำหนัก(รวมทศนิยม) เช่น100.00 แสดง 10000 ทำการเปลี่ยนแปลงแก้ไขโดยใช้ปุ่ม ^ ในการเพิ่มตัวเลข และปุ่ม < ในการเลื่อนหลักตัวเลข

กดปุ่ม ENT SAVE ค่า หน้าจอแสดง CAP

กดปุ่ม MENU หน้าจอแสดง DP (หลักทศนิยม)

กดปุ่ม ENT หน้าจอแสดงจุดทศนิยม แก้ไขได้โดยใช้ปุ่ม <

กดปุ่ม ENT SAVE ค่าหน้าจอแสดง dp

กดปุ่ม MENU หน้าจอแสดง div (ค่าอ่านละเอียด)

กดปุ่ม ENT หน้าจอแสดงค่าอ่านละเอียด ใช้ปุ่ม < แก้ไขตามต้องการ

กดปุ่ม ENT SAVE ค่า div หน้าจอแสดง div

กดปุ่ม MENU หน้าจอแสดงน้ำหนักปกติ

การปรับสัญญาณโหลดเซลล์

กดปุ่ม MENU ให้หน้าจอแสดง CODE

กดปุ่ม ENT ให้หน้าจอแสดง 00000 ป้อนระหัสผ่าน 123 โดยใช้ปุ่ม ^ ในการเพิ่มตัวเลข และใช้ < ในการเลื่อนหลักตัวเลข

กดปุ่ม ENT หน้าจอแสดง CAL

กดปุ่ม ENT หน้าจอแสดง ZERO

กดปุ่ม MENU หน้าจอแสดง SPAN F

กดปุ่ม MENU หน้าจอแสดง ATD

กดปุ่ม ENT จะปรากฏตัวเลขจำนวนหนึ่ง หรือ 00 ให้ปรับค่า VR1 ให้ตัวเลขมีค่าประมาณ -18000 (ไม่มีน้ำหนักได้วางอยู่บนเครื่องชั่ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบค่าอ่านละเอียด

วางน้ำหนักเท่ากับค่าอ่านละเอียดบนแท่นชั่ง ตังเลข ATD จะต้องเพิ่มขึ้นมากกว่า 2 ขึ้นไปจึงจะใช้งานได้ถ้าไม่ได้ให้เพิ่มค่า R3, R4 และ R10

กดปุ่ม ENT หน้าจอแสดง ATD

การปรับน้ำหนักศูนย์

กดปุ่ม MENU จนหน้าจอแสดง ZERO

น้ำหนักบนแท่นชั่งเท่ากับ 0 กดปุ่ม ENT หน้าจอแสดงค่าตัวเลขจำนวนหนึ่งประมาณ 5 วินาที หน้าจอจะกลับมาแสดง ZERO

การปรับน้ำหนัก

กดปุ่ม MENU จนหน้าจอแสดง SPAN วางน้ำหนักบนแท่นชั่ง เช่น 100 ก.ก

กดปุ่ม ENT หน้าจอแสดง 00000 ให้ป้อนค่าน้ำหนักเท่ากับที่อยู่บนแทนชั่ง (โดยใช้ปุ่ม ^ และ <) ให้หน้าจอแสดง 01000 (หน้าจอไม่แสดงทศนิยม) แล้วกดปุ่ม ENT เครื่องแสดงค่าตัวเลขจำนวนหนึ่ง ประมาณ 5 วินาที หน้าจอจะกลับมาแสดง SPAN

กดปุ่ม MENU จนหน้าจอแสดง END กดปุ่ม ENT หน้าจอจะแสดง CODE

กดปุ่ม MENU จนหน้าจอแสดงค่าน้ำหนัก

------ จบขั้นตอน -----

การตั้งค่าน้ำหนัก SP(OPTION)

1. กดปุ่ม MENU ให้หน้าจอแสดง SP

2. กดปุ่ม ENT หน้าจอแสดงค่าน้ำหนักเดิม ที่ตั้งไว้ (1 xxxxx สำหรับ setpoint 1, 2 xxxxx สำหรับ setpoint 2, 3 xxxxx สำหรับ setpoint 3)

3. ใช้ปุ่ม < ^ เปลี่ยนแปลงแก้ไขตัวเลข

4. เมื่อได้ค่าที่ต้องการ กดปุ่ม ENT หน้าจอแสดง SP กดปุ่ม MENU ให้หน้าจอแสดงน้ำหนักปกติ

------ จบขั้นตอน ------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการอุปกรณ์

1. วงจรควบคุม

1.1 คาปาซิเตอร์

เบอร์	จำนวน/ ตัว
10 uF/ 16 V	1
100 uF/ 16 V	2
1000 uF/16 V	2
0.1 uF	5
30 pF	2

ตารางที่ จ.1คาปาซิเตอร์

2. ความต้านทาน

ค่าความต้านทาน	จำนวน/ตัว
120 Ω	2
500 Ω	8
1 kΩ	1
4.7 kΩ	10
10 kΩ	12

ตารางที่ จ.2ความต้านทาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อื่นๆ ประกอบด้วย

AT 89S8252	1 ตัว
คริสตอล 11.0592 MHz	1 ตัว
ไอซี เบอร์ 75176	1 ตัว
ไอซี RS 232	1 ตัว
ไดโอด 1N4001	4 ตัว
ไดโอด 1N4004	8 ตัว
LED	8 ตัว
LCD โมดูล 16 X 2	1 ตัว
ทรานซิสเตอร์ BC 327	1 ตัว
คอนเนคเตอร์ 16 พิน	1 ตัว
คอนเนคเตอร์ 10 พิน	1 ตัว
LM 317T	1 ตัว
บัสเซอร์ 120 Ω	1 ตัว
บัสเซอร์ 1.2 kΩ	1 ตัว
บัสเซอร์ 5 kΩ	1 ตัว

ตารางที่ จ.3 อุปกรณ์อื่นๆ

อุปกรณ์ทางระบบนิวแมติกส์ ประกอบด้วย

รายการ	จำนวน
กระบอกสูบ 2 ทาง	3 ตัว
โซลินอยด์วาล์ว 5/2	2 ตัว
วาล์วควบคุม 3/2	1 ตัว
โหลดเซลล์ 30 กก.	2 ตัว
สวิทช์แบบใช้เท้าเหยียบ	1 ตัว

ตารางที่ จ.4 อุปกรณ์นิวแมติกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อื่นๆ ประกอบด้วย

AT 89S8252	1 ตัว
คริสตอล 11.0592 MHz	1 ตัว
ไอซี เบอร์ 75176	1 ตัว
ไอซี RS 232	1 ตัว
ไดโอด 1N4001	4 ตัว
ไดโอด 1N4004	8 ตัว
LED	8 ตัว
LCD โมดูล 16 X 2	1 ตัว
ทรานซิสเตอร์ BC 327	1 ตัว
คอนเนคเตอร์ 16 พิน	1 ตัว
คอนเนคเตอร์ 10 พิน	1 ตัว
LM 317T	1 ตัว
บัสเซอร์ 120 Ω	1 ตัว
บัสเซอร์ 1.2 kΩ	1 ตัว
บัสเซอร์ 5 kΩ	1 ตัว

ตารางที่ จ.3 อุปกรณ์อื่นๆ

4. อุปกรณ์ทางระบบนิวแมติกส์ ประกอบด้วย

รายการ	จำนวน
กระบอกสูบ 2 ทาง	3 ตัว
โซลินอยด์วาล์ว 5/2	2 ตัว
วาล์วควบคุม 3/2	1 ตัว
โหลดเซลล์ 30 กก.	2 ตัว
ลิมิตสวิทช์แบบใช้เท้าเหยียบ	1 ตัว

ตารางที่ จ.4 อุปกรณ์นิวแมติกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

รายละเอียด และคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- **Compatible with MCS-51™ Products**
- **8K Bytes of In-System Reprogrammable Downloadable Flash Memory**
  - **SPI Serial Interface for Program Downloading**
  - **Endurance: 1,000 Write/Erase Cycles**
- **2K Bytes EEPROM**
  - **Endurance: 100,000 Write/Erase Cycles**
- **4V to 6V Operating Range**
- **Fully Static Operation: 0 Hz to 24 MHz**
- **Three-level Program Memory Lock**
- **256 x 8-bit Internal RAM**
- **32 Programmable I/O Lines**
- **Three 16-bit Timer/Counters**
- **Nine Interrupt Sources**
- **Programmable UART Serial Channel**
- **SPI Serial Interface**
- **Low-power Idle and Power-down Modes**
- **Interrupt Recovery From Power-down**
- **Programmable Watchdog Timer**
- **Dual Data Pointer**
- **Power-off Flag**

Description

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

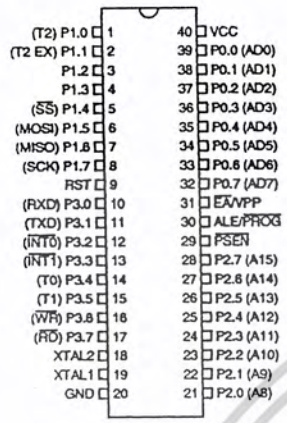
The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator,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

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

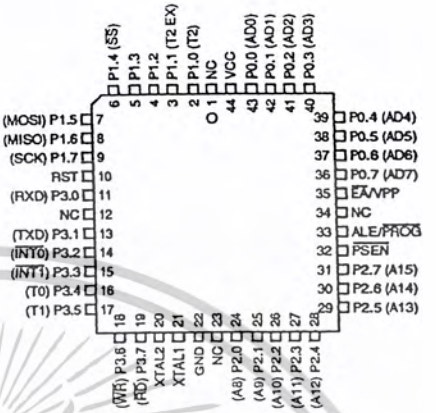
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

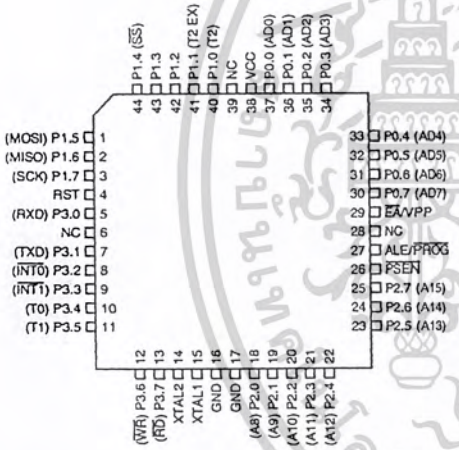
PDIP



PLCC



PQFP/TQFP



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

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

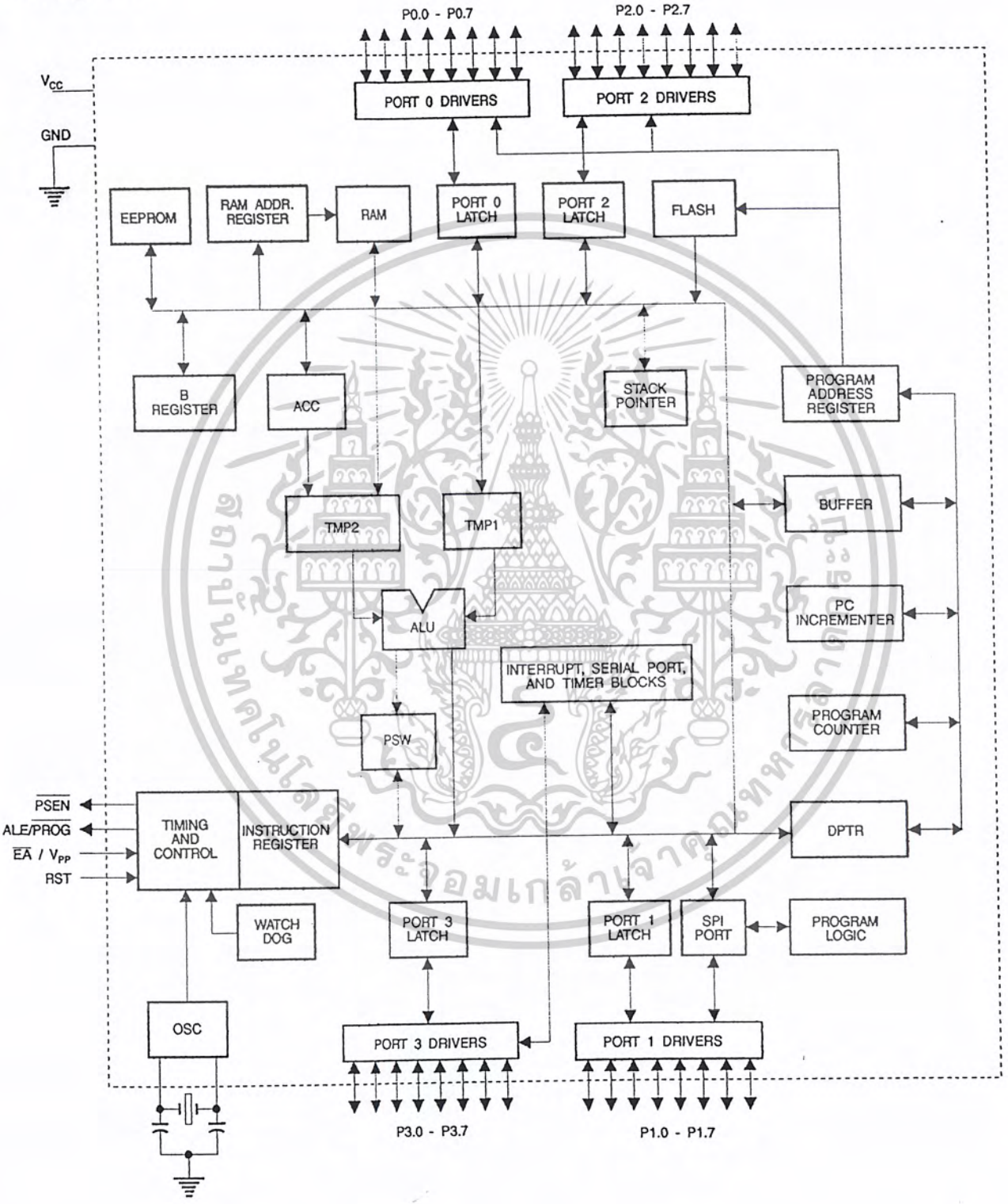
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Pin Description

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

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

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

Port 2

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

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

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

Port 3

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

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

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

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

RST

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

ALE/$\overline{PROG}$

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

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

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

$\overline{PSEN}$

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

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

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external pro-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

gram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

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

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S8252 SFR Map and Reset Values

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

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

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

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

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

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	7	6	5	4	3	2	1	0
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H — Reset Value = 0000 0010B

Bit	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
	7	6	5	4	3	2	1	0

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4. SPCR—SPI Control Register

SPCR Address = D5H — Reset Value = 0000 01XXB

Bit	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
	7	6	5	4	3	2	1	0

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

Table 5. SPSR – SPI Status Register

SPSR Address = AAH — Reset Value = 00XX XXXXB

Bit	SPIF	WCOL	–	–	–	–	–	–
	7	6	5	4	3	2	1	0

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

Table 6. SPDR – SPI Data Register

SPDR Address = 86H — Reset Value = unchanged

Bit	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	7	6	5	4	3	2	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Memory – EEPROM and RAM

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

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

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

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

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

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

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

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

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

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

Programmable Watchdog Timer

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

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

Table 7. Watchdog Timer Period Selection

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

Timer 0 and 1

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

Timer 2

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

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

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

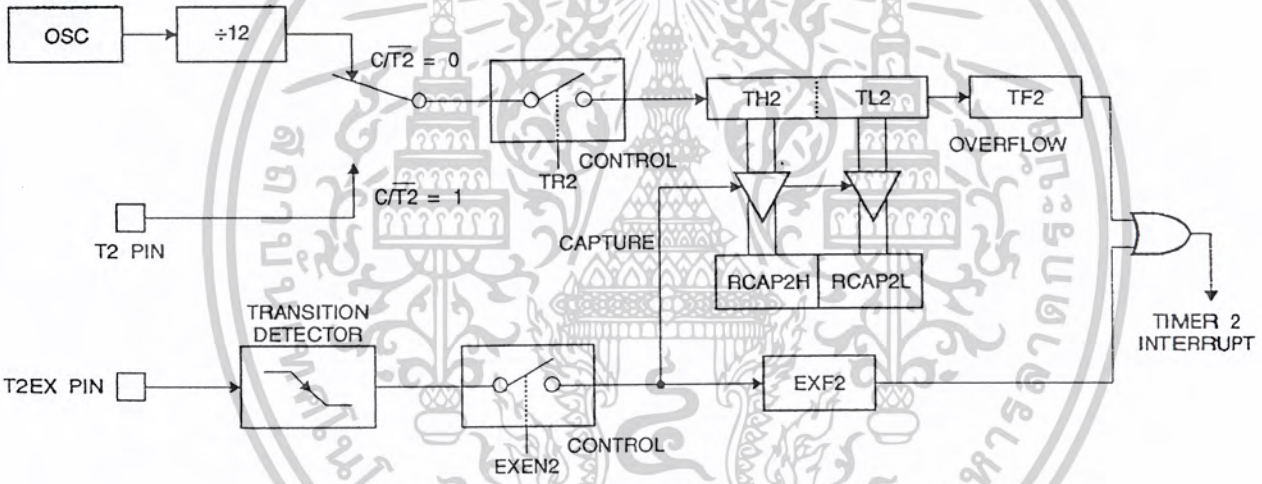
Table 8. Timer 2 Operating Modes

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

Capture Mode

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

Figure 1. Timer 2 in Capture Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Auto-reload (Up or Down Counter)

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

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

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

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

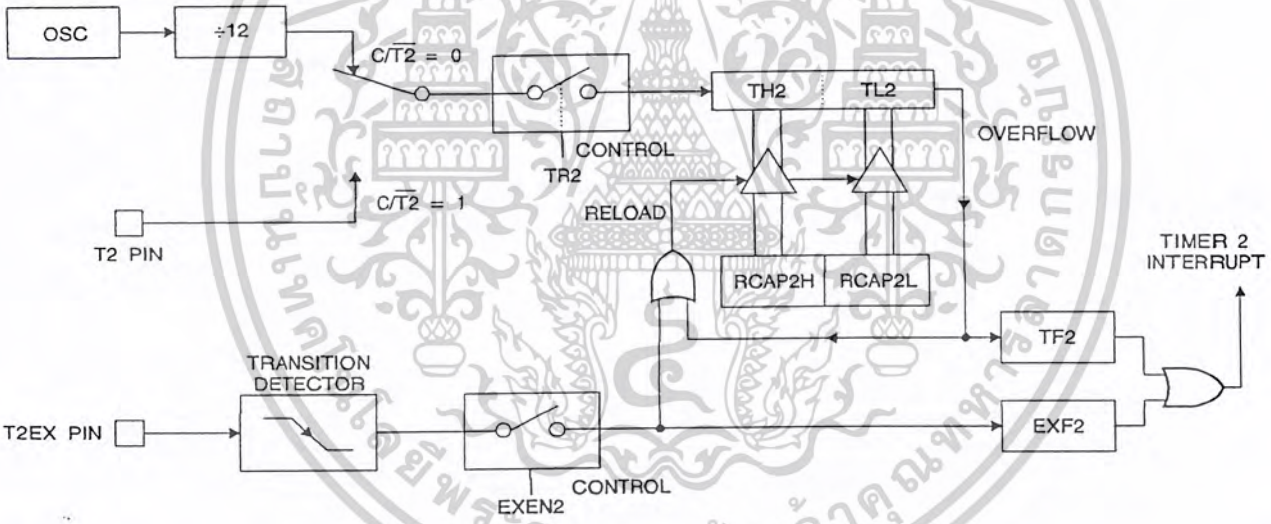


Table 9. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

	–	–	–	–	–	–	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

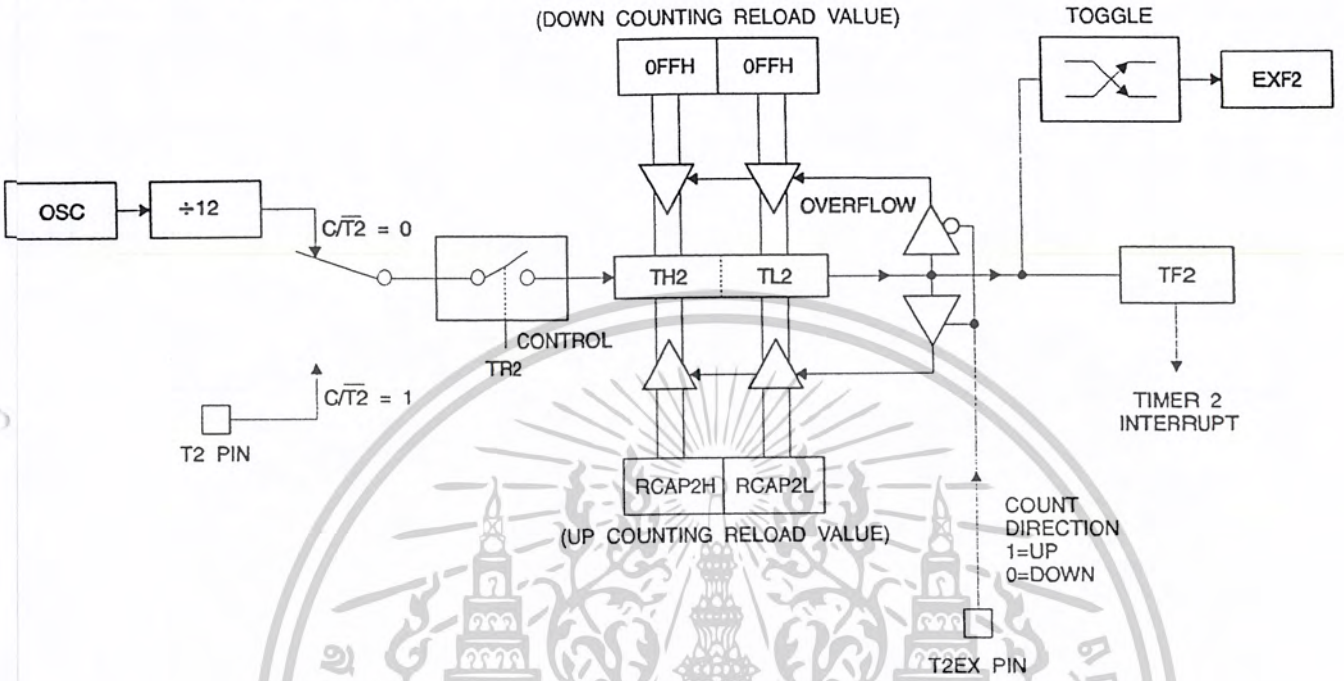
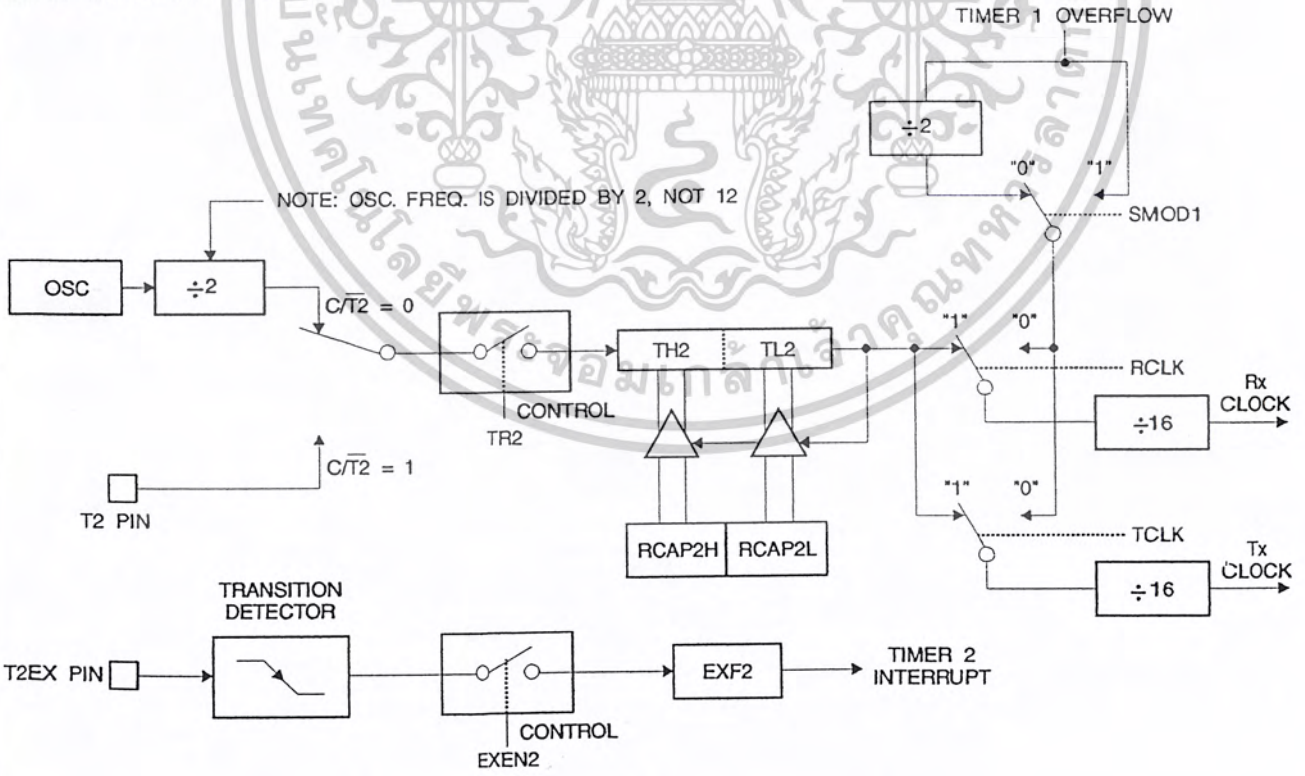


Figure 4. Timer 2 in Baud Rate Generator Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

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

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

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

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

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

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

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

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

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

Programmable Clock Out

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

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

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

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

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 5. Timer 2 in Clock-out Mode

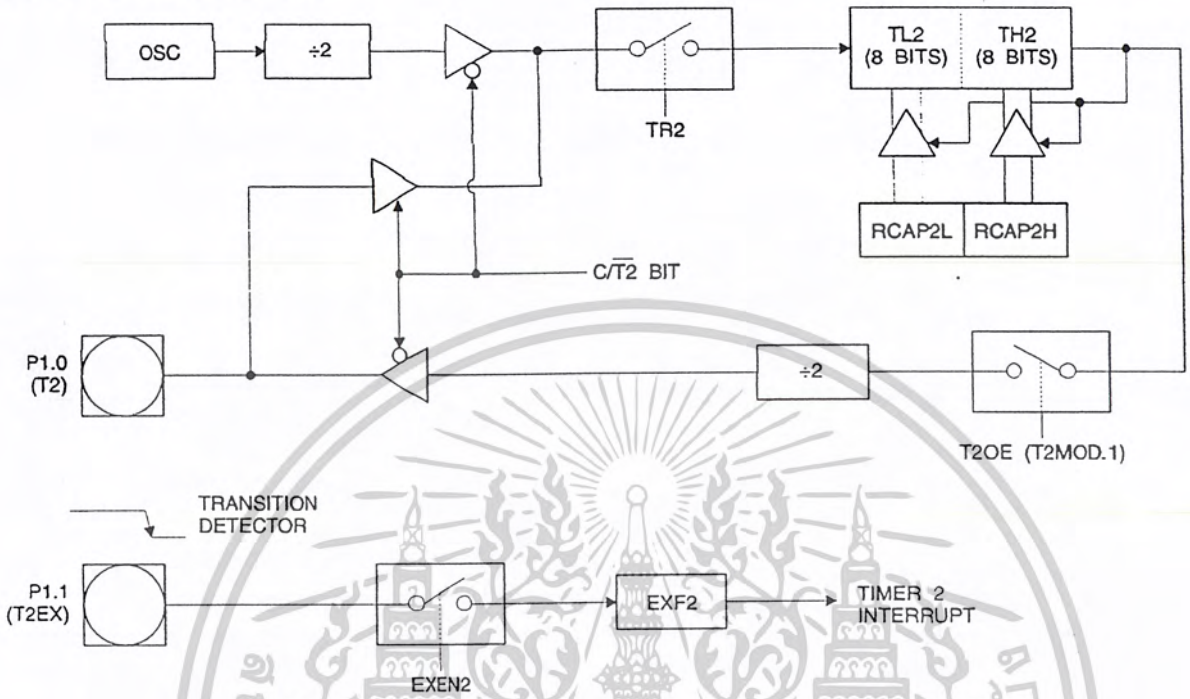
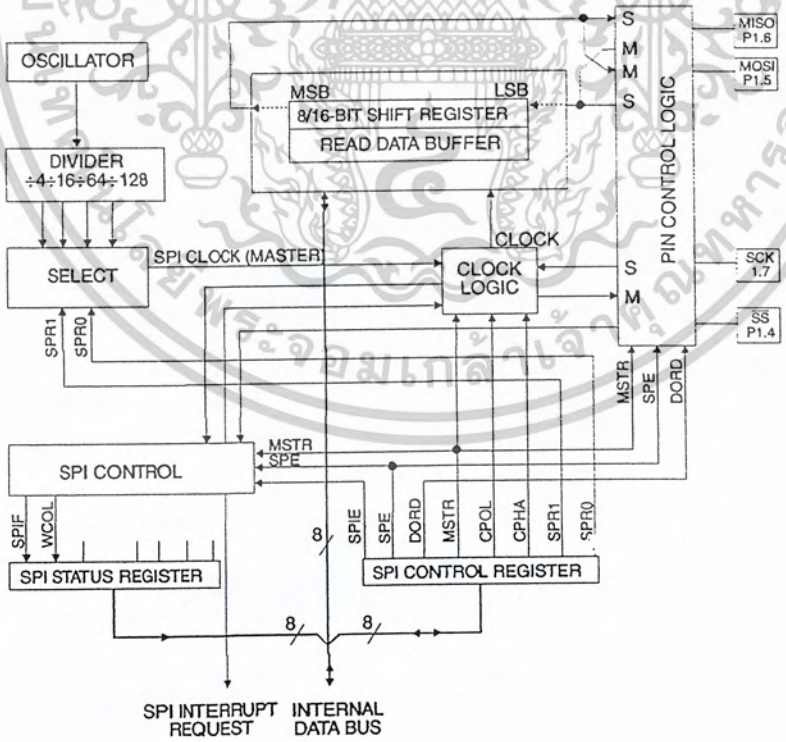


Figure 6. SPI Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JART

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

Serial Peripheral Interface

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

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

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

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

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

Figure 7. SPI Master-slave Interconnection

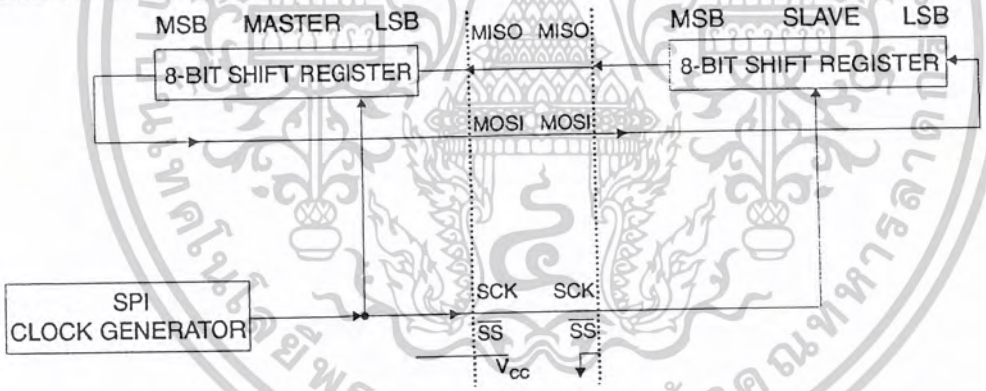
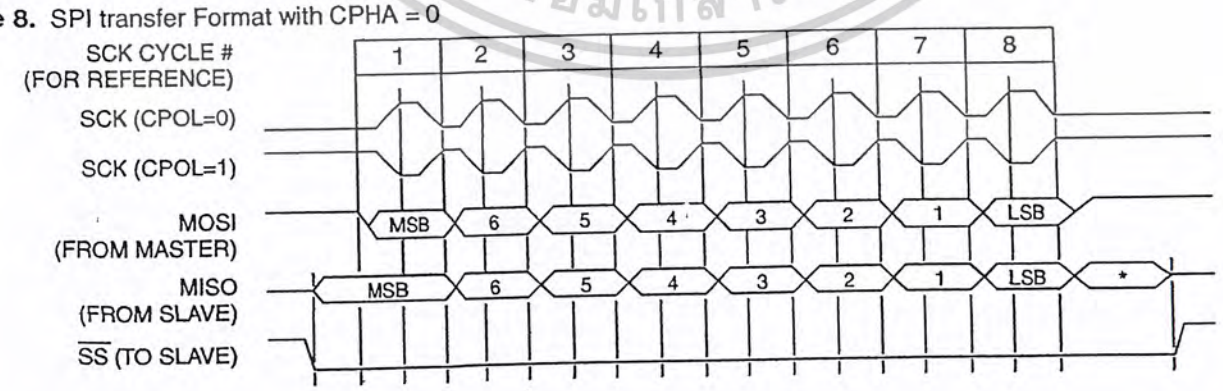


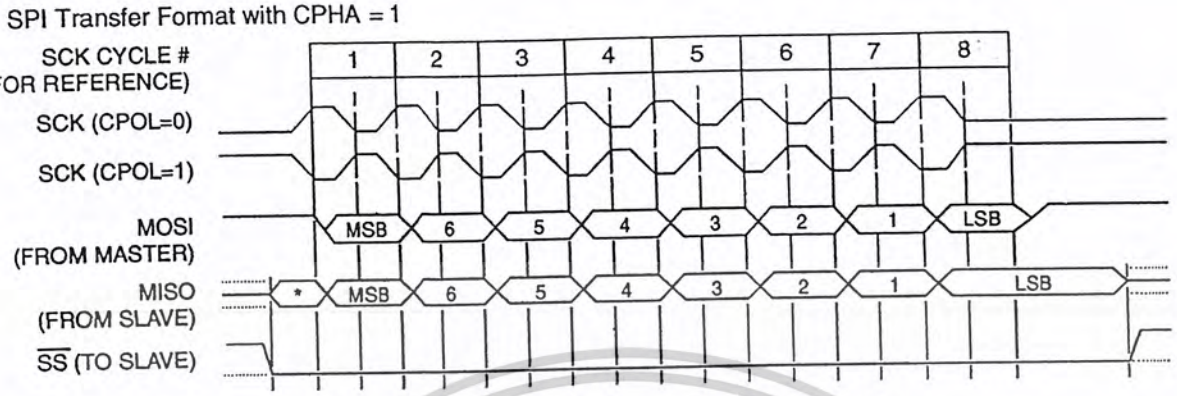
Figure 8. SPI transfer Format with CPHA = 0



*Not defined but normally MSB of character just received

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 9. SPI Transfer Format with CPHA = 1



Not defined but normally LSB of previously transmitted character

Interrupts

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

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

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

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

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

Table 10. Interrupt Enable (IE) Register

(MSB)(LSB)

EA	–	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 10. Interrupt Sources

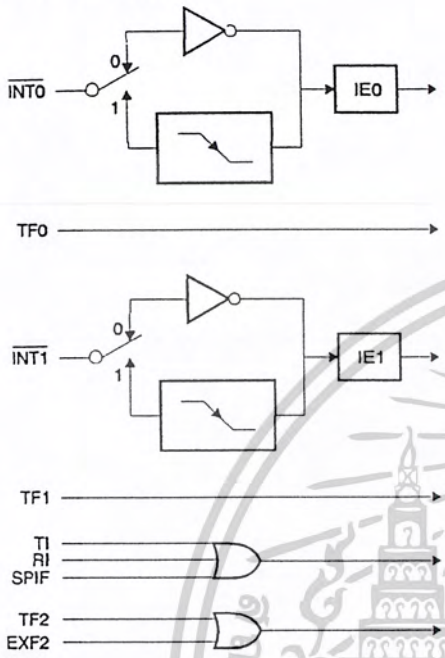
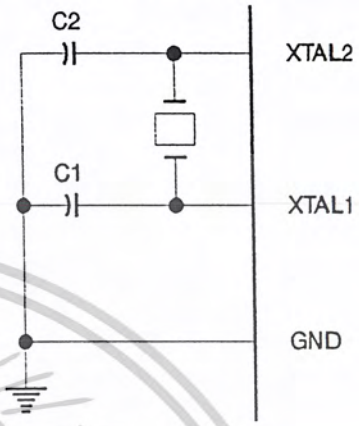
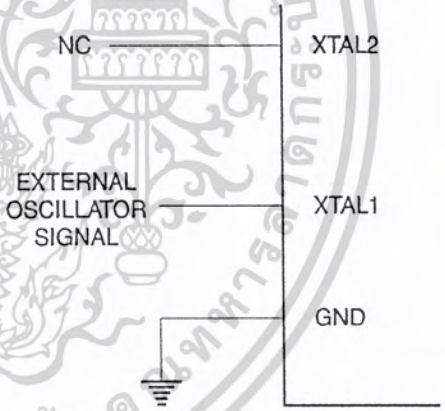


Figure 11. Oscillator Connections



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

Figure 12. External Clock Drive Configuration



Oscillator Characteristics

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Idle Mode

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

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

Status of External Pins During Idle and Power-down Modes

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

Power-down Mode

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

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

Program Memory Lock Bits

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

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

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

Lock Bit Protection Modes[1][2]

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash and EEPROM

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

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

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

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

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

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

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

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

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

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

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

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

The AT89S8252 is shipped with the Serial Programming Mode enabled.

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

(030H) = 1EH indicates manufactured by Atmel
(031H) = 72H indicates 89S8252

Programming Interface

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

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

Serial Downloading

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

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

The Code and Data memory arrays have separate address spaces:

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

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

Serial Programming Algorithm

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

1. Power-up sequence:
   Apply power between VCC and GND pins.
   Set RST pin to "H".
   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):
Set XTAL1 to "L" (if a crystal is not used).
Set RST to "L".
Turn V_{CC} power off.

Serial Programming Instruction

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

Note: 1. DATA polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
2. "aaaaa" = high order address.
3. "x" = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash and EEPROM Parallel Programming Modes

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 13. Programming the Flash/EEPROM Memory

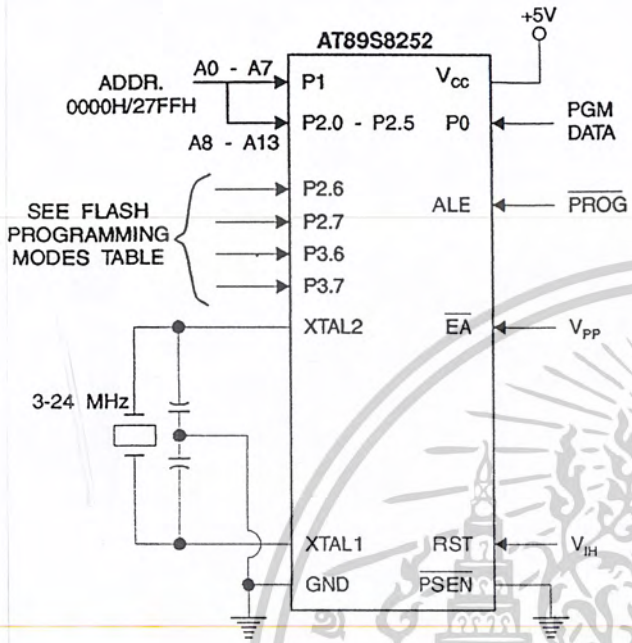


Figure 14. Verifying the Flash/EEPROM Memory

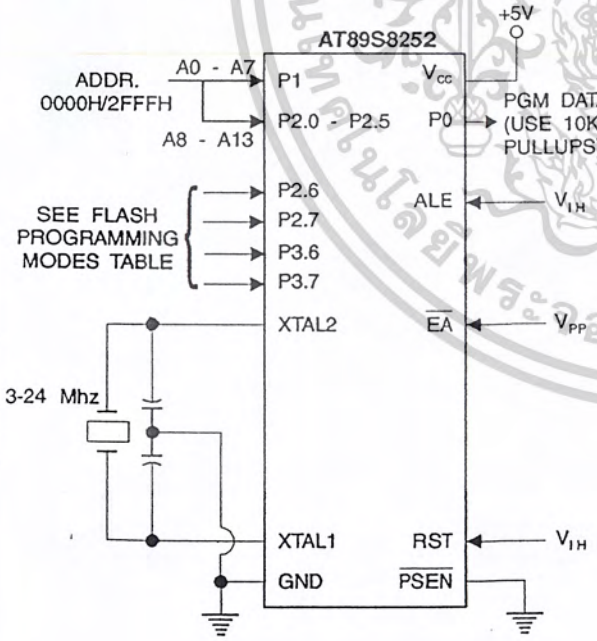
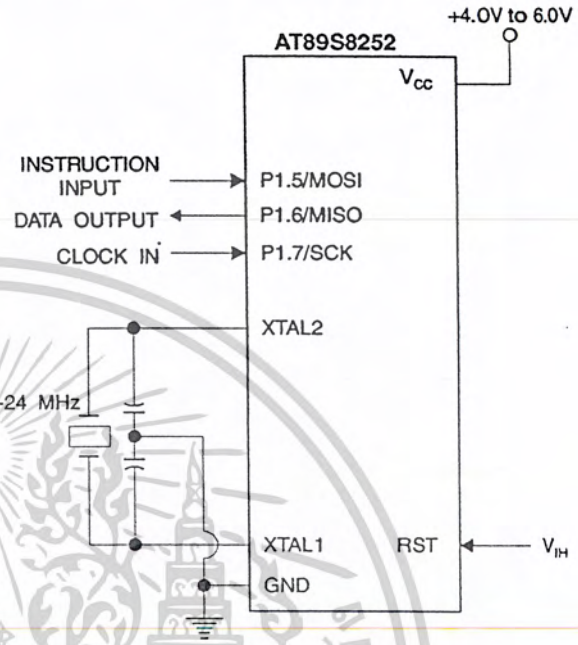


Figure 15. Flash/EEPROM Serial Downloading



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

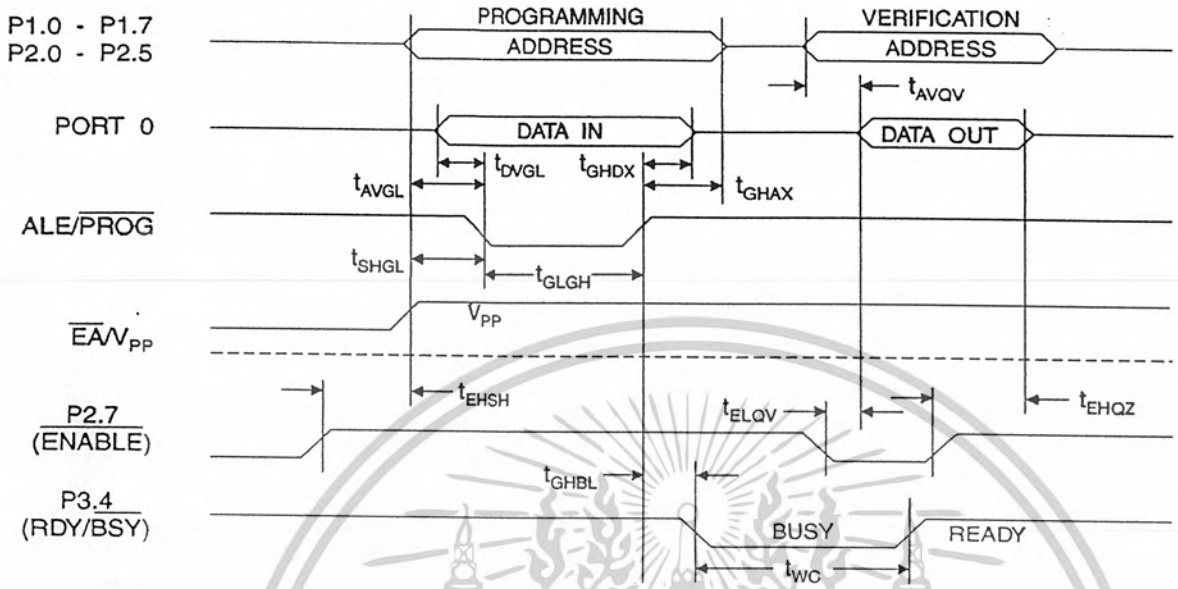
Flash Programming and Verification Characteristics – Parallel Mode

T_A = 0°C to 70°C, V_{CC} = 5.0V ± 10%

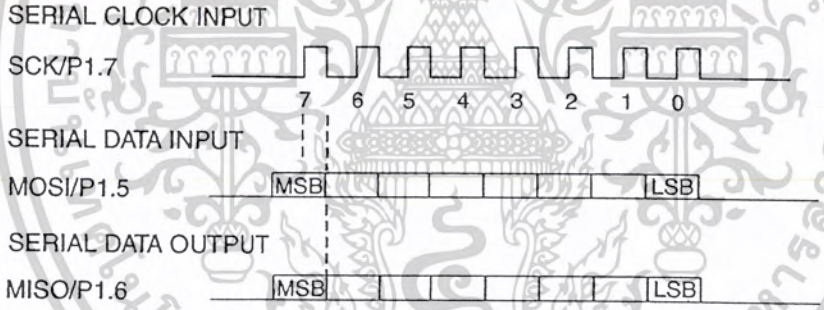
Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash/EEPROM Programming and Verification Waveforms – Parallel Mode



Serial Downloading Waveforms



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

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

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

DC Characteristics

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

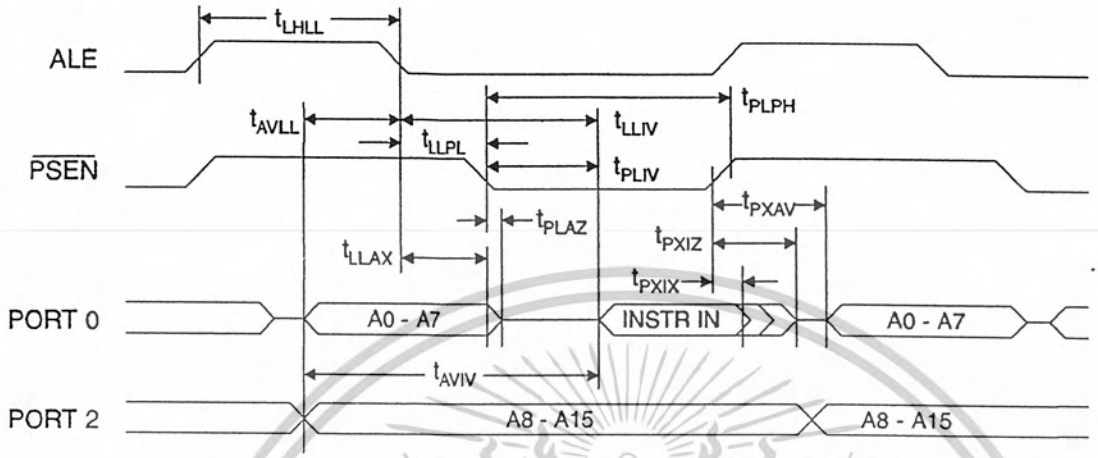
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

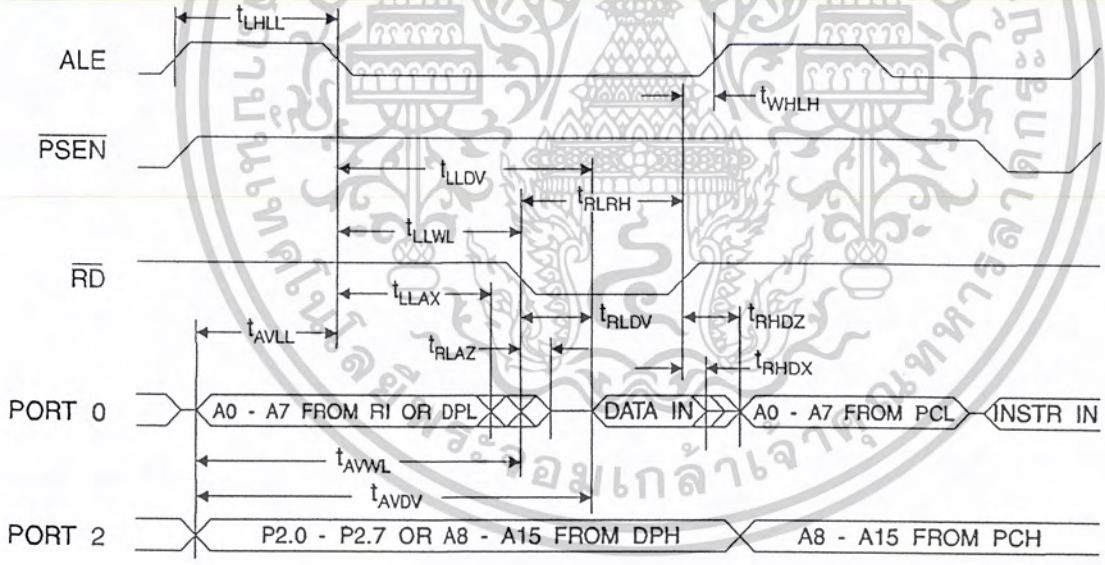
Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{LHLL}	ALE Pulse Width	$2t_{CLCL}$ - 40		ns
t_{AVLL}	Address Valid to ALE Low	t_{CLCL} - 13		ns
t_{LLAX}	Address Hold after ALE Low	t_{CLCL} - 20		ns
t_{LLIV}	ALE Low to Valid Instruction In		$4t_{CLCL}$ - 65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	t_{CLCL} - 13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	$3t_{CLCL}$ - 20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		$3t_{CLCL}$ - 45	ns
t_{PXIX}	Input Instruction Hold after $\overline{PSEN}$	0		ns
t_{PXIZ}	Input Instruction Float after $\overline{PSEN}$		t_{CLCL} - 10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	t_{CLCL} - 8		ns
t_{AVIV}	Address to Valid Instruction In		$5t_{CLCL}$ - 55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	$6t_{CLCL}$ - 100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	$6t_{CLCL}$ - 100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		$5t_{CLCL}$ - 90	ns
t_{RHDX}	Data Hold after $\overline{RD}$	0		ns
t_{RHDZ}	Data Float after $\overline{RD}$		$2t_{CLCL}$ - 28	ns
t_{LLDV}	ALE Low to Valid Data In		$8t_{CLCL}$ - 150	ns
t_{AVDV}	Address to Valid Data In		$9t_{CLCL}$ - 165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	$3t_{CLCL}$ - 50	$3t_{CLCL}$ + 50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	$4t_{CLCL}$ - 75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	t_{CLCL} - 20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	$7t_{CLCL}$ - 120		ns
t_{WHQX}	Data Hold after $\overline{WR}$	t_{CLCL} - 20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{CLCL} - 20	t_{CLCL} + 25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

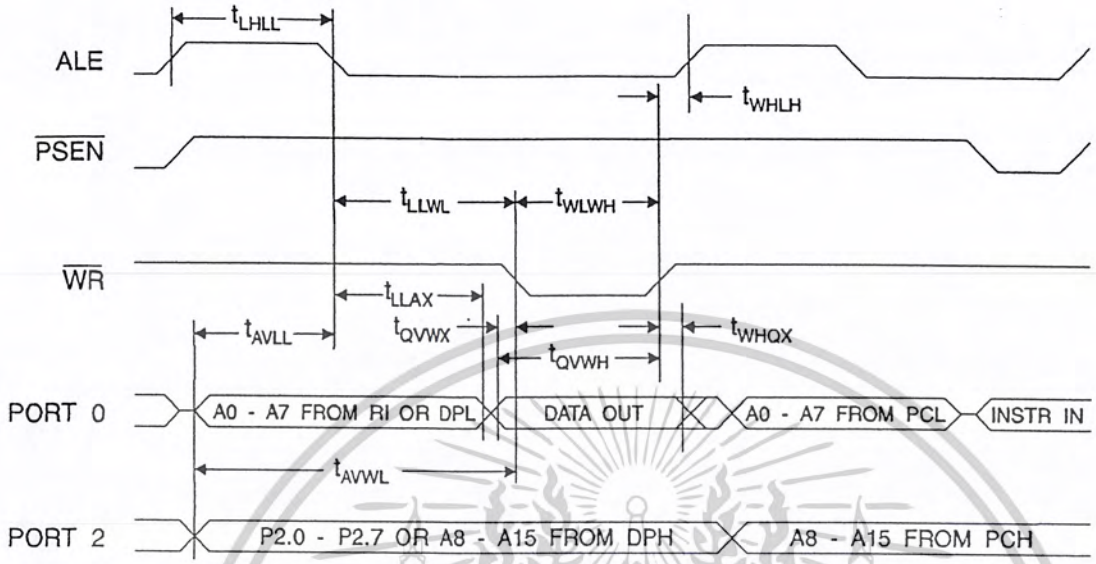


External Data Memory Read Cycle

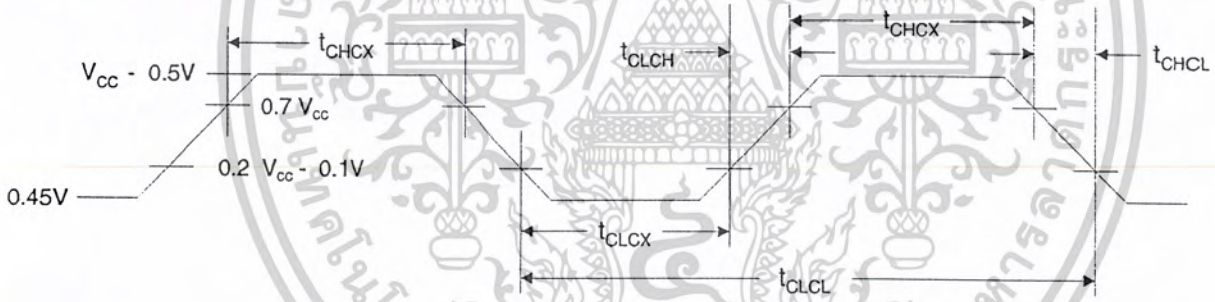


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

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

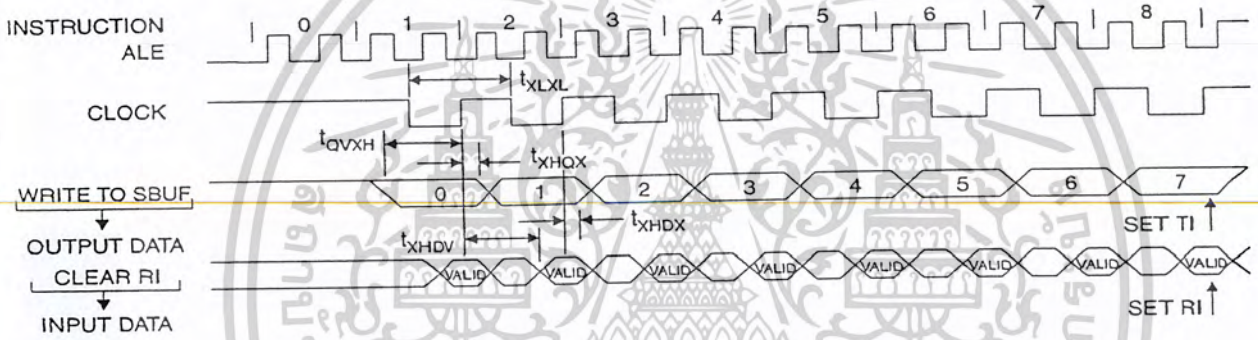
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

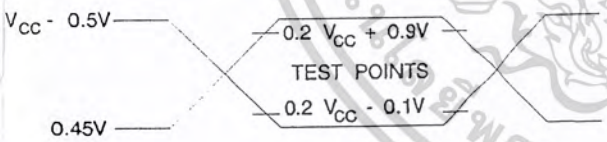
The values in this table are valid for V_{CC} = 4.0V to 6V and Load Capacitance = 80 pF.

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

Shift Register Mode Timing Waveforms

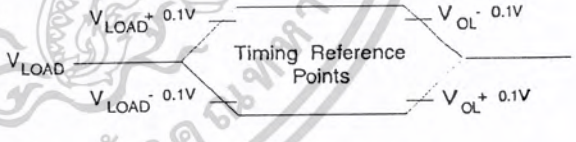


AC Testing Input/Output Waveforms(1)



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

Float Waveforms(1)

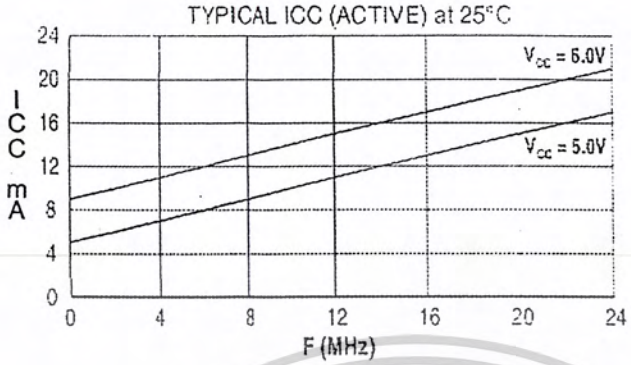


Notes: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

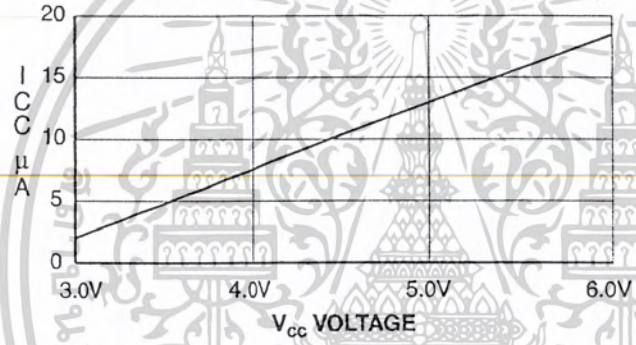
AT89S8252

TYPICAL ICC (ACTIVE) at 25°C



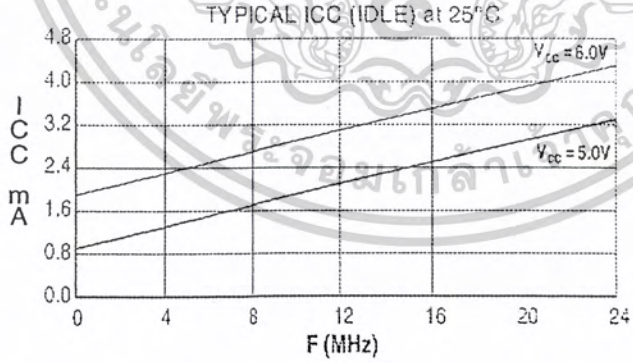
AT89S8252

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



AT89S8252

TYPICAL ICC (IDLE) at 25°C



Notes: 1. XTAL1 tied to GND for Icc (power-down)
2. Lock bits programmed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 6.0V	AT89S8252-24AC AT89S8252-24JC AT89S8252-24PC AT89S8252-24QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
	4.0V to 6.0V	AT89S8252-24AI AT89S8252-24JI AT89S8252-24PI AT89S8252-24QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
33	4.5V to 5.5V	AT89S8252-33AC AT89S8252-33JC AT89S8252-33PC AT89S8252-33QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)

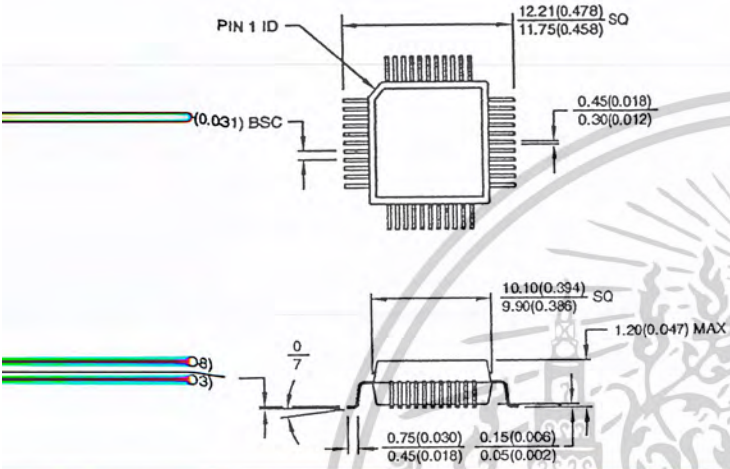
= Preliminary Information

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

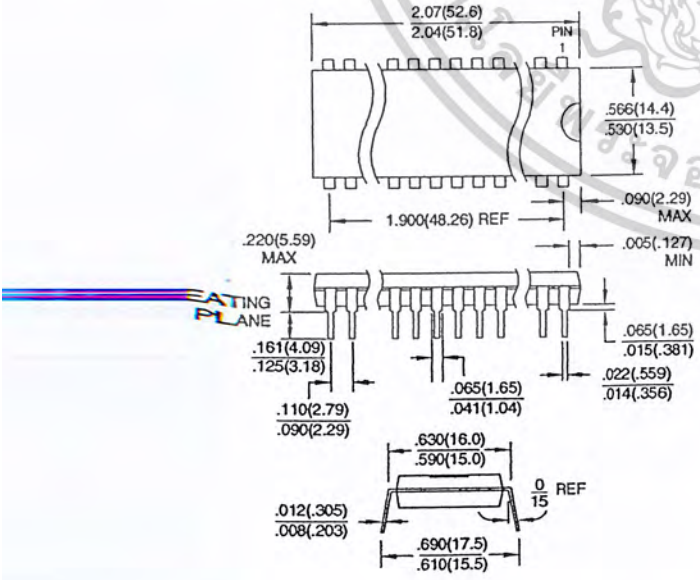
ging Information

44-lead, Thin (1.0 mm) Plastic Gull Wing Quad
ack (TQFP)
nsions in Millimeters and (Inches)*
C STANDARD MS-026 ACB

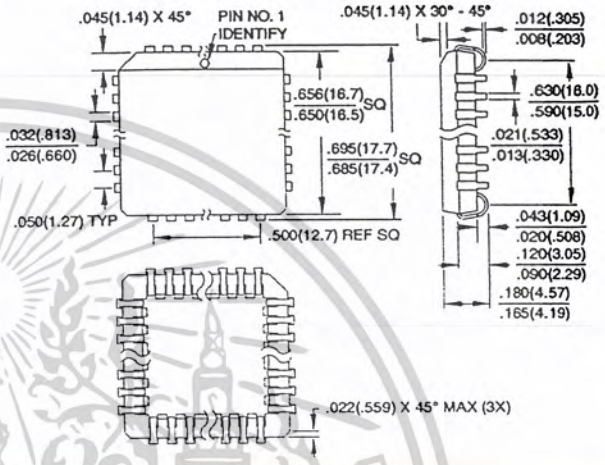


rolling dimension: millimeters

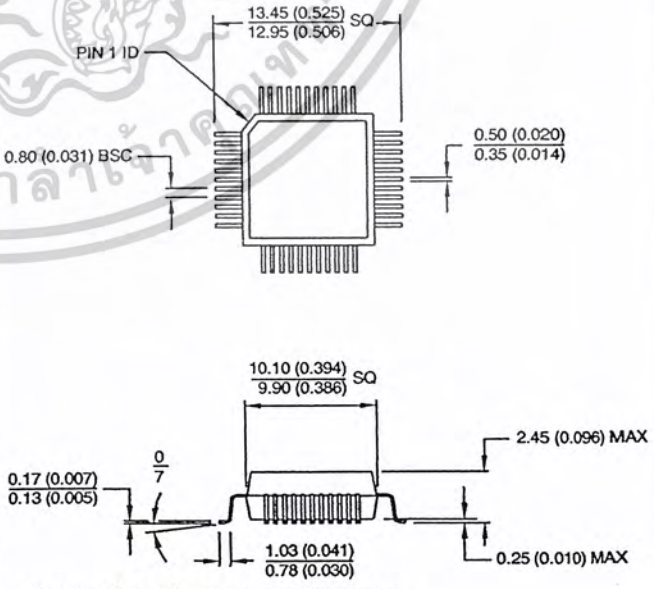
, 40-lead, 0.600" Wide, Plastic Dual Inline
age (PDIP)
nsions in Inches and (Millimeters)



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



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

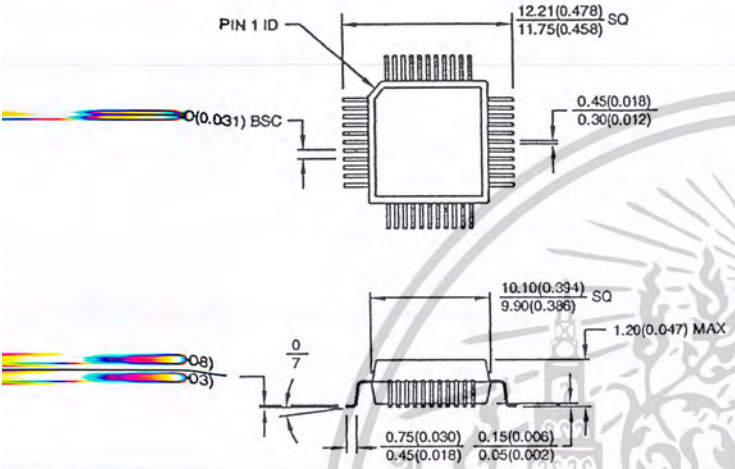


Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

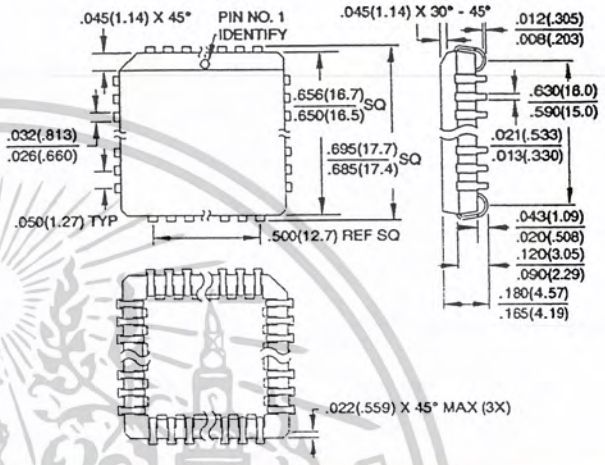
aging Information

, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad
pack (TQFP)
ensions in Millimeters and (Inches)*
EC STANDARD MS-026 ACB

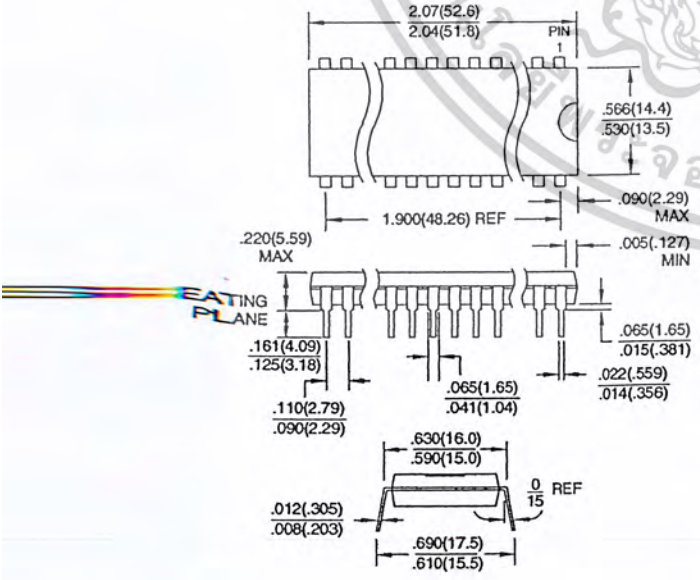


rolling dimension: millimeters

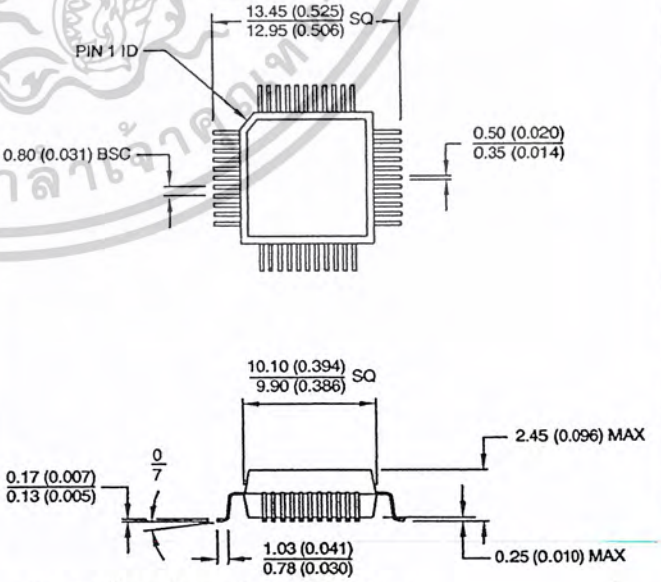
44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC



S, 40-lead, 0.600" Wide, Plastic Dual Inline
age (PDIP)
nsions in Inches and (Millimeters)



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



Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

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

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

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

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

Atmel Operations

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

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

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

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

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ธีรวัฒน์ ประกอบผล , **การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี** , กรุงเทพ , สำนักพิมพ์ ส.ส.ท. พ.ศ. 2545

บัณฑิต จามรภูมิ , **คู่มือการใช้งาน Protel 99** , เชียงใหม่ , พ.ศ. 2544

พรจิต ประทุมสุวรรณ . **การควบคุมนิวแมติกส์** , กรุงเทพฯ , 2535

วรพจน์ กรแก้ววัฒนกุล , ชัยวัฒน์ ลิ้มพรจิตรวิไล , **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายชลอศักดิ์ มหาประทุม
วันเดือนปีเกิด	20 ตุลาคม 2523
สถานที่เกิด	จังหวัดระยอง
ภูมิลำเนาเดิม	128 หมู่4 ตำบลตาขัน อำเภอบ้านค่าย จังหวัดระยอง 21120
ที่อยู่ปัจจุบัน	395/6 หมู่1 หมู่บ้านจินดา ซอยจินดา แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	-
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนหนองตะแบก
มัธยมศึกษาตอนต้น	โรงเรียนบ้านค่าย
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคระยอง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคระยอง
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	คิดจริง ทำจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	ว่าที่ร้อยตรีธานินทร์ คุณุทัย
วันเดือนปีเกิด	8 ธันวาคม 2523
สถานที่เกิด	จังหวัดสมุทรปราการ
ภูมิลำเนาเดิม	143/296 หมู่10 ซอยสันตินคร ตำบลบางปลา อำเภอบางพลี จังหวัดสมุทรปราการ 10540
ที่อยู่ปัจจุบัน	143/296 หมู่10 ซอยสันตินคร ตำบลบางปลา อำเภอบางพลี จังหวัดสมุทรปราการ 10540
โทรศัพท์	0-2750-4507
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนพระสมุทรเจดีย์
มัธยมศึกษาตอนต้น	โรงเรียนสมุทรปราการ
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคสมุทรปราการ
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคสมุทรปราการ
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	พยายาม อดทน มุ่งสู่ความเจริญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายพฤหัส ราชาเดช
วันเดือนปีเกิด	3 กรกฎาคม 2523
สถานที่เกิด	จังหวัดอ่างทอง
ภูมิลำเนาเดิม	168/1 หมู่2 ตำบลหลักแก้ว อำเภอวิเศษชัยชาญ จังหวัดอ่างทอง
ที่อยู่ปัจจุบัน	168/1 หมู่2 ตำบลหลักแก้ว อำเภอวิเศษชัยชาญ จังหวัดอ่างทอง
โทรศัพท์	0-1853-7121
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนวัดนาคู
มัธยมศึกษาตอนต้น	โรงเรียนวิเศษชัยชาญตันติวิทยาภูมิ
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคอ่างทอง
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคอ่างทอง
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	รักชาติยิ่งชีพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายภูวดล ทับสีรัก
วันเดือนปีเกิด	23 มกราคม 2524
สถานที่เกิด	จังหวัดมหาสารคราม
ภูมิลำเนาเดิม	164 หมู่7 ตำบลแกดำ อำเภอแกดำ จังหวัดมหาสารคราม 44190
ที่อยู่ปัจจุบัน	164 หมู่7 ตำบลแกดำ อำเภอแกดำ จังหวัดมหาสารคราม 44190
โทรศัพท์	-
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนชุมชนบ้านหมากค่า
มัธยมศึกษาตอนต้น	โรงเรียนมิตรภาพ
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคมหาสารคราม
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขตขอนแก่น
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	ไม่มีอะไรยากเกินไป ถ้าเรากล้าที่จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้