

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปีการศึกษา 2550

**เครื่องขายข้าวอัตโนมัติ
AUTOMATIC VENDING RICE MACHINE**



โดย

นาย กัมปนาท ล้อมเมตตา
นาย กิตติ จิตรสำราญ
นาย เกรียงศักดิ์ วรรณธนกิจ
นาย ไกรพัฒน์ เสวกวรรณ
นาย จักรพันธ์ บรรจงรอด

รฟ.
ก 393 ค
2550

เลขหมู่.....
เลขทะเบียน..... **81805**
วัน,เดือน,ปี **24 ส.ย. 2551**

b. **1193632x**
i.....

อาจารย์ที่ปรึกษา

อ. ดุสิต สุขสวัสดิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2550


ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องขยายข้าวอัตโนมัติ

ผู้จัดทำ

- 
1. นาย กัมปนาท ล้อมเมตตา
 2. นาย กิตติ จิตรสำราญ
 3. นาย เกรียงศักดิ์ วรรณธนกิจ
 4. นาย ไกรพัฒน์ เสวกวรรณ
 5. นาย จักรพันธ์ บรรจงรอด



.....อาจารย์ที่ปรึกษา

(อาจารย์ ดุสิต สุขสวัสดิ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องขายข้าวอัตโนมัติ

นาย กัมปนาท ล้อมเมตตา

นาย กิตติ จิตรสำราญ

นาย เกรียงศักดิ์ วรรณชนกิจ

นาย ไกรพัฒน์ เสวกวรรณ

นาย จักรพันธ์ บรรจงรอด

อ. ดุสิต สุขสวัสดิ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการออกแบบและการสร้างเครื่องขายข้าวอัตโนมัติ มีการควบคุมโดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีการติดต่อกับหน่วยความจำภายนอก ตัวเลขเจ็ดส่วน เครื่องรับเหรียญ เครื่องทอนเหรียญ รีเลย์ และโมดูล GSM/GPRS เครื่องขายข้าวอัตโนมัตินี้เป็นเครื่องที่ช่วยลดขั้นตอนการจำหน่ายข้าวจากชานาไปยังผู้บริโภค โดยสามารถจำหน่ายข้าวที่ได้บรรจุไว้เป็นถุงๆ ละ 2 กิโลกรัม จำนวน 50 ถุง และสามารถจำหน่ายได้ตลอด 24 ชั่วโมง ทำให้ผู้บริโภคมีความสะดวกสบายมากขึ้น

AUTOMATIC VENDING RICE MACHINE

Gumpanat Lommetta

Kitti Chitsamran

Kriangsak Wantanakij

Kripat Sawagwan

Jakrapan Banjongrod

Dusit Suksawat Advisor

2007

ABSTRACT

This thesis presents the design and the construction of the automatic vending rice machine. It is controlled by microcontroller MCS-51. It connects to EEPROM, 7-segment, coin box, coin dispenser, relays and GSM/GPRS module. This automatic vending rice machine helps for reducing steps of trading rice from farmers to consumers. It can serve that is packed of 2 kilograms for 50 bags and can serve 24 hours a day that makes the consumers have more comfortable than ever.

กิตติกรรมประกาศ

การทำโครงการนี้คณะผู้จัดทำโครงการขอขอบพระคุณบิดา มารดา อันเป็นที่เคารพรัก ที่ช่วยอบรมสั่งสอนเลี้ยงดูคณะผู้จัดทำมาเป็นอย่างดี รวมถึงกำลังใจในการทำงานที่ได้รับมาโดยตลอด และส่งเสริมด้านการศึกษาให้คณะผู้จัดทำได้เป็นบุคคลที่มีความรู้มาถึงปัจจุบันนี้

โครงการนี้จะเสร็จสมบูรณ์มิได้ หากขาดอาจารย์ที่ปรึกษาโครงการ อาจารย์ ดุสิต สุขสวัสดิ์ ซึ่งเป็นผู้ให้คำปรึกษา แนวทางแก้ไขปัญหา และข้อเสนอแนะต่างๆ ในการจัดทำโครงการนี้ให้สำเร็จลุล่วงไปได้ด้วยดี พร้อมกันนี้ใคร่ขอขอบคุณต่อ

ร้านโกวิทการช่าง ที่ให้การสนับสนุนด้านเครื่องมือและอุปกรณ์ในการสร้างเครื่องขยายข่าว

โรงสีข้าวพิจิตร ที่ให้การสนับสนุนข้าวและงบประมาณส่วนใหญ่ที่ใช้ในการทำโครงการในครั้งนี้

สุดท้ายนี้ขอขอบคุณเพื่อน พี่ น้อง ทุกคนที่เป็นกำลังใจ ตลอดจนผู้ที่สนใจโครงการนี้ทุกท่าน ซึ่งเป็นแรงผลักดันอย่างยิ่งที่ทำให้คณะผู้จัดทำมีกำลังใจที่จะพัฒนาโครงการนี้ให้ดียิ่งๆ ขึ้นไป และพร้อมกันนี้ ขอขอบคุณต่อผู้ที่มีส่วนเกี่ยวข้องทุกๆ ท่านที่ช่วยให้โครงการนี้สำเร็จลุล่วงไปด้วยดี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	82
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ	III
สารบัญ.....	IV
สารบัญภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตและข้อกำหนดของโครงการ.....	2
1.4 ขั้นตอนและวิธีการดำเนินงาน	2
1.5 ประโยชน์ที่ได้รับ	2
1.6 การจัดโครงสร้างของเนื้อหาภายในปฏิญานพนธ์.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 ไมโครคอนโทรลเลอร์	3
2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 อนุกรม AT89C51.....	4
2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51.....	5
2.1.3 การทำงานของพอร์ต.....	6
2.1.4 การทำงานของโปรแกรมรูปและการหน่วงเวลา.....	7
2.2 อุปกรณ์ในกล่องควบคุม.....	9
2.2.1 LED ตัวเลข 7 ส่วน.....	9
2.2.2 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51.....	11
2.2.3 การเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำอีอีพรอมอนุกรม EEPROM..	13
2.2.4 รีเลย์ (Relay).....	20
2.3 อุปกรณ์ต่อพ่วง	21
2.3.1 GR64.....	21
2.3.2 เครื่องรับเหรียญ.....	25
2.3.3 เครื่องทอนเหรียญ.....	25

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การออกแบบและการสร้าง	27
3.1 สร้างกล่องควบคุม	31
3.1.1 วงจรไฟเลี้ยง	31
3.1.2 วงจรควบคุม	31
3.1.3 GR64	35
3.1.4 เครื่องรับเหรียญ	35
3.1.5 เครื่องทอนเหรียญ	35
3.2 สร้างโครงตู้	36
3.3 กระบวนการทำงานของเครื่องขายข้าวอัดโนมิตี	38
3.3.1 ขั้นตอนการซื้อ	38
3.3.2 กระบวนการทำงานของเครื่องขายข้าวอัดโนมิตี	39
3.3.3 การปรับโหมด	39
บทที่ 4 ผลการทดลอง	42
4.1 การทดลองการทำงานของเครื่องขายข้าวอัดโนมิตี	42
4.1.1 การทดลองที่ 1	43
4.1.2 การทดลองที่ 2	44
4.1.3 การทดลองที่ 3	45
4.1.4 การทดลองที่ 4	46
4.1.5 การทดลองที่ 5	47
4.1.6 การทดลองที่ 6	48
4.2 การทดลองการใช้งานจริงของเครื่องขายข้าวอัดโนมิตี	49
บทที่ 5 สรุปและข้อเสนอแนะ	51
5.1 สรุปผลการทำงาน	51
5.1.1 ส่วนควบคุม	51
5.1.2 ส่วนแสดงผล	51
5.2 ปัญหาและข้อเสนอแนะ	52
บรรณานุกรม	53
ภาคผนวก	54
ประวัติผู้เขียน	80

สารบัญญภาพ

ภาพที่	หน้า
1.1 แผนภาพขั้นตอนการจำหน่ายข้าวจากผู้ผลิตถึงผู้บริโภค.....	1
2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์.....	3
2.2 โครงสร้างพื้นฐานไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชไมโครโปรเซสเซอร์ AT89C51.....	4
2.3 โฟลวชาร์ตของโปรแกรมลูปลูกอย่างง่าย.....	8
2.4 โฟลวชาร์ตของโปรแกรมรูปแบบใช้ลูปลูกแอนด์เตอร์.....	8
2.5 ลักษณะของ LED ตัวเลข 7 ส่วน.....	9
2.6 LED ตัวเลข 7 ส่วนแบบแคโทดรวม.....	10
2.7 LED ตัวเลข 7 ส่วนแบบแอนโอดรวม.....	10
2.8 วงจรขับ LED.....	11
2.9 ขาของหน่วยความจำอีอีพรอม.....	13
2.10 รูปแบบการเขียนข้อมูลแบบไบต์ของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24C16.....	14
2.11 รูปแบบการเขียนข้อมูลแบบเพจของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24C16.....	15
2.12 โฟลวชาร์ตของกระบวนการตรวจสอบสถานะการเขียนข้อมูลในอีอีพรอม.....	17
2.13 รูปแบบการอ่านข้อมูลแบบแอดเดรสปัจจุบันของหน่วยความจำอีอีพรอม.....	18
2.14 รูปแบบการอ่านข้อมูลแบบสุ่มหรือแบบเจาะจงแอดเดรสของอีอีพรอม.....	18
2.15 รูปแบบการอ่านข้อมูลแบบเรียงลำดับของอีอีพรอม.....	19
2.16 การต่อรีเลย์ในสภาวะปกติและสภาวะจ่ายไฟฟ้า.....	20
2.17 รีเลย์เบอร์ HRS4 (H).....	21
2.18 บอร์ด GR-64.....	22
2.19 การส่ง SMS.....	23
2.20 เครื่องรับเหรียญ.....	25
2.21 เครื่องทอนเหรียญ.....	26
3.1 บล็อกไดอะแกรมการทำงานของเครื่องขายข้าวอัตโนมัติ.....	27
3.2 โฟลวชาร์ตการทำงานของเครื่องขายข้าวอัตโนมัติ.....	28
3.3 ลายทองแดงของแผงควบคุม.....	30
3.4 การลงอุปกรณ์ในแผงควบคุม.....	30
3.5 วงจรไฟเลี้ยง.....	31
3.6 วงจรไมโครคอนโทรลเลอร์.....	32
3.7 วงจร LED ตัวเลข 7 ส่วน.....	32

สารบัญภาพ(ต่อ)

ภาพที่	หน้า
3.8 วงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์	33
3.9 วงจรหน่วยความจำอีพีรอมอนุกรม.....	34
3.10 บล็อกไดอะแกรมของรีเลย์	34
3.11 บล็อกไดอะแกรมของ GR64	35
3.12 บล็อกไดอะแกรมของเครื่องรับเหรียญ	35
3.13 บล็อกไดอะแกรมของเครื่องทอนเหรียญ	36
3.14 โครงสร้างเครื่องขายข้าวอัตโนมัติ	36
3.15 มอเตอร์และชุดหม้อแปลง	37
3.16 ลูกลายลดการเกิด DAMPING	37
3.17 ราวแขวนข้าว	38
3.18 ชุดปลดถุงข้าวและ LIMIT SWITCH	38
3.19 รางจำหน่ายข้าว	38
3.20 การแสดงผลขณะทำการซื้อ	39
3.21 สวิตช์ภายในกล่องควบคุม	40
3.22 การเลือกโหมดของสวิตช์ฟังก์ชัน	40
3.23 การแสดงผลในโหมดการบรรจุข้าว	40
3.24 การแสดงผลในโหมดการตั้งราคา	41
4.1 ตำแหน่งในการแสดงผล.....	43
4.2 เครื่องขายข้าวอัตโนมัติที่ทดลองให้บริการ.....	50
4.3 สถานที่ติดตั้งเครื่องขายข้าวอัตโนมัติ	50
5.1 เครื่องขายข้าวอัตโนมัติ.....	51

สารบัญตาราง

ตารางที่	หน้า
3.1 การต่อใช้งานขาและพอร์ตของไมโครคอนโทรลเลอร์.....	29
4.1 ผลการทดลองที่ 1.....	43
4.2 ผลการทดลองที่ 2.....	44
4.3 ผลการทดลองที่ 3.....	45
4.4 ผลการทดลองที่ 4.....	46
4.5 ผลการทดลองที่ 5.....	47
4.6 ผลการทดลองที่ 6.....	48
4.7 แสดงผลการใช้งานจริงของเครื่องขायข้าวอัตโนมัติ.....	49



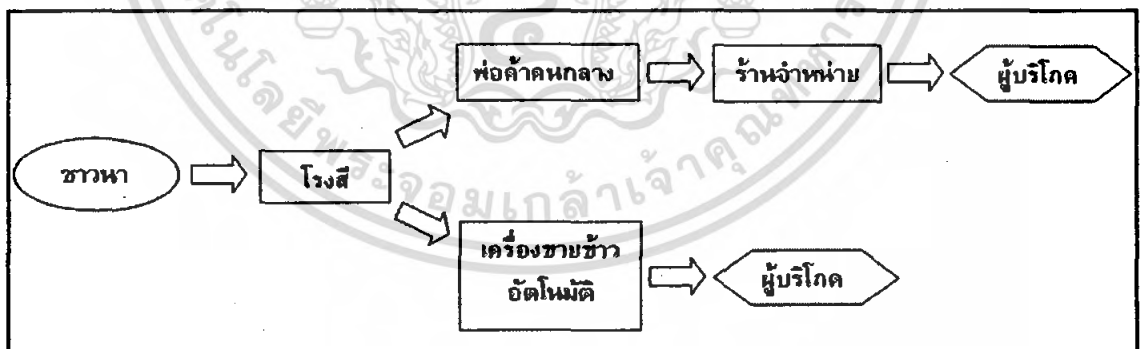
บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ข้าวเป็นสินค้าที่มีความจำเป็นในการดำรงชีวิตของคนไทยเป็นอย่างยิ่ง เนื่องจากประชากรส่วนใหญ่นิยมบริโภคข้าวเป็นอาหารหลัก เมื่อเรามีการบริโภคข้าวแล้ว การจำหน่ายข้าวจึงถือว่ามีค่าจำเป็น แต่ในปัจจุบันกว่าที่ข้าวจะมาถึงมือผู้บริโภคนั้นต้องมีการส่งผ่านมาหลายขั้นตอน ดังแสดงในภาพที่ 1.1 จะเห็นได้ว่า จากชาวนาซึ่งถือว่าเป็นผู้ผลิตรายแรกที่ได้ทำการเก็บเกี่ยวผลผลิต จากนั้นก็นำข้าวไปจำหน่ายให้กับโรงสี ซึ่งจะส่งต่อไปให้กับพ่อค้าคนกลาง จากนั้นจะส่งต่อไปให้บริษัทที่ทำการบรรจุภัณฑ์ข้าว แล้วจึงส่งไปให้ตัวแทนผู้จำหน่ายไปวางขายภายในห้างสรรพสินค้าต่างๆ ซึ่งจากที่กล่าวมามีผลทำให้ราคาของข้าวสูงขึ้นจากความ เป็นจริงมาก อีกทั้งชาวนายังถูกกดราคาของข้าวเพื่อที่นายทุนในขั้นตอนต่างๆจะได้รับกำไรที่มากขึ้น ซึ่งถ้าเราสามารถลดขั้นตอนในการส่งข้าวมายังผู้บริโภคได้ ก็จะทำให้ชาวนาไม่ถูกกดราคาข้าวจากนายทุนในระดับชั้นต่างๆ และราคาข้าวก็ถูกลงเพราะไม่ผ่านการส่งต่อหลายๆทอด

เนื่องจากผู้จัดทำโครงการได้สังเกตเห็นถึงปัญหาเหล่านี้ จึงได้มีแนวความคิดในการจัดทำเครื่องขายข้าวอัตโนมัติ ซึ่งสามารถลดต้นทุนในการส่งผ่านข้าวจากชาวนาไปยังผู้บริโภค และสามารถซื้อขายข้าวได้โดยสะดวก อีกทั้งไม่จำเป็นต้องมีคนมาควบคุมตลอดเวลา เป็นการใช้ทรัพยากรที่มีอยู่ได้อย่างคุ้มค่า



ภาพที่ 1.1 แผนภาพขั้นตอนการจำหน่ายข้าวจากผู้ผลิตถึงผู้บริโภค

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาทำการออกแบบและสร้างเครื่องจำหน่ายข้าวอัตโนมัติ
2. เพื่อลดขั้นตอนการจำหน่ายข้าวจากผู้ผลิตถึงผู้บริโภค
3. เพื่อเพิ่มความสะดวกรบายให้แก่ผู้บริโภคในการซื้อข้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตและข้อกำหนดของโครงการ

1. ระบบของเครื่อง เครื่องขยายข้าวอัดโนมิตินี้ทำงานโดยใช้มอเตอร์หมุน โดยที่ถุงข้าวจะถูกแขวนอยู่ที่ราวแขวนข้าว เมื่อมอเตอร์ทำงานราวแขวนข้าวก็จะหมุนถุงข้าวของเราจะไปชนสลักปลดให้ถุงข้าวตกลงมา

2. ก่อสร้างควบคุม จะใช้ไมโครคอนโทรลเลอร์ในการควบคุม โดยมีการติดต่อกับหน่วยความจำภายนอก ตัวเลขเจ็ดส่วน เครื่องหยุดเหรียญ เครื่องทอนเหรียญ รีเลย์ และโมดูล GSM/GPRS

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาเกี่ยวกับไมโครคอนโทรลเลอร์และอุปกรณ์ต่อพ่วง
2. สร้างวงจรควบคุมการทำงานของเครื่องขยายข้าวอัดโนมิตินี้
3. สร้างชิ้นส่วนต่างๆ ของเครื่องขยายข้าวอัดโนมิตินี้
4. ประกอบอุปกรณ์ต่อพ่วงเข้ากับกล่องควบคุม และชิ้นส่วนต่างๆ
5. ทดสอบการทำงานของเครื่องขยายข้าวอัดโนมิตินี้
6. สรุปผล และให้ข้อเสนอแนะ

1.5 ประโยชน์ที่ได้รับ

1. สามารถนำความรู้ไปประยุกต์ใช้ในการออกแบบและสร้างเครื่องจำหน่ายข้าวอัดโนมิตินี้ได้
2. สามารถลดขั้นตอนการจำหน่ายข้าวจากผู้ผลิตถึงผู้บริโภค
3. สามารถเพิ่มความสะดวกสบายให้แก่ผู้บริโภคในการซื้อข้าว

1.6 การจัดโครงสร้างของเนื้อหาภายในปฏิญญาพันธ

เนื้อหาของรายงานฉบับนี้มี 5 บท แต่ละบทประกอบด้วยเนื้อหาโดยสรุปดังนี้ คือ

บทที่ 1 กล่าวถึงความเป็นมาของปัญหา วัตถุประสงค์ของโครงการ ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงการ และส่วนประกอบของรายงานฉบับนี้

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่ใช้ในโครงการ

บทที่ 3 กล่าวถึงส่วนของการออกแบบระบบของโครงการ และการทำงานของระบบ

บทที่ 4 กล่าวถึงส่วนของการทดลอง และผลการทดลองของโครงการนี้

บทที่ 5 เป็นบทสรุปและวิจารณ์ผลการทดลอง ซึ่งกล่าวถึงบทสรุปของโครงการ วิจารณ์สิ่งที่ได้รับจากโครงการ ปัญหาอุปสรรคและแนวทางแก้ไข และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

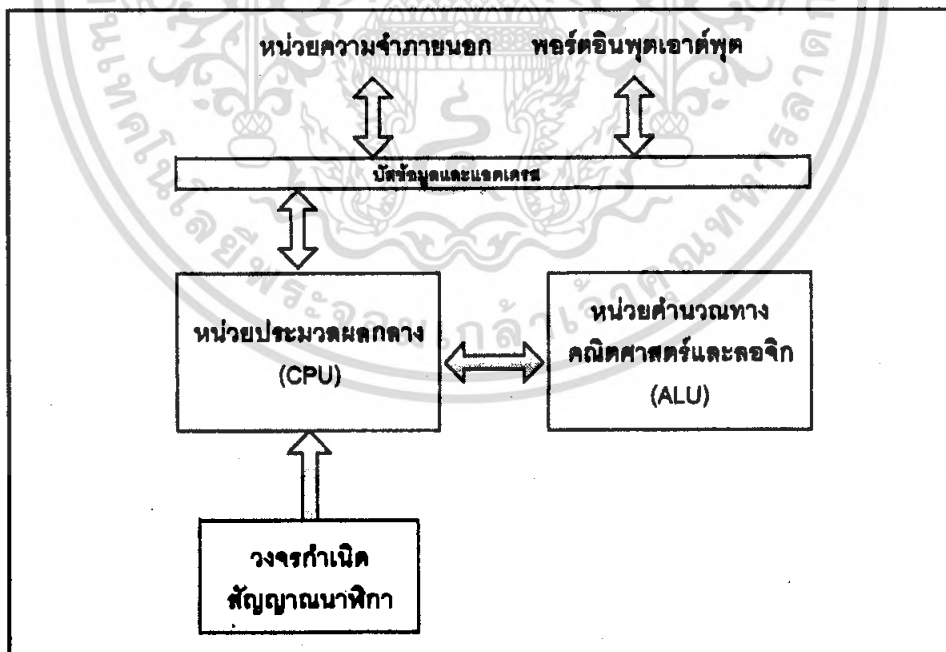
ทฤษฎีที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่ง ที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิกวงจรรับสัญญาณ อินพุต วงจรขับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานนอกวงจรรีเลย์ทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์ และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจาก 2 คำรวมกันคือ “ไมโคร” (micro) ซึ่งหมายถึง ไมโครโปรเซสเซอร์ (microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลางหรือ ซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์” (controller) หมายถึง อุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยสามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์แสดงได้ในภาพที่ 2.1

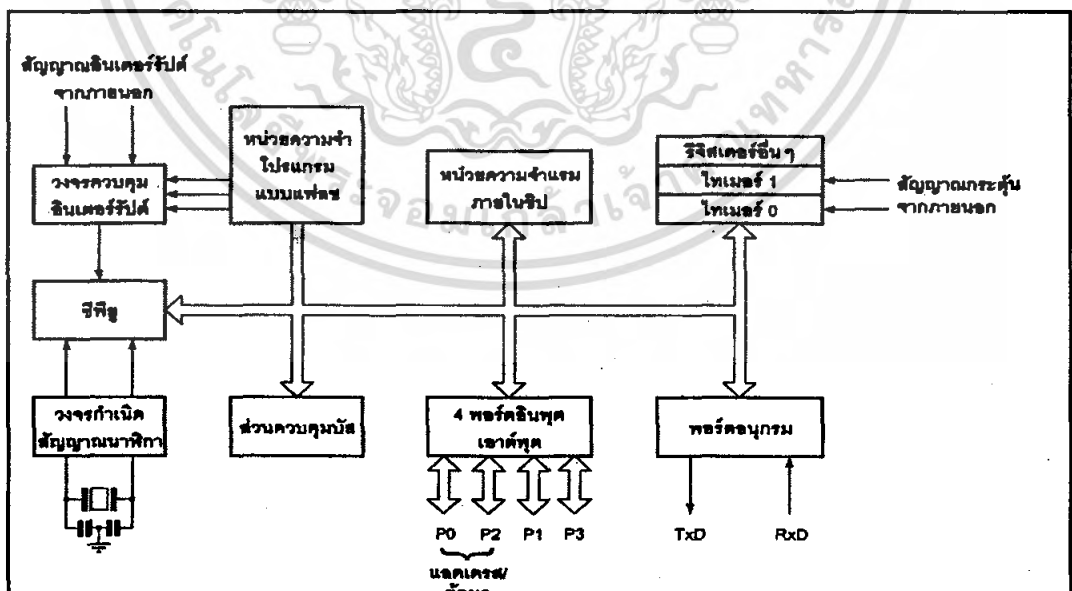


ภาพที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 อนุกรม AT89C51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นแบบแรม
- ขาพอร์ดเป็นแบบกึ่งสองทิศทาง (quasi-bidirectional) ใช้งานได้เป็นทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/คอนไดเตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้อย่างน้อย 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ในชิป

ในภาพที่ 2.2 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C51 จะเห็นได้ว่าโครงสร้างของ AT89C51 จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามาหากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว



ภาพที่ 2.2 โครงสร้างพื้นฐานไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกันโดยมีรายละเอียดขั้นต้นดังนี้

ขา V_{cc} ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่องลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอกและ (A0-A7) ขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อกับแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าไทเมอร์ 2 และขา P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในหน่วยความจำแฟลชภายใน

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่องลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่องลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่ใช้งานพิเศษดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{INT0}$

P3.3 เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{INT1}$

P3.4 เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.5 เป็นขาอินพุตสำหรับรับสัญญาณอินเทอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ \overline{WR} ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ \overline{RD} ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขา รีเซต (Reset) ใช้ในการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซ็ตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมกซ์ไซเคิล โดยที่วงจรถูกเปิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา $\overline{ALE/PROG}$ (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนี้ขานี้ยังเป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขา 2 ครั้งในแต่ละแมกซ์ไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีสัญญาณใด ๆ ออกมา

ขา EAV_{pp} (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อกับหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขาตัวยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.1.3 การทำงานของพอร์ต

1) การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่ต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดเป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟลสที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง

ส่งผลให้ค่าพอร์คนั้นมีค่าลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ค แล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์คอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ควรกำหนดให้ทำงานในสภาวะลอจิก "0" จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

2) การใช้งานเป็นพอร์คเอาต์พุต

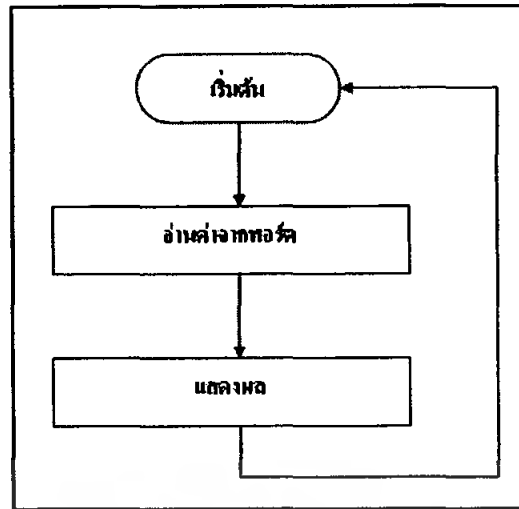
โดยปกติแล้วขาพอร์คจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือเมื่อต้องการส่งข้อมูล "0" ไปยังวงจรแลตช์ซึ่งก็จะส่งต่อไปขับเฟส ทำให้เฟสทำงานที่ขาพอร์คที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล "1" ไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ขาพอร์คที่เชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก "1" ที่ขาพอร์คนั้น ซึ่งจะคล้ายกับกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มี การอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

ในโครงการนี้ใช้ไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C51 ซึ่งเป็นแบบแฟลช มีพอร์คให้ใช้งานทั้งสิ้น 4 พอร์ค คือ พอร์ค 0 ถึง พอร์ค 3 แต่ละพอร์คมีขนาด 8 บิต เป็นพอร์คแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตแล้วเอาต์พุตได้ ทุกพอร์คของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช มีวงจรขับตลอดจนบัฟเฟอร์อินพุต

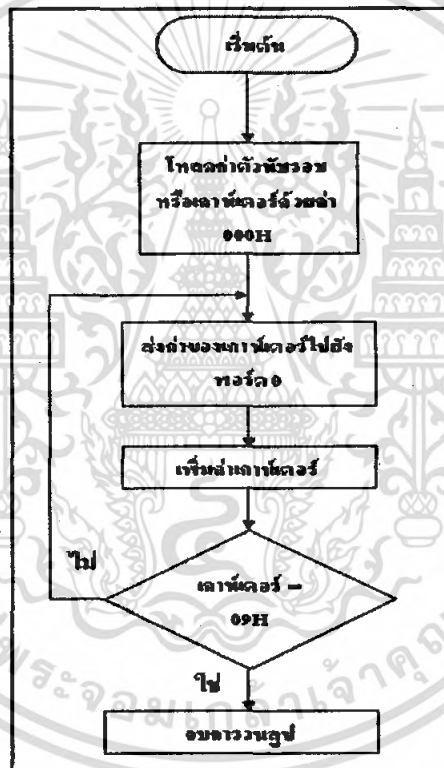
2.1.4 การทำงานของโปรแกรมลูปและการหน่วงเวลา

โปรแกรมลูป (Loop routine) มีประโยชน์อย่างมากในการเขียนโปรแกรมควบคุมของไมโครคอนโทรลเลอร์ เนื่องจากในการทำงานจริงจะมีการกำหนดให้ไมโครคอนโทรลเลอร์ทำงานซ้ำหรือทำงานในลักษณะเดียวกันอยู่เสมอ อาทิ การรอกอยการกดสวิทช์ การแสดงผล เป็นต้น แทนที่จะเขียนโปรแกรมเหมือนๆ กันหลายๆ บรรทัด ก็สามารถทดแทนด้วยการใช้โปรแกรมลูป ทำให้ลดขนาดของโปรแกรมลงได้อย่างมาก

โพลวซาร์ตของโปรแกรมลูปอย่างง่าย ๆ แสดงในภาพที่ 2.3 โดยโปรแกรมจะเริ่มต้นการทำงานด้วยการอ่านค่าแล้วนำข้อมูลที่อ่านได้นี้ไปแสดงผล จากนั้นจึงวนกลับมาอ่านค่าอีกครั้ง นอกจากนั้นการสร้างโปรแกรมลูปอาจอาศัยค่าในรีจิสเตอร์กำหนดเป็นตัวนับจำนวนรอบของการทำงานหรือเรียกว่า ลูปเคาน์เตอร์ (loop counter) ดังแสดงตัวอย่างในโพลวซาร์ตแสดงในภาพที่ 2.4



ภาพที่ 2.3 โฟลวชาร์ตของโปรแกรมลูปอย่างง่าย



ภาพที่ 2.4 โฟลวชาร์ตของโปรแกรมลูปแบบใช้ลูปเคาน์เตอร์

สำหรับโปรแกรมหน่วงเวลา (delay routine) เป็นโปรแกรมประยุกต์ที่ใช้หลักการของโปรแกรมลูป เพียงแต่จุดมุ่งหมายของโปรแกรมหน่วงเวลาคือ การใช้เวลาให้หมดไปโดยไม่ส่งผลต่อข้อมูลหลักของโปรแกรม โปรแกรมหน่วงเวลาจะใช้ลูปเคาน์เตอร์ร่วมกับคาบเวลาของสัญญาณนาฬิกาภายในไมโครคอนโทรเลอร์ในการกำหนดค่าเวลาที่ต้องการหน่วง โดยปกติแล้วผู้เขียนโปรแกรมมักจะเขียนโปรแกรมหน่วงเวลาเป็นค่าเวลาอ้างอิงไว้ค่าหนึ่ง อาจเป็น 1 มิลลิวินาทีหรือ 1 วินาที จากนั้นกำหนดค่าของรีจิสเตอร์เป็นตัวคูณ นำไปคูณเข้ากับค่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาอ้างอิง ก็จะสามารถสร้างการหน่วงเวลาให้แก่ระบบได้อย่างไม่มีข้อจำกัด และใช้เนื้อที่ของหน่วยความจำโปรแกรมสำหรับโปรแกรมหน่วงเวลานี้เพียง 1 หรือ 2 โปรแกรมย่อยเท่านั้น

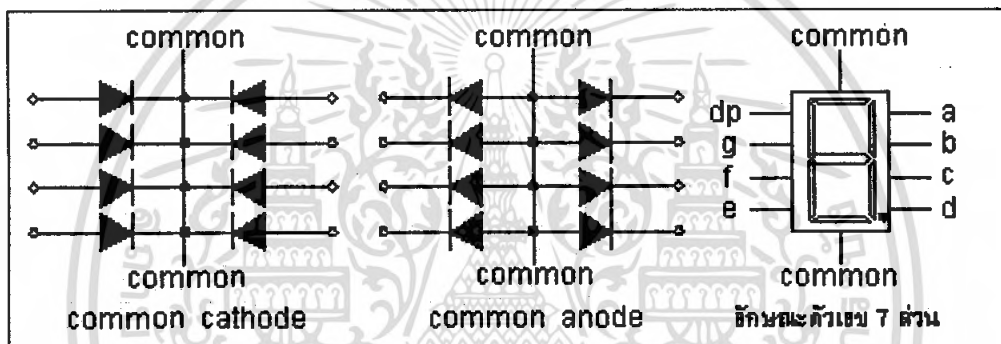
2.2 อุปกรณ์ในกล่องควบคุม

2.2.1 LED ตัวเลข 7 ส่วน

1) โครงสร้าง

LED ตัวเลข 7 ส่วนเป็นอุปกรณ์ที่นำ LED ดวงเดียวมาจัดเป็นรูปแบบการแสดงผลเป็นตัวเลข ปกติจะแสดงตัวเลข 0-9 พร้อมทั้งมีจุดที่จะแสดงเป็นทศนิยม (decimal point)

LED ตัวเลข 7 ส่วนมี 2 ลักษณะ คือ แบบ Cathode ร่วม (Common Cathode) และ แบบ Anode ร่วม (Common Anode) ดังภาพที่ 2.5



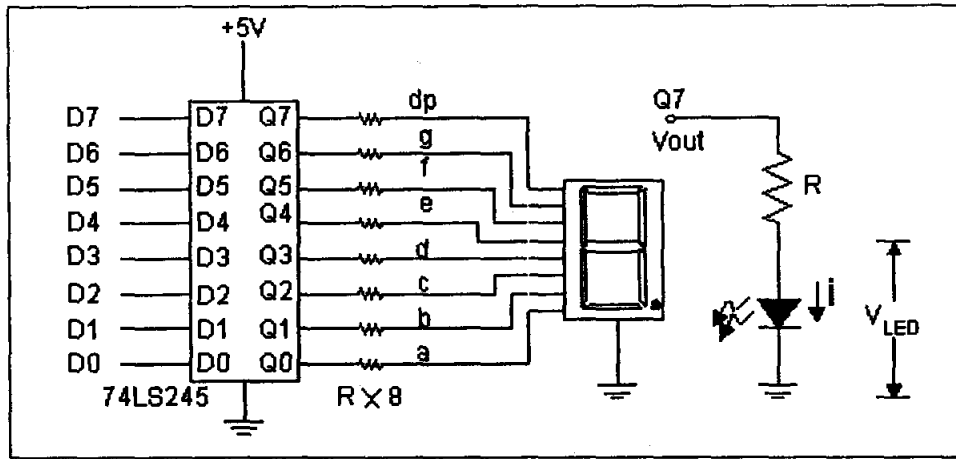
ภาพที่ 2.5 ลักษณะของ LED ตัวเลข 7 ส่วน

2) การหาตำแหน่งขาของ LED ตัวเลข 7 ส่วน

จะวัดได้โดยใช้โอห์มมิเตอร์โดยถือหลักว่า หากให้กระแสไฟฟ้าไหลจาก Anode ไป Cathode ของ Segment ใด ได้แล้ว Segment นั้นจะสว่าง

3) การนำ LED ไปใช้แสดงผล

(1) แบบ Common Cathode แสดงได้ดังภาพที่ 2.6

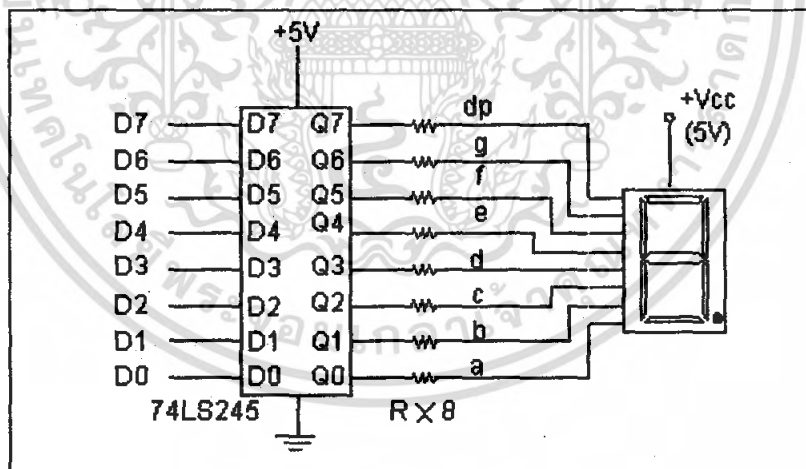


ภาพที่ 2.6 LED ตัวเลข 7 ส่วนแบบแคโทดร่วม

ในแต่ละ Segment จะต้องมีตัวต้านทานจำกัดกระแส โดยผู้ออกแบบจะต้องกำหนดว่าในแต่ละ Segment จะให้กระแสไฟฟ้าไหลได้เท่าใด และเอาต์พุตที่ขับจะต้องให้ลอจิก "1" และ V_{out} ของไอซีที่ขับขณะเป็นลอจิก "1" จะต้องเป็นเท่าใด

$$R = (V_{out} - V_{LED}) / i$$

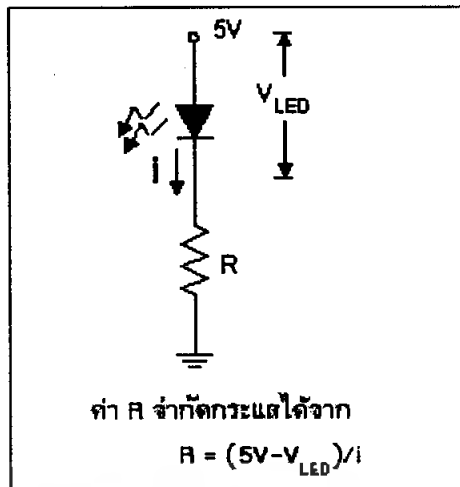
(2) แบบ Common Anode แสดงได้ดังภาพที่ 2.7



ภาพที่ 2.7 LED ตัวเลข 7 ส่วนแบบแอนโนดร่วม

LED ตัวเลข 7 ส่วนแบบ Common Anode จะต้องขับด้วยการให้ลอจิก "0" แก่ Anode โดยมีตัวต้านทาน R จำกัดกระแส

การขับ LED แต่ละ Segment ให้ติด เอาต์พุตของ Segment นั้นจะต้องเป็นลอจิก "0" โดย Anode ของ LED ต่อกับแหล่งจ่ายไฟ 5 V แสดงได้ดังภาพที่ 2.8



ภาพที่ 2.8 วงจรขับ LED

อย่างไรก็ตาม การหาค่า R ไม่ว่าจะจำกัดกระแสให้แก่ LED แบบ Common Anode หรือ Common Cathode ก็ตาม จะใช้หลักการเกี่ยวกับการจำกัดกระแส LED แบบดวงเดียว

2.2.2 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารแบบอนุกรมฟูลดูพล็กซ์ 1 ชุด (วงจรสื่อสารแบบอนุกรมฟูลดูพล็กซ์หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ต 3 ขาคือ P3.0 เป็นขารับข้อมูลขึ้นเข้าหรือ RxD และขา P3.1 เป็นขาส่งข้อมูลออกหรือ TxD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232, RS-442 หรือ RS-485 โดยใช้ไอซีพิเศษที่กำหนดหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

ในการทำงานของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังนี้

1) รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99F ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษหรือ SFR มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลเหล่านั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านขา TxD หรือ ขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไปสำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (serial port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิตมีแอดเดรสอยู่ที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิตมีละเอียดการทำงานดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0-SM1 (Serial port mode bit 0-1): ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2: ใช้ในการเอ็นเอเบิลการสื่อสารในแบบมัลติโพรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกติฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกติฟถ้ายังไม่ได้รับบิตหยุดส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (enable serial reception): ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้เป็น "1"

TB8: ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมดที่ 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8: ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมดที่ 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM 2 เป็น "0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (stop bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งานบิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (transmit interrupt flag): ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้จะต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (receive interrupt flag) : ใช้แสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลส่งเข้าสู่พอร์ตอนุกรม สามารถเซตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการสามารถรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

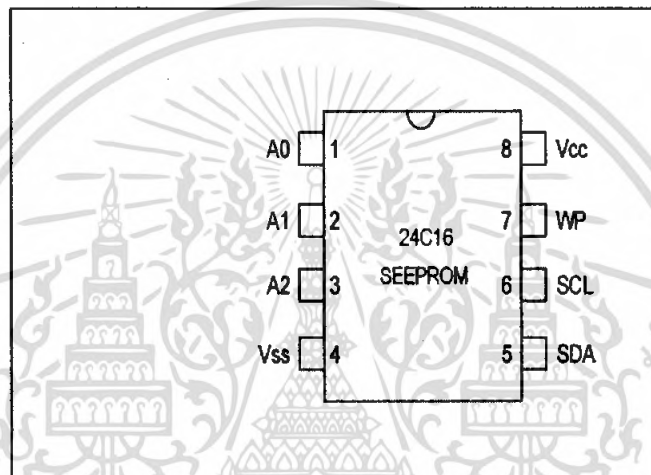
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตนี้จะมีการเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้จะต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

2.2.3 การเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำอีอีพรอมอนุกรม EEPROM

1) หน่วยความจำอีอีพรอมอนุกรม

หน่วยความจำอีอีพรอมอนุกรมเบอร์ 24C16 เป็นหน่วยความจำแบบนอน-โวลตาอิล (non-volatile) หมายความว่า สามารถเก็บรักษาข้อมูลอยู่ได้โดยไม่ต้องจ่ายไฟเลี้ยง สามารถเขียน-อ่าน-ลบได้ด้วยสัญญาณไฟฟ้า ขาของหน่วยความจำอีอีพรอมสามารถแสดงได้ดังภาพที่ 2.9



ภาพที่ 2.9 ขาของหน่วยความจำอีอีพรอม

มีคุณสมบัติทางเทคนิคโดยสรุปดังนี้

- ขนาดของหน่วยความจำ 16 กิโลบิต หรือ 2 กิโลไบต์
- แรงดันใช้งาน 2.5-5.5V
- กระแสขณะสแตนด์บายสูงสุด 100mA
- กระแสขณะอ่านข้อมูลสูงสุด 1mA
- กระแสขณะเขียนข้อมูลสูงสุด 3mA
- ความถี่ของสัญญาณนาฬิกาต่ำสุด-สูงสุด 100-400 kHz
- รอบในการเขียนข้อมูล 1 ล้านครั้ง
- ระยะเวลาการเก็บข้อมูล มากกว่า 200 ปี
- ระยะเวลาในการเขียนสูงสุด 5 มิลลิวินาที
- มีฟังก์ชันป้องกันการเขียนข้อมูลทับ

หน้าที่การทำงานของขาต่างๆของ 24C16 มีดังนี้

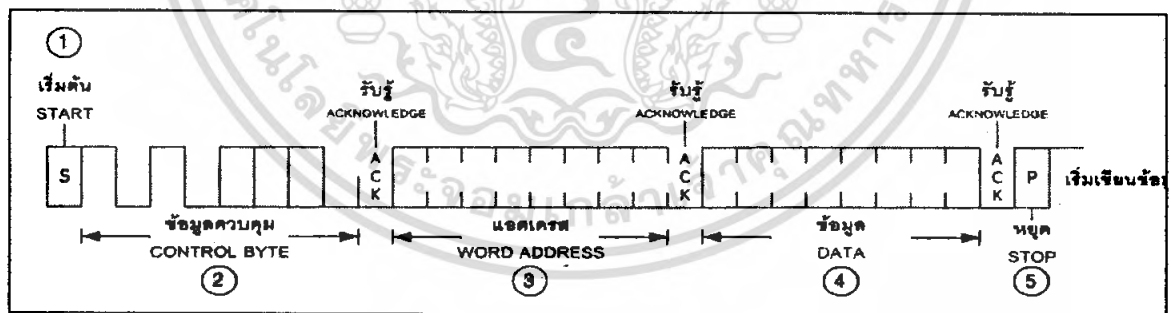
- V_{cc} (ขาที่ 8) เป็นขาป้อนไฟเลี้ยงบวก
- WP (Write project : ขาที่ 7) ถ้าขานี้เป็นลอจิก “1” จะสามารถอ่านได้อย่างเดียว
- SCL (Serial Clock : ขาที่ 6) เป็นขาอินพุตรับสัญญาณนาฬิกา
- SDA (Serial Data : ขาที่ 5) เป็นขาข้อมูลอนุกรม
- V_{ss} (ขาที่ 4) เป็นขาป้อนไฟเลี้ยงลบหรือกราวด์
- A0-A2 (ขาที่ 1-3) เป็นขากำหนดแอดเดรสให้ต่อลงกราวด์ทั้งหมด หรือปล่อยลอยไว้

2) รูปแบบการอ่านและเขียนข้อมูลของอีอีพรอม

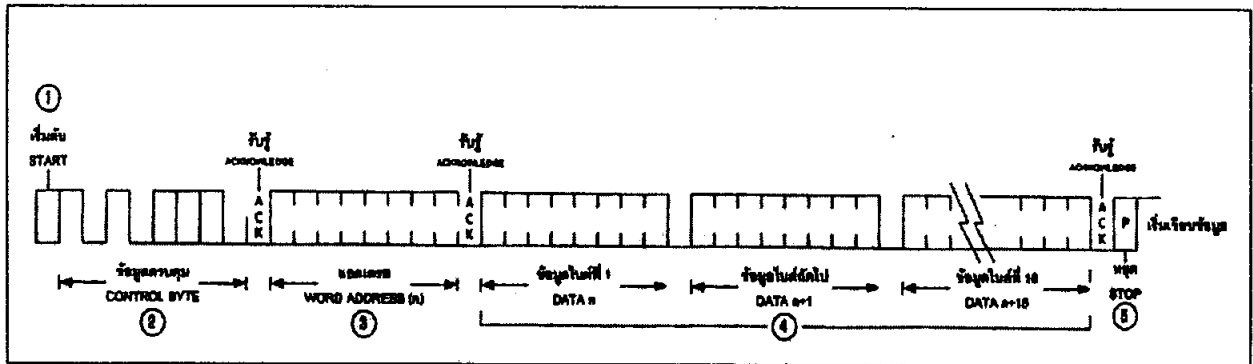
(1) การเขียนข้อมูลของอีอีพรอม มี 2 รูปแบบคือ

1. การเขียนแบบไบต์ (Byte Write) มีรูปแบบการเขียนแสดงในภาพที่ 2.10
2. การเขียนแบบเพจ (Page Write) มีรูปแบบการเขียนแสดงในภาพที่ 2.11

การเข้าถึงข้อมูลของอีอีพรอมนี้ทำได้ในลักษณะของบล็อกข้อมูล ซึ่งแบ่งออกเป็น 8 บล็อกคือ บล็อก 0 ถึง 7 ในแต่ละบล็อกเก็บข้อมูลได้ 256 ตำแหน่ง ตำแหน่งละ 8 บิต และทั้ง 8 บล็อกนี้จะมีแอดเดรสที่เหมือนกันคือ แอดเดรส 0x00 ถึง 0xFF การเลือกตำแหน่งของบล็อกสามารถกำหนดได้ที่บิต 0 ถึง 2 ของข้อมูลควบคุม (Control Byte) ดังแสดงในภาพที่ 2.9



ภาพที่ 2.10 รูปแบบการเขียนข้อมูลแบบไบต์ของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24C16



ภาพที่ 2.11 รูปแบบการเขียนข้อมูลแบบเพจของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24C16

1. การเขียนแบบไบต์ (Byte Write)

พิจารณาภาพที่ 2.10 ไมโครคอนโทรลเลอร์ซึ่งเป็นอุปกรณ์มาสเตอร์จะทำหน้าที่ส่งข้อมูลและสัญญาณนาฬิกา ส่วนอีอีพรอมจะเป็นอุปกรณ์สเลฟทำหน้าที่รับข้อมูล การเขียนข้อมูลแบ่งออกเป็น 5 ช่วงด้วยกันดังนี้

ช่วงที่ 1 เป็นการเริ่มการติดต่อกับอุปกรณ์สเลฟ ในช่วงนี้ตัวมาสเตอร์จะสร้างสัญญาณเริ่มต้นขึ้น เพื่อให้อุปกรณ์สเลฟที่ต่ออยู่เตรียมพร้อมที่จะทำงาน

ช่วงที่ 2 อุปกรณ์มาสเตอร์ส่งข้อมูลควบคุมออกไปที่สายข้อมูลอนุกรม ซึ่งในข้อมูลควบคุมนี้ จะประกอบไปด้วยข้อมูลต่างๆ ดังในภาพที่ 2.10 จากนั้นรอรับสัญญาณรับรู้จากอุปกรณ์ สเลฟ ซึ่งในที่นี้คืออีอีพรอม

ช่วงที่ 3 ไมโครคอนโทรลเลอร์ส่งแอดเดรสที่ต้องการเขียนไปให้อีอีพรอมแล้วรอรับสัญญาณรับรู้

ช่วงที่ 4 ไมโครคอนโทรลเลอร์ส่งข้อมูลที่ต้องการเขียนไปให้อีอีพรอม แล้วรอรับสัญญาณรับรู้ ในช่วงนี้อีอีพรอมจะนำข้อมูลที่รับได้ไปเก็บไว้ที่บัฟเฟอร์ภายในก่อน แต่ยังไม่ทำการเขียนข้อมูล

ช่วงที่ 5 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุด เพื่อหยุดการติดต่อกับอีอีพรอม

เมื่อผ่านช่วงที่ 5 ไปแล้วอีอีพรอมจะเริ่มเขียนข้อมูลลงในหน่วยความจำจริงๆทันที โดยนำข้อมูลที่เก็บไว้ในบัฟเฟอร์ภายในเขียนลงไปยังแอดเดรสที่กำหนดไว้ในช่วงที่ 3 จะใช้เวลาประมาณ 5 มิลลิวินาที ดังนั้นถ้าจะเขียนข้อมูลไบต์ถัดไป จะต้องหน่วงเวลาเพื่อให้การเขียนข้อมูลเสร็จสมบูรณ์ก่อน

โฟลวชาร์ตของกระบวนการตรวจสอบสถานะการเขียนข้อมูลในอีอีพรอมแสดงดังภาพที่ 2.12

2. การเขียนแบบเพจ (Page Write)

มี 5 ช่วงเช่นกัน ดังมีรายละเอียดต่อไปนี้

ช่วงที่ 1 ถึง 3 เหมือนกับการเขียนแบบไบต์ทุกประการ

ช่วงที่ 4 จะแตกต่างจากแบบแรกเล็กน้อยคือ สามารถส่งข้อมูลที่ต้องการเขียนในลักษณะต่อเนื่องกันได้ถึง 32 ข้อมูล (ข้อมูล n ถึง $n+32$ จากรูปที่ 2.13) ต่อการเขียน 1 เพจ โดยค่าของแอดเดรสจะเพิ่มขึ้นโดยอัตโนมัติ ในกรณีที่ต้องส่งข้อมูลมากกว่า 32 ข้อมูล ตำแหน่งของข้อมูลตัวที่เกินจะถูกเขียนทับลงไปตำแหน่งแรกทันที กล่าวคือ ถ้าป้อนข้อมูล 34 ค่า ข้อมูลตัวที่ 33 จะเขียนทับข้อมูลตัวแรก และข้อมูลตัวที่ 34 จะเขียนทับข้อมูลตัวที่สอง ดังนั้นในการป้อนแบบเพจจะต้องระวังในเรื่องของความยาวข้อมูลด้วยเสมอ

ช่วงที่ 5 เช่นเดียวกับแบบแรกคือ ไมโครคอนโทรลเลอร์จะส่งสัญญาณหยุดเพื่อหยุดการติดต่อ หลังจากนั้นกระบวนการเขียนข้อมูลของอีอีพรมจึงเริ่มต้นขึ้น ซึ่งต้องหน่วงเวลาก่อนที่จะส่งข้อมูลชุดต่อไปเช่นกัน

(2) การตรวจสอบสถานะการเขียนข้อมูล

นอกจากใช้การหน่วงเวลาเพื่อรอให้อีอีพรมเขียนข้อมูลเสร็จแล้ว ยังมีอีกวิธีการหนึ่งคือ ตรวจสอบบิตรับรู้ (Acknowledge polling) มีข้อดีคือ สามารถกำหนดช่วงเวลาที่น่านอนได้ โดยไม่เปลี่ยนแปลงตามความเร็วของตัวหน่วยความจำเอง มีโฟลวชาร์ตแสดงการทำงานดังภาพที่ 2.14

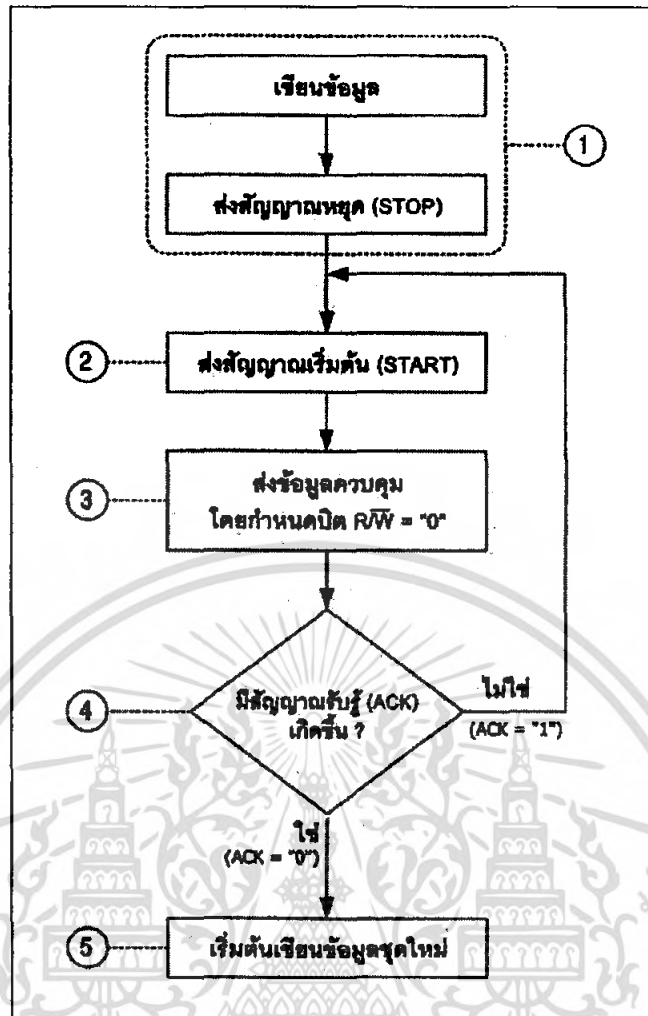
ขั้นตอนที่ 1 ในโฟลวชาร์ตคือช่วงที่ 1 ถึง 5 ของการเขียนข้อมูลตามปกติ

ขั้นตอนที่ 2 ในไมโครคอนโทรลเลอร์จะส่งสัญญาณเริ่มต้นไปให้อีอีพรม

ขั้นตอนที่ 3 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุมไปให้อีอีพรมโดยกำหนดให้บิต $R/W = 0$

ขั้นตอนที่ 4 เป็นขั้นตอนตรวจสอบว่าการเขียนข้อมูลอีอีพรมเสร็จสิ้นหรือยัง โดยตรวจสอบสัญญาณรับรู้หรือบิตรับรู้ที่ตัวอีอีพรมจะส่งออกมา โดยถ้าบิตรับรู้เป็น "1" แสดงว่ายังเขียนข้อมูลไม่เสร็จ แต่ถ้าบิตรับรู้เป็น "0" แสดงว่าการเขียนข้อมูลเสร็จสิ้นแล้ว

ขั้นตอนที่ 5 เมื่อการเขียนข้อมูลภายในของอีอีพรมเสร็จสิ้นแล้ว ไมโครคอนโทรลเลอร์จะสามารถเขียนข้อมูลชุดถัดไปได้ทันที



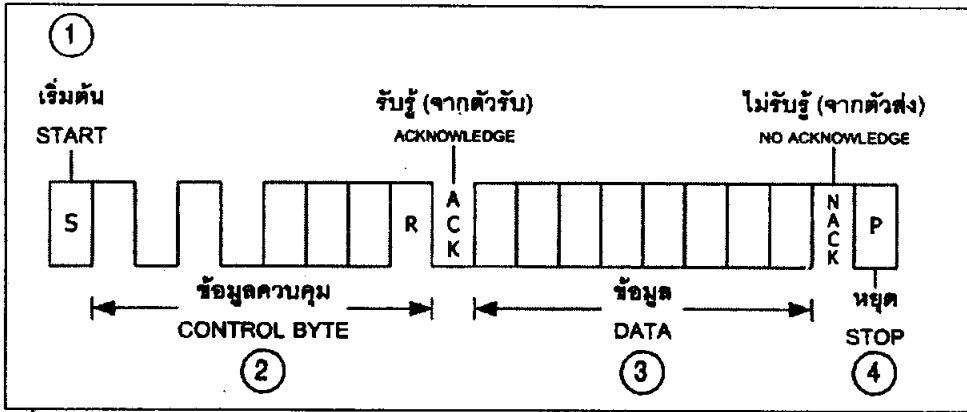
ภาพที่ 2.12 โฟลวชาร์ตของกระบวนการตรวจสอบสถานะการเขียนข้อมูลในออีพรวม

(3) การอ่านข้อมูล

ในการอ่านข้อมูลนี้ ไมโครคอนโทรลเลอร์หรืออุปกรณ์มาสเตอร์จะเป็นตัวรับข้อมูล ส่วนออีพรวมซึ่งเป็นอุปกรณ์สเลฟจะเป็นตัวส่งข้อมูล ในการอ่านข้อมูลมีด้วยกัน 3 แบบ คือ

1. แบบอ่านแอดเดรสปัจจุบัน (Current Address Read)

การอ่านแบบนี้จะเป็นการอ่านข้อมูลจากแอดเดรสปัจจุบัน กล่าวคือ หลังจากเขียนหรืออ่านข้อมูลชุดล่าสุดเสร็จสิ้นแล้ว ตัวชี้แอดเดรสภายในออีพรวม จะเพิ่มค่าขึ้น 1 ค่าโดยอัตโนมัติ ดังนั้นถ้าการอ่านหรือเขียนในครั้งที่แล้วกระทำที่แอดเดรส n การอ่านลักษณะนี้จะอ่านข้อมูลที่แอดเดรส $n+1$ ในภาพที่ 2.13 แสดงรูปแบบของการอ่านข้อมูลแบบนี้



ภาพที่ 2.13 รูปแบบการอ่านข้อมูลแบบแอดเดรสปัจจุบันของหน่วยความจำเบอร์อียิปทอม

ช่วงการทำงานของการอ่านข้อมูลแบบอ่านแอดเดรสปัจจุบันมีดังนี้

ช่วงที่ 1 ตัวมาสเตอร์ซึ่งในที่นี้คือไมโครคอนโทรลเลอร์จะส่งสัญญาณเริ่มต้นเพื่อติดต่อกับอียิปทอม

ช่วงที่ 2 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุม โดยที่บิต RW = "1" จากนั้นรอรับสัญญาณรับรู้จากอียิปทอม

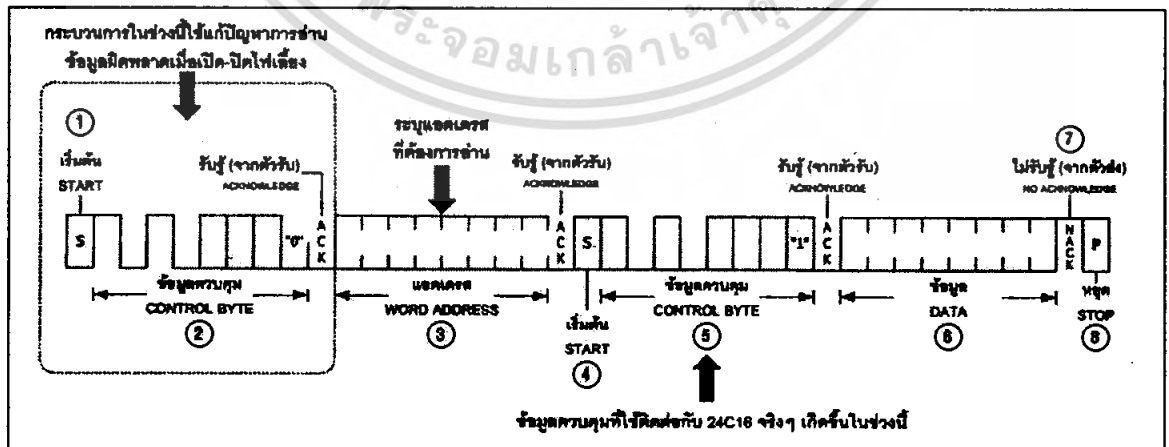
ช่วงที่ 3 อียิปทอม ส่งข้อมูลออกมา

ช่วงที่ 4 ไมโครคอนโทรลเลอร์จะส่งสัญญาณรับรู้ (NACK) ออกไป เพื่อให้ อียิปทอม หยุดการทำงาน แล้วทำให้สายข้อมูลอนุกรม เป็น "1"

ช่วงที่ 5 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุดเพื่อหยุดการติดต่อ

2. การอ่านแบบสุ่ม (Random Read)

เป็นการอ่านข้อมูลแบบเจาะจงแอดเดรส กล่าวคือ ถ้าต้องการอ่านข้อมูลที่ตำแหน่งใด ก็ให้ระบุแอดเดรสของอียิปทอมที่ต้องการอ่านได้ที่เวิร์ด แอดเดรส (Word Address) มีรูปแบบแสดงในภาพที่ 2.14



ภาพที่ 2.14 รูปแบบการอ่านข้อมูลแบบสุ่มหรือแบบเจาะจงแอดเดรสของอียิปทอม

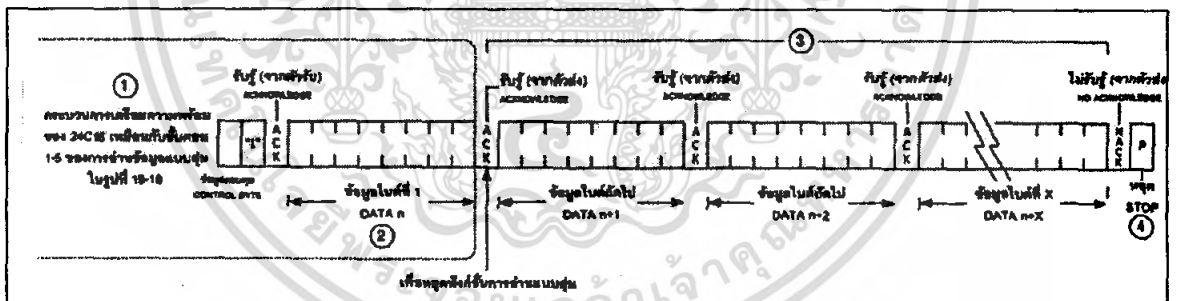
- ช่วงการทำงานมีดังนี้
- ช่วงที่ 1 ไมโครคอนโทรลเลอร์ส่งสัญญาณเริ่มต้นเพื่อติดต่อกับอุปกรณ์สเลฟ
- ช่วงที่ 2 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุมออกไป โดยต้องให้บิต R/W = "0" จากนั้นรอรับสัญญาณรับรู้อีพีรอม
- ช่วงที่ 3 ไมโครคอนโทรลเลอร์ส่งค่าแอดเดรส จากนั้นรอรับสัญญาณรับรู้จากอีพีรอม
- ช่วงที่ 4 ไมโครคอนโทรลเลอร์ส่งสัญญาณเริ่มต้นอีกครั้งหนึ่ง เพื่อระงับการเขียนข้อมูล เนื่องจากในช่วงที่ 2 บิต R/W = "0"
- ช่วงที่ 5 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุมออกไป โดยที่บิต R/W = "1" จากนั้นรอรับสัญญาณรับรู้จากอีพีรอม
- ช่วงที่ 6 อีพีรอมจะส่งข้อมูลในตำแหน่งที่ระบุไว้ในเวิร์ด แอดเดรสมาให้กับไมโครคอนโทรลเลอร์

ช่วงที่ 7 ไมโครคอนโทรลเลอร์ส่งสัญญาณรับรู้ (NACK) ออกไปเพื่อให้อีพีรอมหยุดทำงาน แล้วปล่อยให้สายข้อมูลอนุกรม เป็น "1"

ช่วงที่ 8 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุด

3. การอ่านแบบเรียงลำดับ (Sequential Read)

การอ่านแบบนี้จะเป็นการอ่านข้อมูลออกจากอีพีรอมแบบเรียงกันไปจากแอดเดรสเริ่มต้นจนถึงสุดท้ายแล้ววนใหม่ มีรูปแบบการอ่านแสดงในภาพที่ 2.15



ภาพที่ 2.15 รูปแบบการอ่านข้อมูลแบบเรียงลำดับของอีพีรอม

ช่วงการทำงานมีดังนี้

ช่วงที่ 1 มีการทำงานเหมือนกับการอ่านแบบสุ่มทุกประการ

ช่วงที่ 2 เป็นช่วงที่อีพีรอมส่งข้อมูลมาให้กับไมโครคอนโทรลเลอร์เรียงลำดับกันไปโดยตัวชี้แอดเดรสภายในของอีพีรอมจะเพิ่มค่าขึ้นทีละหนึ่งค่า จากแอดเดรสที่กำหนดไว้ไปจนถึงแอดเดรสสุดท้ายของบล็อก 7 จากนั้นก็จะกลับไปชี้ข้อมูลที่แอดเดรส 0x00 ของบล็อก 0 ใหม่ ดังนั้นการอ่านข้อมูลแบบนี้จะสามารถอ่านข้อมูลได้อย่างไม่มีวันจบ จนกว่าตัวมาสเตอร์จะสร้างสัญญาณรับรู้ (NACK)

ช่วงที่ 3 ไมโครคอนโทรลเลอร์ส่งสัญญาณรับรู้ (NACK) ออกไปเพื่อให้อีอีพรอมหยุดทำงาน แล้วปล่อยให้สายข้อมูลนอนกรม เป็น "1"

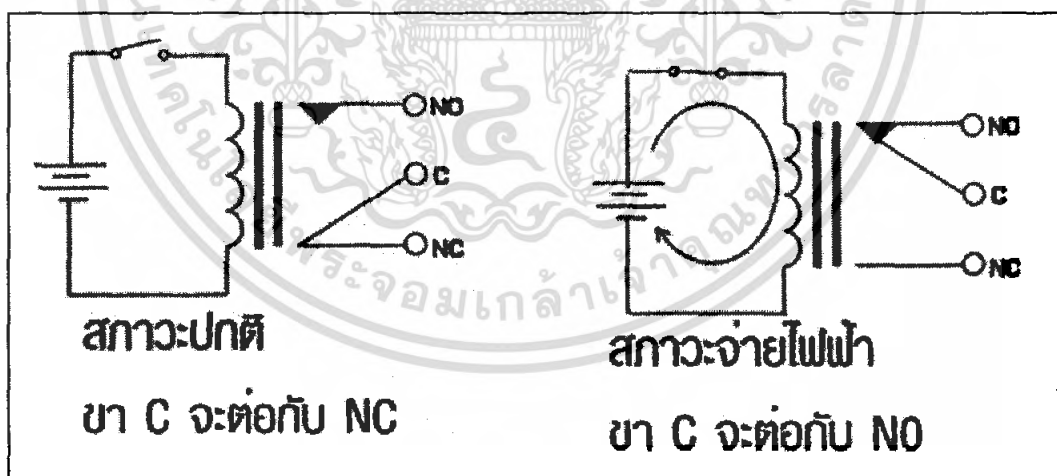
ช่วงที่ 4 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุด

2.2.4 รีเลย์ (Relay)

คือ อุปกรณ์อิเล็กทรอนิกส์ ที่ทำหน้าที่ ดัด-ต่อวงจร คล้ายกับสวิตช์ โดยใช้หลักการหน้าสัมผัส และการที่จะให้รีเลย์ทำงานก็ต้องจ่ายไฟให้มันตามที่กำหนด เพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ รีเลย์จะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิด และตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้ตัวรีเลย์ รีเลย์ก็จะกลายเป็นวงจรเปิด ไฟที่เราใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจาก เพาเวอร์ซ ของเครื่องเรานั้นทันทีที่เปิดเครื่อง ก็จะทำให้รีเลย์ทำงาน

รีเลย์ เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์(Solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภท ได้แก่

1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magnetic contactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา
2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุมบางที่เรียกกันง่าย ๆ ว่า "รีเลย์"



ภาพที่ 2.16 การต่อรีเลย์ในสภาวะปกติและสภาวะจ่ายไฟฟ้า



ภาพที่ 2.17 รีเลย์เบอร์ HRS4 (H)

ซึ่งในโครงการนี้ใช้รีเลย์ประเภทรีเลย์ควบคุมเบอร์ HRS4 (H) เพื่อเป็นเสมือน สวิตช์ใช้ในการควบคุมมอเตอร์

2.3 อุปกรณ์ต่อพ่วง

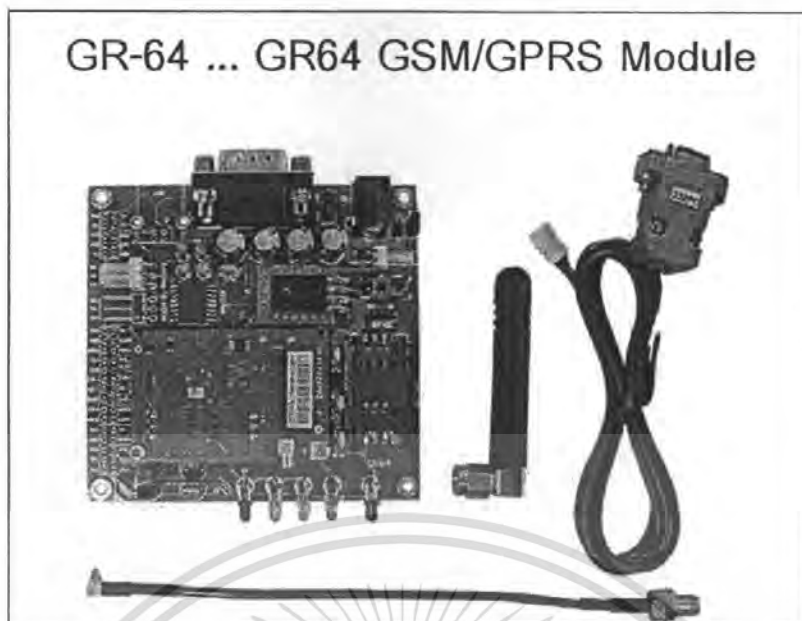
2.3.1 GR64

GR64 เป็นโมดูล GSM รองรับ 4 ความถี่ ได้แก่ GSM 850/900/1800/1900 MHz โดยปกติหากจะใช้งานตัว GR64 ซึ่งเป็นโมดูลที่มีขาต่อใช้งาน 60 pin จะต้องมีส่วนของ power supply ส่วนของ sim card และส่วนของเสาอากาศ

การใช้งานตัว GR64 นี้สามารถติดต่อโดยผ่านทาง พอร์ตอนุกรม ซึ่งรับคำสั่ง AT Command มาจาก MCS - 51

โครงการนี้ใช้ GR64 เป็นตัวส่งข้อความแจ้งเตือนเมื่อข้าวในตู้ใกล้จะหมด

GR-64 ... GR64 GSM/GPRS Module



ภาพที่ 2.18 บอร์ด GR-64

1) หลักการส่ง SMS

SMS: Short Message Service คือ การให้บริการส่งข้อความสั้นไปยังโทรศัพท์มือถือแบบดิจิทัล โดยแต่ละข้อความสามารถบรรจุตัวอักษรสูงสุดได้ 160 ตัวอักษร (อักษรภาษาอังกฤษ) นอกจากนี้ยังสามารถส่งข้อความไปที่เครื่อง Fax, PC หรือ Internet address อีกด้วย

ระบบ SMS ในระบบเครือข่ายโทรศัพท์มือถือ รองรับโดยระบบ GSM (Global System for Mobile Communication), TDMA (time division multiple access) และ CDMA (code division multiple access)

เมื่อ SMS ถูกส่งจากโทรศัพท์มือถือเครื่องหนึ่ง ข้อความนั้นจะถูกส่งไปที่ Short Message Service Center (SMSC) จากนั้นจึงจะส่งไปยังโทรศัพท์มือถือเครื่องรับอีกทอดหนึ่ง โดยมีกระบวนการดังนี้

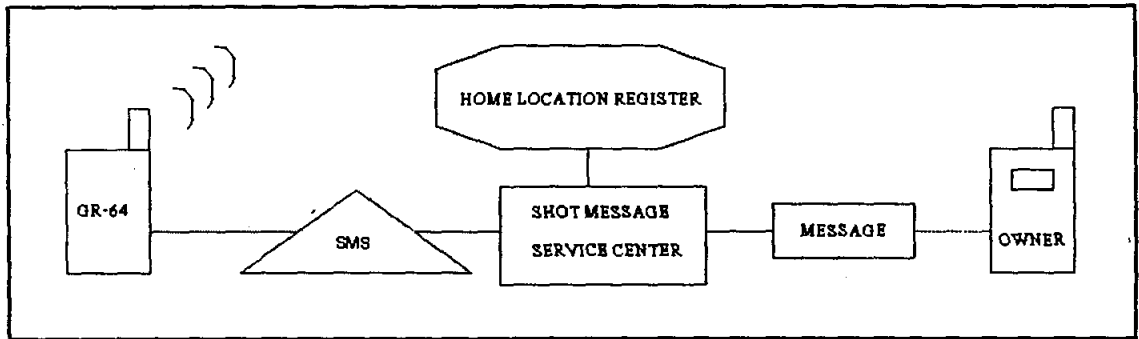
1.1) SMSC จะส่ง SMS Request ไปยัง Home location registers (HLR) เพื่อหาตำแหน่งของผู้รับ

1.2) เมื่อ HLR ได้รับสัญญาณ Request ก็ส่งสถานะของผู้รับ (subscriber's status) กลับมายัง SMSC คือ

1.2.1) สถานะของเครื่องรับ Inactive หรือ Active

1.2.2) ตำแหน่งของเครื่องรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.19 การส่ง SMS

ถ้าสถานะของเครื่องรับเป็น Inactive แล้ว SMSC จะเก็บข้อความไว้ช่วงเวลาหนึ่ง และเมื่อใดที่เครื่องรับมีสถานะ Active แล้ว HLR จะส่ง SMS Notification ไปยัง SMSC และ SMSC ก็จะตอบรับข้อความ นั้นไว้ จากนั้น SMSC จะส่งผ่านข้อความไปในรูปแบบ Short Message Delivery Point-to-Point ไปยังระบบ บริการ โดยระบบจะทำการเรียกไปยังเครื่องรับ และถ้าเครื่องรับมีการตอบรับกลับมา ข้อความก็จะถูกส่งตามไป และ SMSC จะได้รับการตอบยืนยันว่า ข้อความได้ถูกรับโดยปลายทางเรียบร้อยแล้ว หลังจากนั้นข้อความจะมีสถานะเป็น SENT และจะไม่ถูกส่งอีก

การรับส่งข้อมูล SMS มี 2 โหมด คือ Text Mode และ PDU Mode (Protocol Description Unit Mode) การส่งข้อมูล Text Mode นั้นจะเป็นการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อนแล้ว ค่อยส่งข้อมูลใน PDU Mode

2) SERIAL COMMUNICATION

Serial Communication คือ การส่งข้อมูลระหว่างอุปกรณ์ผ่านสายสัญญาณ โดยส่งผ่านข้อมูลไปที่ละบิตต่อเนื่องกัน แบ่งได้เป็น 2 แบบ คือ

2.1) Synchronous มีการส่งสัญญาณ clock ไปพร้อมกับข้อมูล

2.2) Asynchronous ไม่มีการส่งสัญญาณ clock ไปกับข้อมูล

ในการสื่อสารส่งข้อมูลในระยะไกลนั้น จำเป็นต้องใช้ Data Communication Equipment (DCE)

อย่างเช่น Modem เพื่อให้ส่งข้อมูลได้ไกลขึ้น แต่สำหรับการสื่อสารในระยะไกลไม่เกิน 50 ฟุต ไม่จำเป็นต้องใช้ DCE เรียกว่า การสื่อสารแบบ Null-modem และในการติดต่อสื่อสารข้อมูลระหว่างอุปกรณ์ จำเป็นจะต้องมีมาตรฐานในการเชื่อมต่อวงจร ซึ่งมาตรฐานที่ใช้กันมากที่สุดก็คือ มาตรฐาน RS232-C

การสื่อสารข้อมูลแบบอนุกรมระหว่างคอมพิวเตอร์และอุปกรณ์อื่นๆ เกือบทั้งหมดเป็นแบบ Asynchronous ดังนั้นเราจะกล่าวถึงรายละเอียดเกี่ยวกับการสื่อสารข้อมูลแบบ Asynchronous เท่านั้น

3) ตัวอย่างการส่ง SMS

ในส่วนของการส่ง SMS นั้นจะประกอบด้วยข้อมูล พิกัด Latitude และ Longitude ในการส่งงานจะส่งผ่านทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่โดยส่งเป็นชุดคำสั่ง AT-COMMAND ซึ่งชุดคำสั่งที่ใช้จะเป็นดังนี้

AT+CMGF เป็นคำสั่งในการเลือกโหมดของ Message ที่จะส่ง

AT+CMGS เป็นคำสั่งสำหรับส่ง Message ไปยัง Address ที่เลือกไว้

การส่ง SMS นั้นจะใช้ไมโครคอนโทรลเลอร์ ในการส่งโดยจะส่งชุดคำสั่ง AT-COMMAND เข้าไปทางพอร์ตอนุกรมของโทรศัพท์เคลื่อนที่ที่ต่ออยู่กับไมโครคอนโทรลเลอร์ แต่ก่อนที่ส่งนั้นต้องทำการกำหนด อัตรา Baud Rate ให้ตรงกับโทรศัพท์เคลื่อนที่เสียก่อน หลังจากนั้นเราจะใช้คำสั่ง AT+CMGF เป็นคำสั่งในการเลือกโหมดของ Message ที่จะส่ง ซึ่งแบ่งออกเป็น 2 โหมดคือ Text Mode และ PDU Mode (Protocol Description Unit Mode) ถ้าส่งโดยใช้คำสั่ง

AT+CMGF = 1 ส่งแบบ Text Mode

AT+CMGF = 0 ส่งแบบ PDU Mode

หลังจากนั้นจะทำการส่งคำสั่ง AT+CMGS พร้อมกับบอกจำนวนของข้อความตามด้วย "ข้อความที่เราต้องการส่ง" ตามด้วย Ctrl+Z เพื่อบอกว่าสิ้นสุดข้อความแล้ว

AT+CMGF=0	(เซต PDU Mode)
OK	(โทรศัพท์ตอบกลับมาว่า "OK")
AT+CMGS=22	(ส่ง SMS โดยความยาว TPDU = 22 Octets)
>0011000A818073789975000AA0AE8329BFD4697D9EC3	(+CMGS: 64
"โทรศัพท์เคลื่อนที่ตอบกลับ <CR>")	

สามารถแยกข้อความได้เป็น 2 ส่วน คือ SMSC กับ TPDU ซึ่งมีลักษณะดังนี้

SMSC: 00

00: คือความยาวของเบอร์ SMSC = 0 คือให้ใช้ SMSC ที่เก็บอยู่ภายในเครื่อง

TPDU: 11000A8180041133420000AA0AE8329BFD4697D9EC3

1100 : คือสถานะของ SMS ที่ SMSC

0A : คือความยาวของเบอร์โทรศัพท์เคลื่อนที่ที่ต้องการส่ง SMS ไป = 10

81 : คือตัวบอกว่าเบอร์โทรศัพท์เคลื่อนที่เป็นแบบภายในประเทศ

8004113342 : คือเบอร์โทรศัพท์เคลื่อนที่ปลายทางที่ทำการเข้ารหัสแบบสลับ

แล้ว 084-011-3324

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0000 : คือลักษณะการเข้ารหัสของข้อมูล (มาตรฐาน GSM)

AA : คือตัวบอกลักษณะการเข้ารหัส

OA : คือความยาวข้อความก่อนแปลงรหัส = 10

E8329BFD4697D9EC3 : คือข้อความที่แปลงรหัสเป็นเลขฐาน 16 แล้ว ก่อนแปลงคือคำว่า "hellohello"

2.3.2 เครื่องรับเหรียญ

เครื่องรับเหรียญในโครงการนี้ใช้รุ่น Multicoin EU3 เป็นระบบ Micro Processor ที่สามารถตั้งโปรแกรมให้ รู้จักและแยกเหรียญได้ถึง 4 รูปแบบ

- ใช้ได้กับเหรียญ 1, 2, 5 และ 10 บาท
- มีระบบตรวจจับเหรียญเซ็นเซอร์ ป้องกันเหรียญปลอมได้
- ป้องกันการกระตุกเหรียญ หรือการพยายามโกงทุกชนิด
- สามารถต่อกับวงจรอิเล็กทรอนิกส์หรือไมโครคอนโทรเลอร์ได้
- มีสวิทช์ฟังก์ชันสำหรับปรับความเร็วของสัญญาณที่ออกมาได้

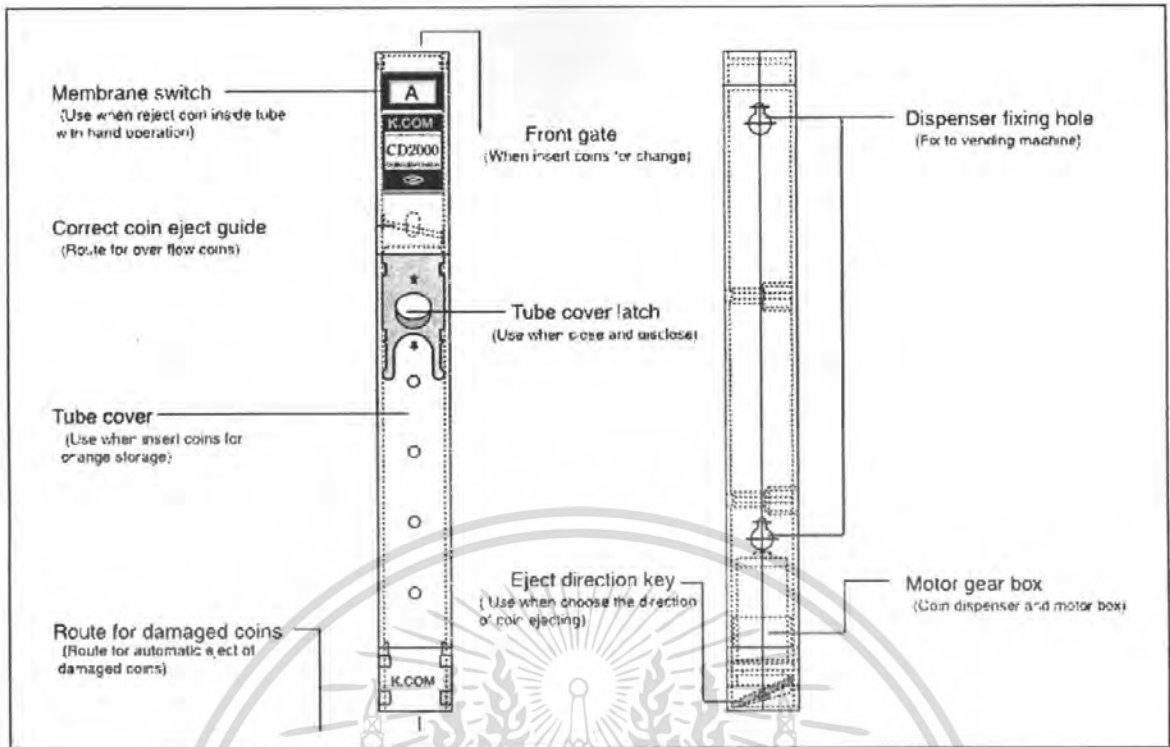


ภาพที่ 2.20 เครื่องรับเหรียญ

2.3.3 เครื่องทอนเหรียญ

เครื่องทอนเหรียญใช้รุ่น KCM-CD2000 ซึ่งสามารถทอนเหรียญได้หลายชนิด แต่ในโครงการนี้จะทดสอบแค่เหรียญ 5 บาทเพียงชนิดเดียว โดยช่องใส่เหรียญ 5 บาท บรรจุได้มากที่สุด 120 เหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



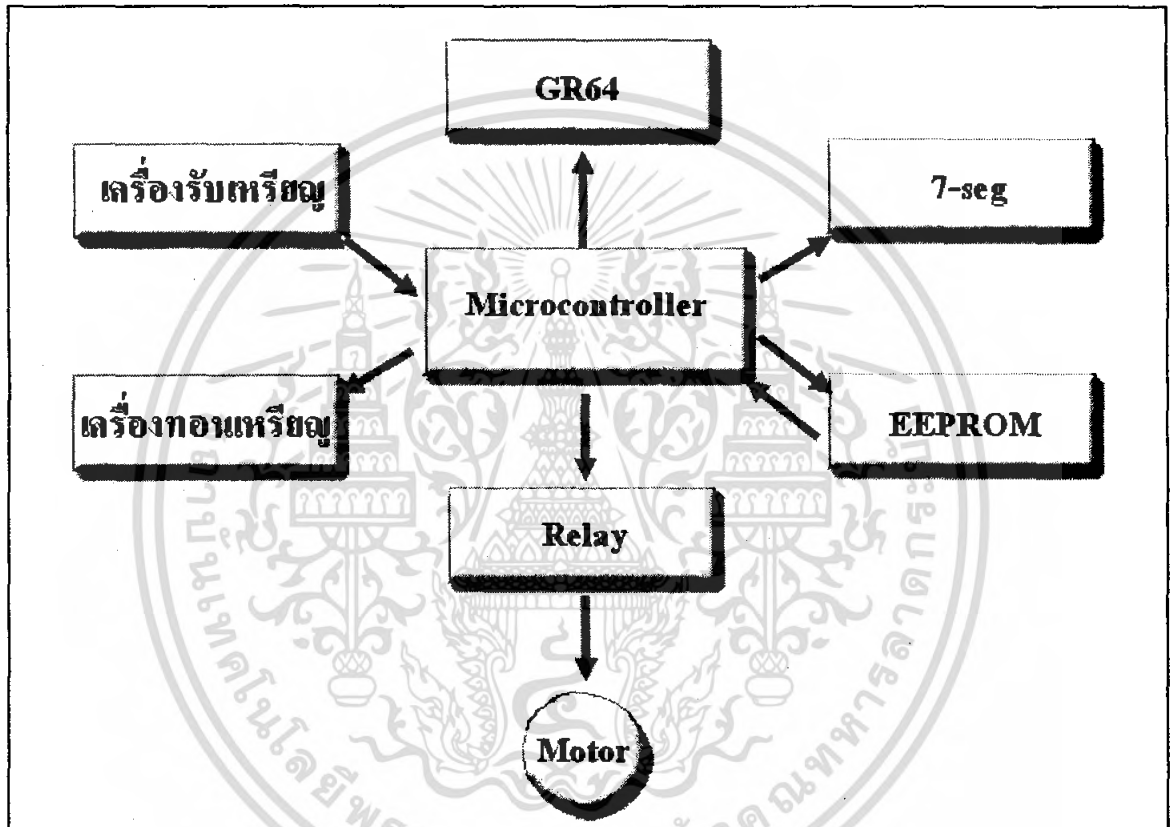
ภาพที่ 2.21 เครื่องทอนเหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

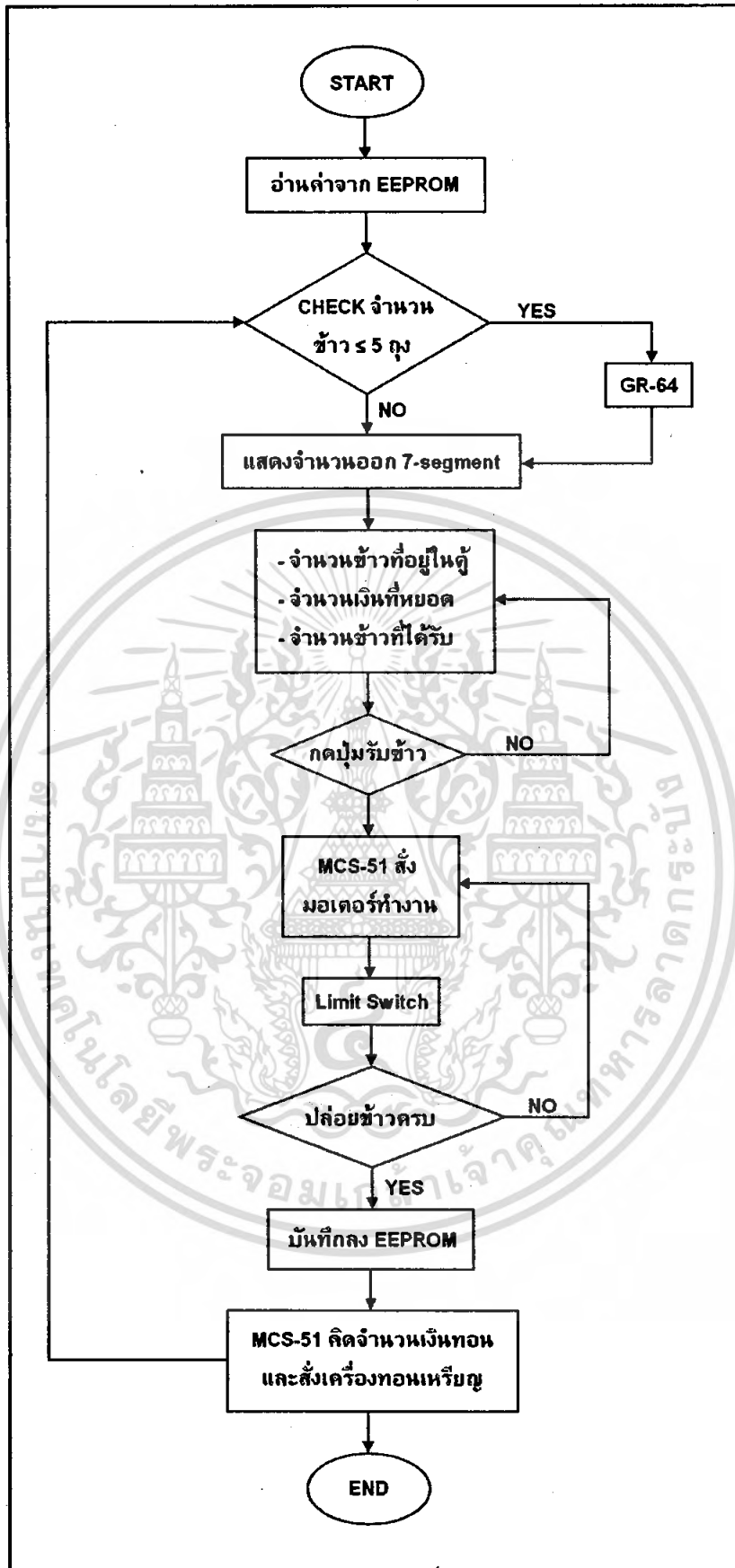
บทที่ 3

การออกแบบและการสร้าง

ในบทนี้จะกล่าวถึงการดำเนินงานโดยประมวลความรู้ที่ได้ศึกษามาทั้งหมด นำมาประกอบกัน และประยุกต์ใช้ออกมาเป็นผลงานที่สมบูรณ์ โดยสามารถแสดงการทำงาน เป็นบล็อกไดอะแกรม ได้ดังภาพที่ 3.1 และโฟลวชาร์ตการทำงานของเครื่องขายข้าวอัตโนมัติ แสดงได้ดังภาพที่ 3.2



ภาพที่ 3.1 บล็อกไดอะแกรมการทำงานของเครื่องขายข้าวอัตโนมัติ



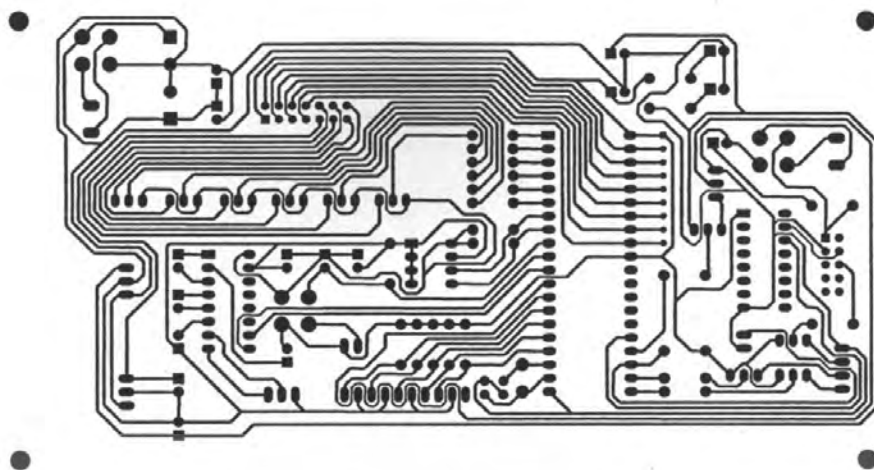
ภาพที่ 3.2 โฟลวชาร์ตการทำงานของเครื่องขายข้าวอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงการนี้ได้ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C51 เป็นตัวควบคุมการทำงาน ซึ่งมี 40 ขา โดยมีพอร์ตอินพุตเอาต์พุต 4 พอร์ต พอร์ตละ 8 บิต ได้แก่ PORT0 (ขา 32-39), PORT1 (ขา 1-8), PORT2 (ขา 21-28) และ PORT3 (ขา 10-17) ซึ่งในโครงการมีการต่อใช้งานขาและพอร์ตของไมโครคอนโทรลเลอร์ ดังแสดงได้ดังตารางที่ 3.1

ตารางที่ 3.1 การต่อใช้งานขาและพอร์ตของไมโครคอนโทรลเลอร์

ขา	พอร์ต	การใช้งาน	ขา	พอร์ต	การใช้งาน
1	P1.0	7-seg#1	21	P0.0	Motor1
2	P1.1	7-seg#2	22	P0.1	Motor2
3	P1.2	7-seg#3	23	P0.2	Limit SW
4	P1.3	7-seg#4	24	P0.3	En. EU3
5	P1.4	7-seg#5	25	P0.4	KCM
6	P1.5	7-seg#6	26	P0.5	-
7	P1.6	EEPROM	27	P0.6	-
8	P1.7	EEPROM	28	P0.7	-
9	-	SW reset	29	-	-
10	P3.0	Serial-Rx	30	-	-
11	P3.1	Serial-Tx	31	-	V _{DD}
12	P3.2	Signal EU3	32	P2.0	7-seg#dp
13	P3.3	SW up	33	P2.1	7-seg#g
14	P3.4	SW down	34	P2.2	7-seg#f
15	P3.5	SW refill	35	P2.3	7-seg#e
16	P3.6	SW price	36	P2.4	7-seg#d
17	P3.7	SW OK	37	P2.5	7-seg#c
18	-	XTAL	38	P2.6	7-seg#b
19	-	XTAL	39	P2.7	7-seg#a
20	-	Ground	40	-	V _{DD}



ภาพที่ 3.3 ลายทองแดงของแผงควบคุม



ภาพที่ 3.4 การลงอุปกรณ์ในแผงควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

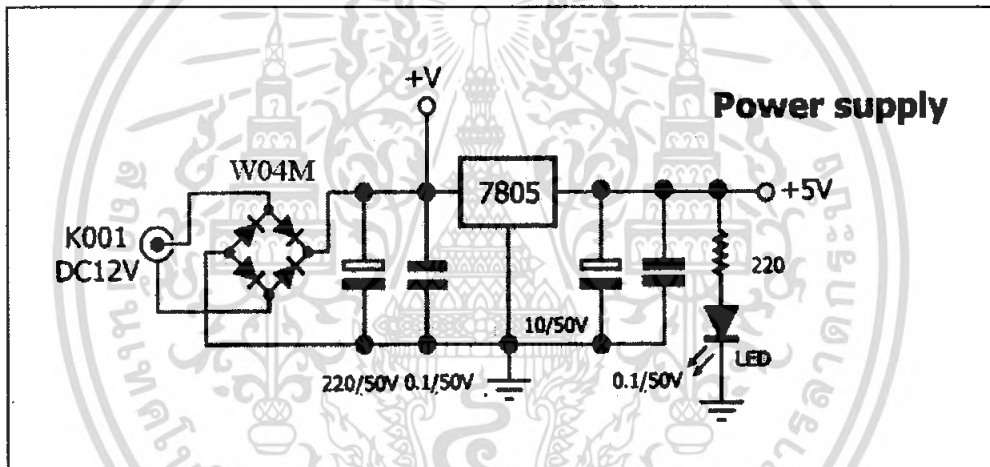
3.1 สร้างกล่องควบคุม

3.1.1 วงจรไฟเลี้ยง

ประกอบด้วย

- | | |
|------------------------|-------|
| 1. หม้อแปลง, 120/12V | 1 ตัว |
| 2. W04M, 400V, 1.5A | 1 ตัว |
| 3. MC7805CTG, 15V, 1A | 1 ตัว |
| 4. C, 220 μ F, 50V | 1 ตัว |
| 5. C, 10 μ F, 50V | 1 ตัว |
| 6. C, 0.1 μ F, 50V | 2 ตัว |
| 7. R, 220k Ω | 1 ตัว |
| 8. LED | 1 ตัว |

วงจรไฟเลี้ยงสามารถแสดงได้ดังภาพที่ 3.5



ภาพที่ 3.5 วงจรไฟเลี้ยง

3.1.2 วงจรควบคุม

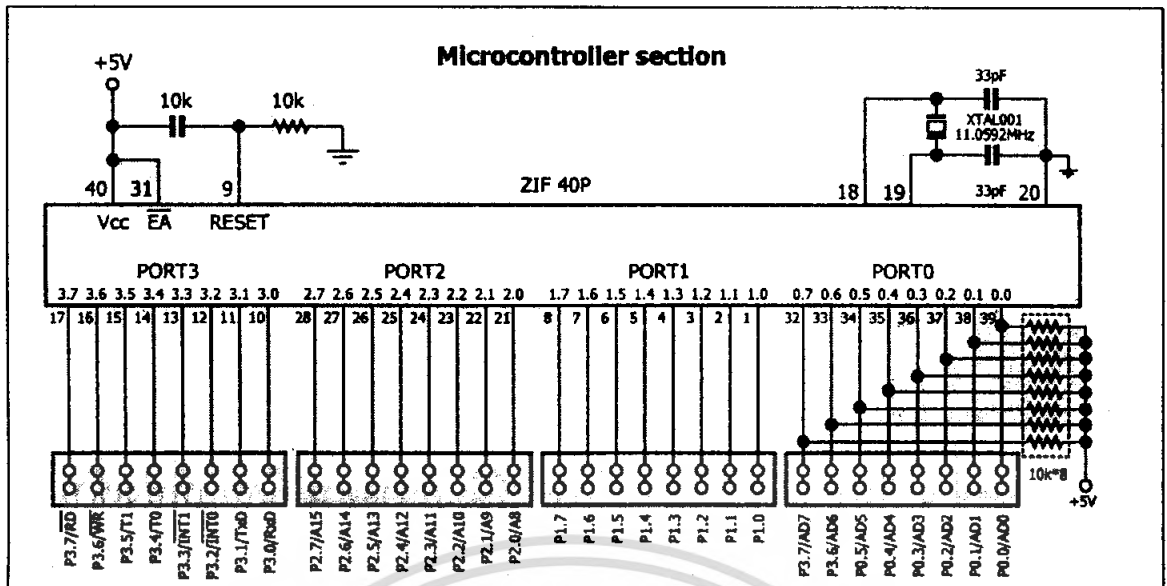
1) วงจรไมโครคอนโทรลเลอร์

ประกอบด้วย

- | | |
|--------------------|-------|
| 1. AT89C51 | 1 ตัว |
| 2. XTAL, 11.0592 | 1 ตัว |
| 3. Switch reset | 1 ตัว |
| 4. C, 33pF | 2 ตัว |
| 5. C, 10 μ F | 1 ตัว |
| 6. R, 10k Ω | 1 ตัว |

วงจรไมโครคอนโทรลเลอร์สามารถแสดงได้ดังภาพที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



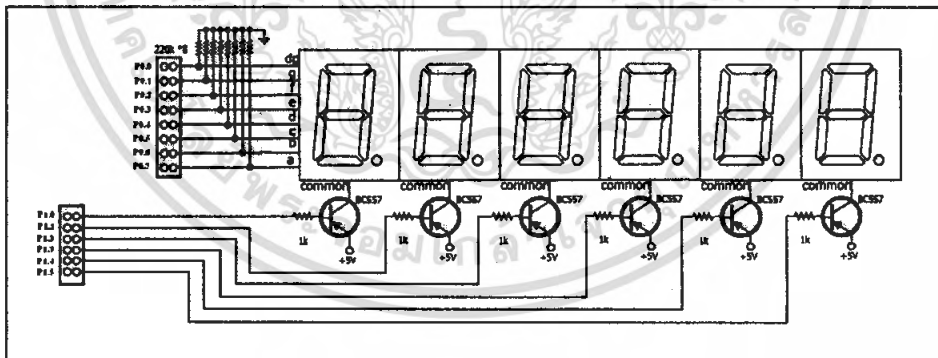
ภาพที่ 3.6 วงจรไมโครคอนโทรลเลอร์

2) วงจร LED ตัวเลข 7 ส่วน

ประกอบด้วย

- 1. 7-seg, 1" 6 ตัว
- 2. BC557 6 ตัว
- 3. R, 1kΩ 6 ตัว

วงจร LED ตัวเลข 7 ส่วนสามารถแสดงได้ดังภาพที่ 3.7



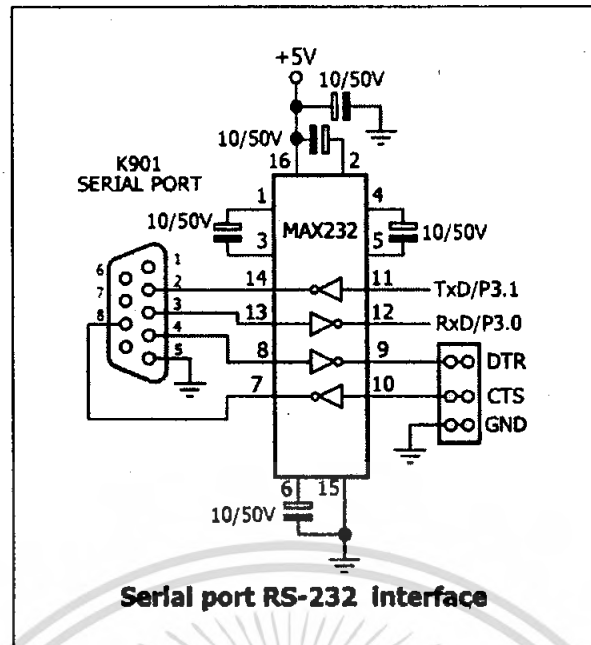
ภาพที่ 3.7 วงจร LED ตัวเลข 7 ส่วน

3) พอร์ตอนุกรมของไมโครคอนโทรลเลอร์

ประกอบด้วย

- 1. MAX232 1 ตัว
- 2. C, 10μF, 50V 5 ตัว

วงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์สามารถแสดงได้ดังภาพที่ 3.8



ภาพที่ 3.8 วงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

4) หน่วยความจำอีอีพรอมอนุกรม

หน่วยความจำอีอีพรอมอนุกรม (Serial EEPROM) เป็นหน่วยความจำแบบนอน-โวลตาไทล์ (non-volatile) หมายความว่า สามารถเก็บรักษาข้อมูลอยู่ได้โดยไม่ต้องจ่ายไฟเลี้ยง สามารถเขียน อ่าน ลบได้ด้วยสัญญาณไฟฟ้าในการเชื่อมต่อในลักษณะอนุกรมแบบระบบบัส I²C สามารถต่อพ่วงกันได้โดยการกำหนดแอดเดรสทางฮาร์ดแวร์ในการเชื่อมต่อกับไมโครคอนโทรลเลอร์ MCS-51 จะใช้ขาพอร์ตเพียง 2 ขา เช่นเดียวกับอุปกรณ์ที่ใช้ในการเชื่อมต่อแบบบัส I²C ตัวอื่นๆ

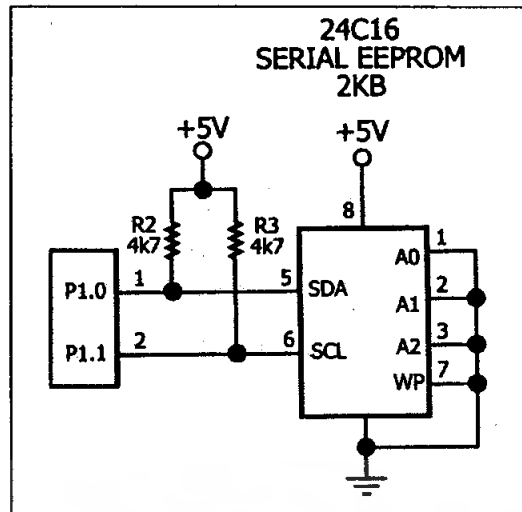
การอ่านและเขียนข้อมูลกับหน่วยความจำอีอีพรอมอนุกรม จะใช้โปรแกรม Hyper Terminal แสดงสถานะการทำงาน

โดยในโครงการนี้ได้ใช้ EEPROM เบอร์ 24LC64 ในการเก็บค่าทั้ง 2 ค่า ได้แก่ จำนวนข้าวที่มีอยู่ในตู้ และราคาข้าว

ประกอบด้วย

- | | |
|------------|-------|
| 1. 24LC16 | 1 ตัว |
| 2. R, 4k7Ω | 1 ตัว |

วงจรหน่วยความจำอีอีพรอมอนุกรมแสดงดังภาพที่ 3.9

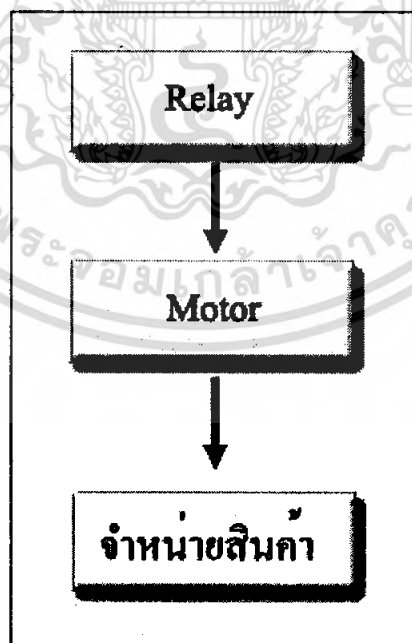


ภาพที่ 3.9 วงจรหน่วยความจำอีอีพรอมอนุกรม

5) รีเลย์

รีเลย์เป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่งที่ทำหน้าที่ในการตัดต่อวงจรหรือเป็นสวิตช์เปิดปิดวงจร โดยอาศัยกระแสไฟฟ้าที่ไหลผ่านขดลวดเกิดเป็นสนามแม่เหล็กไฟฟ้าไปเหนี่ยวนำหน้าสัมผัสของสวิตช์ทำให้สามารถตัดหรือต่อวงจรได้

ในโครงการนี้ใช้รีเลย์เบอร์ HRS4H เป็นสวิตช์ในการสั่งมอเตอร์แต่ไมโครคอนโทรลเลอร์ไม่สามารถเชื่อมต่อดีเลย์เข้ากับพอร์ตของไมโครคอนโทรลเลอร์ได้โดยตรง จึงใช้ไอซี BC557 ทำหน้าที่ให้สัญญาณเพื่อไปขับรีเลย์

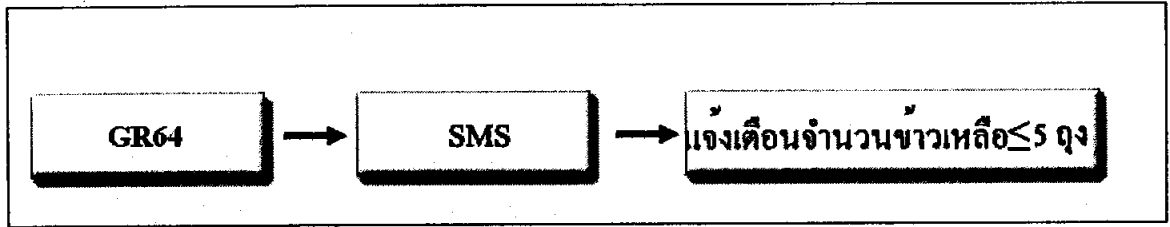


ภาพที่ 3.10 บล็อกไดอะแกรมของรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 GR64

โครงการนี้ใช้ GR64 เป็นตัวส่งข้อความแจ้งเตือนเมื่อข้าวในตู้ใกล้จะหมด สามารถแสดงบล็อกไดอะแกรมได้ดังภาพที่ 3.11



ภาพที่ 3.11 บล็อกไดอะแกรมของ GR64

3.1.4 เครื่องรับเหรียญ

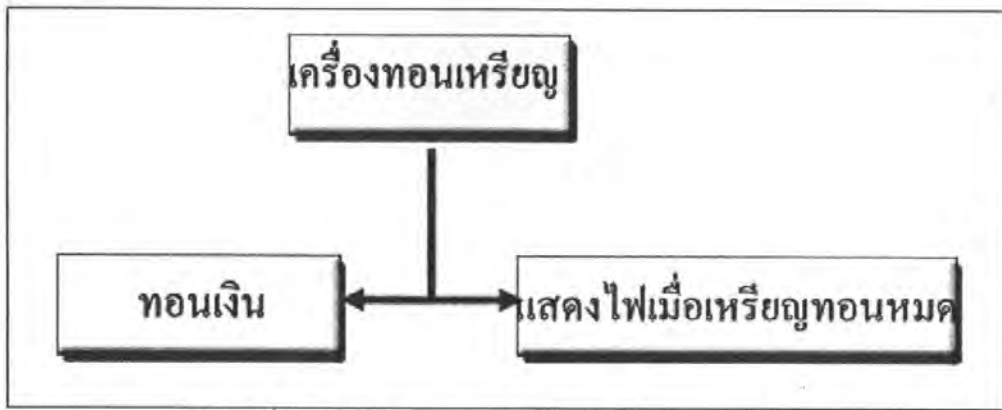
เครื่องรับเหรียญมีการทำงาน คือ เครื่องรับเหรียญจะตรวจสอบว่าเหรียญที่ได้รับเป็นเหรียญชนิดใด แล้วจะทำการส่งสัญญาณแบบพัลส์ มาตามชนิดของเหรียญ คือ เหรียญ 10 บาท จะส่งสัญญาณมา 10 พัลส์ เป็นต้น เครื่องรับเหรียญสามารถปรับเวลาในการส่งสัญญาณในแต่ละลูกคลื่นได้ 3 ค่า ได้แก่ 25 ms, 50 ms และ 80 ms ในโครงการนี้เลือกใช้ 25 ms



ภาพที่ 3.12 บล็อกไดอะแกรมของเครื่องรับเหรียญ

3.1.5 เครื่องทอนเหรียญ

เครื่องทอนเหรียญใช้รุ่น KCM-CD2000 ซึ่งสามารถทอนเหรียญได้หลายชนิด แต่ในโครงการนี้จะทดสอบแค่เหรียญ 5 บาทเพียงชนิดเดียว

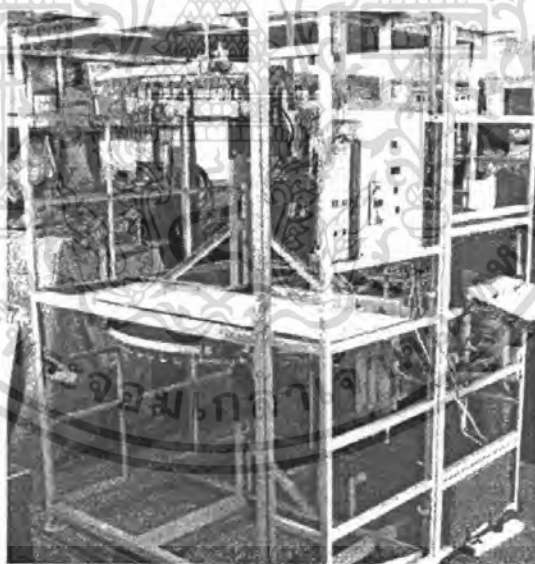


ภาพที่ 3.13 บล็อกไดอะแกรมของเครื่องทอนเหรียญ

3.2 สร้างโครงตู้

เครื่องขายข้าวอัตโนมัติถือเป็นเครื่องมือที่ช่วยอำนวยความสะดวก เพราะเราสามารถซื้อข้าวได้ตลอดเวลา ถ้าเรานำเครื่องขายข้าวอัตโนมัติไปตั้งไว้ในพื้นที่ชุมชน ผู้คนก็จะมีอีกหนึ่งทางเลือกในการซื้อข้าว โดยเครื่องขายข้าวอัตโนมัติสามารถจำหน่ายข้าวได้ 50 ถุง ถุงละ 2 กิโลกรัม

1. ขนาดของตู้ 1.22 m x 1.45 m x 1.95 m (กว้าง x ยาว X สูง)



ภาพที่ 3.14 โครงสร้างเครื่องขายข้าวอัตโนมัติ

2. ส่วนประกอบ

- 1) Motor พิกัด 100 V 1 A ร่วมกับชุดหม้อแปลงสามารถใช้กับไฟบ้านได้ (220 V) แสดงได้ดังภาพที่ 3.15
- 2) ลูกยาง ทำหน้าที่ลดการเกิด damping ของราวแขวนข้าว แสดงได้ดังภาพที่

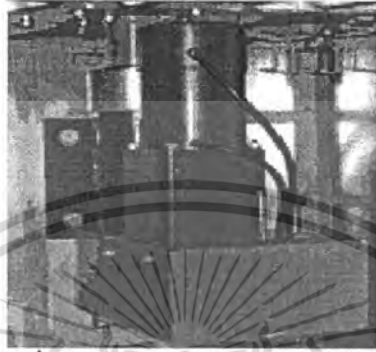
3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ราวแขวนข้าว มีเส้นผ่านศูนย์กลาง 1.125 m. ความเร็วรอบ 0.126 rad/s
บรรจุข้าวได้มากที่สุดจำนวน 25 ถุง จำนวน 2 ราว แสดงได้ดังภาพที่ 3.17

4) ชุดปลดถุงข้าว และ limit switch ทำหน้าที่ปลดถุงข้าวและนับจำนวนข้าวที่
จำหน่ายไป แสดงได้ดังภาพที่ 3.18

5) รางจำหน่ายข้าว แสดงได้ดังภาพที่ 3.19



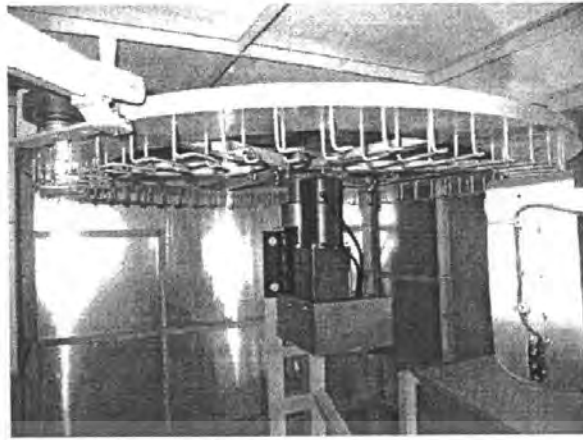
ภาพที่ 3.15 มอเตอร์และชุดหม้อแปลง



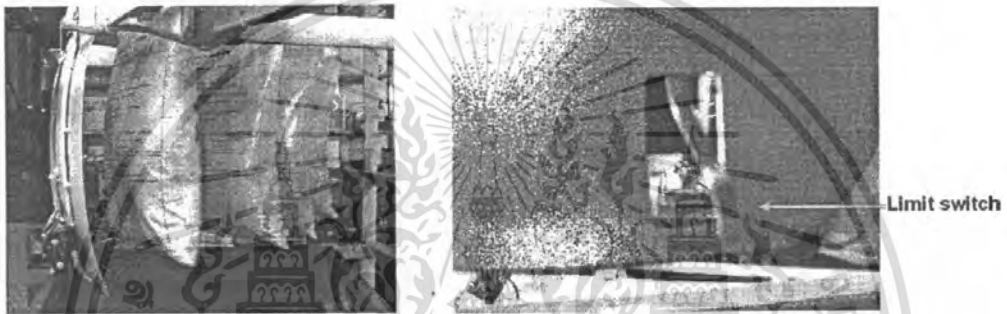
ลูกยางป้องกันการ
เกิด Damping

ภาพที่ 3.16 ลูกยางลดการเกิด damping

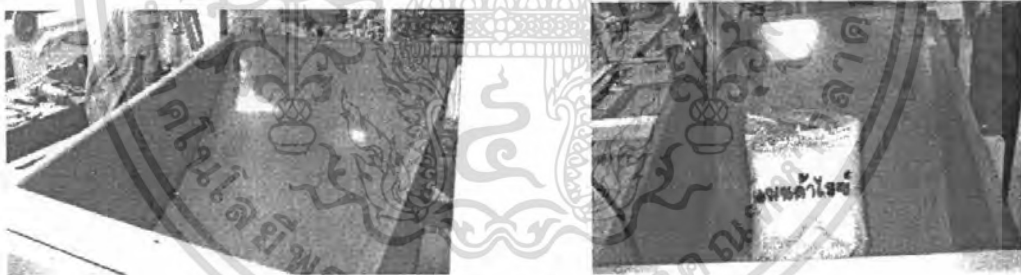
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.17 ราวแขวนข้าว



ภาพที่ 3.18 ชุดปลดถุงข้าวและ limit switch



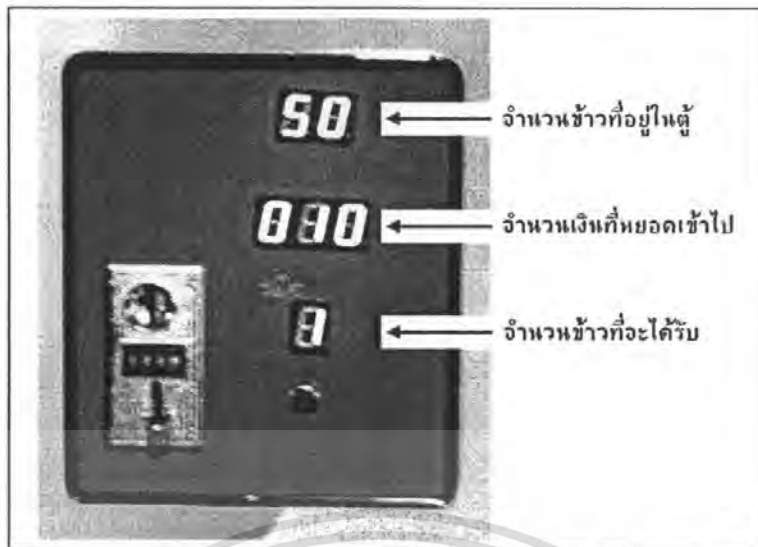
ภาพที่ 3.19 รางจำหน่ายข้าว

3.3 กระบวนการทำงานของเครื่องขายข้าวอัตโนมัติ

3.3.1 ขั้นตอนการซื้อ

- 1) หยอดเหรียญตามราคาข้าวที่กำหนดไว้ จะได้ข้าวจำนวน 1 ถุง ดังแสดงในภาพที่ 3.20
- 2) ถ้าต้องการข้าวมากกว่า 1 ถุง ให้หยอดเหรียญเพิ่มอีกตามจำนวนราคาข้าว แต่สามารถซื้อได้มากที่สุด 9 ถุง
- 3) กดปุ่มรับข้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.20 การแสดงผลขณะทำการชั่ง

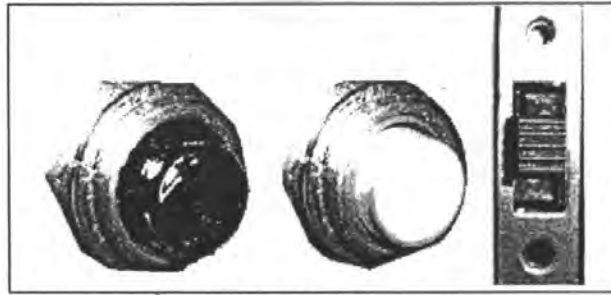
3.3.2 กระบวนการทำงานของเครื่องชั่งอัตโนมัติ

- 1) เครื่องชั่งอัตโนมัติจะแสดงค่า 3 ค่า ได้แก่ จำนวนชั่งที่อยู่ในตู้, จำนวนเงินที่หยอดเข้าไป และจำนวนชั่งที่จะได้รับ โดยเริ่มต้นจะแสดง 50, 0 และ 0 ตามลำดับ
- 2) เมื่อมีการหยอดเหรียญค่าที่ ช่องแสดงจำนวนเงินที่หยอดเข้าไป จะเพิ่มขึ้นตามจำนวนที่หยอดเข้าไป และค่าที่ ช่องแสดงจำนวนชั่งที่จะได้รับ จะเพิ่มขึ้นเมื่อหยอดเงินครบตามราคา
- 3) เมื่อกดปุ่มรับชั่งมอเตอร์ที่อยู่ในเครื่องชั่งอัตโนมัติจะทำงานทำให้ราวแขวนชั่งหมุน
- 4) เมื่อราวแขวนชั่งหมุนชั่งจะไปชนกับชุดปลดตุงชั่ง ทำให้ชั่งหล่นจากราวแขวนไปโดน limit switch
- 5) เมื่อชั่งหล่นครบตามจำนวนที่ชั่งไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์หยุดหมุน
- 6) เครื่องจะทำการลดจำนวนชั่งที่อยู่ในตู้ลงไปตามจำนวนที่ชั่ง
- 7) หากหยอดเงินเข้าไปเกินจะมีการทอนเป็นขั้นตอนสุดท้าย

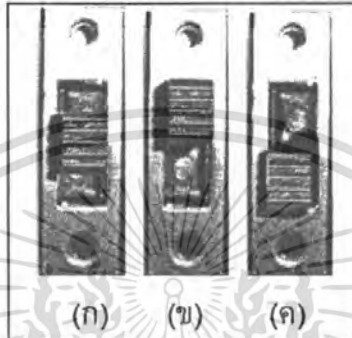
3.3.3 การปรับโหมด

ภายในกล่องควบคุมจะมีสวิทช์อยู่ 3 ตัว ได้แก่ สวิทช์เพิ่ม สวิทช์ลด และสวิทช์ฟังก์ชัน ซึ่งสวิทช์ทั้ง 3 แสดงไว้ในภาพที่ 3.21 โดยสวิทช์ฟังก์ชันสามารถเลือกได้ 3 โหมด ได้แก่ โหมดการขาย แสดงการเลือกโหมดไว้ดังภาพที่ 3.22 (ก) โหมดการบรรจุชั่ง แสดงการเลือกโหมดไว้ดังภาพที่ 3.22 (ข) และโหมดการตั้งราคา แสดงการเลือกโหมดไว้ดังภาพที่ 3.22 (ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.21 สวิตช์ภายในกล่องควบคุม



ภาพที่ 3.22 การเลือกโหมดของสวิตช์ฟังก์ชัน

1) การบรรจุข้าว

(1) เลื่อนสวิตช์ฟังก์ชันไปที่โหมดการบรรจุข้าว จะปรากฏผลที่ช่องแสดงผลดัง

ภาพที่ 3.23



ภาพที่ 3.23 การแสดงผลในโหมดการบรรจุข้าว

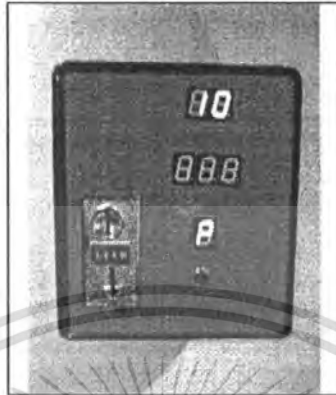
(2) กดสวิตช์เพิ่มหรือลดให้มีค่าเท่ากับจำนวนข้าวที่บรรจุใส่ตู้

(3) กดสวิตช์ยืนยันที่หน้ากล่องควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การปรับตั้งราคาข้าว

(1) เลื่อนสวิตช์ฟังก์ชันไปที่โหมดโหมดการตั้งราคา จะปรากฏผลที่ช่องแสดงผลดังภาพที่ 3.24



ภาพที่ 3.24 การแสดงผลในโหมดการตั้งราคา

- (2) กดสวิตช์เพิ่มหรือลดค่าราคาตามที่ต้องการ
 (3) กดสวิตช์ยืนยันที่หน้ากล่องควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในโครงการนี้ได้ทำการสร้างเครื่องขายข้าวอัตโนมัติ แล้วนำไปทดลองในเรื่องของการทำงานต่าง ๆ ไม่ว่าจะเป็นการทดลองเพื่อหาข้อผิดพลาดของการทำงานของเครื่องขายข้าวอัตโนมัติ และการทดลองนำเครื่องขายข้าวอัตโนมัติที่ได้ทำการปรับปรุงแก้ไขข้อผิดพลาดแล้วนำไปใช้งานจริง

4.1 การทดลองการทำงานของเครื่องขายข้าวอัตโนมัติ

การทดลองในส่วนนี้จะเป็นการทดลองฟังก์ชันการทำงานทั้งหมดของเครื่องขายข้าวอัตโนมัติ ทั้งโหมดการขายข้าว โหมดการปรับตั้งราคาข้าว และโหมดการบรรจุข้าวซึ่งมีวัตถุประสงค์ ดังนี้

1. เพื่อศึกษาคุณลักษณะการทำงานของเครื่องขายข้าวอัตโนมัติว่าเป็นไปตามคุณลักษณะที่ต้องการหรือไม่

2. เพื่อแก้ไขข้อผิดพลาดของเครื่องขายข้าวอัตโนมัติให้สามารถนำไปใช้งานจริงได้

ในการทดลองนี้ได้มีการสังเกตและบันทึกผลการทำงานต่าง ๆ ดังนี้

1. การทำงานของ EEPROM สังเกตผลได้จากการแสดงผลทาง LED ตัวเลข 7 ส่วน โดยในตอนเปิดเครื่องขายข้าวอัตโนมัติจะมีการแสดงจำนวนข้าวที่มีอยู่ในตู้ และจะมีการแสดงผลจำนวนข้าวที่เพิ่มขึ้นเมื่อหยอดเงินครบตามจำนวน

2. การรับเหรียญ สังเกตผลได้จากการแสดงผลขณะซื้อ ซึ่งจะแสดงจำนวนเงินตามที่ได้รับเข้าไป

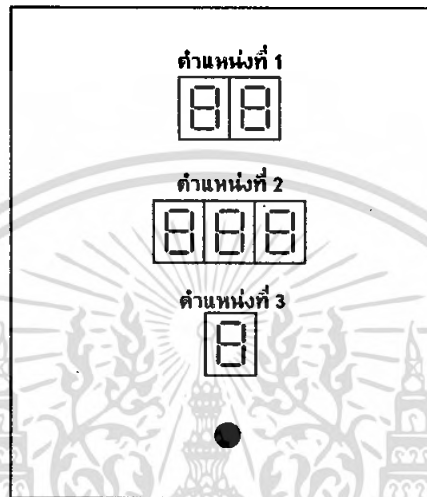
3. การทำงานของมอเตอร์และรางจำหน่ายข้าว สังเกตผลได้จากจำนวนข้าวที่ได้รับ

4. การทอนเหรียญ สังเกตได้จากจำนวนเงินที่มีการทอนออกมา

5. การส่งข้อความแจ้งเตือน สังเกตผลได้จากข้อความที่ได้รับเมื่อจำนวนข้าวเหลือน้อยกว่าหรือเท่ากับ 5 ถุง

4.1.1 การทดลองที่ 1

การออกแบบการทดลองในการทดลองนี้ คือ ตั้งราคาข้าวต่อถุงอยู่ที่ 30 บาท บรรจุก้าวไว้ 50 ถุง แล้วทำการหยอดเหรียญเป็นจำนวนเงิน 150 บาท การแสดงผลที่ตำแหน่งที่ 1, 2 และ 3 ในภาพที่ 4.1 ควรจะเป็น 50, 150 และ 5 ตามลำดับ หลังจากนั้นทำการกดปุ่ม OK เมื่อเครื่องทำงานเสร็จสิ้น การแสดงผลในตำแหน่งที่ 1, 2 และ 3 ควรมีค่าเป็น 45, 000 และ 0 ตามลำดับ ข้าวที่ได้รับควรจะเป็นจำนวน 5 ถุง ซึ่งในการทดลองนี้ได้ทำการทดลองเป็นจำนวน 10 ครั้ง



ภาพที่ 4.1 ตำแหน่งในการแสดงผล

ตารางที่ 4.1 ผลการทดลองที่ 1

ครั้งที่	การแสดงผล						จำนวนข้าวที่ได้รับ		จำนวนเหรียญที่ทอน
	ก่อนกดปุ่ม OK			หลังเครื่องทำงานสิ้นสุด			ช่อง	ช่อง	
	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	จำหน่ายบน	จำหน่ายล่าง	
1	50	150	5	45	000	0	5	0	0
2	50	150	5	45	000	0	5	0	0
3	50	150	5	45	000	0	5	0	0
4	50	150	5	45	000	0	5	0	0
5	50	150	5	45	000	0	5	0	0
6	50	150	5	45	000	0	5	0	0
7	50	150	5	45	000	0	5	0	0
8	50	150	5	45	000	0	5	0	0
9	50	150	5	45	000	0	5	0	0
10	50	150	5	45	000	0	5	0	0

หมายเหตุ จากผลการทดลอง พบว่าไม่พบความผิดพลาดใดๆ ในการทดลองนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การทดลองที่ 2

การออกแบบการทดลองในการทดลองนี้ คือ ตั้งราคาข้าวต่อถุงอยู่ที่ 30 บาท บรรจุก้าวไว้ 50 ถุง แล้วทำการหยอดเหรียญเป็นจำนวนเงิน 165 บาท การแสดงผลที่ตำแหน่งที่ 1, 2 และ 3 ในภาพที่ 4.1 ควรจะเป็น 50, 165 และ 5 ตามลำดับ หลังจากนั้นทำการกดปุ่ม OK เมื่อเครื่องทำงานเสร็จสิ้น การแสดงผลในตำแหน่งที่ 1, 2 และ 3 ควรมีค่าเป็น 45, 000 และ 0 ตามลำดับ ข้าวที่ได้รับควรจะเป็นจำนวน 5 ถุง จำนวนเหรียญที่ทอน (เหรียญ 5 บาท) ควรจะเป็นจำนวน 3 เหรียญ ซึ่งในการทดลองนี้ได้ทำการทดลองเป็นจำนวน 10 ครั้ง

ตารางที่ 4.2 ผลการทดลองที่ 2

ครั้งที่	การแสดงผล						จำนวนข้าวที่ได้รับ		จำนวนเหรียญที่ทอน
	ก่อนกดปุ่ม OK			หลังเครื่องทำงานสิ้นสุด			ชั่งน้ำหนัก	ชั่งน้ำหนัก	
	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3			
1	50	165	5	45	000	0	5	0	3
2	50	165	5	45	000	0	5	0	3
3	50	165	5	45	000	0	5	0	3
4	50	165	5	45	000	0	5	0	3
5	50	165	5	45	000	0	5	0	3
6	50	165	5	45	000	0	5	0	3
7	50	165	5	45	000	0	5	0	3
8	50	165	5	45	000	0	5	0	3
9	50	165	5	45	000	0	5	0	3
10	50	165	5	45	000	0	5	0	3

หมายเหตุ จากผลการทดลอง พบว่าไม่พบความผิดพลาดใดๆ ในการทดลองนี้

4.1.3 การทดลองที่ 3

การออกแบบการทดลองในการทดลองนี้ คือ ตั้งราคาข้าวต่อถุงอยู่ที่ 30 บาท บรรจุข้าวไว้ 27 ถุง แล้วทำการหยอดเหรียญเป็นจำนวนเงิน 150 บาท การแสดงผลที่ตำแหน่งที่ 1, 2 และ 3 ในภาพที่ 4.1 ควรจะเป็น 27, 150 และ 5 ตามลำดับ หลังจากนั้นทำการกดปุ่ม OK เมื่อเครื่องทำงานเสร็จสิ้น การแสดงผลในตำแหน่งที่ 1, 2 และ 3 ควรมีค่าเป็น 22, 000 และ 0 ตามลำดับ ข้าวที่ได้รับควรจะเป็นจำนวน 5 ถุง ซึ่งในการทดลองนี้ได้ทำการทดลองเป็นจำนวน 10 ครั้ง

ตารางที่ 4.3 ผลการทดลองที่ 3

ครั้งที่	การแสดงผล						จำนวนข้าวที่ได้รับ		จำนวนเหรียญที่ทอน
	ก่อนกดปุ่ม OK			หลังเครื่องทำงานสิ้นสุด			ชั่ง น้ำหนัก	ชั่ง น้ำหนัก	
	ตำแหน่ง ที่ 1	ตำแหน่ง ที่ 2	ตำแหน่ง ที่ 3	ตำแหน่ง ที่ 1	ตำแหน่ง ที่ 2	ตำแหน่ง ที่ 3			
1	27	150	5	22	000	0	2	3	0
2	27	150	5	22	000	0	2	3	0
3	27	150	5	22	000	0	2	3	0
4	27	150	5	22	000	0	2	3	0
5	27	150	5	22	000	0	2	3	0
6	27	150	5	22	000	0	2	3	0
7	27	150	5	22	000	0	2	3	0
8	27	150	5	22	000	0	2	3	0
9	27	150	5	22	000	0	2	3	0
10	27	150	5	22	000	0	2	3	0

หมายเหตุ จากผลการทดลอง พบว่าไม่พบความผิดพลาดใดๆ ในการทดลองนี้

4.1.4 การทดลองที่ 4

การออกแบบการทดลองในการทดลองนี้ คือ ตั้งราคาข้าวต่อถุงอยู่ที่ 30 บาท บรรจุข้าวไว้ 27 ถุง แล้วทำการหยอดเหรียญเป็นจำนวนเงิน 165 บาท การแสดงผลที่ตำแหน่งที่ 1, 2 และ 3 ในภาพที่ 4.1 ควรจะเป็น 27, 165 และ 5 ตามลำดับ หลังจากนั้นทำการกดปุ่ม OK เมื่อเครื่องทำงานเสร็จสิ้น การแสดงผลในตำแหน่งที่ 1, 2 และ 3 ควรมีค่าเป็น 22, 000 และ 0 ตามลำดับ ข้าวที่ได้รับควรจะเป็นจำนวน 5 ถุง จำนวนเหรียญที่ทอน (เหรียญ 5 บาท) ควรจะเป็นจำนวน 3 เหรียญ ซึ่งในการทดลองนี้ได้ทำการทดลองเป็นจำนวน 10 ครั้ง

ตารางที่ 4.4 ผลการทดลองที่ 4

ครั้งที่	การแสดงผล						จำนวนข้าวที่ได้รับ		จำนวนเหรียญที่ทอน
	ก่อนกดปุ่ม OK			หลังเครื่องทำงานสิ้นสุด			ช่องจำหน่ายบน	ช่องจำหน่ายล่าง	
	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3			
1	27	165	5	22	000	0	2	3	3
2	27	165	5	22	000	0	2	3	3
3	27	165	5	22	000	0	2	3	3
4	27	165	5	22	000	0	2	3	3
5	27	165	5	22	000	0	2	3	3
6	27	165	5	22	000	0	2	3	3
7	27	165	5	22	000	0	2	3	3
8	27	165	5	22	000	0	2	3	3
9	27	165	5	22	000	0	2	3	3
10	27	165	5	22	000	0	2	3	3

หมายเหตุ จากผลการทดลอง พบว่าไม่พบความผิดพลาดใดๆ ในการทดลองนี้

4.1.5 การทดลองที่ 5

การออกแบบการทดลองในการทดลองนี้ คือ ตั้งราคาข้าวต่อถุงอยู่ที่ 30 บาท บรรจุน้ำไว้ 7 ถุง แล้วทำการหยุดเหรียญเป็นจำนวนเงิน 150 บาท การแสดงผลที่ตำแหน่งที่ 1, 2 และ 3 ในภาพที่ 4.1 ควรจะเป็น 7, 150 และ 5 ตามลำดับ หลังจากนั้นทำการกดปุ่ม OK เมื่อเครื่องทำงานเสร็จสิ้น การแสดงผลในตำแหน่งที่ 1, 2 และ 3 ควรมีค่าเป็น 2, 000 และ 0 ตามลำดับ ข้าวที่ได้รับควรจะเป็นจำนวน 5 ถุง และควรมีการแจ้งเตือนผ่านทาง SMS เมื่อข้าวเหลือน้อยกว่า 5 ถุง ซึ่งในการทดลองนี้ได้ทำการทดลองเป็นจำนวน 10 ครั้ง

ตารางที่ 4.5 ผลการทดลองที่ 5

ครั้งที่	การแสดงผล						จำนวนข้าวที่ได้รับ		จำนวนเหรียญที่ทอน
	ก่อนกดปุ่ม OK			หลังเครื่องทำงานสิ้นสุด			ชั่งน้ำหนัก	ชั่งน้ำหนัก	
	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3			
1	7	150	5	2	000	0	0	5	0
2	7	150	5	2	000	0	0	5	0
3	7	150	5	2	000	0	0	5	0
4	7	150	5	2	000	0	0	5	0
5	7	150	5	2	000	0	0	5	0
6	7	150	5	2	000	0	0	5	0
7	7	150	5	2	000	0	0	5	0
8	7	150	5	2	000	0	0	5	0
9	7	150	5	2	000	0	0	5	0
10	7	150	5	2	000	0	0	5	0

หมายเหตุ จากผลการทดลอง พบว่ามีการแจ้งเตือนผ่านทาง SMS ทุกครั้ง และไม่พบความผิดปกติใดๆ ในการทดลองนี้

4.1.6 การทดลองที่ 6

การออกแบบการทดลองในการทดลองนี้ คือ ตั้งราคาข้าวต่อถุงอยู่ที่ 30 บาท บรรจุงข้าวไว้ 7 ถุง แล้วทำการหยอดเหรียญเป็นจำนวนเงิน 165 บาท การแสดงผลที่ตำแหน่งที่ 1, 2 และ 3 ในภาพที่ 4.1 ควรจะเป็น 7, 165 และ 5 ตามลำดับ หลังจากนั้นทำการกดปุ่ม OK เมื่อเครื่องทำงานเสร็จสิ้น การแสดงผลในตำแหน่งที่ 1, 2 และ 3 ควรมีค่าเป็น 2, 000 และ 0 ตามลำดับ ข้าวที่ได้รับควรจะเป็นจำนวน 5 ถุง จำนวนเหรียญที่ทอน (เหรียญ 5 บาท) ควรจะเป็นจำนวน 3 เหรียญ และควรมีการแจ้งเตือนผ่านทาง SMS เมื่อข้าวเหลือน้อยกว่า 5 ถุง ซึ่งในการทดลองนี้ได้ทำการทดลองเป็นจำนวน 10 ครั้ง

ตารางที่ 4.6 ผลการทดลองที่ 6

ครั้งที่	การแสดงผล						จำนวนข้าวที่ได้รับ		จำนวนเหรียญที่ทอน
	ก่อนกดปุ่ม OK			หลังเครื่องทำงานสิ้นสุด			ช่อง	ช่อง	
	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	ตำแหน่งที่ 1	ตำแหน่งที่ 2	ตำแหน่งที่ 3	จำหน่ายบน	จำหน่ายล่าง	
1	7	165	5	2	000	0	0	5	3
2	7	165	5	2	000	0	0	5	3
3	7	165	5	2	000	0	0	5	3
4	7	165	5	2	000	0	0	5	3
5	7	165	5	2	000	0	0	5	3
6	7	165	5	2	000	0	0	5	3
7	7	165	5	2	000	0	0	5	3
8	7	165	5	2	000	0	0	5	3
9	7	165	5	2	000	0	0	5	3
10	7	165	5	2	000	0	0	5	3

หมายเหตุ จากผลการทดลอง พบว่ามีการแจ้งเตือนผ่านทาง SMS ทุกครั้ง และไม่พบความผิดพลาดใดๆ ในการทดลองนี้

4.2 การทดลองการใช้งานจริงของเครื่องขายข้าวอัตโนมัติ

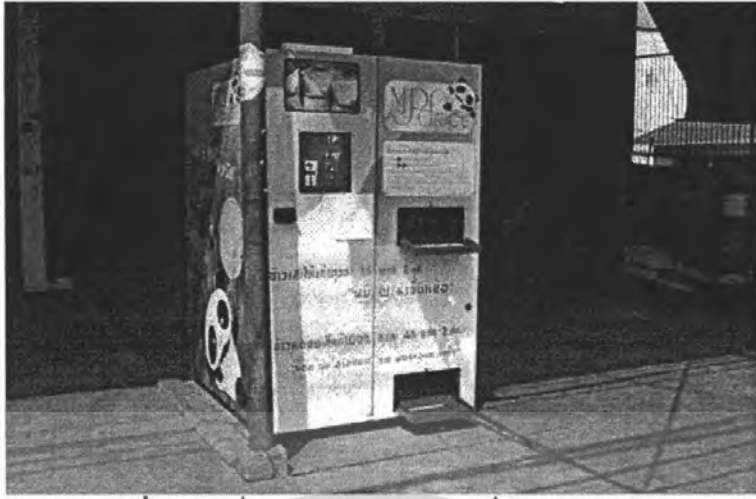
ในส่วนผลการทดลองในส่วนการใช้งานจริง คณะผู้จัดทำโครงการได้นำเครื่องขายข้าวอัตโนมัติไปติดตั้งเพื่อทดลองให้บริการและเก็บผลการทดลอง ณ บริเวณตลาดสนามแดง ซึ่งเป็นย่านโรงงานอุตสาหกรรมด้วย ตั้งแต่วันที่ 1 มีนาคม พ.ศ.2551 – 10 มีนาคม พ.ศ. 2551 เป็นเวลาทั้งหมด 10 วัน ซึ่งแสดงผลได้ดังตารางที่ 4.7

ตารางที่ 4.7 แสดงผลการใช้งานจริงของเครื่องขายข้าวอัตโนมัติ

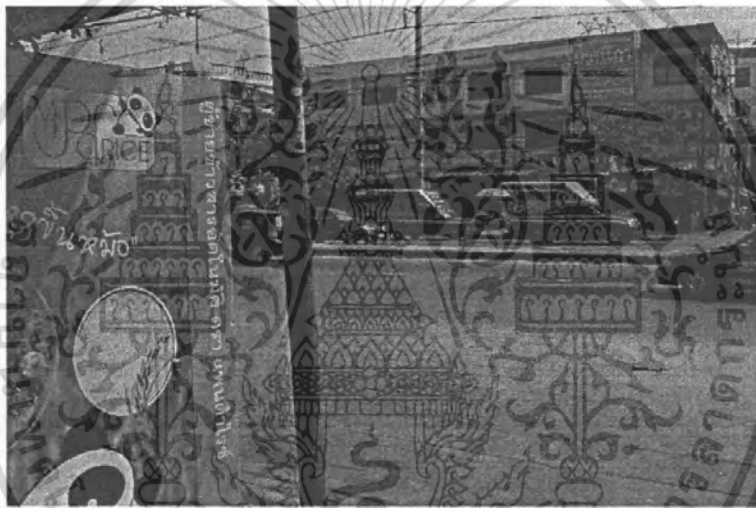
วัน \ ช่วงเวลา	06:00-12:00	12:00-18:00	18:00-00:00	00:00-06:00	รวม
1	0	2	0	0	2
2	0	2	1	0	3
3	0	2	1	0	3
4	1	1	0	0	2
5	0	3	2	1	6
6	0	0	0	0	0
7	1	1	1	1	4
8	0	4	0	1	5
9	0	2	2	0	4
10	1	2	1	0	4
รวม	3	19	8	3	33

จากตารางที่ 4.7 จะเห็นว่าช่วงเวลาที่มีการใช้บริการเครื่องขายข้าวอัตโนมัติเป็นจำนวนมาก คือช่วงเวลา 12:00-18:00 และ 18:00-00:00 ซึ่งช่วงเวลาดังกล่าวเป็นช่วงเวลาในการซื้อของเพื่อมาทำการบริโภคภายในครอบครัวและช่วงเวลาเลิกงาน เนื่องจากสถานที่นำเครื่องขายข้าวอัตโนมัติไปติดตั้งอยู่ใกล้กับบริเวณตลาดและโรงงานอุตสาหกรรม ดังนั้นกลุ่มผู้ใช้บริการจึงเป็นกลุ่มของแม่บ้านที่ออกมาเพื่อทำการจับจ่ายใช้สอยซื้อของเพื่อนำไปบริโภคภายในครอบครัว และกลุ่มของคนทำงานในโรงงานอุตสาหกรรม

หมายเหตุ มีการสนับสนุนในส่วนของขวัญที่ทำการบรรจุในเครื่องขายข้าวอัตโนมัติที่นำไปใช้ทดลองให้บริการจากโรงสีข้าวพิจิตร



ภาพที่ 4.2 เครื่องขายข้าวอัตโนมัติที่ทดลองให้บริการ



ภาพที่ 4.3 สถานที่ติดตั้งเครื่องขายข้าวอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการทำงาน

5.1.1 ส่วนควบคุม

กล่องควบคุมไมโครคอนโทรลเลอร์สามารถควบคุมอุปกรณ์ต่างๆ ให้สามารถรับค่าจากเครื่องหยอดเหรียญ จากนั้นทำการประมวลผลและส่งการให้มอเตอร์หมุนเพื่อทำการปล่อยข้าวตามจำนวนที่ผู้ซื้อต้องการ ซึ่งหากผู้ซื้อได้ทำการหยอดจำนวนเงินที่เกินจากราคาจริงที่ต้องจ่าย กล่องควบคุมจะสามารถทำการประมวลผลเพื่อทอนเหรียญออกมาแก่ผู้ซื้อได้ และหากจำนวนข้าวภายในตู้มีจำนวนน้อยกว่าที่กำหนดไว้ บอร์ดจะทำการส่งข้อความถึงเจ้าของเครื่องขายข้าวอัตโนมัติเพื่อให้ทราบและมาทำการเพิ่มจำนวนข้าวในเครื่องขายข้าวอัตโนมัติ

5.1.2 ส่วนแสดงผล

บอร์ดควบคุมไมโครคอนโทรลเลอร์สามารถควบคุมให้บอร์ดแสดงผลการหยอดเหรียญ, ผลการทอนเหรียญ, จำนวนข้าวที่ซื้อ, จำนวนข้าวที่เหลืออยู่จริง และราคาข้าวได้ผ่านทาง 7-segment



ภาพที่ 5.1 เครื่องขายข้าวอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาและข้อเสนอแนะ

จากการนำเครื่องขายข้าวอัดโนมิตีไปทดลองใช้งานที่ตลาดหนามแดง อำเภอเมือง จังหวัดสมุทรปราการ คณะผู้จัดทำได้รวบรวมข้อมูลจากผู้ให้บริการและผู้ดูแลเครื่องขายข้าวอัดโนมิตี พบว่ามีปัญหาและมีข้อเสนอแนะดังนี้

1. ในการหยอดเหรียญในบางครั้งเครื่องก็ไม่รับเหรียญ
2. ช่องจำหน่ายไม่มีความปลอดภัยเพียงพอเพราะสามารถนำมือล้วงเข้าไปหยิบข้าวได้ ควรจะออกแบบช่องจำหน่ายให้ไม่สามารถล้วงมือไปถึงราวแขวนถุงข้าวได้
3. ช่องจำหน่ายด้านสว่างอยู่ในตำแหน่งที่ต่ำเกินไป ทำให้ไม่สะดวกในการรับข้าว และอยู่ในระดับที่สัตว์เลี้ยงสามารถปัสสาวะรดช่องจำหน่ายได้ ควรจะออกแบบช่องจำหน่ายให้สูงขึ้นกว่าระดับที่สัตว์เลี้ยงสามารถปัสสาวะรดได้ และสะดวกในการรับข้าว
4. ขนาดเครื่องมีขนาดใหญ่เกินไป ใช้พื้นที่ในการติดตั้งมาก ควรจะลดขนาดตู้ในมีขนาดเล็กลง หรือสามารถบรรจุถุงข้าวได้มากขึ้น
5. ทำเลที่ตั้งมีผู้ผ่านไปผ่านมาน้อยเกินไป
6. เครื่องไม่มีความโดดเด่น ผู้ที่ผ่านไปผ่านไม่ทราบว่าเป็นเครื่องขายข้าว ควรจะออกแบบเครื่องให้มีความโดดเด่น สามารถให้ผู้พบเห็นรู้ได้ว่าเป็นเครื่องขายข้าว
7. ช่องแสดงตัวอย่างถุงข้าวไม่สวยงามเท่าที่ควร
8. ระบบการเปิดปิดตู้ยังมีความปลอดภัยไม่เพียงพอ เนื่องจากสามารถใช้ประแจหกเหลี่ยมไขเปิดตู้ได้ ควรจะออกแบบให้ล็อคเครื่องด้วยแม่กุญแจได้

บรรณานุกรม

- [1] ชัยวัฒน์ ลิ้มพรจิตรวิไล และประจัน พลังสันติกุลม, ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 กับ Keil C51 คอมไพเลอร์, Innovative Experiment Co., Ltd.
- [2] ดอนสัน ปงผาบ, ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน1, สมาคมส่งเสริม (ไทย-ญี่ปุ่น), 2549.
- [3] ดอนสัน ปงผาบ, ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน2, สมาคมส่งเสริม (ไทย-ญี่ปุ่น), 2549.
- [4] ชีรบูลย์ หล่อวิเชียรรุ่ง และคณะ, ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C, Innovative Experiment Co., Ltd.
- [5] ชีรวัฒน์ ประกอบผล, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์, สมาคมส่งเสริม (ไทย-ญี่ปุ่น), 2544.
- [6] ชีรวัฒน์ ประกอบผล, การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี, สมาคมส่งเสริม (ไทย-ญี่ปุ่น), 2545.
- [7] วรายุทธ คัมภีรวัฒน์ และสุชาดา ไชยสุข, ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS ปริณญาณิพนธ์ภาควิศวกรรมไฟฟ้า มหาวิทยาลัยขอนแก่น, 2547.
- [8] Han-Way Huang, **Using the MCS-51 microcontroller**, Oxford University Press, 2000.
- [9] John B. Peatman, **Design with microcontrollers** , McGraw-Hill, 1988.
- [10] Kenneth J. Hintz **Microcontrollers : architecture, implementation, and programming**, McGraw-Hill, 1992.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Program : AUTOMATIC VENDING RICE MACHINE *****/
#include<reg51.h>
#include<stdio.h>

sbit seg1=P1^0;      /*****1 7-segment *****/
sbit seg2=P1^1;      /*****2 7-segment *****/
sbit seg3=P1^2;      /*****3 7-segment *****/
sbit seg4=P1^3;      /*****4 7-segment *****/
sbit seg5=P1^4;      /*****5 7-segment *****/
sbit seg6=P1^5;      /*****6 7-segment *****/
sbit scl=P1^6;       /*****7 eeprom *****/
sbit sda=P1^7;       /*****8 eeprom *****/
sbit mot1=P2^0;      /*****12 EU3-Money *****/
sbit mot2=P2^1;      /*****13 Up *****/
sbit sen=P2^2;       /*****14 Down *****/
sbit coin=P2^3;      /*****15 Refill *****/
sbit torn=P2^4;      /*****16 Price *****/
sbit money=P3^2;     /*****17 OK *****/
sbit uup=P3^3;       /*****21 motor1 *****/
sbit ddown=P3^4;     /*****22 motor2 *****/
sbit rref=P3^5;      /*****23 Limit Switch *****/
sbit ppric=P3^6;     /*****24 EU3-Enable *****/
sbit ok=P3^7;        /*****25 KCM *****/

unsigned char val1;
unsigned char val2=0;
unsigned int val3=0;
unsigned char vala;
unsigned char vals;
unsigned char pric=10;
unsigned char a1=0;
unsigned char a2=0;
unsigned char b1=0;
unsigned char b2=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char b3=0;
unsigned char c1=0;
unsigned char k=0;
unsigned bdata mm=0;
unsigned int ii=0;
unsigned int jj=0;
unsigned int ij=1400;
unsigned char *j;
unsigned char t=0;
unsigned char num[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

```

```
void delay(unsigned char);
```

```
void deli2c();
```

```
char writebyte(unsigned int,unsigned char);
```

```
char readbyte(unsigned int,unsigned char *);
```

```
void sendbyte(unsigned char);
```

```
unsigned char getbyte();
```

```

/*****

```

```
/* write data 1 byte to eeprom */
```

```

*****/

```

```
char writebyte(unsigned int a,unsigned char d)
```

```
{
```

```
    unsigned char i;
```

```
    i = (a >> 7) & 0x0E;
```

```
    i |= 0xA0;
```

```
    sda = 0;
```

```
    //sda,scl = 0 start data transfer for send control byte
```

```
    scl = 0;
```

```
    sendbyte(i);
```

```
    /* control byte for write RW = 0*/
```

```
    sda = 1;
```

```
    scl = 1; deli2c();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(sda) {}
scl = 0; deli2c();

sendbyte(0x00);
sda = 1;
scl = 1; deli2c();
while(sda) {}
scl = 0; deli2c();

sendbyte(a);
sda = 1;
scl = 1; deli2c();
while(sda) {}
scl = 0; deli2c();

sendbyte(d);
sda = 1;
scl = 1; deli2c();
while(sda) {}
scl = 0; deli2c();
sda = 0;

scl = 1; //back to Idle
sda = 1;

delay(255);

return 1;
}

```



```

/*****/
/*  read data 1 byte from eeprom  */
/*****/
char readbyte(unsigned int a,unsigned char *p)
{
    unsigned char i;

    i = (a >> 7) & 0x0E;
    i |= 0xA0;

    sda = 0;
    scl = 0;

    sendbyte(i);          /* control byte for write RW = 0 */
    sda = 1;
    scl = 1; deli2c();
    while(sda) {}
    scl = 0; deli2c();

    sendbyte(0x00);
    sda = 1;
    scl = 1; deli2c();
    while(sda) {}
    scl = 0; deli2c();

    sendbyte(a);
    sda = 1;
    scl = 1; deli2c();
    while(sda) {}
    scl = 0; deli2c();

    scl = 1;
    sda = 1;
    sda = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

scl = 0;

sendbyte(i | 0x01);    /* control byte for read RW = 1*/
sda = 1;
scl = 1; deli2c();
while(sda) {}
scl = 0; deli2c();

*p = getbyte();

sda = 1;
scl = 1;
scl = 0;

scl = 1;
sda = 1;

delay(255);

return 1;
}
/*****
/* send data 1 byte to eeprom */
*****/
void sendbyte(unsigned char i)
{
char j;

for(j = 0;j < 8;j++)
{
if((i & 0x80) == 0x80) {sda = 1;}
else {sda = 0;}

scl = 1; deli2c();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    scl = 0; deli2c();
    i <<= 1;
}
return;
}
/*****
/*    get data 8 bit    */
*****/

unsigned char getbyte(void)
{
    unsigned char i;
    char j;

    for(j=0;j<8;j++)
    {
        i <<= 1;

        sda = 1;
        scl = 1; deli2c();
        if(sda == 1) {i |= 0x01;}
        else        {i &= 0xFE;}
        scl = 0; deli2c();
    }
    return i;
}
/*****
/*    Function delay    */
*****/

void delay(unsigned char i)
{
    unsigned int j;

    while(i>0)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    j = 0x02FF; while(j>0) j--;
    i--;
    }
return;
}
/*****/
/* Function delay for i2c device */
/*****/
void deli2c(void)
{
    unsigned char i;

    i = 100; while(i>0) i--;

    return;
}
/*****/
/* Function delay1 */
/*****/
void delay1(int count)
{
    int i,j;
    for(i=0;i<count;i++)
        for(j=0;j<250;j++);
}
/*****/
/* Function display */
/*****/
void display()
{
    b1=val2%10;
    b2=(val2%100)/10;
    b3=val2/100;
}

```

```

a1=val1%10;
a2=val1/10;

seg1=0;
P0=~num[val3];
delay1(5);
seg1=1;

seg2=0;
P0=~num[b1];
delay1(5);
seg2=1;

seg3=0;
P0=~num[b2];
delay1(5);
seg3=1;

seg4=0;
P0=~num[b3];
delay1(5);
seg4=1;

seg5=0;
P0=~num[a1];
delay1(5);
seg5=1;

seg6=0;
P0=~num[a2];
delay1(5);
seg6=1;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
/*   Function display1   */
/*****/

void display1()
{
    a1=val1%10;
    a2=val1/10;

    seg1=0;
    P0=~num[val3];
    delay1(10);
    seg1=1;
    seg2=0;
    seg3=0;
    seg4=0;
    P0=0xBF;
    delay1(10);
    seg2=1;
    seg3=1;
    seg4=1;

    seg5=0;
    P0=~num[a1];
    delay1(10);
    seg5=1;

    seg6=0;
    P0=~num[a2];
    delay1(10);
    seg6=1;
}

```

```

/*****/
/* Function Refill Switch */
/*****/

void rreff()
{
    if(uup==0)
    {
        delay1(30);
        if(uup==0)
        {
            if(val1<50)
                val1++;
            else
                val1=0;
        }
        delay1(250);
    }
    if(ddown==0)
    {
        delay1(30);
        if(ddown==0)
        {
            if(val1>0)
                val1--;
            else
                val1=50;
        }
        delay1(250);
    }
    if(ok==0)
    {
        delay1(30);
        if(ok==0)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        coin=0;
        writebyte(1,val1);
        delay1(300);
    }
}
}
/*****
/* Function Price Switch */
*****/
void ppricc()
{
    if(uup==0)
    {
        delay1(30);
        if(uup==0)
        {
            if(pric<99)
                pric++;
            else
                pric=0;
        }
        delay1(250);
    }
    if(ddown==0)
    {
        delay1(30);
        if(ddown==0)
        {
            if(pric>0)
                pric--;
            else
                pric=99;
        }
        delay1(250);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if(ok==0)
    {
        delay1(30);
        if(ok==0)
        {
            coin=0;
            writebyte(2,pric);
            delay1(300);
        }
    }
}

/*****
/* Function Limit Switch */
*****/
void sensor()
{
    if(sen==0)
    {
        delay1(30);
        if(sen==0)
        {
            val3--;
            val1--;
            writebyte(1,val1);
            k++;
        }
    }
}

/*****
/* Function send message */
*****/
void ssms()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char s;
char sms[]={ 'a','t','d','0','8','4','0','1','1','3','3','2','4',';',0x0D};
char sss;
for(s=0;s<15;s++)
{
    SBUF=sms[s];
    delay1(50);
}
RI=0;
sss=SBUF;
}
/*****/
/* Function change money */
/*****/
void ttorn()
{
    char tt;
    for (tt=0;tt<t;tt++)
    {
        torn=1;
        delay1(860);
    }
}
/*****/
/* Function Main */
/*****/
void main()
{
    TCON = 0x21;
    TMOD = 0x21;
    SCON = 0x50;          // Mode serial port TX
    TH1 = 0xFD;          // Set 9600 bps Timer1 default
    TL1 = 0xFD;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TF1=0;
TI = 0;
ES = 1;
TR1 = 1;
RI=0;
torn=0;
coin=0;
seg1=1;
seg2=1;
seg3=1;
seg4=1;
seg5=1;
seg6=1;
mot1=1;
mot2=1;
readbyte(1,j);
val1=*j;
vala=val1;
readbyte(2,j);
pric=*j;

while(vala>5)
{
    display();

    if(rref==0)
    {
        coin=1;
        seg1=1;
        seg2=1;
        seg3=1;
        seg4=1;
        seg5=1;
        seg6=1;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(rref==0)
{
    rreff();
    a1=val1%10;
    a2=val1/10;

    seg5=0;
    P0=~num[a1];
    delay1(10);
    seg5=1;

    seg6=0;
    P0=~num[a2];
    delay1(10);
    seg6=1;
    seg1=0;
    P0=0x8E;
    delay1(10);
    seg1=1;
}
coin=0;
}
if(ppric==0)
{
    coin=1;
    seg1=1;
    seg2=1;
    seg3=1;
    seg4=1;
    seg5=1;
    seg6=1;
    while(ppric==0)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.ppricc());
a1=pric%10;
a2=pric/10;

seg5=0;
P0=~num[a1];
delay1(10);
seg5=1;

seg6=0;
P0=~num[a2];
delay1(10);
seg6=1;

seg1=0;
P0=0x8C;
delay1(10);
seg1=1;
}
coin=0;
}
if(val3==val1)
    coin=1;
else
    coin=0;

if(money==0)
{
    delay1(10);
    if(money==0)
    {
        val2++;
        val3=val2/pric;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(ok==0)
{
    delay1(30);
    if(ok==0)
    {
        vals=val3;
        coin=1;
        if(val3>0)
        {
            if(val1>25)
            {
                ii=0;
                jj=0;
                while(vals!=k)
                {
                    mot1=0;
                    mot2=1;
                    sensor();
                    display1();
                    if(val1<=25)
                    {
                        while(vals!=k)
                        {
                            mot1=1;
                            mot2=0;
                            sensor();
                            display1();
                        }
                    }
                }
            }
        }
    }
    if(val1<=25)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

coin=1;
seg1=1;
seg2=1;
seg3=1;
seg4=1;
seg5=1;
seg6=1;
while(rref==0)
{
    rreff();
    a1=val1%10;
    a2=val1/10;

    seg5=0;
    P0=~num[a1];
    delay1(10);
    seg5=1;

    seg6=0;
    P0=~num[a2];
    delay1(10);
    seg6=1;

    seg1=0;
    P0=0x8E;
    delay1(10);
    seg1=1;
}
coin=0;
}
if(ppric==0)
{
    coin=1;
    seg1=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

seg2=1;
seg3=1;
seg4=1;
seg5=1;
seg6=1;
while(pric==0)
{
    pricc();
    a1=pric%10;
    a2=pric/10;

    seg5=0;
    P0=~num[a1];
    delay1(10);
    seg5=1;
    seg6=0;
    P0=~num[a2];
    delay1(10);
    seg6=1;

    seg1=0;
    P0=0x8C;
    delay1(10);
    seg1=1;
}
    coin=0;
}
if(val3==val1)
    coin=1;
else
    coin=0;
if(money==0)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

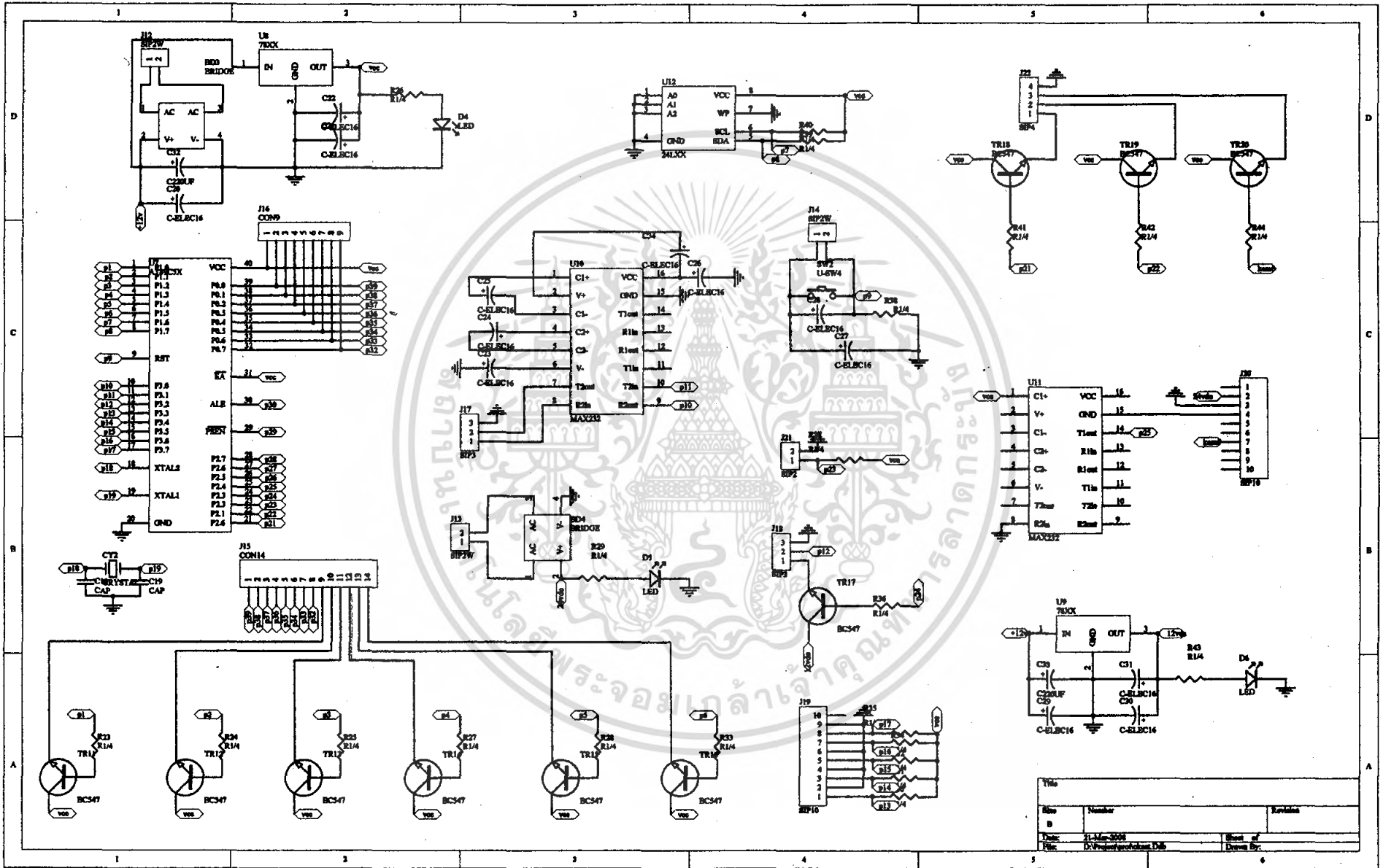
delay1(10);
if(money==0)
{
    val2++;
    val3=val2/pric;
}
}
if(ok==0)
{
    delay1(30);
    if(ok==0)
    {
        vals=val3;
        coin=1;
        if(val3>0)
        {
            if(val1>25)
            {
                ii=0;
                jj=0;
                while(vals!=k)
                {
                    mot1=0;
                    mot2=1;
                    sensor();
                    display1();
                    if(val1<=25)
                    {
                        mm=1;
                        while(vals!=k)
                        {
                            mot1=1;
                            mot2=0;
                            sensor();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ประวัติผู้เขียน



1. นาย กัมปนาท ล้อมเมตตา

เกิดวันที่ 21 มีนาคม พ.ศ. 2529

สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (ไฟฟ้ากำลัง) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2550 ที่อยู่ 13 ซอย 35 อัสวานนท์6 ตำบลปากน้ำ อำเภอเมือง จังหวัดสมุทรปราการ 10270

เบอร์โทร 087-560-5880

อีเมล kongump@msn.com



2. นาย กิตติ จิตรสำราญ

เกิดวันที่ 22 ธันวาคม พ.ศ. 2529

สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (ไฟฟ้ากำลัง) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2550 ที่อยู่ 85 หมู่ 2 ตำบลศรีพราน อำเภอแสวงหา จังหวัดอ่างทอง 14150

เบอร์โทร 080-282-4990

อีเมล ch_kitti@hotmail.com



3. นาย เกรียงศักดิ์ วรรณธนกิจ

เกิดวันที่ 4 ธันวาคม พ.ศ. 2528

สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (ไฟฟ้ากำลัง) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2550 ที่อยู่ 917 หมู่ 8 ซอยวัดด่าน ตำบลสำโรงเหนือ อำเภอเมือง จังหวัดสมุทรปราการ 10270

เบอร์โทร 086-660-5744

อีเมล galungdee@hotmail.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



4. นาย ไกรพัฒน์ เสวกวรรณ

เกิดวันที่ 2 ตุลาคม พ.ศ. 2528

สำเร็จการศึกษาระดับปริญญาตรี (ไฟฟ้ากำลัง) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2550 ที่อยู่ 420 ถ.ราชดำเนิน ตำบลในเมือง อำเภอเมือง จังหวัดกำแพงเพชร 60200

เบอร์โทร 087-359-2189

อีเมล rainbow_ton@hotmail.com



5. นาย จักรพันธ์ บรรจงรอด

เกิดวันที่ 10 พฤษภาคม พ.ศ. 2527

สำเร็จการศึกษาระดับปริญญาตรี (ไฟฟ้ากำลัง) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2550 ที่อยู่ หมู่ 3 ตำบลคูขาด อำเภอกง จังหวัดนครราชสีมา 30260

เบอร์โทร 084-0110024

อีเมล sekpower@hotmail.com



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้