สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

ระบบจอดรถในอาคาร

PARKING LOT

โดย

นาย พิศาล ภูวนกุลชัย

นาย พีระพงษ์ ทีปประพันธ์ณี

เลขหมู่.....
เลขทะเบียน 45722
วัน, เดือน, ปี 13 ก.พ. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจอดรถในอาคาร

PARKING LOT

โดย

นาย พิศาล ภูวนกุลชัย รหัส 41014310

นาย พีระพงษ์ ทีปประพันธ์ณี รหัส 41014314

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. จงกล งามวิวิทย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2544

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบจอดรถในอาคาร
PARKING LOT

ผู้จัดทำ นาย พิศาล ภูวนกุลชัย รหัสประจำตัว 41014310
นาย พีระพงษ์ ทีปประพันธ์ณี รหัสประจำตัว 41014314

.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. จงกล งามวิวิทย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจอดรถในอาคาร

โดย
นาย พิศาล ภูวนกุลชัย
นาย พีระพงษ์ ทีปประพันธ์ณี

อาจารย์ที่ปรึกษา
รองศาสตราจารย์ ดร. จงกล งามวิวิทย์

ปีการศึกษา 2544

บทคัดย่อ

ปริญญานิพนธ์นี้ได้นำเสนอแนวคิดของระบบจอดรถในอาคารที่ใช้ เครื่องอ่านบัตรแถบแม่เหล็กมาคิดค่าบริการจอดรถ และใช้ไมโครคอนโทรลเลอร์เป็นตัวกลางในการเชื่อมโยงสัญญาณของอุปกรณ์ทั้งหมด โดยเริ่มต้นเมื่อมีรถเข้ามาจอด รีโมทอินฟาเรดจะส่งสัญญาณไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมให้ที่กั้นใต้ท้องรถยกสูงขึ้นจากระดับพื้น พร้อมกับบันทึกเวลาและตรวจสอบจำนวนรถที่เข้าจอด หากจำนวนรถเท่ากับจำนวนที่จอดรถ ไฟสีแดงจะปรากฏหน้าทางเข้าของอาคารจอดรถ จากนั้นผู้ขับขี่จะต้องรับบัตรแถบแม่เหล็ก ณ จุดจอดรถ และเมื่อผู้ขับขี่ต้องการนำรถออก ผู้ขับขี่จะต้องนำบัตรแถบแม่เหล็กไปสอดยังเครื่องอ่านบัตรแถบแม่เหล็กที่ใช้คิดค่าจอดรถ เพื่อคำนวณค่าจอดรถพร้อมทั้งชำระค่าบริการจอดรถ หลังจากนำบัตรแถบแม่เหล็กสอดคืน ณ จุดจอดรถ ที่กั้นใต้ท้องรถจะเลื่อนลง จึงจะนำรถออกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARKING LOT

By

Mr. Pisan Puwanakunchai

Mr. Peerapong Teepprapunnee

Adviser

Assoc. Prof. Dr. Jongkol Ngamwiwit

Academic Year: 2001

ABSTRACT

This project aims to investigate the parking lot using the magnetic reader for computing the parking fee. The another important part is a microcontroller which is the agent in linking the signal among all devices. The process will start when a car runs pass the infrared remote that will forward the signal to the microcontroller for controlling the plate under the car to be lifted for blocking, and the time will be recorded at that moment. The driver is required to receive and keep a magnetic card provided at that parking point. When the driver wants to drive the car out of the parking lot, it is necessary that the driver inserts the magnetic card into the magnetic reader which to be used or payment in order to calculate the parking fee, then the driver pays for it. Next, the driver needs to insert the magnetic card, which has already been registered as paid, into the magnetic reader at the parking point. After that, the plate under the car will return to its place, so the driver is able to bring the car out of the parking lot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงาน	1
1.2 ขอบเขตของโครงงาน	1
1.3 ขั้นตอนและวิธีดำเนินงาน	1
1.4 แนวความคิดในการสร้างระบบ	2
บทที่ 2 บัตรแถบแม่เหล็ก และเครื่องอ่านบัตรแถบแม่เหล็ก	5
2.1 บัตรแถบแม่เหล็ก	5
2.2 เครื่องอ่านบัตรแถบแม่เหล็ก	5
2.3 การอ่านข้อมูลจากบัตรแถบแม่เหล็ก	7
บทที่ 3 สเตปปิงมอเตอร์	10
3.1 หลักการควบคุมสเตปปิงมอเตอร์	10
3.2 การควบคุมสเตปปิงมอเตอร์แบบ 2 ขั้ว	11
3.3 การควบคุมสเตปปิงมอเตอร์แบบ 2 ขั้ว ในส่วนของการขับมอเตอร์	13
การควบคุมแบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส สลับ 2 เฟส	16
3.5 การกำหนดพิกัดของกระแสสำหรับมอเตอร์	17
บทที่ 4 รีโมทอินฟาเรด 25 ฟุต (รับและส่ง) และเครื่องหยอดเหรียญ	18
4.1 รีโมทอินฟาเรด	18
4.2 เครื่องหยอดเหรียญ	20
บทที่ 5 โมดูล LCD	21
5.1 รายละเอียดเกี่ยวกับโมดูล LCD	21
5.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD	21
5.3 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16 x 2)	22
5.4 คำสั่งควบคุมโมดูล LCD	23
5.5 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD	26
บทที่ 6 ผลการทดลอง	28
6.1 ผลการทดลองที่ 1	28
6.2 ผลการทดลองที่ 2	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่ 7 สรุปและวิเคราะห์ผลการดำเนินงาน 37

กิตติกรรมประกาศ

เอกสารอ้างอิง

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

บทที่ 1

รูปที่ 1.1 แสดงโครงสร้างทางด้านฮาร์ดแวร์ของระบบจอดรถในอาคาร 2

รูปที่ 1.2 แผนผังแสดงขั้นตอนทางด้านซอฟต์แวร์ของระบบจอดรถในอาคาร 3

รูปที่ 1.3 แสดงแบบจำลองอาคารจอดรถ 4

บทที่ 2

รูปที่ 2.1 แสดงความสัมพันธ์ของสัญญาณ data, clock และ present ของเครื่องอ่านบัตรแถบแม่เหล็ก 6

บทที่ 3

รูปที่ 3.1 รูปแสดงโครงสร้างของไบโพลาร์สเตปปิงมอเตอร์ 11

รูปที่ 3.2 แสดงบล็อคไดอะแกรมของ L298N 14

รูปที่ 3.3 แสดงลักษณะของไบโพลาร์สเตปปิงมอเตอร์ 15

รูปที่ 3.4 แสดงการควบคุมแบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส สลับ 2 เฟส (ฮาล์ฟ สเตป) 16

บทที่ 4

รูปที่ 4.1 แสดงวงจรรีโมทอินฟาเรดส่วนของตัวรับ 19

รูปที่ 4.2 แสดงวงจรรีโมทอินฟาเรดส่วนของตัวส่ง 19

บทที่ 5

รูปที่ 5.1 แสดงขาต่าง ๆของจอ LCD แบบ 16 ตัวอักษร 2 บรรทัด 22

บทที่ 6

รูปที่ 6.1 แสดงการนำรถเข้าจอด ณ จุดจอดรถ 28

รูปที่ 6.2 แสดงรีโมทอินฟาเรด 28

รูปที่ 6.3 แสดงที่กั้นใต้ท้องรถที่ถูกยกขึ้นด้วยการทำงานของสเตปปิงมอเตอร์ 29

รูปที่ 6.4 แสดงบัตรแถบแม่เหล็กที่สอดอยู่ที่กล่องอ่านบัตรแถบแม่เหล็ก 29

รูปที่ 6.5 แสดงขณะทำการสอดบัตรแถบแม่เหล็กสอดในกล่องบัตรแถบแม่เหล็กตัวที่ใช้คิดค่าบริการจอดรถ 30

รูปที่ 6.6 แสดงจอ LCD ที่ปรากฏขั้นตอนการคิดค่าบริการจอดรถ 31

รูปที่ 6.7 แสดงขณะหยอดเหรียญลงในเครื่องหยอดเหรียญ 32

รูปที่ 6.8 แสดงหน้าจอ LCD 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปที่ 6.9 แสดงขณะทำการดึงบัตรแถบแม่เหล็กออกจากกล่องบัตรแถบแม่เหล็ก ตัวที่ใช้คิดค่าบริการจอดรถ 33

รูปที่ 6.10 แสดงการสอดบัตรคืน ณ จุดจอดรถ 33

รูปที่ 6.11 แสดงที่กั้นใต้ท้องรถเลื่อนลงมา ณ ระดับพื้นดังเดิม 34

รูปที่ 6.12 แสดงการนำรถออกจากที่จอดรถ 34

รูปที่ 6.13 แสดงไฟแดงหน้าทางเข้าอาคารจอดรถที่ดับอยู่ 35

รูปที่ 6.14 แสดงไฟแดงหน้าทางเข้าอาคารจอดรถสว่างขึ้น 35

รูปที่ 6.15 แสดงไฟแดงหน้าทางเข้าอาคารจอดรถดับลง 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

บทที่ 2

ตารางที่ 2.1 แสดงรายละเอียดของบัตรแถบแม่เหล็ก 5

ตารางที่ 2.2 แสดงข้อมูลของรหัส BCD สำหรับเครื่องอ่านบัตรแถบแม่เหล็ก 9

บทที่ 3

ตารางที่ 3.1 แสดงการควบคุมแบบจ่ายไฟให้ขดลวด
ครั้งละ 1 เฟส 12

ตารางที่ 3.2 แสดงการควบคุมแบบจ่ายไฟให้ขดลวด
พร้อมกัน 2 เฟส 12

ตารางที่ 3.3 แสดงการควบคุมแบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส
สลับ 2 เฟส 13

ตารางที่ 3.4 แสดงการควบคุมแบบไบโพลาร์สเตปปิงมอเตอร์ 14

ตารางที่ 3.5 แสดงการจ่ายกระแสให้มอเตอร์และการสร้างสัญญาณ
ควบคุม L298N 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ในช่วงระยะเวลาที่ผ่านมาการวิวัฒนาการทางด้านเทคโนโลยีได้พัฒนาไปอย่างรวดเร็ว จนได้เข้าไปมีอิทธิพลต่อความเป็นอยู่ของมนุษย์ เทคโนโลยีทำให้เกิดการทำงานที่รวดเร็วและแม่นยำยิ่งขึ้น การดำเนินชีวิตในปัจจุบันจึงต้องมีสิ่งอำนวยความสะดวกที่เกิดจากเทคโนโลยีแทบทั้งสิ้นซึ่งรวมทั้งการใช้บัตรต่างๆ เพื่อเป็นเครื่องอำนวยความสะดวกด้วย เช่น บัตรเอทีเอ็ม บัตรเครดิต บัตรเช็คเวลาเข้าออกของพนักงาน บัตรสมาชิกต่างๆ และอื่นๆ อีกมากมาย

ดังนั้น โครงงานนี้จึงเห็นประโยชน์ในการใช้งานของบัตรแถบแม่เหล็ก ในการประยุกต์ใช้กับที่จอดรถเพื่อทำการป้องกันการสูญหายของรถยนต์จากการโจรกรรม อีกทั้งยังมีความสะดวกในการคำนวณค่าบริการที่จอดรถ จึงได้เกิดแนวความคิดในการทำโครงงานระบบจอดรถในอาคาร

1.2 ขอบเขตของโครงงาน

1.2.1 ส่วนของบัตร ใช้บัตรแถบแม่เหล็ก

1.2.2 ส่วนของการอ่านข้อมูล ใช้เครื่องอ่านบัตรแถบแม่เหล็ก

1.2.3 ส่วนควบคุม ใช้ไมโครคอนโทรลเลอร์

1.2.4 ส่วนของการตรวจจับรถยนต์ ใช้รีโมทอินฟาเรด

1.2.5 ส่วนของเครื่องกั้นใต้ท้องรถ ใช้สเตปปิงมอเตอร์

1.2.6 ส่วนที่แสดงค่าบริการจอดรถ ใช้จอ LCD

1.2.7 ส่วนแสดงผลการทำงาน ใช้แบบจำลองอาคารจอดรถ จำลองการทำงานของโครงงานนี้

1.3 ขั้นตอนและวิธีดำเนินงาน

1.3.1 ศึกษาค้นคว้าข้อมูลในส่วนที่เกี่ยวข้องกับโครงงาน

1.3.2 จัดเตรียมอุปกรณ์ที่เกี่ยวข้อง

1.3.3 ศึกษารายละเอียดของบัตรแถบแม่เหล็ก

1.3.4 ศึกษาการทำงานของเครื่องอ่านบัตรแถบแม่เหล็ก

1.3.5 ศึกษาการทำงานของสเตปปิงมอเตอร์

1.3.6 ศึกษาการทำงานของรีโมทอินฟาเรด

1.3.7 ศึกษาการทำงานของจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

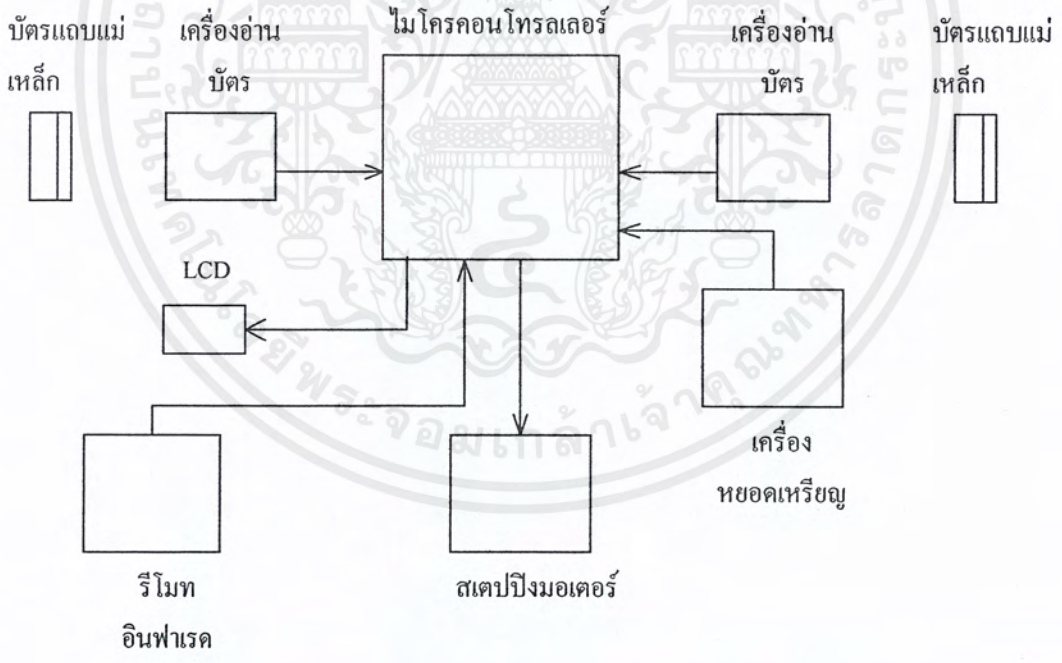
1.3.8 ศึกษาการทำงานของไมโครคอนโทรลเลอร์

1.3.9 ออกแบบวงจรและติดตั้งอุปกรณ์

1.3.10 เขียนโปรแกรมควบคุมการทำงานทั้งหมด

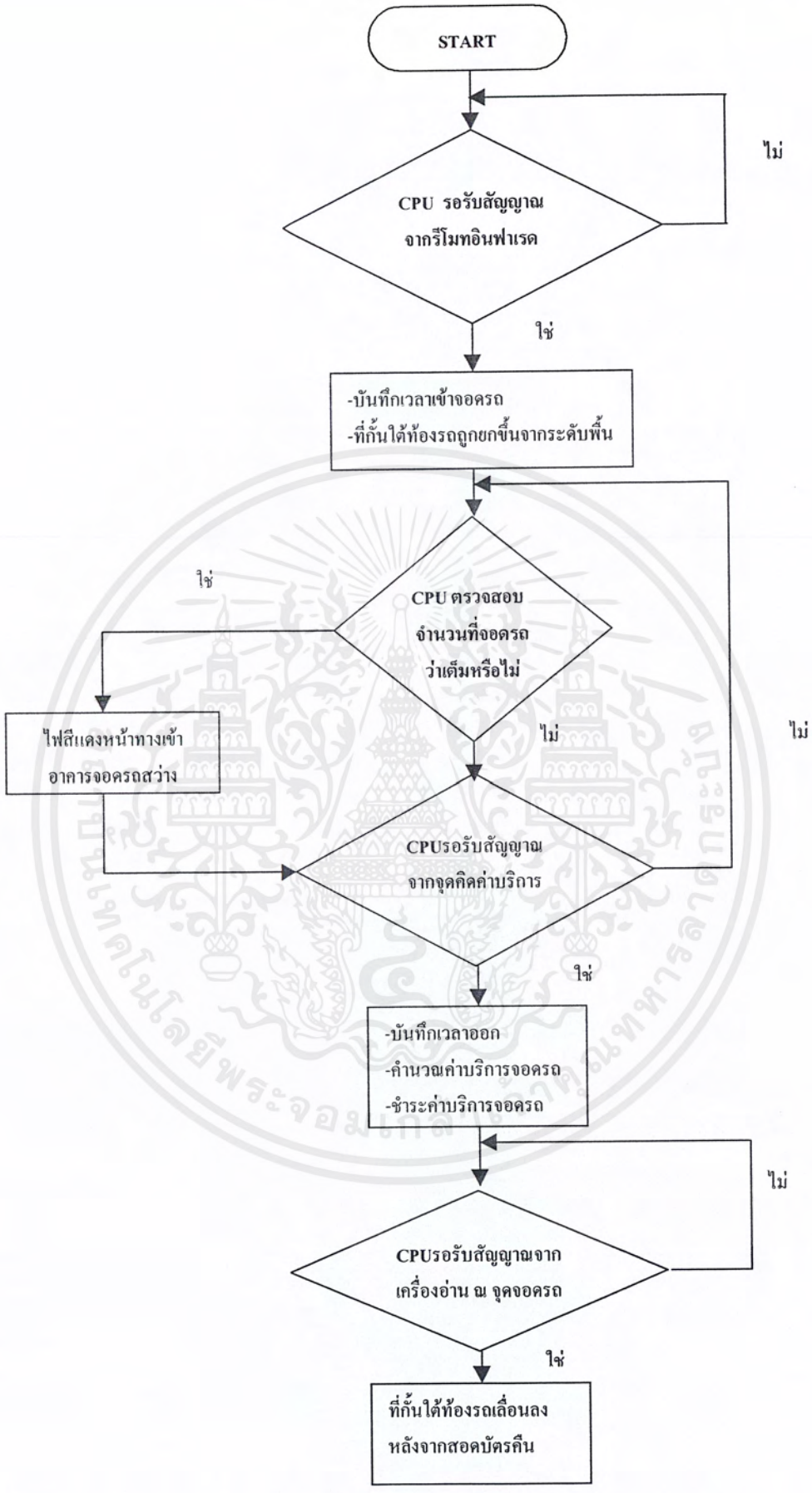
1.4 แนวความคิดในการสร้างระบบ

เครื่องคิดค่าบริการที่จอดรถโดยใช้บัตรแถบแม่เหล็ก จะมีโครงสร้างทางด้านฮาร์ดแวร์ (hardware) ของระบบจอดรถในอาคารดังรูป 1.1 ซึ่งประกอบไปด้วยบัตรแถบแม่เหล็ก เครื่องอ่านบัตรแถบแม่เหล็ก ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 ซึ่งใช้ควบคุมและประมวลผล จอ LCD ใช้สำหรับแสดงข้อความ เครื่องหยอดเหรียญสำหรับการคิดค่าบริการจอดรถ สเตปปิงมอเตอร์สำหรับยกที่กั้นใต้ท้องรถให้สูงขึ้นจากระดับพื้น และยังมีรีโมทอินฟาเรดเพื่อใช้ควบคุมสเตปปิงมอเตอร์ด้วย



รูปที่ 1.1 แสดงโครงสร้างทางด้านฮาร์ดแวร์ของระบบจอดรถในอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

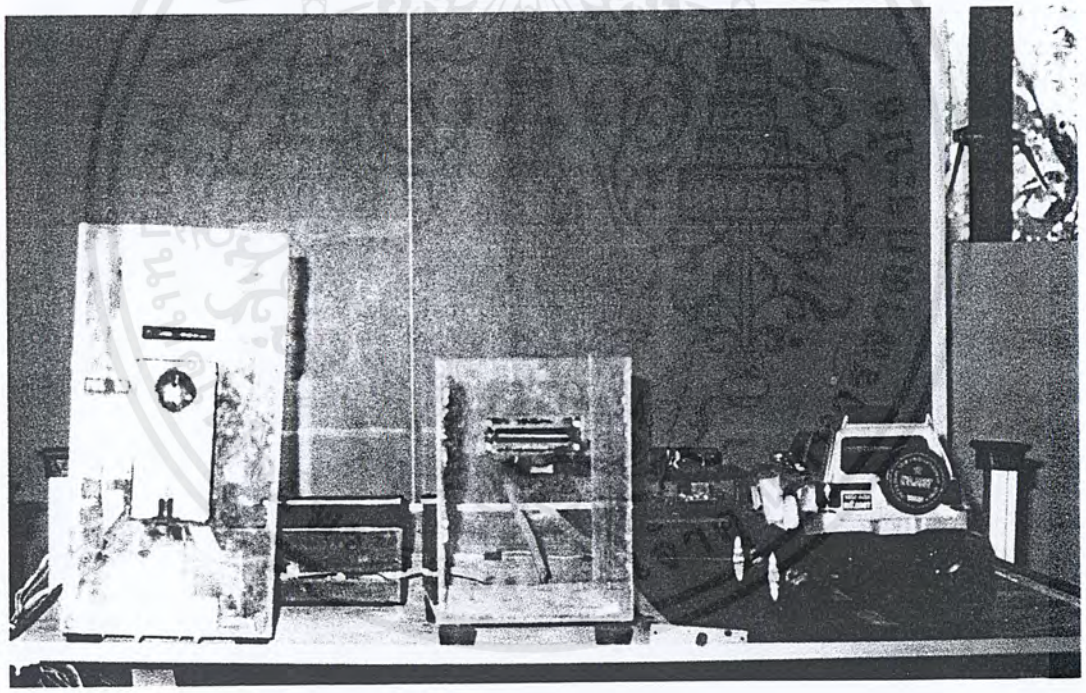


รูปที่ 1.2 แผนผังแสดงขั้นตอนทางด้านซอฟต์แวร์(software)ของระบบจอดรถในอาคาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1.2 เริ่มต้นเมื่อมีรถเข้ามาจอด รีโมทอินฟาเรดจะส่งสัญญาณไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมให้ที่กั้นใต้ท้องรถยกสูงขึ้นจากระดับพื้น พร้อมกับบันทึกเวลาและตรวจสอบจำนวนรถที่เข้าจอด หากจำนวนรถเท่ากับจำนวนที่จอดรถ ไฟสีแดงจะปรากฏหน้าทางเข้าของอาคารจอดรถ ซึ่งหมายถึงที่จอดรถในอาคารจอดรถเต็ม ผู้ขับขี่จะต้องรับบัตรแถบแม่เหล็ก ณ จุดจอดรถ หลังจากนั้นเมื่อผู้ขับขี่ต้องการนำรถออก ผู้ขับขี่จะต้องนำบัตรแถบแม่เหล็กไปสอดยังเครื่องอ่านบัตรแถบแม่เหล็กที่ใช้คิดค่าจอดรถ เพื่อคำนวณค่าจอดรถพร้อมทั้งชำระค่าบริการจอด จากนั้นเมื่อนำบัตรแถบแม่เหล็กสอดคืน ณ จุดจอดรถที่กั้นใต้ท้องรถจะเลื่อนลง จึงจะนำรถออกได้ รายละเอียดต่าง ๆ ของซอฟต์แวร์ได้แสดงที่ภาคผนวก ก

ในส่วนของแบบจำลองอาคารจอดรถที่ใช้ในโครงงานนี้จะมีลักษณะดังรูป



รูปที่ 1.3 แสดงแบบจำลองอาคารจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
บัตรแถบแม่เหล็กและเครื่องอ่านบัตรแถบแม่เหล็ก

2.1 บัตรแถบแม่เหล็ก

บัตรแถบแม่เหล็ก ที่เราคุ้นเคยกันดีในรูปแบบของ บัตรเงินสด บัตรเอทีเอ็ม บัตรเครดิต บัตรวีซ่า และบัตรอื่นๆ อีกมากมาย ฯลฯ เป็นการ์ดแถบแม่เหล็ก ซึ่งบนตัวบัตรเองจะบันทึกข้อมูลและรายละเอียดต่างๆ ของบัตรไว้ในรูปของเส้นแรงแม่เหล็ก ภายในส่วนที่เป็นแถบแม่เหล็กในการ์ด โดยที่แถบแม่เหล็กที่อยู่บนการ์ด ซึ่งเรียกว่า "Track" นั้น ตามปกติแล้วจะถูกแบ่งออกเป็น 3 ส่วนด้วยกัน ซึ่งในแต่ละส่วนจะใช้เก็บข้อมูลซึ่งมีความหนาแน่นและลักษณะของข้อมูลที่มีความแตกต่างกัน ดังนี้คือ

แถบแม่เหล็กที่	ความหนาแน่นในการบันทึก	การเข้ารหัสข้อมูล	จำนวนตัวอักษรสูงสุด	ลักษณะของข้อมูลที่เก็บในบัตร
Track 1	210 BPI	ALPHA	79	ชื่อเจ้าของบัตรและหมายเลขบัตร
Track 2	75 BPI	BCD	40	หมายเลขบัตรและวันหมดอายุ
Track 3	210 BPI	BCD	107	หมายเลขบัตรและรหัสพิเศษ

ตารางที่ 2.1 ตารางแสดงรายละเอียดของบัตรแถบแม่เหล็ก

หมายเหตุ ความหนาแน่นของการบันทึกข้อมูลมีหน่วยเป็น BPI (Byte Per Inch)

2.2 เครื่องอ่านบัตรแถบแม่เหล็ก

เครื่องอ่านแถบแม่เหล็ก ของ Panasonic รุ่น ZU-M2121S451 นี้จะอ่านได้เฉพาะข้อมูล ที่บันทึกด้วยความละเอียด 75 BPI คือ เฉพาะในส่วนของแถบแม่เหล็กแถบที่ 2 เท่านั้น โดยมีสัญญาณที่ใช้เชื่อมต่อเพื่ออ่านข้อมูลจากเครื่องอ่านแถบแม่เหล็กทั้งหมด 5 สัญญาณ ซึ่งเป็นสัญญาณเอาต์พุทส่งออกมาจากเครื่องอ่านแถบแม่เหล็กทั้งหมด และมีขาไฟเลี้ยงวงจรที่ต้องต่อให้เครื่องอ่านอีก ดังนี้คือ

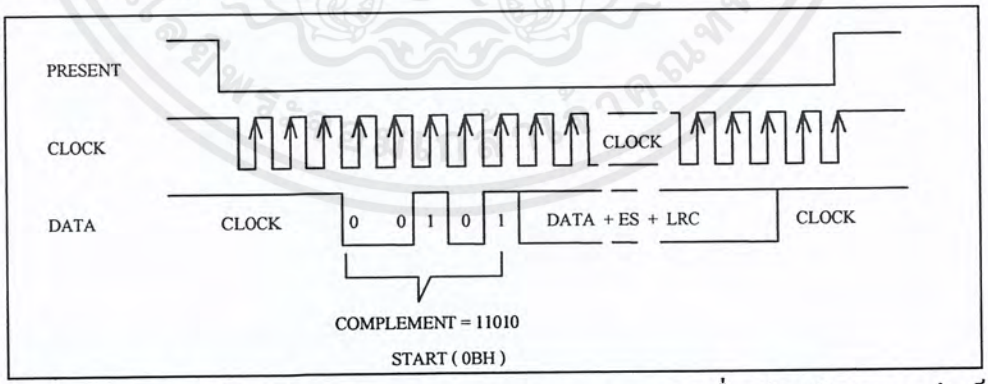
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ไฟเลี้ยงวงจร (+VCC) ต้องมีค่าเป็น +5 V DC ซึ่งต่อจากภายนอกให้เครื่องอ่านแถบแม่เหล็ก

2.2.2 สัญญาณข้อมูล เป็นสัญญาณข้อมูลเอาต์พุทส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก ซึ่งเป็นสัญญาณของข้อมูลที่อ่านได้จากบัตรแถบแม่เหล็ก การอ่านสัญญาณข้อมูลนี้ต้องอ่านแบบอนุกรมทีละบิตโดยต้องพิจารณาให้สัมพันธ์สอดคล้องกับสัญญาณนาฬิกาของบัตรด้วย สัญญาณข้อมูลที่อ่านได้นี้จะกลับสภาวะกับสัญญาณจริง คือ มีสภาวะเป็นตรงข้ามกันอยู่ ดังนั้นเมื่ออ่านสัญญาณจากเครื่องอ่านแถบแม่เหล็กได้แล้วจะต้องกลับสภาวะของสัญญาณเสียก่อนจึงจะได้ข้อมูลที่ถูกต้อง

2.2.3 สัญญาณนาฬิกา เป็นสัญญาณนาฬิกาเอาต์พุทส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก ใช้เป็นสัญญาณอ้างอิงในการอ่านสัญญาณข้อมูลจากบัตร โดยการอ่านสัญญาณข้อมูลแต่ละบิตนั้นต้องอ่านในขณะที่สัญญาณนาฬิกานี้เป็นขอบขาขึ้น (rising edge) เสมอ สัญญาณข้อมูลที่อ่านได้นั้นจะเริ่มต้นจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อนเป็นอันดับแรก

2.2.4 สัญญาณ Present เป็นสัญญาณสถานะเอาต์พุทส่งออกมาจากเครื่องอ่านแถบแม่เหล็ก ทำงานที่ลอจิกศูนย์เมื่อสัญญาณนี้เกิด (“0”) จะบ่งบอกให้ทราบว่าเครื่องอ่านแถบแม่เหล็กเริ่มต้นส่งข้อมูลออกมา หรือมีการนำบัตรแถบแม่เหล็กไปรูดผ่านหัวอ่านสัญญาณของเครื่อง เมื่อสัญญาณนี้หมดลง (กลับเป็นลอจิกหนึ่ง) อีกครั้งหนึ่ง แสดงว่าข้อมูลจากการอ่านในครั้งนั้นถูกส่งออกไปหมดสิ้นเรียบร้อยแล้ว ซึ่งเราอาจใช้ประโยชน์จากสัญญาณนี้ ส่งเป็นสัญญาณอินเทอร์รัพท์ให้ไมโครคอนโทรลเลอร์เพื่อทำการอ่านข้อมูล



รูปที่ 2.1 แสดงความสัมพันธ์ของสัญญาณ data, clockและ present ของเครื่องอ่านบัตรแถบแม่เหล็ก

2.2.5 สัญญาณ Card In Detect เป็นสัญญาณจากอุปกรณ์ตรวจจับ ปกติมีค่าเป็น “1” แต่เมื่อมีการสอดบัตรเข้าไปยังช่องรับบัตร สัญญาณนี้จะมีค่าเป็น “0” ค้างไว้ตลอด และจะกลับมีค่าเป็น “1” อีกครั้งเมื่อดึงบัตรออกจากช่องรับบัตรเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 สัญญาณ End Stop เป็นสัญญาณจากอุปกรณ์ตรวจจับที่อยู่ภายในช่องรับบัตรด้านในสุด โดยปกติมีค่าเป็น "1" แต่เมื่อมีการสอดบัตรเข้าไปยังช่องรับบัตรจนสุด ก็จะทำให้สัญญาณนี้มีค่าเป็น "0" ค้างไว้ตลอด และจะกลับมีค่าเป็น "1" อีกครั้งก็ต่อเมื่อดึงบัตรกลับออกมา

2.2.7 Ground เป็นจุดอ้างอิงของสัญญาณต่างๆ ระหว่างเครื่องอ่านแถบแม่เหล็กกับบอร์ดไมโครคอนโทรลเลอร์

2.3 การอ่านข้อมูลจากบัตรแถบแม่เหล็ก

การใช้เครื่องอ่านแถบแม่เหล็กเพื่อให้อ่านข้อมูลจากบัตรแถบแม่เหล็กนั้น ต้องสอดบัตรเข้าไปในช่องรับบัตรจากด้านหน้า ตามทิศทางและตำแหน่งที่กำหนดไว้เท่านั้น โดยให้สังเกตจากตำแหน่งของหัวอ่านกับส่วนที่เป็นแถบแม่เหล็กของบัตร ต้องอยู่ในแนวเดียวกัน สำหรับการอ่านข้อมูลจากบัตรนั้น ขอแนะนำให้อ่านในขณะที่สอดบัตรเข้าไปในช่องรับบัตรเท่านั้น เพราะถ้าไปอ่านในขณะที่ดึงบัตรกลับออกมาจะทำให้ได้ข้อมูลที่ย้อนถอยหลังกลับ ซึ่งจะทำให้ยากแก่การถอดรหัสที่ได้จากการอ่าน และเนื่องจาก เครื่องอ่านบัตรแถบแม่เหล็กสามารถอ่านข้อมูล จากบัตรแถบแม่เหล็กได้เฉพาะข้อมูลของแถบแม่เหล็กที่ 2 เท่านั้น ดังนั้นจะกล่าวถึงเฉพาะวิธีการอ่านข้อมูลจากแถบแม่เหล็กที่ 2 เพียงอย่างเดียวโดยรูปแบบของข้อมูลในแถบแม่เหล็กส่วนที่ 2 ที่ใช้ในการบันทึกสัญญาณ ด้วยความหนาแน่นของสนามแม่เหล็ก 75 BPI ซึ่งจะบรรจุจำนวนรหัสของข้อมูลต่างๆ ในแถบแม่เหล็กนี้ได้สูงสุดไม่เกิน 40 ตัวอักษร โดยนับรวมรหัสควบคุมและรหัสตรวจสอบด้วยจะมีรายละเอียดดังนี้

SS	PAN	FS	Addition Data	ES	LRC

SS = รหัสข้อมูลแสดงการเริ่มต้นของข้อมูลในบัตร (start sentinel) มีรหัสข้อมูลเป็น 0BH (;)

Additional Data = เป็นข้อมูลเพิ่มเติมอื่นๆ ของบัตร เช่น เดือน/ปี ที่ออกบัตรและหมดอายุ

ES = รหัสข้อมูลแสดงการสิ้นสุดของข้อมูลในบัตร (end sentinel) มีรหัสข้อมูลเป็น 0FH (?)

LRC = ข้อมูลตรวจสอบความผิดพลาด(longitudinal redundancy check)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0000000000	SS	Data,Data,Data,...,Data	ES	LRC	0000000000

จากภาพแสดงลักษณะการจัดเรียงของข้อมูลที่เก็บไว้ในบัตรแถบแม่เหล็กในส่วนของแถบแม่เหล็กส่วนที่ 2 ซึ่งจัดเรียงลำดับความสำคัญจากซ้ายไปขวาแบบอนุกรมทีละบิตโดยจะสังเกตเห็นว่ามีข้อมูลซึ่งเป็นลอจิกศูนย์หรือเรียกว่า "Clocking Bit" นำหน้าและปิดท้ายข้อมูลจริงอยู่ ลักษณะการจัดเก็บข้อมูลของแถบแม่เหล็กส่วนที่ 2 จะเก็บข้อมูลด้วยการเข้ารหัสแบบ "Modulo5" ซึ่งในแต่ละชุดข้อมูลจะประกอบด้วยข้อมูล 5 บิต โดย 4 บิตแรก (D0..D3) เป็นรหัสข้อมูลแบบ BCD ส่วนบิตที่ 5 (D4) เป็นพาริตี้บิตแบบคี่ (ODD) เพื่อใช้ตรวจสอบความถูกต้องของข้อมูล ในแต่ละไบท์ (5 Bit) ที่อ่านได้ หากค่าของบิตพาริตี้ผิดพลาด แสดงว่าการอ่านข้อมูลนั้นล้มเหลว โดยเริ่มทำการอ่านข้อมูลเมื่อสัญญาณ Present เริ่มเป็นศูนย์ก่อน และทำการอ่านข้อมูลในทุกๆ ขอบขาขึ้นของสัญญาณนาฬิกาเสมอ ซึ่งข้อมูลในส่วนก่อนเริ่มต้นและหลังจากสิ้นสุดของการอ่านนี้จะมีค่าเป็นศูนย์ (สัญญาณ ข้อมูล = ลอจิก "1") นำหน้าและปิดท้ายด้วยข้อมูลจริงอยู่ ข้อมูลในส่วนนี้เราไม่ต้องสนใจ แต่ให้ตรวจสอบและรอจนกว่าจะเริ่ม เป็นข้อมูล Start บิต (0BH หรือ 11010 ในที่นี้เรียงลำดับความสำคัญจากซ้ายไปขวา ซึ่งบิตเริ่มต้นของรหัส Start หรือ 0BH ต้องเริ่มด้วย 1 เป็นบิตแรกเสมอ) ดังนั้นในการอ่านเราต้อง รอจนกว่าจะพบสัญญาณข้อมูลมีค่าเป็น "1" (สัญญาณข้อมูล = ลอจิก "0" เพราะกลับสภาวะกันอยู่) จึงเริ่มเก็บข้อมูลชุดละ 5 บิตไปเรื่อยๆ จนถึงรหัสจบ (0FH) เมื่อพบรหัสจบแล้วจะมีข้อมูลตาม มาอีก 1 ไบท์ ซึ่งเป็นข้อมูลสำหรับตรวจสอบความผิดพลาดของการอ่านข้อมูลในแต่ละไบท์ (ไม่คิดพาริตี้บิต) ตั้งแต่เริ่มต้น จนสิ้นสุดมาทำการ XOR กัน โดยครั้งแรกให้นำไบท์ เริ่มต้น (0BH) ทำการ XOR กับศูนย์ แล้วนำผลลัพท์ที่ได้ไป XOR กับไบท์ถัดไปเรื่อยๆ ตามลำดับ หากผลลัพท์สุดท้ายที่ได้ไม่เท่ากับ ค่าของ "LRC" ที่อ่านมาได้แสดงว่าการอ่านข้อมูลทั้งหมดล้มเหลว

สำหรับข้อมูลของรหัส BCD ใช้สำหรับเครื่องอ่านแถบแม่เหล็กนี้จะเป็นข้อมูลชุดละ 5 บิต โดยเป็นข้อมูลจริง 4 บิต และเป็นรหัสตรวจสอบพาริตี้อีก 1 บิต ซึ่งมีจำนวนทั้งหมด 16 อักขระ ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parity	D3	D2	D1	D0	Character	Function
1	0	0	0	0	0 (0H)	Data
0	0	0	0	1	1 (1H)	Data
0	0	0	1	0	2 (2H)	Data
1	0	0	1	1	3 (3H)	Data
0	0	1	0	0	4 (4H)	Data
1	0	1	0	1	5 (5H)	Data
1	0	1	1	0	6 (6H)	Data
0	0	1	1	1	7 (7H)	Data
0	1	0	0	0	8 (8H)	Data
1	1	0	0	1	9 (9H)	Data
1	1	0	1	0	: (AH)	Control
0	1	0	1	1	; (BH)	Start Sentinel
1	1	1	0	0	< (CH)	Control
0	1	1	0	1	= (DH)	Field Seprator
0	1	1	1	0	> (EH)	Control
1	1	1	1	1	? (FH)	End Sentinel

ตารางที่ 2.2 ตารางแสดงข้อมูลของรหัส BCD สำหรับเครื่องอ่านแถบแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
สเตปปิงมอเตอร์

3.1 หลักการควบคุมสเตปปิงมอเตอร์

การที่สเตปปิงมอเตอร์ จะสามารถเคลื่อนที่ไปในทิศทางที่เราต้องการได้นั้น เราจะจ่ายกระแสไฟฟ้าให้กับขดลวดของมอเตอร์ เพื่อให้เกิดเป็นอำนาจของสนามแม่เหล็กไฟฟ้าขึ้น อย่างเป็นลำดับขั้นที่ถูกต้องและสัมพันธ์กัน ซึ่งที่ใช้ในการควบคุมการเคลื่อนที่ของสเตปปิงมอเตอร์นั้นสามารถทำได้หลายวิธี ในแต่ละวิธีการนั้น ต่างมีข้อดีข้อเสียต่างกันไป เพื่อให้เข้าใจถึงแนวทางในการควบคุมการเคลื่อนที่ของสเตปปิงมอเตอร์ได้ดียิ่งขึ้น จึงขอแบ่งแยกประเภทการควบคุมการทำงานของสเตปปิงมอเตอร์ ตามลักษณะโครงสร้างของขดลวด โดยเราสามารถแบ่งแยกได้ 2 แบบด้วยกัน คือ

1. สเตปปิงมอเตอร์แบบ 2 ขั้ว (bipolar stepping motor)
2. สเตปปิงมอเตอร์แบบหลายขั้ว (unipolar stepping motor)

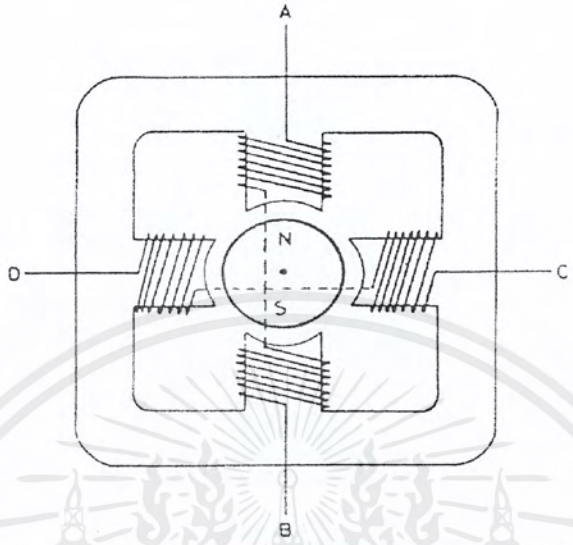
ซึ่งวิธีการสำหรับการควบคุมการทำงานของสเตปปิงมอเตอร์ ทั้ง 2 แบบนี้ มีวิธีการที่คล้ายกันโดยแบ่งเทคนิคการควบคุมการทำงานของสเตปปิงมอเตอร์ ได้ 3 แบบคือ

- การควบคุมแบบจ่ายกระแสไฟให้กับขดลวดครั้งละ 1 ขด
- การควบคุมแบบจ่ายกระแสไฟให้กับขดลวดครั้งละ 2 ขด
- การควบคุมแบบจ่ายกระแสไฟให้กับขดลวด 1 ขด สลับกับ 2 ขด

ในโครงงานนี้ได้สเตปปิงมอเตอร์ แบบ 2 ขั้ว หรือมอเตอร์ที่มีสาย 4 เส้น เพื่อใช้ในการยกที่กั้นใต้ท้องรถขึ้นจากระดับพื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การควบคุมสเตปปิงมอเตอร์แบบ 2 ขั้ว



รูปที่ 3.1 รูปแสดงโครงสร้างของไบโพลาร์สเตปปิงมอเตอร์

โดยทั่วๆ ไปแล้ว สเตปปิงมอเตอร์ที่มี 2 ขั้วนี้ จะเป็นสเตปปิงมอเตอร์ ชนิด PM เป็นส่วนใหญ่ ซึ่งจะมีขดลวดพันอยู่บนแกนสเตเตอร์ จำนวน 2 ขด ดังรูป การที่เราจะควบคุมให้สเตปปิงมอเตอร์ เคลื่อนที่ไปได้นั้น ทำได้โดยการจ่ายไฟเข้าขดลวดเพื่อให้เกิดเป็นสนามแม่เหล็กไฟฟ้าขึ้น ซึ่งจะส่งผลให้อำนาจของขั้วแม่เหล็กถาวรและแม่เหล็กไฟฟ้าเกิดการดูดผลักกัน จึงทำให้แม่เหล็กถาวรที่อยู่บนโรเตอร์เคลื่อนที่ไปได้ตามต้องการ ซึ่งเราสามารถกำหนดวิธีการควบคุมการจ่ายไฟเข้าขดลวดของมอเตอร์เพื่อให้มอเตอร์หมุนไปในทิศทางที่เราต้องการได้ 3 ลักษณะด้วยกันคือ

3.2.1 การควบคุมมอเตอร์แบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส หรือการควบคุมแบบฮาล์ฟไดร์ฟ (half drive) วิธีการควบคุมแบบนี้อาจเรียกได้หลายอย่างแตกต่างกันไป เช่น เวฟ ไดร์ฟ (wave drive)หรือ วันเฟสออน (one-phase-on) หรือ วันเอ็กไซต์เตชั่น (one-excitaton)ก็ได้ โดยในวิธีนี้สามารถทำได้โดยการจ่ายกระแสไฟฟ้าให้กับขดลวดของมอเตอร์เรียงกันไปครั้งละ 1 ขด เรียงต่อเนื่องกันไป วิธีการนี้จะมีข้อดีคือ การเคลื่อนที่ของมอเตอร์จะราบเรียบสม่ำเสมอดี แต่จะให้แรงบิดน้อย โดยลำดับขั้นตอนสามารถแสดงได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับที่	D	C	B	A	การทำงาน	ตำแหน่ง
1	0	0	0	1	กระแสไหลจาก A→ B ขดเดียว	
2	0	1	0	0	กระแสไหลจาก C→ D ขดเดียว	
3	0	0	1	0	กระแสไหลจาก B→ A ขดเดียว	
4	1	0	0	0	กระแสไหลจาก D→ C ขดเดียว	

ตารางที่ 3.1 ตารางแสดงการควบคุมแบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส

3.2.2 การควบคุมมอเตอร์โดยการจ่ายไฟให้กับขดลวดพร้อมกัน 2 เฟส หรือการควบคุมแบบฟูล สเตป (full step)ซึ่งในบางครั้งอาจเรียก ทูเฟสออน (two-phase-on) หรือ ทูเอ็กซ์ไซเตชั่น (two excitation)ก็ได้เช่นกัน ในวิธีนี้เราจะจ่ายกระแสไฟฟ้าให้กับขดลวดของมอเตอร์ครั้งละ 2 เฟส พร้อมกัน การควบคุมแบบนี้จะให้แรงบิดมาก โดยมีลำดับขั้นดังต่อไปนี้

ลำดับที่	D	C	B	A	การทำงาน	ตำแหน่ง
1	0	1	0	1	กระแสไหลจาก A→ B และ D→ C	
2	0	1	1	0	กระแสไหลจาก C→ D และ B→ A	
3	1	0	1	0	กระแสไหลจาก B→ A และ D→ C	
4	1	0	0	1	กระแสไหลจาก D→ C และ A→ B	

ตารางที่ 3.2 ตารางแสดงการควบคุมแบบจ่ายไฟให้กับขดลวดพร้อมกัน 2 เฟส

3.2.3 การควบคุมมอเตอร์โดยการจ่ายไฟให้กับขดลวด 1 เฟส สลับ 2 เฟส หรือการควบคุมแบบฮาล์ฟสเตป (half step) หรือในบางครั้งอาจเรียก วันทู เอ็กซ์ไซเตชั่น(one-two excitation) ในวิธีนี้เราจะจ่ายกระแสไฟฟ้าให้กับขดลวดของมอเตอร์ 1 เฟส สลับกับ 2 เฟส การควบคุมแบบนี้จะให้แรงบิดเฉลี่ยลดลง แต่ได้ความละเอียดแม่นยำของระยะทางการเคลื่อนที่เพิ่มขึ้นเป็น 2 เท่า ลำดับขั้นตอนสามารถแสดงได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับที่	D	C	B	A	การทำงาน	ตำแหน่ง
1	0	0	0	1	กระแสไหลจาก A→ B ขดเดียว	
2	0	1	0	1	กระแสไหลจาก A→ B และ D→ C	
3	0	1	0	0	กระแสไหลจาก C→ D ขดเดียว	
4	0	1	1	0	กระแสไหลจาก C→ D และ B→ A	
5	0	0	1	0	กระแสไหลจาก B→ A ขดเดียว	
6	1	0	1	0	กระแสไหลจาก B→ A และ D→ C	
7	1	0	0	0	กระแสไหลจาก D→ C ขดเดียว	
8	1	0	0	1	กระแสไหลจาก D→ C และ A→ B	

ตารางที่ 3.3 ตารางแสดงการควบคุมแบบจ่ายไฟให้ขดลวด 1 เฟส สลับ 2 เฟส

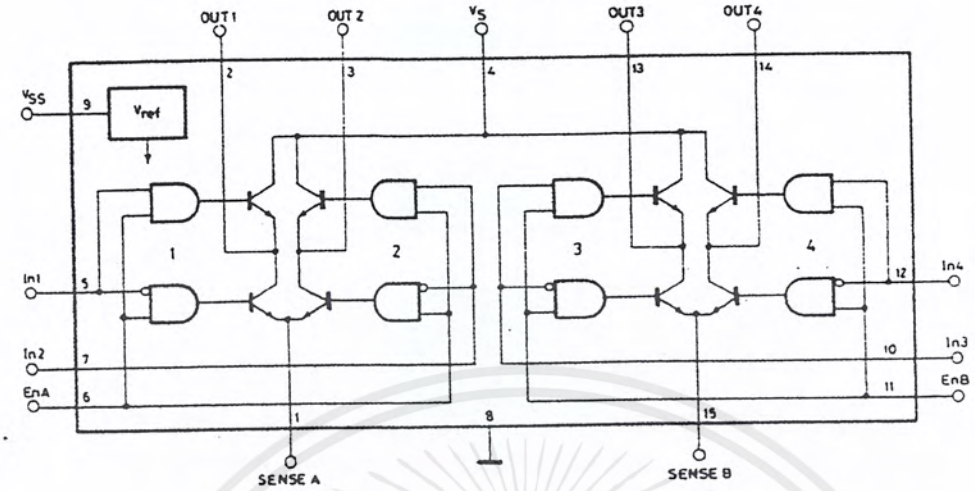
วิธีการที่กล่าวมาทั้ง3วิธีนั้นเป็นการควบคุมทิศทางการเคลื่อนที่ของมอเตอร์เท่านั้น สำหรับในเรื่องของความเร็วในการเคลื่อนที่นั้นต้องกำหนดจากอัตราความเร็วของสัญญาณพัลส์ที่ใช้สำหรับควบคุมการทำงานในแต่ละสเตป ซึ่งความถี่ของสัญญาณพัลส์นี้ต้องขึ้นอยู่กับความสามารถของมอเตอร์ที่ใช้งานด้วยว่ามีอัตราการสเตป (step rate) เป็นเท่าไร เนื่องจากตัวมอเตอร์นั้นเป็นอุปกรณ์ทางกลซึ่งทำงานได้ช้าถ้าหากเราสร้างสัญญาณควบคุมที่มีอัตราการสเต็ปที่เร็วเกินความสามารถของมอเตอร์ เช่นสั่งให้เคลื่อนที่ไปยังสเตปที่ 2 ในขณะที่มอเตอร์ยังเคลื่อนที่ไปยังสเตปที่ 1 ไม่เสร็จสมบูรณ์แล้วก็จะทำให้เกิดการผิดพลาดขึ้น ซึ่งมอเตอร์อาจแสดงออกด้วยอาการสั่น หรือไม่หมุนเลย หรืออาจหมุนกลับไปกลับมาไม่แน่นอน เป็นต้น

3.3 การควบคุมสเตปปิงมอเตอร์แบบ 2 ขั้ว ในส่วนของการขับมอเตอร์

ในส่วนของวงจรขับกระแสให้กับมอเตอร์นั้นจะใช้ IC เบอร์ L298N ของ SGS-THOMSON ซึ่งสามารถแบ่งสัญญาณการควบคุมการทำงานออกเป็น 2 ชุด คือ

- EN1, INPUT-A, INPUT-B สำหรับขับกระแสให้กับขดลวดขดที่ 1 ของมอเตอร์
- EN2, INPUT-C, INPUT-D สำหรับขับกระแสให้กับขดลวดขดที่ 2 ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงบล็อกไดอะแกรมของ L298N

โดยสัญญาณ EN1 เป็นสัญญาณควบคุมหลักทำงานที่ลอจิก “1” เพื่อใช้สำหรับเปิด-ปิดการไหลของกระแสของเอาต์พุต A และ B

ส่วนสัญญาณ EN2 เป็นสัญญาณควบคุมหลักทำงานที่ลอจิก “1” เพื่อใช้สำหรับเปิด-ปิดการไหลของกระแสของเอาต์พุต C และ D

ซึ่งเราสามารถแสดงให้เห็นความสัมพันธ์ของสัญญาณต่างๆ ได้ดังตาราง

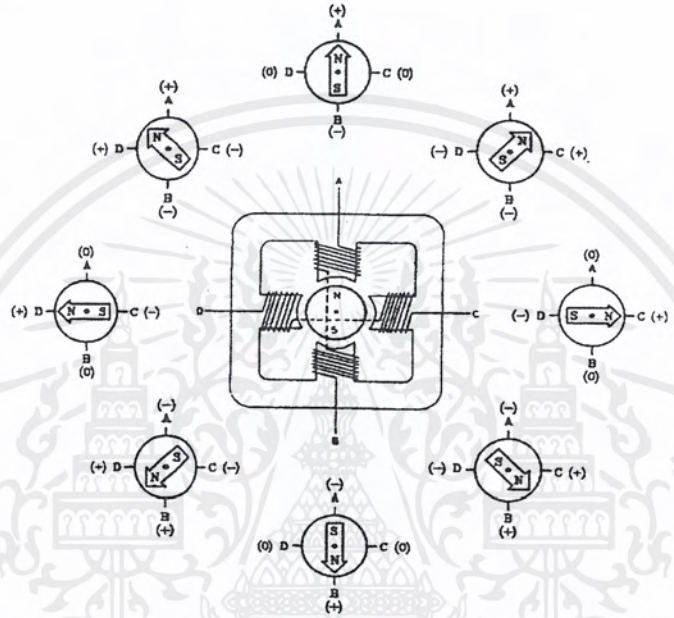
EN1	INPUT-B	INPUT-A	การทำงานของมอเตอร์
“1”	“0”	“0”	ไม่มีกระแสไหลมอเตอร์หยุดหมุน
	“0”	“1”	กระแสไหลจาก A ไป B
	“1”	“0”	กระแสไหลจาก B ไป A
	“1”	“1”	ไม่มีกระแสไหลมอเตอร์หยุดหมุน
“0”	X	X	ไม่มีกระแสไหลมอเตอร์หยุดหมุน

EN2	INPUT-D	INPUT-C	การทำงานของมอเตอร์
“1”	“0”	“0”	ไม่มีกระแสไหลมอเตอร์หยุดหมุน
	“0”	“1”	กระแสไหลจาก C ไป D
	“1”	“0”	กระแสไหลจาก D ไป C
	“1”	“1”	ไม่มีกระแสไหลมอเตอร์หยุดหมุน
“0”	X	X	ไม่มีกระแสไหลมอเตอร์หยุดหมุน

ตารางที่ 3.4 ตารางแสดงการควบคุมไบโพลาร์สเตปปิงมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่า L298N แบ่งสัญญาณออกเป็น 2 กลุ่ม เพื่อใช้สำหรับแยกควบคุมขดลวดของมอเตอร์ 2 ขด โดยในแต่ละขดจะแยกออกจากกันอย่างอิสระไม่เกี่ยวของซึ่งกันและกัน และในขณะเดียวกันในแต่ละขดลวดของมอเตอร์นั้นก็ยังสามารถกำหนดทิศทางการไหลของกระแสในขดลวดได้ทั้งสองทิศทาง



รูปที่ 3.3 แสดงลักษณะของไบโพลาร์สเตปปิงมอเตอร์

ลำดับที่	EN2	EN1	IN-D	IN-C	IN-B	IN-A	ตำแหน่ง
1	1	1	1	0	1	0	
2	1	0	1	0	0	0	
3	1	1	1	0	0	1	
4	0	1	0	0	0	1	
5	1	1	0	1	0	1	
6	1	0	0	1	0	0	
7	1	1	0	1	1	0	
8	0	1	0	0	1	0	

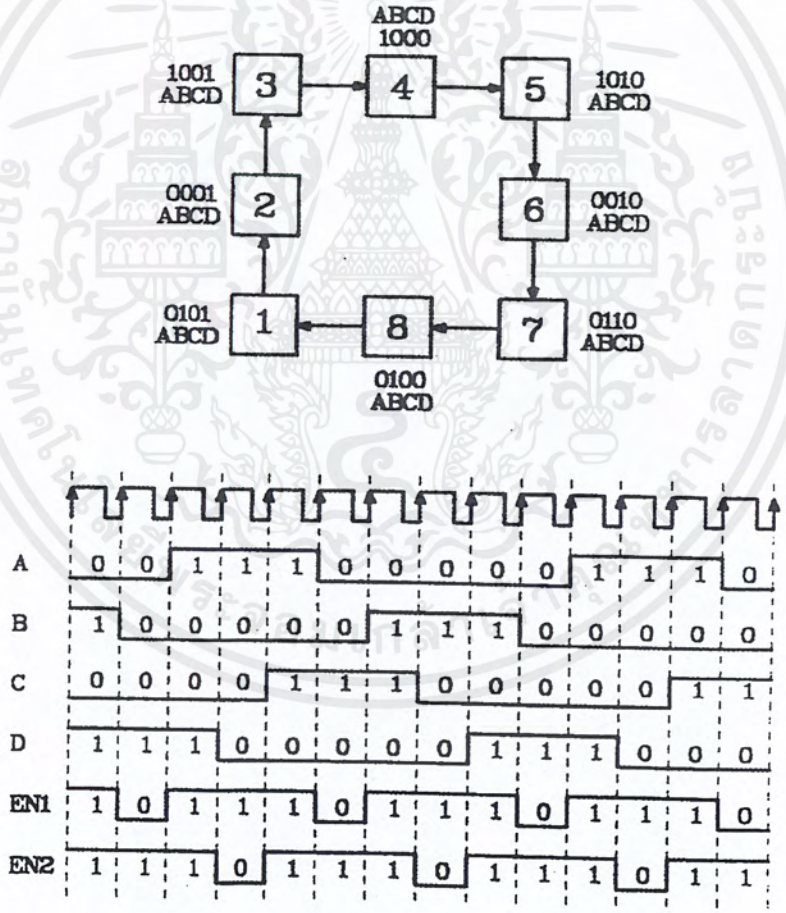
ตารางที่ 3.5 ตารางแสดงการจ่ายกระแสให้มอเตอร์และการสร้างสัญญาณควบคุม L298N

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานนี้จะทำการควบคุมสเตปปิงมอเตอร์ แบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส สลับ 2 เฟส

3.4 การควบคุมแบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟส สลับ 2 เฟส

สำหรับการควบคุมมอเตอร์แบบนี้ เหมาะกับงานที่ต้องการความละเอียดแม่นยำในการเคลื่อนที่มากๆ เพราะสามารถให้ความละเอียดของการเคลื่อนที่มากกว่าแบบอื่นๆ ถึง 2 เท่า เพราะในแบบอื่นๆ นั้น เมื่อป้อนกระแสไฟฟ้าให้เข้ากับขดลวด 1 จังหวะจะทำให้มอเตอร์เคลื่อนที่ไปได้ 1 สเตป แต่การควบคุมแบบนี้จะทำให้มอเตอร์เคลื่อนที่ได้เพียงครึ่งสเตปเท่านั้น จึงให้ความละเอียดในการเคลื่อนที่ดีกว่าแบบอื่นๆ แต่แรงบิดเฉลี่ยจะลดลง



รูปที่ 3.4 แสดงการควบคุมแบบจ่ายไฟให้ขดลวดครั้งละ 1 เฟสสลับ 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.5 การกำหนดพิกัดของกระแสสำหรับมอเตอร์

สำหรับการกำหนดพิกัดกระแสสูงสุดของเอาต์พุทนั้น สามารถกำหนดได้ด้วยการปรับค่าความต้านทานแบบปรับค่าได้ VR1 แล้ววัดระดับแรงดันที่ขา 3 หรือ 5 ของออปแอมป์ LM358 เทียบกับกราวน์ของวงจร ซึ่งค่าระดับของแรงดันนี้สามารถคำนวณได้จากสูตร

$$\text{แรงดัน} = \text{ค่ากระแสสูงสุดที่ต้องการ} \times 0.5$$

ตัวอย่างเช่น ต้องการพิกัดกระแสสูงสุด 2 แอมป์ จะได้ว่า

$$\text{แรงดัน} = 2 \times 0.5$$
$$= 1 \text{ โวลท์}$$

ดังนั้นจึงต้องทำการปรับค่าความต้าน VR1 จนได้ค่าแรงดันที่ขา 3 หรือ 5 ของออปแอมป์ LM358 จนได้แรงดันประมาณ 1 โวลท์

อย่างไรก็ดี การกำหนดพิกัดของกระแสนี้ควรกำหนดให้สูงกว่ากระแสที่มอเตอร์ต้องการเล็กน้อย เช่น มอเตอร์ต้องการกระแส 1 แอมป์ เราอาจกำหนดไว้เป็น 1.5 แอมป์ ทั้งนี้ก็เนื่องมาจากถ้าเราตั้งพิกัดของกระแสเท่ากับความต้องการของมอเตอร์พอดีอาจทำให้มอเตอร์ไม่ทำงาน แต่ทั้งนี้ทั้งนั้นการกำหนดพิกัดกระแสสูงสุดนี้ต้องไม่เกิน 4 แอมป์ (2 โวลท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

45722

บทที่ 4
รีโมทอินฟาเรด 25 ฟุต(รับและส่ง)และเครื่องหยอดเหรียญ

4.1 รีโมทอินฟาเรด 25 ฟุต (รับและส่ง)

รีโมทอินฟาเรด นี้สามารถนำไปควบคุมอุปกรณ์ไฟฟ้า ให้ปิดเปิดได้ตามที่ต้องการ วงจรนี้จะแยกออก 2 ส่วน คือ ภาคส่ง จะใช้ไฟ 9 โวลต์ และภาครับจะใช้ไฟ 9 โวลต์ โดยภาครับนี้จะไปควบคุมอุปกรณ์ไฟฟ้า โดยสามารถต่อกับเครื่องใช้ไฟฟ้าได้ไม่เกิน 300 วัตต์

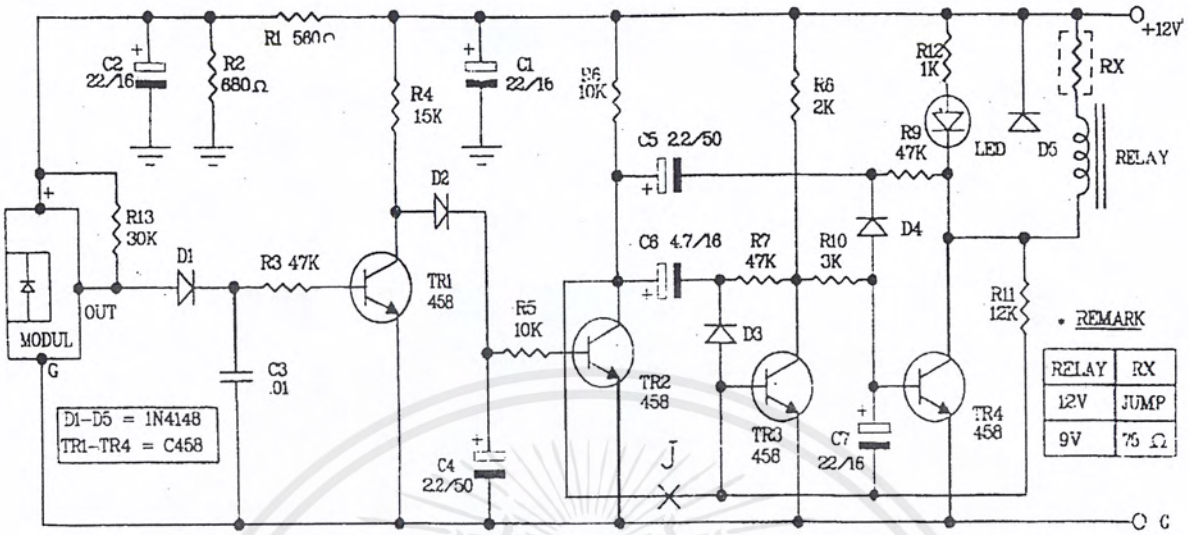
การทำงาน แยกการทำงานออกเป็น 2 ภาค

ภาคส่ง TR1, TR2 ต่อเป็นวงจรสำหรับกำเนิดความถี่ โดยความถี่นี้จะส่งไปควบคุม TR3, TR4 โดย TR3, TR4 นี้จะสร้างความถี่ประมาณ 39-41 KHz ความถี่ทั้ง 2 ความถี่นี้จะส่งเข้าTR5 เพื่อทำการขยาย และส่งออกทาง LED อินฟาเรด

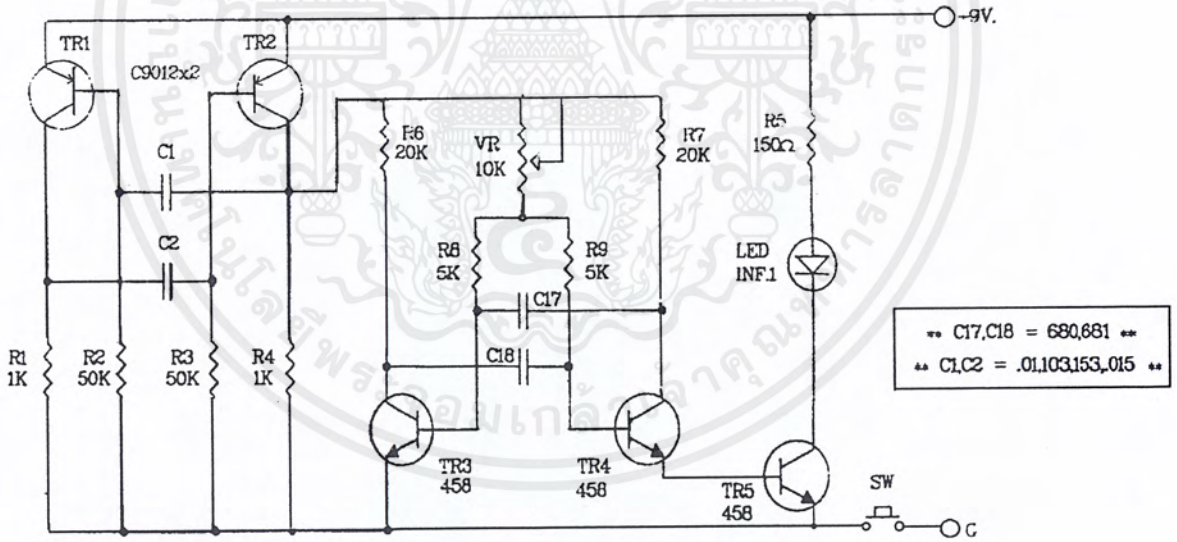
ภาครับ โมดูลภาครับจะทำหน้าที่รับคลื่นแสงอินฟาเรดความถี่ 39-41 KHz ที่ส่งมา เพื่อถอดรหัส แล้วให้เอาท์พุตออกมาทางจุด OUT ซึ่งจะต่อผ่าน D1 โดยมี C3 ทำหน้าที่เป็นฟิลเตอร์แล้วส่งเข้า TR1, TR2 เพื่อทำการขยายสัญญาณให้แรงขึ้น ที่ขาคอลเลคเตอร์ของ TR2 จะต่อเข้าที่ฟลิบฟลอบเพื่อควบคุมให้รีเลยทำงานตามที่เราควบคุมที่ภาคส่ง ชุดฟลิบฟลอบนี้จะประกอบด้วย TR3, TR4 โดย TR ทั้งสองตัวนี้จะสลับกันทำงาน เมื่อได้รับสัญญาณจากการส่ง

การทดสอบ จากรูปที่ 4.1 จุด J ยังไม่ต้องต่อถึงกัน จ่ายไฟเข้าที่ภาครับและที่ภาคส่ง จากนั้นหันหน้า LED อินฟาเรดให้ตรงกับโฟโต้ไดโอดตัวสีดำซึ่งอยู่ที่ภาครับ แล้วกดสวิตซ์ที่ภาคส่ง ตอนนี้ LED ที่ภาครับจะติด และรีเลย์จะหยุดทำงาน ถ้าวงจรไม่ทำงานตามที่กล่าวมาให้ทดลองปรับโวลุ่มเกือกม้าที่ภาคส่ง ถ้าทดลองได้ตามที่กล่าวมา ให้ค่อย ๆ เอาภาคส่งออกห่างจากภาครับแล้วกดสวิตซ์ที่ภาคส่ง สังเกต LED จะติดดับ ตามที่เรากดสวิตซ์ที่ภาคส่ง ถ้าไม่ทำงานให้ทดลองปรับโวลุ่มเกือกม้าที่ภาคส่งอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงวงจรรีโมทอินฟาเรดส่วนของตัวรับ



รูปที่ 4.2 แสดงวงจรรีโมทอินฟาเรดส่วนของตัวส่ง

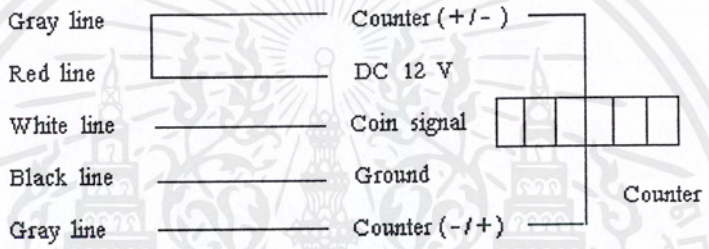
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 เครื่องหยอดเหรียญ

ข้อกำหนด : - ใช้กับไฟ ดีซี 12 โวลต์

- ใช้กับเหรียญที่มีเส้นผ่านศูนย์กลาง 18 – 31 มม.
- ใช้กับเหรียญที่มีความหนา 1.2 – 3.0 มม.
- ใช้ในช่วงอุณหภูมิ -15 °C - 75 °C

สำหรับขาที่ใช้ต่อ ของเครื่องหยอดเหรียญ มีดังต่อไปนี้ :



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โมดูล LCD

5.1 รายละเอียดเกี่ยวกับโมดูล LCD

ในโมดูล LCD ประกอบไปด้วย 3 ส่วนหลักๆ คือ

5.1.1 ตัวแสดงผล (display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็น โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

5.1.2 ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟฟิก

5.1.3 ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

5.2 โครงสร้างภายในของตัวควบคุม โมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน

5.2.1 บัฟเฟอร์อินพุตเอาท์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

5.2.2 รีจิสเตอร์คำสั่ง (instruction register: IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

5.2.3 รีจิสเตอร์ข้อมูล (data register: DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

5.2.4 แรมเก็บข้อมูลแสดงผล (display data RAM: DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

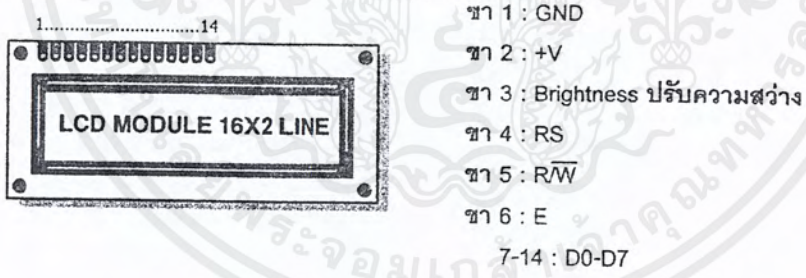
5.2.5 รอมเก็บตัวอักษร (character generator ROM: CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านข้อมูลแล้วไปแสดงที่ตัวแสดงผลได้ ซึ่งมีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

5.2.6 แรมเก็บตัวอักษร (character generator RAM: CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

5.2.7 แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

5.3 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16 x 2)

โมดูล LCD ขนาด 16 x 2 มีขาต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดการทำงานของแต่ละขามีดังนี้



รูปที่ 5.1 แสดงขาต่าง ๆ ของจอ LCD แบบ 16 ตัวอักษร 2 บรรทัด

Vss (ขา 1) ใช้สำหรับต่อกราวด์

Vdd (ขา 2) ต่อไฟเลี้ยง +5 โวลท์

Vo (ขา 3) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4) เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โ คยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R/W (ขา 5) เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E (ขา 6) เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14) เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

5.4 คำสั่งควบคุม โมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น "0" แล้วเขียนคำสั่งควบคุม คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมีดังนี้

5.4.1 คำสั่งเคลียร์ตัวแสดงผล (clear display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างเข้าไปใน DDRAM เมื่อตัวควบคุมเอ็กซ์คิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น "0" เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวทีหลัง) ให้เป็น "1"

5.4.2 คำสั่ง return home

ต้องกำหนดให้บิต1 ของข้อมูลเป็น"1"เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

5.4.3 คำสั่งเลือกโหมดการป้อนข้อมูล (entry mode set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต 6	บิต 5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น "1"เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น "0" เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0" แอดเดรสจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H - 07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือและแอดเดรสของ DDRAM เพิ่มขึ้น

5.4.4 คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น “1” จะเป็นการเปิดจอแสดงผล ถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดเป็น “0” จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกะพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกะพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H – 0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กะพริบ

5.4.5 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต 3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

5.4.6 คำสั่งกำหนดฟังก์ชั่นการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	1	DL	N	F	*	*

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น "0" จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น "1" จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น "0" จะแสดงผล 1 บรรทัด ถ้าเป็น "1" จะแสดง 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น "1"

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น "0"จะเป็นการแสดงผลแบบ 5 x 7 จุด และถ้าเป็น "1" จะแสดงผลเป็นแบบ 5 x 10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนด โมดูล LCDให้ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5 x 7 จุด

5.4.7 คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น "0" บิต 6 เป็น "1" ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

5.4.8 คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูลนั้น ที่บิต 7 ต้องเป็น "1" และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของDDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น "0" แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.9 คำสั่งอ่านแฟลก BUSY และแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (ในตารางจะกำหนดเป็น BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น "0" แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น "1" แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น "1" ด้วย แต่สัญญาณที่ RS ยังต้องเป็น "0" อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0-6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

5.5 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องป้อนคำสั่ง แล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูลที่ต้องการแสดงผล เนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก "0" หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่ง ในทางตรงกันข้าม หากขา RS ได้รับลอจิก "1" ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น "1" เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น “1” ข้อมูลในขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

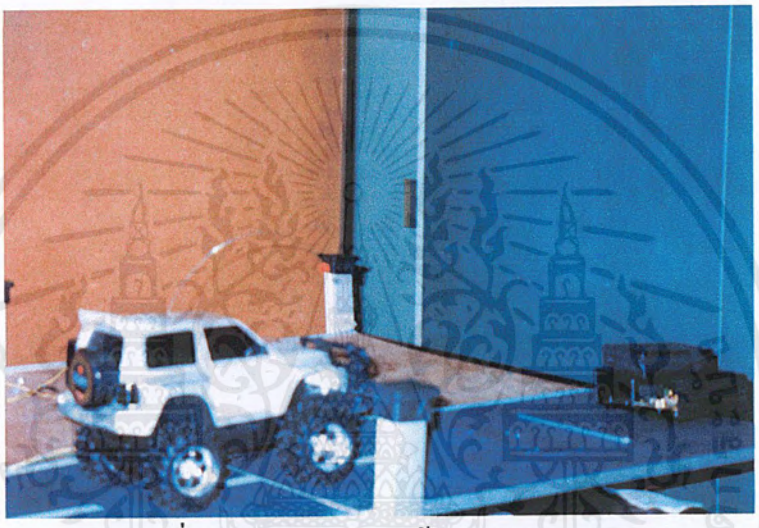
ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “1” ให้ขา RS แล้ว ต้องกำหนดให้ขา R/W เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ผลการทดลอง

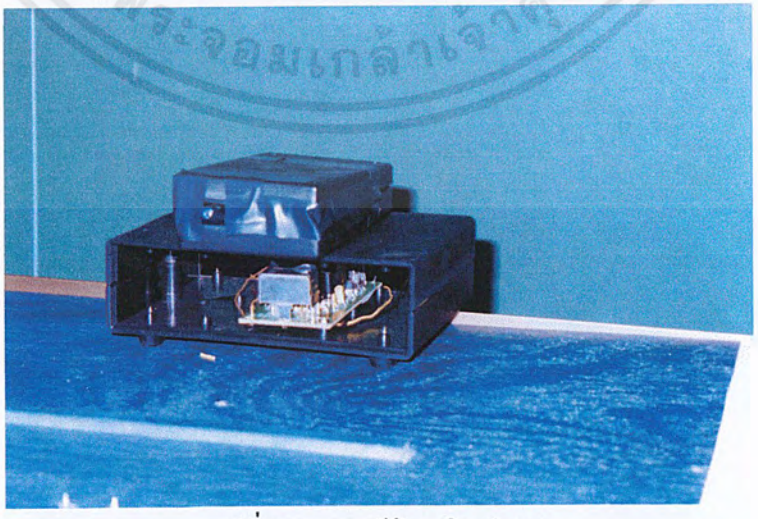
ผลการทดลองที่ 1 อธิบายขั้นตอนการนำรถเข้าจอดในอาคารจอดรถจำลอง จนกระทั่งนำรถออกจากอาคารจอดรถจำลอง

ขั้นที่ 1.1 นำรถเข้าจอด ณ จุดจอดรถ



รูปที่ 6.1 แสดงการนำรถเข้าจอด ณ จุดจอดรถ

ขั้นที่ 1.2 หลังจากรีโมทอินฟาเรดตรวจจับรถได้รีโมทอินฟาเรดจะทำการส่งสัญญาณไปยังไมโครคอนโทรเลอร์



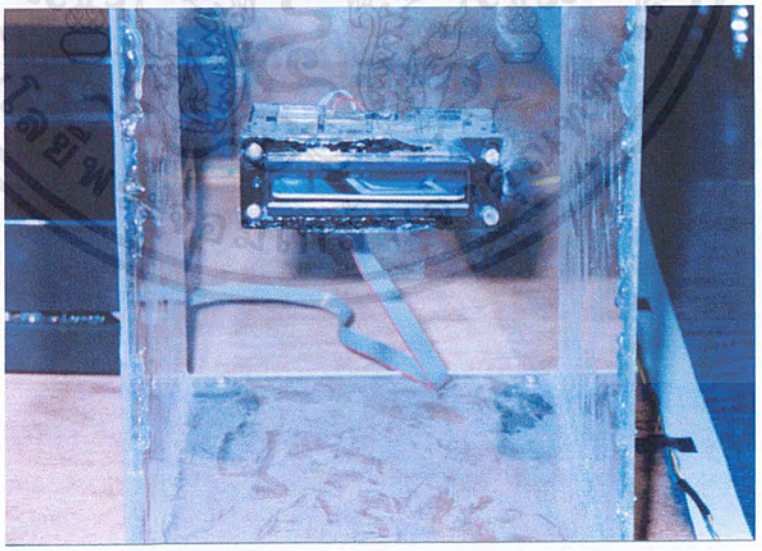
รูปที่ 6.2 แสดงรีโมทอินฟาเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 1.3 ไมโครคอนโทรลเลอร์ทำการบันทึกเวลาเข้าจอดรถ พร้อมทั้งสั่งการให้สเตปปิงมอเตอร์ทำงาน เพื่อยกที่กั้นใต้ท้องรถขึ้นจากระดับพื้น

รูปที่ 6.3 แสดงที่กั้นใต้ท้องรถที่ถูกยกขึ้นด้วยการทำงานของสเตปปิงมอเตอร์

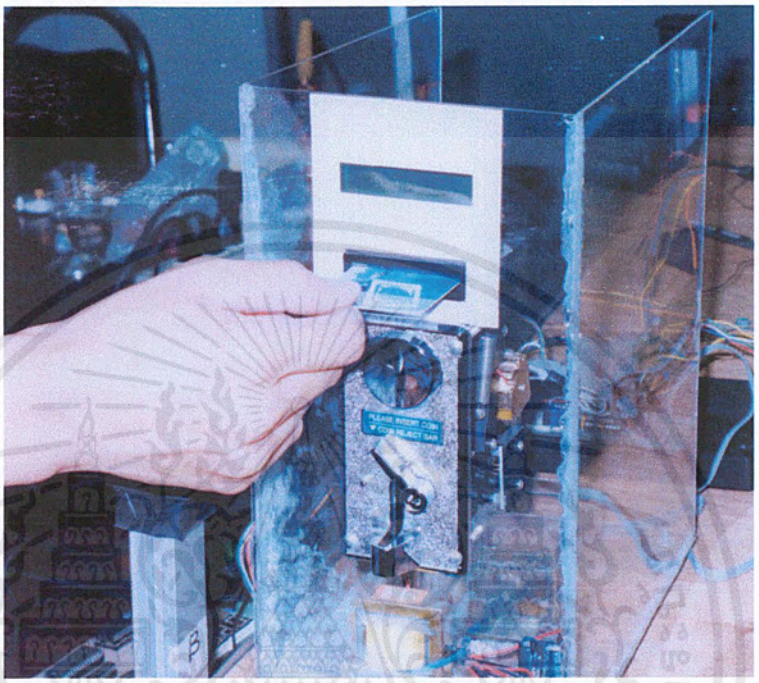
ขั้นที่ 1.4 ทำการดึงบัตรแถบแม่เหล็กออกจากกล่องอ่านบัตรแถบแม่เหล็กที่อยู่ ณ จุดจอดรถ



รูปที่ 6.4 แสดงบัตรแถบแม่เหล็กที่สอดอยู่ที่กล่องอ่านบัตรแถบแม่เหล็ก ณ จุดจอดรถ

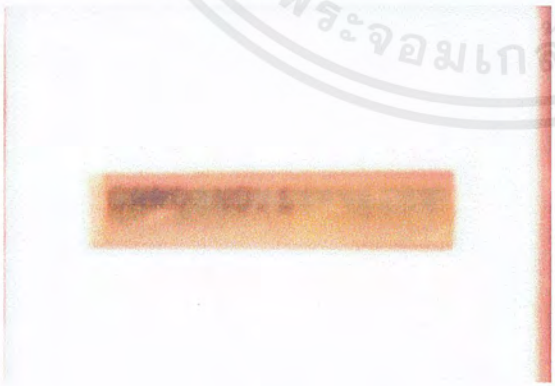
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 1.5 เมื่อต้องการนำรถออกจากอาคารจอดรถ ให้นำบัตรแถบแม่เหล็กสอดยังกล่องอ่านบัตรแถบแม่เหล็กที่ใช้คิดค่าบริการจอดรถเพื่อบันทึกเวลาออก พร้อมทั้งคำนวณค่าบริการจอดรถ

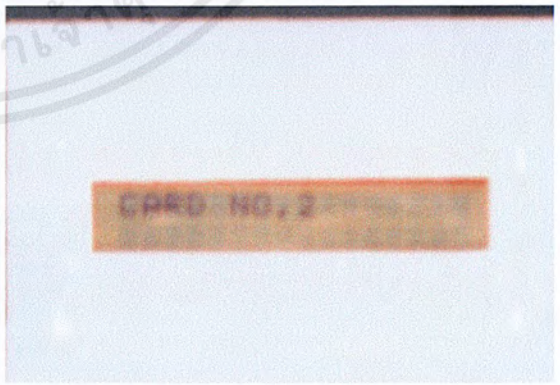


รูปที่ 6.5 แสดงขณะทำการสอดบัตรแถบแม่เหล็กในกล่องอ่านบัตรแถบแม่เหล็กตัวที่ใช้คิดค่าบริการจอดรถ

ขั้นที่ 1.6 ไมโครคอนโทรลเลอร์คำนวณค่าบริการจอดรถ โดยแสดงค่าบริการจอดรถทางจอ LCD

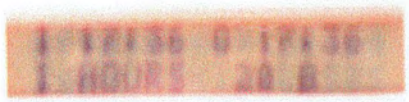


รูปที่ 6.6.1



รูปที่ 6.6.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6.3

รูปที่ 6.6.4

รูปที่ 6.6.5

รูปที่ 6.6.6

รูปที่ 6.6 แสดงจอ LCD ที่ปรากฏขั้นตอนการคิดค่าบริการจอดรถ

จากรูป แสดงผลที่ปรากฏทางจอ LCD หลังจากสอดบัตรจอดรถ โดยใช้บัตรจอดรถ 2 ใบ บัตรจอดรถชั้นล่าง จะแสดงผลทางจอ LCD เป็น “CARD NO.1” บัตรจอดรถชั้นบน จะแสดงผลทางจอ LCDเป็น “CARD NO.2” ดังรูปที่ 6.6.1 และรูปที่ 6.6.2 ตามลำดับ จากนั้นที่หน้าจอ LCD จะแสดงเวลาออก เวลาทั้งหมดที่เข้าจอด พร้อมทั้งจำนวนเงินที่ต้องชำระ โดยเศษของเวลาจะทำการปัดเป็น 1 ชั่วโมง

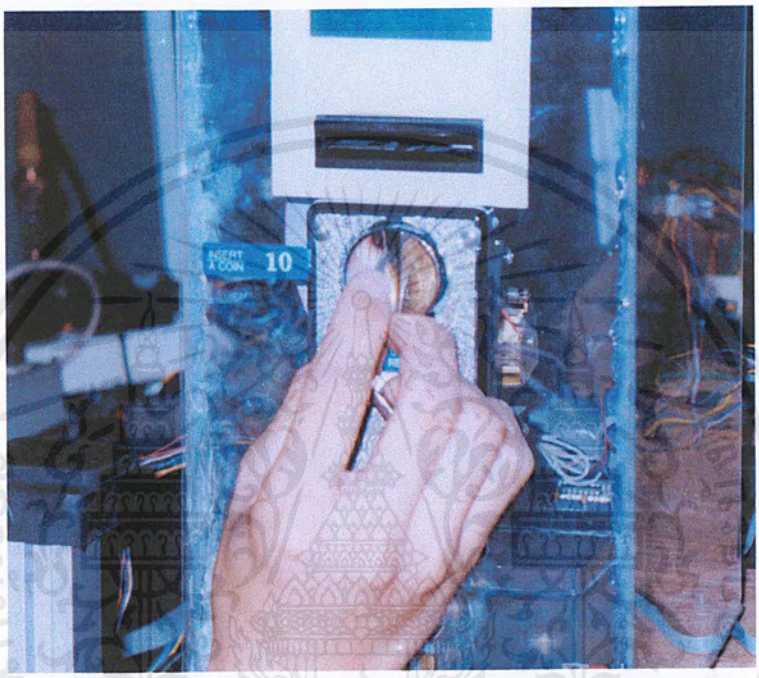
กรณีรูปที่ 6.6.3 ซึ่งเป็นการจอดรถชั้นล่าง จะคิดค่าบริการจอดรถ 20 บาทต่อ 1 ชั่วโมง แต่ถ้าจอดรถชั้นบน จะคิดค่าบริการ 10 บาทต่อ 1 ชั่วโมง ดังรูปที่ 6.6.4

กรณีรูปที่ 6.6.5 และรูปที่ 6.6.6 หน้าจอ LCD จะแสดงข้อความ “INSERT COIN” ในบรรทัดบนของจอ LCD ส่วนบรรทัดล่างทางด้านซ้ายจะแสดงจำนวนเงินที่ได้ทำการหยอดเหรียญลงไปแล้ว ส่วนทางขวาจะแสดงจำนวนเงินทั้งหมดที่ต้องชำระ โดยเริ่มต้นหากยังไม่มีการหยอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

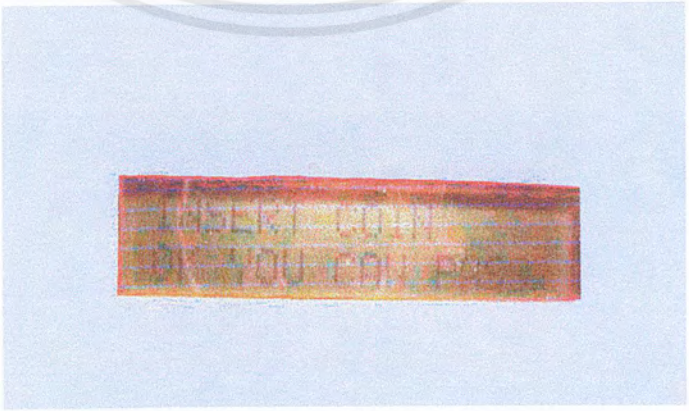
เหรียญ ทางด้านซ้ายของจอ LCD จะยังแสดงข้อความ “00 B” และเมื่อมีการหยอดเหรียญแล้ว ทางด้านซ้ายของจอ LCD จะแสดงจำนวนเงินที่ได้หยอดเหรียญลงไป

ขั้นที่ 1.7 ทำการหยอดเหรียญ 10 บาท เพื่อชำระค่าบริการจอดรถ



รูปที่ 6.7 แสดงขณะหยอดเหรียญลงในเครื่องหยอดเหรียญ

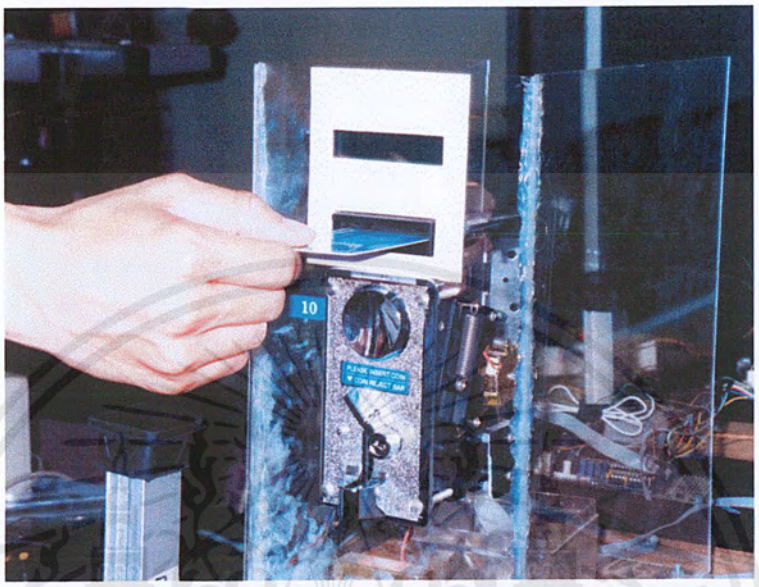
ขั้นที่ 1.8 เมื่อทำการชำระค่าบริการจอดรถเรียบร้อยแล้ว หน้าจอ LCD จะปรากฏข้อความ “ OK. YOU CAN PASS ”



รูปที่ 6.8 แสดงหน้าจอ LCD

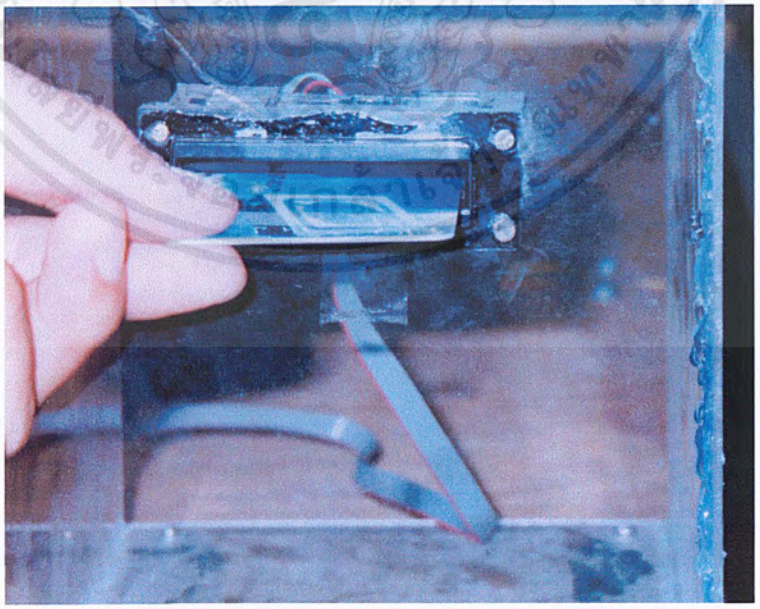
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 1.9 เมื่อทำการดึงบัตรแถบแม่เหล็กออกจากกล่องบัตรแถบแม่เหล็กตัวที่ใช้คิดค่าบริการจอดรถ ข้อความบนจอ LCD จะหายไป



รูปที่ 6.9 แสดงขณะทำการดึงบัตรแถบแม่เหล็กออกจากกล่องบัตรแถบแม่เหล็กตัวใช้คิดค่าบริการจอดรถ

ขั้นที่ 1.10 นำบัตรแถบแม่เหล็กสอดคืนที่กล่องอ่านบัตรแถบแม่เหล็ก ณ จุดจอดรถ



รูปที่ 6.10 แสดงการสอดบัตรคืน ณ จุดจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 1.11 หลังจากสอดบัตรคีนไมโครคอนโทรลเลอร์จะสั่งการให้ สเตปปิงมอเตอร์หมุนกลับทาง เพื่อลดระดับที่กั้นใต้ท้องรถลงดังเดิม

รูปที่ 6.11 แสดงที่กั้นใต้ท้องรถเลื่อนลงมา ณ ระดับพื้นดังเดิม

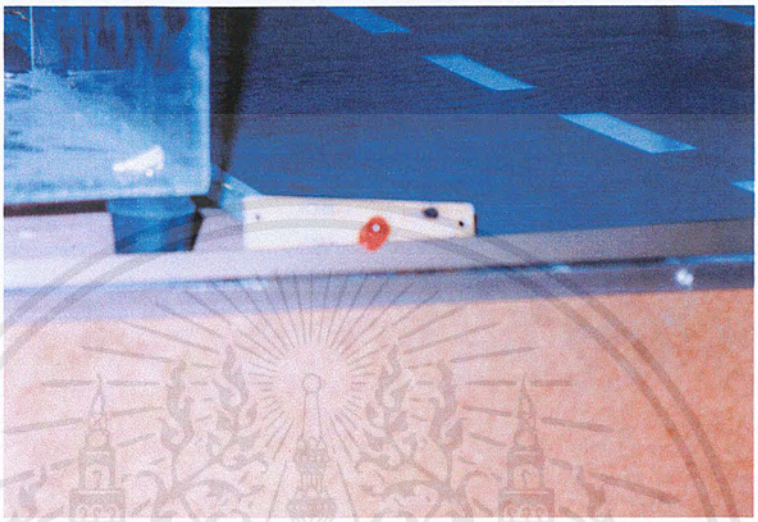
ขั้นที่ 1.12 นำรถออกจากที่จอดรถ

รูปที่ 6.12 แสดงการนำรถออกจากที่จอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองที่ 2 แสดงสถานะของที่จอดรถด้วยไฟแดงหน้าทางเข้าของอาคารจอดรถ

ขั้นที่ 2.1 จำนวนที่จอดรถยังไม่เต็ม ไฟแดงหน้าทางเข้าอาคารจอดรถจะยังไม่สว่าง



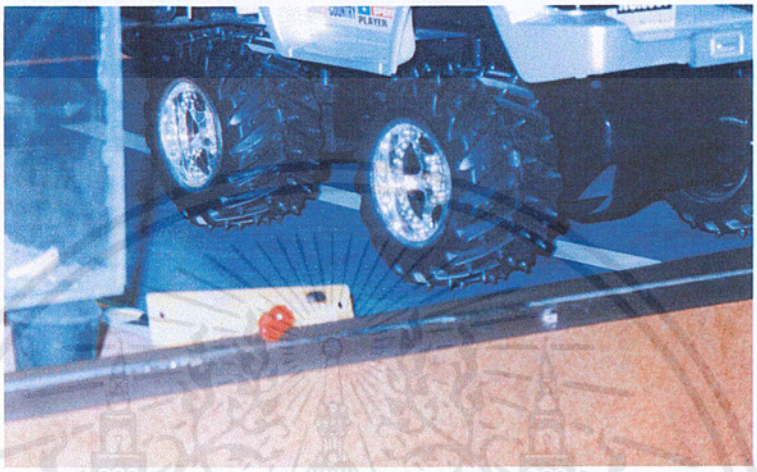
รูปที่ 6.13 แสดงไฟแดงหน้าทางเข้าอาคารจอดรถที่ดับอยู่

ขั้นที่ 2.2 เมื่อมีการดึงบัตรจอดรถออกจากกล่องอ่านบัตรแถบแม่เหล็ก 2 ใบ(แทนการจอดรถ 2 คัน ในแต่ละชั้นของอาคาร) ไฟแดงหน้าทางเข้าอาคารจะสว่างขึ้น แสดงความหมายว่า ที่จอดรถในอาคารเต็มแล้ว รถคันต่อมาไม่สามารถเข้าจอดในอาคารได้

รูปที่ 6.14 แสดงไฟแดงหน้าทางเข้าอาคารจอดรถสว่างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 2.3 เมื่อเจ้าของรถคันหนึ่งทำการชำระเงินค่าจอดรถ พร้อมทั้งดึงบัตรจอดรถออกจากกล่องอ่านบัตรแถบแม่เหล็กที่ใช้คิดค่าบริการจอดรถ ไฟแดงหน้าทางเข้าอาคารจอดรถจะดับลง เพื่อบอกให้รู้ว่าขณะนี้มีที่จอดรถว่างอยู่ สามารถนำรถเข้าจอดได้



รูปที่ 6.15 รูปแสดงไฟแดงหน้าทางเข้าอาคารจอดรถดับลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปและวิเคราะห์ผลการดำเนินงาน

7.1 สรุปผลการดำเนินงาน

จากการศึกษาและทดลองของโครงงานนี้จะแบ่งงานออกเป็น 2 ส่วน ได้แก่ ส่วนโครงสร้างทางเครื่องกลและวงจรอิเล็กทรอนิกส์ กับส่วนควบคุมคือส่วนของโปรแกรม ในส่วนของโครงสร้างและวงจรนั้นได้ทำการออกแบบและติดตั้งอุปกรณ์ไว้อย่างเหมาะสม ทำให้สามารถตรวจสอบจุดผิดพลาดขณะทำการทดลองได้ง่าย และในส่วนโปรแกรมจะใช้โปรแกรมภาษาแอสเซมบลีเพื่อให้โครงงานดังกล่าวทำงานได้อัตโนมัติ ไม่จำเป็นต้องติดต่อผ่านคอมพิวเตอร์

7.2 ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

ปัญหาหลักในการดำเนินงานแบ่งออกได้ดังนี้

7.2.1 โครงสร้างทางเครื่องกล และวงจรอิเล็กทรอนิกส์

ที่จุดต่อระหว่างเพลามอเตอร์และเกลียวน็อต ควรจะเชื่อมต่อกันให้แน่นสนิทเพื่อป้องกันไม่ให้เพลาและเกลียวเกิดการขัดกันขณะเพลาหมุน และวงจรอิเล็กทรอนิกส์ควรจะมีการเชื่อมสายอย่างระมัดระวัง เพื่อป้องกันสายหลุดขณะทำการทดลอง

7.2.2 ส่วนของโปรแกรม ควรจะมีการแบ่งแยกข้อมูลออกเป็นส่วน ๆ ให้ชัดเจน เพื่อให้ง่ายต่อการเรียกใช้และแก้ไขโปรแกรม

7.3 แนวทางการพัฒนาในอนาคต

โครงงานนี้สามารถนำไปพัฒนาและประยุกต์ใช้งานในอาคารจอดรถ โดยการเพิ่มจำนวนชุดของเครื่องอ่านบัตรแถบแม่เหล็กตามจำนวนที่จอดรถ และในส่วนของที่กั้นใต้ท้องรถเราอาจจะเปลี่ยนจากสเตปปิงมอเตอร์เป็นชุดไฮดรอลิก เพื่อเพิ่มแรงดันให้ทำการยกที่กั้นใต้ท้องรถขึ้นแทน

ในความเป็นจริงปัญหาที่เกิดขึ้นคือ รถที่จอดแต่ละคันนั้นมีลักษณะแตกต่างกัน เป็นผลให้ช่วงล่างของรถแต่ละคันมีความสูงเหนือพื้นที่แตกต่างกัน ดังนั้นหากเรากำหนดให้ที่กั้นใต้ท้องรถยกขึ้นเหนือพื้นดินเป็นมุมเอียงที่เท่าเดิมตลอด อาจจะทำให้ที่กั้นดังกล่าวชนกับใต้ท้องรถบางคันที่ต่ำเกินไป หรือกรณีที่รถบางคันมีช่วงล่างสูงเหนือพื้นมาก ทำให้สามารถขับข้ามที่กั้นใต้ท้องรถนั้นได้ ซึ่งปัญหาดังกล่าวแก้ไขได้โดยติดตั้งเซ็นเซอร์เพื่อวัดระดับความสูงใต้ท้องรถพอประมาณและทำการคำนวณ เพื่อหาระดับความสูงที่กั้นใต้ท้องรถจะถูกยกขึ้นให้พอเหมาะกับความสูงของใต้ท้องรถแต่ละคัน

ในกรณีถ้ามีรถคันสุดท้ายที่จะทำให้ที่จอดรถในอาคารจอดรถเต็มขับเข้ามา แต่ไฟสีแดงหน้าอาคารจอดรถจะติดเมื่อรถคันสุดท้ายเข้าไปจอด ณ จุดจอดรถ และได้ทำการดึงบัตรแถบแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล็กที่จุดจอดรถออกแล้ว ซึ่งในกรณีนี้จะทำให้เกิดปัญหาคือ จะทำให้รถคันที่ขับตามรถคันสุดท้ายเข้ามานั้นไม่รู้ว่าที่จอดรถเต็มในอาคารจอดรถนั้นเต็มแล้ว ทั้งนี้เนื่องจากโครงงานนี้ได้ใช้พอร์ตของไมโครคอนโทรลเลอร์เบอร์ AT89S8252 หมดทุกพอร์ตแล้ว จึงต้องมีการพัฒนาสำหรับการตรวจสอบจำนวนของรถที่เข้าในอาคารจอดรถ โดยติดตั้งเซนเซอร์ที่ทางเข้าและทางออกของอาคารจอดรถ แล้วใช้สัญญาณจากเซนเซอร์ไปคำนวณจำนวนรถที่เข้าและออกจากอาคารจอดรถโดยไมโครคอนโทรลเลอร์ แล้วแสดงผลเป็นไฟสีแดงหน้าอาคารจอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับความกรุณาจาก รองศาสตราจารย์ ดร. จงกล งามวิวิทย์ อาจารย์ที่ปรึกษา ที่ได้ให้คำแนะนำแก่ผู้จัดทำตลอดมา และคณาจารย์ทุกท่านที่ประสิทธิประสาทวิชาความรู้ให้แก่ผู้จัดทำ ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง ณ โอกาสนี้ด้วย

ขอกราบขอบพระคุณ คุณพ่อคุณแม่ของผู้จัดทำที่คอยให้กำลังใจแก่ผู้จัดทำตลอดมา ขอขอบคุณแกง เพื่อนที่ให้คำปรึกษาด้านโปรแกรม พี่วุฒิช่วยแนะนำการติดตั้งอุปกรณ์ น้องป้อมที่ช่วยตรวจสอบคำสะกด และเพื่อน ๆ ทุกคนที่ให้กำลังใจแก่ผู้จัดทำมาโดยตลอด

ผู้จัดทำ

นาย พิศาล ภูวนกุลชัย

นาย พีระพงษ์ ทีปประพันธ์ณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. บริษัท อีทีที จำกัด "คู่มือการใช้งานบอร์ดควบคุมสเตปปิงมอเตอร์ อีที เอสเอ็มซีซี เวอร์ชั่น 2" หน้า 3-14
2. บริษัท อีทีที จำกัด "คู่มือการใช้งานบอร์ดควบคุม ซีพี-เอส 8252 เวอร์ชั่น 1" หน้า 22-23
3. บริษัท อีทีที จำกัด "เครื่องอ่านบัตรแถบแม่เหล็ก พานาโซนิค แซดยู-เอ็ม2121เอส 451" หน้า 1-4
4. วรพจน์ กรแก้ววัฒนกุล "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ เอ็มซีเอส 51" บริษัทอินโนเวตีฟ เอ็กเพอริเมนต์จำกัด หน้า 236 – 255
5. บริษัท ฟิวเจอร์คิท มาร์เก็ตติ้ง จำกัด "รีโมทอินฟาเรด 25 ฟุต (รับและส่ง)" 2 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

```
;*************************************************
*            WRITE RTC DS1307                   *
*            SXA51  ASEMBLER                    *
;*************************************************
ORG  0H
INFRA           EQU   P3.6
PORTA           EQU   P3.3
PORTB           EQU   P3.2
PORTC           EQU   P3.1
PORTD           EQU   P3.0
EN1             EQU   P3.5
EN2             EQU   P3.4
COIN            EQU   P1.1
CS_LCD          EQU   P1.3      ; E  LCD (PIN INT1)
RS_LCD          EQU   P1.2      ; RS LCD (Pin T0)
DATA_MAG        EQU   P2.0          ; Data From Magnetic Card1
PRESENT         EQU   P2.1          ; Enable Magnetic Card1
CLK_MAG         EQU   P2.2          ; Clock Sync Magnetic Card1
CARD_INS        EQU   P2.3          ; Card Input  Detect Active "0"1
DATA_MAG1       EQU   P1.0          ; Data From Magnetic Card2
PRESENT1        EQU   P3.7          ; Enable Magnetic Card2
CLK_MAG1        EQU   P2.4          ; Clock Sync Magnetic Card2
CARD_INS1       EQU   P2.7          ; Card Input  Detect Active "0"2
DATE            EQU   20H
MONTH           EQU   22H
YEAR            EQU   24H
SCL             EQU   P2.5
SDA             EQU   P2.6
CONT_BYTE_W     EQU   0D0H
CONT_BYTE_R     EQU   0D1H
init55:LJMP   RESET_VEC          ;reset vector
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ADD_LOW:        DS   1
DATA:           DS   1
SEC:            DS   1
MIN:            DS   1
HOUR:           DS   1
DAY:            DS   1
SEC1:           DS   1
MIN1:           DS   1
HOUR1:          DS   1
SEC3:           DS   1
MIN3:           DS   1
HOUR3:          DS   1
SEC2:           DS   1
MIN2:           DS   1
HOUR2:          DS   1
INT_1:          DS   1
INT_2:          DS   1

FL1:            DW   1

FL2:            DW   1
;
org 20h
FLAG_BUF:  DS   1          ; Flag Buffer
CHECKSUM:  DS   1
BUFFER:    DS   40         ; 25 Byte Buffer
;
BUFFER1:   DS   40         ; 25 Byte Buffer
;
START      EQU  FLAG_BUF.4     ; Start Flag Status
STOP       EQU  FLAG_BUF.5     ; Stop Flag Status
ERROR      EQU  FLAG_BUF.2     ; Error Flag Status
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PARITY     EQU    FLAG_BUF.3        ; Parity Error Flag Status
IN1        EQU    FLAG_BUF.0        ; Parity Error Flag Status
IN2        EQU    FLAG_BUF.1        ; Parity Error Flag Status


RESET_VEC:          MOV    SP,#256-32
                    CLR    EA
                                ;*********************************
                                ;*      Init LCD                 *
                                ;*********************************
                    SETB   CS_LCD
                    LCALL  DELAY
                    LCALL  INIT_LCD
                    MOV    A,#0
                    LCALL  GOTO_LCD
                                ;*********************************
                                ;*      Init Infrared            *
                                ;*********************************
;call delay11
;jmp INIT_SER

HAHA: JNB  INFRA,INF0
    CLR P1.4
     call delay12;
     JMP HAHA
INF1:;       JNB INFRA,INF0
     ;       JMP INF1
INF0:        JB INFRA,pass
CLR P1.4
call delay12
          JMP INF0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ;**********
                    ;jmp INIT_SER
                    ;********
pass:

MOV R6,#80h
A1:  CLR     PORTA
     SETB    PORTB
     CLR     PORTC
     SETB    PORTD
     SETB    EN1
     SETB    EN2
     CALL DELAY11
A2:  CLR     PORTA
     CLR     PORTB
     CLR     PORTC
     SETB    PORTD
     CLR     EN1
     SETB    EN2
     CALL DELAY11
A3:  SETB    PORTA
     CLR     PORTB
     CLR     PORTC
     SETB    PORTD
     SETB    EN1
     SETB    EN2
     CALL DELAY11
A4:  SETB    PORTA
     CLR     PORTB
     CLR     PORTC
     CLR     PORTD
     SETB    EN1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CLR     EN2
    CALL DELAY11
A5: SETB    PORTA
    CLR     PORTB
    SETB    PORTC
    CLR     PORTD
    SETB    EN1
    SETB    EN2
    CALL DELAY11
A6: CLR     PORTA
    CLR     PORTB
    SETB    PORTC
    CLR     PORTD
    CLR     EN1
    SETB    EN2
    CALL DELAY11
A7:  CLR    PORTA
    SETB    PORTB
    SETB    PORTC
    CLR     PORTD
    SETB    EN1
    SETB    EN2
    CALL DELAY11
A8: CLR     PORTA
    SETB    PORTB
    CLR     PORTC
    CLR     PORTD
    SETB    EN1
    CLR     EN2
    CALL DELAY11
   DJNZ  R6,A1
  JMP END1
  DELAY11:  MOV R4,#25H
  DLY1:   MOV R5,#200H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      DJNZ  R5,$
      DJNZ   R4,DLY1
      RET
DELAY12:  MOV R4,#550H
DLY11:   MOV R5,#5000H
      DJNZ  R5,$
      DJNZ   R4,DLY11
      RET
END1:;          CLR        PORTA
 ;  clr        PORTB
 ; CLR         PORTC
 ;   CLR       PORTD
;SETB EN1
;SETB EN2
;call delay11
;              CLR        PORTA
;   clr        PORTB
;  CLR         PORTC
;    CLR       PORTD
;CLR EN1
;SETB EN2
;call delay11
;              CLR        PORTA
;   clr        PORTB
;  CLR         PORTC
;    CLR       PORTD
;CLR EN1
;CLR EN2
;call delay11
;              CLR        PORTA
;   clr        PORTB
;  CLR         PORTC
;    CLR       PORTD
;SETB EN1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;CLR EN2
;call delay11
INIT_SER:
        MOV  A,0
        MOV  IE,A
        MOV   A,#0FBH          ; Set baud rate 9600
        MOV   TH1,A
        MOV   TL1,A
        MOV   TMOD,#00100000B   ; Timer1 Mode 2
        CLR   ES               ; Disable serial interupt
        CLR   ET1              ; Disable timer1 interupt
        SETB  TR1              ; Set timer1 control
        MOV   SCON,#01010000B   ; Serial mode 1
        SETB  DATA_MAG         ; Stanby Signale
        SETB  PRESENT
        SETB  CLK_MAG
        SETB  CARD_INS
        SETB  DATA_MAG1        ; Stanby Signale
        SETB  PRESENT1
        SETB  CLK_MAG1
        SETB  CARD_INS1
        LCALL CLR_BUFF

INIT_PORT:      SETB  SCL
        SETB  SDA
        MOV   A,#0CH
        MOV   DPTR,#INIT_TAB
        MOV   A,#00H
INIT0:    PUSH  ACC
        MOVC  A,@A+DPTR
        CJNE  A,#00H,INIT1
        LJMP  INIT2
INIT1:    POP   ACC
        INC   A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SJMP    INIT0

INIT2:              MOV    A,#0
                    MOV    FL1,A
                    MOV    FL2,A


                    ; MOV     ADD_LOW,#00H
                   ; MOV     DATA,#00H
                   ; LCALL   WRITE_BYTE
                   ; MOV     ADD_LOW,#01H
                   ; MOV     DATA,#20H
                   ; LCALL   WRITE_BYTE
                   ; MOV     ADD_LOW,#02H
                   ; MOV     DATA,#20H
                   ; LCALL   WRITE_BYTE


INIT3:              MOV     A,FL2
                    ANL     A,FL1
                    JZ      NOLIGHT
LIGHT:              Setb    P1.4
                    JMP     WAIT_HAHA

NOLIGHT:            clr     P1.4

WAIT_HAHA:          SETB    CARD_INS1       ;CHECK SENSOR CARD INPUT
                    JNB     CARD_INS1,GET_2
                    SETB    CARD_INS        ;CHECK SENSOR CARD INPUT
                    JNB     CARD_INS,GET_1
                    jmp     INIT3
GET_1:              LCALL   GET_DATA
                    jmp     HIGH22
GET_2:              LCALL   GET_DATA1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HIGH22:          MOV     A,#0
                 LCALL   GOTO_HLCD
                 MOV     R0,#BUFFER+1
                 MOV     A,@R0
                 ADD     A,#'0'
                 CJNE    A,#'1', UNDERtemp
OVER:            MOV     A,FL1
                 JZ      NOTSET1
     ;if card 1 is 0out
SET1:            MOV     ADD_LOW,#02H
                 LCALL   READ_BYTE
                 MOV     A,DATA
                 MOV     HOUR3,A
                 MOV     ADD_LOW,#01H
                 LCALL   READ_BYTE
                 MOV     A,DATA    ;;;;;;;;
                 MOV     MIN3,A
                 MOV     ADD_LOW,#00H
                 LCALL   READ_BYTE
                 MOV     A,DATA
                 MOV     SEC3,A
                 LCALL   COUNT_MON21
                 MOV     A,#00
                 MOV     FL1,A

                 JMP     INIT3
undertemp:       jmp under;
  ;if card 1 is  net set yet
NOTSET1:         MOV     A,#01
                 MOV     FL1,A
                 MOV     ADD_LOW,#02H
                 LCALL   READ_BYTE
                 MOV     A,DATA
                 MOV     HOUR1,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    ADD_LOW,#01H
        LCALL  READ_BYTE
        MOV    A,DATA
        MOV    MIN1,A
        MOV    ADD_LOW,#00H
        LCALL  READ_BYTE
        MOV    A,DATA
        MOV    SEC1,A
        JMP    INIT3

UNDER:
        MOV    A,FL2
        JZ     NOTSET2

SET2:     MOV    ADD_LOW,#02H
        LCALL  READ_BYTE
        MOV    A,DATA
        MOV    HOUR3,A
        MOV    ADD_LOW,#01H
        LCALL  READ_BYTE
        MOV    A,DATA
        MOV    MIN3,A
        MOV    ADD_LOW,#00H
        LCALL  READ_BYTE
        MOV    A,DATA
        MOV    SEC3,A
        LCALL  COUNT_MON2
        LCALL INIT_LCD
        MOV    A,#00H
        MOV    FL2,A
        JMP    HAHA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NOTSET2:        MOV    A,#01H
                MOV    FL2,A
                MOV    ADD_LOW,#02H
                LCALL  READ_BYTE
                MOV    A,DATA
                MOV    HOUR2,A
                MOV    ADD_LOW,#01H
                LCALL  READ_BYTE
                MOV    A,DATA
                MOV    MIN2,A
                MOV    ADD_LOW,#00H
                LCALL  READ_BYTE
                MOV    A,DATA
                MOV    SEC2,A
                JMP    INIT3


                MOV    A,#' '
                LCALL  TX_BYTE
                MOV    ADD_LOW,#02H
                LCALL  READ_BYTE
                LCALL  SHOWASCTOLCD
                MOV    A,20
                LCALL  TX_BYTE
                MOV    A,#' '
                LCALL  TX_BYTE
                MOV    A,#':'
                LCALL  WR_LCD
                LCALL  TX_BYTE
                MOV    A,#' '
                LCALL  TX_BYTE
                MOV    A,#' '
                LCALL  TX_BYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    ADD_LOW,#01H
        LCALL  READ_BYTE
        LCALL  SHOWASCTOLCD

        MOV    A,#':'
        LCALL  WR_LCD
        MOV    ADD_LOW,#00H

        LCALL  READ_BYTE
        LCALL  SHOWASCTOLCD
        MOV    A,#' '
        LCALL  TX_BYTE
        MOV    A,#' '
        LCALL  TX_BYTE
        MOV    A,#':'
        LCALL  TX_BYTE
        MOV    A,#' '
        LCALL  TX_BYTE
        MOV    A,#' '
        LCALL  TX_BYTE
        MOV    ADD_LOW,#00H

        LCALL  READ_BYTE
        LCALL  SHOWASC

        MOV    A,#0DH
        LCALL  TX_BYTE
        LJMP   INIT3

SHOWASC:        PUSH   ACC
        SWAP   A
        ANL    A,#0FH
        LCALL  HEXASC
        LCALL  TX_BYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP    ACC
        ANL    A,#0FH
        LCALL  HEXASC
        LCALL  TX_BYTE
        RET
SHOWASCTOLCD:    PUSH   ACC
        SWAP   A
        ANL    A,#0FH
        LCALL  HEXASC
        LCALL  WR_LCD
        POP    ACC
        ANL    A,#0FH
        LCALL  HEXASC
        LCALL  WR_LCD
        RET

        ;*************************************************
        ;*      WRITE DATA TO EEPROM 1 BYTE              *
        ;*INPUT : ADD_HIGH                               *
        ;*     : ADD_LOW                                  *
        ;*     : DATA                                     *
        ;*************************************************
WRITE_BYTE:      CLR    SDA          ;start bit
        CLR    SCL
        MOV    A,#CONT_BYTE_W       ;send control byte
        LCALL  LOOP_BYTE
        SETB   SDA
        SETB   SCL
        JB     SDA,WRITE_BYTE       ;loop until busy
        CLR    SCL
        MOV    A,ADD_LOW            ;send address low
        LCALL  LOOP_BYTE
        SETB   SDA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB   SCL
        JB     SDA,WRITE_BYTE       ;loop until busy
        CLR    SCL
        MOV    A,DATA               ;send data
        LCALL  LOOP_BYTE
        SETB   SDA
        SETB   SCL
        JB     SDA,WRITE_BYTE       ;loop until busy
        CLR    SDA
        CLR    SCL
        SETB   SCL                  ;stop bit
        SETB   SDA
        RET

        ;*****************************************************
        ;*        READ DATA FROM EEPROM 1 BYTE               *
        ;* INPUT  : ADD_HIGH                                 *
        ;*        : ADD_LOW                                  *
        ;* OUTPUT : DATA                                     *
        ;*****************************************************

READ_BYTE:  CLR    SDA              ;start bit
        CLR    SCL
        MOV    A,#CONT_BYTE_W       ;send control byte
        LCALL  LOOP_BYTE
        SETB   SDA
        SETB   SCL
        JB     SDA,READ_BYTE        ;loop until busy
        CLR    SCL
        MOV    A,ADD_LOW            ;send address low
        LCALL  LOOP_BYTE
        SETB   SDA
        SETB   SCL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JB     SDA,READ_BYTE        ;loop until busy
        CLR    SCL

        SETB   SCL
        SETB   SDA
        CLR    SDA                  ;start bit
        CLR    SCL
        MOV    A,#CONT_BYTE_R       ;send control byte
        LCALL  LOOP_BYTE
        SETB   SDA
        SETB   SCL
        JB     SDA,READ_BYTE        ;loop until busy
        CLR    SCL
        LCALL  LOOP_READ
        SETB   SDA
        SETB   SCL
        CLR    SCL

        SETB   SCL                  ;stop bit
        SETB   SDA
        RET

                ;*************************************************
                ;*              WRITE                            *
                ;* INPUT: ACC                                    *
                ;*************************************************
LOOP_BYTE:      PUSH   02H
        MOV    R2,#08H
LOOP_SEND:      RLC    A
        MOV    SDA,C
        SETB   SCL
        CLR    SCL
        DJNZ   R2,LOOP_SEND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP    02H
        RET

                ;*****************************************************
                ;*                  READ                             *
                ;* OUTPUT: ACC                                       *
                ;*****************************************************
LOOP_READ:      PUSH   02H
        MOV    R2,#08H
LOOP_READ1:     SETB   SCL
        MOV    C,SDA
        CLR    SCL
        RLC    A
        DJNZ   R2,LOOP_READ1
        MOV    DATA,A
        POP    02H
        RET

DELAY4M:        PUSH   DPH
        PUSH   DPL
        MOV    DPTR,#0000H
DEL4:           INC    DPTR
        MOV    A,DPL
        ORL    A,DPH
        JNZ    DEL4
        POP    DPL
        POP    DPH
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ;*****************************************************
                ;*          Send 1-Byte to SERIAL                    *
                ;*          Input  : ACC                             *
                ;*          Output : Serial port                     *
                ;*****************************************************
TX_BYTE:        PUSH  IE
                CLR   TI
                MOV   SBUF,A
                JNB   TI,$
                CLR   TI
                POP   IE
                RET

                ;*****************************************************
                ;*          Receive Data From SERIAL                 *
                ;*          Input  : Serial Port                     *
                ;*          Output : ACC                             *
                ;*****************************************************
RX_BYTE:        PUSH  IE
                JNB   RI,$          ; Wait data
                CLR   RI
                MOV   A,SBUF
                LCALL TX_BYTE
                POP   IE
                RET


HEXASC:         PUSH  DPH
                PUSH  DPL
                MOV   DPTR,#HEXASC_TAB
                MOVC  A,@A+DPTR
                POP   DPL
                POP   DPH
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HEXASC_TAB:         DB    30H,31H,32H,33H,34H,35H,36H,37H,38H,39H
                    DB    41H,42H,43H,44H,45H,46H


INIT_TAB:           DB    'CP-S8252 & CP-S53',0AH,0DH
                    DB    'TEST DS1307',0AH,0DH
                    DB    'HOUSE   MIN    SEC',0AH,0DH,00H

                    ;*************************************************
                    ;*        Program Test 4 Bits LCD                *
                    ;*        Dot Matrix TM1620 16X2                 *
                    ;*************************************************
                    ;*          Input  : ACC  (ASCII)                *
                    ;*          Output : Data bus LCD                *
                    ;*************************************************
WR_LCD:     MOV     B,A
            ANL     A,#0F0H
            ORL     A,#00001100B    ;RS/CS HI
            MOV     R2,A
            MOV     A,P1
            ANL     A,#00000011B
            ORL     A,R2
            MOV     P1,A
            LCALL   EN_LCD
            MOV     A,B             ;Low byte
            SWAP    A
            ANL     A,#0F0H
            ORL     A,#00001100B
            MOV     R2,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    A,P1
        ANL    A,#00000011B
        ORL    A,R2
        MOV    P1,A
        LCALL  EN_LCD
        RET

                    ;*****************************************************
                      ;*          Write Instruction LCD          *
                      ;*          Input  : ACC (Command)         *
                      ;*          Output : Data bus LCD          *
                    ;*****************************************************
WR_INS:   MOV    B,A
        ANL    A,#0F0H
        SETB   ACC.3          ;CS HI
        MOV    R2,A
        MOV    A,P1
        ANL    A,#00000011B   ;else bit
        ORL    A,R2
        MOV    P1,A           ; High byte
        LCALL  EN_LCD
        MOV    A,B            ; Low byte
        SWAP   A
        ANL    A,#0F0H
        SETB   ACC.3
        MOV    R2,A
        MOV    A,P1
        ANL    A,#00000011B
        ORL    A,R2
        MOV    P1,A
        LCALL  EN_LCD
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************
;*              Goto position of LCD             *
;*              Input : ACC (addr.)              *
;*************************************************
GOTO_LCD:  PUSH B
           SETB   ACC.7
           LCALL  WR_INS
           POP B
           RET

;*************************************************
;*           Goto high position of LCD           *
;*              Input : ACC (addr.)              *
;*************************************************
GOTO_HLCD: PUSH B
           SETB   ACC.7
           SETB   ACC.6
           LCALL  WR_INS
           POP B
           RET

;*************************************************
;*                 Initial LCD                   *
;*                 4-Bit Interface               *
;*************************************************
INIT_LCD:  CLR    RS_LCD
           MOV    A,#33H       ; Set DL = 1 3-time
           LCALL  WR_INS
           MOV    A,#32H       ; Clear DL = 0 1-time
           LCALL  WR_INS
           MOV    A,#28H       ; Function set
           LCALL  WR_INS       ; DL=0 4Bit,N=1 2Line,F=0 5X7
           MOV    A,#0CH
           LCALL  WR_INS       ; Entry display,cursor off,cursor not blink
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ;*************************************************
                     ;*                COUNT MONEY                *
                    ;*************************************************
COUNT_MON1:  MOV    A,HOUR1
        LCALL  BCD2HEX
        MOV    B,#60D
        MUL    AB
        MOV    B,MIN1
        PUSH   ACC
        MOV    A,B
        LCALL  BCD2HEX
        MOV    B,A
        POP    ACC
        ADD    A,B
        MOV    INT_1,A
        MOV    A,HOUR3
        LCALL  BCD2HEX
        MOV    B,#60D
        MUL    AB
        MOV    B,MIN3
        PUSH   ACC
        MOV    A,B
        LCALL  BCD2HEX
        MOV    B,A
        POP    ACC
        ADD    A,B
        MOV    INT_2,A
        MOV    B,INT_1
        SUBB   A,B
        MOV    B,#60
        DIV    AB
        ADD    A,#'0'
        LCALL    WR_LCD
        MOV    A,#' '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL    WR_LCD
        MOV     A,#'H'
        LCALL    WR_LCD
        MOV     A,#'O'
        LCALL    WR_LCD
        MOV     A,#'U'
        LCALL    WR_LCD
        MOV     A,#'R'
        LCALL    WR_LCD
        MOV     A,#'S'
        LCALL    WR_LCD
        MOV     A,#' '
        LCALL    WR_LCD
        MOV     A,#' '
        LCALL    WR_LCD
        MOV     A,#' '
        LCALL    WR_LCD

        RET
COUNT_MON21: MOV A,#00H
        LCALL GOTO_LCD
        MOV     A,#'C'
        LCALL    WR_LCD
        MOV     A,#'A'
        LCALL    WR_LCD
        MOV     A,#'R'
        LCALL    WR_LCD
        MOV     A,#'D'
        LCALL    WR_LCD
        MOV     A,#' '
        LCALL    WR_LCD
        MOV     A,#'N'
        LCALL    WR_LCD
        MOV     A,#'O'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL    WR_LCD
        MOV      A,#'.'
        LCALL    WR_LCD
        MOV      A,#'1'
        LCALL    WR_LCD
        LCALL    VERY_DELAY
        MOV    A,HOUR3
      ; LCALL  BCD2HEX
        MOV    B,HOUR1
      ; PUSH   ACC
      ; MOV    A,B
      ;; LCALL  BCD2HEX
      ; MOV    B,A
      ; POP    ACC
        SUBB    A,B
        PUSH    ACC
        MOV     A,MIN3
      ;  LCALL  BCD2HEX
        MOV     B,MIN1
      ; PUSH   ACC
      ; MOV    A,B
      ; LCALL  BCD2HEX
      ; MOV    B,A
      ; POP    ACC
        SUBB   A,B
        POP    ACC
        JC     ATHANB1
        INC    ACC
ATHANB1:    PUSH ACC
        PUSH ACC
        PUSH ACC
        MOV A,#00H
        LCALL GOTO_HLCD
        POP ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP ACC
        RET



LCD_TAB:    DB   ' TIME :',00H
                                    ;*********************
                                    ;* Clear Data Buffer *
                                    ;* Register : R0,ACC *
                                    ;*********************
;
CLR_BUFF:   PUSH   B
        MOV    B,#40
        MOV    R0,#BUFFER        ; Clear Data Buffer
CLR_BUF1:   CLR    A
        MOV    @R0,A
        INC    R0
        DJNZ   B,CLR_BUF1
        POP    B
        RET

                                ;*********************************
                                ;* Read Data From Magnetic Card *
                                ;* Get Number of Card (Modulo5) *        0
                                ;* Output : BUFFER (BCD 40Byte) *
                                ;* Reg.   : R1,R2,ACC,CHECKSUM  *
                                ;*********************************
;
GET_DATA1:  SETB   DATA_MAG1         ; Standy Signale
        SETB   PRESENT1
        SETB   CLK_MAG1
        LCALL  CLR_BUFF          ; Clear Buffer
        MOV    CHECKSUM,#0
        MOV    R1,#BUFFER-1      ; Pointer to Save Data
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR    START           ; Clear Any Flag
        CLR    STOP
        CLR    ERROR
        ;
WAIT_CARD1: SETB  CARD_INS1     ;CHEEK SENSOR CARD INPUT
         JB     CARD_INS1,WAIT_CARD1
WAIT_STR1: SETB  PRESENT1     ;
      JB     PRESENT1,WAIT_STR1    ;WAIT START READY
GET_DAT01:  LCALL  GET_CLK1          ; Get Start Sentinel
        MOV    C,DATA_MAG1
        CPL    C
        JB     DATA_MAG1,GET_DAT01  ; Loop Until Start Bit
        SETB   START           ; Strat Bit OK
        MOV    R3,#25     ;MODULO 5 FORMAT COUNTER
        ;
GET_DAT11:  MOV    R2,#5           ; Modulo 5 Format Counter
        CLR    A
        CLR    PARITY
        INC    R1              ; Point to Next Byte Save
GET_DAT21:  JB     START,GET_DAT31     ; Get 1-Byte Data
        LCALL  GET_CLK1
        MOV    C,DATA_MAG1
        CPL    C
GET_DAT31:  RRC    A
        CLR    START
        DJNZ   R2,GET_DAT21
        RR     A
        RR     A
        RR     A
        JB     P,GET_DAT41        ; Parity Odd OK
        SETB   PARITY           ; Parity Error
        SJMP   GET_DAT91
        ;
GET_DAT41:  ANL    A,#0FH           ; Ignore Parity
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JB     STOP,GET_DAT51
        PUSH   ACC
        XRL    A,CHECKSUM        ; Checksum Data
        MOV    CHECKSUM,A
        POP    ACC
GET_DAT51:  MOV    @R1,A          ; Save Data
        JB     STOP,GET_DAT71    ; Stop Operation
        CJNE   A,#0FH,GET_DAT61
        SETB   STOP
        SJMP   GET_DAT11
GET_DAT61:  DJNZ   R3,GET_DAT11
            INC    R1
            MOV    @R1,#0FH      ;GENERATE STOP CODE
        SJMP   GET_DAT91         ; MAXIMUM LIMIT READ
GET_DAT71:  CJNE   A,CHECKSUM,GET_DAT81
        CLR    ERROR             ; Checksum OK
        SJMP   GET_DAT91
GET_DAT81:  SETB   ERROR         ; Checksum Error
GET_DAT91:     SETB   CARD_INS1
           JNB    CARD_INS1,GET_DAT91   :WAIT CARD EXIT
              RET


GET_DATA:  SETB   DATA_MAG       ; Standy Signale
        SETB   PRESENT
        SETB   CLK_MAG
        LCALL  CLR_BUFF          ; Clear Buffer
        MOV    CHECKSUM,#0
        MOV    R1,#BUFFER-1      ; Pointer to Save Data
        CLR    START             ; Clear Any Flag
        CLR    STOP
        CLR    ERROR
        ;
WAIT_CARD: SETB  CARD_INS      ;CHEEK SENSOR CARD INPUT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JB      CARD_INS,WAIT_CARD
WAIT_STR: SETB  PRESENT     ;
        JB      PRESENT,WAIT_STR    ;WAIT START READY
GET_DAT0:  LCALL  GET_CLK          ; Get Start Sentinel
        MOV     C,DATA_MAG
        CPL     C
        JB      DATA_MAG,GET_DAT0   ; Loop Until Start Bit
        SETB    START              ; Strat Bit OK
        MOV     R3,#25      ;MODULO 5 FORMAT COUNTER
        ;
GET_DAT1:  MOV     R2,#5            ; Modulo 5 Format Counter
        CLR     A
        CLR     PARITY
        INC     R1                 ; Point to Next Byte Save
GET_DAT2:  JB      START,GET_DAT3     ; Get 1-Byte Data
        LCALL   GET_CLK
        MOV     C,DATA_MAG
        CPL     C
GET_DAT3:  RRC     A
        CLR     START
        DJNZ    R2,GET_DAT2
        RR      A
        RR      A
        RR      A
        JB      P,GET_DAT4         ; Parity Odd OK
        SETB    PARITY             ; Parity Error
        SJMP    GET_DAT9
        ;
GET_DAT4:  ANL     A,#0FH           ; Ignore Parity
        JB      STOP,GET_DAT5
        PUSH    ACC
        XRL     A,CHECKSUM         ; Checksum Data
        MOV     CHECKSUM,A
        POP     ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GET_DAT5:  MOV    @R1,A            ; Save Data
           JB     STOP,GET_DAT7    ; Stop Operation
           CJNE   A,#0FH,GET_DAT6
           SETB   STOP
           SJMP   GET_DAT1
GET_DAT6:  DJNZ   R3,GET_DAT1
           INC    R1
           MOV    @R1,#0FH         ;GENERATE STOP CODE
           SJMP   GET_DAT9         ; MAXIMUM LIMIT READ
GET_DAT7:  CJNE   A,CHECKSUM,GET_DAT8
           CLR    ERROR            ; Checksum OK
           SJMP   GET_DAT9
GET_DAT8:  SETB   ERROR            ; Checksum Error
GET_DAT9:  SETB   CARD_INS
           JNB    CARD_INS,GET_DAT9    :WAIT CARD EXIT
           RET

                                   ;**********************
                                   ;* Get Magnetic Clock *
                                   ;**********************
;
GET_CLK:   JB     CLK_MAG,$        ; Wait Falling Clock
           NOP
           NOP
           JNB    CLK_MAG,$        ; Wait Rising Clock
           NOP
           NOP
           RET
GET_CLK1:  JB     CLK_MAG1,$       ; Wait Falling Clock
           NOP
           NOP
           JNB    CLK_MAG1,$       ; Wait Rising Clock
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        NOP
        NOP
        RET
                                ;**********************
                                ;*    bcdTOhex     *
                                ;**********************
BCD2HEX:    PUSH B
        MOV B,r0
        PUSH B
        MOV R0,A
        SWAP A
        ANL A,0FH
        MOV B,#10D
        MUL AB
        MOV B,A
        MOV A,R0
        ANL A,0FH
        ADD A,B
        POP B
        mov r0,B
        POP B
        RET

        END
```

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.

8-bit Microcontroller with 8K Bytes Flash

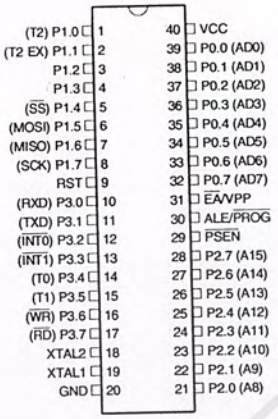
AT89S8252

Rev. 0401E–02/00

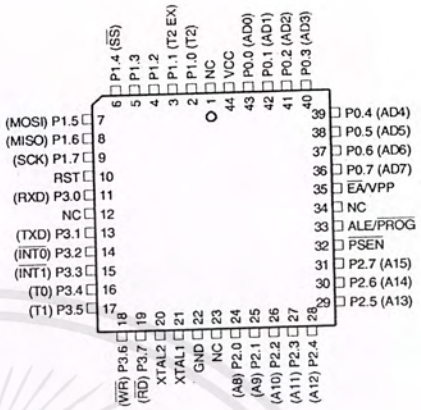
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

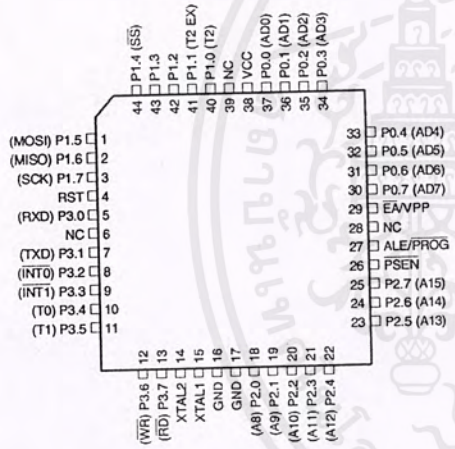
PDIP



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.

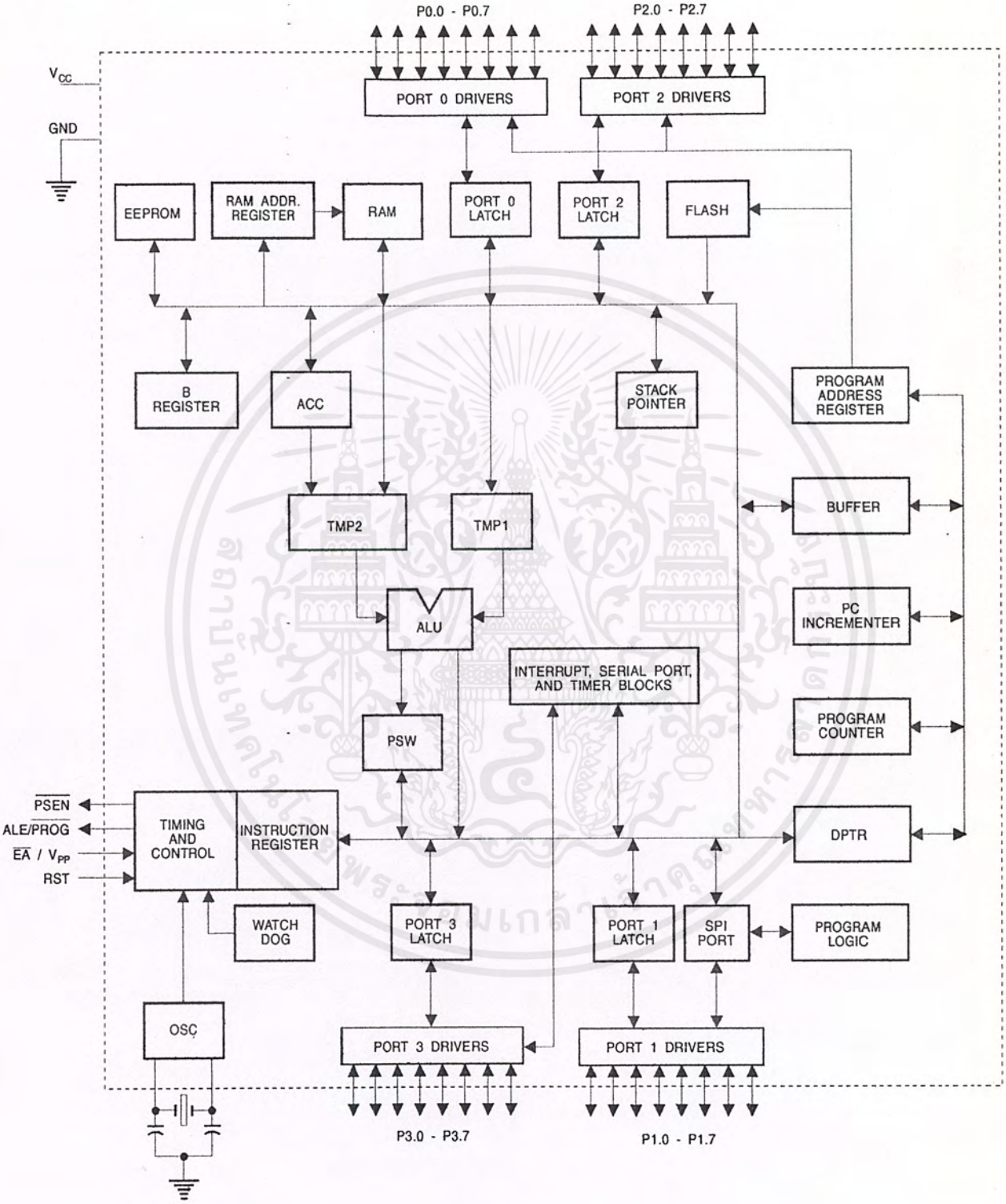
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

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

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

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2–wire serial interface
- Programmable squarewave output signal
- Automatic power fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode at 25°C
- Optional Industrial temperature range –40°C to +85°C (IND)
- Available in 8–pin DIP or SOIC

ORDERING INFORMATION

DS1307	Serial Timekeeping Chip; 8–pin DIP
DS1307Z	Serial Timekeeping Chip; 8–pin SOIC (150 mil)
DS1307N	8–pin DIP (IND)
DS1307ZN	8–pin SOIC (IND)

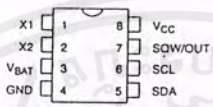
DESCRIPTION

The DS1307 Serial Real Time Clock is a low power full BCD clock calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24–hour or 12–hour format with AM/PM indicator. The DS1307 has a built–in power sense circuit which detects power failures and automatically switches to the battery supply.

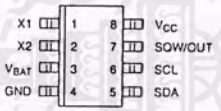
OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT} the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT}+0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock. The following paragraphs describe the function of each pin.

PIN ASSIGNMENT



DS1307 8–PIN DIP (300 MIL)



DS1307Z 8–PIN SOIC (150 MIL)

PIN DESCRIPTION

V_{CC}	–	Primary Power Supply
X1, X2	–	32.768 KHz Crystal Connection
V_{BAT}	–	+3 Volt Battery Input
GND	–	Ground
SDA	–	Serial Data
SCL	–	Serial Clock
SQW/OUT	–	Square wave/Output Driver

©Copyright 1995 by Dallas Semiconductor Corporation.
All Rights Reserved. For important information regarding patents and other intellectual property rights, please refer to Dallas Semiconductor data books.

041697 1/11

MICROCHIP 24AA256/24LC256/24FC256

256K Bit I²C™ CMOS Serial EEPROM

DEVICE SELECTION TABLE

Part Number	Vcc Range	Max Clock Frequency	Temp Ranges
24AA256	1.8-5.5V	400 kHz†	I
24LC256	2.5-5.5V	400 kHz‡	I, E
24FC256	2.5-5.5V	1 MHz	I

†100 kHz for Vcc < 2.5V.
‡100 kHz for E temperature range.

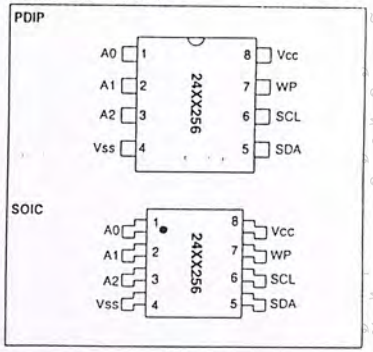
FEATURES

- Low power CMOS technology
  - Maximum write current 3 mA at 5.5V
  - Maximum read current 400 µA at 5.5V
  - Standby current 100 nA typical at 5.5V
- 2-wire serial interface bus, I²C compatible
- Cascadable for up to eight devices
- Self-timed ERASE/WRITE cycle
- 64-byte page-write mode available
- 5 ms max write-cycle time
- Hardware write protect for entire array
- Output slope control to eliminate ground bounce
- Schmitt trigger inputs for noise suppression
- 100,000 erase/write cycles guaranteed
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP and SOIC (208 mil) packages
- Temperature ranges:
  - Industrial (I): -40°C to +85°C
  - Automotive (E): -40°C to +125°C

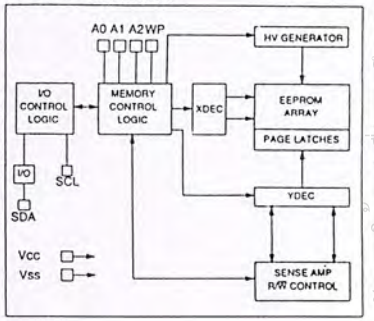
DESCRIPTION

The Microchip Technology Inc. 24AA256/24LC256/24FC256 (24XX256*) is a 32K x 8 (256K bit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low power applications such as personal communications or data acquisition. This device also has a page-write capability of up to 64 bytes of data. This device is capable of both random and sequential reads up to the 256K boundary. Functional address lines allow up to eight devices on the same bus, for up to 2M bit address space. This device is available in the standard 8-pin plastic DIP and 8-pin SOIC (208 mil) packages.

PACKAGE TYPE



BLOCK DIAGRAM



I²C is a trademark of Philips Corporation.
*24XX256 is used in this document as a generic part number for the 24AA256/24LC256/24FC256 devices.

© 1998 Microchip Technology Inc.

CP-S8252 Control Board with In-System ET-SPI Down Load

FEATURES of

LIQUID CRYSTAL DISPLAY MODULE

MODEL NO. : MTC-S16100XRGHS

Character Type Dot Matrix LCD Module
Easy interface of 4-bit or 8-bit
Display character pattern : 5 × 7 font with cursor(208 kinds)
The special character pattern can be programmable by Character Generator RAM directly.
A customer character pattern can be programmable by mask option.
Automatically power on reset.
Internal Memory:
- Character Generator ROM (CGROM) : 10,080 bits (204 characters × 5 × 8 dot)
- Character Generator RAM (CGRAM) : 64 × 8 bits (8 characters × 5 × 8 dot)

Low power operation:
- Power supply voltage range : 2.7 ~ 5.5 V (V_{DD})
- LCD drive voltage range : 3.0 ~ 13.0 V (V_{DD} - Vo)

Low power consumption
CMOS process
Duty-cycle : 1/16

View Direction	☑ 6 O'clock			☐ 12 O'clock	
LCD Type	☐ FSTN Positive			☐ FSTN Negative	
	☑ STN Gray	☐ STN Yellow Green		☐ STN Blue	
Rear Polarizer	☑ Reflective	☐ Transflective		☐ Transmissive	
Backlight Type	☐ LED	☐ Internal Power	☐ EL	☐ 5V input	
		☐ External Power	☐ CCFL	☐ 12V input	
Backlight Color	☐ White	☐ Amber	☐ Blue Green	☐ Yellow Green	☐ Other
Temperature Range	☐ Normal	☑ Wide		☐ Super Wide	
CCFL Inverter	☐ Build-in	☑ Not Build-in			
EL Driver IC	☐ Build-in	☑ Not Build-in			
Touch Screen	☐ With	☑ Without			

TO BE VERY CAREFUL !

The LCD driver ICs are made of CMOS process, which is very easy to be damaged by static charge, make sure the user is grounded when handling the LCM.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Commands

Instruction	Instruction Code										Description	Execution time(f_{osc} is 270kHz)
	RS	R/$\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.53mS
Return Home	0	0	0	0	0	0	0	0	1	*	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53mS
Entry Mode	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and make shift of entire display enable.	39 µS
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 µS
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	39 µS
Function Set	0	0	0	0	1	DL	N	F	*	*	Set interface data length (DL : 4-bit/8-bit), numbers of display line (N : 1-line/2-line), display font type(F : 5 X 8 dots/ 5 X 11 dots)	39 µS
Set CG RAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 µS
Set DD RAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 µS
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 µS
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 µS
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 µS

* means don't care

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMMANDS DESCRIPTION

Clear Display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing "20H" (space code) to all DDRAM address, and set DDRAM address to "00H" into AC (address counter). Return cursor to the original status. namely, bring the cursor to the left edge on first line of the display. Make entry mode increment (I/D = "1").

Return Home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

Return Home is cursor return home instruction. Set DDRAM address to "00H" into the address counter. Return cursor to its original site and return display to its original status, if shifted. Content of DDRAM is not changed.

Entry Mode Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

Set the moving direction of cursor and display.

I/D : Increment / decrement of DDRAM address (cursor or blink)

When I/D = "High", cursor/blink moves to right and DDRAM address is increased by 1.

When I/D = "Low", cursor/blink moves to left and DDRAM address is decreased by 1.

* CGRAM operates the same as DDRAM, when read from or write to CGRAM.

SH: Shift of entire display

When DDRAM read (CGRAM read/write) operation or SH = "Low", shift of entire display is not performed. If SH = "High" and DDRAM write operation, shift of entire display is performed according to I/D value (I/D = "1" , shift left, I/D = "0" : shift right).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Display ON/OFF Control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Control display/cursor/blink ON/OFF 1 bit register.

D : Display ON/OFF control bit

When D = "High", entire display is turned on.

When D = "Low", display is turned off, but display data is remained in DDRAM.

C : Cursor ON/OFF control bit

When C = "High", cursor is turned on.

When C = "Low", cursor is disappeared in current display, but I/D register remains its data.

B : Cursor Blink ON/OFF control bit

When B = "High", cursor blink is on, that performs alternate between all the high data and display character at the cursor position.

When B = "Low", blink is off.

Cursor or Display Shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

Without writing or reading of display data, shift right/left cursor position or display.

This instruction is used to correct or search display data. (Refer to Table 4)

During 2-line mode display, cursor moves to the 2nd line after 40th digit of 1st line.

Note that display shift is performed simultaneously in all the line.

When displayed data is shifted repeatedly, each line shifted individually.

When display shift is performed, the contents of address counter are not changed.

S/C	R/L	Operation
0	0	Shift cursor to the left, AC is decreased by 1.
0	1	Shift cursor to the right, AC is increased by 1.
1	0	Shift all of the display to the left, cursor moves according to the display.
1	1	Shift all of the display to the right, cursor moves according to the display.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
00	0	0	0	1	DL	N	F	*	*

DL : Interface data length control bit
When DL = "High", it means 8-bit bus mode with MPU.
When DL = "Low", it means 4-bit bus mode with MPU. So to speak, DL is a signal to select 8-bit or 4-bit bus mode. When 4-bit bus mode, it needs to transfer 4-bit data by two times.
N : Display line number control bit
When N = "Low", it means 1-line display mode.
When N = "High", 2-line display mode is set.
F : Display font type control bit
When F = "Low", it means 5 X 8 dots format display mode
When F = "High", 5 x11 dots format display mode.

Set CG RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.
This instruction makes CGRAM data available from MPU.

Set DD RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.
This instruction makes DDRAM data available from MPU.
When 1-line display mode (N = 0), DDRAM address is from "00H" to "4FH".
In 2-line display mode (N = 1), DDRAM address in the 1st line is from "00H" to "27H", and DDRAM address in the 2nd line is from "40H" to "67H".

Read Busy Flag and Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether KS0066U is in internal operation or not. If the resultant BF is High, it means the internal operation is in progress and you have to wait until BF to be Low, and then the next instruction can be performed. In this instruction you can read also the value of address counter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Write Data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.
The selection of RAM from DDRAM, CGRAM, is set by the previous address set instruction : DDRAM address set, CGRAM address set. RAM set instruction can also determine the AC direction to RAM. After write operation, the address is automatically increased/decreased by 1, according to the entry mode.

Read Data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.
The selection of RAM is set by the previous address set instruction. If address set instruction of RAM is not performed before this instruction, the data that read first is invalid, because the direction of AC is not determined. If you read RAM data several times without RAM address set instruction before read operation, you can get correct RAM data from the second, but the first data would be incorrect, because there is no time margin to transfer RAM data.
In case of DDRAM read operation, cursor shift instruction plays the same role as DDRAM address set instruction : it also transfers RAM data to output data register. After read operation address counter is automatically increased/decreased by 1 according to the entry mode. After CGRAM read operation, display shift may not be executed correctly.

NOTE: In case of RAM write operation, after this AC is increased/decreased by 1 like read operation. In this time, AC indicates the next address position, but you can read only the previous data by read instruction.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DD RAM ADDRESSING

For 16×1 or 8×1 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47

For 16×2 or 8×2 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

For 16×4 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

For 20×2 Display

Character	1	2	3	4	5	6	7	8	9	10	…	…	17	18	19	20
DD RAM Address	00	01	02	03	04	05	06	07	08	09	…	…	10	11	12	13
	40	41	42	43	44	45	46	47	48	49	…	…	50	51	52	53

For 20×4 Display

Character	1	2	3	4	5	6	7	8	9	10	…	…	17	18	19	20
	00	01	02	03	04	05	06	07	08	09	…	…	10	11	12	13
DD RAM Address	40	41	42	43	44	45	46	47	48	49	…	…	50	51	52	53
	14	15	16	17	18	19	1A	1B	1C	1D	…	…	24	25	26	27
	54	55	56	57	58	59	5A	5B	5C	5D	…	…	64	65	66	67

For 40×2 Display

Character	1	2	3	4	5	6	7	8	9	10	…	…	37	38	39	40
DD RAM Address	00	01	02	03	04	05	06	07	08	09	…	…	24	25	26	27
	40	41	42	43	44	45	46	47	48	49	…	…	64	65	66	67

For 40×4 Display

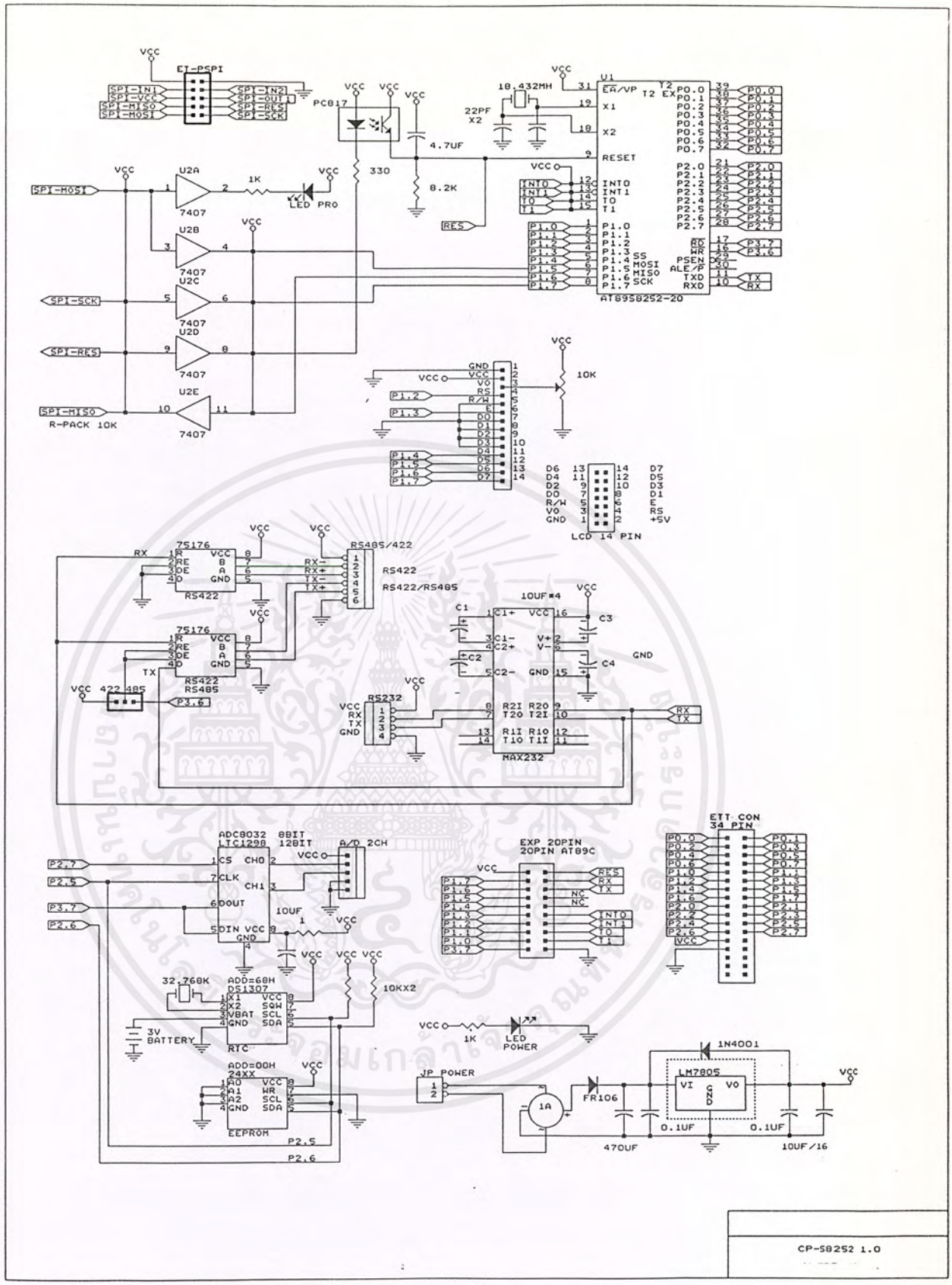
Character	E	1	2	3	4	5	6	7	8	9	10	…	…	37	38	39	40
DD RAM Address	E1	00	01	02	03	04	05	06	07	08	09	…	…	24	25	26	27
		40	41	42	43	44	45	46	47	48	49	…	…	64	65	66	67
	E2	00	01	02	03	04	05	06	07	08	09	…	…	24	25	26	27
		40	41	42	43	44	45	46	47	48	49	…	…	64	65	66	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

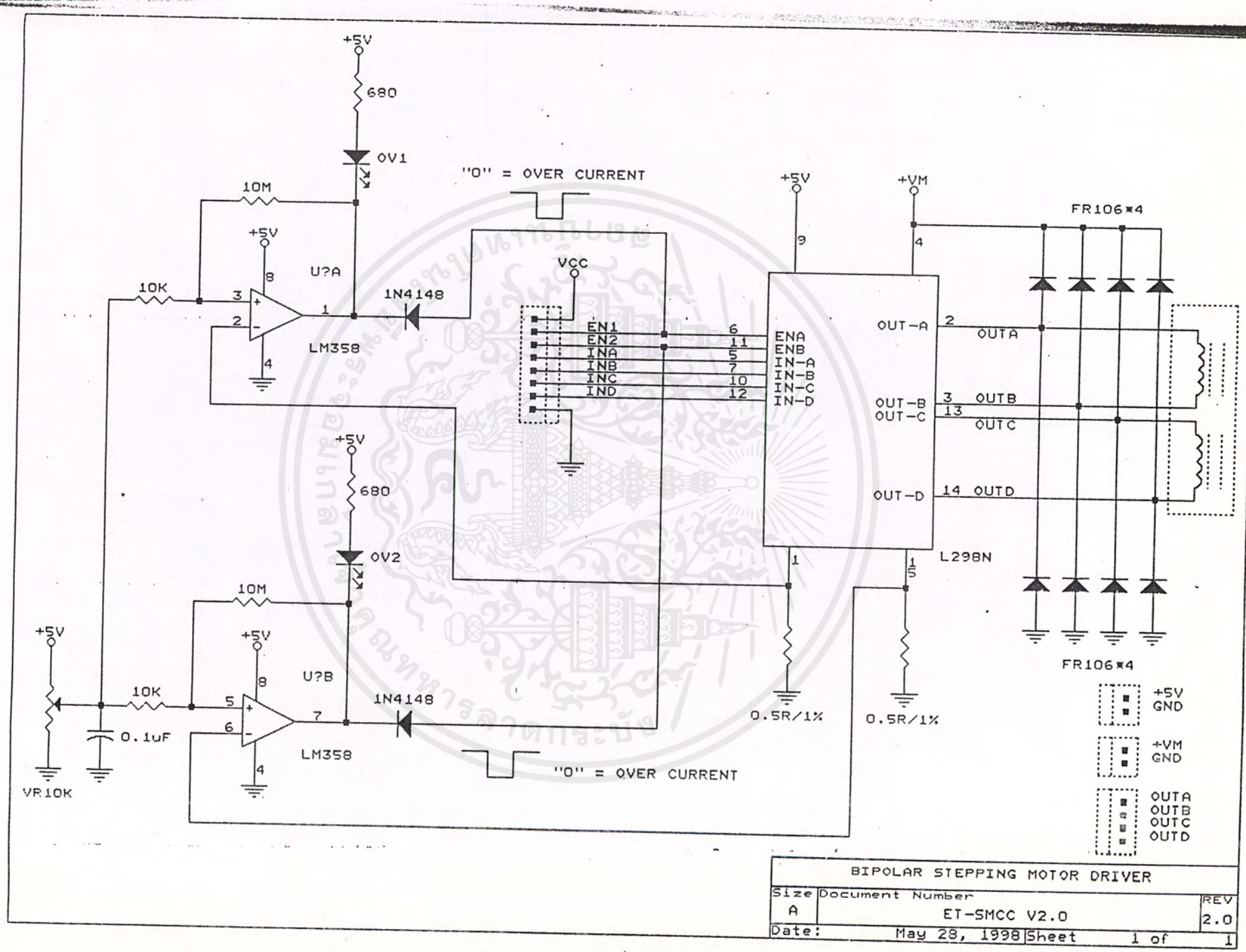
CG RAM MAPPING

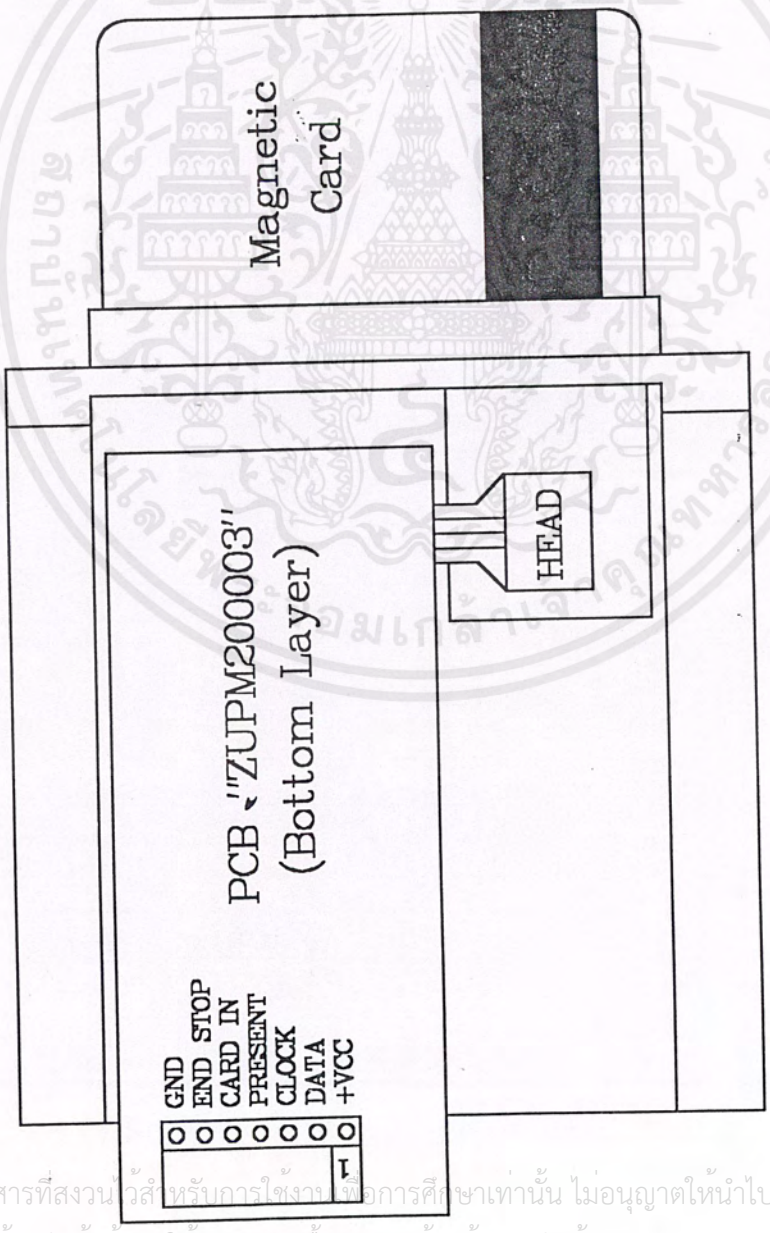
Character Code (DD RAM data)								CG RAM Address						Character Patterns (CG RAM data)								
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
High							Low	High					Low	High							Low	
											0	0	0				0	1	1	0	0	
											0	0	1				1	0	0	1	0	
											0	1	0				0	0	1	0	0	←Character
											0	1	1				0	1	0	0	0	Pattern
0	0	0	0	*	0	0	0	0	0	0	1	0	0	*	*	*	1	1	1	1	0	
											1	0	1				0	0	0	0	0	
											1	1	0				0	0	0	0	0	
											1	1	1				0	0	0	0	0	←Cursor
											0	0	0				1	1	1	1	1	
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	1	0	1	←Character
											0	1	1				1	0	1	1	1	Pattern
0	0	0	0	*	0	0	1	0	0	1	1	0	0	*	*	*	1	0	1	0	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	1	1	1	1	
											1	1	1				0	0	0	0	0	←Cursor
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
											0	0	0				1	1	1	1	1	
											0	0	1				1	0	0	0	1	
											0	1	0				1	1	1	0	1	←Character
											0	1	1				1	0	0	0	1	Pattern
0	0	0	0	*	1	1	1	1	1	1	1	0	0	*	*	*	1	0	1	1	1	
											1	0	1				1	0	0	0	1	
											1	1	0				1	1	1	1	1	
											1	1	1				0	0	0	0	0	←Cursor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


+5V
680
OV1
10M
+5V
U?A
8
10K
3
2
1
4
LM358
1N4148
"0" = OVER CURRENT
VCC
EN1
EN2
INA
INB
INC
IND
6
11
5
7
10
12
ENA
ENB
IN-A
IN-B
IN-C
IN-D
+5V
9
+VM
4
OUT-A
OUT-B
OUT-C
OUT-D
2
3
13
14
OUTA
OUTB
OUTC
OUTD
L298N
1
15
0.5R/1%
0.5R/1%
FR106*4
FR106*4
+5V
680
OV2
10M
+5V
U?B
8
10K
5
6
7
4
LM358
1N4148
0.1uF
+5V
VR10K
"0" = OVER CURRENT
+5V
GND
+VM
GND
OUTA
OUTB
OUTC
OUTD
BIPOLAR STEPPING MOTOR DRIVER
Size
A
Document Number
ET-SMCC V2.0
REV
2.0
Date:
May 28, 1998
Sheet
1 of
1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้