

ระบบการจัดการบนรถโดยสารประจำทางโดยแสดงผลภาษาไทย  
BUS MANAGEMENT SYSTEM WITH OUTPUT IN THAI



เลขหมู่.....  
เลขทะเบียน..... 28862  
วัน, เดือน, ปี 1.0 พ.ย. 2540

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ. 2540

ISBN 974-621-915-4

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**BUS MANAGEMENT SYSTEM WITH OUTPUT IN THAI**



**WORAPHON LILAKIATSAKUN**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE  
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**1997**

**ISBN 974-621-915-4**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบการจัดการบนรถโดยสารประจำทางโดยแสดงผลภาษาไทย
นักศึกษา	นายวรพล สีลาเกียรติสกุล
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.กอบชัย เดชหาญ
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2540

### บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอ ระบบจัดการบนรถประจำทาง ซึ่งจะมีการทำงานเป็นสองลักษณะ คือ ระบบจัดเก็บค่าโดยสารแบบไม่มีพนักงานขายตั๋ว โดยทำงานผ่านบัตรหน่วยความจำซึ่งสามารถโปรแกรมมูลค่าใหม่ได้ หรือให้ผู้โดยสารชำระเองก่อนลงจากรถ ซึ่งสามารถจะทราบค่าโดยสารจากส่วนแสดงผล และอีกลักษณะหนึ่งคือ ระบบบอกตำแหน่งของการเดินรถ ซึ่งจะแสดงผลเป็นข้อความเสียงภาษาไทย

**Thesis Title** Bus Management System with output in Thai  
**Student** Mr.Worraphon Lilakiatsakun  
**Thesis Advisor** Assoc.Prof.Kobchai Dejhan  
**Level of study** Master of Engineering in Electrical Engineering  
**Department** Telecommunication Engineering Faculty Of Engineering  
King Mongkut's Institute of Technology Ladkrabang  
**Year** 1997

### ABSTARCT

This thesis proposes a design of Bus Management System operating with two units. The first unit concerns about the fare ticket without ticket seller. It is operated by using memory smart card and it can be reprogrammed the new value. The another method is the passengers can pay cash before getting off the bus. The fare will be displayed for the passengers. The second unit deals with the bus position system. It announces the sentences and phrases by voice in Thai.

## กิตติกรรมประกาศ

วิทยานิพนธ์เรื่องนี้เสร็จด้วยความช่วยเหลือ และกำลังใจของ เพื่อน รุ่นพี่ น้อง รวมถึง รศ.ดร.กอบชัย เดชหาญ ซึ่งเป็นอาจารย์ที่ปรึกษา ขอขอบพระคุณอย่างสูงมา ณ. ที่นี้ และ ถ้าผลงานชิ้นนี้ทำประโยชน์แก่สังคมส่วนรวมได้บ้างก็ขออุทิศ ผลงานนี้ เป็นสาธารณะ เพื่อที่จะ นำไปศึกษา พัฒนา และเผยแพร่ต่อไปได้ และขอขอบพระคุณทุนอุดหนุนการทำวิทยานิพนธ์ จากบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วรพล สีลาเกียรติสกุล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VIII
<b>บทที่</b>	
1. บทนำ.....	1
จุดประสงค์ของโครงการในการทำวิทยานิพนธ์.....	1
ภาพรวมของระบบจัดการบทรกโดยสารประจำทาง.....	1
2. ทฤษฎีที่เกี่ยวข้อง.....	6
ไมโครคอนโทรลเลอร์ชิพเดี่ยว MCS51.....	6
โมดูลแสดงผล LCD รุ่น LM6061SYR-2.....	19
หน่วยความจำ EEPROM 93C46.....	30
ชิพรักษาเวลาจริง DS1202.....	38
เซนเซอร์ระยะทาง.....	42
ชิพสังเคราะห์เสียงแบบ ADPCM รุ่น OKI MSM6376.....	44
3. การออกแบบ.....	63
การออกแบบทางฮาร์ดแวร์.....	63
การออกแบบทางซอฟต์แวร์.....	75
4. การทดลองและผลการทดลอง.....	83
การทดลองเชื่อมต่อเซนเซอร์.....	83
การทดลองเชื่อมต่อเครื่องพิมพ์.....	84
การทดลองเชื่อมต่อโมดูล LCD.....	85

## สารบัญ (ต่อ)

บทที่	หน้า
การทดลองเชื่อมต่อชิพฐานเวลาจริง.....	87
การทดลองเชื่อมต่อ EEPROM.....	87
การทดลองเชื่อมต่อส่วนสังเคราะห์เสียง.....	88
5. สรุปผล.....	91
บรรณานุกรม.....	92
ภาคผนวก.....	93
ภาคผนวก ก. วงจรสมบูรณีสวนต่าง ๆ .....	94
ภาคผนวก ข. รายละเอียดโปรแกรมควบคุมระบบจัดการบนรถประจำทาง.....	101
ภาคผนวก ค. รายละเอียดโปรแกรมส่วนแสดงผลจุดเมตริกซ์.....	134
ประวัติผู้เขียน.....	144



## สารบัญตาราง

ตารางที่	หน้า
1 แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ SCON.....	11
2 แสดงการเลือกโหมดการรับส่งข้อมูลอนุกรมของบิต SM1.SM0.....	13
3 แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ TMOD.....	14
4 แสดงการเลือกโหมดโดยบิต M1 และ M0.....	14
5 แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ TCON.....	15
6 แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ IE.....	16
7 แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ IP.....	17
8 แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ PCON.....	18
9 แสดงรายละเอียดข้อกำหนดในการเชื่อมต่อ LCM.....	19
10 แสดงการกำหนดขนาดตัวอักษรของจอ LCD.....	21
11 แสดงการกำหนดจุดแวนอนต่อหนึ่งตัวอักษร.....	21
12 แสดงการกำหนดจำนวนของตัวอักษรในหนึ่งบรรทัด.....	22
13 แสดงการกำหนดจำนวนครั้งของการแบ่งเวลาในการแสดงผล.....	22
14 แสดงการกำหนดพื้นที่การแสดงผลเคอร์เซอร์.....	23
15 แสดงการกำหนดตำแหน่งเริ่มต้นการแสดงผลไบต์ต่ำ.....	23
16 แสดงการกำหนดตำแหน่งเริ่มต้นการแสดงผลไบต์สูง.....	24
17 แสดงการกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์ต่ำ.....	24
18 แสดงการกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์สูง.....	25
19 แสดงการเขียนข้อมูลเพื่อแสดงผล.....	25
20 แสดงการอ่านข้อมูลที่ตำแหน่งเคอร์เซอร์.....	26
21 แสดงการเคลียร์บิตข้อมูลของการแสดงผล.....	27

## สารบัญตาราง

ตารางที่	หน้า
22 แสดงการเซตบิตข้อมูลของการแสดงผล.....	27
23 แสดงการอ่านแฟล็กบิวซี (Busy Flag).....	28
24 แสดงความหมายของสัญลักษณ์ที่ใช้ในการแสดงผลบนจอ LCD.....	29
25 แสดงชุดคำสั่งของ 93C46.....	33
26 แสดงรายละเอียดไปต์คำสั่ง.....	39
27 แสดงความสัมพันธ์ระหว่างจำนวนพัลส์ของ ST.....	49
28 แสดงกสณเลือกรหัสคำพูดและรหัสเสียงอื่นๆ.....	53
29 แสดงรหัสการเลือกเสียงบีบเสียงต่างๆ.....	56
30 แสดงความสัมพันธ์ของจำนวนพัลส์ ST กับการลดทอนสัญญาณ.....	57
31 แสดงระดับสัญญาณเอาท์พุตจากส่วน D/A.....	59
32 แสดงความถี่คัทออฟของวงจรกรองความถี่ต่ำผ่าน.....	59
33 แสดงระดับเอาท์พุตของวงจรกรองความถี่ต่ำผ่าน.....	60
34 แสดงการจัดวางตำแหน่งของหน่วยความจำและอินพุต/เอาท์พุต บนวงจรควบคุมการทำงานหลัก.....	64
35 แสดงการจัดวางตำแหน่งของหน่วยความจำและอุปกรณ์อินพุต/เอาต์พุต บนวงจรควบคุมการแสดงผลแบบจุดเมตริกซ์.....	72

## สารบัญภาพ

	หน้า
1 แสดงโครงสร้างการทำงานของระบบจัดการบนรถโดยสารประจำทาง.....	1
2 แสดงตัวอย่างตารางการคิดค่าโดยสารของตัวช่วงต่างๆ.....	2
3 แสดงโครงสร้างการจัดเก็บค่าโดยสาร.....	3
4 แสดงขั้นตอนการจัดเก็บค่าโดยสารผ่านบัตรหน่วยความจำ.....	3
5 แสดงขั้นตอนการจัดเก็บค่าโดยสารโดยให้ผู้โดยสารชำระเอง.....	4
6 แสดงสถานที่สำคัญของเส้นทางเดินรถของรถประจำทางสาย ปอ3.....	5
7 แสดงโครงสร้างการบอกระยะทาง.....	5
8 ขาสัญญาของ MCS51.....	20
9 รายละเอียดคำสั่งย่อยของโหมดควบคุม.....	26
10 แสดงขั้นตอนการอ่านข้อมูลที่ตำแหน่งเคอร์เซอร์.....	26
11 แสดงสัญลักษณ์ที่ใช้ในการกำหนดค่าต่าง ๆ.....	29
12 วิธีการแสดงผลข้อมูลในหน่วยความจำในโหมดตัวอักษรและโหมดกราฟฟิก.....	30
13 แสดงบล็อกไดอะแกรมและการจัดเรียงขาสัญญาของ 93C46.....	31
14 แสดงไดอะแกรมเวลาทั่วไป.....	34
15 แสดงไดอะแกรมเวลาการอ่าน.....	34
16 แสดงไดอะแกรมเวลาการไม่ยอมให้เขียนและลบข้อมูล.....	35
17 แสดงไดอะแกรมเวลาการไม่ยอมให้เขียนและลบข้อมูล.....	35
18 แสดงไดอะแกรมเวลาการเขียน.....	36
19 แสดงไดอะแกรมเวลาการลบ.....	36
20 แสดงไดอะแกรมเวลาการเขียนทั้งหมด.....	37
21 แสดงไดอะแกรมเวลาการลบทั้งหมด.....	37
22 แสดงบล็อกไดอะแกรมและการจัดเรียงขาสัญญาของ DS1202.....	38
23 แสดงไดอะแกรมเวลาการติดต่อแบบไบต์เดี่ยว.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ VIII ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ (ต่อ)

	หน้า
24 แสดงไดอะแกรมเวลาการติดต่อแบบหลายไบต์.....	41
25 แสดงไดอะแกรมเวลาการอ่านข้อมูล.....	42
26 แสดงไดอะแกรมเวลาการเขียนข้อมูล.....	42
27 แสดงโครงสร้างของเซนเซอร์วัดระยะทาง.....	43
28 แสดงการต่อเซนเซอร์เพื่อใช้งาน.....	43
29 แสดงบล็อกไดอะแกรมภายในของ MSM 6376.....	45
30 แสดงสัญญาณเวลาขณะเริ่มต้นจ่ายไฟเลี้ยง.....	46
31 แสดงสัญญาณเวลาเริ่มต้นทำงานและเข้าสู่สแตนด์บายโหมด.....	46
32 แสดงสัญญาณเวลาการทำงานกับช่องสัญญาณที่ 1 เท่านั้น.....	47
33 แสดงสัญญาณเวลาการทำงานกับช่องสัญญาณที่ 2 เท่านั้น.....	47
34 แสดงสัญญาณเวลาการเล่นเสียงสะท้อน ในช่องสัญญาณที่ 1.....	48
35 แสดงสัญญาณเวลาการเล่นเสียงพร้อมกัน 2 ช่องสัญญาณ.....	49
36 แสดงการจัดเรียงขาสัญญาณของ MSM 6376.....	50
37 แสดงสัญญาณการรับข้อมูลจาก CPU ( $ST \leq 350 \mu S$ ).....	54
38 แสดงสัญญาณการรับข้อมูลจาก CPU ( $ST \geq 800 \mu S$ ).....	54
39 แสดงสัญญาณเวลาการรับข้อมูลจากสวิตช์ (เล่นเสียง 1 คำ).....	55
40 แสดงสัญญาณเวลาการรับข้อมูลจากสวิตช์ (เล่นเสียงซ้ำ).....	55
41 แสดงการใส่รหัสหยุด.....	56
42 แสดงสัญญาณเวลาของการเล่นเสียงผสม 2 ช่องสัญญาณ.....	58
43 แสดงคุณสมบัติของวงจรกรองความถี่ต่ำผ่านที่ $F_s = 8$ กิโลเฮิรซ์.....	60
44 แสดง Noise ของวงจรกรองความถี่ต่ำผ่าน.....	61
45 แสดงวงจรจับ Pop Noise ภายนอก.....	61
46 แสดงการต่อวงจร RC ออสซิลเลเตอร์.....	61
47 แสดงคุณสมบัติของวงจร RC ออสซิลเลเตอร์.....	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญญภาพ (ต่อ)

	หน้า
48 แสดงวงจรการต่อคริสตอลออสซิลเลเตอร์.....	62
49 บล็อกไดอะแกรมหลักของการออกแบบระบบจัดการ.....	63
50 แสดงบล็อกไดอะแกรมของส่วนควบคุมระบบหลัก.....	64
51 แสดงวงจรควบคุมบนส่วนควบคุมระบบหลัก.....	65
52 แสดงการเชื่อมต่อกับเครื่องพิมพ์.....	66
53 แสดงการเชื่อมต่อโมดูล LCD.....	66
54 แสดงการเชื่อมต่อแป้นพิมพ์.....	67
55 แสดงการเชื่อมต่อกับบัตรหน่วยความจำ.....	68
56 แสดงการเชื่อมต่อกับชิพฐานเวลาจริง.....	68
57 แสดงการเชื่อมต่อกับเซนเซอร์.....	69
58 แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนควบคุมหลัก.....	69
59 แสดงบล็อกไดอะแกรมส่วนแสดงผลแบบจุดเมตริกซ์.....	70
60 แสดงวงจรควบคุมของส่วนแสดงผลแบบจุดเมตริกซ์.....	71
61 แสดงวงจรชุดขับกระแสของส่วนแสดงผลแบบจุดเมตริกซ์.....	72
62 แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนการแสดงผล.....	73
63 แสดงบล็อกไดอะแกรมการแสดงผลเสียงพูด.....	73
64 แสดงวงจรของส่วนแสดงผลเสียงพูด.....	74
65 แสดงไฟล์ชาร์ตแสดงการทำงานโปรแกรมหลัก.....	76
66 แสดงไฟล์ชาร์ตการเพิ่มจำนวนเงินในบัตร.....	77
67 แสดงไฟล์ชาร์ตการตรวจสอบข้อมูลการใช้บริการ.....	78
68 แสดงไฟล์ชาร์ตการทำงานระบบจัดการ.....	79
69 แสดงไฟล์ชาร์ตการคำนวณระยะทาง.....	80
70 แสดงไฟล์ชาร์ตการใช้บัตรหน่วยความจำ.....	81
71 แสดงไฟล์ชาร์ตการทำงานบนส่วนแสดงผลจุดเมตริกซ์.....	82

## สารบัญญภาพ (ต่อ)

	หน้า
72 แสดงสัญญาณที่ได้จากการหมุนแกนของเซนเซอร์.....	83
73 แสดงเซนเซอร์สำหรับวัดระยะทาง.....	84
74 แสดงผลการทดลองพิมพ์ในรูปแบบต่าง ๆ.....	84
75 แสดงการเชื่อมต่อเครื่องพิมพ์.....	85
76 แสดงผลการทดลองแสดงตัวอักษรบนจอ LCD.....	86
77 แสดงการเชื่อมต่อจอ LCD.....	86
78 ผลการทดลองกำหนดเวลาลงในชิพฐานเวลา.....	87
79 ผลการทดลองโปรแกรมมูลค่าจำนวนเงินลงบน EEPROM.....	87
80 สัญญาณเสียงที่ได้จากการสังเคราะห์.....	88
81 วงจรส่วนสังเคราะห์ เสียง.....	89
82 ส่วนแสดงผลจุดเมตริกซ์.....	89
83 แสดงต้นแบบของทั้งระบบ.....	90
84 แสดงวงจรควบคุมบนส่วนควบคุมระบบหลัก.....	95
85 แสดงการเชื่อมต่อกับเครื่องพิมพ์.....	95
86 แสดงการเชื่อมต่อโมดูล LCD.....	96
87 แสดงการเชื่อมต่อแป้นพิมพ์.....	96
88 แสดงการเชื่อมต่อกับบัตรหน่วยความจำ.....	97
89 แสดงการเชื่อมต่อกับชิพฐานเวลาจริง.....	97
90 แสดงการเชื่อมต่อกับเซนเซอร์.....	97
91 แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนควบคุมหลัก.....	98
92 แสดงวงจรชุดขับกระแสของส่วนแสดงผลแบบจุดเมตริกซ์.....	98
93 แสดงวงจรควบคุมของส่วนแสดงผลแบบจุดเมตริกซ์.....	99
94 แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนการแสดงผล.....	100
95 แสดงวงจรของส่วนแสดงผลเสียงพูด.....	100

# บทที่ 1

## บทนำ

### จุดประสงค์ของโครงการในการทำวิทยานิพนธ์

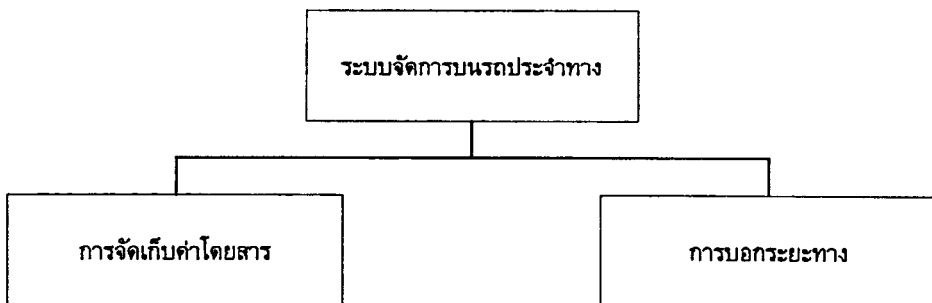
เพื่อสร้างวิธีจับเก็บค่าโดยสาร และการบอกระยะบนเส้นทางรถโดยสาร โดยไม่ใช้พนักงาน บนรถโดยสาร และสามารถ ตรวจสอบค่าโดยสาร และจำนวนผู้ใช้บริการ ในแต่ละเที่ยว ได้สะดวก และมีประสิทธิภาพมากขึ้น การบอกระยะบนเส้นทางรถโดยสาร นั้น เพื่อให้ผู้ใช้บริการ สามารถรู้ตำแหน่งคร่าว ๆ ของตัวเอง เพื่อที่จะเตรียมตัวลงที่จุดหมายได้ถูกต้อง ซึ่งจะมีการแสดงผล เป็นข้อความภาษาไทย และเสียงพูดภาษาไทย ซึ่งระบบเสียงนี้จะเป็นประโยชน์ต่อคนพิการทางสายตา ที่สามารถจะคาดคะเน จุดที่จะลงจากรถได้ถูกต้อง

### ภาพรวมของระบบจัดการบนรถโดยสารประจำทาง

ระบบจัดการบนรถประจำทางนี้ จะควบคุมการทำงานทั้งหมดด้วยระบบคอมพิวเตอร์ ซึ่งจะแบ่งโครงสร้างการทำงาน เป็น 2 ส่วน แสดงดังแผนภูมิที่ 1

#### แผนภูมิที่ 1

แสดงโครงสร้างการทำงานของระบบจัดการบนรถโดยสารประจำทาง

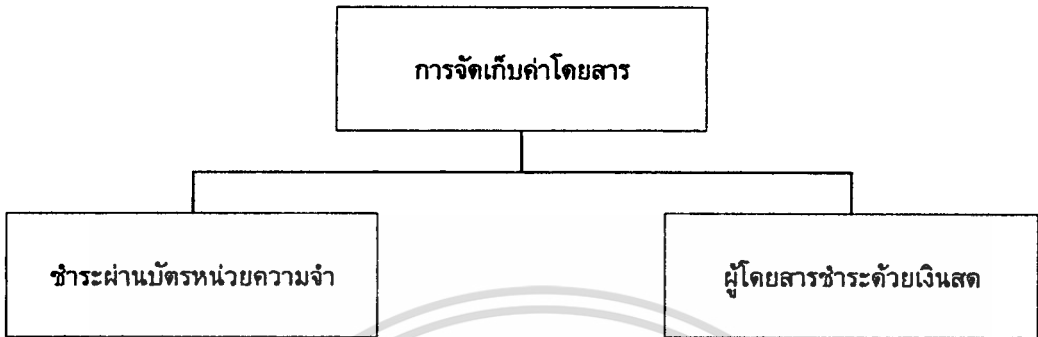


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### แผนภูมิที่ 3

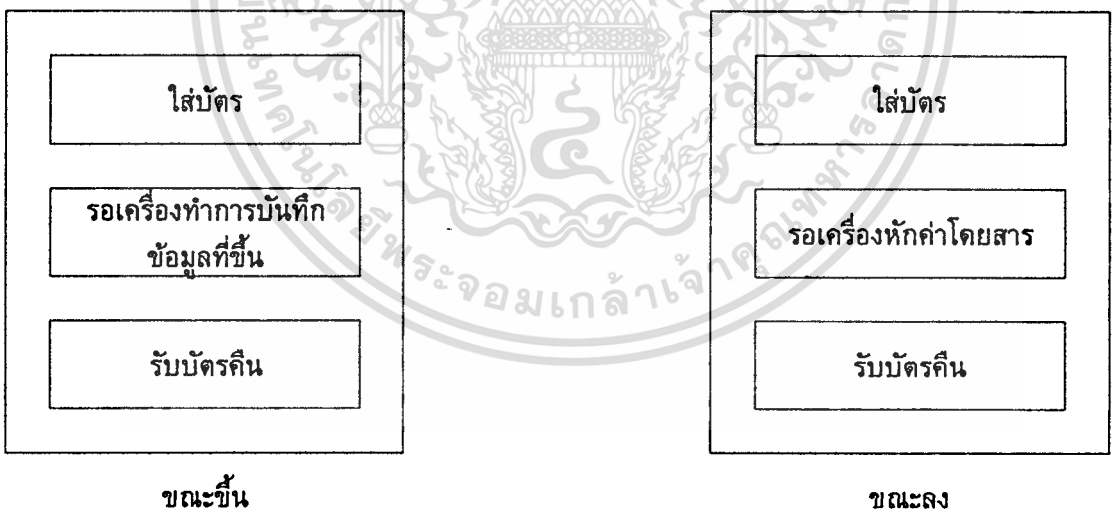
แสดงโครงสร้างการจัดเก็บค่าโดยสาร



1.1 การจ้ดเก็บค่าโดยสารผ่านบัตรหน่วยความจำ จะมีขั้นตอนการจ้ดเก็บดังแผนภูมิที่ 4

#### แผนภูมิที่ 4

แสดงขั้นตอนการจ้ดเก็บค่าโดยสารผ่านบัตรหน่วยความจำ



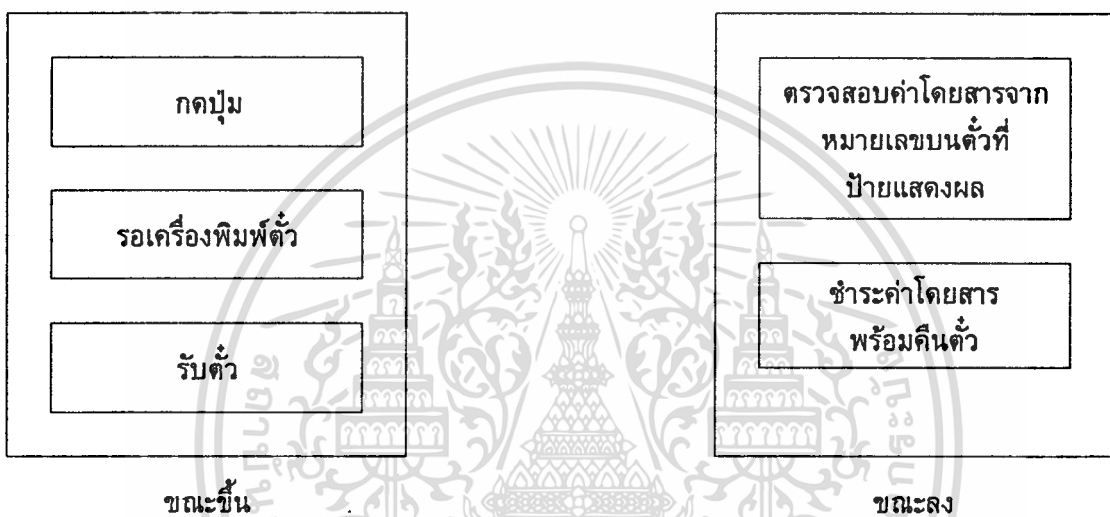
เมื่อผู้โดยสารขึ้นรถก็จะเสียบบัตรหน่วยความจำ เข้าไปในเครื่องอ่านบัตร หลังจากนั้นจะทำการบันทึกหมายเลขของช่วง ที่ขึ้นลงในบัตรพร้อมกับแสดงจำนวนเงินที่มีอยู่ในบัตร บนจอแสดงผล จากนั้นผู้โดยสารก็รับบัตรคืน และเมื่อต้องการจะลงจากรถ ผู้โดยสารก็ใส่บัตรเข้าไปในเครื่องอ่านบัตรอีกครั้งหนึ่ง เครื่องอ่านบัตรจะอ่านข้อมูลระยะเริ่มต้นที่โดยสาร เพื่อนำมาคำนวณ

ค่าโดยสาร ณ จุดที่ลง แล้วทำการหักค่าโดยสารจากบัตร แล้วแสดงจำนวนเงินที่เหลือ พร้อมกับค่าโดยสารที่ได้หักไป ผู้โดยสาร รับบัตรคืน การเก็บ โดยวิธีนี้จะไม่มีการพิมพ์ตัวโดยสาร

1.2 การจัดเก็บค่าโดยสารโดยให้ผู้โดยสารชำระเอง จะมีขั้นตอนการจัดเก็บดังแผนภูมิ

### แผนภูมิที่ 5

แสดงขั้นตอนการจัดเก็บค่าโดยสารโดยให้ผู้โดยสารชำระเอง



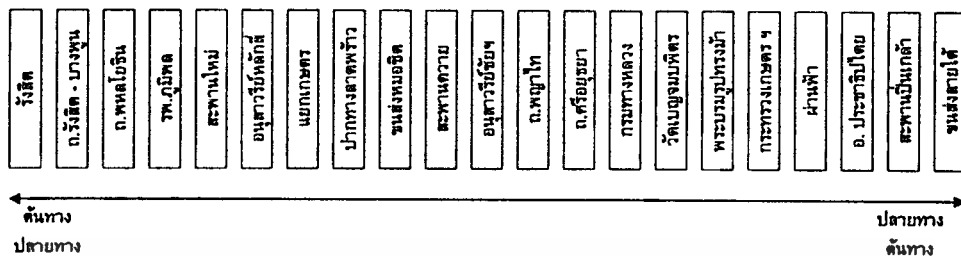
เมื่อผู้โดยสารขึ้นรถ ก็กดปุ่มเพื่อรับตั๋ว ซึ่งบนตั๋วจะมีการพิมพ์หมายเลขแสดงช่วงเริ่มต้นของการขึ้นรถ จากนั้นผู้โดยสาร ก็จะถือตั๋วโดยสารไว้ และเมื่อต้องการจะลง ก็ดูราคาค่าโดยสารจากป้ายแสดงผลภายในรถ และชำระค่าโดยสารพร้อมกับตั๋วลง กล่องรับเงินก่อนลงจากรถ

ทั้งสองวิธีนี้จะมีการบันทึก ข้อมูลของการใช้บริการทั้งหมดลงหน่วยความจำ เพื่อสามารถตรวจสอบในภายหลัง โดยการ พิมพ์รายละเอียด การใช้บริการในแต่ละเที่ยวโดยสารได้

2. การบอกระยะทางของเส้นทางรถ การบอกระยะทางนี้จะใช้การบอกเป็นระยะเช่น ทุกๆ 1 กิโลเมตร หรือ ทุกๆ ช่วงที่กำหนด โดยแบ่งเส้นทางรถเป็นช่วงๆ และเก็บข้อมูลสถานที่สำคัญต่างๆ ของช่วงนั้นๆ เมื่อรถเดินทางถึงระยะที่กำหนดก็จะบอกชื่อสถานที่สำคัญเหล่านั้น ซึ่งเปรียบเสมือนการบอกระยะทางรถ ตัวอย่างของเส้นทางรถของประจำทางสาย ปอ.3 แสดงดังแผนภูมิที่ 6

### แผนภูมิที่ 6

แสดงสถานที่สำคัญของเส้นทางเดินรถของรถประจำทางสาย ปอ.3



การบอกระยะทางนี้ จะใช้ วิธีการ บอกสถานที่ ซึ่งจะสัมพันธ์กับระยะทางที่รถเคลื่อนไปโดย สถานที่ที่จะใช้บอกนั้นจะห่าง กันประมาณ 2 กิโลเมตร หรืออาจจะน้อยกว่า วิทยานิพนธ์นี้เสนอการบอกระยะทางนี้เป็น 2 ลักษณะ ซึ่งใช้วิธีการบอกระยะทาง ดังที่กล่าวมา โดยแสดงดังแผนภูมิที่ 7

### แผนภูมิที่ 7

แสดงโครงสร้างการบอกระยะทาง



2.1 บอกเป็นข้อความ จะทำงานผ่านส่วนแสดงผลแบบจุดเมตริกซ์ (Dot Matrix) ซึ่งจะแสดง ข้อความของระยะต้นทาง ระยะปลายทาง และระยะปัจจุบัน รวมถึงข้อความอื่นๆ

2.2 บอกเป็นเสียงพูด จะพูดข้อความต่าง ๆ ที่กำหนดผ่านลำโพง เช่น สถานที่สำคัญต่าง ๆ และคำพูดอื่นๆ

ทั้งสองวิธีนี้จะทำงานอัตโนมัติ โดยระบบจะตรวจสอบระยะทางที่รถเคลื่อนที่ไปได้ แล้วแสดงผลตามระยะต่างๆ และถ้ามีการเปลี่ยนแปลงเข้าสู่ระยะใหม่ก็จะแสดงผลข้อความใหม่

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### ไมโครคอนโทรลเลอร์ชิพเดี่ยว MCS51

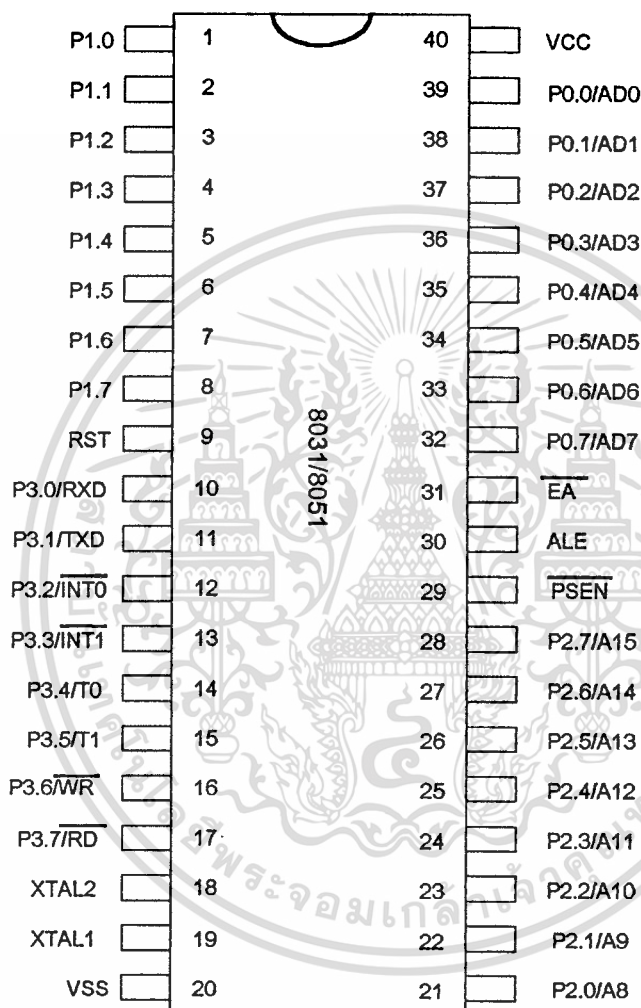
1. คุณสมบัติเฉพาะของ MCS51 นั้นมีการทำงานที่มีความสามารถพิเศษดังต่อไปนี้

- 1.1 สามารถนำเอาข้อมูลมา AND, OR หรือทำ Complement ทั้งแบบทีละ 8 บิต และ 1 บิต
- 1.2 สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บชุดคำสั่งที่จะ ให้ MCS51 ทำงานได้สูงสุด 64 กิโลไบต์
- 1.3 สามารถต่อกับหน่วยความจำสำหรับข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการ ทำงานของโปรแกรมได้สูงสุด 64 กิโลไบต์
- 1.4 ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมจำนวน 4 กิโลไบต์ อยู่ในวงจรรวม ทำให้ไม่ต้องต่อหน่วย ความจำสำหรับโปรแกรมอยู่ภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็ก
- 1.5 มีพอร์ตแบบขนาน (Parallel Port) สำหรับข้อมูลเข้า และออก จำนวน 32 บิต ที่ข้อมูลแต่ละบิตเป็นอิสระต่อกัน
- 1.6 มีวงจร Timer/Counter ขนาด 16 บิต 2 ชุด (8052 มี 3 ชุด) ที่ทำงานในโหมดต่าง ๆ ได้ถึง 4 โหมด
- 1.7 มี Universal Asynchronous Receiver Transmitter (UART) สำหรับ รับ - ส่งข้อมูลอนุกรมแบบ Full Duplex ที่ สามารถเลือกรูปแบบการรับ - ส่งข้อมูลได้ 4 แบบ
- 1.8 มีแหล่งกำเนิดสัญญาณขอขัดจังหวะการทำงานของโปรแกรม (Interrupt Request Signal) 6 แหล่ง ซึ่งสามารถ กระโดดไปทำงานตอบสนองการขัดจังหวะได้ต่าง ๆ กัน 5 ตำแหน่ง
- 1.9 สามารถเลือกการทำงานให้อยู่ในโหมดของ Idle และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. รายละเอียดขาสัญญาณของ MCS51 จะบรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งจะมีขาสัญญาณรวมทั้งหมด 40 ขานั้น จะใช้งาน ต่างๆ กัน แสดงดังภาพที่ 8

ภาพที่ 8  
แสดงขาสัญญาณของ MCS51



2.1 VCC ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง +5 โวลท์เข้าไป เพื่อให้วงจรรวมทำงานได้

2.2 VSS ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ

2.3 Port 0 เป็นพอร์ตขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับดังภาพที่ 1.6 แต่ละขาจะเขียนว่า P0.0, P0.1, ..., P0.7 สำหรับ P0.7 หมายถึงบิต 7 ของพอร์ต 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด (MSB) และ P0.0 คือบิต 0 ของพอร์ต 0 ซึ่งเป็นบิตที่มีนัยสำคัญต่ำสุด (LSB) พอร์ต 0 นี้ ใช้ได้ทั้งการรับ - ส่งตำแหน่ง และข้อมูลกับหน่วยความ (Data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือใช้เป็น พอร์ตรับ - ส่งข้อมูล (I/O Port) ก็ได้ ข้อมูลที่ส่งออกทางพอร์ต 0 จะถูกค้างสภาวะไว้ที่ขาของพอร์ต โครงสร้างแต่ละบิตของพอร์ต 0 เป็น แบบ Open Drain Bidirectional

2.4 Port 1 เป็นพอร์ตขนานขนาด 8 บิต คือขา P1.0 ถึง P1.7 (ขา 1 - 8) P1.0 หมายถึง บิต 0 ของพอร์ต 1 ซึ่งเป็นบิตนัยสำคัญต่ำสุด และบิต P1.7 หมายถึงบิตที่ 7 ของพอร์ต 1 ซึ่งเป็นบิตนัยสำคัญสูงสุด พอร์ตนี้จะทำงานเป็น อินพุต/เอาต์พุต พอร์ต เท่านั้น

2.5 Port 2 พอร์ตขนานขนาด 8 บิต คือขา P2.0 ถึง P2.7 (ขา 21 - 28) โครงสร้างของพอร์ต 2 แต่ละบิตลักษณะโครงสร้างจะเหมือน กับพอร์ต 0 แตกต่างกันในพอร์ต 2 นั้น จะใช้งานเพียง 2 ลักษณะคือ

2.5.1 ใช้ส่งค่าแอดเดรส 8 บิตบนให้หน่วยความจำภายนอก (A8 - A15)

2.5.2 ใช้เป็นพอร์ตรับ - ส่งข้อมูลกับภายนอก

2.6 Port 3 เป็นพอร์ตขนานขนาด 8 บิต คือขา P3.0 ถึง P3.7 (ขา 10 - 17) P3.0 หมายถึงบิต 0 ของพอร์ต 3 ซึ่งเป็นบิตนัยสำคัญ ต่ำสุด และบิต P3.7 หมายถึงบิตที่ 7 ของพอร์ต 3 ซึ่งเป็นบิตนัยสำคัญสูงสุด นอกเหนือจากการทำงานเพื่อรับ - ส่งข้อมูลแล้ว แต่ละบิตของพอร์ตนี้ยังสามารถทำหน้าที่อื่น ๆ ดังนี้

2.6.1 P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

2.6.2 P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

2.6.3 P3.2/ $\overline{\text{INT0}}$  (External Interrupt) ใช้รับสัญญาณขั้วจ้งหระจากภายนอก

2.6.4 P3.3/INT1 (External Interrupt) ใช้รับสัญญาณขั้วจ้งหระจากภายนอก

2.6.5 P3.4/T0 (Timer/Counter 0 External) ขารับสัญญาณเข้าไปยังวงจร

Timer/Counter 0

2.6.6 P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร

Timer/Counter 1

2.6.7 P3.6/ $\overline{\text{WR}}$  (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก

2.6.8 P3.7/ $\overline{\text{RD}}$  (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

2.7 RST ขานี้ใช้ทำการรีเซ็ตการทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ ถ้าป้อน สัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการรีเซ็ตการทำงานของ 8051 ดังนั้นจึงสามารถต่อตัวเก็บประจุภายนอก ระหว่าง ขา RST กับไฟเลี้ยง +5 โวลท์ เพื่อให้เกิดการรีเซ็ตเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power on reset การรีเซ็ตเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำให้ ค่าในรีจิสเตอร์ต่าง ๆ เปลี่ยนไปเป็นค่าหนึ่ง เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซ็ต 8051 จะเริ่มทำงานจาก คำสั่งที่อยู่ในหน่วยความจำโปรแกรม ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรม ที่จะทำงานถูกเปลี่ยนให้เป็น 0000H

2.8 ALE Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์ สัญญาณนี้ จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 8051 สัญญาณนี้จะใช้บอกกับ อุปกรณ์ภายนอก 8051 ว่าขณะนี้สัญญาณนี้แอกทีฟ (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่งหน่วยความจำ ภายนอก 8051 ที่ต้องการติดต่อออกไปทางพอร์ต 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ต 0 จะส่ง ค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ต 0 จะใช้รับ - ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุต

2.9 PSEN Program Store Enable ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไป ทำงานมาจากหน่วยความจำโปรแกรมภายนอก 8051 ในกรณีทีอ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำ แต่สำหรับโปรแกรมภายใน 8051 แล้ว สัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุตของ TTL ชนิด LS ได้ถึง 8 อินพุต

2.10  $\overline{EA}$  External Access ขานี้เป็นขาอินพุตที่ต่อเข้าไปยังวงจร Timing and Control ใน เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา  $\overline{EA}$  นี้ แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอก เพื่อทำการอ่านคำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา  $\overline{EA}$  เป็น 1 หมายความว่า โปรแกรมในตำแหน่ง 0000H ถึง 0FFFFH ถูกเก็บไว้ภายใน 8051 การทำงานในตำแหน่งหน่วยความจำ ช่วงนี้จะ อ่านคำสั่งต่าง ๆ จาก ROM ภายใน 8051

2.11 XTAL1 ขานี้ต่ออยู่กับวงจรออสซิลเลเตอร์ภายใน ซึ่งจะใช้งานร่วมกับขา XTAL2 เพื่อสร้างสัญญาณนาฬิกาให้กับระบบ โดยการ ต่อคริสตอล และตัวเก็บประจุภายนอก ซึ่งมีค่าประมาณ 20 pf แต่ถ้าไม่ต้องการใช้การกำเนิดสัญญาณนาฬิกาจากภายใน สามารถป้อนสัญญาณนาฬิกาเข้าทางขานี้ได้ทันที

2.12 XTAL2 ขานี้จะใช้ร่วมกับขา XTAL1 เพื่อกำเนิดสัญญาณนาฬิกาจากภายใน ในกรณีใช้สัญญาณนาฬิกาภายนอก ให้ปล่อยขานี้ลอยไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register, SFR) จะเป็นรีจิสเตอร์ที่อยู่ภายใน MCS51 เพื่อใช้ทำงานหน้าที่พิเศษและควบคุมฮาร์ดแวร์ภายใน

3.1 Accumulator (A) ตำแหน่งหน่วยความจำภายในเท่ากับ 0E0H รีจิสเตอร์นี้มีขนาด 8 บิต เป็นรีจิสเตอร์ที่ใช้มาก ซึ่งในรหัสคำสั่งช่วยจำจะอ้างอิงถึงรีจิสเตอร์นี้โดยใช้สัญลักษณ์ A เช่น MOV A,#15H คำสั่งที่จะอ่าน หรือเก็บข้อมูลกับหน่วยความจำภายนอกจะต้องกระทำผ่านรีจิสเตอร์นี้เท่านั้น เช่น MOVX @R0,A หรือ MOVX A,@R0 เป็นต้น และข้อมูลที่อยู่ภายในรีจิสเตอร์นี้ก็สามารถที่จะให้โปรแกรม ตรวจสอบเพื่อกระโดดการทำงานไปยังตำแหน่งอื่นได้ เช่น JZ rel

3.2 B Register (B) ตำแหน่งหน่วยความจำภายในเท่ากับ 0F0H รีจิสเตอร์ขนาด 8 บิต ที่ใช้ในคำสั่งการคูณ (MUL AB) และคำสั่งหาร (DIV AB) เท่านั้น โดยรีจิสเตอร์ B นี้จะเก็บ ตัวคูณ และผลลัพธ์ 8 ถึง บิต 15 ในคำสั่งการคูณ ส่วนในคำสั่งหารนั้นรีจิสเตอร์ B จะเก็บตัวหารและผลหาร การเขียนข้อมูล ไปยังรีจิสเตอร์นี้ จะต้องใช้คำสั่งเคลื่อนย้ายข้อมูลไปยังตำแหน่ง 0F0H เช่น MOV 0F0H,25H จะเป็นการกำหนดค่า 25H ให้กับ รีจิสเตอร์

3.3 Program Status Word (PSW) ตำแหน่งหน่วยความจำภายในเท่ากับ 0D0H รีจิสเตอร์ขนาด 8 บิต ที่แต่ละบิตจะบอกสถานะต่าง ๆ ดังรายละเอียด แต่บิตของ PSW จะสามารถกำหนดให้เป็น 1 หรือ 0 ได้ด้วยคำสั่ง SETB หรือ CLR ตามลำดับ ค่าตำแหน่งบิต 0 ถึง บิต 7 ของรีจิสเตอร์ PSW เท่ากับ D0H ตามลำดับ

3.4 Stack Pointer (SP) ตำแหน่งหน่วยความจำภายในเท่ากับ 081H รีจิสเตอร์ขนาด 8 บิต รีจิสเตอร์นี้จะชี้ตำแหน่งความจำภายใน 8051 ที่ใช้เก็บตำแหน่งเดิม ของโปรแกรม ก่อนทำงานคำสั่ง CALL หรือตำแหน่งที่จะใช้เก็บข้อมูลด้วยคำสั่ง PUSH และตำแหน่งที่จะอ่านข้อมูลออกมาในคำสั่ง POP เมื่อทำ การรีเซ็ต 8051 โดยการป้อนสัญญาณสถานะลอจิก 1 เข้าไปที่ขา RST ของ 8051 จะทำให้ข้อมูลในรีจิสเตอร์นี้มีค่าเป็น 07H หมายความว่ารีจิสเตอร์ SP ชี้หน่วยความจำภายใน 8051 ที่ตำแหน่ง 07H ค่าของ SP จะเปลี่ยนแปลงไปโดยการ ใช้ คำสั่งเคลื่อนย้าย ข้อมูล หรือการทำงานของคำสั่ง PUSH, POP, CALL

3.5 Data Pointer (DPTR) ตำแหน่งหน่วยความจำภายในเท่ากับ 82H และ 83H รีจิสเตอร์ DPTR มีขนาด 16 บิต หน้าที่ของรีจิสเตอร์นี้คือใช้สำหรับชี้ตำแหน่งในหน่วยความจำรีจิสเตอร์ DPTRสามารถใช้อ้างอิงตำแหน่งหน่วยความจำได้ สูงสุด 64 K ตำแหน่ง เช่น คำสั่ง MOVXA,@DPTR รีจิสเตอร์ DPTR ประกอบด้วยรีจิสเตอร์ ขนาด 8 บิต 2 ตัวคือ DPH และ DPL ดังนั้นการแก้ไขข้อมูลในรีจิสเตอร์ DPTR จึงทำได้ทั้งทีละ 16 บิต หรือทำการทีละ 8 บิต โดยการแก้ไขข้อมูลใน DPH หรือ DPL

3.6 Port 0 ถึง 3 (P0 - P3) ตำแหน่งหน่วยความจำภายในเท่ากับ 80H, 90H, 0A0H และ 0B0H สำหรับข้อมูลภายใน 8051 ที่ตำแหน่ง 80H, 90H, 0A0H และ 0B0H ตามลำดับ การเขียนข้อมูลลงไปยังหน่วยความจำ แต่ละตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ตนั้น ๆ ของ 8051 ข้อมูลที่เขียนออกไปจะถูก LATCH ค้างไว้ และปรากฏอยู่ที่แต่ละบิต ของพอร์ต ในการอ่านข้อมูล จากรีจิสเตอร์แต่ละตัวก็จะเป็นการอ่านสภาวะลอจิกของสัญญาณที่ปรากฏอยู่ที่แต่ละขาของ พอร์ตนั้น ๆ เช่น MOV A,80H เป็นการอ่านสภาวะลอจิกจากพอร์ต 0 เข้ามายัง Accumulator การอ่านข้อมูลจากพอร์ต จะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ตนั้น ๆ เสียก่อน ทุกบิตของ พอร์ต 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้ โดยใช้คำสั่ง SETB bit และ CLR bit

3.7 Serial Data Buffer (SBUF) ตำแหน่งหน่วยความจำภายในเท่ากับ 99H รีจิสเตอร์ นี้มีขนาด 8 บิต โครงสร้างภายในรีจิสเตอร์นี้มี 2 ตัว ที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับข้อมูลที่จะส่ง แบบอนุกรม ออกจาก 8051 และอีกตัวหนึ่งสำหรับข้อมูลแบบอนุกรมที่เข้ามา ดังนั้น พอร์ตอนุ กรมของ 8051 จึงเรียกว่ามีการทำงานแบบ Full Duplex เพราะสามารถส่ง และรับข้อมูลได้ใน เวลาเดียวกัน เนื่องจากมีรีจิสเตอร์สำหรับส่ง และรับ แยกออกจากกัน ข้อมูลที่ต้องการ จะส่ง ออกก็ให้เขียนไปยังรีจิสเตอร์ SBUF แล้วสั่งงานให้ส่งข้อมูลออกมา ข้อมูลในรีจิสเตอร์จะเริ่มส่ง ออกโดยเริ่มจากบิต 0 ถึง 7 ตามลำดับ ถ้ามีข้อมูลเข้ามาทางขา RXD ก็จะถูกเก็บไปไว้ในรีจิส เตอร์นี้ โดยถือว่าข้อมูลบิตแรกที่เข้ามาคือบิต 0

3.8 Serial Port Control (SCON) ตำแหน่งหน่วยความจำภายในเท่ากับ 98H รีจิส เตอร์ SCON มีขนาด 8 บิต ใช้สำหรับควบคุมการส่ง และรับข้อมูล ผ่านทางพอร์ตอนุกรม แต่ละ บิตของข้อมูลใน รีจิสเตอร์นี้ มีความหมายเฉพาะดังนี้

### ตารางที่ 1

แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ SCON

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SMO	SMI	SM2	REN	TB8	RB8	TI	RI

3.8.1 บิต RI (Receive Interrupt Flag) บิตนี้จะถูกกำหนดโดยฮาร์ดแวร์ ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโหมด 0 นั้น บิต RB8 จะมีค่าเป็น 1 เมื่อมีข้อมูล เข้ามาครบทั้ง 8 บิต ส่วนในโหมดอื่น บิต RB8 จะเป็น 1 ก็ต่อเมื่อข้อมูลถึงเวลาครึ่งหนึ่งของ Stop Bit (ยกเว้นบาง กรณี ให้ดูที่เรื่องบิต SM2 ของรีจิสเตอร์ SCON) บิตนี้จะสามารถเคลียร์ให้มีค่าเป็น 0 ได้โดยใช้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง CLR Bit โดยค่าตำแหน่งของบิตมีค่าเท่ากับ 98H บิตนี้มีประโยชน์ให้รู้ว่า ข้อมูลได้เข้ามา อยู่ใน SBUF ครบทั้งชุด แล้วพร้อมที่ CPU จะอ่านไปเก็บในหน่วยความจำต่อไป หรืออาจกำหนด ค่าในรีจิสเตอร์ IE และ IP เพื่อว่าเมื่อมีข้อมูลเข้ามาทางพอร์ตอนุกรมแล้วจะทำให้เกิดการขัด จังหวะ (Interrupt) การทำ งานของโปรแกรมหลัก แล้วกระโดดไปทำงานในโปรแกรมตอบสนอง การขัดจังหวะ

3.8.2 บิต TI (Transmit Interrupt Flag) ค่าในบิต TI จะถูกกำหนดให้เป็น 1 หรือ 0 ด้วยฮาร์ดแวร์ โดยในการส่งข้อมูลแบบอนุกรมโหมด 0 บิตนี้จะเป็น 1 เพื่อจะ บอกว่าการส่งข้อมูล ในรีจิสเตอร์ SBUF ออกไปทางพอร์ตอนุกรมครบทั้ง 8 บิต แต่ถ้าเป็นการส่งข้อมูลแบบอนุกรมใน โหมดอื่น จะทำ ให้ข้อมูลในบิต TI เป็น 1 เมื่อเริ่มการส่ง Stop Bit ข้อมูลบิตนี้สามารถเคลียร์เป็น 0 ได้ด้วยคำสั่ง CLR bit โดยที่ค่าตำแหน่งของ บิต นี้เท่ากับ 99H บิตนี้ยังมีประโยชน์เพื่อบอกว่า การส่งข้อมูลจาก SBUF ออกไปทางพอร์ตอนุกรมนั้นสิ้นสุดแล้ว พร้อมทั้งจะให้ โปรแกรม เขียน ข้อมูลลงไปยัง SBUF สำหรับการส่งออกไปได้ นอกจากนี้การกำหนดค่าในรีจิสเตอร์ IE และ IP ยังสามารถที่จะ กำหนด ให้เกิดการขัดจังหวะการทำงานของโปรแกรมได้ เมื่อบิตนี้ถูกฮาร์ดแวร์ ทำให้มีค่าเป็น 1

3.8.3 บิต RB8 เมื่อมีการกำหนดให้รับข้อมูลในโหมด 2 และ 3 จะใช้บิตนี้สำหรับ เก็บข้อมูลบิตที่ 9 ที่เข้ามาทางพอร์ตอนุกรม ส่วนในโหมด 1 นั้น บิตนี้จะเก็บ Stop bit ซึ่งมีค่าเป็น 1 นั้นเอง ในโหมด 0 บิตนี้จะไม่ถูกใช้งาน ค่าตำแหน่งของบิตนี้คือ 9AH

3.8.4 บิต TB8 ในการส่งข้อมูลแบบอนุกรมโหมด 2 และ 3 จะใช้บิตนี้เก็บข้อมูลบิตที่ 9 ส่วนโหมดอื่นจะไม่ใช้งานบิตนี้ การกำหนดค่าใน บิตนี้ สามารถทำได้โดยใช้คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิตนี้คือ 9BH

3.8.5 บิต REN (Receive Enable) เป็นบิตที่จะใช้กำหนดให้ทำการรับข้อมูลเข้ามา จากทางพอร์ตอนุกรม หรือไม่ถ้าบิตนี้เป็น 1 ก็จะได้รับข้อมูลเข้ามา แต่ถ้า เป็น 0 ก็จะไม่รับข้อมูลที่ ษา RXD เข้ามา การให้บิตนี้เป็น 1 หรือ 0 ทำได้โดยใช้คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่ง ของบิต นี้คือ 9CH

3.8.6 บิต SM2 ถ้ากำหนดบิต SM2 เป็น 0 จะมีการทำงานตามปกติที่ได้กล่าวมาใน เรื่องของการแยกที่ฟลล็ก RI แต่ถ้ากำหนด SM2 เป็น 1 และอยู่ในโหมด 2 และ 3 ซึ่งปกติแล้ว บิต RI จะเป็น 1 เมื่อข้อมูลบิตที่ 9 เข้ามา แต่เมื่อ SM2 เป็น 1 แล้ว RI จะเป็น 1 ก็ต่อเมื่อ ข้อมูล บิตที่ 9 ที่เข้ามามีค่าเป็น 1 ถ้าข้อมูลบิตที่ 9 ที่เข้ามามีค่าเป็น 0 จะไม่ทำให้บิต RI มีค่าเป็น 1 ใน โหมด 1 บิต RI มีค่าเป็น 1 เมื่อข้อมูล Stop Bit เข้ามายังพอร์ตอนุกรมถูกต้อง แต่ถ้า Stop bit ไม่

เข้ามายังพอร์ตนุกรม อันอาจเกิดจากปัญหาในการส่งข้อมูลแล้ว บิต RI จะมีค่าเป็น 0 ในโหมด 0 บิตนี้จะมีค่าเป็น 0 เสมอ

3.8.7 บิต SM0, SM1 เป็น 2 บิตที่ใช้งานร่วมกันเพื่อกำหนดโหมดของการรับ - ส่งข้อมูลของพอร์ตนุกรม ค่าใน 2 บิตนี้ จะกำหนดโหมดได้ดังนี้

## ตารางที่ 2

แสดงการเลือกโหมดการรับส่งข้อมูลนุกรมของบิต SM1, SM0

SM0	SM1	โหมด	การทำงาน	อัตราการส่งข้อมูล
0	0	0	ซีพรีจิสเตอร์	ความถี่คงที่ = $F_{osc}/12$
0	1	1	8 บิต UART	ความถี่สามารถเปลี่ยนแปลงได้
1	0	2	9 บิต UART	ความถี่คงที่ = $F_{osc}/32$ หรือ $F_{osc}/64$
1	1	3	9 บิต UART	ความถี่สามารถเปลี่ยนแปลงได้

3.9 Timer Register (TH0, TL0, TH1, TL1) ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH, 8AH, 8DH, 8BH ใน 8051 จะมีวงจร Timer อยู่ 2 ชุดคือ Timer 0 และ Timer 1 ใน Timer แต่ละชุดจะมีรีจิสเตอร์ขนาด 8 บิตอยู่ 2 ตัว เพื่อ เก็บค่าการนับได้สูงสุดถึง 16 บิต ใน Timer 0 รีจิสเตอร์นี้คือ TH0, TL0 และใน Timer 1 คือรีจิสเตอร์ TH1, TL1 TLx (x หมายถึง 0 หรือ 1) จะเก็บค่าของการนับ 8 บิตล่าง และ THx จะเก็บค่าของการนับ 8 บิตบน การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิลของสัญญาณนาฬิกาที่ผ่านวงจรหาร 12 เมื่อการนับครบถึงค่าสูงสุดที่รีจิสเตอร์ TLx และ THx จะเก็บได้คือค่า FFFFH แล้วยังนับต่อไป ค่าที่ได้จากการนับจะเป็น 0000H ทำให้เกิดการ Set บิตบางบิตในรีจิสเตอร์ TCON เพื่อบอกสถานะโอเวอร์ โฟลว์ ส่วนการทำงานเป็น Counter ก็คือ การใช้รีจิสเตอร์ THx และ TLx ทำการนับจำนวนไซเคิลของ สัญญาณที่เข้ามาทางขา T0 หรือ T1 ของ 8051 สัญญาณที่เข้ามาทางขา T0 หรือ T1 ซึ่งสัญญาณอาจจะมาจากอุปกรณ์ตรวจจับ (Sensor) แต่สถานะของสัญญาณนี้จะต้องมีระดับโวลเตจของสถานะลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีแรงดัน ไม่เกิน 0.6 โวลท์ และลอจิก 1 จะต้องมีแรงดันมากกว่า 2.4 โวลท์

3.10 Timer/Counter Mode Register (TMOD) ตำแหน่งหน่วยความจำภายในเท่ากับ 89H TMOD เป็นรีจิสเตอร์ขนาด 8 บิต ที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer 1 แต่ละบิตในรีจิสเตอร์มีความหมายเฉพาะดังนี้

### ตารางที่ 3

แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ TMOD

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
GATE	C/T	M1	MO	GATE	C/T	M1	MO

3.10.1 GATE เป็นบิตที่ใช้ควบคุมให้ Timer ทำงานหรือไม่ ถ้าบิตนี้ของ Timer x ถูกตั้งเป็น 1 จะทำให้ Timer ทำงานก็ต่อเมื่อ ที่ขา INTx มีสภาวะลอจิกเป็น 1 และบิต TRx ในรีจิสเตอร์ TCON เป็น 1 ด้วย

3.10.2  $\overline{C/T}$  บิตนี้ใช้สำหรับเลือกการทำงานของ Timerว่าจะใช้เป็น Timer หรือ Counter ถ้าบิตนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา Tx

3.10.3 M1, MO เป็น 2 บิตที่ใช้ร่วมกัน เพื่อเลือกโหมดการทำงานของ Timer การทำงานโหมด 0,1 และ 2 ของ Timer 0 จะเหมือนกับ Timer 1 แต่ในโหมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M1 และ MO จะเลือกโหมดการทำงานดังนี้

### ตารางที่ 4

แสดงการเลือกโหมดโดยบิต M1 และ MO

M1	MO	โหมด	การทำงาน
0	0	0	รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิต ค่าจากการนับ 8 บิตบนมาจาก 8 บิตของ THx และอีก 5 บิตล่างมาจากค่า 5 บิตล่างของรีจิสเตอร์ TLx โดยที่ 3 บิตบนของ TLx จะไม่ต้องสนใจเลย
0	1	1	รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 16 บิต ค่าจากการนับ 8 บิตบนอยู่ในรีจิสเตอร์ THx และค่าจากการนับ 8 บิตล่างอยู่ในรีจิสเตอร์ TLx

**ตารางที่ 4 (ต่อ)**  
เลือกโหมดโดยบิต M1 และ M0

M1	M0	โหมด	การทำงาน
0	0	0	รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิต ค่าจากถารนับ 8 บิตบนมาจาก 8 บิตของ THx และอีก 5 บิตล่างมาจากค่า 5 บิตล่างของรีจิสเตอร์ TLx โดยที่ 3 บิตบนของ TLx จะไม่ต้องสนใจเลย
1	1	3	Timer 1 จะไม่ทำงาน ส่วน Timer 2 จะเสมือนมีตัวนับขนาด 8 บิต 2 ตัวทำงานอยู่

3.11 Timer Control (TCON) ตำแหน่งหน่วยความจำภายในเท่ากับ 088H รีจิสเตอร์ขนาด 8 บิตนี้ใช้ควบคุมการทำงาน และบอกสถานะของ Timer 0 และ Timer 1 แต่ละบิตของรีจิสเตอร์นี้จะทำงานต่างกันดังนี้

**ตารางที่ 5**  
แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ TCON

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
TF1	TR1	TFO	TRO	IE1	IT1	IE0	IT0

3.11.1 IT0 เป็นบิตที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องมาจากสถานะของสัญญาณที่เข้ามาทางขา  $\overline{INT0}$  ถ้า IT0 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสถานะของสัญญาณที่ขา  $\overline{INT0}$  เปลี่ยนจาก 1 เป็น 0 แต่ถ้า IT0 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสถานะของสัญญาณที่ขา  $\overline{INT0}$  เป็น 0

3.11.2 IE0 บิตนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา  $\overline{INT0}$  มีสถานะลอจิกของสัญญาณตามที่กำหนดในบิต IT0 แล้วทำให้เกิดการขัดจังหวะโปรแกรม และเมื่อกระโดดไปทำงานยังโปรแกรมตอบสนองการขัดจังหวะแล้ว จะทำให้บิตนี้กลับเป็น 0

3.11.3 IT1 เป็นบิตที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องมาจากสถานะของสัญญาณที่เข้ามาทางขา  $\overline{INT1}$  ถ้า IT1 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสถานะของ

สัญญาณที่ขา INT1 เปลี่ยนจาก 1 เป็น 0 แต่ถ้า IT1 เป็น 0 จะเกิดการขัดจังหวะ โปรแกรม ถ้าสถานะของสัญญาณที่ขา INT1 เป็น 0

3.11.4 IE1 บิตนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา  $\overline{\text{INT1}}$  มีสถานะลอจิกของสัญญาณตามที่กำหนดในบิต IT1 แล้วทำให้เกิดการขัดจังหวะโปรแกรม และเมื่อกระโดดไปทำงานยังโปรแกรมตอบสนองการขัดจังหวะแล้ว จะทำให้บิตนี้กลับเป็น 0

3.11.5 TR0 Timer 0 Run Control บิตนี้ถ้าเป็น 0 Timer 0 ไม่ทำการนับสัญญาณไม่ว่ากรณีใดทั้งสิ้น แต่ถ้าบิตนี้เป็น 1 จะทำให้ Timer 0 ทำงานโดยขึ้นกับสัญญาณ GATE และ INTO ข้อมูลในบิตนี้จะสามารถเซต เป็น 1 หรือ เคลียร์ เป็น 0

3.11.6 TF0 Timer 0 Overflow flag บิตนี้จะเป็น 1 เมื่อการนับของรีจิสเตอร์ใน Timer 0 (TL0 หรือ TH0 ขึ้นกับโหมดของการทำงาน) เกิด Overflow ขึ้น คือการนับเพิ่มไปจนถึงค่าสูงสุดแล้วนับต่อไป ทำให้ค่าการนับกลับมาเริ่มต้นใหม่ที่ 0 หรือค่า Reload เมื่อ 8051 กระโดดไปทำงานยังโปรแกรมตอบสนองการขัดจังหวะจะทำให้บิตนี้กลับเป็น 0

3.11.7 TR1 Timer 1 Run Control Bit การทำงานจะเหมือนกับการทำงานของบิต TR0 แต่บิตนี้จะทำงานกับ Timer 1

3.11.8 TF1 Timer 1 Overflow flag บิตนี้เหมือนกับบิต TF0 ต่างกันที่ขึ้นกับการทำงานของ Timer 1

3.12 Interrupt Enable Register (IE) ตำแหน่งหน่วยความจำภายในเท่ากับ 0A8H รีจิสเตอร์นี้ ใช้สำหรับการยอม หรือไม่ยอม ให้เกิดการอินเทอร์รัพท์จากแหล่งต่าง ๆ ได้ 6 ชนิด โดยแหล่งกำเนิดสัญญาณ ทั้งหมดสามารถทำให้เกิดการขัดจังหวะได้ 5 แบบดังนี้

#### ตารางที่ 6

แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ IE

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
EA	X	ET2	ES	ET1	EX1	ETO	EXO

ถ้าต้องการ Enable บิตใด ก็ให้โปรแกรมกำหนดค่าในบิตนั้นเป็น 1 ถ้าค่าในบิตนั้นเป็น 0 หมายถึง Disable ซึ่งจะทำให้ ไม่มีการขัดจังหวะการทำงานของโปรแกรม เนื่องจากสัญญาณขอขัดจังหวะนั้น ๆ

3.12.1 EX0 บิตนี้จะใช้ Enable หรือ Disable สัญญาณที่เข้ามาทางขา INTO ให้เกิดการขัดจังหวะหรือไม่

3.12.2 ET0 Timer 0 Interrupt Enable Bit บิตนี้จะใช้ Enable หรือ Disable สัญญาณ ขัดจังหวะจาก Timer 0 (TF0)

3.12.3 EX1 บิตนี้จะใช้ Enable หรือ Disable สัญญาณที่เข้ามาทางขา INT1 ให้เกิดการขัดจังหวะหรือไม่

3.12.4 ET1 Timer 1 Interrupt Enable Bit บิตนี้จะใช้ Enable หรือ Disable สัญญาณขัดจังหวะจาก Timer 1 (TF1)

3.12.5 ES ข้อมูลในบิตนี้จะ Disable หรือ Enable การขัดจังหวะจากพอร์ตนุกรม อันเนื่องมาจากมีข้อมูลเข้ามายัง SBUF หรือข้อมูลจาก SBUF ได้ส่งออกผ่านทาง Serial Port หมดแล้ว

3.12.6 ET2 Timer 2 Internal Enable bit จะใช้งานเฉพาะใน 8052 และ 83152 เท่านั้น บิตนี้จะใช้ Enable หรือ Disable สัญญาณขอการขัดจังหวะที่มาจาก Timer 2

3.12.7 EA บิตนี้จะควบคุมทั้ง 6 บิตที่กล่าวมาแล้ว ถ้าข้อมูลในบิตนี้เป็น 0 จะเป็นการ Disable ทุกบิตที่กล่าวมาแล้วซึ่งจะ ไม่มีการขัดจังหวะโปรแกรมได้เลย แต่ถ้าบิตนี้เป็น 1 การอินเทอร์รัพท์จะเกิดขึ้นหรือไม่ จะขึ้นอยู่กับข้อมูลในแต่ละบิตนั้น

3.13 IP Interrupt (IP) ตำแหน่งหน่วยความจำภายในเท่ากับ 0B8H รีจิสเตอร์ IP จะทำหน้าที่ในการจัดลำดับความสำคัญของการอินเทอร์รัพท์ เมื่อมีการอินเทอร์รัพท์เข้ามาพร้อมกัน โดย แต่ละ บิตของรีจิสเตอร์ IP นั้น จะบอกลำดับความสำคัญของแหล่งกำเนิดสัญญาณขัดจังหวะดังนี้

### ตารางที่ 7

แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ IP

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
X	X	PT2	PS	PT1	PX1	PT0	PX0

3.13.1 PX0 บิต 0 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจากภายนอก 8051 คือ INTO

3.13.2 PT0 บิต 1 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 0

3.13.3 PX1 บิต 2 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจากภายนอก 8051 คือ INT1

3.13.4 PT1 บิต 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 1

3.13.5 PT2 บิต 5 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 2 บิตนี้ได้เฉพาะใน 8052 ที่มี Timer 2

3.13.6 PS บิต 4 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Serial Port ในกรณีที่มีข้อมูลเข้ามา หรือส่งข้อมูล ออกสิ้นสุดแล้ว

3.14 Power Control (PCON) ตำแหน่งหน่วยความจำภายในเท่ากับ 87H

รีจิสเตอร์ PCON ทำหน้าที่ในการควบคุมการใช้พลังงานของไมโครคอนโทรลเลอร์ และหน้าที่อื่น ๆ ดังนี้

### ตารางที่ 8

แสดงรายละเอียดการกำหนดค่าของรีจิสเตอร์ PCON

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SMOD	X	X	X	GF1	GF0	PD	IDL

3.14.1 IDL ถ้าบิตนี้ถูกกำหนดให้เป็น 1 MCS51 จะเข้าสู่การทำงานใน Idle Mode ทันที

3.14.2 PD ถ้าบิตนี้ถูกกำหนดให้เป็น 1 MCS51 จะเข้าสู่การทำงานใน Power Down Mode ทันที

3.14.3 GF0, GF1 เป็น General Purpose Flag Bit บิตนี้สามารถกำหนดให้มีค่าเป็น 0 หรือ 1 ได้โดยโปรแกรม เพื่อส่ง ผ่านสถานะการทำงานของ 8051 ระหว่างแต่ละโปรแกรมย่อย

3.14.4 SMOD เป็นบิตที่ใช้ร่วมในการกำหนดอัตราการส่งข้อมูล (Baud Rate) ผ่านทางพอร์ตอนุกรม ซึ่งในการรับ - ส่งข้อมูล ผ่านทางพอร์ตอนุกรม โหมด 1 และ 3 จะสามารถกำหนดอัตราการส่งข้อมูลได้ตามอัตราการเกิดโอเวอร์โฟลว์ใน Timer 1 ถ้าบิตนี้เป็น 1 จะทำให้อัตราการส่งข้อมูลเพิ่มขึ้น 2 เท่า

## โมดูลแสดงผล LCD รุ่น LM6061SYR-2

ส่วนแสดงผล LCD รุ่น LM601SYR-2 ซึ่งมีขนาด 240 x 60 จุด สามารถแสดงผลได้ 2 ลักษณะ คือการแสดงในโหมดตัวอักษร และการแสดงผลในโหมดกราฟฟิก

1 รายละเอียดขาสัญญาณ การใช้งานและความหมายของขาสัญญาณ แสดงดังตารางที่ 9

**ตารางที่ 9**  
แสดงรายละเอียดขาสัญญาณในการเชื่อมต่อ LCM

ขา	สัญลักษณ์	หน้าที่การทำงาน
1	VSS	ขาราวนด์ ( 0 โวลท์ )
2	VDD	ขาไฟเลี้ยงวงจร ( 5 โวลท์ )
3	RS	ขาสัญญาณเลือกรีจิสเตอร์ ลอจิก 0 - รีจิสเตอร์ข้อมูล ลอจิก 1 - รีจิสเตอร์คำสั่ง
4	RW	ขาสัญญาณควบคุมการอ่าน หรือ เขียนข้อมูลกับ LCM
5	E	ขาสัญญาณยอมให้มีการติดต่อกับ LCM
6	DB0	ขาสัญญาณข้อมูลบิต 0
7	DB1	ขาสัญญาณข้อมูลบิต 1
8	DB2	ขาสัญญาณข้อมูลบิต 2
9	DB3	ขาสัญญาณข้อมูลบิต 3
10	DB4	ขาสัญญาณข้อมูลบิต 4
11	DB5	ขาสัญญาณข้อมูลบิต 5
12	DB6	ขาสัญญาณข้อมูลบิต 6
13	DB7	ขาสัญญาณข้อมูลบิต 7
14	CS	ขาสัญญาณเลือกชิพของ LCM
15	RES	ขาสัญญาณรีเซท

2.2 คำสั่งควบคุมการแสดงผล การแสดงผลบนโหมดดู LCD จะทำงานโดยการเขียนข้อมูลลงในรีจิสเตอร์ คำสั่ง ( Instruction Register ) และรีจิสเตอร์ ข้อมูล ( Data Register ) โดยใช้สัญญาณ RS แบ่งแยกระหว่างการติดต่อกับรีจิสเตอร์ทั้งสองนี้ ดูจากหัวข้อยละเอียดขาสัญญาณ ในการทำงานของโหมดดู LCD นั้นจะแสดงสภาวะการทำงานทางแฟล็ก Busy ถ้าแฟล็กนี้เป็น 1 แสดงว่ากำลังทำงานคำสั่งอยู่ ซึ่งจะไม่สามารถเขียนข้อมูลมายังโหมดดู LCD จะต้องรอจนกว่าแฟล็กนี้จะ เป็น 0 ก่อนจึงจะสามารถติดต่อกับโหมดดูได้

2.2.1 คำสั่งโหมดควบคุม เมื่อมีการเขียนข้อมูล 00H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่โหมดควบคุม โดยมีรายละเอียดการทำงานดังภาพที่ 9

ภาพที่ 9  
รายละเอียดคำสั่งย่อยของโหมดควบคุม

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	0	0	0	0
ควบคุมโหมด	0	0	0	0	ข้อมูลเลือกโหมด					

DB5	DB4	DB3	DB2	DB1	DB0	เคอร์เซอร์/กระพริบ	CG	โหมดกราฟฟิก/ตัวอักษร	
1/0	1/0	0	0	0	0	เคอร์เซอร์ OFF	ภายใน	โหมดตัวอักษร	
						เคอร์เซอร์ ON			
						เคอร์เซอร์ OFF ตัวอักษรกระพริบ			
						เคอร์เซอร์ กระพริบ			
		1	0	0	1	เคอร์เซอร์ OFF	ภายนอก		
						เคอร์เซอร์ ON			
						เคอร์เซอร์ OFF ตัวอักษรกระพริบ			
						เคอร์เซอร์ กระพริบ			
		0	0	1	0	<del>XXXXXXXXXX</del>			โหมดกราฟฟิก
สถานะแสดงผล ON/OFF Master/Slave	การกระพริบ	เคอร์เซอร์	กราฟฟิกโหมด	CGภายใน	ภายนอก				

1: Master

0: Slave

1: ส่วนแสดงผล ON

0: ส่วนแสดงผล OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 คำสั่งกำหนดขนาดตัวอักษร เมื่อมีการเขียนข้อมูล 01H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การกำหนดขนาดตัวอักษร โดยมีรายละเอียดการกำหนด คำดังนี้

### ตารางที่ 10

แสดงการกำหนดขนาดตัวอักษรของจอ LCD

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	1	0	0	0	0	0	0	0
Char Pitch	0	0	(Vp-1) Binary			0	(Hp-1) Binary			

Vp แสดงจำนวนของ จุดแนวตั้งต่อหนึ่งตัวอักษร ( ความสูงของตัวอักษร ) ไม่รวมช่องว่างระหว่างตัวอักษรที่เรียงในแนว ตั้ง ค่าของ Vp นี้จะใช้ในการแสดงผลโหมดตัวอักษรเท่านั้น ในโหมดกราฟฟิกจะไม่สนใจค่านี้

Hp แสดงจำนวนของ จุดแนวนอนต่อหนึ่งตัวอักษร (ความกว้างของตัวอักษร) ซึ่งรวมช่องว่างระหว่างตัวอักษรที่เรียงกัน ตามแนวนอนด้วย ในโหมดกราฟฟิกจะหมายถึงจำนวนบิตที่ใช้ต่อข้อมูลหนึ่งตัว โดยค่าของ Hp จะสามารถกำหนดได้ 3 คำดังนี้

### ตารางที่ 11

แสดงการกำหนดจุดแนวนอนต่อหนึ่งตัวอักษร

Hp	DB2	DB1	DB0	ขนาดตัวอักษร (ความกว้าง)
6	1	0	1	6
7	1	1	0	7
8	1	1	1	8

2.2.3 คำสั่งกำหนดจำนวนตัวอักษรในหนึ่งบรรทัด เมื่อมีการเขียนข้อมูล 02H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การกำหนดจำนวนตัวอักษรที่จะแสดงในหนึ่งบรรทัด โดยมีรายละเอียดการกำหนดคำดังนี้

## ตารางที่ 12

แสดงการกำหนดจำนวนของตัวอักษรในหนึ่งบรรทัด

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	1	0	0	0	0	0	0	0
Num of char	0	0	0	(Hn-1) Binary						

Hn ถ้าใช้ในโหมดตัวอักษร หมายถึง จำนวนตัวอักษรในหนึ่งบรรทัด แต่ถ้าในโหมดกราฟฟิก หมายถึงจำนวนไบต์ที่จะใช้ในหนึ่งบรรทัด ดังนั้นจำนวนจุดทั้งหมดที่จะใช้บนจอภาพในหนึ่งบรรทัด สามารถคำนวณได้จากสมการ

$$N = H_p \times H_n$$

Hn เป็นจำนวนคู่ตั้งแต่ 2 - 128

Hp เป็นจำนวนจุดต่อหนึ่งตัวอักษร (ความกว้าง)

2.2.4 คำสั่งกำหนดจำนวนครั้งของการแบ่งเวลา เมื่อมีการเขียนข้อมูล 03H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การกำหนดจำนวนครั้งของการแบ่งเวลาในการแสดงผล โดยมีรายละเอียดการกำหนดค่าดังนี้

## ตารางที่ 13

แสดงการกำหนดจำนวนครั้งของการแบ่งเวลาในการแสดงผล

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	0	1	1	0
Cursor pos	0	0	0	(Nx-1) Binary						

Nx คือ จำนวนของการแบ่งเวลาในการแสดงผลแบบมัลติเพล็กซ์ ( Multiplex ) ค่าของ NX จะอยู่ในช่วง 1 - 128

2.2.5 คำสั่งกำหนดพื้นที่ในการแสดงเคอร์เซอร์ เมื่อมีการเขียนข้อมูล 04H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การกำหนดพื้นที่ในการแสดงเคอร์เซอร์ โดยมีรายละเอียด การกำหนดค่าดังนี้

**ตารางที่ 14**  
**แสดงการกำหนดพื้นที่การแสดงผลเคอร์เซอร์**

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	1	0	0	0	0	0	0	1
Cursor pos	0	0	0	0	0	0	(Cp - 1) Binary			

Cp ถ้าใช้ในโหมดตัวอักษร หมายถึง ตำแหน่งที่จะให้เคอร์เซอร์แสดง ภายในตัวอักษร เช่น ในตัวอักษรขนาด 5 x 7 ถ้า ต้องการแสดงผลเคอร์เซอร์ที่บรรทัดล่างสุด ก็กำหนดค่าของ Cp = 8 ส่วนความยาวของเคอร์เซอร์จะเท่ากับขนาดของตัวอักษร คือค่า ของ Hp ค่าของ Cp สามารถกำหนดอยู่ในช่วงตั้งแต่ 1 - 16 ในกรณีที่ค่าของ Cp < Vp และ เกิดการทับซ้อนกับตัวอักษรขึ้นใหม่ LCD จะถือว่าการแสดงผลเคอร์เซอร์มีความสำคัญสูงกว่า ถ้าค่าของ Cp > Vp จะไม่สามารถแสดงผลเคอร์เซอร์ให้เห็นได้

2.2.6 คำสั่งกำหนดตำแหน่งเริ่มต้นของการแสดงผลไบต์ต่ำ เมื่อมีการเขียนข้อมูล 08H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การกำหนดตำแหน่งเริ่มต้นของการแสดงผลไบต์ต่ำ โดยมีรายละเอียดการกำหนดค่าดังนี้

**ตารางที่ 15**  
**แสดงการกำหนดตำแหน่งเริ่มต้นการแสดงผลไบต์ต่ำ**

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	0	0	0
display start address	0	0	แอดเดรสเริ่มต้น (ไบต์ล่าง)							

2.2.7 คำสั่งกำหนดตำแหน่งเริ่มต้นของการแสดงผลไบต์สูง เมื่อมีการเขียนข้อมูล 09H ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การกำหนดตำแหน่งเริ่มต้นของการแสดงผลไบต์สูง โดยมีรายละเอียดการกำหนดค่าดังนี้

### ตารางที่ 16

แสดงการกำหนดตำแหน่งเริ่มต้นการแสดงผลไบต์สูง

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	0	0	1
display start address	0	0	แอดเดรสเริ่มต้น (ไบต์บน)							

คำสั่งทั้งสองนี้จะเป็นการกำหนดข้อมูลลงใน รีจิสเตอร์ตำแหน่งเริ่มต้น ( Start Address Register ) ซึ่งตำแหน่งเริ่มต้น ของการแสดงผลก็คือ ตำแหน่งของแรมแสดงผล (Display RAM) ซึ่งจะเริ่มต้นแสดงผลที่ด้านบนซ้ายของจอแสดงผล ในโหมด กราฟฟิกจะใช้ข้อมูลทั้ง 16 บิต แต่ในกรณีโหมดตัวอักษรจะใช้เพียง 12 บิตเท่านั้น โดยข้อมูล 4 บิตบนของไบต์สูงจะไม่ใช้งาน

2.2.8 คำสั่งกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์ต่ำ เมื่อมีการเขียนข้อมูล 0AH ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสูการกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์ต่ำ โดยมีรายละเอียดการกำหนดค่าดังนี้

### ตารางที่ 17

แสดงการกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์ต่ำ

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	0	0	0
display start address	0	0	แอดเดรสเริ่มต้น (ไบต์ล่าง)							

2.2.9 คำสั่งกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์สูงเมื่อมีการเขียนข้อมูล 0BH ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสูการกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์สูง โดยมีรายละเอียดการกำหนดค่าดังนี้

### ตารางที่ 18

แสดงการกำหนดตำแหน่งเคอร์เซอร์บนจอแสดงผลไบต์สูง

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	0	1	1
Cursor add counter	0	0	แอดเดรสเคอร์เซอร์ (ไบต์บน)							

คำสั่งทั้งสองนี้เป็นการกำหนดตำแหน่งเคอร์เซอร์ลงใน ตัวนับตำแหน่งเคอร์เซอร์ (Cursor Address Counter) ตำแหน่ง เคอร์เซอร์จะบอกถึงตำแหน่งของ แรมแสดงผล ที่จะเขียนหรืออ่านข้อมูล และในโหมดตัวอักษรก็จะมีการแสดงเคอร์เซอร์ที่ตำแหน่ง นั้นๆด้วย

การกำหนดตำแหน่งเคอร์เซอร์มีขั้นตอนดังนี้

2.2.9.1 เมื่อต้องการเปลี่ยนค่าของตำแหน่งเคอร์เซอร์ใหม่ ให้ทำการเขียนข้อมูลตำแหน่งไบต์ต่ำก่อนแล้วตามด้วยการเขียน ข้อมูลตำแหน่งไบต์สูง

2.2.9.2 เมื่อต้องการเปลี่ยนค่าของตำแหน่งไบต์ต่ำเท่านั้น ให้ทำการเขียนข้อมูลตำแหน่งไบต์ต่ำก่อนแล้วตามด้วยการเขียน ข้อมูลตำแหน่งไบต์สูง

2.2.9.3 เมื่อต้องการเปลี่ยนค่าของตำแหน่งไบต์สูงเท่านั้น ให้เขียนข้อมูลตำแหน่งไบต์สูงเพียงอย่างเดียว โดยไม่จำเป็นต้อง เขียนข้อมูลตำแหน่งไบต์ต่ำ

2.2.10 คำสั่งการเขียนข้อมูลเพื่อแสดงผล เมื่อมีการเขียนข้อมูล 0CH ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การเขียนข้อมูลเพื่อแสดงผล โดยมีรายละเอียดการกำหนด ค่าดังนี้

### ตารางที่ 19

แสดงการเขียนข้อมูลเพื่อแสดงผล

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	1	0	1
RAM	1	1	ข้อมูลรหัสตัวอักษรหรือข้อมูลแพทเทิร์น							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ถูกเขียนมายังรีจิสเตอร์ข้อมูลภายหลังจากการเขียนข้อมูล OCH ไปยังรีจิสเตอร์คำสั่ง จะถูกเก็บลงในแรมแสดงผล ที่ตำแหน่งเคอร์เซอร์ปัจจุบัน หลังจากทำงานแสดงผลแล้ว ตำแหน่งเคอร์เซอร์จะเพิ่มขึ้นอีก 1 ตำแหน่ง

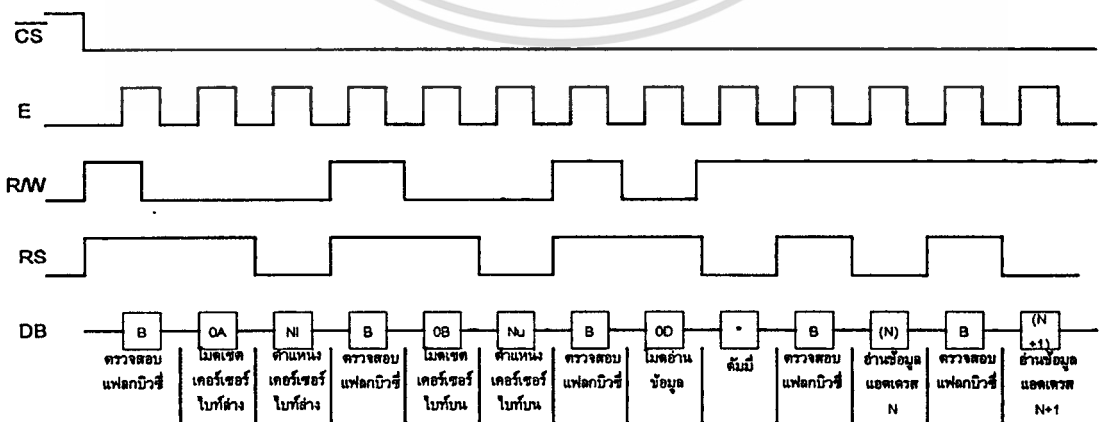
2.2.11 คำสั่งการอ่านข้อมูลที่ตำแหน่งเคอร์เซอร์ เมื่อมีการเขียนข้อมูล ODH ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเข้าสู่การอ่านข้อมูลที่ตำแหน่งเคอร์เซอร์ โดยมีรายละเอียดดังนี้

ตารางที่ 20  
แสดงการอ่านข้อมูลที่ตำแหน่งเคอร์เซอร์

รีจิสเตอร์	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	1	0	1
RAM	1	1	ข้อมูลรหัสตัวอักษรหรือข้อมูลแพทเทิร์น							

ภายหลังจากการเขียนข้อมูล ODH ไปที่รีจิสเตอร์คำสั่งแล้ว ข้อมูลจะถูกอ่านมาจากแรมแสดงผล ซึ่งขั้นตอน การอ่านข้อมูลจะเป็นดังภาพที่ 10

ภาพที่ 10  
แสดงขั้นตอนการอ่านข้อมูลที่ตำแหน่งเคอร์เซอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการอ่านข้อมูล จะเริ่มจากการกำหนดตำแหน่งเคอร์เซอร์ หลังจากนั้นจะต้องทำการอ่านข้อมูล 2 ครั้ง ซึ่งการอ่าน ข้อมูลครั้งแรกจะไม่มีการนำข้อมูลไปใช้งาน ส่วนการอ่านข้อมูลครั้งที่สองจึงจะได้ข้อมูลที่ถูกต้องเพื่อนำไปใช้งาน ดังนั้นถ้ามีการ กำหนดตำแหน่งเคอร์เซอร์ใหม่ จะต้องมีการอ่านข้อมูลสองครั้งเสมอ แต่ถ้าไม่มีการกำหนดตำแหน่งเคอร์เซอร์ใหม่ ก็สามารถอ่าน ข้อมูลเพียงครั้งเดียว เพื่อนำไปใช้งานได้

คำสั่งนี้จะให้ผลลัพธ์เป็นข้อมูลที่แสดงผลอยู่ตรงตำแหน่งเคอร์เซอร์ปัจจุบันลงบนดาต้าบัส (DB0-DB7) พร้อมกับเพิ่มค่า เคอร์เซอร์ขึ้นอีก 1 ตำแหน่ง

2.2.12 คำสั่งเคลียร์บิตข้อมูลของการแสดงผล เมื่อมีการเขียนข้อมูล 0EH ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเคลียร์บิตภายในข้อมูลที่จะนำไปใช้สำหรับการแสดงผล รายละเอียดของการเคลียร์บิตข้อมูลแสดงดังนี้

ตารางที่ 21  
แสดงการเคลียร์บิตข้อมูลของการแสดงผล

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	1	0	1
Bit clear	0	0	0	0	0	0	0	(Nb-1) Binary		

2.2.13 คำสั่งเซตบิตข้อมูลของการแสดงผล เมื่อมีการเขียนข้อมูล 0FH ไปที่รีจิสเตอร์คำสั่ง จะเป็นการเซตบิตภายในข้อมูลที่จะนำไปใช้สำหรับการแสดงผล รายละเอียดของการเซตบิตข้อมูลแสดงดังนี้

ตารางที่ 22  
แสดงการเซตบิตข้อมูลของการแสดงผล

รีจิสเตอร์	R/W	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
คำสั่ง	0	1	0	0	0	0	1	1	1	1
Bit set	0	0	0	0	0	0	0	(Nb-1) Binary		

คำสั่งเคลียร์บิต และเซตบิตข้อมูลนี้ จะใช้ในการกำหนดบิตต่าง ๆ ภายในไบต์ข้อมูลของ แรมแสดงผล ให้เป็น 0 หรือ 1 ตามลำดับ ตำแหน่งของบิตจะกำหนดโดย Nb และตำแหน่งของ ไบต์จะกำหนดโดยเคอร์เซอร์ปัจจุบัน หลังจากทำงานคำสั่งเคลียร์ หรือเซตบิตแล้ว ตำแหน่งของ เคอร์เซอร์จะเพิ่มขึ้น 1 ตำแหน่ง

ค่า Nb จะอยู่ในช่วง 1 - 8 โดย

Nb = 1 หมายถึง บิตน้อยสำคัญต่ำสุด (LSB)

Nb = 8 หมายถึง บิตน้อยสำคัญสูงสุด (MSB)

2.2.14 คำสั่งอ่านแฟล็กบิวซี (Busy Flag) เมื่อส่งสัญญาณ RW = 1 และ RS = 1 ไปยังโหมดูลแสดงผล LCD จะเป็นการอ่านสถานะของการทำงาน โดย แฟล็กบิวซี จะถูกส่งออกที่บิต 7 ของดาต้าบัส (DB7)

### ตารางที่ 23

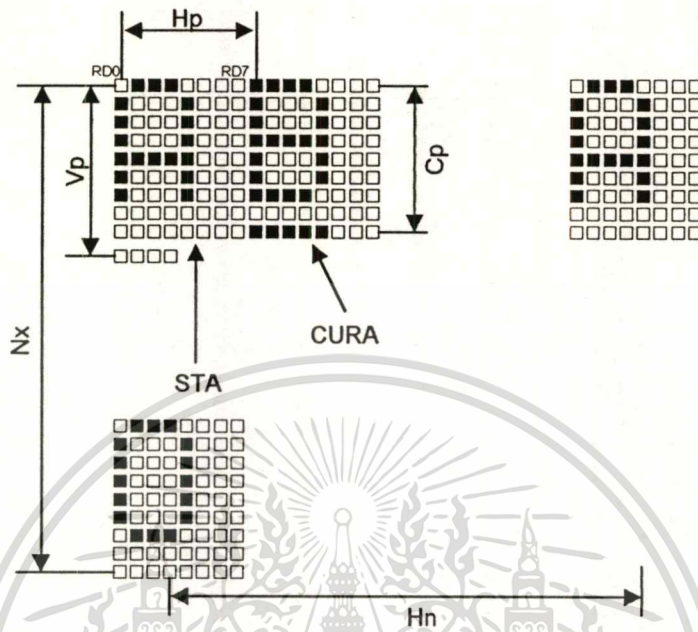
#### แสดงการอ่านแฟล็กบิวซี (Busy Flag)

รีจิสเตอร์	RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
แฟล็ก Busy	1	1	1/0				****			

แฟล็กบิวซีจะเซตเป็น 1 เมื่อมีการทำงานคำสั่งในข้อ 1-13 หลังจากทำงานเสร็จ แฟล็กบิวซีจะถูกเคลียร์ค่าเป็น 0 ขณะที่ แฟล็กบิวซีเซตเป็น 1 จะไม่สามารถทำงานคำสั่งอื่น ๆ ได้ ดังนั้นก่อนที่จะทำการติดต่อกับโหมดูล LCD ต้องแน่ใจว่าแฟล็กบิวซีมีค่า เป็น 0 ยกเว้นในกรณีที่เขียนข้อมูลไปยังรีจิสเตอร์คำสั่ง ก็ไม่จำเป็นต้องมีการตรวจสอบค่าของแฟล็กบิวซี

### ภาพที่ 11

แสดงสัญลักษณ์ที่ใช้ในการกำหนดค่าต่าง ๆ



### ตารางที่ 24

แสดงความหมายของสัญลักษณ์ที่ใช้ในการแสดงผลบนจอ LCD

สัญลักษณ์	ชื่อ	ความหมาย	ค่า
$H_p$	Horizontal Character Pitch	ความกว้างของตัวอักษร	6 - 8 จุด
$H_n$	Number of Horizontal Character	โหมดตัวอักษร	จำนวนตัวอักษร/บรรทัด
		โหมดกราฟฟิก	2 - 128
			จำนวนไบต์/บรรทัด
$V_p$	Vertical Character Pitch	ความสูงตัวอักษร	เฉพาะเลขคู่ 1 - 16 จุด
$C_p$	Cursor Position	หมายเลขบรรทัดที่จะแสดงเคอร์เซอร์	1 - 16 บรรทัด
$N_x$	Number of Time Division	จำนวนของการแบ่งเวลาในการแสดงผล	1 - 128 บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ข้อสังเกต

ถ้าให้จำนวนจุดในแนวตั้งเท่ากับ  $m$  และจำนวนจุดในแนวนอนเท่ากับ  $n$  จะได้ความสัมพันธ์ดังนี้

$$1/m = 1/Nx = \text{อัตราความถี่ของการแสดงผล}$$

$$n = H_p \times H_n, m/V_p = \text{จำนวนบรรทัดที่สามารถแสดงตัวอักษร}$$

$$C_p \leq V_p$$

ภาพที่ 12

แสดงวิธีการแสดงผลข้อมูลในหน่วยความจำในโหมดตัวอักษร และโหมดกราฟฟิก

โหมดการแสดงผล	ข้อมูล	RAM	หน้าจอ LCD																								
โหมดตัวอักษร	รหัสแอสกี	แอดเดรส เริ่มต้น → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	B7	B6	B5	B4	B3	B2	B1	B0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<p>Hp : 6, 7 หรือ 8 จุด</p>
B7	B6	B5	B4	B3	B2	B1	B0																				
0	0	0	0	0	0	0	0																				
0	0	0	0	0	0	0	0																				
โหมดกราฟฟิก	แพทเทิร์นตัวอักษร	แอดเดรส เริ่มต้น → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	B7	B6	B5	B4	B3	B2	B1	B0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<p>Hp : 8 จุด</p>
B7	B6	B5	B4	B3	B2	B1	B0																				
0	0	0	0	0	0	0	0																				
0	0	0	0	0	0	0	0																				

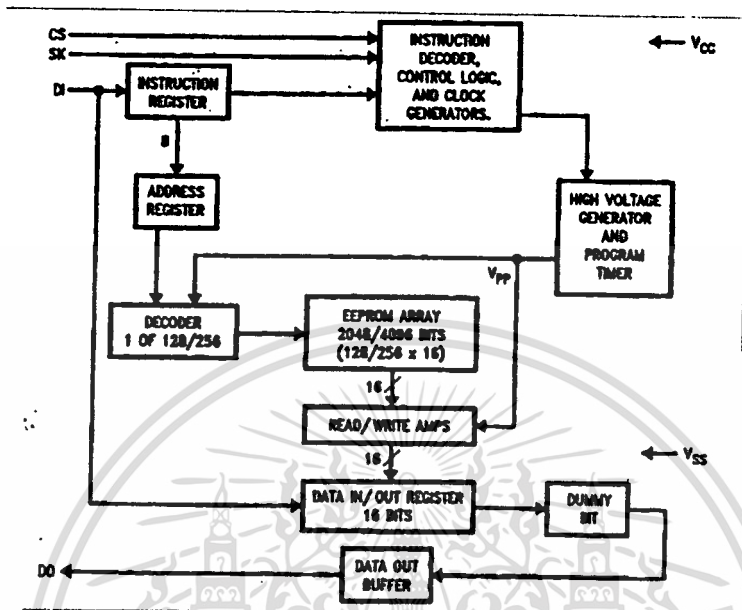
### หน่วยความจำ EEPROM 93C46

1. ข้อมูลทั่วไป ไอซี 93C46 เป็นชิพหน่วยความจำแบบ EEPROM ขนาด 1024 บิต โดยจะแบ่งพื้นที่ ในแต่ละตำแหน่งเป็น 16 บิต ซึ่งจะ ทำงานที่ระดับไฟเลี้ยง 5 โวลท์ และใช้การติดต่อในลักษณะอนุกรม 3 ขาสัญญาณ (MICROWIRE) คือ DI (Data Input) DO (Data output) และ SK (Shift clock) คำสั่งในการติดต่อกับ 93C46 นี้จะมี 7 ชุดคำสั่ง ในการอ่าน เขียน และ ลบข้อมูลภายในหน่วย ความจำ และ สำหรับ 93C46 นี้สามารถเขียนข้อมูลโดยไม่ต้องลบข้อมูลเดิมก่อน การเขียน หรือการลบข้อมูล ภายในหน่วย ความจำจะมีขาสัญญาณ DO เป็นขาที่บอกสถานะ READY/BUSY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 13

แสดงบล็อกไดอะแกรมและการจัดเรียงขาสัญญาณของ 93C46



## 2. คุณสมบัติเฉพาะ สามารถแบ่งเป็นดังนี้

2.1 ขณะทำงานจะกินกระแสไฟ 400 ไมโครแอมป์ ส่วนเข้าสู่สถานะที่ไม่ทำงานจะกินกระแสเพียง 25 ไมโครแอมป์

2.2 การติดต่อกับระบบอื่น ๆ จะเป็นแบบอนุกรม 3 สายสัญญาณ

2.3 สามารถรักษาข้อมูลได้นานถึง 10 ปี

2.4 สามารถเขียนข้อมูลได้ถึง 40000 ครั้ง

3. การทำงานคำสั่ง 93C46 นั้นจะมีคำสั่งในการทำงานอยู่ 7 คำสั่ง โดยบิตแรกสุด (MSB) ของทุกๆคำสั่ง จะต้องเป็น " 1 " เสมอ แล้วตามด้วย ออปโค้ด 2 บิต จากนั้น จะตามด้วยแอดเดรส 6 บิต และข้อมูล 16 บิต

3.1 คำสั่งยอมให้มีการเขียนและลบข้อมูล (Erase/Write Enable ,EWEN) คำสั่ง EWEN จะใช้เพื่อยอมให้ ชิพสามารถถูกเขียนหรือ ลบข้อมูลภายในได้ ดังนั้น เมื่อเริ่มจ่ายไฟเลี้ยงให้กับชิพ จำเป็นที่ จะต้องมีการเขียนคำสั่งนี้ก่อนที่จะทำงานคำสั่งที่เกี่ยวกับการเขียน หรือ การลบข้อมูล สถานะของคำสั่งนี้จะคงอยู่ จนกระทั่งมีการ ทำงานคำสั่ง EWDS หรือ หยุดจ่ายไฟเลี้ยงให้กับชิพ

3.2 คำสั่งไม่ยอมให้มีการเขียนและลบข้อมูล (Erase/Write Disable, EWDS) คำสั่ง EWDS จะใช้เพื่อป้องกันการเขียนข้อมูลโดยไม่ตั้งใจ ซึ่งเมื่อใช้คำสั่งนี้แล้วจะทำให้คำสั่งที่เกี่ยวกับการลบข้อมูลไม่ทำงานอีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนและลบ จะไม่สามารถทำงานได้จนกว่าจะมีคำสั่ง EWEN ส่วนการทำงานคำสั่ง EWDS จะไม่มีผลกับคำสั่ง READ

3.3 คำสั่งการอ่าน (READ) คำสั่ง READ จะให้ข้อมูลเอาท์พุตมาที่สัญญาณ DO การทำงานการอ่านจะเริ่มต้นเมื่อ ชิพได้รับคำสั่ง READ จากนั้น แอดเดรสจะถูกถอดรหัส และข้อมูลจะถูกปล่อยออกมาเมื่อมีการเปลี่ยนแปลงสัญญาณ SK จาก " 0 " เป็น "1" โดยก่อนที่จะปล่อยข้อมูลภายในหน่วยความจำ จะมีการส่งข้อมูล "0" ออกมาหลังจากที่ป้อนแอดเดรสจนครบให้กับ ชิพ

3.4 คำสั่งการเขียน (WRITE) คำสั่ง WRITE จะเป็นการเขียนข้อมูล 16 บิตลงไปยังตำแหน่งที่ต้องการ หลังจากที่บิตสุดท้ายของข้อมูลถูกใส่เข้าไปที่ขา DI สัญญาณ CS จะต้องเป็น LOW ก่อนขอบขาขึ้นของ สัญญาณ SK และเมื่อ CS เป็น LOW จะทำให้ชิพเริ่มวงจรการโปรแกรม ข้อมูลภายใน โดยแสดงสถานะ READY/BUSY ออกมาที่ขา DO หลังจากนั้นก็สัญญาณขา CS กลับมาเป็น HIGH ได้ 250 นาโน วินาที ถ้า DO เป็น "0" แสดงถึงสถานะที่กำลังเขียนข้อมูลอยู่ และถ้า DO เป็น "1" แสดงถึง สถานะที่การทำงานเสร็จสมบูรณ์แล้ว และสามารถที่จะรับคำสั่งเพื่อทำงานต่อไปได้

3.5 คำสั่งการลบ ( ERASE ) คำสั่ง ERASE นี้จะทำให้ข้อมูลทุกบิตในตำแหน่งที่กำหนดเป็น " 1 " ทั้งหมด หลังจากที่ป้อนคำสั่งและแอดเดรสแล้ว เมื่อ สัญญาณขา CS เป็น LOW ชิพจะเริ่มการลบข้อมูล โดยขาสัญญาณ DO จะแสดงสถานะ READY/BUSY หลังจากนั้นก็สัญญาณ CS กลับเป็น HIGH ได้ 250 นาโนวินาที ถ้า DO เป็น "0" แสดงถึงสถานะที่กำลังเขียนข้อมูลอยู่ และถ้า DO เป็น "1" แสดงถึง สถานะที่ การ ทำงานเสร็จสมบูรณ์แล้ว และสามารถที่จะรับคำสั่งเพื่อทำงานต่อไปได้

3.6 คำสั่งการเขียนทุกตำแหน่ง (Write All, WRAL) คำสั่ง WRAL จะทำการเขียนข้อมูลในหน่วยความจำทุกตำแหน่ง วงจรการทำงานจะเหมือนกับการทำงานคำสั่ง WRITE แต่จะแตกต่างกันเฉพาะข้อนี้ได้

3.7 คำสั่งการลบข้อมูลทุกตำแหน่ง (Erase All ,ERAL) คำสั่ง ERAL จะทำการลบข้อมูลในหน่วยความจำทุกตำแหน่ง (ข้อมูลทุกบิตเป็น "1") วงจรการทำงานจะเหมือนกับการ ทำงาน คำสั่ง ERASE แต่จะแตกต่างกันเฉพาะข้อนี้ได้

ตารางที่ 25  
แสดงชุดคำสั่งของ 93C46

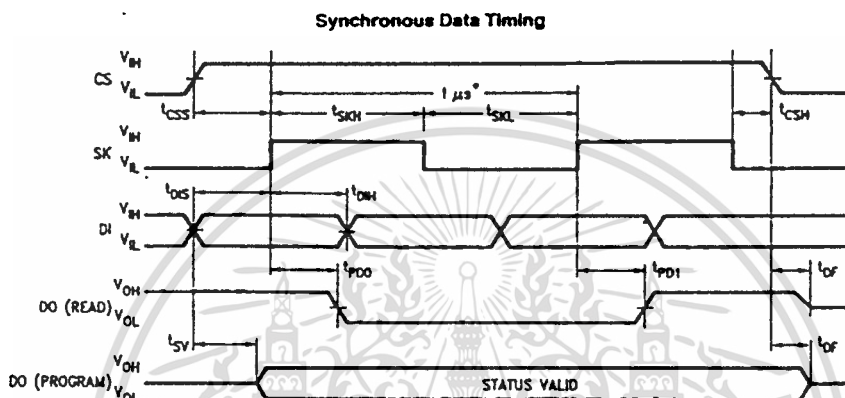
คำสั่ง	บิตเริ่ม	ออปโค้ด	แอดเดรส	ข้อมูล	การทำงาน
READ	1	10	A5-A0		อ่านข้อมูลที่ถูกเก็บในหน่วยความจำตำแหน่งที่ถูกระบุ จาก A5 - A0
EWEN	1	00	11XXXX		ยอมให้มีการเขียนหรือลบข้อมูลในหน่วยความจำได้
ERASE	1	11	A5-A0		ลบข้อมูลในหน่วยความจำตำแหน่งที่ระบุจาก A5 - A0
WRITE	1	01	A5-A0	D15-D0	เขียนข้อมูลลงในตำแหน่งที่ระบุจาก A5 - A0
ERAL	1	00	10XXXX		ลบข้อมูลทุกตำแหน่งของหน่วยความจำ
WRAL	1	00	01XXXX	D15-D0	เขียนข้อมูลลงทุกตำแหน่งของหน่วยความจำ
EWDS	1	00	00XXXX		ไม่ยอมให้มีการเขียนหรือลบข้อมูลในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ไตอะแกรมเวลาการทำงานคำสั่ง

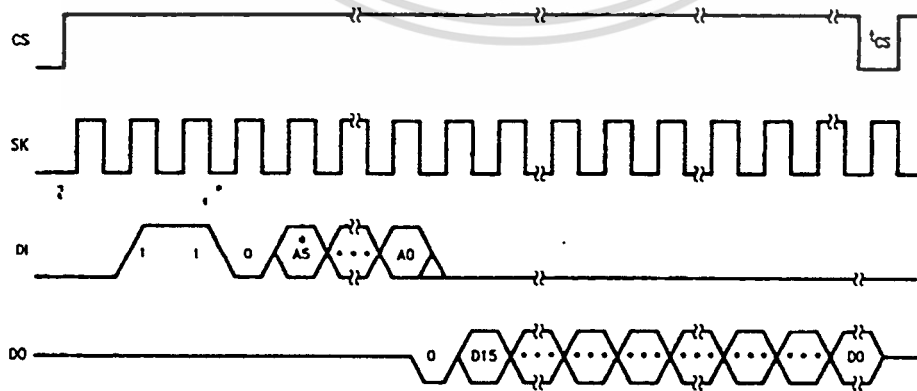
4.1 ไตอะแกรมเวลาทั่วไป

ภาพที่ 14  
แสดงไตอะแกรมเวลาทั่วไป



4.2 ไตอะแกรมเวลาการอ่าน

ภาพที่ 15  
แสดงไตอะแกรมเวลาการอ่าน

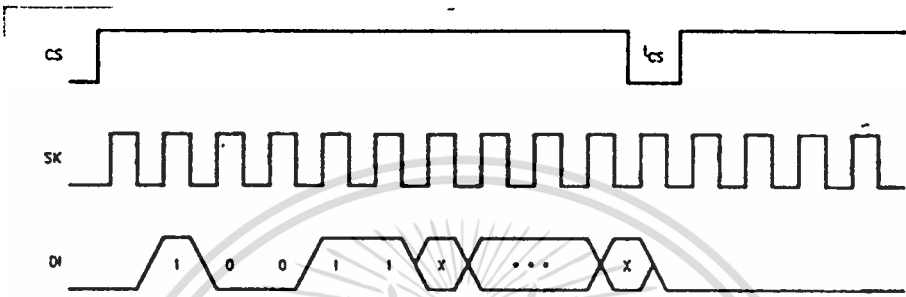


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 ไตอะแกรมเวลาการยอมให้เขียนและลบข้อมูล

ภาพที่ 16

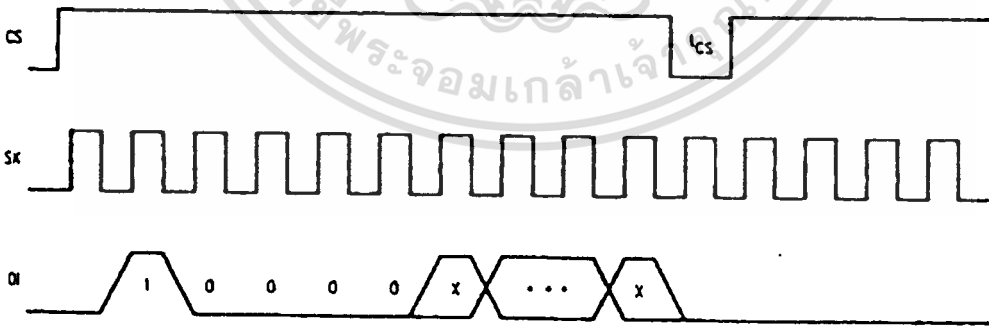
แสดงไตอะแกรมเวลาการยอมให้เขียนและลบข้อมูล



### 4.4 ไตอะแกรมเวลาการไม่ยอมให้เขียนและลบข้อมูล

ภาพที่ 17

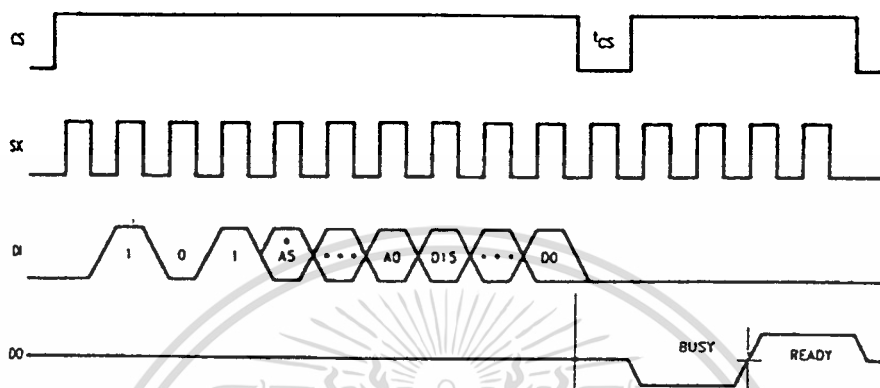
แสดงไตอะแกรมเวลาการไม่ยอมให้เขียนและลบข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

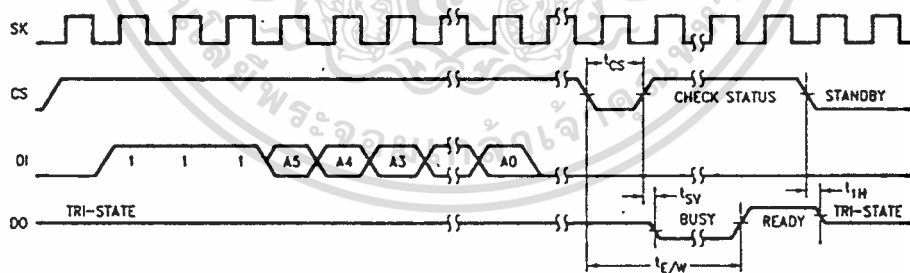
4.5 ไตอะแกรมเวลาการเขียน

ภาพที่ 18  
แสดงไตอะแกรมเวลาการเขียน



4.6 ไตอะแกรมเวลาการลบ

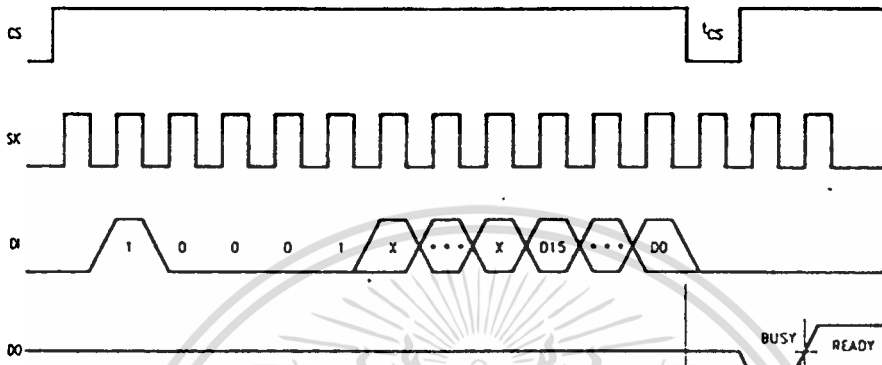
ภาพที่ 19  
แสดงไตอะแกรมเวลาการลบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

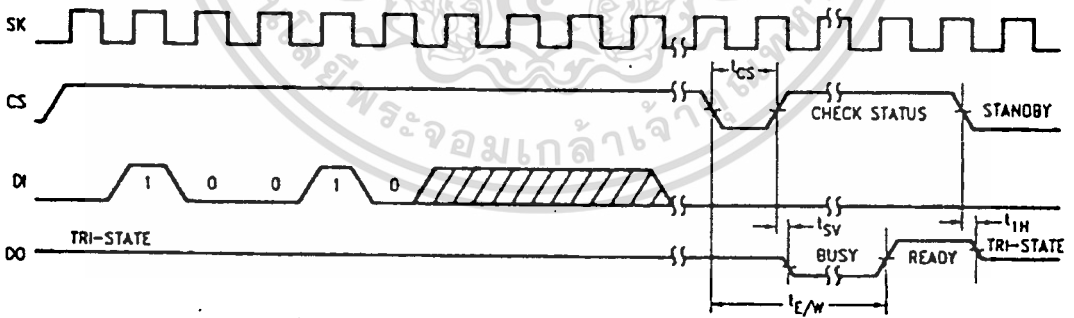
4.7 ไตอะแกรมเวลาการเขียนทั้งหมด

ภาพที่ 20  
แสดงไตอะแกรมเวลาการเขียนทั้งหมด



4.8 ไตอะแกรมเวลาการลบทั้งหมด

ภาพที่ 21  
แสดงไตอะแกรมเวลาการลบทั้งหมด

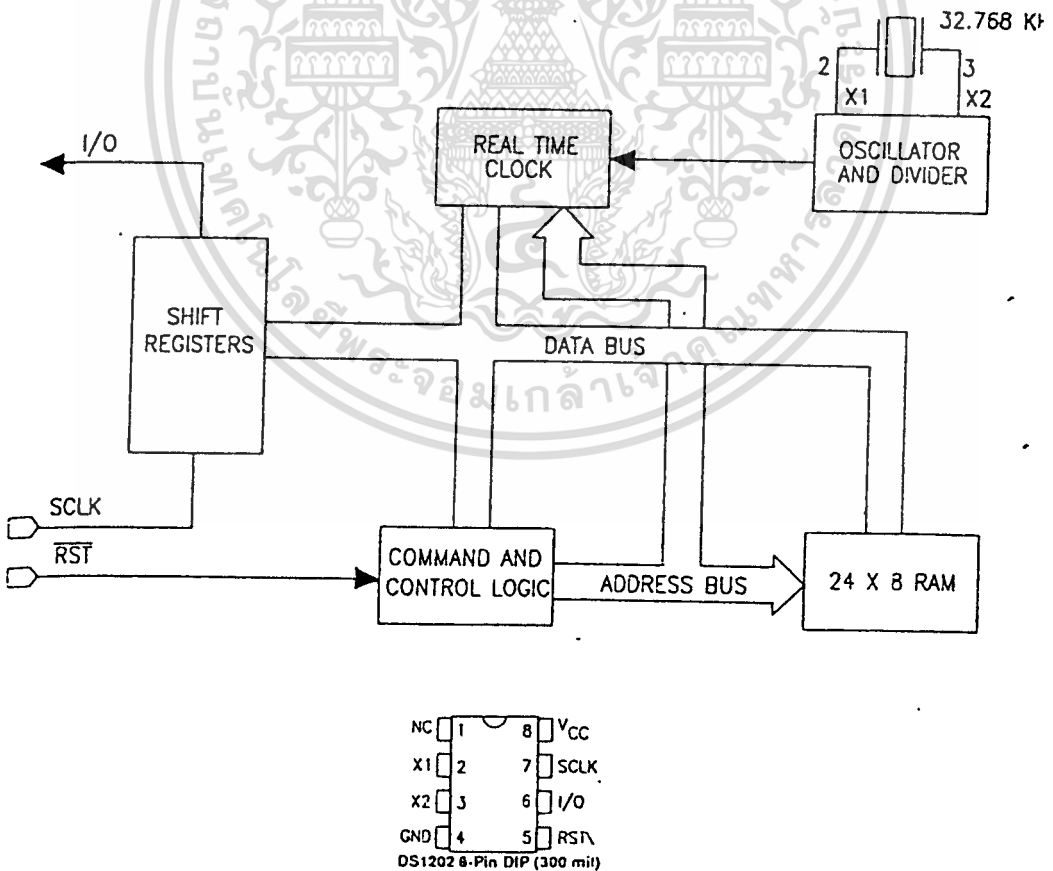


## ชิพรักษาเวลาจริง DS1202

1 ข้อมูลทั่วไป DS1202 เป็นชิพที่ทำหน้าที่นับ เวลา วัน เดือน ปี และเก็บข้อมูลไว้ในรีจิสเตอร์ นอกจากนั้นยังมีส่วนหน่วยความจำ ภายใน ซึ่งสามารถเก็บข้อมูล ได้อีก 24 ไบต์ การติดต่อข้อมูลกับชิพนี้จะทำงานในลักษณะอนุกรมแบบ 3 สายสัญญาณ โดยจะให้ข้อมูล วินาที นาที ชั่วโมง วันของสัปดาห์ วันของเดือน เดือน และปี ในเดือนใดถ้ามีวันน้อยกว่า 31 วัน รวมถึงปีอธิกสุรทิน นั้น จะมีการปรับข้อมูลเองโดยอัตโนมัติ ขาสัญญาณในการเชื่อมต่อจะมี 3 เส้น คือ RST (Reset) I/O (Data Input/Output) SCLK (Serial Clock) ซึ่งมีรูปแบบการติดต่อเป็นสองลักษณะ คือ การติดต่อแบบไบต์เดียว และการติดต่อแบบหลายไบต์

ภาพที่ 22

แสดงบล็อกไดอะแกรมและการจัดเรียงขาสัญญาณของ DS1202



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. คุณสมบัติเฉพาะ แบ่งเป็นส่วนต่างๆดังนี้

2.1 สามารถนับเวลาเป็น วินาที นาที ชั่วโมง วันของสัปดาห์ วันของเดือน เดือน ปี รวมถึงปีอธิกสุรทิน

2.2 มีหน่วยความจำ RAM ขนาด 24 X 8 สำหรับเก็บข้อมูลทั่วไป

2.3 การติดต่อจะใช้ในลักษณะอนุกรม 3 สายสัญญาณ ซึ่งเป็นการประหยัดการเชื่อมต่อภายนอก

2.4 สามารถทำงานที่แรงดัน 2.0 - 5.5 โวลท์

2.5 การทำงานที่แรงดัน 2 โวลท์ จะกินกระแสเพียง 300 นาโนแอมป์

2.6 การติดต่อข้อมูลสามารถกระทำได้ในลักษณะ ไบต์เดียว หรือ หลายไบต์

2.7 สามารถทำงานในอุณหภูมิ - 40 ถึง + 85 องศาเซลเซียส

3. การทำงานภายในของชิพ บล็อกไดอะแกรมของ DS1202 จะประกอบด้วยส่วนต่าง ๆ คือ ชิพ รีจิสเตอร์ ส่วนควบคุมลอจิก ออสซิลเลเตอร์ ฐานเวลาจริง และ หน่วยความจำ RAM การทำงานของชิพจะเริ่มต้นเมื่อ RST เป็น HIGH เมื่อมีสัญญาณ SCLK เข้ามา ข้อมูลที่ขา I/O จะถูกเลื่อนเข้าไปใน ชิพรีจิสเตอร์ ที่ขอบขาขึ้นของ SCLK หลังจากทีเลื่อนข้อมูลครบ 8 บิต ข้อมูลนี้จะถูกตีความเพื่อให้ชิพ พร้อมทั้งจะทำงานต่อ ซึ่งอาจจะเป็นการรับข้อมูล หรือการส่งข้อมูลกับภายนอก

4. ไบต์คำสั่ง ไบต์คำสั่ง จะมีรูปแบบดังตารางที่ 26 ไบต์คำสั่งนี้จะเป็นข้อมูลชุดแรกที่จะถูกส่งเข้าไปใน DS1202 โดยบิต 7 (MSB) จะต้องเป็น "1" เสมอ ถ้าเป็น "0" จะทำให้การทำงานสิ้นสุดทันที บิต 6 ถ้าเป็น "0" จะระบุถึงการติดต่อกับข้อมูลในรีจิสเตอร์ นาฬิกาหรือปฏิทิน แต่ถ้าเป็น "1" จะระบุถึงการติดต่อกับข้อมูลใน RAM ส่วนบิต 1 ถึง บิต 5 จะหมายถึงตำแหน่งของรีจิสเตอร์หรือหน่วยความจำที่ต้องการติดต่อกับ บิต 0 (LSB) ถ้าเป็น "1" จะเป็นการอ่านข้อมูลออกจากชิพ และถ้าเป็น "0" จะเป็นการเขียนข้อมูลลงในชิพ บิตเริ่มต้นที่จะส่งเข้าไปในชิพจะต้องเป็นบิต 0 เสมอ

### ตารางที่ 26

#### แสดงรายละเอียดไบต์คำสั่ง

D7	D6	D5	D4	D3	D2	D1	D0
1	RAM/CK	A4	A3	A2	A1	A0	RD/W

5. โหมดหลายไบต์ การทำงานในโหมดนี้จะต้องกำหนดไบต์คำสั่ง บิต 1 ถึงบิต 5 ให้เป็น "1" ทั้งหมด ส่วนบิต 6 จะเป็นทางเลือก ในการจะติดต่อกับรีจิสเตอร์นาฬิกาปฏิทิน หรือ RAM ถ้าเป็นการติดต่อกับรีจิสเตอร์นาฬิกาปฏิทินแล้วจะต้องป้อน SCLK ทั้งหมด 72 พัลส์ (ไบต์คำสั่ง 8 พัลส์ + ข้อมูล 64 พัลส์) และถ้าเป็นการติดต่อกับ RAM แล้วจะต้องป้อน SCLK ทั้งหมด 200 พัลส์ (ไบต์คำสั่ง 8 พัลส์ + ข้อมูล 192 พัลส์)

6. คำสั่งป้องกันการเขียน ก่อนที่จะมีการเขียนข้อมูลใดๆ ลงในรีจิสเตอร์ หรือ RAM ของ DS1202 จะต้องเคลียร์แฟลกป้องกันการเขียนก่อน โดยการเขียนไบต์คำสั่งเป็น 8EH แล้วส่งข้อมูล 00H ตามเข้าไป แต่ถ้าต้องการป้องกันการเขียนข้อมูลใดๆ ก็ทำโดยการเขียนไบต์คำสั่งเป็น 8EH แล้วส่งข้อมูล 80H ตามเข้าไป

7. แฟลกป้องกันการเขียน แฟลกนี้จะอยู่ที่บิต 7 ของรีจิสเตอร์ป้องกันการเขียน โดยถ้าบิตนี้เป็น "1" จะไม่สามารถเขียนข้อมูลใดๆ ลงในชิพได้ แต่ถ้าบิตนี้เป็น "0" จะสามารถเขียนข้อมูลลงในชิพได้ ส่วนบิตอื่นๆ ของรีจิสเตอร์ป้องกันการเขียนจะเป็น "0" ทั้งหมด

8. การควบคุมสัญญาณ RST และ SCLK การติดต่อระหว่างชิพกับภายนอกจะเริ่มต้นเมื่อ RST เป็น HIGH เพื่อให้ส่วนควบคุมลอจิกภายในยอมให้ข้อมูล ภายนอกสามารถ ส่งเข้ามายังชิพรีจิสเตอร์ได้ และเมื่อ RST เป็น LOW จะเป็นการสิ้นสุดการรับหรือส่งข้อมูล และทำให้ขา I/O อยู่ในสภาวะอิมพีแดนซ์สูง การรับข้อมูลเข้านั้น ข้อมูลจะถูกดึงเข้ามาที่ขอบขาขึ้นของ SCLK ส่วนการส่งข้อมูลออกนั้น ข้อมูลจะถูกปล่อยออกมาที่ขอบขาลงของ SCLK

9. การรับข้อมูลเข้า เมื่อมีการเขียนไบต์คำสั่ง สำหรับการเขียนข้อมูลแล้ว ไบต์ข้อมูลต่อไปจะเริ่มต้นถูกดึงเข้าภายใน ที่ขอบขาขึ้นของ SCLK ซึ่งถ้ามีการป้อนพัลส์ SCLK มากกว่าค่าที่กำหนดแล้วในขณะที่ RST ยังเป็น HIGH จะไม่มีผลกับการเขียนข้อมูลแต่อย่างใด

10. การส่งข้อมูลออก เมื่อมีการเขียนไบต์คำสั่งสำหรับการอ่านข้อมูลแล้ว ไบต์ข้อมูลจะเริ่มต้นถูกส่งออกมาภายนอก ที่ขอบขาลงของ SCLK ซึ่งถ้ามีการป้อนพัลส์ SCLK มากกว่าค่าที่กำหนดแล้วในขณะที่ RST ยังเป็น HIGH จะเกิดการส่งข้อมูลซ้ำเกิดขึ้น

11. รีจิสเตอร์นาฬิกาปฏิทิน รีจิสเตอร์กลุ่มนี้จะมี 8 ตัว ซึ่งข้อมูลที่เก็บอยู่ภายในจะเป็นรูปแบบ BCD

12. แฟลกหยุดนาฬิกา บิต 7 ของรีจิสเตอร์นาฬิกา จะถูกกำหนดไว้เป็น แฟลกหยุดนาฬิกา เมื่อกำหนดให้บิตนี้เป็น "1" ออสซิลเลเตอร์ภายในจะหยุดทำงาน ชิพจะเข้าสู่โหมดพลังงานต่ำ ซึ่งกินกระแสไม่เกิน 100 นาโนแอมป์ และถ้ากำหนดบิตนี้เป็น "0" นาฬิกาจะสามารถทำงานตามปกติ

13. โหมด AM/PM 12 หรือ 24 ชั่วโมง บิต 7 ของรีจิสเตอร์ชั่วโมง ถูกกำหนดให้เป็นทางเลือกในโหมด 12 หรือ 24 ชั่วโมง ถ้ากำหนดบิตนี้เป็น "1" จะทำงานเป็นโหมด 12 ชั่วโมง และบิต 5 จะเป็น บิต AM/PM ถ้าบิต 5 เป็น "1" จะหมายถึง PM และถ้ากำหนดบิต 7 เป็น "0" จะเป็นการเลือกโหมด 24 ชั่วโมง โดยบิต 5 จะเป็นบิต 10 ชั่วโมง

14. หน่วยความจำ RAM หน่วยความจำส่วนนี้จะมีความจุขนาด 24 X 8 ซึ่งสามารถอ่านและเขียนข้อมูลได้สองลักษณะ คือ แบบไบต์เดียว และ แบบ หลายไบต์

15. คริสตัล ที่ต่ออยู่ภายนอกใช้เพื่อสร้างฐานเวลาจริงของชิพ โดยใช้ความถี่ 32.768 กิโลเฮิรตซ์ ต่อเข้ากับขา X1 และ X2 และควรจะต่อโหลดคาปาซิเตอร์ (CI) ขนาด 6 Pf รวมด้วย

16. สัญญาณเวลาการทำงานของ DS1202

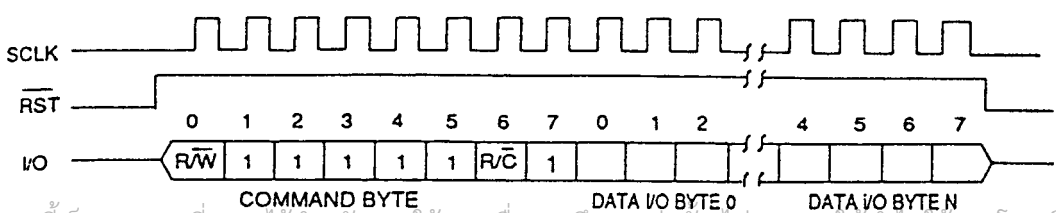
16.1 การติดต่อแบบไบต์เดียว



16.2 การติดต่อแบบหลายไบต์

**ภาพที่ 24**

แสดงไดอะแกรมเวลาการติดต่อแบบหลายไบต์

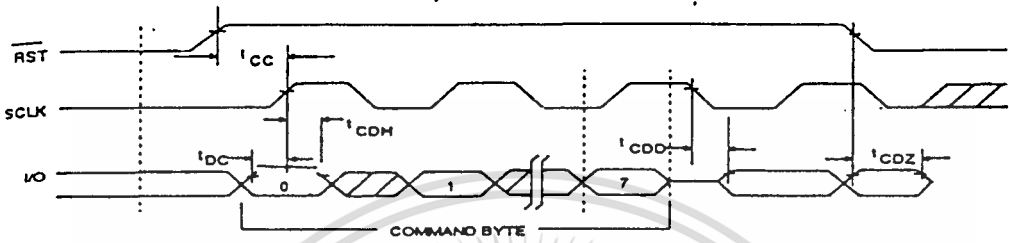


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 16.3 การอ่านข้อมูล

ภาพที่ 25

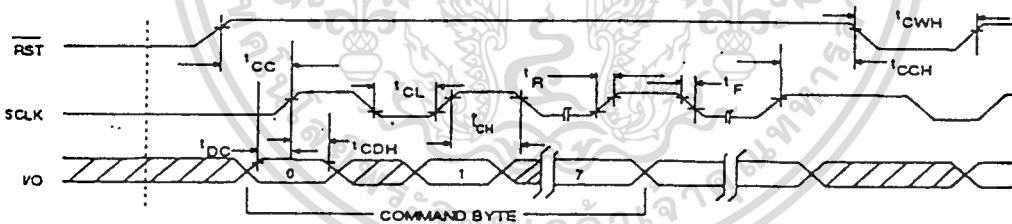
แสดงไทม์แอกแกรมเวลาการอ่านข้อมูล



### 16.4 การเขียนข้อมูล

ภาพที่ 26

แสดงไทม์แอกแกรมเวลาการเขียนข้อมูล

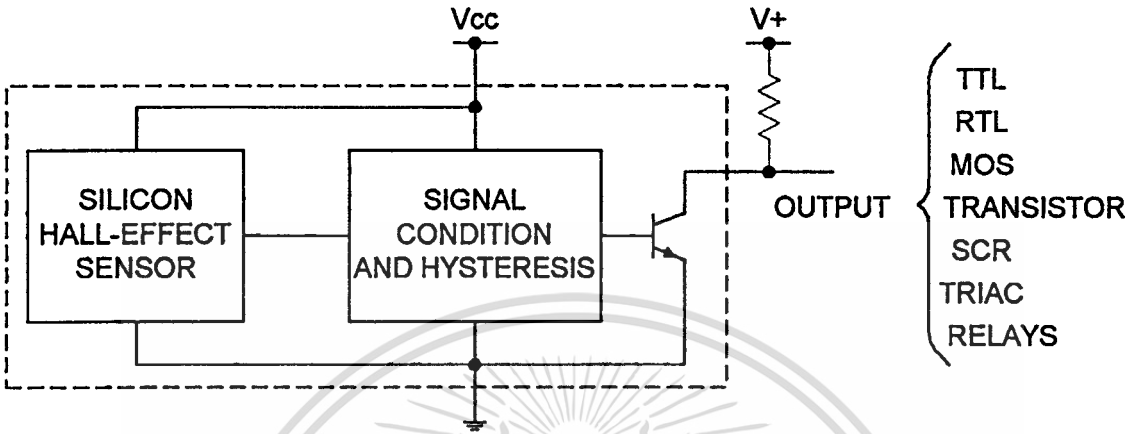


### เซนเซอร์ระยะทาง

ทำหน้าที่ตรวจจับความเร็ว และระยะทางที่รถเคลื่อนที่ไปได้ ซึ่งตัวตรวจจับความเร็วและระยะทางนี้จะมี 2 แบบ แบบหนึ่งเป็นแบบใบพัดตัดแสง และอีกแบบหนึ่งเป็นแบบแม่เหล็กที่เคลื่อนที่ผ่านตัวตรวจจับสนามแม่เหล็กแบบ Hall effect ซึ่งในปัจจุบันนี้จะใช้เฉพาะในแบบที่สองเท่านั้น เนื่องจากแบบแรกจะมีปัญหาการใช้งานเมื่อมีฝุ่นละอองมาก ในเซนเซอร์ แบบที่สองนี้จะมีโครงสร้างดังภาพที่ 27

ภาพที่ 27

แสดงโครงสร้างของเซนเซอร์วัดระยะทาง



การใช้งานเซนเซอร์วัดระยะทางและความเร็วนี้ จะมีแท่งแม่เหล็กติดอยู่กับแถบเหล็ก และมีตัวตรวจจับสนามแม่เหล็ก แบบ Hall effect ติดอยู่ใกล้กับแถบเหล็ก โดยแถบเหล็กที่ติดแม่เหล็กจะถูกติดตั้งในตำแหน่งระหว่างชุดเครื่องยนต์ กับมาตรวัดความเร็ว ดังภาพที่ 28

ภาพที่ 28

แสดงการต่อเซนเซอร์เพื่อใช้งาน



ส่วนประมวลผลกลาง

8031

เมื่อรถยนต์เคลื่อนที่จะทำให้แถบเหล็กของเซนเซอร์ที่ต่อกับเครื่องยนต์หมุน สนามแม่เหล็กที่ส่งไปยังตัวตรวจจับ แบบ Hall effect จะเปลี่ยนแปลง เมื่อแกนเหล็กหมุน 1 รอบ จะทำให้ความต้านทานระหว่าง คอลเลกเตอร์ (Collector) และ อีมีเตอร์ (Emitter) เพิ่มขึ้นแล้วลดลง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 ครั้ง ถือเป็น 1 ไชเคิล และเมื่อต้องวงจรภายนอกอีกเล็กน้อย จะทำให้เซนเซอร์ส่ง สัญญาณออกมาเป็น 2 ระดับ ถือเป็น 0 โวลต์ และเท่ากับไฟเลี้ยงระยะเซนเซอร์ (ประมาณ 12 โวลต์) นั่นคือ เมื่อแถบ ของเซนเซอร์หมุนไป 2 รอบ จะทำให้มีพัลส์ออกมา 1 ไชเคิล ถ้ารดยิ่งเร็ว ความกว้างของไชเคิลก็จะยิ่งลดลง ดังนั้นการวัดความกว้างของไชเคิล ก็จะไม่บอก ความเร็วของรดยนต์ได้ ดังนั้นเซนเซอร์แบบที่ใช้นี้จะให้พัลส์ซึ่งสามารถนำไปคำนวณหาระยะ ทางจากสมการ

$$\text{ระยะทาง} = \text{จำนวนพัลส์} \times \text{เส้นรอบวง} \quad (1)$$

ปัจจุบันในรดยนต์ซึ่งรวมถึงรถโดยสารรุ่นใหม่โดยเฉพาะที่มีระบบจุดระเบิดเป็นแบบอิเล็กทรอนิกส์ แล้วจะมีเซนเซอร์ชนิดนี้ต่อกับชุดเครื่องยนต์ และมีสายสัญญาณต่อออกมาเพื่อไปเข้ากับมาตรวัดอัตราเร็ว ดังนั้นจึงไม่จำเป็นจะต้องต่อเซนเซอร์เพิ่มเติมอีกและสามารถ นำสัญญาณพัลส์ที่ส่งออกมาใช้ในการคำนวณระยะทางได้

### ชิพสังเคราะห์เสียงแบบ ADPCM OKI MSM 6376

1. โครงสร้างของชิพ MSM 6376 เป็นชิพสังเคราะห์เสียงด้วยวิธีการ ADPCM ซึ่งมี 2 ช่องสัญญาณภายใน โดยใช้หน่วยความจำภายนอกได้สูงถึง 16 เมกกะบิต เช่น ROM หรือ EPROM นอกจากนั้นยังมีส่วนแปลงสัญญาณ D/A (Digital to Analog Converter) และ วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter) อยู่ภายในทำให้การต่อเพื่อใช้งานทำได้โดยง่าย เพียงต่อวงจรขยายกำลัง และลำโพงเท่านั้น

#### 2. คุณสมบัติเฉพาะ

2.1 สามารถต่อหน่วยความจำภายนอกได้สูงสุด 16 เมกกะบิต

2.2 สามารถเล่นเสียงสะท้อน (Echo) และเล่นเสียงพร้อมกันทั้ง 2 ช่องสัญญาณ

2.3 สามารถเลือกเสียงบีบ (Beep) ได้ 2 ความถี่ คือ 1 เฮิรซ์ และ 2 เฮิรซ์

2.4 สามารถเลือกความถี่สุ่ม (Sampling Frequency) ได้ตั้งแต่ 4, 6, 8.0 จนถึง 32

กิโลเฮิรซ์

2.5 เวลาในการพูดสูงสุดประมาณ 10.9 นาที (ที่ความถี่สุ่ม 6.4 กิโลเฮิรซ์) เมื่อใช้

ROM ขนาด 16 เมกกะบิต

2.6 สามารถเลือกคำที่จะพูดได้ทั้งหมด 111 คำ

2.7 มีวงจรแปลงสัญญาณ D/A และขจัด Pop Noise อยู่ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

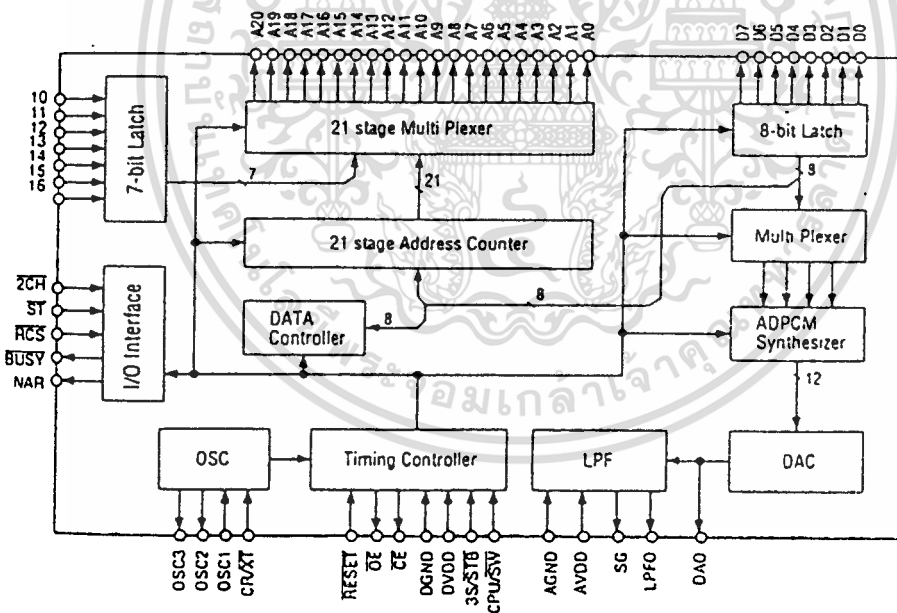
2.8 มีวงจรรองสัญญาณความถี่ต่ำผ่าน ซึ่งมีค่าการลดทอนเป็น 24 dB/OCT อยู่ในโหมดสแตนด์บาย (Standby Mode) เพื่อหยุดการทำงานของออสซิลเลเตอร์ และการทำงานอื่นๆ ภายในชิพ

2.9 มีวงจรถ่ายความถี่ออสซิลเลเตอร์ภายใน ซึ่งสามารถใช้การต่อวงจร RC หรือคริสตัล ภายนอกได้

2.10 ความถี่ออสซิลเลเตอร์ของ เอาท์พุท LPF เป็น 64-128 กิโลเฮิรซ์ เอาท์พุท D/A เป็น 64-256 กิโลเฮิรซ์

2.11 ไฟเลี้ยงวงจร 4.5-5.5 โวลท์

ภาพที่ 29  
แสดงบล็อกไดอะแกรมภายในของ MSM 6376

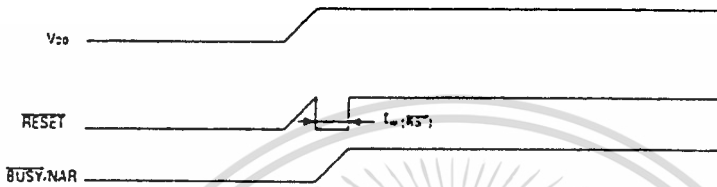


### 3. ไตอะแกรมเวลาการทำงานของ MSM 6376

#### 3.1 เริ่มต้นจ่ายไฟเลี้ยง

ภาพที่ 30

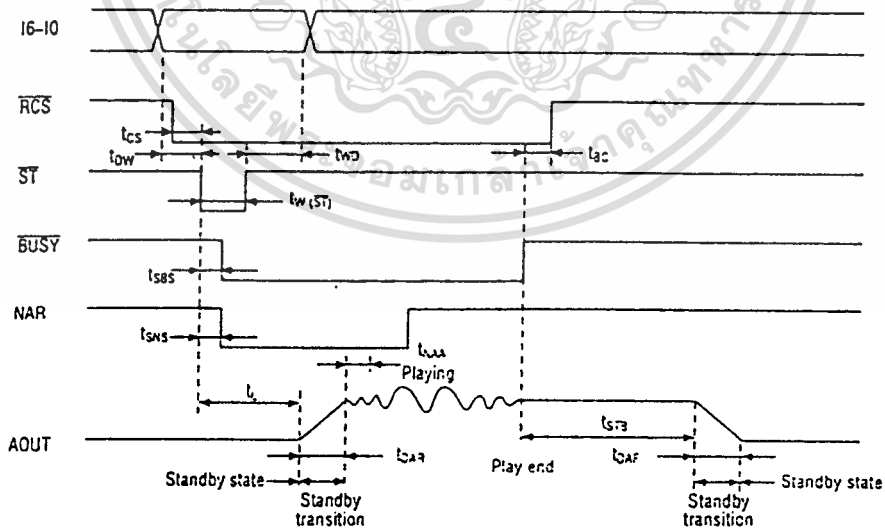
แสดงสัญญาณเวลาขณะเริ่มต้นจ่ายไฟเลี้ยง



#### 3.2 เริ่มต้นการทำงานและเข้าสู่สภาวะสแตนด์บาย

ภาพที่ 31

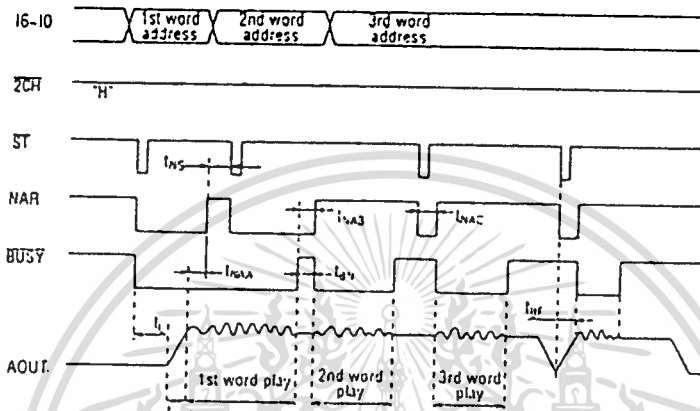
แสดงสัญญาณเวลาการทำงานและเข้าสู่สแตนด์บายโหมด



### 3.3 การทำงานกับช่องสัญญาณที่ 1 เท่านั้น (ใช้การเชื่อมต่อกับโปรเซสเซอร์)

ภาพที่ 32

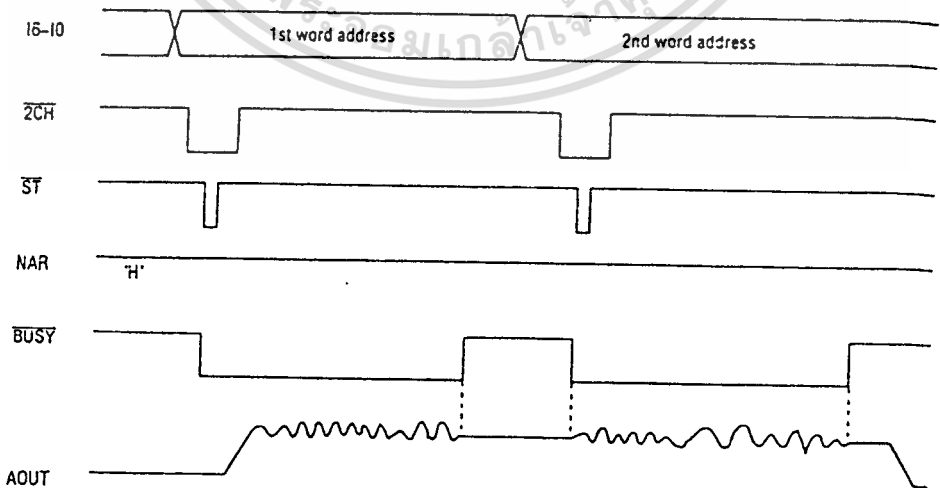
แสดงสัญญาณเวลาการทำงานกับช่องสัญญาณที่ 1 เท่านั้น



### 3.4 การทำงานกับช่องสัญญาณที่ 2 เท่านั้น (ใช้การเชื่อมต่อกับโปรเซสเซอร์)

ภาพที่ 33

แสดงสัญญาณเวลาการทำงานกับช่องสัญญาณที่ 2 เท่านั้น

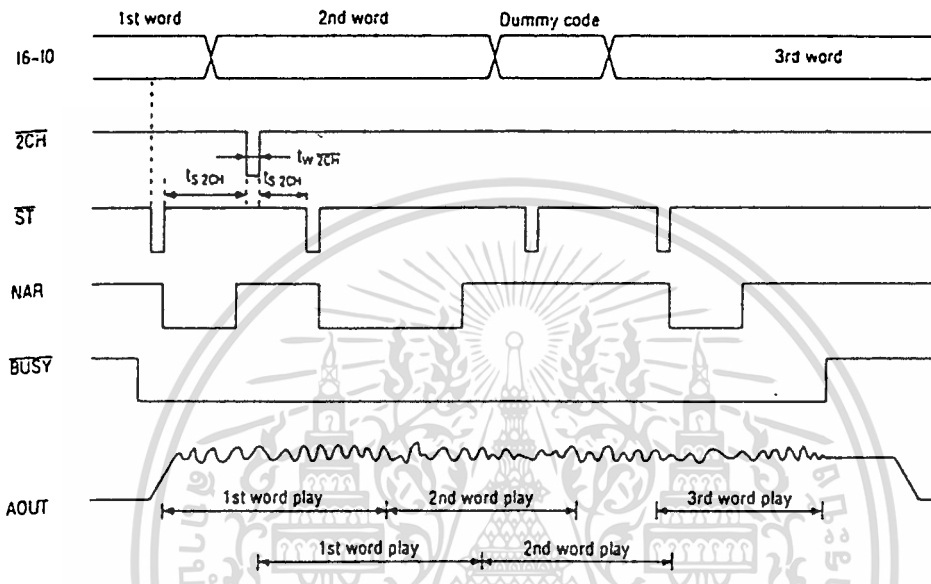


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 การเล่นเสียงสะท้อน ในช่องสัญญาณที่ 1

ภาพที่ 34

แสดงสัญญาณเวลาการเล่นเสียงสะท้อน ในช่องสัญญาณที่ 1



3.5.1 ถ้าสัญญาณ  $\overline{2CH}$  แยกที่ฟิซขณะที่ สัญญาณ  $\overline{ST}$  ไม่แยกที่ฟิซ จะเข้าสู่การเล่นเสียงสะท้อน เมื่อเสร็จสิ้นการทำงานการเล่นเสียงสะท้อนจะถูกยกเลิก ยกเว้นเป็นการเล่นเสียงสะท้อนอย่างต่อเนื่อง

3.5.2 ขณะเล่นเสียงสะท้อน สัญญาณเสียงจะมาจากสัญญาณเสียงของช่องสัญญาณที่ 1 (รับเข้ามาโดย  $\overline{ST}$ ) รวมกับสัญญาณเสียงของช่องสัญญาณที่ 2 ขนาดลดลง 6 dB (รับเข้ามาโดย  $\overline{2CH}$ )

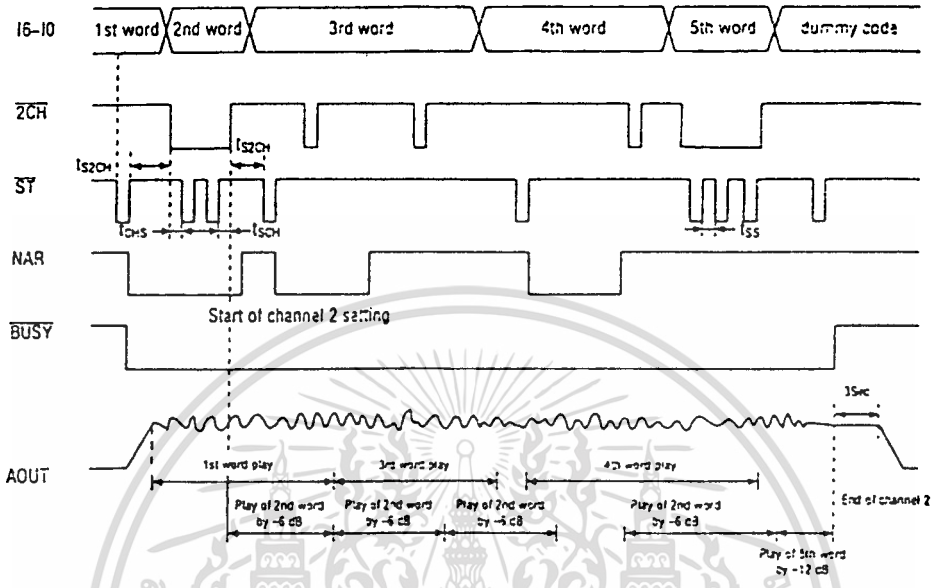
3.5.3 การเล่นเสียงอย่างต่อเนื่อง นั้นจะทำให้เสียงสะท้อนเกิดขึ้นในคำที่จะพูดต่อไปด้วย

3.5.4 ควรป้อนรหัสที่ไม่ได้ใช้งานให้กับชิพ ภายหลังจากที่การเล่นเสียงสะท้อนสิ้นสุดลง เพื่อให้เกิดการหน่วงเวลา ประมาณ 3 วินาทีก่อนเข้าสู่โหมดสแตนด์บาย

### 3.6 การเล่นเสียงพร้อมกับ 2 ช่องสัญญาณ

## ภาพที่ 35

แสดงสัญญาณเวลาการเล่นเสียงพร้อมกัน 2 ช่องสัญญาณ



3.6.1 เมื่อสัญญาณ ST แยกที่ขณะที่ 2CH เป็น "L" จะทำให้ช่องสัญญาณที่ 2 เริ่มเล่นเสียงซ้อนเข้าไป และระดับความดังของเสียงที่ซ้อนเข้าไบนั้นขึ้นอยู่กับจำนวนของพัลส์ ST ตามตารางด้านล่าง

3.6.2 ระดับความดังของการเล่นเสียงซ้อนของช่องสัญญาณที่ 2 นี้จะใช้ตลอด จนกว่าจะเข้าสู่โหมดสแตนด์บาย หรือมีการรีเซตช่องสัญญาณที่ 2 เกิดขึ้น

3.6.3 ควรบ่อนรหัสที่ไม่ได้ใช้งานให้กับซีพ ภายหลังจากที่การเล่นเสียงสะท้อนสิ้นสุดลง เพื่อให้เกิดการหน่วงเวลาประมาณ 3 วินาทีก่อนเข้าสู่โหมดสแตนด์บาย

## ตารางที่ 27

แสดงความสัมพันธ์ระหว่างจำนวนพัลส์ของ ST

จำนวนพัลส์ของ ST	การลดทอน
1	ไม่ลดทอน (100% ข้อมูลเสียง)
2	ลดทอน 6 dB (1/2 ข้อมูลเสียง)
3	ลดทอน 12 dB (1/4 ข้อมูลเสียง)

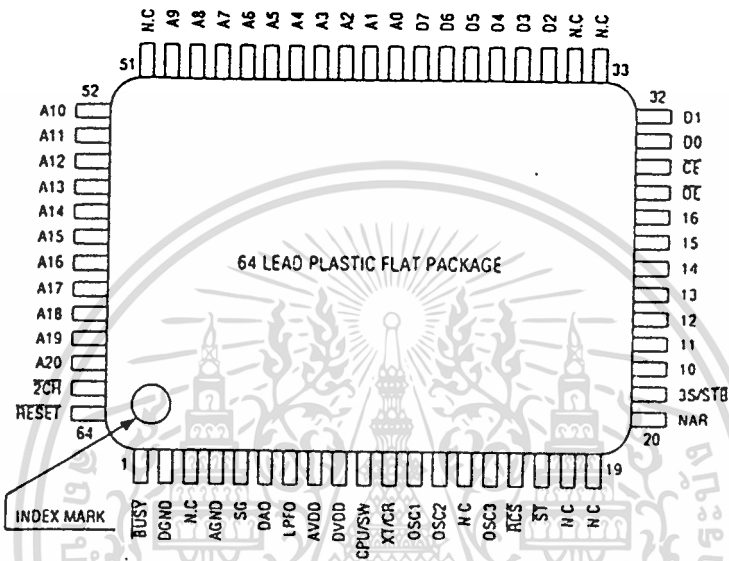
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. การจัดเรียงและรายละเอียดของสัญญาณ

##### 4.1 การจัดเรียงขาสัญญาณ

ภาพที่ 36

แสดงการจัดเรียงขาสัญญาณของMSM6376



##### 4.2 รายละเอียดขาสัญญาณ

ชื่อ	อินพุต/เอาต์พุต	
IO-16	อินพุต	เลือกข้อความที่จะพูด โดยข้อมูลนี้จะถูกอ่านเข้า เมื่อระดับของสัญญาณ ST เป็น "L" และจะค้างข้อมูลไว้ภายในชิพที่ขอบขาขึ้นของสัญญาณ ST
A0-A20	เอาต์พุต	สัญญาณแอดเดรสเพื่อติดต่อกับหน่วยความจำภายนอก
D0-D7	อินพุต	สัญญาณดาต้าที่รับเข้ามาจากหน่วยความจำภายนอก ข้อมูลจะส่งมาเมื่อสัญญาณ RCS เป็น "L"
2CH	อินพุต	ใช้เล่นเสียงสะท้อน หรือ เล่นเสียงพร้อมกันทั้ง 2 ช่อง สัญญาณ 1 อยู่จะเข้าสู่โหมดการเล่นเสียงสะท้อน และช่วงเวลาหน่วง (Delay) ในการเล่นเสียงสะท้อนก็ขึ้นอยู่กับเวลาในการป้อนสัญญาณ 2CH ถ้าสัญญาณ ST แยกที่ฟเมื่อสัญญาณ 2CH เป็น "L" จะมีการเล่นเสียงของช่องสัญญาณที่ 2 ใหม่

ST	อินพุต	เมื่อสัญญาณ ST แยกทีฟ (ขอบขาลง) จะเป็นการอ่านข้อมูลจากขา 10-16 และจะค้างข้อมูลไว้ภายในที่ขอบขาขึ้นของ ST ถ้าระดับของสัญญาณ NAR เป็น "H" จะเป็นการเล่นข้อมูลของช่องสัญญาณที่ 2 ใหม่ โดยสามารถปรับระดับความดังจากจำนวนพัลส์ของ ST (ขณะที่ 2CH เป็น "L") ในการเชื่อมต่อกับสวิทช์ การเล่นเสียงจะทำซ้ำ เมื่อ ST เป็น "L"
RCS	อินพุต	เมื่อสัญญาณ RCS นี้แยกทีฟ (เป็น "L") จะทำให้ขาสัญญาณในการเชื่อมต่อๆ สามารถทำงานได้ เช่น ST เป็นอินพุต A0-A20 OE และ CE เป็นเอาต์พุต แต่เมื่อสัญญาณนี้ไม่แยกทีฟ (เป็น "H") จะทำให้ขาสัญญาณเอาต์พุตอยู่ในสถานะอิมพีแดนซ์สูง (High Impedance)
BUSY	เอาต์พุต	ขณะที่กำลังเล่นเสียงอยู่จะทำให้ขาสัญญาณนี้เป็น "L"
NAR	เอาต์พุต	เมื่อสัญญาณนี้เป็น "H" สามารถส่งข้อมูลต่อไปเข้ามา
CR/XT	อินพุต	เมื่อให้ระดับลอจิกขาสัญญาณนี้เป็น "H" จะเป็นการ กำหนดขา OSC1 OSC2 และ OSC3 เป็นเทอร์มินัลของวงจร คริสตอล ออสซิลเลเตอร์นี้ ซึ่งจะมีตัวต้านทาน 2 เมกกะโอห์ม ต่อระหว่างขา OCS1 และ OCS2
OCS1	อินพุต	เทอร์มินัลของคริสตอลหรือ RC ออสซิลเลเตอร์
OSC2, OSC3	เอาต์พุต	ถ้าใช้สำหรับคริสตอลออสซิลเลเตอร์ จะปล่อยขา OSC3 ลอยไว้ ถ้าใช้สัญญาณนาฬิกาจากภายนอก จะต่อเข้ากับขา OSC1 และปล่อยขา OSC2 และ OSC3 ลอยไว้
CPU/SW	เอาต์พุต	ถ้าใช้สำหรับคริสตอลออสซิลเลเตอร์ จะปล่อยขา OSC3 ลอยไว้ ถ้าใช้สัญญาณนาฬิกาจากภายนอก จะต่อเข้ากับขา OSC1 และปล่อยขา OSC2 และ OSC3 ลอยไว้
CPU/SW	อินพุต	ถ้าระดับลอจิกขาสัญญาณนี้เป็น "H" เป็นการเลือกใช้งานร่วมกับ CPU ภายนอก ถ้าระดับลอจิกขาสัญญาณนี้เป็น "L" เป็นการเลือกใช้งานร่วมกับสวิทช์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3S/STB	อินพุต	ถ้าระดับลอจิกขาสัญญาณนี้เป็น "H" ชิพจะเข้าสู่สภาวะสแตนด์บาย หลังจากการสังเคราะห์เสียงเสร็จ 3 วินาที ถ้าระดับลอจิกขาสัญญาณนี้เป็น "L" จะทำให้สัญญาณเอาต์พุตของวงจร D/A เป็น 1/2 ของ Vdd หลังจากการสังเคราะห์เสียงเสร็จ
RESET	อินพุต	ถ้าระดับลอจิกขาสัญญาณนี้เป็น "L" ชิพจะเข้าสู่สภาวะสแตนด์บาย ขณะที่มีการรีเซต ออสซิลเลเตอร์จะหยุดทำงาน เอาต์พุตของ D/A จะเป็นระดับกราวด์ MSM 6376 จะมีส่วน Power on Reset อยู่ภายใน และเพื่อให้การทำงานรีเซตภายในมีความถูกต้องควรให้ไฟเลี้ยงวงจรขึ้นภายใน 1 มิลลิวินาที ถ้าไม่สามารถทำได้ควรให้พัลส์รีเซตเมื่อมีการจ่ายไฟเลี้ยง
CE	เอาต์พุต	สัญญาณเลือกชิพหน่วยความจำภายนอกสัญญาณนี้จะให้เอาต์พุตได้เมื่อ RCS เป็น "L"
OE	เอาต์พุต	สัญญาณจะควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก สัญญาณนี้จะให้เอาต์พุตได้เมื่อ RCS เป็น "L"
DAO	เอาต์พุต	สัญญาณเอาต์พุตอนาล็อก (เสียง) ที่ส่งมาจากส่วน D/A
AOUT	เอาต์พุต	สัญญาณเอาต์พุตอนาล็อก (เสียง) ที่ส่งมาจากส่วน LPF
SG	เอาต์พุต	เทอร์มินัลสำหรับการปรับอัตรา SN ของส่วน LPF ภายใน ถ้าต้องการเอาต์พุตจาก LPF ไปใช้งานควรต่อตัวเก็บประจุประมาณ 1 uF
AVDD		แหล่งจ่ายไฟส่วนอนาล็อกภายในชิพ
AGND		กราวด์ส่วนอนาล็อกภายในชิพ
DVDD		แหล่งจ่ายไฟส่วนดิจิตอลภายในชิพ
DGND		กราวด์ส่วนดิจิตอลภายในชิพ

## 5. รายละเอียดการทำงาน

5.1 การเลือกรหัสของคำพูด ผู้ใช้สามารถเลือกคำที่จะพูดได้ทั้งหมด 111 คำ โดยเลือกจากข้อมูล 10-16 ตามตารางที่ 29

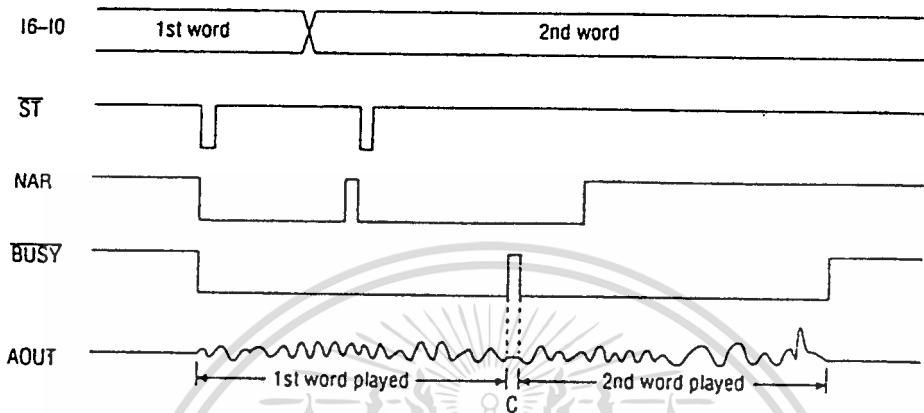
ตารางที่ 28  
แสดงการเลือกรหัสคำพูดและรหัสเสียงอื่นๆ

IO - I6	ข้อมูล
0000000	รหัสหยุด
0000001	ข้อมูลที่ผู้ใช้กำหนด
ถึง	111 คำ
1101111	
111000	
ถึง	รหัสบีบ
11101111	
1111000	
ถึง	รหัสทดสอบ
1111111	

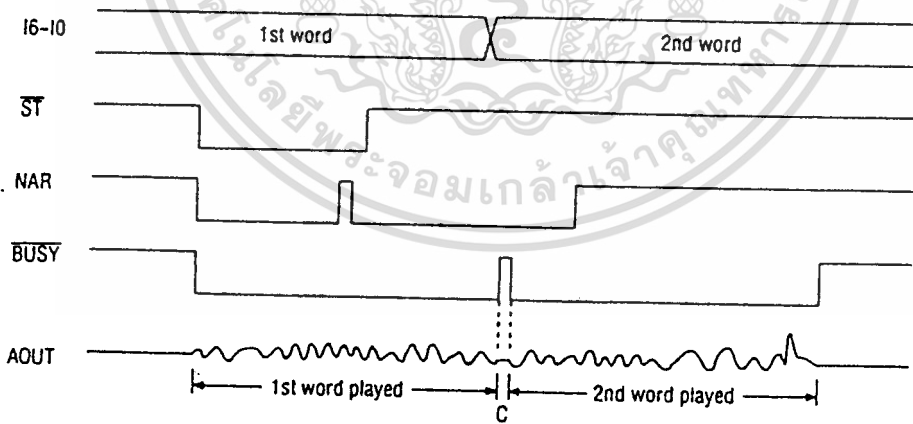
5.2 การรับสัญญาณข้อมูลจาก CPU หรือ สวิตช์ การที่ชิพจะรับสัญญาณจาก CPU หรือ สวิตช์นั้น จะเลือกจากระดับสัญญาณขา CPU/SW ถ้าป้อนระดับแรงดันเป็น "H" ก็จะเป็นการทำงานเพื่อรับข้อมูลจาก CPU แต่ถ้าป้อนระดับลอจิกให้ขานี้เป็น "L" ก็จะทำงาานเพื่อรับข้อมูลจากสวิตช์ ซึ่งการรับข้อ

5.2.1 การรับข้อมูลจาก CPU เมื่อสัญญาณ NAR เป็น "H" และมีพัลส์ ST เกิดขึ้น ข้อมูล IO-16 จะถูกดึงเข้าภายในชิพและนำข้อมูลนั้นไปใช้เพื่อเป็นรหัสในการสร้างคำพูด การรับสัญญาณจาก CPU นี้สามารถเล่นเสียงต่อเนื่องได้ ถ้าพัลส์  $\overline{ST}$  นั้นเป็น "L" นานเกินกว่า 800  $\mu$ S จะทำให้เกิดการเล่นเสียงซ้ำอีก 1 ครั้ง (ที่อัตราสุ่ม 8 กิโลเฮิรท์)

## ภาพที่ 37

แสดงสัญญาณการรับข้อมูลจาก CPU ( $\overline{ST} \leq 350 \mu\text{s}$ )

## ภาพที่ 38

แสดงสัญญาณการรับข้อมูลจาก CPU ( $\overline{ST} > 800 \mu\text{s}$ )

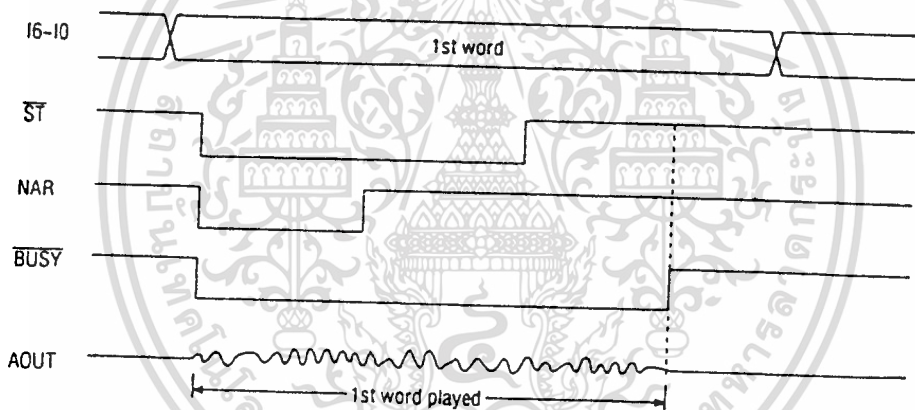
5.2.2 การรับสัญญาณข้อมูลจากสวิทช์ เมื่อสัญญาณ  $\overline{ST}$  เป็น "L" จะมีการเล่นเสียงจากข้อมูลรหัส 10-16 ตามปกติ แต่ถ้าเล่นเสียงเสร็จสมบูรณ์แล้วสัญญาณ  $\overline{ST}$  ยังเป็น "L" ก็จะทำให้เกิดการเล่นเสียงของข้อมูลเดิมซ้ำอีกครั้งหนึ่ง จนกว่า  $\overline{ST}$  จะเป็น "H"

แต่ถ้าต้องการเปลี่ยนคำพูดใหม่ และให้การพูดต่อเนื่องจะทำโดยการให้  $\overline{ST}$  เป็น "L" แล้วเปลี่ยนข้อมูล 10-16 ในขณะที่กำลังพูดคำแรกอยู่

การรับสัญญาณข้อมูลจากสวิทช์ จะไม่สามารถเล่นใหม่ดเสียงสะท้อน และใหม่ดเล่นเสียงพร้อมกันทั้ง 2 ช่องสัญญาณ"

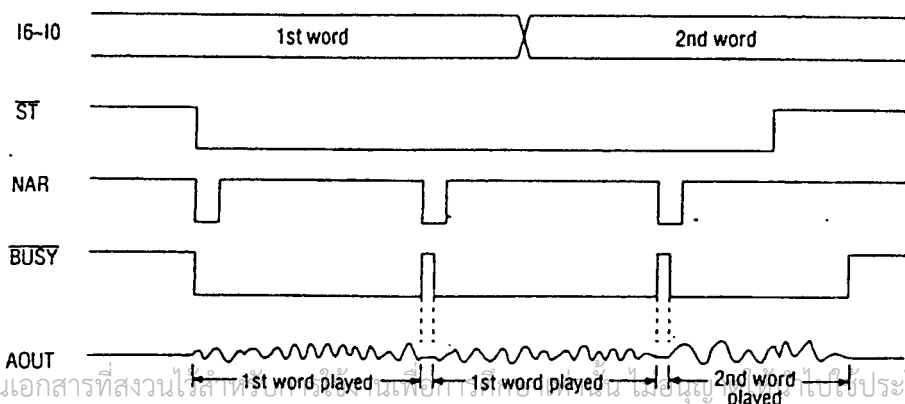
ภาพที่ 39

แสดงสัญญาณเวลาการรับข้อมูลจากสวิทช์ (เล่นเสียง 1 คำ)



ภาพที่ 40

แสดงสัญญาณเวลาการรับข้อมูลจากสวิทช์ (เล่นเสียงซ้ำ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้เผยแพร่ไปยังหน่วยงานภายนอกโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การสร้างเสียงบีบ ภายในชิพจะมีวงจรสร้างเสียงบีบ ดังนั้น จึงสามารถเลือกเล่นเสียงบีบ ได้จาก I0-I6 โดยที่แอมพลิจูดของเสียงประมาณ  $1/4 V_{dd}$  ดังตารางด้านล่าง

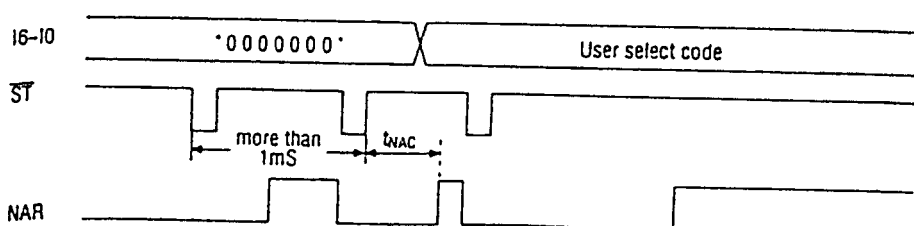
ตารางที่ 29  
แสดงรหัสการเลือกเสียงบีบเสียงต่างๆ

16	15	14	13	12	11	10	ความถี่เสียงบีบ	ช่วงเวลา (วินาที)
1	1	1	0	0	0	0	2.0	0.064
					0	1		0.125
					1	0		0.250
					1	1		0.500
				0	0	0	1.0	0.064
					0	1		0.125
					1	0		0.250
					1	1		0.500

5.4 รหัสหยุด การเล่นเสียงพูดจะสิ้นสุดเมื่อมีการรับรหัสหยุดคือ 10-16 เป็น "0" ทั้งหมด ซึ่งจะทำให้เอาต์พุตของส่วน D/A จะเป็น  $1/2 V_{dd}$

การให้สัญญาณ ST นั้นจะกระทำได้ 2 ลักษณะคือ ให้สัญญาณ St เป็น "L" นานกว่า 1 มิลลิวินาที หรือให้สัญญาณ ST ตามภาพที่ 41

ภาพที่ 41  
แสดงการใส่รหัสหยุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสหยุดนั้นจะทำหน้าที่หยุดเฉพาะการเล่นเสียงเท่านั้น ส่วนวงจรรอซซิลเลเตอร์ และ วงจรอื่น ๆ ยังคงทำงานต่อไปแต่รหัสหยุดจะไม่สามารถหยุดการเล่นเสียงบีบได้

5.5 ความถี่การสุ่ม (Sampling Frequency) ความถี่การสุ่มนี้จะถูกกำหนด เพื่อที่จะ อ่านข้อมูลเสียงที่จะเก็บไว้ใน ROM ซึ่งจะมีความสัมพันธ์กับความถี่หลัก คือ  $F_{osc}$  ดังนั้น ความถี่ การสุ่ม ( $F_s$ ) จะเป็นดังนี้

$$F_{s1} = F_{osc}/8, F_{s2} = F_{osc}/10, F_{s3} = F_{osc}/16$$

5.6 การเล่นเสียงสะท้อน และการเล่นเสียงพร้อมกันทั้งสองช่องสัญญาณ

5.6.1 การเล่นเสียงสะท้อน การเล่นเสียงสะท้อนจะสร้างมาจาก สัญญาณเสียง ของช่องที่ 1 รวมกับสัญญาณเสียงเดิมที่ถูกหน่วงเวลาและลดทอนลง 6 dB (ประมาณ ๗ แอมพลิจูดสัญญาณเดิม)

5.6.2 การเล่นเสียงพร้อมกันทั้งสองช่องสัญญาณ การเล่นเสียงแบบนี้จะสามารถ เล่นเสียงที่ต่างกันในเวลาเดียวกัน ซึ่งจะใช้งานเกี่ยวกับ BGM (Back Ground Music) เป็นต้น ข้อมูลเสียงของช่องสัญญาณที่ 2 สามารถเล่นซ้อนเข้าไปได้ แต่ระดับความดังของเสียงสามารถ เปลี่ยนแปลงได้เป็น 1, 1/2 และ 1/4 ของความดังของช่องสัญญาณ 1 ขึ้นกับจำนวนพัลส์ของ ST ดังตารางด้านล่าง

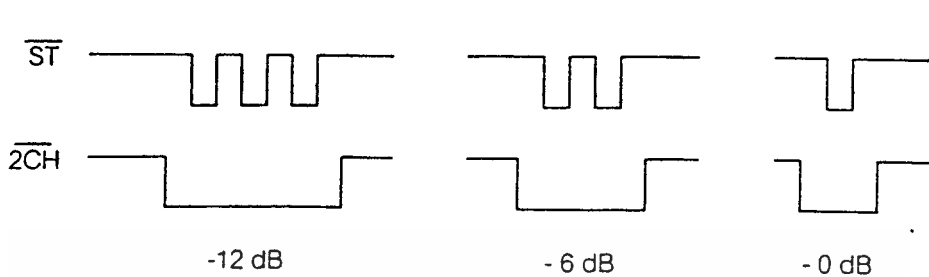
### ตารางที่ 30

แสดงความสัมพันธ์ของจำนวนพัลส์ ST กับการลดทอนสัญญาณ

จำนวนพัลส์ ST	การลดทอน
1	ไม่ลดทอน (100% ข้อมูลเสียง)
2	ลดทอน 6 dB (1/2 ข้อมูลเสียง)
3	ลดทอน 12 dB (1/4 ข้อมูลเสียง)

## ภาพที่ 42

แสดงสัญญาณเวลาของการเล่นเสียงผสม 2 ช่องสัญญาณ



เมื่อมีการกำหนดให้เล่นเสียงผสม 2 ช่องสัญญาณใหม่นี้จะยังทำงานอยู่จนกระทั่งมีการเข้าสู่โหมดสแตนด์บาย หรือจนกว่าจะมีการรีเซ็ตช่องสัญญาณ 2

5.7 การเข้าสู่โหมดสแตนด์บาย เมื่อมีการเอ็นเนเบิลการเข้าโหมดสแตนด์บายแล้วชิพจะเข้าสู่สภาวะสแตนด์บาย หลังจากเสร็จสิ้นการสังเคราะห์เสียง เป็นเวลา 3 วินาที เมื่อเข้าสู่สภาวะนี้ จะทำการหยุดทำงานทุกชนิด ดังนั้น เมื่อจะมีการทำงานใหม่จะต้องใช้เวลาประมาณ 100 มิลลิวินาที ก่อนที่จะเริ่มเล่นเสียงได้เนื่องจากต้องรอเวลาในการทำงานของวงจรจับ Pop Noise ถ้าไม่มีการเอ็นเนเบิลการเข้าสแตนด์บายใหม่เมื่อเสร็จการสังเคราะห์เสียง เอาท์พุทที่ขา AOUT จะมีระดับแรงดันประมาณ 1/2 ของ Vdd เมื่อมีการทำงานใหม่จะใช้เวลาเริ่มต้นประมาณ 350  $\mu$ S ในกรณีที่ต้องการเข้าสู่โหมดสแตนด์บาย จะต้องป้อนพัลส์ RESET แทน แต่จะเกิด Pop Noise ที่ AOUT

5.8 สัญญาณเสียงเอาท์พุท สัญญาณเสียงนั้นสามารถเลือกการใช้งานได้จาก 2 ขา สัญญาณ คือ เอาท์พุทของ D/A หรือเอาท์พุทจากส่วน LPF

5.8.1 เอาท์พุทของส่วนแปลงสัญญาณ D/A ระดับเอาท์พุทจากส่วนนี้จะมีค่าสูงสุดเป็น 4095/4096 Vdd เมื่อเลือกเอาท์พุทจากส่วนนี้จำเป็นต้องต่อวงจร กรองความถี่ต่ำผ่านภายนอก อิมพีแดนซ์ของส่วนเอาท์พุท D/A จะอยู่ในช่วง 15 ถึง 35 กิโลโอห์ม ดังนั้น ควรจะหา วงจรกรองที่เหมาะสม เพื่อไม่ให้มีผลกับความถี่คัตออฟ (Cut off frequency) ของวงจรกรองได้ตามตารางที่ 31 แสดงระดับเอาท์พุทจากส่วน DA

ตารางที่ 31  
แสดงระดับสัญญาณเอาต์พุตจากส่วน D/A

เงื่อนไข	ระดับต่ำสุด	ระดับกลาง	ระดับสูงสุด	หน่วย
เล่น 1 ช่อง	1/4 Vdd	1/2 Vdd	3/4 Vdd	V
เล่นผสม 2 ช่อง	0.0	1/2 Vdd	Vdd	V
เล่นเสียงบีบ	3/8 Vdd	1/2 Vdd	3/4 Vdd	V

5.8.2 เอาต์พุตของวงจรรองความถี่ต่ำผ่าน เนื่องจากว่าวงจรรองความถี่ต่ำผ่าน นั้นประกอบด้วย สวิทช์คาปาซิเตอร์ ดังนั้น ความถี่คัทออฟ จะแปรผันกับความถี่ของออสซิลเลเตอร์ เมื่อความถี่การสุ่ม ( $F_s$ ) เป็น 1/8 และ 1/10 ของความถี่ออสซิลเลเตอร์ ความถี่คัทออฟ ( $F_{cut}$ ) จะเป็น 3/64 ของ  $F_{osc}$  และถ้า  $F_s = 1/16$  แล้ว  $F_{cut} = 3/128 \times F_{osc}$

ตารางที่ 32  
แสดงความถี่คัทออฟของวงจรรองความถี่ต่ำผ่าน

ความถี่สุ่ม ( $F_s$ )	ความถี่หลัก ( $F_{osc}$ )	ความถี่คัทออฟ ( $F_{cut}$ )
4.0 KHz	64 KHz	1.5 KHz
6.4 KHz	64 KHz	3.0 KHz
8.0 KHz	64 KHz	3.0 KHz
12.8 KHz	128 KHz	6.0 KHz
16.0 KHz	128 KHz	6.0 KHz

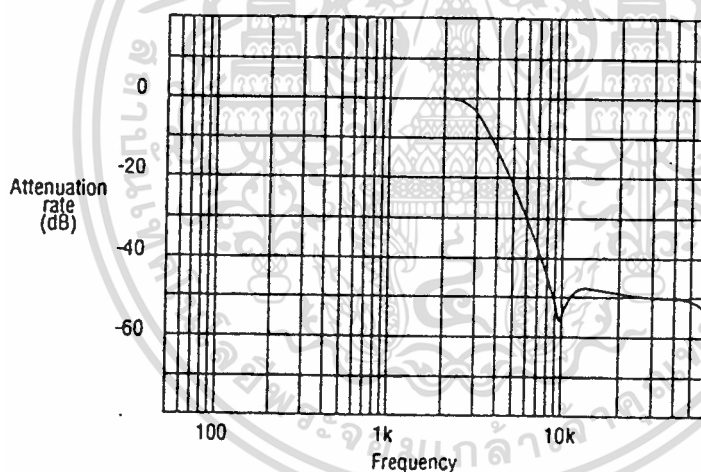
คุณสมบัติของวงจรรองความถี่ต่ำผ่านเมื่อมีความถี่สุ่มเป็น 8 กิโลเฮิรท์ จะแสดงดังภาพที่ 43 และตารางที่ 34 แสดงระดับเอาต์พุตจากส่วนวงจรรองความถี่ต่ำผ่าน

ตารางที่ 33  
แสดงระดับเอ๊าท์พุทของวงจรรองความถี่ต่ำผ่าน

เงื่อนไข	ระดับต่ำสุด	ระดับกลาง	ระดับสูงสุด	หน่วย
เล่น 1 ช่อง	1/4 Vdd	1/2 Vdd	3/4 Vdd	V
เล่นผสม 2 ช่องสัญญาณ	0.0	1/2 Vdd	Vdd - 0.7	V
เล่นเสียงบีบ	3/8 Vdd	1/2 Vdd	3/8 Vdd	V

ภาพที่ 43

แสดงคุณสมบัติของวงจรรองความถี่ต่ำผ่านที่  $F_s = 8$  กิโลเฮิรท์

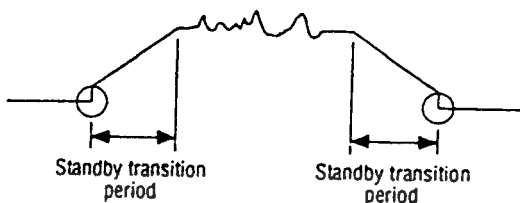


5.8.3 Pop Noise ของวงจรรองความถี่ต่ำผ่าน แม้ว่าภายในชิพจะมีวงจรถัด Pop Noise แต่ก็มี Pop Noise ได้ในช่วงที่แสดงดังภาพที่ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

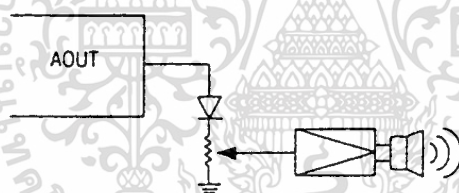
### ภาพที่ 44

แสดง Noise ของวงจรรองความถี่ต่ำผ่าน



ถ้าต่อไดโอดที่เอาต์พุตของ AOUT ดังภาพที่ 45 Pop Noise จะลดลงได้

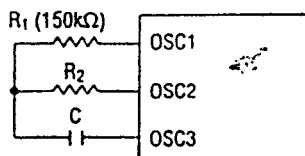
ภาพที่ 45  
แสดงวงจรจัด Pop Noise ภายนอก



5.9 RC ออสซิลเลเตอร์ จะมีลักษณะการต่อใช้งานดังภาพที่ 46

ภาพที่ 46

แสดงการต่อวงจร RC ออสซิลเลเตอร์

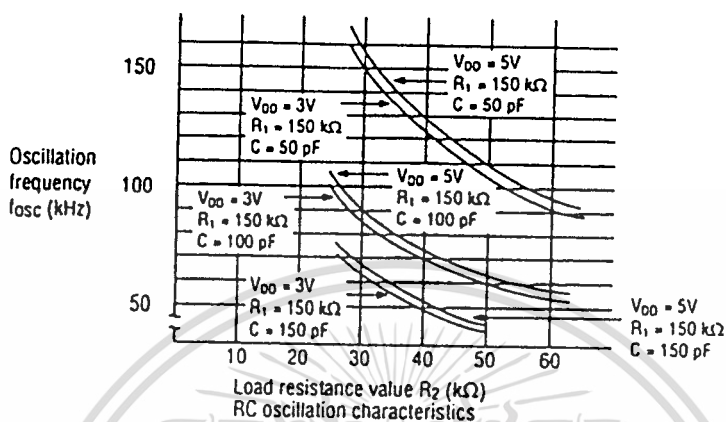


5.9.1 การคำนวณค่าคงที่ RC ในภาพที่ 47 จะแสดงคุณสมบัติของการใช้ RC ออสซิลเลเตอร์ภายใน ถ้าต้องการ  $F_{osc}$  เป็น 64 กิโลเฮิรท์ จะเลือกค่า C และ R2 ได้ดังนี้ คือ  $C = 100$

$\mu F$ ,  $R_1 = 150K$ ,  $R_2 = 50K$  สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ภาพที่ 47

แสดงคุณสมบัติของวงจร RC ออสซิลเลเตอร์

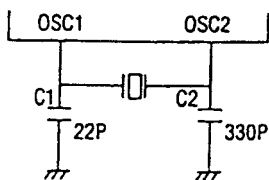


5.9.2 การกระเพื่อม (Fluctuation) ของการใช้ RC ออสซิลเลเตอร์ การใช้ RC ออสซิลเลเตอร์นั้นจะทำให้ความถี่กำเนิดขึ้นมีการกระเพื่อมขึ้นเนื่องจาก กระบวนการภายในชิพ เมื่อใช้ความต้านทาน  $R_2$  50 กิโลโอห์ม การผิดพลาดจาก  $R_2$  จะเป็น 2% และตัวเก็บประจุ จะเป็น 1% ส่วนกระบวนการภายในจะเป็น 4% ดังนั้น ค่าโดยรวมของการกระเพื่อม จะเป็น 7%

5.10 คริสตอลออสซิลเลเตอร์ การต่อวงจรเพื่อใช้คริสตอลออสซิลเลเตอร์ ใช้คริสตอลความถี่ 64 กิโลเฮิรซ์ แสดงดังภาพที่ 48

### ภาพที่ 48

แสดงวงจรการต่อคริสตอลออสซิลเลเตอร์



# บทที่ 3

## การออกแบบ

### การออกแบบทางฮาร์ดแวร์

ระบบจัดการบรรณโดยสารประจำทางนี้ จะแบ่งการออกแบบเป็น 3 ส่วนหลัก ดังไดอะแกรมแผนภูมิที่ 49

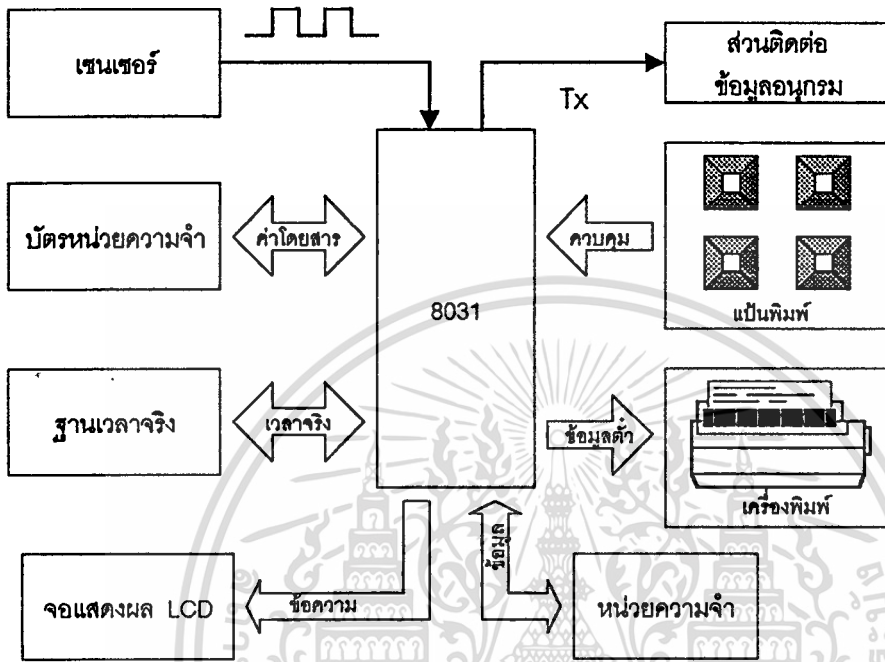
#### แผนภูมิที่ 49

แสดงบล็อกไดอะแกรมหลักของการออกแบบระบบจัดการ



1. ส่วนควบคุมระบบหลัก ทำหน้าที่ ควบคุมการทำงานทั้งหมด โดยจะรับสัญญาณพัลส์จากอุปกรณ์เซนเซอร์ เพื่อคำนวณระยะทางและแสดงผลในส่วนต่าง ๆ นอกจากนี้ ยังทำหน้าที่จัดเก็บค่าโดยสารผ่านบัตรหน่วยความจำ การทำงานของส่วนควบคุมระบบนี้จะประกอบด้วยส่วนต่าง ๆ ดังไดอะแกรมภาพที่ 50

**แผนภูมิที่ 50**  
แสดงบล็อกไดอะแกรมของส่วนควบคุมระบบหลัก



1.1 วงจรควบคุม วงจรส่วนนี้จะใช้ไมโครคอนโทรลเลอร์ 8031 เป็นตัวประมวลผลกลางโดยมีการจัดตำแหน่งของหน่วยความจำและอุปกรณ์อินพุต/เอาต์พุต ดังตารางที่แสดงวงจรดังตารางที่ 35

ตารางที่ 34

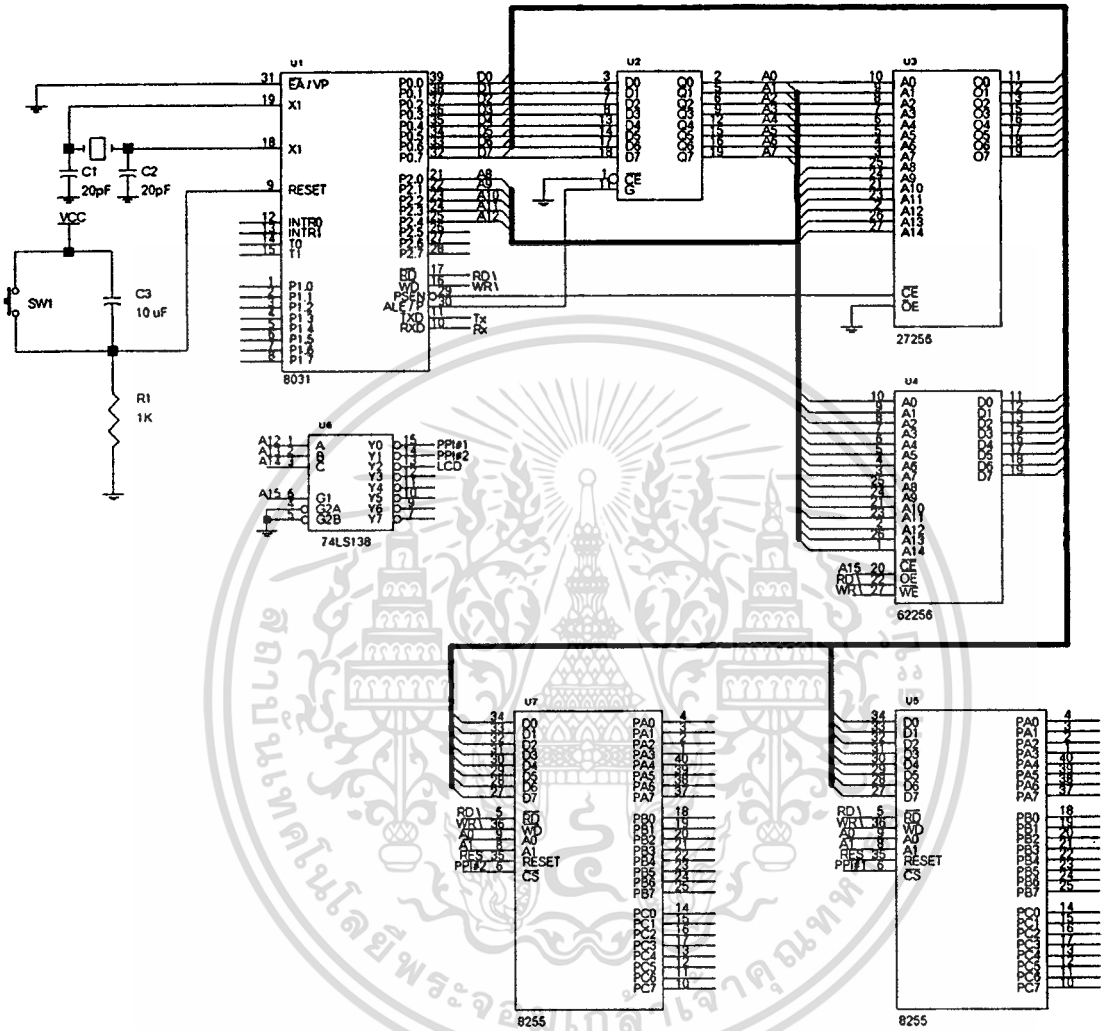
แสดงการจัดวางตำแหน่งของหน่วยความจำและอินพุต/เอาต์พุตบนวงจร  
ส่วนควบคุมการทำงานหลัก

อุปกรณ์	ตำแหน่ง
ROM	0000H - 7FFFH (Program memory)
RAM	0000H - 7FFFH (Data memory)
8255#1	8000H - 8003H (I/O)
8255#2	9000H - 9003H (I/O)
LCM	0A000H - 0A001H (I/O)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิใช่เพื่อให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

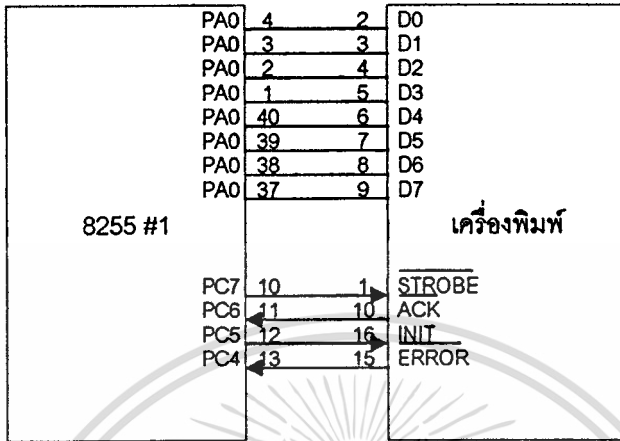
ภาพที่ 51

แสดงวงจรควบคุมบนส่วนควบคุมระบบหลัก



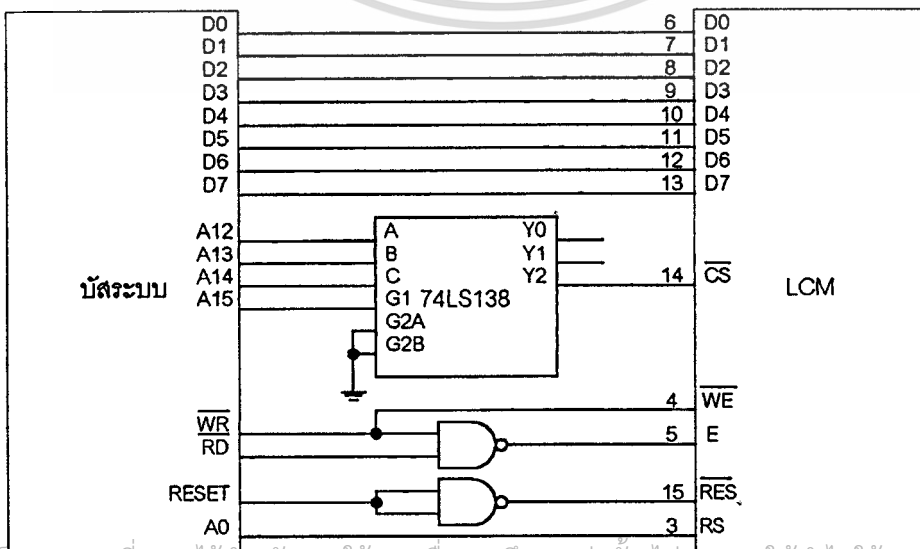
1.2 การเชื่อมต่อกับเครื่องพิมพ์ เครื่องพิมพ์จะทำหน้าที่ พิมพ์ตัวโดยสาร ซึ่งพิมพ์หมายเลขของระยะที่ขึ้นโดยสาร รวมถึง วันและเวลา การใช้ใน อีกกรณีหนึ่ง เมื่อพิมพ์รายละเอียดเกี่ยวกับการใช้บริการทั้งหมด การเชื่อมต่อกับเครื่องพิมพ์จะกระทำผ่านชิพ 8255 ตัวที่ 1 ซึ่งจะทำงานในโหมด 1 โดยมีการเชื่อมต่อขา สัญญาดังภาพที่ 52

ภาพที่ 52  
แสดงการเชื่อมต่อกับเครื่องพิมพ์



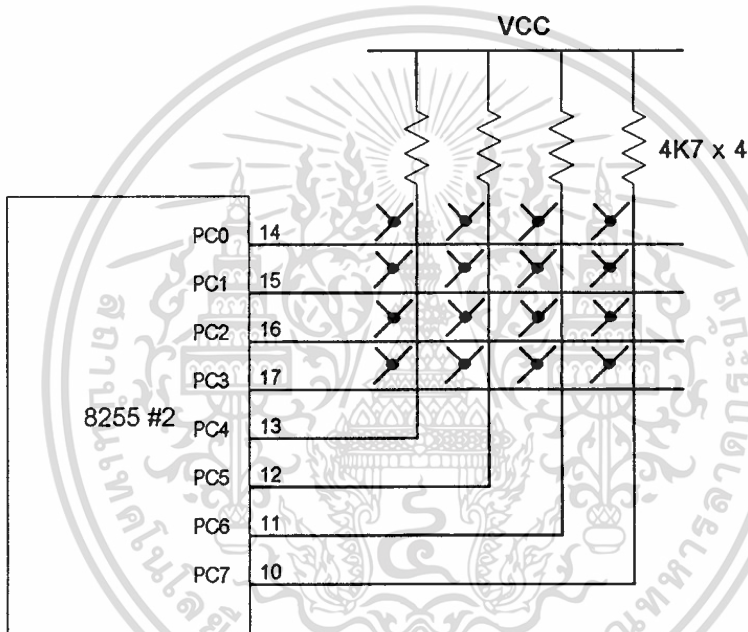
1.3 การเชื่อมต่อกับโมดูล LCD โมดูล LCD ทำหน้าที่ แสดงข้อความต่าง ๆ เช่น จำนวนเงินที่เหลือในบัตร ค่าโดยสารและข้อความอื่น ๆ การเชื่อมต่อกับโมดูล LCD นี้จะกระทำโดยวางตำแหน่งของ LCD โมดูลอยู่ในพื้นที่ของอุปกรณ์อินพุต/เอาต์พุต สัญญาณ D0 - D7 จะสามารถต่อเข้ากับดาต้าบัสของระบบได้ทันทีและสัญญาณ E จะสร้างมาจากสัญญาณ RD และ WR ผ่านแนนด์เกต สัญญาณรีเซทจะสร้างมาจากสัญญาณรีเซทของระบบโดยต่อเป็น อินพุตของแนนด์เอท ที่ต่ออยู่ใน ลักษณะของน้อทเกต และสัญญาณ RS จะใช้สัญญาณ A0 ดังวงจรในภาพที่ 53

ภาพที่ 53  
แสดงการเชื่อมต่อโมดูล LCD



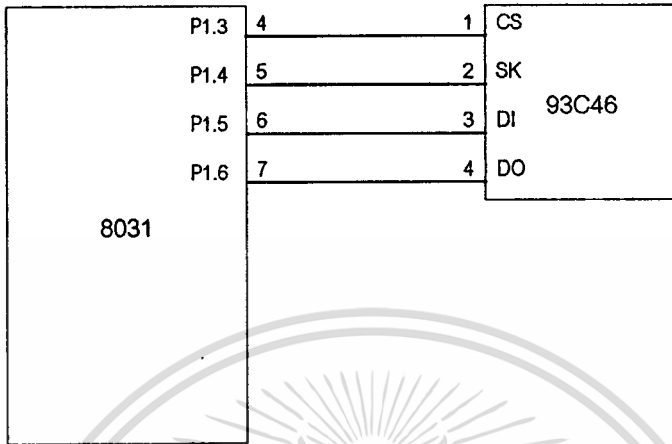
1.4 การเชื่อมต่อกับไมโครแป้นพิมพ์ ไมโครแป้นพิมพ์ ทำหน้าที่รับข้อมูล เพื่อนำไปทำงาน เช่น ปุ่มกดเพื่อรับตัวอักษร ปุ่มกดเพื่อบอกมูลค่าของบัตร โดยสาร เป็นต้น การเชื่อมต่อกับไมโครแป้นพิมพ์นี้ จะกระทำผ่านชิพสับสนุน 8255 ตัวที่ 2 โดยใช้การสแกนแบบ มัลติเพล็กซ์ ซึ่งจะต่อแต่ละแถวเข้ากับพอร์ต PC0 - PC3 และหลักเข้ากับ PC4 - PC7 ดังภาพที่ 54

ภาพที่ 54  
แสดงการเชื่อมต่อแป้นพิมพ์



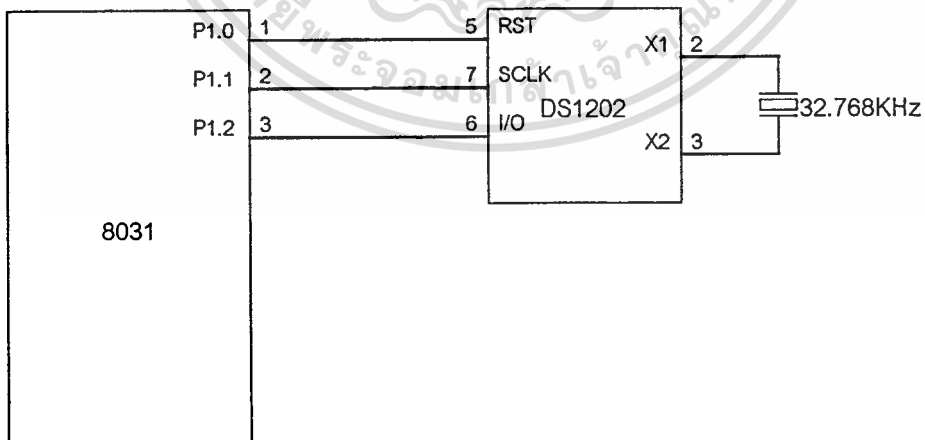
1.5 การเชื่อมต่อกับไมโครอ่านเขียนบัตรหน่วยความจำ ไมโครอ่านเขียนบัตรหน่วยความจำ ทำหน้าที่อ่านข้อมูลจำนวนเงิน และบันทึกจำนวนเงินใหม่ที่เปลี่ยนแปลงลงในบัตร การเชื่อมต่อกับไมโครอ่านเขียนบัตรหน่วยความจำนี้ จะใช้พอร์ตภายในของ 8031 คือ P 1.0 - P 1.3 เพื่อ สร้างสัญญาณ CS, DI, DO และ CLK ให้กับ EEPROM ที่อยู่บนบัตรเพื่อทำการอ่านและเขียนข้อมูลกับบัตร แสดงดังภาพที่ 55

ภาพที่ 55  
แสดงการเชื่อมต่อกับบัตรหน่วยความจำ



1.6 การเชื่อมต่อกับไมโครฐานเวลาจริง ไมโครฐานเวลาจริง ทำหน้าที่เป็นฐานเวลาอ้างอิงให้กับระบบ การเชื่อมต่อกับไมโครฐานเวลาจริง จะใช้พอร์ตภายในของ 8031 คือ P 1.4 - 1.6 เพื่อสร้างสัญญาณ RST,IO, และ SCLK ให้กับชิพฐานเวลา แสดงดังภาพที่ 56

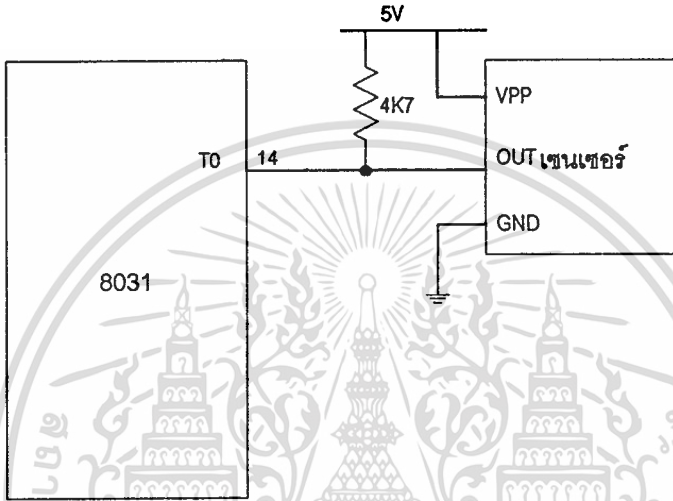
ภาพที่ 56  
แสดงการเชื่อมต่อกับชิพฐานเวลาจริง



1.7 เซนเซอร์ระยะทาง ไมโครเซนเซอร์วัดระยะทาง ทำหน้าที่ส่งสัญญาณพัลส์ 1 ลูก มาให้กับวงจรควบคุมเมื่อล้อหมุนไปครบ 1 รอบเซนเซอร์ที่ใช้งานเพื่อวัดระยะทาง จะมีเอาต์พุต เป็นลักษณะคอลเลคเตอร์เปิด (Open Collector) ซึ่งจะต้องต่อ ตัวต้านทาน पुलอัฟ (Pull up) โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

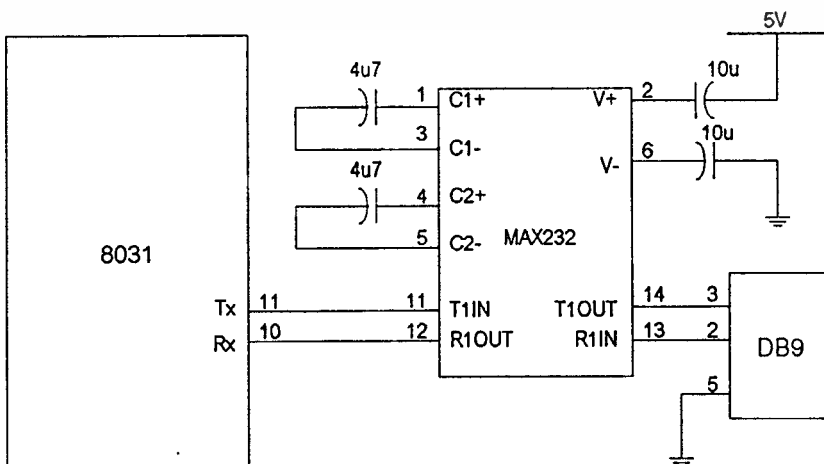
เลือกค่าเป็น 4.7 กิโลโหม์ และต่อสายสัญญาณนี้เข้าที่ขา T0 ของ 8031 เพื่อทำการนับ จำนวนพัลส์ที่ส่งออกมาจากเซนเซอร์ วงจรแสดงดังภาพที่ 57

ภาพที่ 57  
แสดงการเชื่อมต่อกับเซนเซอร์



1.8 ส่วนติดต่อข้อมูลอนุกรม โมดูลการส่งข้อมูลอนุกรม ทำหน้าที่เปลี่ยนระดับแรงดันให้เป็นมาตรฐาน RS232 เพื่อติดต่อกับส่วนอื่นๆ ในลักษณะอนุกรม วงจรแสดงดังภาพที่ 58

ภาพที่ 58  
แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนควบคุมหลัก

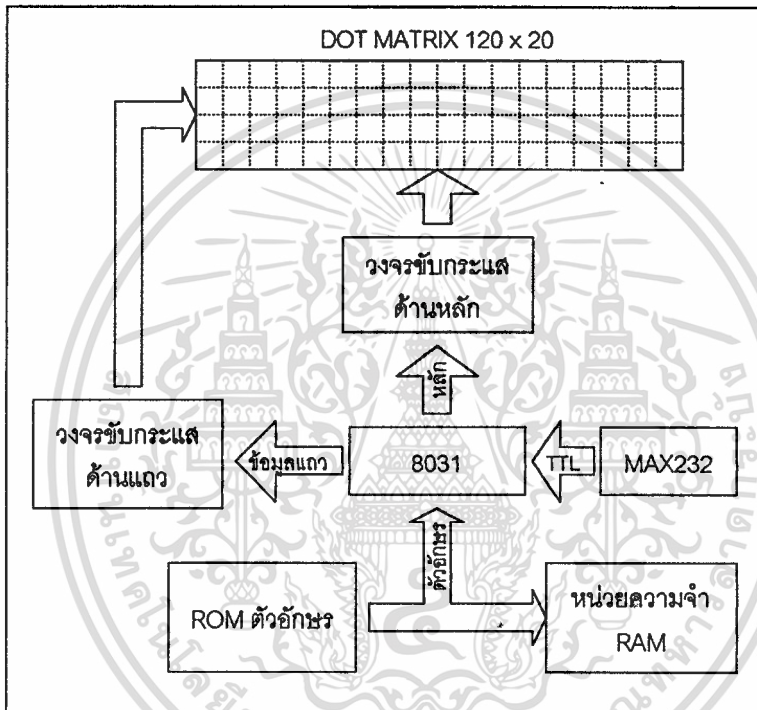


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนแสดงผลแบบจุดเมตริกซ์ ทำหน้าที่ แสดงข้อความต่าง ๆ ได้แก่ เส้นทางเดินรถ ตำแหน่งปัจจุบันของการ เดินทาง ราคาค่าโดยสาร และอื่น ๆ การทำงานของส่วนแสดงผลแบบจุดเมตริกซ์ จะแสดงดังไดอะแกรมแผนภูมิที่ 59

### แผนภูมิที่ 59

แสดงบล็อกไดอะแกรมส่วนแสดงผลแบบจุดเมตริกซ์

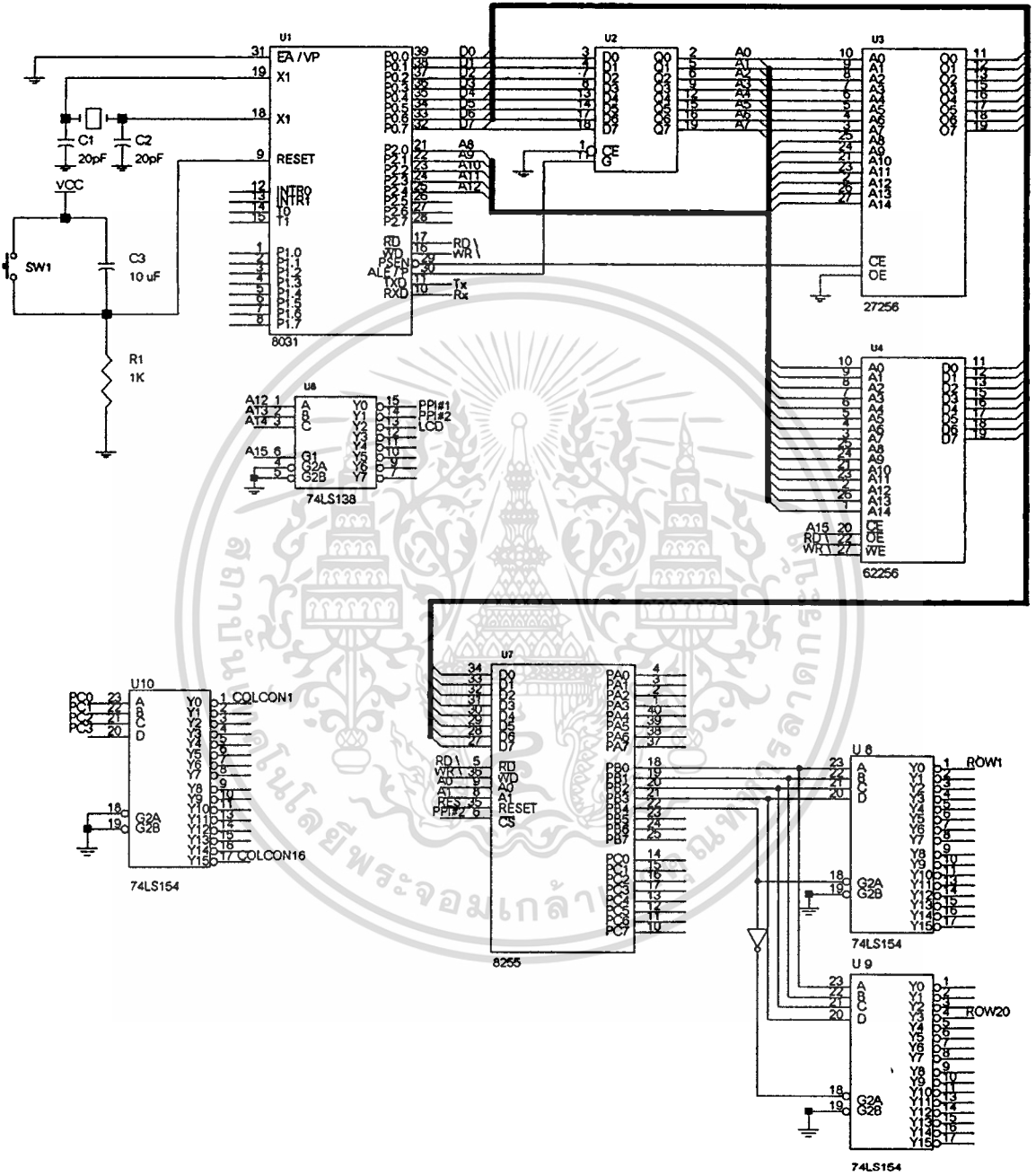


จากไดอะแกรมจะเห็นว่าการทำงานนั้นจะแบ่งเป็น 2 ส่วนหลัก คือวงจรควบคุมซึ่งทำหน้าที่รับข้อมูลรหัสแอสกี(ASCII) จากภายนอก เพื่อนำมาแปลงเป็นข้อมูลที่จะนำไปแสดงผล ส่วนที่สองคือ วงจรชุดขับกระแส (Driver) ทำหน้าที่รับกระแสเพื่อ ให้ LED สว่าง ดังนั้นการออกแบบวงจรทั้งสองส่วนจะเป็นดังนี้

2.1 การออกแบบวงจรควบคุม การทำงานของวงจรทั้งหมดจะควบคุมด้วย 8031 โดยจะมีการจัดวางหน่วยความจำ และอินพุต/เอาต์พุตดังตารางที่ 35 และวงจรแสดงดังภาพที่ 60

### ภาพที่ 60

แสดงวงจรควบคุมของส่วนแสดงผลแบบจุดเมตริกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 35

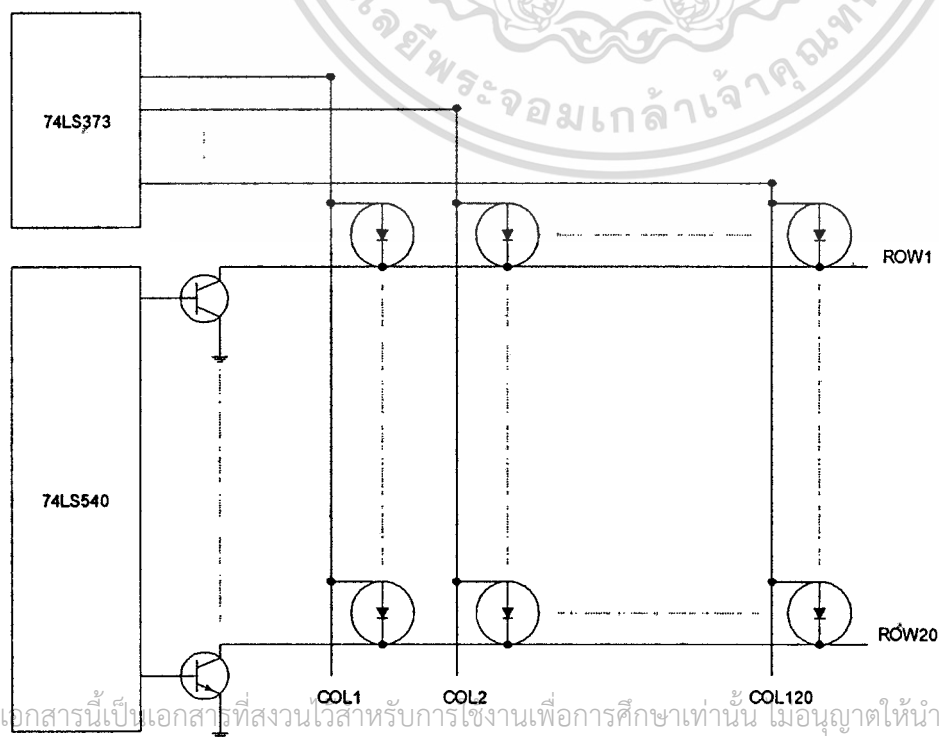
แสดงการจัดวางตำแหน่งของหน่วยความจำและอุปกรณ์อินพุต/เอาต์พุต  
บนวงจรควบคุมการแสดงผลแบบจุดเมตริกซ์

อุปกรณ์	ตำแหน่ง
ROM	0000H - 7FFFH (Program memory)
RAM	0000H - 7FFFH (Data memory)
8255#1	8000H - 8003H (I/O)

2.2 การออกแบบวงจรชุดขับกระแส ชุดขับกระแสจะแบ่งออกเป็น 2 ส่วน คือ ด้านแถว (ROW) และส่วนหลัก (COLUMN) ส่วนด้านหลักนั้น จะใช้ ไอซี 74LS373 เพื่อค้างข้อมูลที่แสดงผลในแต่ละแถว ส่วนวงจรชุดขับกระแสด้านแถวจะใช้ไอซี 74LS540 ขับกระแสต่อก่อนที่จะใช้ทรานซิสเตอร์ เพื่อ ขับกระแสของ LED ทั้งแถวซึ่งประมาณ 1 แอมป์ โดยมีวงจรดังภาพที่ 61

ภาพที่ 61

แสดงวงจรชุดขับกระแสของส่วนแสดงผลแบบจุดเมตริกซ์

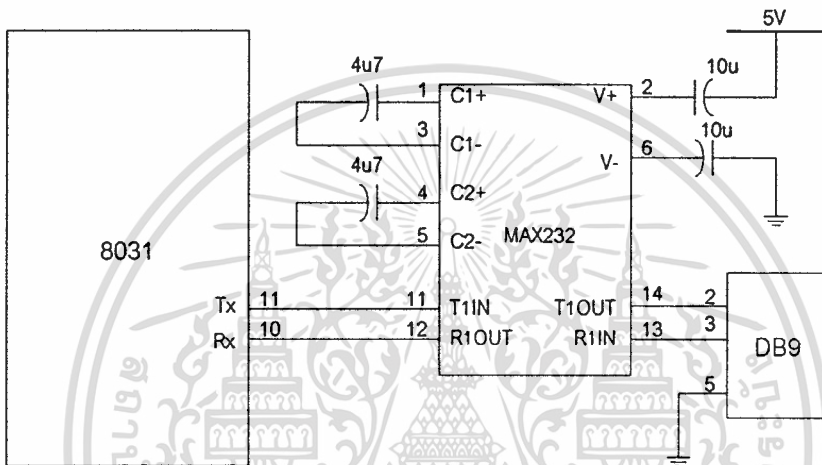


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนติดต่อข้อมูลอนุกรม ส่วนนี้จะทำหน้าที่เปลี่ยนระดับแรงดันของสัญญาณจาก TTL ให้อยู่ในสัญญาณมาตรฐาน RS232 เพื่อติดต่อกับส่วนควบคุมระบบหลัก ในลักษณะอนุกรมเพื่อรับข้อความที่จะแสดงผล รวมถึงรหัสควบคุมต่าง ๆ วงจรแสดงดังภาพที่ 62

ภาพที่ 62

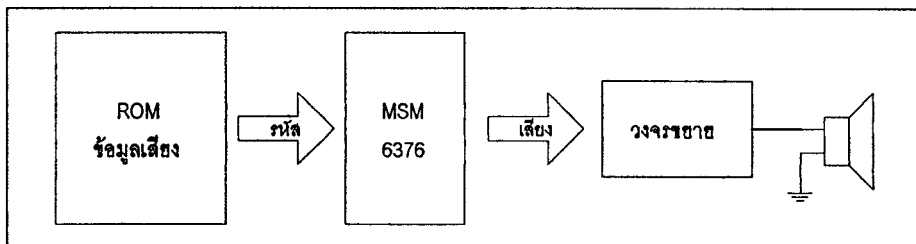
แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนการแสดงผล



3. ส่วนแสดงผลด้านเสียงพูด ทำหน้าที่ พูดข้อความต่าง ๆ เช่นเดียวกับ ส่วนแสดงผลแบบจุดเมตริกซ์ โดยการรับข้อมูลรหัสมาจาก ส่วนควบคุมระบบหลัก การทำงานของส่วนแสดงผลจะแสดงดังบล็อกไดอะแกรมแผนภูมิที่ 63

แผนภูมิที่ 63

แสดงบล็อกไดอะแกรมการแสดงผลเสียงพูด



จากไดอะแกรมจะเห็นว่าการทำงานนั้นจะแบ่งเป็น 3 ส่วนที่เชื่อมต่อกัน คือชิพสังเคราะห์

เอเสียง (Speech Synthesis) หน่วยความจำ และวงจรรายชากำลัง ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

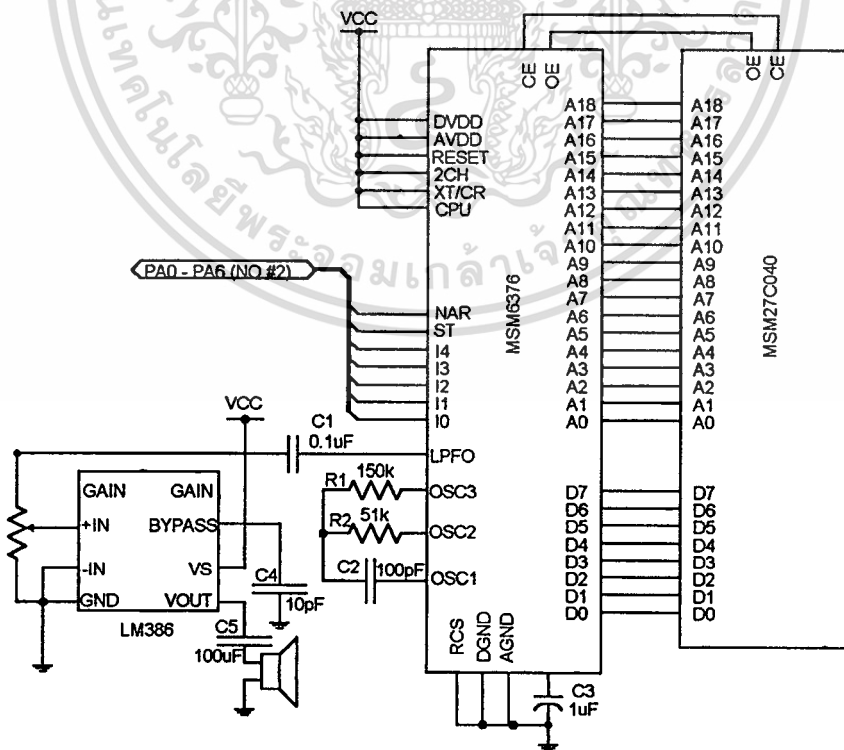
3.1 วงจรชีพสังเคราะห์เสียง จะใช้ชีพสังเคราะห์เสียงของบริษัท OKI เบอร์ MSM 6376 จะทำหน้าที่รับสัญญาณรหัสคำพูดมาจากภายนอก หรือสวิตช์ จากนั้นก็จะส่งสัญญาณไปอ่านข้อมูลที่ตรงกับรหัสคำพูดเพื่อสร้างสัญญาณเสียงออกที่เข้าที่พหูพชา LPFO ความถี่ของออลซิลเลเตอร์นั้นจะใช้จากวงจรรภายใน โดยต่อตัวต้านทานและตัวเก็บประจุภายนอกเพิ่มเติม แสดงดังวงจรรภาพที่ 64

3.2 หน่วยความจำ จะใช้หน่วยความจำ ROM ขนาด 4M บิต เบอร์ 27C040 ซึ่งทำหน้าที่เก็บข้อมูลเสียงที่มีการเข้ารหัสแบบ ADPCM ซึ่งสามารถเชื่อมต่อกัน MSM 6376 ได้ทันที ดังวงจรรภาพที่ 64

3.3 วงจรขยายกำลัง จะใช้ออปแอมป์เบอร์ LM 386 เพื่อขยายสัญญาณออกลำโพงต่อไป เนื่องจากว่า สัญญาณที่ออกจากขา LPFO ได้มีการกรองสัญญาณความถี่ต่ำแล้ว จึงขยายกำลังเพื่อส่งออกลำโพง 8 โอห์มได้โดยไม่ต้องสร้างวงจรรองอีก ดังภาพที่ 64

ภาพที่ 64

แสดงวงจรรของส่วนแสดงผลเสียงพูด



## การออกแบบทางซอฟต์แวร์

การออกแบบซอฟต์แวร์เพื่อควบคุมการทำงานทั้งระบบ จะแบ่งการทำงานเป็น 2 ส่วน คือ โปรแกรมบนส่วนควบคุมหลัก และ โปรแกรมบนส่วนแสดงผล เนื่องจากแต่ละส่วนจะใช้ไมโครโปรเซสเซอร์ควบคุมการทำงาน โดยจะมีการติดต่อข้อมูลผ่านทาง พอร์ตอนุกรม

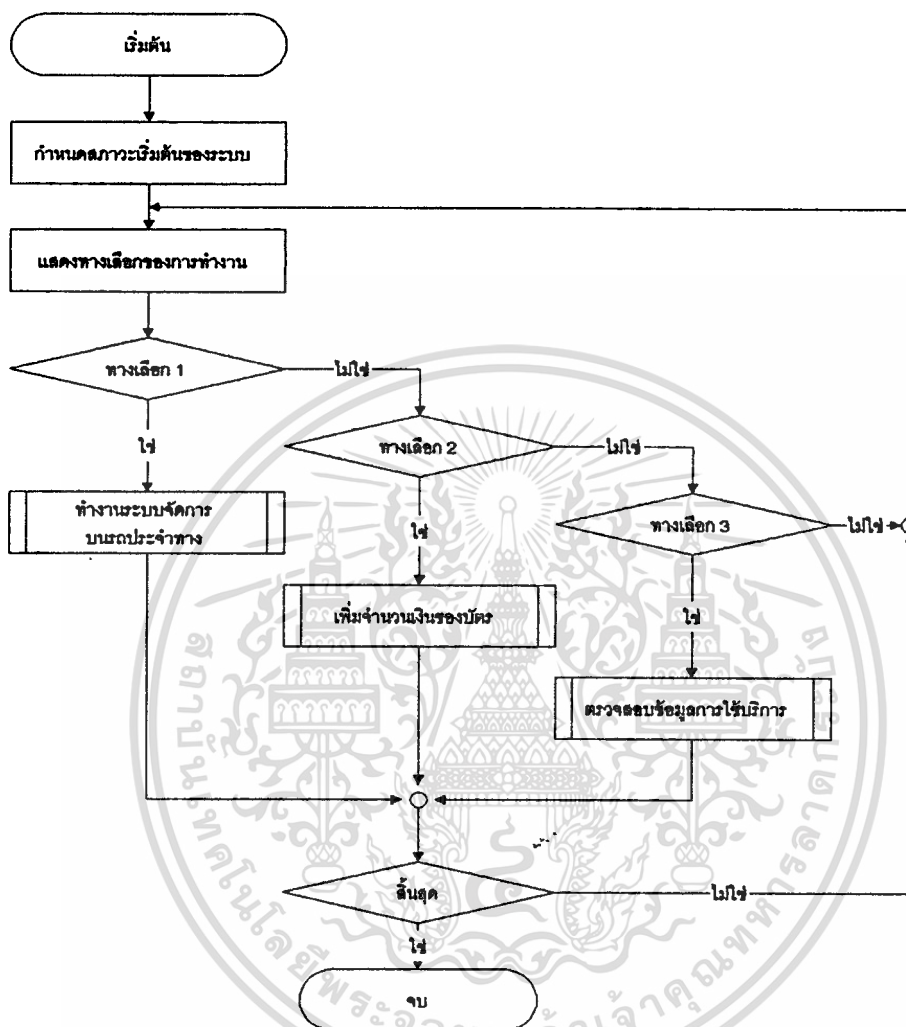
1 โปรแกรมบนส่วนควบคุมหลัก โปรแกรมบนส่วนควบคุมหลักจะทำหน้าที่ ควบคุมการทำงานของระบบทั้งหมด โดยมีหน้าที่ คำนวณระยะทาง แสดงผลด้านเสียง และส่งข้อมูล เพื่อแสดงข้อความระยะทางต่างๆ การหักค่าโดยสารจากบัตรหน่วยความจำ การโปรแกรมมูลค่าของบัตร เพิ่มเติม และ การตรวจสอบข้อมูลการใช้บริการในแต่ละเที่ยวโดยสาร

1.1 โปรแกรมหลัก จะเริ่มต้นโดยการกำหนดสถานะทางฮาร์ดแวร์ ให้พร้อมที่จะทำงาน และเคลียร์ค่าของบัพเฟอร์ และแฟลทที่ ใช้งานในโปรแกรมใหม่ เช่น บัพเฟอร์ระยะทาง เป็นต้น เมื่อกำหนดสถานะต่างๆ เสร็จก็จะทำการแสดงทางเลือกของการทำงาน เป็น 3 ลักษณะ คือ การทำงานระบบจัดการ การทำงานเพิ่มค่าจำนวนเงินของบัตร หรือ การทำงานตรวจสอบข้อมูลในเที่ยว โดยสารเมื่อมีการเลือกก็จะไปทำงานตามโปรแกรมย่อยเหล่านั้น โฟลว์ชาร์ตการทำงาน แสดงดังแผนภูมิที่ 65



## แผนภูมิที่ 65

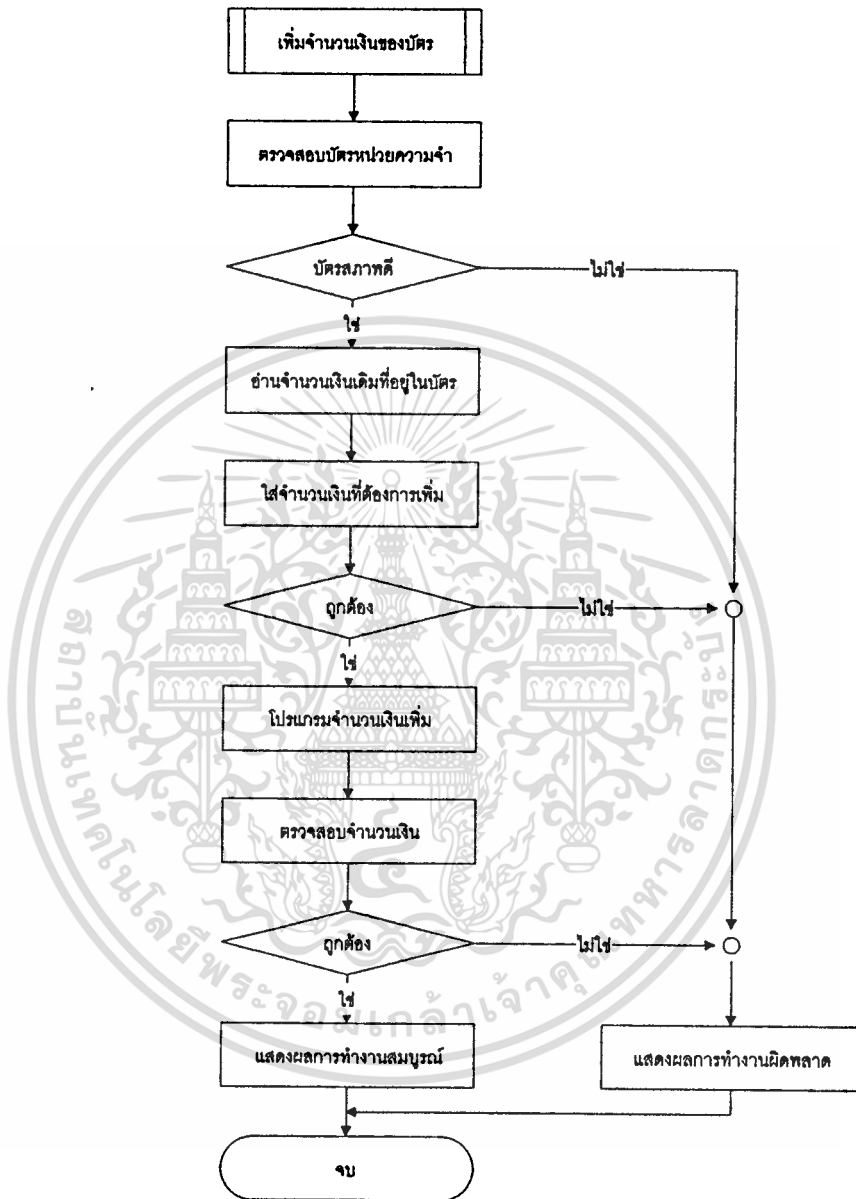
### แสดงโฟลว์ชาร์ตแสดงการทำงานของโปรแกรมหลัก



1.2 โปรแกรมย่อยส่วนเพิ่มค่าจำนวนเงินของบัตร การทำงานของโปรแกรมย่อยส่วนนี้จะเริ่มต้นโดยการตรวจสอบบัตรหน่วยความจำที่ใช้ว่าถูกต้องหรือไม่ โดยการอ่านรหัสที่ได้มีการโปรแกรมไว้ก่อน ถ้ารหัสที่อ่านมาได้ถูกต้องก็จะอ่านจำนวนเงินที่มีอยู่ในบัตรออกมาแสดง แล้วทำการใส่จำนวนเงินที่ ต้องการจะเพิ่มเข้าไป เมื่อป้อนจำนวนเงินถูกต้องแล้ว ก็จะทำการบวกจำนวนเงินเดิมเข้ากับจำนวนเงินใหม่ แล้วจะทำการตรวจสอบ จำนวนเงินที่ได้โปรแกรมใหม่บนบัตร ว่า การโปรแกรมเกิดข้อผิดพลาดหรือไม่ ถ้าไม่เกิดข้อผิดพลาดก็จะแสดงข้อความการ ทำงานเสร็จสมบูรณ์ แต่ถ้าการทำงานในขั้นตอนใดมีข้อผิดพลาดเกิดขึ้นก็จะแสดงข้อผิดพลาดที่เกิดขึ้น

## แผนภูมิที่ 66

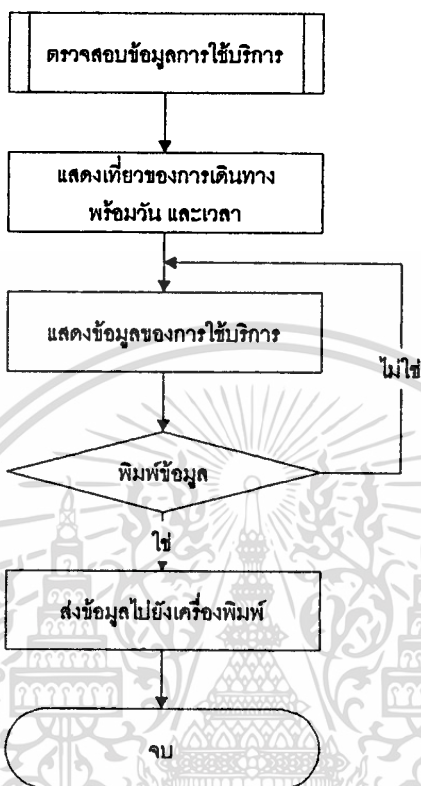
### แสดงไฟล์ชาร์ตการเพิ่มจำนวนเงินในบัตร



1.3 โปรแกรมย่อยส่วนตรวจสอบข้อมูลการใช้ บริการ โปรแกรมส่วนนี้ทำหน้าที่ตรวจสอบข้อมูลของการใช้บริการ โดยจะอ่านข้อมูลจากหน่วยความจำ ซึ่งข้อมูลเหล่านี้จะถูก บันทึกไว้ขณะที่ เริ่มต้นทำงานระบบ จนกระทั่ง เสร็จสิ้นการทำงาน จากนั้นถ้ามีการสั่งให้พิมพ์ข้อมูลเหล่านี้ ก็จะทำ การส่งข้อมูลไปยัง เครื่องพิมพ์

## แผนภูมิที่ 67

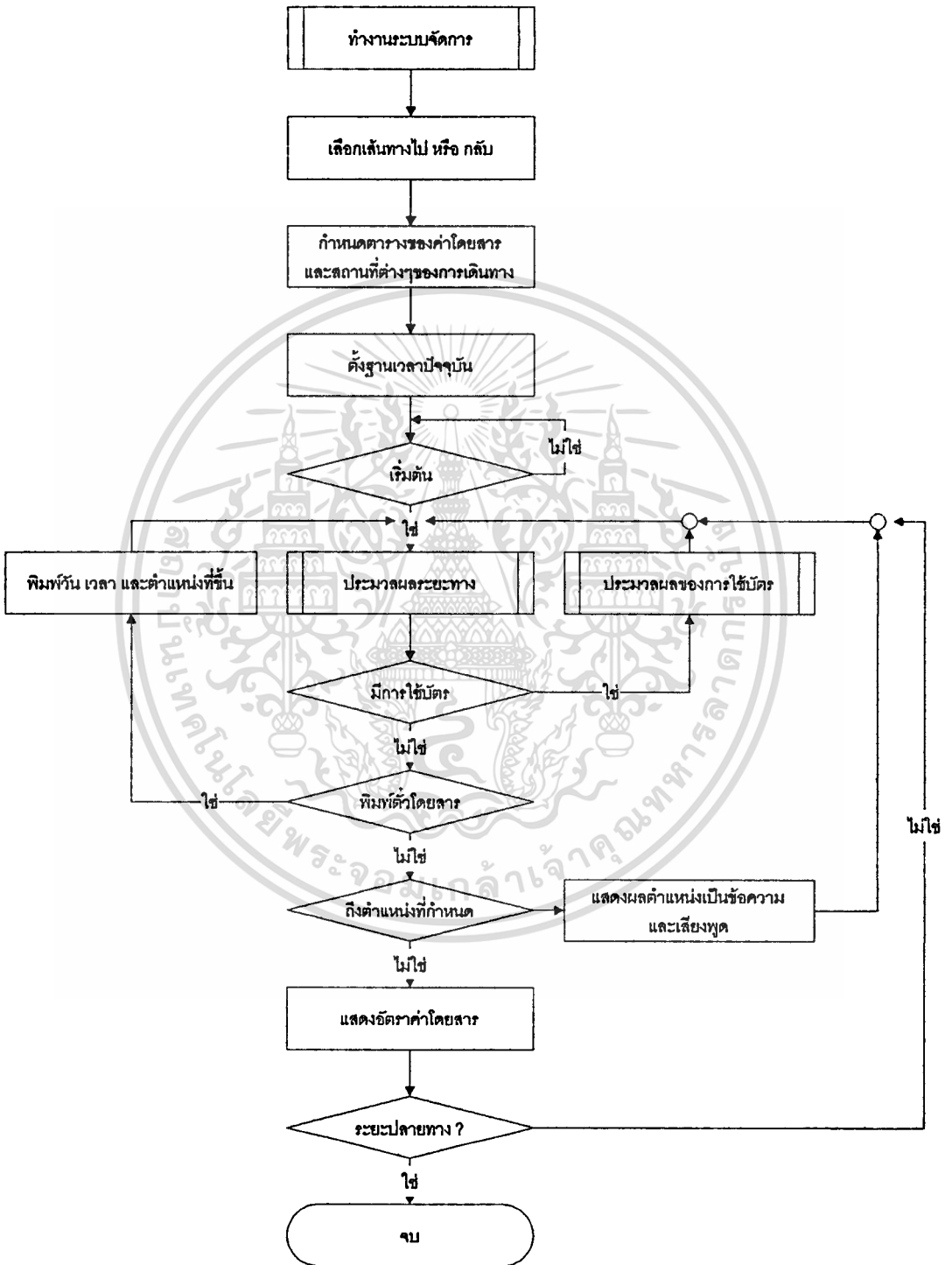
แสดงไฟล์ข่าวสารการตรวจสอบข้อมูลการใช้บริการ



1.4 โปรแกรมส่วนระบบจัดการ การทำงานโปรแกรมส่วนนี้จะเริ่มต้นโดยการเลือกเส้นทางของการเดินทาง เพื่อที่จะนำไปเลือกตารางข้อมูลเกี่ยวกับ เส้นทางการเดินทางโดยสาร จากนั้นจะเข้าสู่การตั้งเวลาของระบบให้ตรงกับเวลาปัจจุบัน โดยการทำงานทั้งหมดจะอ้างอิงกับฐานเวลาในตอนเริ่มต้น เช่นเวลาในการเริ่มต้นระบบ เมื่อถึงจุดหมาย และอื่นๆ หลังจากนั้นจะเริ่มทำงานระบบโดยจะเริ่มคำนวณระยะทางที่ รถเคลื่อนที่ และจะมีการตรวจสอบการทำงาน 2 ลักษณะคือถ้ามีการใช้บัตรหน่วยความจำ ก็จะเข้าไปประมวลผลการใช้บัตร แต่ถ้าไม่ ใช้บัตร ก็จะกดปุ่มเพื่อพิมพ์ตัว เครื่องก็จะพิมพ์ข้อมูล วัน เวลา และตำแหน่งที่ขึ้นโดยสาร จากนั้นก็จะทำการตรวจสอบว่า ถึงจุดที่สำคัญต่างๆ ถ้าถึงตำแหน่งที่กำหนดแล้วก็จะมีการแสดงตำแหน่งเป็นข้อความและเสียงพูด และจะมีการแสดงอัตราค่าโดยสารของตัวระยะต่างๆ การทำงานจะดำเนินไปจนกระทั่งถึงจุดปลายทางของเส้นทาง

## แผนภูมิที่ 68

### แสดงโฟลว์ชาร์ตการทำงานระบบจัดการ

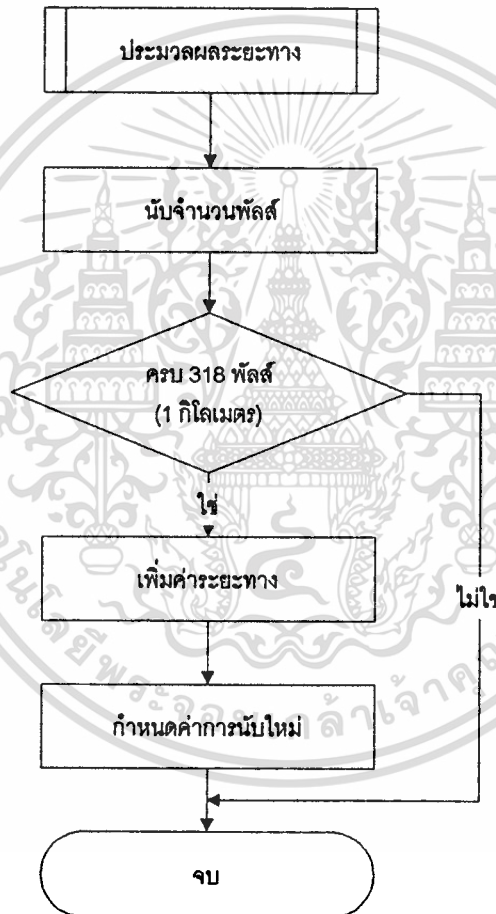


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.1 โปรแกรมย่อยประมวลผลการนับระยะทาง การประมวลผลระยะทางจะทำการนับพัลส์ที่ได้จากเซนเซอร์จากการหมุนของล้อ 1 รอบ และจะตรวจสอบจำนวนพัลส์ ที่ได้รับว่าครบ 318 พัลส์ หรือประมาณ 1 กิโลเมตร ถ้าครบก็จะทำการเพิ่มค่าของระยะทางขึ้น และทำการกำหนดค่าของวงจรมับใหม่เพื่อที่จะเริ่มต้นนับจำนวนพัลส์ใหม่

### แผนภูมิที่ 69

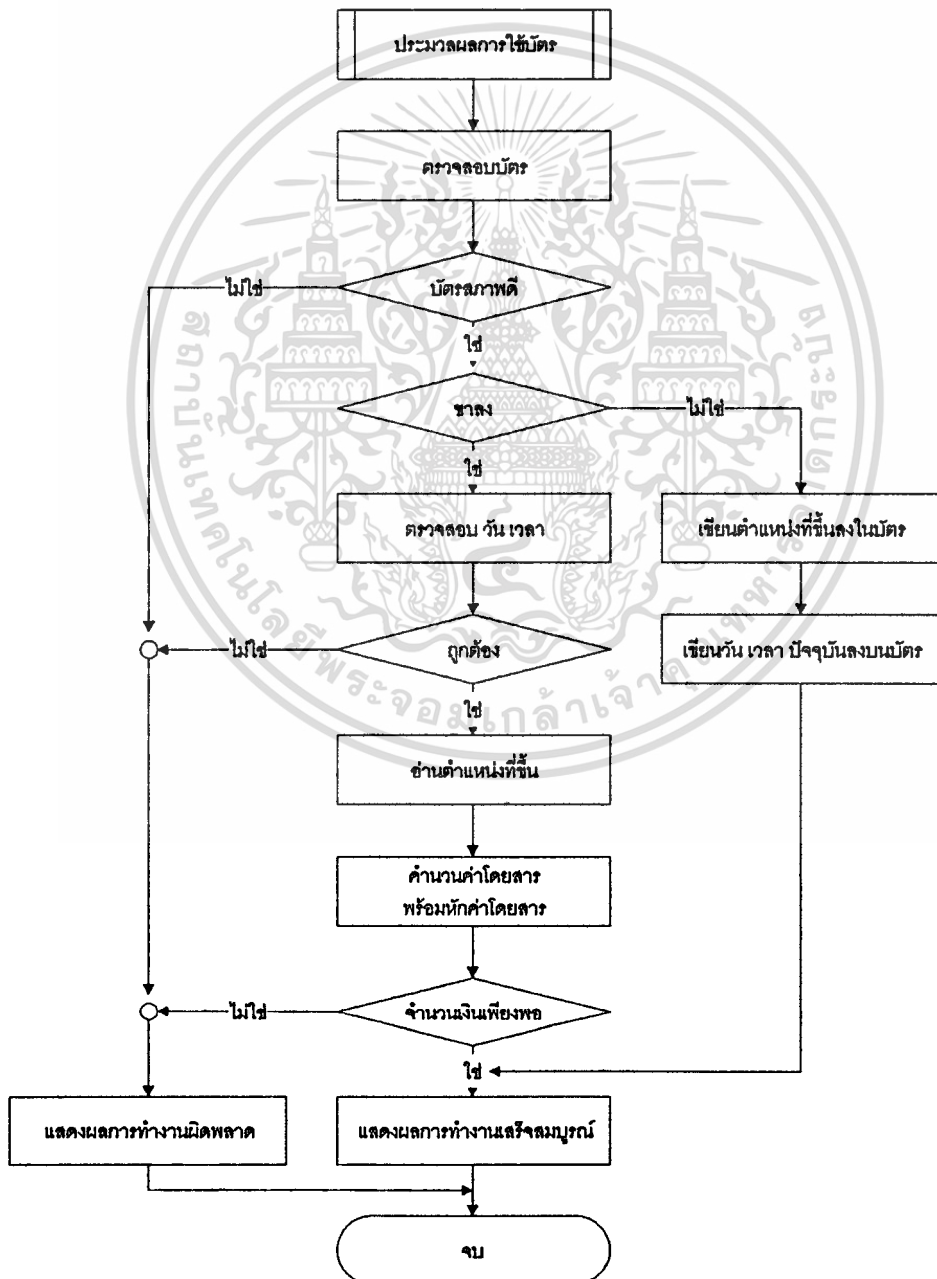
#### แสดงโฟลว์ชาร์ตการคำนวณระยะทาง



1.4.2 โปรแกรมย่อยประมวลผลการใช้บัตร การทำงานจะเริ่มต้นโดยการตรวจสอบบัตรหน่วยความจำ ว่าอยู่ในสภาพที่ใช้งานดีหรือไม่ โดยการอ่าน ข้อมูล จาก ตำแหน่งที่กำหนดแล้วเปรียบเทียบกับไบต์ทดสอบ ถ้าอยู่ในสภาพดีก็จะตรวจสอบว่าการสอด้บัตรนั้นเป็นขาขึ้นหรือขาลง ถ้าเป็นขาขึ้น ก็จะทำการบันทึกวัน เวลา รวมถึงตำแหน่งที่ขึ้น จากนั้นก็แสดงข้อความการทำงานเสร็จสมบูรณ์ เพื่อดึงบัตรออกจากช่อง เสียบ แต่ถ้าเป็นการสอด้บัตรขาลง ก็จะทำการอ่านวันและ เวลา ที่ขึ้น เปรียบเทียบกับเวลาปัจจุบัน ว่าถูกต้องสัมพันธ์กัน หรือไม่ถ้าถูกต้อง ก็ดำเนินการคำนวณไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะอ่านตำแหน่งที่ขึ้นเพื่อคำนวณค่าโดยสาร จนถึงระยะปัจจุบัน แล้วจะตรวจสอบว่าภายในบัตร นั้นมีจำนวนเงินเพียงพอหรือไม่ ถ้ามีเพียงพอก็จะหักค่าโดยสารจากบัตร พร้อมแสดงจำนวนเงิน ที่เหลืออยู่ในบัตร และแสดงข้อความเสร็จสมบูรณ์ส่วนการคิดพลาด จากการตรวจสอบที่ส่วน ต่าง ๆ จะมีการแสดงผลตามข้อผิดพลาดแต่ละแบบที่เกิดขึ้น แสดงผังแผนภูมิที่ 70

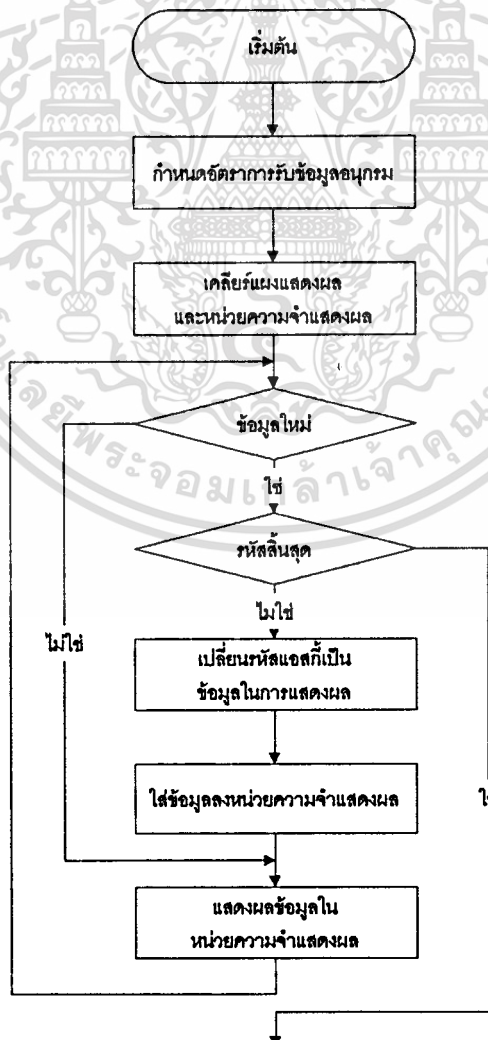
### แผนภูมิที่ 70 แสดงโฟลว์ชาร์ตการใช้บัตรหน่วยความจำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมบนส่วนแสดงผลจุดเมตริกซ์ การทำงานจะเริ่มต้นโดยการกำหนดอัตราการรับข้อมูลอนุกรมเป็น 9600 บิต/วินาที เพื่อรับข้อมูลที่จะถูกส่งมาจากส่วนควบคุมหลัก จากนั้นจะทำการเคลียร์แผงแสดงผลโดยจะให้ ไดโอดเปล่งแสง (LED)ดับทุกดวง และเคลียร์ค่าในหน่วยความจำ แสดงผลให้เป็นศูนย์ และจะเริ่มรับข้อมูลที่ส่งมาจากส่วนควบคุมหลัก โดยข้อมูลที่ส่งมาจะประกอบด้วย รหัสควบคุม และข้อมูลซึ่ง รหัสควบคุมจะควบคุมการทำงานของแผงแสดงผลว่าให้ทำงานหรือไม่ ส่วนข้อมูลจะเป็นรหัสแอสกี ส่วนรหัสภาษาไทยจะใช้ของสมาคมมาตรฐานอุตสาหกรรมไทย (สมอ) เมื่อรับข้อมูลเข้ามาก็จะทำการแปลงรหัสแอสกีให้เป็นข้อมูลที่จะนำไปแสดงผลและเก็บ ลงหน่วยความจำแสดงผล

แผนภูมิที่ 71  
แสดงโฟลว์ชาร์ตการทำงานบนส่วนแสดงผลจุดเมตริกซ์



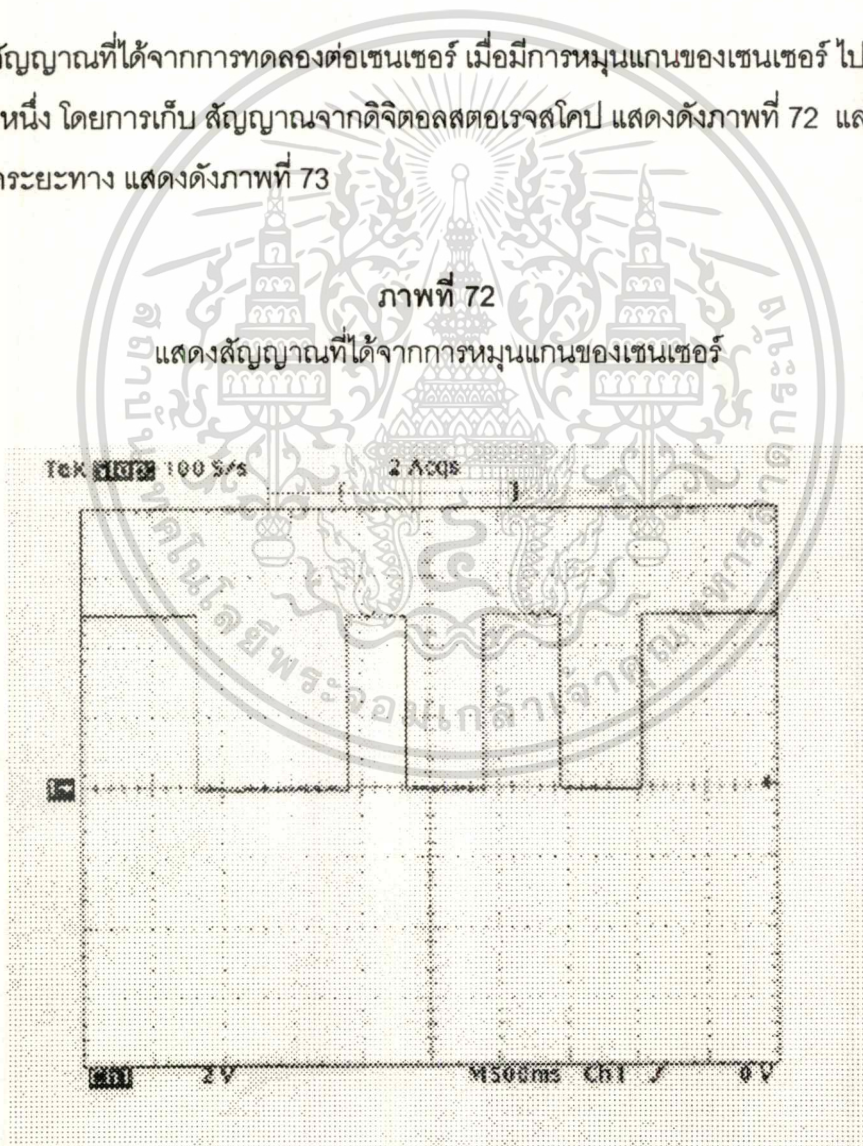
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

#### การทดลองเชื่อมต่อเซนเซอร์

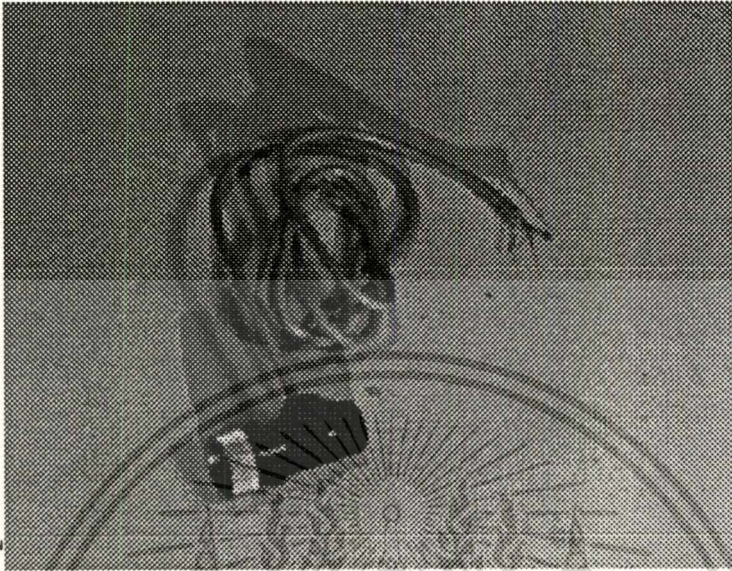
สัญญาณที่ได้จากการทดลองต่อเซนเซอร์ เมื่อมีการหมุนแกนของเซนเซอร์ ไปด้วยความเร็ว ระดับหนึ่ง โดยการเก็บ สัญญาณจากดิจิตอลสโตเรจสโคป แสดงดังภาพที่ 72 และเซนเซอร์ สำหรับวัดระยะทาง แสดงดังภาพที่ 73



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ภาพที่ 73

แสดงเซนเซอร์สำหรับวัดระยะทาง

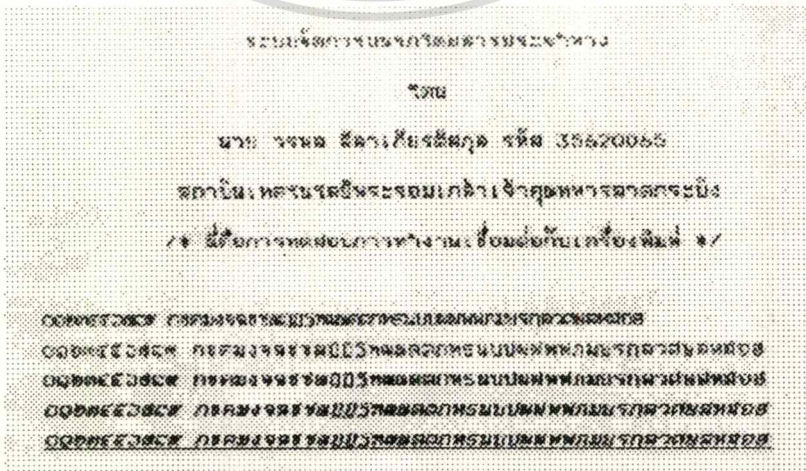


### การทดลองเชื่อมต่อเครื่องพิมพ์

การทดลองจะทำโดยการเชื่อมต่อเครื่องพิมพ์กับส่วนควบคุมหลักโดยใช้คำสั่งเพื่อให้เครื่องพิมพ์ ทำการพิมพ์ตัวอักษรในลักษณะต่างๆดังภาพที่ 74

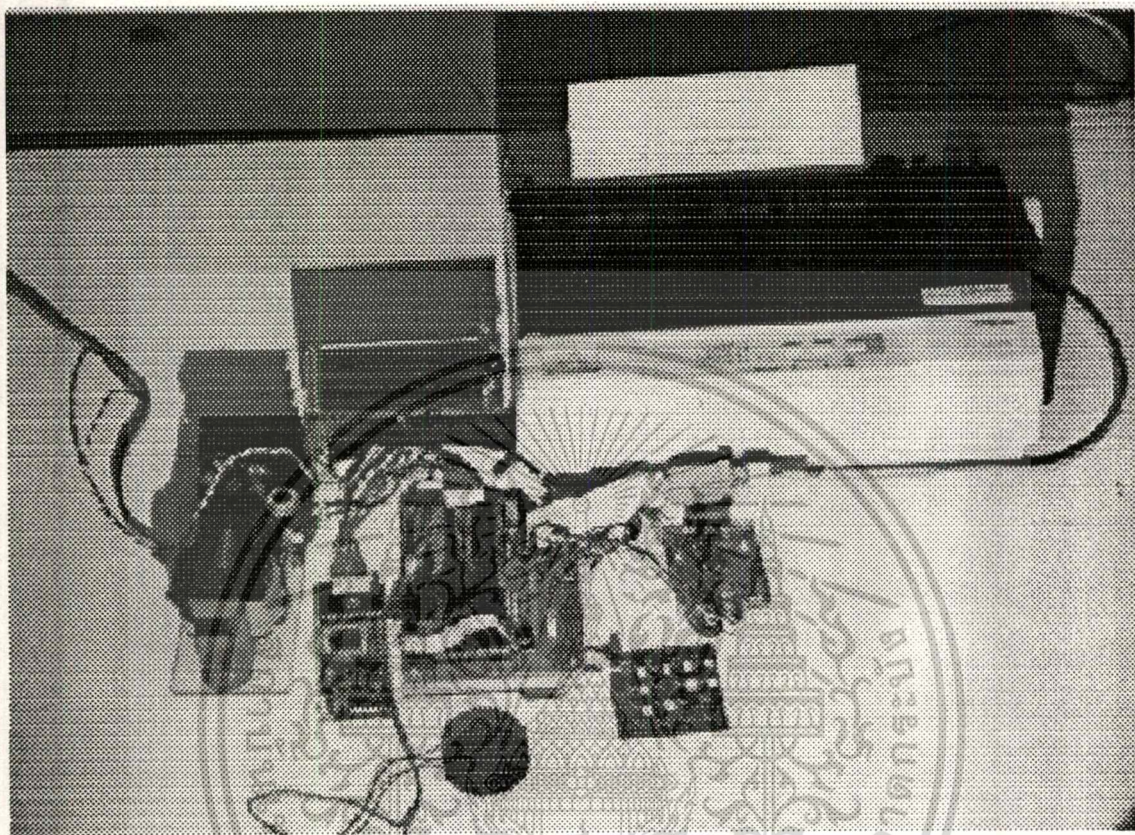
### ภาพที่ 74

แสดงผลการทดลองพิมพ์ในรูปแบบต่าง ๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 75  
แสดงการเชื่อมต่อเครื่องพิมพ์



การทดลองเชื่อมต่อโมดูล LCD

การทดลองจะทำการเชื่อมต่อกับ LCD กับส่วนควบคุมหลัก โดยการแสดงผลบนจอ LCD เป็นภาษาไทยและ ภาษาอังกฤษแสดงดังภาพที่ 76 และการเชื่อมต่อจอ LCD แสดงดังภาพที่ 77

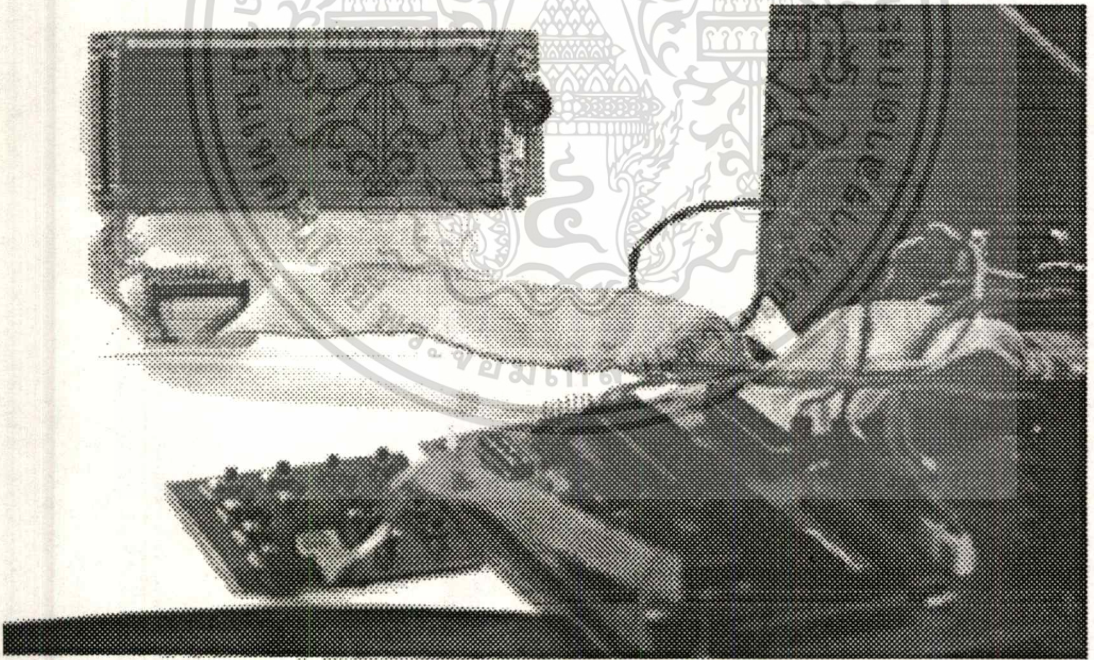
## ภาพที่ 76

แสดงผลการทดลองแสดงตัวอักษรบนจอ LCD



## ภาพที่ 77

แสดงการเชื่อมต่อจอ LCD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทดลองเชื่อมต่อกับชิพฐานเวลาจริง

การทดลองจะทำโดยการเชื่อมต่อกับชิพฐานเวลาจริง กับส่วนควบคุมหลัก โดยการกำหนดฐานเวลา เช่น วัน เดือน ปี และ เวลา แสดงดังภาพที่ 78

ภาพที่ 78

ผลการทดลองกำหนดเวลาลงในชิพฐานเวลา



## การทดลองเชื่อมต่อกับ EEPROM

การทดลองจะทำโดยการเชื่อมต่อกับ EEPROM กับส่วนควบคุมหลัก โดยการกำหนดจำนวนเงิน และ โปรแกรมมูลค่า ใหม่เพิ่มเติม แสดงดังภาพที่ 79

ภาพที่ 79

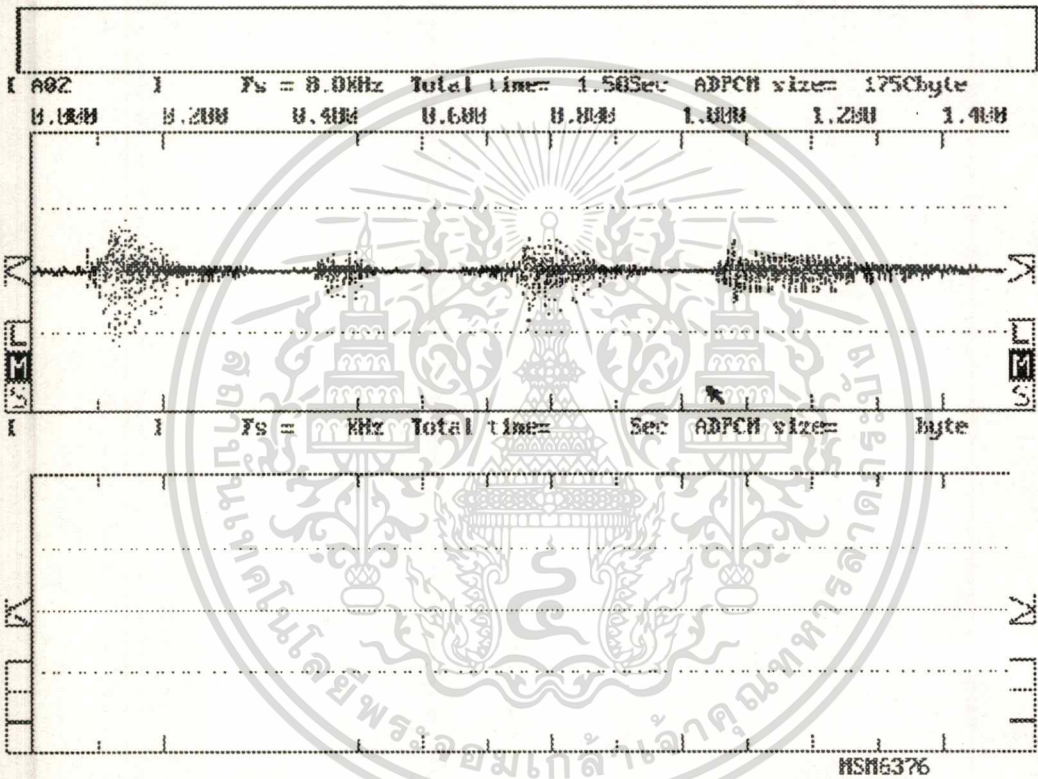
ผลการทดลองโปรแกรมมูลค่าจำนวนเงินลงบน EEPROM



## การทดลองเชื่อมต่อส่วนสังเคราะห์เสียง

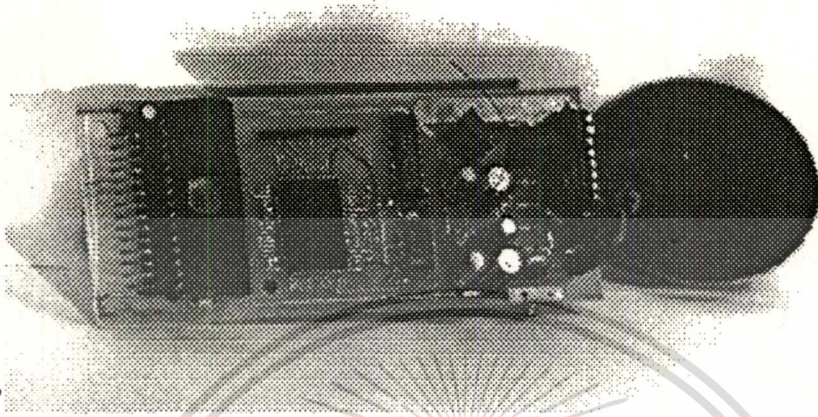
การทดลองจะทำการเชื่อมต่อกับส่วนสังเคราะห์เสียงกับส่วนควบคุมหลัก โดยแสดงรูปของสัญญาณเสียงของ คำว่า " รังสิต บางพูน" ที่สังเคราะห์ได้จาก MSM6376 แสดงดังภาพที่ 80

ภาพที่ 80  
สัญญาณเสียงที่ได้จากการสังเคราะห์



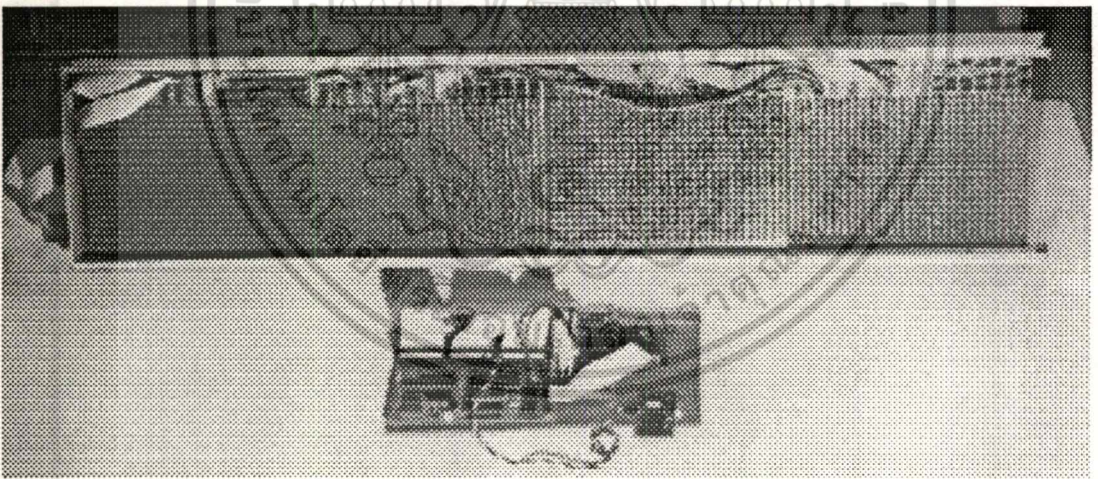
## ภาพที่ 81

วงจรส่วนสังเคราะห์เสียง



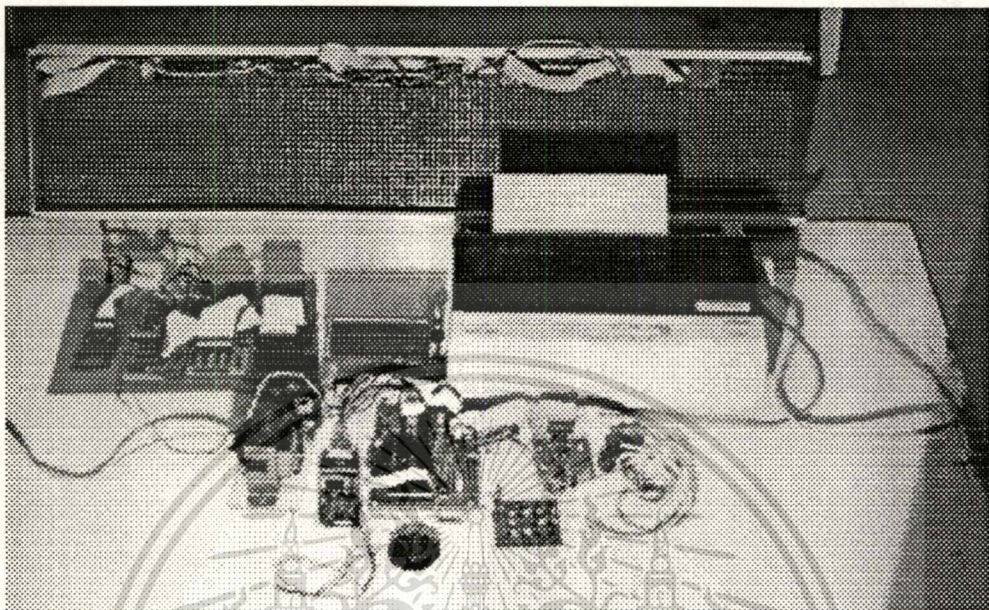
## ภาพที่ 82

ส่วนแสดงผลจุดเมตริกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 83  
แสดงต้นแบบของทั้งระบบ



## บทที่ 5

### สรุปผล

ระบบการจัดบนรถประจำทางนี้ สามารถทำงานตามที่ได้ออกแบบ คือ การคิดค่าโดยสาร และการบอกระยะทาง โดยไม่ใช้ พนักงานขายตั๋ว รวมถึงการโปรแกรมมูลค่าของบัตรหน่วยความจำ แต่ในลักษณะการใช้งานจริงจำเป็นต้องมีการปรับปรุงโครงสร้างหลายส่วนเพื่อให้เหมาะสมกับรถประจำทางแบบต่าง ๆ โดยระบบที่จัดทำขึ้นนี้ จะเหมาะสำหรับการใช้กับรถโดยสารที่มีทางขึ้นและลง เป็นทางเดียวกัน เนื่องจากมีช่องเสียบบัตรเพียงจุดเดียว และสัญญาณที่จะนำมาคำนวณระยะทางนั้น ถ้าเป็นรถโดยสารที่ใช้เครื่องยนต์ระบบหัวฉีดซึ่งไม่จำเป็นต้องต่อเซนเซอร์ เนื่องจากว่าจะมีการให้สัญญาณพัลส์เพื่อวัดระยะทาง ออกมาอยู่แล้วก็จะต้องมีการปรับจำนวนพัลส์ต่อกิโลเมตร เนื่องจากว่ารถแต่ละรุ่นจะกำหนดสัญญาณพัลส์ที่ส่งออกมาจากชุดวัดระยะทางไม่เท่ากัน

ระบบนี้สามารถที่จะออกแบบการจัดการเพิ่มเติม เช่น ให้ความคุ้มครองการเปิดปิดประตู หรือการเชื่อมต่อกับเซนเซอร์ชนิดต่าง ๆ โดยไม่จำเป็นต้องออกแบบระบบใหม่ เนื่องจากมีพอร์ตสำหรับรับสัญญาณ ส่งสัญญาณกับภายนอกเหนืออยู่ ซึ่งจะทำให้ระบบมีประสิทธิภาพเพิ่มขึ้น โดยใช้แผงควบคุมเพียงชุดเดียวก็สามารถทราบสถานะและควบคุมการทำงานของรถโดยสารได้

ในการพัฒนาระบบจัดการ มีแนวทางดังนี้

1. การจัดเก็บค่าโดยสารผ่านบัตรหน่วยความจำแบบไม่ ต้องใช้ หน้าสัมผัส (Contactless Smart Card) ซึ่งจะทำให้การคิดค่าโดยสารไม่เสียเวลาในการทำงาน ซึ่งมีความเป็นไปได้สูงสุด
2. การจัดทำฐานข้อมูลของการใช้บริการให้สมบูรณ์ ซึ่งจะทำให้การตรวจสอบ การทำงาน ตลอดจนปริมาณสภาพการโดยสารของผู้ใช้ได้ เพื่อจัดระบบการบริการให้มีประสิทธิภาพสูงสุด
3. เนื่องจากมีส่วนแสดงผลอยู่บนรถโดยสารแล้ว ถ้าสามารถเพิ่มระบบการรับข้อมูลแบบ RDS (Radio Data System) ก็จะสามารถที่จะกระจายข่าวสารให้กับรถประจำทางได้ โดยการนำข้อความต่าง ๆ มาแสดงผล

## บรรณานุกรม

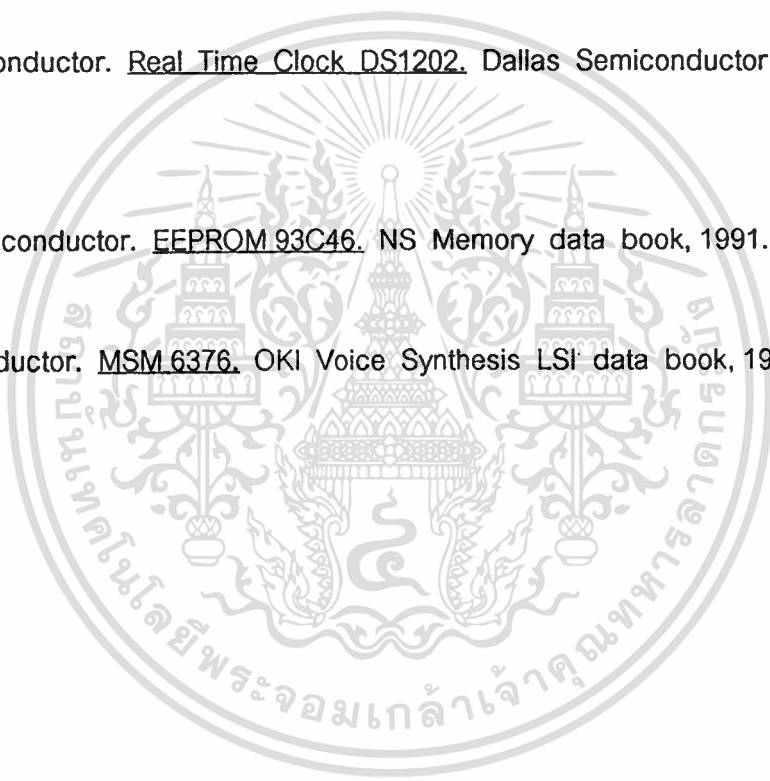
สุเจตน์ จันทพงษ์. ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051. ครั้งที่ 1 กรุงเทพฯ. วิทยาลัยมหานคร, 2535.

วรพล ลีลาเกียรติสกุล. "ระบบคิดค่าโดยสารรถประจำทาง" เอกสารการประชุมวิชาการทางไฟฟ้า. ครั้งที่ 18, 2538.

Dallas Semiconductor. Real Time Clock DS1202. Dallas Semiconductor data book, 1991.

National Semiconductor. EEPROM 93C46. NS Memory data book, 1991.

OKI Semiconductor. MSM 6376. OKI Voice Synthesis LSI data book, 1992/1993.





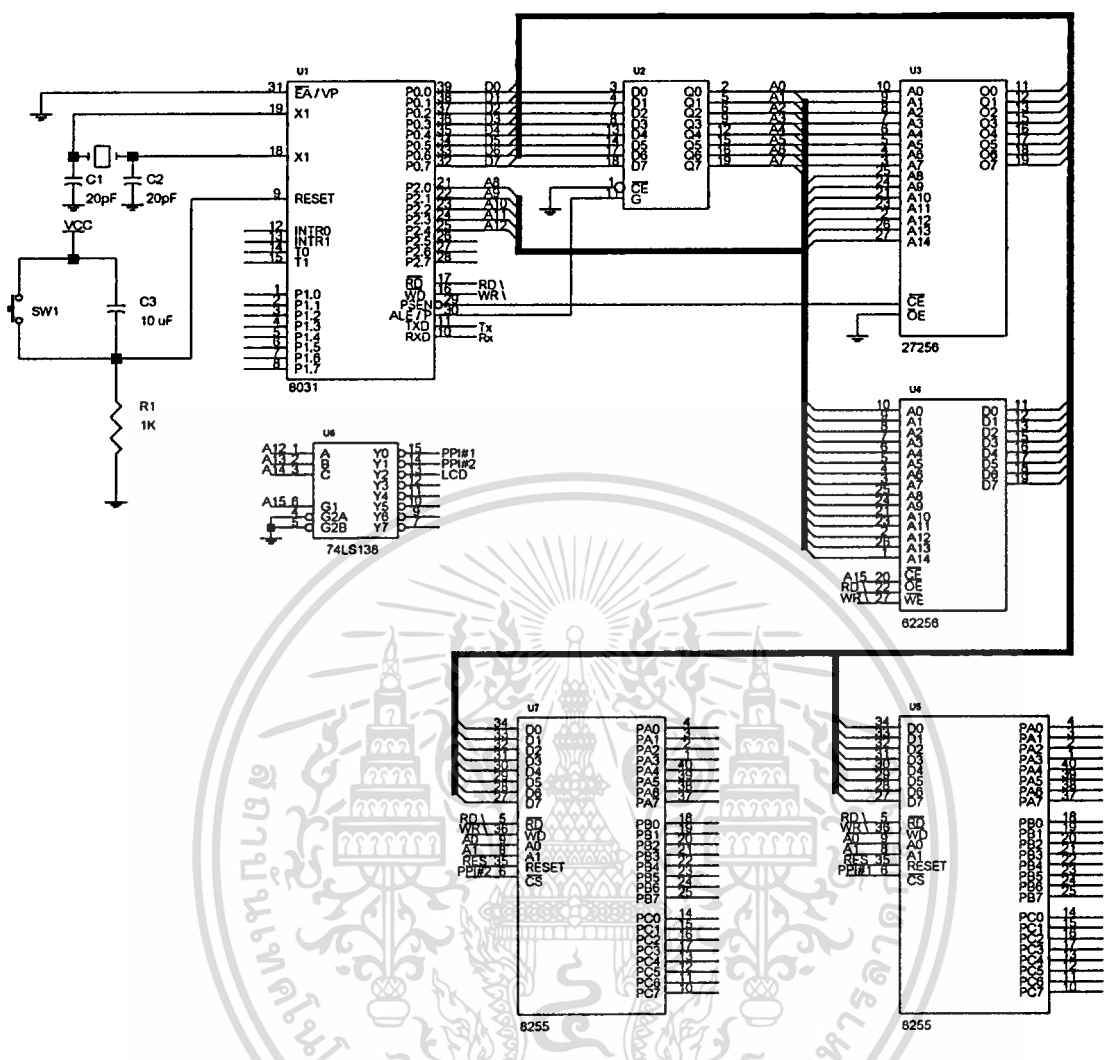
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

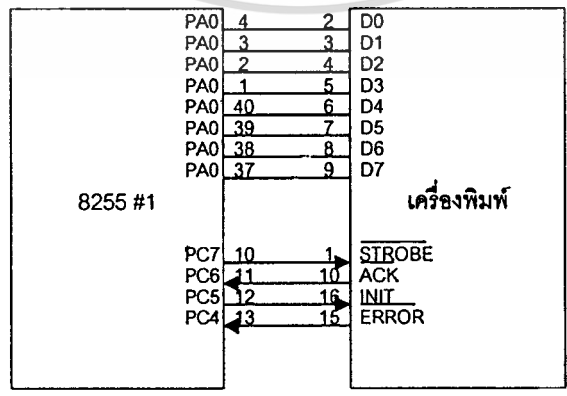
ภาพที่ 84

แสดงวงจรควบคุมส่วนควบคุมระบบหลัก



ภาพที่ 85

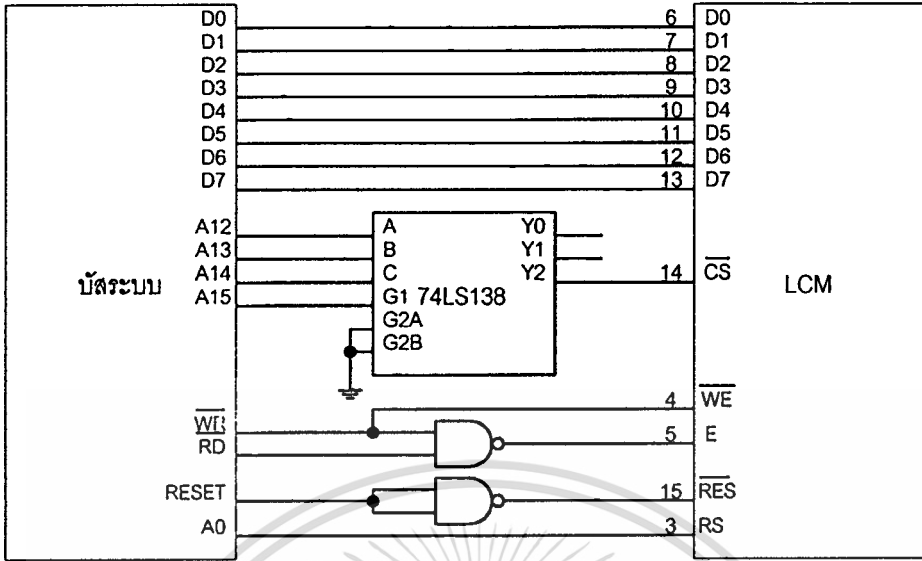
แสดงการเชื่อมต่อกับเครื่องพิมพ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

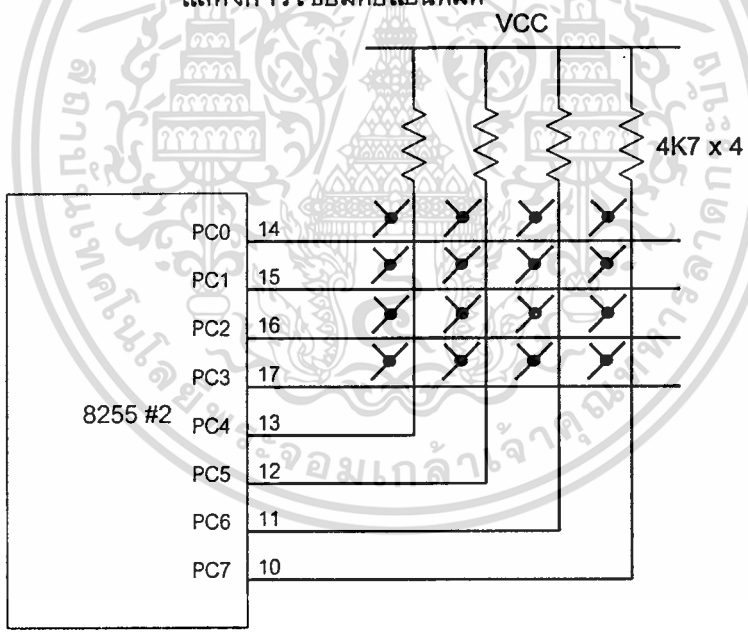
ภาพ 86

แสดงการเชื่อมต่อโมดูล LCD



ภาพ 87

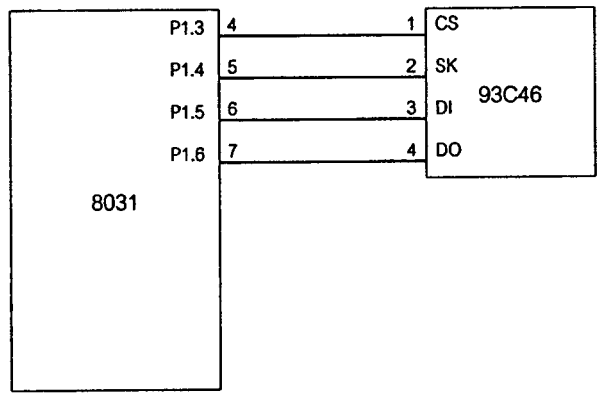
แสดงการเชื่อมต่อเป็นพินท์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

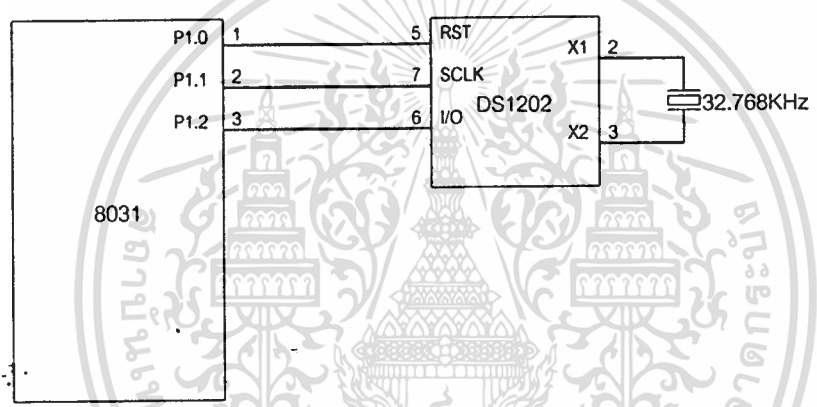
ภาพที่ 88

แสดงการเชื่อมต่อกับบัตรหน่วยความจำ



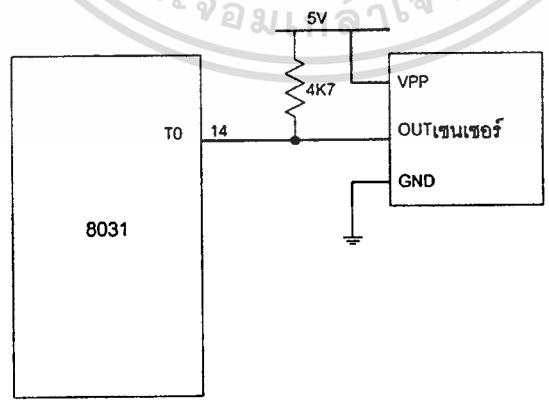
ภาพที่ 89

แสดงการเชื่อมต่อกับชิพฐานเวลาจริง



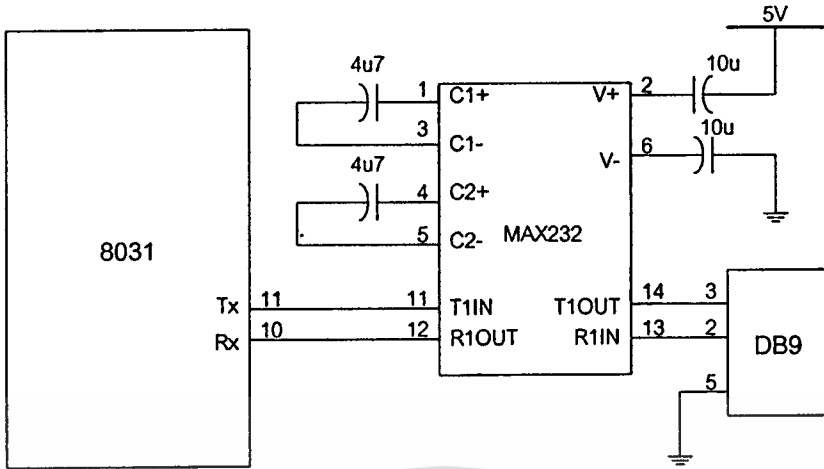
ภาพที่ 90

แสดงการเชื่อมต่อกับเซนเซอร์



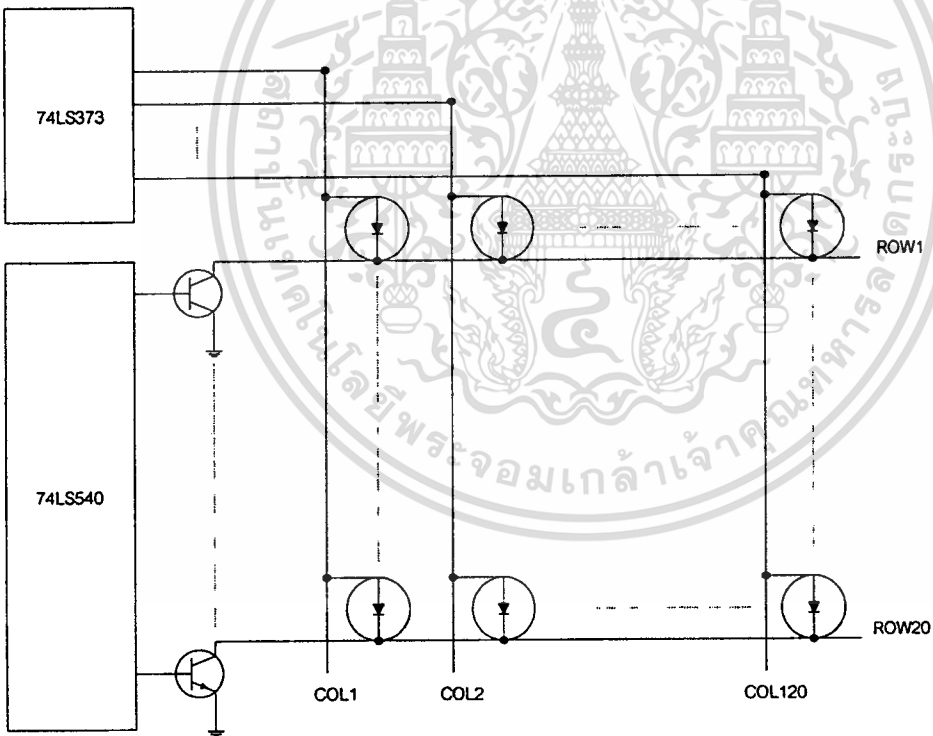
ภาพที่ 91

แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนควบคุมหลัก



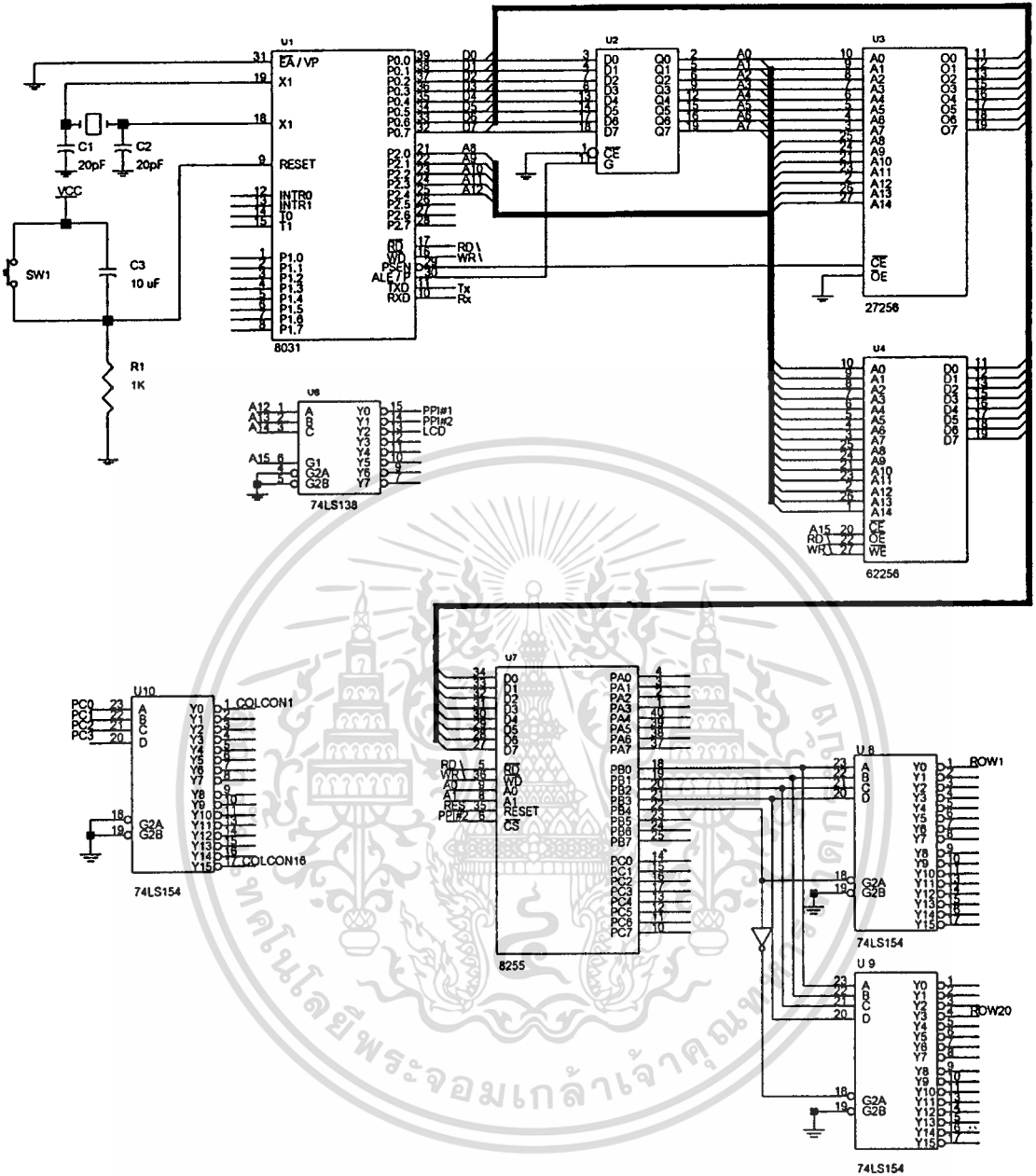
ภาพที่ 92

แสดงวงจรชุดขับกระแสของส่วนแสดงผลแบบจุดเมทริกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

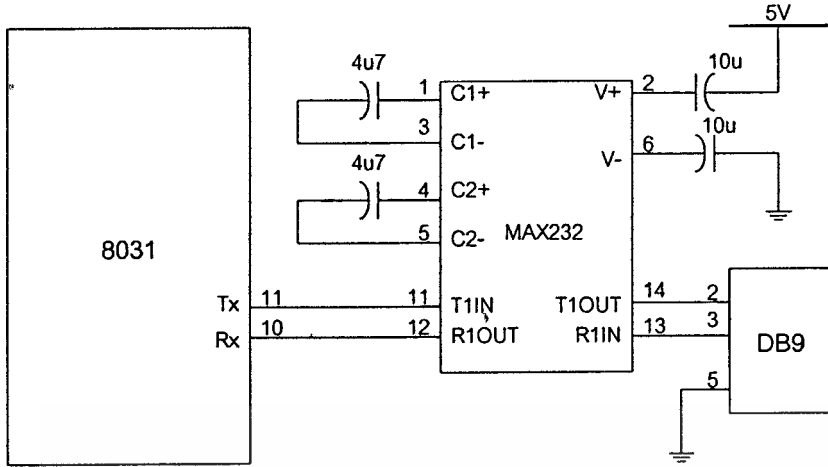
แสดงวงจรควบคุมของส่วนแสดงผลแบบจุดเมตริกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

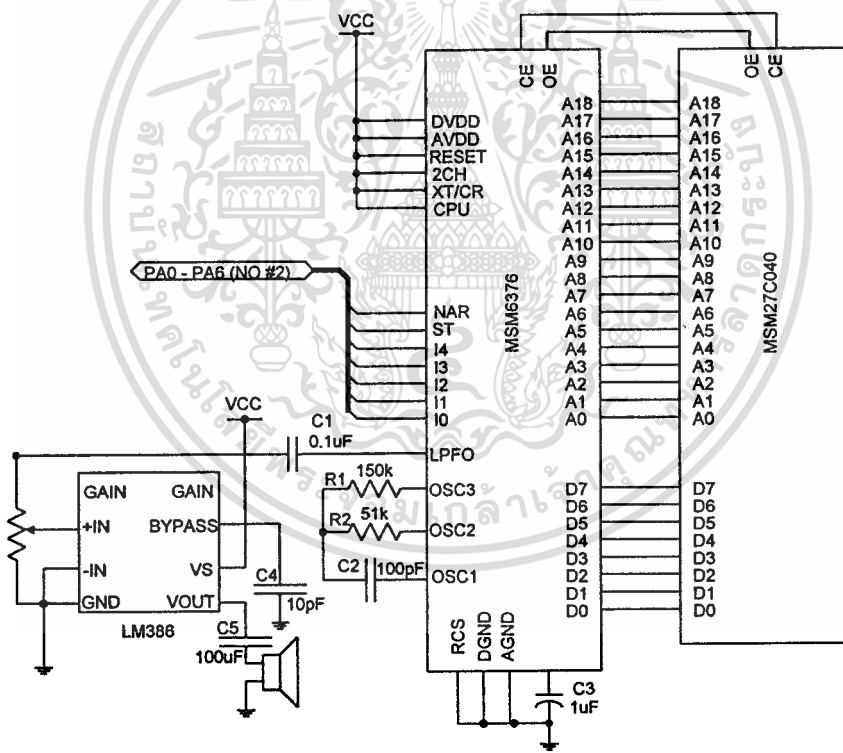
ภาพที่ 94

แสดงการเชื่อมต่อกับส่วนติดต่อข้อมูลอนุกรมบนส่วนการแสดงผล .



ภาพที่ 95

แสดงวงจรของส่วนแสดงผลเสียงพูด





ภาคผนวก ข.  
รายละเอียดโปรแกรมควบคุมระบบจัดการบรรณประจำทาง

```

; CPU 8031
; CLOCK 11.059 MHz
; ROM 32 KBytes
; RAM 32 KBytes
; PP1 ADDRESS 8000H - 8003H
; PP2 ADDRESS 9000H - 9003H
; LCM ADDRESS A000H - A001H
; KEYBOARD 16 KEYS
;
0000 CPU "8051.TBL"
0000 HOF "INT8"
;
;FIRST DEFINE REGISTER FOR COMPILER
;
0000 INCL "HEAD51.H"
;MCS-51 INTERNAL REGISTERS
00F0 = B: EQU 0F0H ;B REGISTER
00E0 = A: EQU 0E0H ;ACCUMULATOR
00D0 = PSW: EQU 0D0H ;PROGRAM STATUS WORD
00B8 = IPC: EQU 0B8H ;INTERRUPT PRIORITY
00B0 = P3: EQU 0B0H ;PORT 3
00B0 = P3.0: EQU 0B0H
00B1 = P3.1: EQU 0B1H
00B2 = P3.2: EQU 0B2H
00B3 = P3.3: EQU 0B3H
00B4 = P3.4: EQU 0B4H
00B5 = P3.5: EQU 0B5H
00B6 = P3.6: EQU 0B6H
00A8 = IEC: EQU 0A8H ;INTERRUPT ENABLE
00A0 = P2: EQU 0A0H ;PORT 2
0098 = SBUF: EQU 098H ;SEND BUFFER
0088 = SCON: EQU 088H ;SERIAL CONTROL
0090 = P1: EQU 090H
0090 = P1.0: EQU 090H ;PORT 1
0091 = P1.1: EQU 091H
0092 = P1.2: EQU 092H
0093 = P1.3: EQU 093H
0097 = P1.7: EQU 097H
008D = TH1: EQU 8DH ;TIMER 1 HIGH
008C = TH0: EQU 8CH ;TIMER 0 HIGH
008B = TL1: EQU 8BH ;TIMER 1 LOW
008A = TL0: EQU 8AH ;TIMER 0 LOW
0089 = TMOD: EQU 89H ;TIMER MODE
0088 = TCON: EQU 88H ;TIMER CONTROL
0087 = PCON: EQU 87H ;POWER CONTROL REGISTER
0083 = DPH: EQU 83H ;DATA POINTER HIGH
0082 = DPL: EQU 82H ;DATA POINTER LOW
0081 = SP: EQU 81H ;STACK POINTER
0080 = P0: EQU 80H ;PORT 0
;
;MCS-51 INTERNAL BIT ADDRESSES
00D7 = CY: EQU 0D7H ;CARRY FLAG
00D6 = AC: EQU 0D6H ;AUXILIARY-CARRY FLAG
00D5 = F0: EQU 0D5H ;USER FLAG 0
00D4 = RS1: EQU 0D4H ;REGISTER SELECT MSB
00D3 = RS0: EQU 0D3H ;REGISTER SELECT LSB
00D2 = OV: EQU 0D2H ;OVERFLOW FLAG
00D0 = P: EQU 0D0H ;PARITY FLAG
00BC = PS: EQU 0BCH ;PRIORITY SERIAL PORT
00BB = PT1: EQU 0BBH ;PRIORITY TIMER 1
00BA = PX1: EQU 0BAH ;PRIORITY EXTERNAL 1
00B9 = PT0: EQU 0B9H ;PRIORITY TIMER 0
00B8 = PX0: EQU 0B8H ;PRIORITY EXTERNAL 0
00AF = EA: EQU 0AFH ;ENABLE ALL INTERRUPT
00AC = ES: EQU 0ACH ;ENABLE SERIAL INTERRUPT
00AB = ET1: EQU 0ABH ;ENABLE TIMER 1 INTERRUPT
00AA = EX1: EQU 0AAH ;ENABLE EXTERNAL 1 INTERR
00A9 = ET0: EQU 0A9H ;ENABLE TIMER 0 INTERRUPT
00A8 = EX0: EQU 0A8H ;ENABLE EXTERNAL 0 INTERR
00A7 = SM0: EQU 0A7H ;SERIAL MODE 0
00A6 = SM1: EQU 0A6H ;SERIAL MODE 1
00A5 = SM2: EQU 0A5H ;SERIAL MODE 2
00A4 = REN: EQU 0A4H ;SERIAL RECEPTION ENABLE
00A3 = TB8: EQU 0A3H ;TRANSMIT BIT 8
009A = RB8: EQU 09AH ;RECEIVE BIT 8
0099 = TI: EQU 099H ;TRANSMIT INTERRUPT FLAG
0098 = RI: EQU 098H ;RECEIVE INTERRUPT FLAG
008F = TF1: EQU 08FH ;TIMER 1 OVERFLOW FLAG
008E = TR1: EQU 08EH ;TIMER 1 RUN CONTROL BIT
008D = TF0: EQU 08DH ;TIMER 0 OVERFLOW FLAG
008C = TR0: EQU 08CH ;TIMER 0 RUN CONTROL BIT
008B = IE1: EQU 08BH ;EXT INTERR. 1 EDGE FLAG
008A = IT1: EQU 08AH ;EXT INTERR. 1 TYPE FLAG
0089 = IE0: EQU 089H ;EXT INTERR. 0 EDGE FLAG
0088 = IT0: EQU 088H ;EXT INTERR. 0 TYPE FLAG
;
;INTERNAL RAM DEFINE
;
;REAL TIME DATA BUFFER
0010 = MINUTE: EQU 10H
0011 = HOUR: EQU 11H
0012 = DATE: EQU 12H
0013 = MONTH: EQU 13H
0014 = YEAR: EQU 14H
0015 = second: equ 15h
0016 = TMP_BUF0: EQU 16H
0017 = TMP_BUF1: EQU 17H
0018 = BUF_VAL1: EQU 18H
0019 = BUF_VAL2: EQU 19H
001A = BUF_VAL3: EQU 1AH
001B = INDEXPOS: EQU 1BH
001C = BUF_DIST: EQU 1CH
001D = BUF_CURP: EQU 1DH
0020 = CURROW: EQU 20H ;CURRENT CURSOR POSITION IN ROW
0 - 2
0021 = CURCOL: EQU 21H ;CURRENT CURSOR POSITION IN COL 0
- 29
0022 = POSHI: EQU 22H ;SAVE HEX POSITION HI BYTE
0023 = POSLO: EQU 23H ;SAVE HEX POSITION LO BYTE
0024 = POS_L: EQU 24H ;BUFFER TO STORE LOW BYTE OF
POSITION DISPLAY RAM
0025 = POS_H: EQU 25H ;BUFFER TO STORE HIGH BYTE OF
POSITION DISPLAY RAM
0026 = BUF_ROUTEL: EQU 26H
0027 = BUF_ROUTEH: EQU 27H
0028 = BUF_INDEXL: EQU 28H
0029 = BUF_INDEXH: EQU 29H
0070 = FGPAGE0: EQU 70H ;STATUS FOR FIRST PAGE OF MENU
0071 = FGCAR: EQU 71H ;STORE CARRY FLAG
0072 = FGFOR: EQU 72H ;STATUS FORWARD ROUTE
;
; LCD MODULE INTERFACE
A001 = CNTLCM: EQU 0A001H ;CONTROL REGISTER
A000 = DATLCM: EQU 0A000H ;DATA REGISTER
;
; PRINTER INTERFACE & OUTPUT PORT
8000 = PP1A: EQU 8000H
8001 = PP1B: EQU PP1A+1
8002 = PP1C: EQU PP1A+2
8003 = PP1CO: EQU PP1A+3
;
; KEYBOARD & SPEECH SYNTHESIS INTERFACE
9000 = PPI2A: EQU 9000H
9001 = PPI2B: EQU PPI2A+1
9002 = PPI2C: EQU PPI2A+2
9003 = PPI2CO: EQU PPI2A+3
;
;DATA DEFINE
00A2 = CWORD1: EQU 0A2H ;MODE 1 PORT A OUTPUT
0088 = CWORD2: EQU 88H ;MODE 0 PORT A & B & C0-C3 OUTPUT
;PORT C4-C7 INPUT
FE88 = KILOP: EQU -(330) ;PULSE/KILOMETER
0000 = CRDVALID: EQU 00H
0001 = UPCODE: EQU 01H
0002 = UPPOS: EQU 02H
0003 = UNITVAL: EQU 03H
0004 = TENVAL: EQU 04H
0005 = HUNDVAL: EQU 05H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0006 = UPMIN: EQU 06H
0007 = UPHOUR: EQU 07H

;EXTERNAL DEFINE
1000 = BACKUP: EQU 1000H ;CHECK OLD DATA
1001 = DISTANCE: EQU 1001H ;ALL DISTANCE
1002 = CURRENTP: EQU 1002H ;CURRENT POSITION

;*****
0000 ORG 0000H
0000 020040 START: LJMP START1 ;JUMP PASS INTERRUPT SERVICE
ROUTINE

000B ORG 000BH
000B 02080C LJMP CHKDIST ;CHECK DISTANCE

0040 ORG 40H
;
;DELAY FOR START UP
0040 START1:
0040 D8FE DJNZ R0,$
0042 D9FC DJNZ R1,START1 ;DELAY FOR START SYSTEM
;
;INITIAL STACK
0044 758140 MOV SP,#40H ;INITIAL STACK
;
0047 12059A LCALL RAMCLR ;CLEAR BUFFER RAM

;*****
;INITIAL INTERNAL COUNTER INTERRUPT
004A INITCNT:
004A 75A882 MOV IEC,#82H ;ENABLE INTERRUPT COUNTER
004D 758825 MOV TMOD,#25H ;COUNTER 0 MODE 1
0050 758CFE MOV TH0,#HIGH{KILOP}
0053 758AB6 MOV TL0,#LOW{KILOP} ;SET INITIAL CONDITION
0056 751C00 MOV BUF_DIST,#00 ;INITIAL CURSOR FIST TIME BY DEFINE ROW NUMBER IN INTERNAL
0059 751D00 MOV BUF_CURP,#00 RAM FOLLOW,

;*****
;INITIAL SERIAL PORT
005C INITSER:
005C 759852 MOV SCON,#52H ;ENABLE RECIEVE SERIAL PORT
MODE 1
005F 758DFD MOV TH1,#0FDH ;BUAD RATE 9600
0062 D28E SETB TR1

;*****
;INITIAL PARALLEL PORT
0064 INITPAR:
0064 909003 MOV DPTR,#PPI2CO
0067 7488 MOV A,#CWORD2
0069 F0 MOVX @DPTR,A

;*****
;INITIAL PRINTER AND CHARACTERISTIC
;INITPRN:
; MOV DPTR,#PPI1CO
; MOV A,#0A2H ;MODE 1 GROUP A IS OUTPUT
; ;PORT C IS INPUT
;
; MOVX @DPTR,A
; MOV A,#02H ;ACTIVE INIT PIN
; MOVX @DPTR,A
; LCALL DELAYM ;DELAY FOR A WHILE
; MOV A,#03H
; MOVX @DPTR,A
; LCALL DELAYM
; MOV A,#0DH ;DE-ACTIVE INIT PIN
; MOVX @DPTR,A
; LCALL DELAYM
; LCALL CHKBUSY
; MOV R0,#10H

;*****
;INIT LCM TO GRAPHIC MODE
006A 120FB1 LCALL INITLCD ;INIT IN GRAPHIC MODE
;
;CLEAR SCREEN , ALL DOT OFF
006D 12100A LCALL CLRSCRN ;BLANK ALL SCREEN
;
;CHECK DATA EVER BE BACK UP IN RAM
;
; MOV DPTR,#BACKUP
; MOVX A,@DPTR
; CJNE A,#55H,NXTBACK1
; INC DPTR
; MOVX A,@DPTR
; MOV BUF_DIST,A ;LOAD OLD DISTANCE
; INC DPTR
; MOVX A,@DPTR
; MOV BUF_CURP,A ;LOAD CURRENT POSITION
; SJMP NXTBACK2
;
;NXTBACK1:
; INC DPTR
; MOV A,#00
; MOV BUF_DIST,A
; MOVX @DPTR,A ;CLEAR DISTANCE
; INC DPTR
; MOV BUF_CURP,A
; MOVX @DPTR,A ;CLEAR CURRENT POSITION
;
;NXTBACK2:
; ADDRESS 50H = ROW NUMBER 0,1,2
; ADDRESS 51H = COL NUMBER 0 - 29
; FIRST TIME, SET CURSOR AT ROW0,COLO
0070 7820 MOV R0,#CURROW ;INIT CURSOR POSITION
0072 7600 MOV @R0,#00H ;ROW 0 , COL 0
0074 08 INC R0
0075 7600 MOV @R0,#00
0077 1210C0 LCALL CURCONV ;CONVERT TO HEX POSITION
007A MAIN:
007A 120546 LCALL SHW_LOGO ;SHOW MEEASURE
007D 1200A1 LCALL WAITKEY ;WAIT KEYPRESS
0080 LPMAIN:
0080 1200AA LCALL SHW_MMENU
0083 1200A1 LCALL WAITKEY
0086 MAIN0:
0086 B40106 CJNE A,#1,MAIN1
0089 1200F3 LCALL WORKNORM ;GO OPERATION ON
008C 020080 LJMP LPMAIN
008F MAIN1:
008F B40206 CJNE A,#2,MAIN2
0092 120480 LCALL PROGCARD ;PROGRAM CARD VALUE
0095 020080 LJMP LPMAIN
0098 MAIN2:
0098 B403E5 CJNE A,#3,LPMAIN
009B 120542 LCALL CHECKDATA ;CHECK NUMBER OF USAGE
009E 020080 LJMP LPMAIN

;*****
;ROUTINE WAIT FOR ANY KEY PRESSED
00A1 WAITKEY:
00A1 120609 LCALL SCANKEY
00A4 50FB JNC WAITKEY
00A6 120591 LCALL BEEP
00A9 22 RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;SUBROUTINE FOR PROCESS WITH REAL TIME CLOCK CHIP
;DS1202
;PORT PIN DEFINE
SCLKRTC: EQU P1.0 ;SERIAL CLOCK DS1202 RTC
IORTC: EQU P1.1 ;I/O DS1202 RTC
RSTRTC: EQU P1.2 ;RST DS1202 RTC

```

RET

```

;----- SUB -----
; WRITE SINGLE BYTE TO DS1202
; IN = R3 COMMAND
; R4 DATA
; REG = A,R2,R3,R4,R5

```

;----- SETTIME SUB -----

```

; SETTIME SUBROUTINE
; INPUT = MINUTE, HOUR
; REG = R2,R3

```

RTCWR:

```

CLR SCLKRTC ;CLK=0
CLR IORTC
LCALL RTCDL
SETB RSTRTC ;RST=1
LCALL RTCDL
MOV A,R3 ;COMMAND
LCALL RTCWRC ;WRITE COMMAND/DATA
MOV A,R4 ;DATA
LCALL RTCWRC ;WRITE COMMAND/DATA
CLR RSTRTC ;RST=0
LCALL RTCDL
RET

```

SETTIME:

```

CLR RSTRTC
CLR IORTC
CLR SCLKRTC
MOV R3,#8EH ;WRITE PROTECT = 0
MOV R4,#00H
LCALL RTCWR
MOV R3,#80H ;SECOND
MOV R4,#00
LCALL RTCWR

MOV R3,#82H ;MINUTE = 0
MOV R4,MINUTE
LCALL RTCWR

MOV R3,#84H ;HOUR = 0
MOV R4,HOUR
LCALL RTCWR

MOV R3,#8EH ;WRITE PROTECT = 1
MOV R4,#80H
LCALL RTCWR
RET

```

RTCWRC:

```

MOV R1,#8 ;WRITE COMMAND/DATA 8 BIT
RTCWRC1: RRC A
MOV IORTC,C ;PUT DATA
SETB SCLKRTC ;RISING EDGE CLOCK
LCALL RTCDL
CLR SCLKRTC ;FALLING EDGE CLOCK
LCALL RTCDL
DJNZ R1,RTCWRC1
RET

```

;----- SETDATE SUB -----

```

; SETTIME SUBROUTINE
; INPUT = DATE, YEAR, MONTH
; REG = R2,R3

```

;----- RDTIME SUB -----

```

; READ TIME SUBROUTINE
; OUTPUT = SECOND, MINUTE, HOUR, DAY, DATE, YEAR, MONTH
; REG = R2,R3,A,DPTR

```

SETDATE:

```

CLR RSTRTC
CLR IORTC
CLR SCLKRTC
MOV R3,#8EH ;WRITE PROTECT = 0
MOV R4,#00H
LCALL RTCWR

MOV R3,#86H ;DATE
MOV R4,DATE
LCALL RTCWR

MOV R3,#88H ;MONTH
MOV R4,MONTH
LCALL RTCWR

MOV R3,#8CH ;YEAR
MOV R4,YEAR
LCALL RTCWR

MOV R3,#8EH ;WRITE PROTECT = 1
MOV R4,#80H
LCALL RTCWR

```

RDTIME:

```

MOV R3,#81H
LCALL RTCRD
MOV SECOND,R4

MOV R3,#83H ;READ MINUTE TIME
LCALL RTCRD
MOV MINUTE,R4

MOV R3,#85H ;READ HOUR TIME
LCALL RTCRD
MOV HOUR,R4

MOV R3,#8DH ;READ YEAR TIME
LCALL RTCRD
MOV YEAR,R4

MOV R3,#89H ;READ MONTH TIME
LCALL RTCRD
MOV MONTH,R4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV DATE,R4
```

```
; MOV R3,#8BH ;READ DAY TIME
; LCALL RTCRD
; MOV DAY,R4
RET
```

```
;----- RTCRD SUB -----
; READ SINGLE BYTE FROM DS1202
; IN = R3 COMMAND
; OUT = R4 DATA
; REG = A,R2,R3,R4
```

```
RTCRD:
CLR SCLKRTC ;CLK=0
CLR IORTC
LCALL RTCDL
SETB RSTRTC ;RST=1
LCALL RTCDL
MOV A,R3 ;COMMAND
LCALL RTCWRC ;WRITE COMMAND/DATA

SETB IORTC
MOV R1,#8 ;READ DATA
CLR A
```

```
RTCRD1:
MOV C,IORTC
RRC A
SETB SCLKRTC
LCALL RTCDL
CLR SCLKRTC
LCALL RTCDL
DJNZ R1,RTCRD1
MOV R4,A
CLR RSTRTC ;RST = 0
LCALL RTCDL
RET
```

```
;-----
;ROUTINE FOR DELAY TIME OF REAL TIME CLOCK
;REG = R5
```

```
RTCDL: MOV R5,#5H
DJNZ R5,$
;RET
```

```
00CE 1210C0 LCALL CURCONV
00D1 9008B7 MOV DPTR,#MSG_MAIN2
00D4 752500 MOV POS_H,#00H
00D7 752450 MOV POS_L,#50H ;SHOW AT LINE 1
00DA 1207E5 LCALL DSP_MSG
```

```
00DD 752002 MOV CURROW,#02
00E0 752100 MOV CURCOL,#00
00E3 1210C0 LCALL CURCONV
00E6 9008D5 MOV DPTR,#MSG_MAIN3
00E9 752500 MOV POS_H,#00H
00EC 7524A0 MOV POS_L,#0A0H ;SHOW AT LINE 2
00EF 1207E5 LCALL DSP_MSG
00F2 22 RET
```

```
-----
;ROUTINE FOR BUS MANAGEMENT SYSTEM WORK
-----
```

```
00F3 WORKNORM:
00F3 1203A4 LCALL SHW_ROUTE ;SHOW ROUTE MESSAGE
00F6 1200A1 LCALL WAITKEY
00F9 LPWORKNORM:
00F9 1203ED LCALL SHW_RMENU ;SHOW ROUTE MENU
00FC 1200A1 LCALL WAITKEY
00FF WORKNORM0:
00FF B40117 CJNE A,#1,WORKNORM1
0102 C272 CLR FGFOR ;SELECT REVERSE TABLE
0104 900C88 MOV DPTR,#ROUTETABR
0107 858327 MOV BUF_ROUTEH,DPH
010A 858226 MOV BUF_ROUTEH,DPL
010D 900C73 MOV DPTR,#INDEXR
0110 858329 MOV BUF_INDEXH,DPH
0113 858228 MOV BUF_INDEXL,DPL
0116 020138 LJMP WORKNORM3
0119 WORKNORM1:
0119 B40217 CJNE A,#2,WORKNORM2
011C D272 SETB FGFOR ;SELECT FORWARD TABLE
011E 900C49 MOV DPTR,#ROUTETABF
0121 858327 MOV BUF_ROUTEH,DPH
0124 858228 MOV BUF_ROUTEH,DPL
0127 900C34 MOV DPTR,#INDEXF
012A 858329 MOV BUF_INDEXH,DPH
012D 858228 MOV BUF_INDEXL,DPL
0130 020138 LJMP WORKNORM3
0133 WORKNORM2:
0133 0200F9 LJMP LPWORKNORM
0136 WORKNORM3:
0136 1201E5 LCALL TIMESET ;SET REAL TIME CLOCK
0139 120437 LCALL SHW_CONFIRM ;SHOW CONFIRM MESSAGE
013C 1200A1 LCALL WAITKEY
013F B40A08 CJNE A,#10,WORKNORM31
0142 D28C SETB TR0 ;START COUNTER
0144 751C00 MOV BUF_DIST,#00
0147 020150 LJMP WORKNORM32 ;START OPERATION
014A WORKNORM31:
014A B40BE9 CJNE A,#11,WORKNORM3
014D 0200F9 LJMP LPWORKNORM ;RETURN BACK TO SELECT NEW
OPTION
```

```
0150 WORKNORM32:
0150 12100A LCALL CLRSCRN
0153 WORKNORM4:
0153 120188 LCALL SHW_PROCESS ;SHOW LIST OF DATA
0156 120609 LCALL SCANKEY
0159 4003 JC NXTWORKNORM4
; LCALL CHKCRD
; JNC WORKNORM4
; LCALL CARDPRO ;PROCESS WITH SMART CARD
015B 020153 LJMP WORKNORM4
015E NXTWORKNORM4:
015E 120591 LCALL BEEP
0161 B40C06 CJNE A,#12,WORKNORM41
0164 758140 MOV SP,#40H ;END PROCESS AND RESTART
PROGRAM
0167 02007A LJMP MAIN ;WARM BOOT
016A WORKNORM41:
016A B40D06 CJNE A,#13,WORKNORM42
016D 120436 LCALL DISPRATE ;DISPLAY RATE OF CHARGE
```

```
-----
;ROUTINE FOR MAIN MENU
-----
```

```
00AA SHW_MMENU:
00AA 12059A LCALL RAMCLR
00AD 12100A LCALL CLRSCRN
00B0 1210AC LCALL CURHOME

00B3 752000 MOV CURROW,#00 ;SHOW AT LINE 0
00B6 752100 MOV CURCOL,#00
00B9 1210C0 LCALL CURCONV
00BC 90089D MOV DPTR,#MSG_MAIN1
00BF 752500 MOV POS_H,#00H
00C2 752400 MOV POS_L,#00H ;SHOW AT LINE 0 PAGE 0
00C5 1207E5 LCALL DSP_MSG

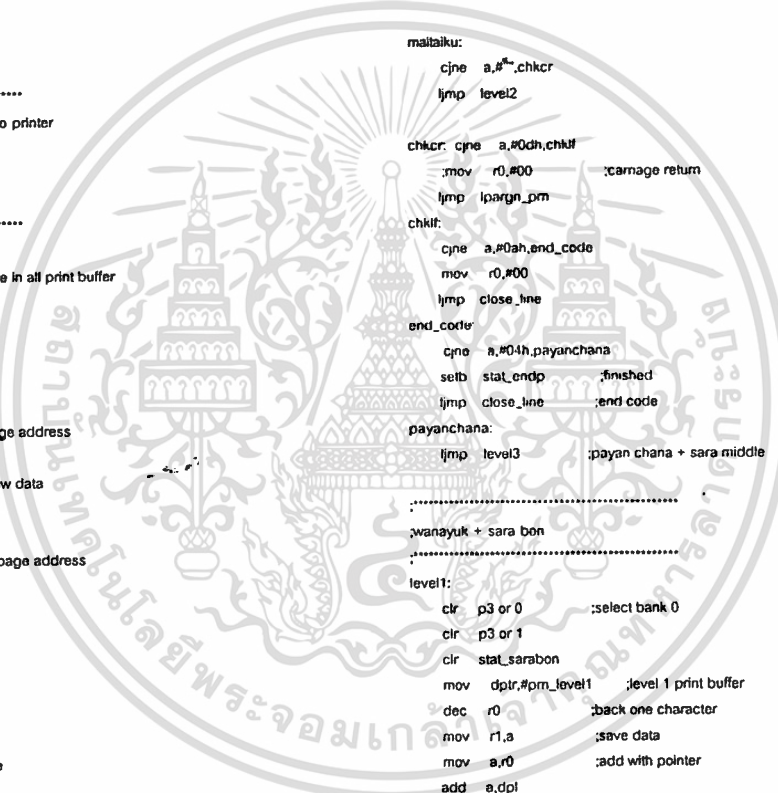
00C8 752001 MOV CURROW,#01
00CB 752100 MOV CURCOL,#00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;register use : r6,r7
;.....
delays:
    mov r6,#00 ;initial data
    mov r7,#00
delaypb:
    djnz r6,$ ;
    djnz r7,delaypbs
    ret
;.....
;routine delay millisecond
;input - acc (number of millisecond)
;output - none
;register use : r6,r7
;.....
delaypm:
    mov r6,#100 ;initial data
    mov r7,#5
delayppm:
    djnz r6,$ ;
    djnz r7,delayppm
    ret
;.....
;routine arrange and print information to printer
;input -
;output -
;register use :r0,r1,dptr,acc
;.....
argn_pm:
    lcall putspace ;put space in all print buffer
    mov r0,#00 ;pointer
    mov dptr,#page_buf
    mov page1,dpl
    mov pageh,dph
lpargn_pm:
    mov dpl,page1 ;load page address
    mov dph,pageh
    movx a,@dptr ;read raw data
    inc dptr
    mov page1,dpl
    mov pageh,dph ;save page address
    cjne a,#" ",sara_uu
    jmp level4 ;sara u
sara_uu:
    cjne a,#" ",sara_lee
    jmp level4 ;sara UU
sara_lee:
    cjne a,#" ",sara_lee
    setb stat_sarabon ;sara lee
    jmp level2
sara_lee:
    cjne a,#" ",sara_uh
    setb stat_sarabon ;sara uuh
    jmp level2
sara_uuh:
    cjne a,#" ",maihaan
    setb stat_sarabon ;sara uuh
    jmp level2
maihaan:
    cjne a,#" ",karan
    setb stat_sarabon ;maihaan agart
    jmp level2
karan:
    cjne a,#" ",malek
    jmp level2 ;karan
malek:
    cjne a,#" ",maltee
;.....
jnb stat_sarabon,malek1 ;malek
ljmp level1 ;follow by sara upper
malek1:
    jmp level2 ;not follow
maltee:
    cjne a,#" ",maitee
    jnb stat_sarabon,maltee1 ;mai tee
    jmp level1 ;follow by sara upper
maltee1:
    jmp level2 ;not follow
maitee:
    cjne a,#" ",maijatava
    jnb stat_sarabon,maitee1 ;mai tee
    jmp level1 ;follow by sara upper
maitee1:
    jmp level2 ;not follow
maijatava:
    cjne a,#" ",maitaiku
    jnb stat_sarabon,maijatava1 ;mai jatawa
    jmp level1 ;follow by sara upper
maijatava1:
    jmp level2 ;not follow
maitaiku:
    cjne a,#" ",chkcr
    jmp level2
chkcr:
    cjne a,#"0dh",chkif
    mov r0,#00 ;carnage return
    jmp lpargn_pm
chkif:
    cjne a,#"0ah",end_code
    mov r0,#00
    jmp close_line
end_code:
    cjne a,#"04h",payanchana
    setb stat_endp ;finished
    jmp close_line ;end code
payanchana:
    jmp level3 ;payan chana + sara middle
;.....
;wanayuk + sara bon
;.....
level1:
    clr p3 or 0 ;select bank 0
    clr p3 or 1
    clr stat_sarabon
    mov dptr,#pm_level1 ;level 1 print buffer
    dec r0 ;back one character
    mov r1,a ;save data
    mov a,r0 ;add with pointer
    add a,dpl
    mov dpl,a
    mov a,#00
    addc a,dph
    mov dph,a ;position of print character
    mov a,r1 ;restore data
    movx @dptr,a
    inc r0 ;restore old
    cjne r0,#page_width-1,nxlevel1
    mov r0,#00
    jmp close_line
nxlevel1:
    jmp lpargn_pm
;.....
;sara bon - sara lee,sara lee and etc.
;.....
level2:
    clr p3 or 0 ;select bank 0
    clr p3 or 1
    mov dptr,#pm_level2 ;level 2 print buffer
    dec r0 ;back one character

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov r1,a ;save data
mov a,r0 ;add with pointer
add a,dpl
mov dpl,a
mov a,#00
addc a,dph
mov dph,a ;position of print character
mov a,r1 ;restore data
movx @dptr,a
inc r0 ;restore old
cjne r0,#page_width-1,nxtlevel2
mov r0,#00
ljmp close_line
nxtlevel2:
ljmp lpargn_pm

```

.....  
:payahchana & sara klang & mai muan mai malal mai yamok  
.....

```

level3:
clr p3 or 0 ;select bank 0
clr p3 or 1
clr sta_sarabon
mov dptr,#pm_level3 ;level 3 print buffer
mov r1,a ;save data
mov a,r0 ;add with pointer
add a,dpl
mov dpl,a
mov a,#00
addc a,dph
mov dph,a ;position of print character
mov a,r1 ;restore data
movx @dptr,a
inc r0 ;point to next position
cjne r0,#page_width-1,nxtlevel3
mov r0,#00
ljmp close_line
nxtlevel3:
ljmp lpargn_pm

```

.....  
:sara lang - sara u,sara uu  
.....

```

level4:
clr p3 or 0 ;select bank 0
clr p3 or 1
clr sta_sarabon
mov dptr,#pm_level4 ;level 4 print buffer
dec r0 ;back one character
mov r1,a ;save data
mov a,r0 ;add with pointer
add a,dpl
mov dpl,a
mov a,#00
addc a,dph
mov dph,a ;position of print character
mov a,r1 ;restore data
movx @dptr,a
inc r0 ;restore old
cjne r0,#page_width-1,nxtlevel4
mov r0,#00
ljmp close_line
nxtlevel4:
ljmp lpargn_pm

```

close\_line:

```

mov a,#04h
mov dptr,#pm_level1+page_width
movx @dptr,a ;write end code in level 1
mov dptr,#pm_level2+page_width
movx @dptr,a ;write end code in level 2
mov dptr,#pm_level3+page_width
movx @dptr,a ;write end code in level 3
mov dptr,#pm_level4+page_width

```

```

movx @dptr,a ;write end code in level 4
transfer:
mov dptr,#leftmar
lcall setcode
mov dptr,#pm_level1 ;print level1
lcall print
mov dptr,#v1fd ;feed between level
lcall setcode
mov dptr,#leftmar
lcall setcode
mov dptr,#pm_level2 ;print level2
lcall print
mov dptr,#v2fd ;feed between level
lcall setcode
mov dptr,#leftmar
lcall setcode
mov dptr,#pm_level3 ;print level3
lcall print
mov dptr,#v3fd ;feed between level
lcall setcode
mov dptr,#leftmar
lcall setcode
mov dptr,#pm_level4 ;print level4
lcall print
mov dptr,#v4fd ;feed between level
lcall setcode
jb sta_endp,qargn_pm
lcall putspace
ljmp lpargn_pm ;loop back until end code
qargn_pm:
clr sta_endp
ret

```

.....  
:routine put space into printer buffer  
:register user0,r1,acc  
.....

```

putspace:
clr p3 or 0 ;select bank 0
clr p3 or 1
mov r1,#02h
mov r0,#00h ;:512 bytes
mov dptr,#print_buf
mov a,#20h ;:space character
lpputspace:
movx @dptr,a
inc dptr
djnz r0,lpputspace ;:0000 - 0001fh (512 byte)
djnz r1,lpputspace
ret

```

.....  
:routine set control code  
:input -dptr  
.....

```

setcode:
mov r0,dpl
mov r1,dph ;save code address
mov r2,#00
lpsetcode:
mov a,r2
mov dpl,r0 ;load code address
mov dph,r1
movc a,@a+dptr ;read code
cjne a,#04h,nxtsetcode
sjmp qsetcode
nxtsetcode:
inc r2
clr p3 or 0 ;select bank 2
setb p3 or 1
mov dptr,#pdata
movx @dptr,a
lcall chkbusy

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ljmp lpselcode
qselcode:
ret
;.....
;routine print all page to printer
;register use: r2,r3
;.....
print:
mov r2,dpl ;save print address
mov r3,dph
lpprintb:
mov dpl,r2 ;load print address
mov dph,r3
clr p3 or 0 ;select bank 0
clr p3 or 1
movx a,@dptr ;read code from buffer
inc dptr
mov r2,dpl ;save print address
mov r3,dph
cjne a,#04h,nxtprintb
sjmp qprintb
nxtprintb:
clr p3 or 0 ;select bank2
setb p3 or 1
mov dptr,#pmdata
movx @dptr,a ;send data to printer
lcall chkbusy
ljmp lpprintb
qprintb:
ret

```

```

;.....
;routine check busy
;input - none
;output - none
;.....
chkbusy:
clr p3 or 0
setb p3 or 1 ;bank 2
push dph
push dpl
push a
lpchk:
mov dptr,#pmcrlr
movx a,@dptr
jnb a or 3,lpchk
pop a
pop dpl
pop dph
ret

```

```

;.....
;routine adjust BCD to ASCII
;input - acc
;output - r3 (high digit)
; - r2 (low digit)
;.....
bcdasc:
mov r3,a
ani a,#0fh ;adjust low digit
ori a,#30h
mov r2,a ;save low digit
mov a,r3
swap a ;adjust high digit
ani a,#0fh
ori a,#30h
mov r3,a ;save high digit
ret

```

```

;.....
;routine print one character
;input : a
;.....
printch:
lcall chkbusy
mov dptr,#pmdata
movx @dptr,a
ret

```

```

;.....
;printer define control data
;.....
smopat: dfb 1bh,28h,74h,03h,00h,30h,14h,00h,04h
M1fd: dfb 1bh,33h,0ah,0ah,04h
M2fd: dfb 1bh,33h,17h,0ah,04h
M3fd: dfb 1bh,33h,16h,0ah,04h
M4fd: dfb 1bh,33h,18h,0ah,04h
crlf: dfb 0ah,0dh,04h
leftmar: dfb 1bh,6ch,02h,04h
num_line: dfb 1bh,43h,1eh,04h ;30 line/page
char_size: dfb 1bh,50h,04h ;10 cpi
cpi10: dfb 1bh,50h,04h
cpi12: dfb 1bh,4dh,04h
emp_char: dfb 1bh,45h,04h ;select emphasized
nemp_char: dfb 1bh,46h,04h ;cancel emphasized
dbstik_char: dfb 1bh,47h,04h ;select double strike
ndbstik_char: dfb 1bh,48h,04h ;cancel double strike
ital_char: dfb 1bh,34h,04h ;select italic
nital_char: dfb 1bh,35h,04h ;cancel italic
under_char: dfb 1bh,2ch,01h,04h ;turn on underline character
nunder_char: dfb 1bh,2ch,00h,04h ;turn off underline character
dbwid_char: dfb 1bh,0eh,04h ;select double width character
ndbwid_char: dfb 1bh,14h,04h ;cancel double width character
onwid_char: dfb 1bh,57h,01h,04h ;turn on double width caharacter
offwid_char: dfb 1bh,57h,00h,04h ;turn off double width character
cond_char: dfb 1bh,0fh,04h ;select condense mode
ncond_char: dfb 1bh,12h,04h ;cancel condense mode

```

```

form0: dfb " BUS MANAGEMENT SYSTEM",0AH,0DH
dfb " BY",0AH,0DH
dfb " Mr. WORAPHON LILAKIATSAKUN 35620065",0AH,0DH
dfb "KING MONGKUT INSTITUTE OF TECHNOLOGY CHAOKUNTHAHARN
LADKRABANG",0AH,0DH
dfb " / Functional test with Printer interface ",0Ah,0Dh,04H
form1: dfb "0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ",0AH,0DH,04H
form2: dfb " ระบบจัดการรถโดยสารประจำทาง",0ah,0dh
dfb " โดย",0ah,0dh
dfb " นาย วรพล สีลาเกียรติสกุล รหัส 35620065",0ah,0dh
dfb " สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง",0ah,0dh
dfb " นี้คือการทดสอบการทำงานเชื่อมต่อกับเครื่องพิมพ์ * ",0ah,0dh ,04h

```

```

form3: dfb "๐๑๒๓๔๕๖๗๘๙ กขคดงจชฌฎฏฐฑฒณดตถทธนบปผฝพฟภท
มยรฤศษสฬฮอย",0ah,0dh,04h
TEST: DFB "TEST",04H

```

```

0170 020153 LJMP WORKNORM4
0173 WORKNORM4:
0173 B40E08 CJNE A,#14,WORKNORM3
0176 1203A2 LCALL SPEAKPOS ;SPEAK POSITION
0179 020153 LJMP WORKNORM4
017C WORKNORM3:
017C B40FD4 CJNE A,#15,WORKNORM4
017F 1203A3 LCALL PRNTICK ;PRINT TICKET
0182 020153 LJMP WORKNORM4
0185 QWOERKNORM:
0185 22 RET

```

```

;.....
;ROUTINE FOR SHOW PROCESS MESSAGE
;.....
0186 SHW_PROCESS:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

MOV A,#04H
LCALL CURHI
MOV R5,#20 ;20 SCAN LINES
CLRLN22: MOV R4,#30 ;240 DOTS PER LINE
CLRLN21: MOV DPTR,#CNTLCM
MOV A,#0CH
MOVX @DPTR,A
DEC DPL
MOV A,#00H ;WRITE BLANK ( 8 DOTS/TIME )
MOVX @DPTR,A
DJNZ R4,CLRLN21
DJNZ R5,CLRLN22
RET

```

```

;WRITE 1 BYTE TO LCM
;DATA IN : ACC
;LOCATION AT PRESENT CURSOR POSITION NOT EFFECT.
;
WR1BYTE: MOV R1,A
MOV DPTR,#CNTLCM ;WRITE TO LCM
MOV A,#0CH
MOVX @DPTR,A
DEC DPL
MOV A,R1 ;WRITE 8 DOTS/TIME
MOVX @DPTR,A
RET

```

```

;SET CURSOR LOW ADDRESS POINTER
;DATA IN A
;NOTE : CURSOR POSITION = POSITION OF THE DOT ON LCM
; SEPARATE IN HIGH BYTE AND LOW BYTE
;
CURLOW: PUSH A ;SET POSITION OF CURSOR
MOV DPTR,#CNTLCM
MOV A,#0AH
MOVX @DPTR,A
DEC DPL
POP A
MOVX @DPTR,A
RET

```

```

;SET CURSOR HIGH ADDRESS POINTER
;DATA IN A
;
CURHI: PUSH A ;SET POSITION OF CURSOR
MOV DPTR,#CNTLCM
MOV A,#0BH
MOVX @DPTR,A
DEC DPL
POP A
MOVX @DPTR,A
RET

```

```

;SET CURSOR TO HOME POSITION
;
CURHOME: MOV R0,#CURROW ;INIT CURSOR POSITION
MOV @R0,#00 ;ROW 0 , COL 0
INC R0
MOV @R0,#00
LCALL CURCONV ;CONVERT TO HEX POSITION
RET

```

```

;SET CURSOR TO BOTTOM POSITION
;
CURBOTM: MOV R0,#CURROW
MOV @R0,#29
INC R0
LCALL CURCONV
RET

```

```

;CAL ROW NUMBER AND COL NUMBER IN HEX POSITION OF LCM
;DATA IN INTERNAL RAM 50H(ROW),51H(COL) STORE ROW,COL
;DATA IN INTERNAL RAM 52H(ROW),53H(COL) STORE HEX POS
;

```

```

CURCONV:
MOV R0,#CURROW ;GET ROW
MOV A,@R0
CJNE A,#00,CONV1 ;IF ROW 0
MOV DPTR,#240 ;COL 0 ARE 240
SJMP CONV2
CONV1: CJNE A,#01,CONV3 ;IF ROW 1
MOV DPTR,#840 ;COL 0 ARE 1080
SJMP CONV2
CONV3: CJNE A,#02,CONV4 ;IF ROW 2
MOV DPTR,#1440 ;COL 0 ARE 1440
SJMP CONV2
CONV4: MOV DPTR,#1800 ;IF NOT ABOVE,SET TO 1800
CONV2: INC R0 ;R0 = 51H
CLR C
MOV A,@R0 ;GET COL NO.
ADDC A,DPL ;ADD WITH COL 0
MOV DPL,A ;SAVE BACK
INC R0 ;R0 = 53H = HEX LO
INC R0
MOV @R0,A
MOV A,#00 ;IF CARRY
ADDC A,DPH ;ADD CARRY
MOV DPH,A ;SAVE BACK
DEC R0 ;R0 = 52H = HEX HI
MOV @R0,A

```

```

;THEN DISPLAY CURSOR AT SET CURSOR POSITION
MOV A,DPH
PUSH A
MOV A,DPL
LCALL CURLOW
POP A
LCALL CURHI
RET

```

```

;WRITE THAI CHARATER ( NORMAL LEVEL IS PAYANCHANA).
;AFTER DID IT,CURSOR POSITION INCREMENT BY ONE
;ASCII CODE IN ACC.
;CUR POS IN HEX IN 52H & 53H , TEMP USE 54H & 55H
;

```

```

WRCHR: MOV DPTR,#THAIPAT ;CAL START ADDRESS OF PATTERN
MOV B,#08H ;8 BYTE / CHARACTER
MUL AB ;ASCII CODE * 8
CLR C
ADDC A,DPL ;DPTR + DATA IN AB
MOV DPL,A ;DPL = DPL + A
MOV A,B ;DPH = DPH + B
ADDC A,DPH
MOV DPH,A
PUSH DPH ;SAVE POINTER WHICH POINT TO PATTERN
PUSH DPL

```

```

;TRANSFER HEX CUR POS IN TEMP
;
CLR C ;COPY CURRENT CUR POS TO TEMP
MOV R0,#22H ;BECAUSE IN THIS SUBROUTINE WE MUST
MOV R1,#24H ;MODIFY CURRENT POSITION IN TEMP
MOV A,@R0 ;(52) = (54) , (53) = (55)
MOV @R1,A
INC R0
INC R1
MOV A,@R0
MOV @R1,A

```

```

;WRITE PATTERN OF CHARACTER TO LCM
MOV R3,#08 ;WRITE 8 BYTE / CHARACTER
WRCHR1: MOV R0,#25H ;SET CURSOR ADDRESS
MOV A,@R0

```



```

LCALL CURLOW ;SET LOW ADDRESS
MOV R0,#24H
MOV A,@R0
LCALL CURHI ;SET HIGH ADDRESS
POP DPL ;GET POINTER BACK
POP DPH
MOV A,#00H
MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
INC DPTR ;NEXT BYTE OF PATTERN
PUSH DPH ;SAVE BACK
PUSH DPL
LCALL WR1BYTE ;WRITE DATA ON LCM 1 BYTE
LCALL ADD1LN ;SHIFT CURSOR TO NEXT SCAN LINE
DJNZ R3,WRCHR1 ;COMPLETE ?
POP DPH ;YES POP DPTR OUT
POP DPL
;INCREMENT CURSOR POSITION ;INCREMENT CURSOR TO NEXT POSITION
MOV R0,#CURCOL
INC @R0
LCALL CURCONV
RET

```

```

MOV @R0,A
JNC ADD8LN1
DEC R0 ; R0 = 54H
MOV A,@R0
INC A
MOV @R0,A
ADD8LN1: RET
;
;.....
;WRITE U,UU AFTER DID IT CURSOR STILL THE SAME
;.....
WRU: MOV DPTR,#THAIPT ;CAL START ADDRESS OF PATTERN
MOV B,#08H ; 8 BYTE / CHARACTER
MUL AB ;ASCII CODE * 8
CLR C
ADDC A,DPL ; DPTR + DATA IN AB
MOV DPL,A ; DPL = DPL + A
MOV A,B ; DPH = DPH + B
ADDC A,DPH
MOV DPH,A
PUSH DPH ;SAVE POINTER WHICH POINT TO PATTERN
PUSH DPL

```

```

;.....
;SHIFT CURSOR BACK 4 SCAN LINES FOR WRITE I,EE
;.....
SUB4LN: CLR C ;SUB 4 LINE SCAN
MOV R0,#25H ;GET LOW BYTE OF CUR POS
MOV A,@R0
SUBB A,#120 ;4 LINE SCAN
MOV @R0,A
JNC SUB4LN1
DEC R0 ; R0 = 54H
MOV A,@R0
DEC A
MOV @R0,A
SUB4LN1: RET

```

```

;TRANSFER HEX CUR POS IN TEMP
CLR C ;COPY CURRENT CUR POS TO TEMP
MOV R0,#22H
MOV R1,#24H
MOV A,@R0 ;(52) = (54) , (53) = (55)
MOV @R1,A
INC R0
INC R1
MOV A,@R0
MOV @R1,A
LCALL ADD8LN
MOV R3,#04 ;WRITE 4 BYTE / CHARACTER
WRU1: MOV R0,#25H ;SET CURSOR ADDRESS
MOV A,@R0
LCALL CURLOW
MOV R0,#24H
MOV A,@R0
LCALL CURHI
POP DPL ;GET POINTER BACK
POP DPH
MOV A,#00
MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
INC DPTR ;NEXT BYTE OF PATTERN
PUSH DPH ;SAVE BACK
PUSH DPL
LCALL WR1BYTE ;WRITE DATA ON LCM 1 BYTE
LCALL ADD1LN ;SHIFT CURSOR TO NEXT SCAN LINE
DJNZ R3,WRU1 ;COMPLETE ?
POP DPH ;YES POP DPTR OUT
POP DPL
RET

```

```

;.....
;SHIFT CURSOR BACK 8 SCAN LINES FOR WRITE EK,TOW
;.....
SUB8LN: CLR C ;SUB 8 LINE SCAN
MOV R0,#25H ;GET LOW BYTE OF CUR POS
MOV A,@R0
SUBB A,#240 ;8 LINE SCAN
MOV @R0,A
JNC SUB8LN1
DEC R0 ; R0 = 54H
MOV A,@R0
DEC A
MOV @R0,A
SUB8LN1: RET

```

```

;.....
;WRITE I,EE AT FIST LEVEL
;AFTER DID IT CUSOR STILL THE SAME
;.....
WRIEE: MOV DPTR,#THAIPT ;CAL START ADDRESS OF PATTERN
MOV B,#08H ; 8 BYTE / CHARACTER
MUL AB ;ASCII CODE * 8
CLR C
ADDC A,DPL ; DPTR + DATA IN AB
MOV DPL,A ; DPL = DPL + A
MOV A,B ; DPH = DPH + B
ADDC A,DPH
MOV DPH,A
PUSH DPH ;SAVE POINTER WHICH POINT TO PATTERN
PUSH DPL
;.....
;TRANSFER HEX CUR POS IN TEMP
;.....
CLR C ;COPY CURRENT CUR POS TO TEMP
MOV R0,#22H

```

```

;.....
;SHIFT CURSOR TO NEXT SCAN LINE FOR WRITE OTHER DOTS
;.....
ADD1LN: CLR C ;ADD 1 LINE SCAN
MOV R0,#25H ;GET LOW BYTE OF CUR POS
MOV A,@R0
ADDC A,#30 ;1 LINE SCAN
MOV @R0,A
JNC ADD1LN1
DEC R0
MOV A,@R0
INC A
MOV @R0,A
ADD1LN1: RET

```

```

;.....
;SHIFT CURSOR TO NEXT 8 SCAN LINES FOR WRITE U,UU
;.....
ADD8LN: CLR C ;ADD 8 LINE SCAN
MOV R0,#25H ;GET LOW BYTE OF CUR POS
MOV A,@R0
ADDC A,#240 ;8 LINE SCAN
MOV @R0,A
JNC ADD8LN1
DEC R0
MOV A,@R0
INC A
MOV @R0,A
ADD8LN1: RET

```

```

;.....
;SHIFT CURSOR TO NEXT 8 SCAN LINES FOR WRITE U,UU
;.....
ADD8LN: CLR C ;ADD 8 LINE SCAN
MOV R0,#25H ;GET LOW BYTE OF CUR POS
MOV A,@R0
ADDC A,#240 ;8 LINE SCAN
MOV @R0,A
JNC ADD8LN1
DEC R0
MOV A,@R0
INC A
MOV @R0,A
ADD8LN1: RET

```

```

;.....
;TRANSFER HEX CUR POS IN TEMP
;.....
CLR C ;COPY CURRENT CUR POS TO TEMP
MOV R0,#22H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV R1,#24H
MOV A,@R0      ;(52) = (54) , (53) = (55)
MOV @R1,A
INC R0
INC R1
MOV A,@R0
MOV @R1,A
;DEC 4 SCAN LINES DUE TO WRITE FROM TOP TO BOTTOM
LCALL SUB4LN
MOV R3,#04      ;WRITE 4 BYTE / CHARACTER
WR1: MOV R0,#25H ;SET CURSOR ADDRESS
MOV A,@R0
LCALL CURLOW
MOV R0,#24H
MOV A,@R0
LCALL CURHI
POP DPL         ;GET POINTER BACK
POP DPH
MOV A,#00
MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
INC DPTR       ;NEXT BYTE OF PATTERN
PUSH DPH       ;SAVE BACK
PUSH DPL
LCALL WR1BYTE  ;WRITE DATA ON LCM 1 BYTE
LCALL ADD1LN   ;SHIFT CURSOR TO NEXT SCAN LINE
DJNZ R3,WR1    ;COMPLETE ?
POP DPH        ;YES POP DPTR OUT
POP DPL
RET

```

```

;WRITE EK,TOW AT SECOND LEVEL
;AFTER DID IT CURSOR STILL THE SAME
;*****
WREKTOW: MOV DPTR,#THAIPTAT ;CAL START ADDRESS OF PATTERN
MOV B,#08H      ; 8 BYTE / CHARACTER
MUL AB          ; ASCII CODE * 8
CLR C
ADDC A,DPL      ; DPTR + DATA IN AB
MOV DPL,A      ; DPL = DPL + A
MOV A,B         ; DPH = DPH + B
ADDC A,DPH
MOV DPH,A
PUSH DPH       ;SAVE POINTER WHICH POINT TO PATTERN
PUSH DPL
;TRANSFER HEX CUR POS IN TEMP
CLR C          ;COPY CURRENT CUR POS TO TEMP
MOV R0,#22H
MOV R1,#24H
MOV A,@R0      ;(52) = (54) , (53) = (55)
MOV @R1,A
INC R0
INC R1
MOV A,@R0
MOV @R1,A
;DEC 8 SCAN LINES DUE TO WRITE FROM TOP TO BOTTOM
LCALL SUB8LN
MOV R3,#04      ;WRITE 4 BYTE / CHARACTER
WR2: MOV R0,#25H ;SET CURSOR ADDRESS
MOV A,@R0
LCALL CURLOW
MOV R0,#24H
MOV A,@R0
LCALL CURHI
POP DPL         ;GET POINTER BACK
POP DPH
MOV A,#00
MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
INC DPTR       ;NEXT BYTE OF PATTERN
PUSH DPH       ;SAVE BACK
PUSH DPL
LCALL WR1BYTE  ;WRITE DATA ON LCM 1 BYTE
LCALL ADD1LN   ;SHIFT CURSOR TO NEXT SCAN LINE
DJNZ R3,WR2    ;COMPLETE ?

```

```

POP DPH        ;YES POP DPTR OUT
POP DPL
RET
;*****
;WRITE DATA FROM CODE ROM
;DPTR POINT TO START ADDRESS OF BUFFER
;END OF DATA MUST BE 00H
;*****
WRTDAT:
WRTDAT2:
MOVX A,@DPTR   ;GET ASCII CHAR
CJNE A,#04H,WRTDAT1 ;IF EOT CODE
RET           ;RET
WRTDAT1: INC DPTR ;IF NOT,INC POINTER TO NEXT CHAR
PUSH DPH    ;AND SAVE
PUSH DPL
CJNE A,#0D1H,ZZ1 ;IF MAIHANARGARD
LJMP ZZ2
ZZ1: CJNE A,#0D4H,ZZ4 ;IF SARA-H
LJMP ZZ2
ZZ4: CJNE A,#0D5H,ZZ5 ;IF SARA-E
LJMP ZZ2
ZZ5: CJNE A,#0D6H,ZZ6 ;IF SARA-AU
LJMP ZZ2
ZZ6: CJNE A,#0D7H,ZZ7 ;IF SARA-AUU
LJMP ZZ2
ZZ7: CJNE A,#0E7H,ZZ8 ;IF MAITAKOO
LJMP ZZ2
ZZ8: CJNE A,#0E8H,ZZ9 ;IF KARAN
LJMP ZZ2
ZZ9: CJNE A,#0EDH,ZZ10 ;IF NARUKAHID
LJMP ZZ2
ZZ10: CJNE A,#0E9H,ZZ11 ;IF MAI-EK
LJMP ZZ2
ZZ11: CJNE A,#0E9H,ZZ12 ;IF MAI-TOW
LJMP ZZ2
ZZ12: CJNE A,#0EAH,ZZ13 ;IF MAI-TREE
LJMP ZZ2
ZZ13: CJNE A,#0EBH,ZZ14 ;IF MAI-JADTAVAR
LJMP ZZ2
ZZ14: CJNE A,#0D8H,ZZ15 ;IF SARA-U
LJMP ZZ2
ZZ15: CJNE A,#0D9H,ZZ16 ;IF SARA-UU
LJMP ZZ2
ZZ16: CJNE A,#0DAH,ZZ17 ;IF UNDERDOT
LJMP ZZ2
ZZ17: CJNE A,#0AH,ZZ18 ;IF LINE FEED
LJMP ZZ2
ZZ18: CJNE A,#0DH,ZZ19 ;IF CARIAGE RETURN
LJMP ZZ2
ZZ19: LCALL WRCHR ;IF PAYANCHANA OR SARA IN NORMAL LEVEL
ZZ3: POP DPL ;GET NEXT ASCII BACK
POP DPH
LJMP WRTDAT2 ;WRITE NEXT CHARACTER

```

```

;*****
ZZ2: PUSH A ;DISPLAY UPPER FIRST LEVEL
MOV R0,#CURCOL ;SHIFT CURSOR BACK 1 POSITION
DEC @R0
LCALL CURCONV ;CONVERT TO HEX
POP A
LCALL WRREE ;WRITE TO LCM
MOV R0,#CURCOL ;SHIFT CURSOR TO OLD POSITION
INC @R0
LCALL CURCONV ;CONVERT TO HEX
LJMP ZZ3
;*****
ZZ2: PUSH A ;DISPLAY UPPER SECOND LEVEL
MOV R0,#CURCOL ;SHIFT CURSOR BACK 1 POSITION
DEC @R0
LCALL CURCONV ;CONVERT TO HEX
POP A
LCALL WREKTOW ;WRITE TO UPPER SECOND LEVEL
MOV R0,#CURCOL ;SHIFT BACK 1 POSITION

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำออกจำหน่ายได้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC @R0
LCALL CURCONV ;CONVERT TO HEX
LJMP ZZ3
;
ZZ22: PUSH A ;DISPLAY IN LOWWER LEVEL
MOV R0,#CURCOL ;MOVE CURSOR BACK ONE POSITION
DEC @R0
LCALL CURCONV ;CONVERT TO HEX
POP A
LCALL WRU ;WRITE TO LCM
MOV R0,#CURCOL ;MOVE BACK ONE POSITION
INC @R0
LCALL CURCONV ;CONVERT TO HEX
LJMP ZZ3
;
ZZ23: MOV R0,#CURROW
INC @R0
LCALL CURCONV
LJMP ZZ3
ZZ24: MOV R0,#CURCOL
MOV @R0,#00H
LCALL CURCONV
LJMP ZZ3
;
;WRITE DATA FROM REG A AND CHECK LEVEL
;.....
WRFMBUF: CJNE A,#0D1H,AZZ1 ;IF MAIHANARGARD
LJMP AZZ2
AZZ1: CJNE A,#0D4H,AZZ4 ;IF SARA-I
LJMP AZZ2
AZZ4: CJNE A,#0D5H,AZZ5 ;IF SARA-E
LJMP AZZ2
AZZ5: CJNE A,#0D6H,AZZ6 ;IF SARA-AU
LJMP AZZ2
AZZ6: CJNE A,#0D7H,AZZ7 ;IF SARA-AUU
LJMP AZZ2
AZZ7: CJNE A,#0E7H,AZZ8 ;IF MAITAKOO
LJMP AZZ2
AZZ8: CJNE A,#0ECH,AZZ9 ;IF KARAN
LJMP AZZ2
AZZ9: CJNE A,#0EDH,AZZ10 ;IF NARUKAHID
LJMP AZZ2
AZZ10: CJNE A,#0E8H,AZZ11 ;IF MAI-EK
LJMP AZZ22
AZZ11: CJNE A,#0E9H,AZZ12 ;IF MAI-TOW
LJMP AZZ22
AZZ12: CJNE A,#0EAH,AZZ13 ;IF MAI-TREE
LJMP AZZ22
AZZ13: CJNE A,#0EBH,AZZ14 ;IF MAI-JADTAVAR
LJMP AZZ22
AZZ14: CJNE A,#0D8H,AZZ15 ;IF SARA-U
LJMP AZZ222
AZZ15: CJNE A,#0D9H,AZZ16 ;IF SARA-UU
LJMP AZZ222
AZZ16: CJNE A,#0DAH,AZZ17 ;IF UNDERDOT
LJMP AZZ222
AZZ17: CJNE A,#0AH,AZZ18 ;IF LINE FEED
LJMP AZZ223
AZZ18: CJNE A,#0DH,AZZ19 ;IF CARIAGE RETURN
LJMP AZZ224
AZZ19: LCALL WRCHR ;IF PAYANCHANA OR SARA IN NORMAL LEVEL
AZZ3: RET ;WRITE NEXT CHARACTER
;
AZZ2: *PUSH A
MOV R0,#CURCOL
DEC @R0
LCALL CURCONV
POP A
LCALL WRIEE ;WRITE TO LCM
MOV R0,#CURCOL
INC @R0
LCALL CURCONV
RET
;
AZZ22: PUSH A
MOV R0,#CURCOL
DEC @R0
LCALL CURCONV
POP A
LCALL WRU ;WRITE TO LCM
MOV R0,#CURCOL
INC @R0
LCALL CURCONV
RET
;
AZZ223: MOV R0,#CURROW ;CARIAGE RETURN
INC @R0
LCALL CURCONV
LJMP AZZ3
;
AZZ224: MOV R0,#CURCOL ;LINE FEED
MOV @R0,#00H
LCALL CURCONV
RET
INCL "THAIPT.ASM"
0186 12018D LCALL SHW_TIMEPRO
0189 1201AF LCALL SHW_DISTPRO
018C 22 RET
;.....
;ROUTINE FOR SHOW TIME IN PROCESS
;.....
018D SHW_TIMEPRO:
018D 120E6F LCALL RDTIME
0190 752000 MOV CURROW,#00
0193 752105 MOV CURCOL,#5
0196 1210C0 LCALL CURCONV
0199 1205C1 LCALL SHOWDATE
019C 7420 MOV A,#"
019E 1210FF LCALL WRCHR
01A1 1205E3 LCALL SHWTIME
01A4 743A MOV A,#";"
01A6 1210FF LCALL WRCHR
01A9 E515 MOV A,SECOND
01AB 1205B5 LCALL DISPBCD2
01AE 22 RET
;.....
;ROUTINE FOR SHOW DISTANCE IN PROCESS
;.....
01AF SHW_DISTPRO:
01AF 752001 MOV CURROW,#01
01B2 752100 MOV CURCOL,#00
01B5 1210C0 LCALL CURCONV
01B8 752500 MOV POS_H,#00
01BB 752450 MOV POS_L,#50H
01BE 900B52 MOV DPTR,#MSG_DISTP
01C1 1207E5 LCALL DSP_MSG
01C4 E51C MOV A,BUF_DIST
01C8 1201D2 LCALL DECBCD
01C9 7003 JNZ NXTDISTPRO
01CB 751C00 MOV BUF_DIST,#00
01CE NXTDISTPRO:
01CE 1205B5 LCALL DISPBCD2
01D1 22 RET
;.....
;ROUTINE FOR ADJUST DECIMAL TO ASCII

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; THAI PATTERN FOR SMO STANDARD

THAIPAT:

DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-000,00h  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-001,01H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-002,02H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-003,03H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-004,04H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-005,05H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-006,06H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-007,07H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-008,08H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-009,09H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-010,0AH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-011,0BH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-012,0CH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-013,0DH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-014,0EH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-015,0FH  
  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-016,10H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-017,11H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-018,12H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-019,13H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-020,14H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-021,15H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-022,16H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-023,17H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-024,18H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-025,19H  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-026,1AH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-027,1BH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-028,1CH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-029,1DH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-030,1EH  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-031,1FH  
  
 DFB 00H,00H,00H,00H,00H,00H,00H,00H : code-032,20H  
 DFB 0CH,1EH,1EH,0CH,0CH,00H,0CH,00H : code-033,21H,I  
 DFB 36H,36H,36H,24H,00H,00H,00H,00H : code-034,22H,"  
 DFB 36H,36H,7FH,36H,7FH,36H,36H,00H : code-035,23H,#  
 DFB 0CH,3EH,03H,1EH,30H,1FH,0CH,00H : code-036,24H,\$  
 DFB 00H,63H,33H,18H,0CH,66H,63H,00H : code-037,25H,%  
 DFB 0CH,12H,0CH,0CH,52H,22H,5CH,00H : code-038,26H,&  
 DFB 06H,06H,03H,00H,00H,00H,00H,00H : code-039,27H,'  
 DFB 18H,0CH,06H,06H,06H,0CH,18H,00H : code-040,28H,(  
 DFB 06H,0CH,18H,18H,18H,0CH,06H,00H : code-041,29H,)  
 DFB 00H,66H,3CH,0FFH,3CH,66H,00H,00H : code-042,2AH,\*  
 DFB 00H,0CH,0CH,3FH,0CH,0CH,00H,00H : code-043,2BH,+  
 DFB 00H,00H,00H,00H,00H,0CH,0CH,06H : code-044,2CH,,  
 DFB 00H,00H,00H,3FH,00H,00H,00H,00H : code-045,2DH,-  
 DFB 00H,00H,00H,00H,00H,0CH,0CH,00H : code-046,2EH,./  
 DFB 60H,30H,18H,0CH,06H,03H,01H,00H : code-047,2FH/

DFB 3EH,63H,73H,7BH,6FH,67H,3EH,00H : code-048,30H,0  
 DFB 0CH,0EH,0CH,0CH,0CH,0CH,3FH,00H : code-049,31H,1  
 DFB 1EH,33H,30H,1CH,06H,33H,3FH,00H : code-050,32H,2  
 DFB 1EH,33H,30H,1CH,30H,33H,1EH,00H : code-051,33H,3  
 DFB 3BH,3CH,3BH,33H,7FH,30H,7BH,00H : code-052,34H,4  
 DFB 3FH,03H,1FH,30H,30H,33H,1EH,00H : code-053,35H,5  
 DFB 1CH,06H,03H,1FH,33H,33H,1EH,00H : code-054,36H,6  
 DFB 3FH,33H,30H,18H,0CH,0CH,0CH,00H : code-055,37H,7  
 DFB 1EH,33H,33H,1EH,33H,33H,1EH,00H : code-056,38H,8  
 DFB 1EH,33H,33H,3EH,30H,18H,0EH,00H : code-057,39H,9  
 DFB 00H,0CH,0CH,00H,00H,0CH,0CH,00H : code-058,3AH,:  
 DFB 00H,0CH,0CH,00H,00H,0CH,0CH,06H : code-059,3BH,:  
 DFB 18H,0CH,06H,03H,06H,0CH,18H,00H : code-060,3CH,<  
 DFB 00H,00H,3FH,00H,00H,00H,3FH,00H,00H : code-061,3DH,=  
 DFB 06H,0CH,18H,30H,18H,0CH,06H,00H : code-062,3EH,>  
 DFB 1EH,33H,30H,18H,0CH,00H,0CH,00H : code-063,3FH,?  
 DFB 3EH,63H,7BH,7BH,7BH,03H,3EH,00H : code-064,40H,@  
 DFB 0CH,1EH,33H,33H,3FH,33H,33H,00H : code-065,41H,OA  
 DFB 3FH,66H,66H,3FH,66H,66H,3FH,00H : code-066,42H,OB  
 DFB 3CH,66H,03H,03H,03H,66H,3CH,00H : code-067,43H,OC

DFB 1FH,36H,66H,66H,66H,36H,1FH,00H : code-068,44H,OD  
 DFB 7FH,46H,16H,1EH,16H,46H,7FH,00H : code-069,45H,OE  
 DFB 7FH,46H,16H,1EH,16H,06H,0FH,00H : code-070,46H,OF  
 DFB 3CH,66H,03H,03H,7BH,66H,7CH,00H : code-071,47H,OG  
 DFB 33H,33H,33H,3FH,33H,33H,33H,00H : code-072,48H,OH  
 DFB 1EH,0CH,0CH,0CH,0CH,0CH,1EH,00H : code-073,49H,I  
 DFB 78H,30H,30H,30H,33H,33H,1EH,00H : code-074,4AH,J  
 DFB 67H,66H,1EH,1EH,36H,66H,67H,00H : code-075,4BH,K  
 DFB 0FH,06H,06H,06H,46H,66H,7FH,00H : code-076,4CH,L  
 DFB 63H,77H,7FH,7FH,6BH,63H,63H,00H : code-077,4DH,M  
 DFB 63H,67H,67H,7BH,73H,63H,63H,00H : code-078,4EH,N  
 DFB 1CH,36H,63H,63H,63H,36H,1CH,00H : code-079,4FH,OO  
  
 DFB 3FH,66H,66H,3EH,06H,06H,0FH,00H : code-080,50H,P  
 DFB 1EH,33H,33H,33H,33H,1EH,3BH,00H : code-081,51H,Q  
 DFB 3FH,66H,66H,3EH,36H,66H,67H,00H : code-082,52H,R  
 DFB 1EH,33H,06H,0CH,18H,33H,1EH,00H : code-083,53H,S  
 DFB 3FH,2DH,0CH,0CH,0CH,0CH,1EH,00H : code-084,54H,T  
 DFB 33H,33H,33H,33H,33H,33H,3FH,00H : code-085,55H,U  
 DFB 33H,33H,33H,33H,33H,1EH,0CH,00H : code-086,56H,V  
 DFB 63H,63H,63H,6AH,7FH,77H,63H,00H : code-087,57H,W  
 DFB 63H,63H,36H,1CH,1CH,36H,63H,00H : code-088,58H,X  
 DFB 33H,33H,33H,1EH,0CH,0CH,1EH,00H : code-089,59H,Y  
 DFB 7FH,63H,31H,18H,4CH,66H,6FH,00H : code-090,5AH,Z  
 DFB 1EH,06H,06H,06H,06H,06H,1EH,00H : code-091,5BH,[  
 DFB 03H,06H,0CH,18H,30H,60H,40H,00H : code-092,5CH,\  
 DFB 1EH,18H,18H,18H,18H,18H,1EH,00H : code-093,5DH,]  
 DFB 08H,1CH,36H,63H,00H,00H,00H : code-094,5EH,^  
 DFB 00H,00H,00H,00H,00H,00H,0FFH : code-095,5FH,\_  
  
 DFB 0CH,0CH,18H,00H,00H,00H,00H : code-096,60H,\*  
 DFB 00H,00H,1EH,30H,3EH,33H,6EH,00H : code-097,61H,a  
 DFB 07H,06H,06H,3EH,66H,66H,3BH,00H : code-098,62H,b  
 DFB 00H,00H,1EH,33H,03H,33H,1EH,00H : code-099,63H,c  
 DFB 38H,30H,30H,3EH,33H,33H,6EH,00H : code-100,64H,d  
 DFB 00H,00H,1EH,33H,3FH,03H,1EH,00H : code-101,65H,e  
 DFB 1CH,36H,06H,0FH,06H,06H,0FH,00H : code-102,66H,f  
 DFB 00H,00H,6EH,31H,31H,3EH,30H,1FH : code-103,67H,g  
 DFB 07H,06H,36H,6EH,66H,66H,67H,00H : code-104,68H,h  
 DFB 0CH,00H,0EH,0CH,0CH,0CH,1EH,00H : code-105,69H,i  
 DFB 30H,00H,30H,30H,30H,33H,33H,1EH : code-106,6AH,j  
 DFB 07H,06H,65H,36H,1EH,36H,67H,00H : code-107,6BH,k  
 DFB 0EH,0CH,0CH,0CH,0CH,0CH,1EH,00H : code-108,6CH,l  
 DFB 00H,00H,33H,7FH,7FH,6BH,63H,00H : code-109,6DH,m  
 DFB 00H,00H,1FH,33H,33H,33H,33H,00H : code-110,6EH,n  
 DFB 00H,00H,1EH,33H,33H,33H,1EH,00H : code-111,6FH,o  
  
 DFB 00H,00H,3BH,66H,66H,3EH,06H,0FH : code-112,70H,p  
 DFB 00H,00H,6EH,33H,33H,3EH,30H,7BH : code-113,71H,q  
 DFB 00H,00H,3BH,6EH,66H,06H,0FH,00H : code-114,72H,r  
 DFB 00H,00H,3EH,03H,1EH,30H,1FH,00H : code-115,73H,s  
 DFB 08H,0CH,3EH,0CH,0CH,2CH,18H,00H : code-116,74H,t  
 DFB 00H,00H,33H,33H,33H,33H,6EH,00H : code-117,75H,u  
 DFB 00H,00H,33H,33H,33H,1EH,0CH,00H : code-118,76H,v  
 DFB 00H,00H,63H,6BH,7FH,7FH,36H,00H : code-119,77H,w  
 DFB 00H,00H,63H,36H,1CH,36H,63H,00H : code-120,78H,x  
 DFB 00H,00H,33H,33H,33H,3EH,30H,1FH : code-121,79H,y  
 DFB 00H,00H,3FH,19H,0CH,26H,3FH,00H : code-122,7AH,z  
 DFB 3BH,0CH,0CH,07H,0CH,0CH,3BH,00H : code-123,7BH,{  
 DFB 18H,18H,18H,00H,18H,18H,18H,00H : code-124,7CH,|  
 DFB 07H,0CH,0CH,3BH,0CH,0CH,07H,00H : code-125,7DH,}  
 DFB 0CEH,3BH,00H,00H,00H,00H,00H : code-126,7EH,~  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-127,7FH

DFB 00H,00H,00H,00H,00H,00H,00H : code-128,80H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-129,81H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-130,82H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-131,83H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-132,84H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-133,85H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-134,86H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-135,87H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-136,88H  
 DFB 00H,00H,00H,00H,00H,00H,00H : code-137,89H

โครงการพัฒนาระบบงานเพื่อการศึกษาและส่งเสริมการศึกษานานาชาติ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-138,8AH	DFB 00H,44H,2AH,1CH,44H,2AH,1CH,00H	: smo-208,0D0H,SARA-A
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-139,8BH	DFB 00H,88H,54H,38H,00H,00H,00H,00H	: smo-209,0D1H,MAIHANARGARD
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-140,8CH	DFB 00H,18H,24H,20H,20H,20H,20H,00H	: smo-210,0D2H,SARA-AR
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-141,8DH	DFB 03H,18H,24H,20H,20H,20H,20H,00H	: smo-211,0D3H,SARA-AM
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-142,8EH	DFB 00H,10H,28H,7CH,00H,00H,00H,00H	: smo-212,0D4H,SARA-I
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-143,8FH	DFB 40H,50H,68H,7CH,00H,00H,00H,00H	: smo-213,0D5H,SARA-E
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-144,90H	DFB 00H,50H,0A8H,7CH,00H,00H,00H,00H	: smo-214,0D6H,SARA-AU
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-145,91H	DFB 00H,0A8H,0B4H,7EH,00H,00H,00H,00H	: smo-215,0D7H,SARA-AUU
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-146,92H	DFB 20H,50H,60H,40H,00H,00H,00H,00H	: smo-216,0D8H,SARA-U
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-147,93H	DFB 48H,54H,58H,20H,00H,00H,00H,00H	: smo-217,0D9H,SARA-UU
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-148,94H	DFB 00H,00H,00H,00H,00H,20H,50H,20H	: smo-218,0DAH,
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-149,95H	DFB 00H,00H,00H,0FFH,00H,00H,00H,00H	: smo-219,0DBH,
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-150,96H	DFB 08H,08H,08H,0FFH,00H,00H,00H,00H	: smo-220,0DCH,
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-151,97H	DFB 00H,00H,00H,0FFH,08H,08H,08H,08H	: smo-221,0DDH,
DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-152,98H	DFB 08H,08H,08H,0FFH,08H,08H,08H,08H	: smo-222,0DEH,
DFB 00H,00H,00H,0F8H,08H,08H,08H,08H	: code-153,99H	DFB 0BFH,66H,66H,3EH,6EH,66H,3EH,01H	: smo-223,0DFH,baht
DFB 00H,00H,00H,0FH,08H,08H,08H,08H	: code-154,9AH	DFB 00H,08H,08H,08H,18H,28H,10H,00H	: smo-224,0E0H,SARA-A
DFB 08H,08H,08H,0F8H,00H,00H,00H,00H	: code-155,9BH	DFB 00H,12H,12H,12H,36H,5AH,24H,00H	: smo-225,0E1H,SARA-AJR
DFB 08H,08H,08H,0FH,00H,00H,00H,00H	: code-156,9CH	DFB 0FCH,02H,1CH,10H,30H,50H,20H,00H	: smo-226,0E2H,SARA-O
DFB 08H,08H,08H,08H,08H,08H,08H,08H	: code-157,9DH	DFB 1CH,22H,24H,10H,30H,50H,20H,00H	: smo-227,0E3H,SARA-AIMAIMUAN
DFB 08H,08H,08H,0F8H,08H,08H,08H,08H	: code-158,9EH	DFB 0AH,14H,10H,10H,30H,50H,20H,00H	: smo-228,0E4H,SARA-AIMAIMALAJ
DFB 08H,08H,08H,0FH,08H,08H,08H,08H	: code-159,9FH	DFB 00H,18H,24H,20H,20H,20H,20H,20H	: smo-229,0E5H,
DFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH	: smo-160,0A0H,all black	DFB 00H,14H,2AH,24H,30H,08H,08H,08H	: smo-230,0E6H,MAIYAMOK
DFB 00H,3CH,42H,44H,44H,44H,44H,00H	: smo-161,0A1H,kokai	DFB 80H,58H,0A8H,58H,00H,00H,00H,00H	: smo-231,0E7H,MAI-TAIKOO
DFB 00H,48H,54H,58H,50H,50H,30H,00H	: smo-162,0A2H,khokhal	DFB 00H,40H,40H,40H,00H,00H,00H,00H	: smo-232,0E8H,MAI-EK
DFB 00H,5AH,55H,52H,50H,50H,30H,00H	: smo-163,0A3H,khokhuad	DFB 00H,98H,50H,38H,00H,00H,00H,00H	: smo-233,0E9H,MAI-TOR
DFB 00H,3CH,42H,42H,5AH,56H,44H,00H	: smo-164,0A4H,korkwai	DFB 80H,0A8H,54H,48H,00H,00H,00H,00H	: smo-234,0EAH,MAI-TREE
DFB 00H,24H,5AH,42H,5AH,56H,44H,00H	: smo-165,0A5H,korkon	DFB 00H,40H,0E0H,40H,00H,00H,00H,00H	: smo-235,0EBH,MAI-JATTAVAR
DFB 00H,4AH,55H,50H,4CH,4AH,34H,00H	: smo-166,0A6H,korrakang	DFB 0C0H,30H,28H,10H,00H,00H,00H,00H	: smo-236,0ECH,KARAN
DFB 00H,20H,54H,64H,44H,48H,30H,00H	: smo-167,0A7H,ngomgoo	DFB 00H,00H,00H,00H,40H,40H,0A0H,40H	: smo-237,0EDH,
DFB 00H,3CH,40H,48H,54H,58H,30H,00H	: smo-168,0A8H,jorjam	DFB 00H,00H,00H,00H,80H,80H,20H	: smo-238,0EEH,
DFB 00H,1EH,21H,24H,6AH,0ACH,58H,00H	: smo-169,0A9H,chorching	DFB 00H,00H,1CH,22H,2AH,22H,1CH,00H	: smo-239,0EFH,
DFB 80H,48H,34H,58H,50H,48H,70H,00H	: smo-170,0AAH,chorchang	DFB 00H,3CH,42H,42H,42H,42H,3CH,00H	: smo-240,0F0H,thai 0
DFB 80H,4AH,35H,32H,50H,48H,70H,00H	: smo-171,0ABH,sorso	DFB 00H,3CH,42H,5AH,5EH,40H,38H,00H	: smo-241,0F1H,thai 1
DFB 00H,5EH,0A1H,0A2H,0B3H,0ABH,0D2H,00H	: smo-172,0ACH,chorakachur	DFB 00H,54H,6CH,55H,4AH,42H,3CH,00H	: smo-242,0F2H,thai 2
DFB 00H,8EH,91H,92H,93H,0F5H,82H,0F0H	: smo-173,0ADH,yorying	DFB 00H,6CH,6AH,52H,46H,4AH,24H,00H	: smo-243,0F3H,thai 3
DFB 00H,38H,44H,48H,4CH,4AH,0C4H,7EH	: smo-174,0AEH,chorchadar	DFB 80H,7CH,02H,12H,2AH,1AH,7CH,00H	: smo-244,0F4H,thai 4
DFB 00H,38H,44H,48H,4CH,4AH,0F4H,2AH	: smo-175,0AFH,torpatak	DFB 0B0H,7CH,02H,12H,2AH,1AH,7CH,00H	: smo-245,0F5H,thai 5
DFB 40H,3CH,62H,48H,54H,58H,22H,75H	: smo-176,0B0H,tortam	DFB 01H,3EH,40H,44H,4AH,46H,3CH,00H	: smo-246,0F6H,thai 6
DFB 00H,6AH,55H,4AH,48H,48H,48H,00H	: smo-177,0B1H,tomangmonto	DFB 80H,94H,0AAH,6AH,66H,6AH,64H,00H	: smo-247,0F7H,thai 7
DFB 00H,9AH,0A5H,0A5H,0BBH,0ADH,56H,00H	: smo-178,0B2H,torpoota	DFB 80H,7CH,02H,0AH,2AH,5AH,34H,00H	: smo-248,0F8H,thai 8
DFB 00H,4EH,51H,52H,53H,0B5H,52H,00H	: smo-179,0B3H,nomend	DFB 80H,54H,3AH,12H,26H,4AH,04H,00H	: smo-249,0F9H,thai 9
DFB 00H,3CH,42H,4AH,56H,5AH,4CH,00H	: smo-180,0B4H,dordex	DFB 00H,5AH,5DH,56H,50H,28H,14H,00H	: smo-250,0FAH,thai
DFB 00H,34H,4AH,4AH,56H,5AH,4CH,00H	: smo-181,0B5H,tortoa	DFB 00H,00H,02H,15H,56H,0AEH,00H,00H	: smo-251,0FBH,thai
DFB 00H,3CH,42H,44H,46H,4AH,44H,00H	: smo-182,0B6H,tortung	DFB 10H,30H,70H,0FFH,70H,30H,10H,00H	: smo-252,0FCH,thai
DFB 00H,34H,4AH,4CH,48H,48H,48H,00H	: smo-183,0B7H,tortaham	DFB 01H,03H,07H,0FFH,07H,03H,01H,00H	: smo-253,0FDH,thai
DFB 40H,38H,7CH,48H,48H,48H,1CH,00H	: smo-184,0B8H,tortong	DFB 01H,1CH,3EH,7FH,01H,01H,01H,01H	: smo-254,0FEH,thai
DFB 00H,24H,2AH,2CH,168H,0A4H,5CH,00H	: smo-185,0B9H,nomoo	DFB 01H,01H,01H,01H,7FH,3EH,1CH,01H	: smo-255,0FFH,thai
DFB 00H,44H,4AH,4CH,48H,48H,38H,00H	: smo-186,0BAH,borbaimai		
DFB 40H,44H,4AH,4CH,48H,48H,38H,00H	: smo-187,0BBH,porpla		
DFB 00H,48H,54H,4CH,54H,54H,28H,00H	: smo-188,0BCH,porpaung		
DFB 40H,48H,54H,4CH,54H,54H,28H,00H	: smo-189,0BDH,forfa		
DFB 00H,42H,45H,46H,54H,54H,28H,00H	: smo-190,0BEH,porpam		
DFB 40H,42H,45H,46H,54H,54H,28H,00H	: smo-191,0BFH,fortun		
DFB 00H,38H,44H,48H,4CH,4AH,4CH,00H	: smo-192,0C0H,porsampoa		
DFB 00H,44H,4AH,4CH,4CH,5AH,24H,00H	: smo-193,0C1H,morma		
DFB 00H,24H,2AH,26H,2CH,24H,1CH,00H	: smo-194,0C2H,yoryak		
DFB 00H,7CH,02H,3CH,30H,28H,10H,00H	: smo-195,0C3H,romu		
DFB 00H,1CH,22H,24H,26H,2AH,24H,20H	: smo-196,0C4H,tormu		
DFB 00H,3CH,42H,5CH,66H,6AH,24H,00H	: smo-197,0C5H,loring		
DFB 00H,1CH,22H,24H,26H,25H,22H,20H	: smo-198,0C6H,Jormu		
DFB 00H,78H,44H,40H,60H,50H,20H,00H	: smo-199,0C7H,vorvan		
DFB 40H,7CH,42H,52H,6AH,5AH,4CH,00H	: smo-200,0C8H,sorsatar		
DFB 00H,44H,6AH,0ECH,48H,44H,78H,00H	: smo-201,0C9H,sornusi		
DFB 40H,3CH,42H,5CH,66H,6AH,24H,00H	: smo-202,0CAH,sorsua		
DFB 00H,22H,55H,26H,54H,52H,5CH,00H	: smo-203,0CBH,horheep		
DFB 0A0H,52H,65H,48H,54H,54H,28H,00H	: smo-204,0CCH,lorjular		
DFB 00H,3CH,42H,44H,4AH,46H,3CH,00H	: smo-205,0CDH,orang		
DFB 60H,3CH,5EH,44H,42H,46H,3CH,00H	: smo-206,0CEH,hormkhook		
DFB 00H,24H,52H,52H,4CH,40H,40H,70H	: smo-207,0CFH,paiyarnoi		

01D2	DECBCD:	CLR	C
01D2 C3		SUBB	A,#100
01D3 9464		JC	NXTDECB
01D5 4004		MOV	A,#00
01D7 7400		SJMP	QDECB
01D9 8009			
01DB	NXTDECB		
01DB 2464		ADD	A,#100
01DD 75F00A		MOV	B,#10
01E0 84		DIV	AB
01E1 C4		SWAP	A
01E2 45F0		ORL	A,B
01E4	QDECB		
01E4 22		RET	

:ROUTINE FOR CHECK TIME BEFORE START TRAVEL

01E5	TIMESSET:		
01E5 120E6F		LCALL	RDTIME
01E5 120E6F			:READ OLD VALUE
01E8 120351		LCALL	SHW_TIMESSET

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

01EB LPTIMESET:
01EB 1200A1 LCALL WAITKEY
01EE B40A03 CJNE A,#10,TIMESET_1
01F1 0201FD LJMP QTIMESET ;CORRECT DATA
01F4 TIMESET_1:
01F4 B40BF4 CJNE A,#11,LPTIMESET
01F7 1201FE LCALL PROGALLTIME ;CHANGE CURRENT TIME
01FA 0201E5 LJMP TIMESET
01FD QTIMESET:
01FD 22 RET

;*****
;ROUTINE FOR PROGRAM TIME & DATE
;*****
01FE PROGALLTIME:
01FE 120205 LCALL PROGDATE
0201 1202AC LCALL PROGTIME
0204 22 RET

;*****
;ROUTINE FOR PROGRAM DATE ,MONTH ,YEAR (BUDDHA)
;*****
0205 PROGDATE:
0205 12100A LCALL CLRSCRN
0208 752000 MOV CURROW,#00 ;SHOW AT LINE 0
020B 752100 MOV CURCOL,#00
020E 1210C0 LCALL CURCONV
0211 900AF8 MOV DPTR,#MSG_DATE
0214 752500 MOV POS_H,#00H
0217 752400 MOV POS_L,#00H ;SHOW AT LINE 0 PAGE 0
021A 1207E5 LCALL DSP_MSG
021D 1205C1 LCALL SHOWDATE
0220 752001 MOV CURROW,#01
0223 752100 MOV CURCOL,#00
0226 1210C0 LCALL CURCONV
0229 900B2D MOV DPTR,#MSG_DATEF ;SHOW FORM TO FILL DATA
022C 752500 MOV POS_H,#00
022F 752450 MOV POS_L,#50H
0232 1207E5 LCALL DSP_MSG

0235 752001 MOV CURROW,#01
0238 752108 MOV CURCOL,#08
023B 1210C0 LCALL CURCONV ;BACK SPACE
023E 120321 LCALL IN2BCD
0241 6046 JZ QPROGDATE
0243 F512 MOV DATE,A ;LOAD DATE

0245 752108 MOV CURCOL,#11
0248 1210C0 LCALL CURCONV
024B 120321 LCALL IN2BCD
024E 6039 JZ QPROGDATE
0250 F513 MOV MONTH,A ;LOAD MONTH

0252 752110 MOV CURCOL,#16
0255 1210C0 LCALL CURCONV
0258 120321 LCALL IN2BCD
025B 602C JZ QPROGDATE
025D F514 MOV YEAR,A ;LOAD YEAR
025F 12028A LCALL CHRISTADJ

0262 752002 MOV CURROW,#02
0265 752100 MOV CURCOL,#00
0268 1210C0 LCALL CURCONV
026B 752500 MOV POS_H,#00
026E 7524A0 MOV POS_L,#0A0H
0271 900B0B MOV DPTR,#MSG_SETTM
0274 1207E5 LCALL DSP_MSG
0277 LPPROGDATE1:
0277 1200A1 LCALL WAITKEY
027A B40A06 CJNE A,#10,PROGDATE1
027D 120E19 LCALL SETDATE
0280 020289 LJMP QPROGDATE
0283 PROGDATE1:
0283 B40BF1 CJNE A,#11,LPPROGDATE1
0286 020205 LJMP PROGDATE
0289 QPROGDATE:
0289 22 RET

028A CHRISTADJ:
028A E514 MOV A,YEAR
028C C3 CLR C
028D 9443 SUBB A,#43H
028F F514 MOV YEAR,A
0291 5005 JNC CHRISTADJ1
0293 C3 CLR C
0294 9460 SUBB A,#60H
0296 F514 MOV YEAR,A ;TEMPORARY STORE
0298 CHRISTADJ1:
0298 540F ANL A,#0FH
029A C3 CLR C
029B 940A SUBB A,#10
029D 400C JC QCHRISTADJ
029F 2404 ADD A,#4 ;SUB WITH 6
02A1 F516 MOV TMP_BUF0,A
02A3 E514 MOV A,YEAR
02A5 54F0 ANL A,#0F0H
02A7 4516 ORL A,TMP_BUF0
02A9 F514 MOV YEAR,A
02AB QCHRISTADJ:
02AB 22 RET

;*****
;ROUTINE FOR PROGRAM TIME HOUR (24),MINUTE
;*****
02AC PROGTIME:
02AC 12100A LCALL CLRSCRN
02AF 752000 MOV CURROW,#00 ;SHOW AT LINE 0
02B2 752100 MOV CURCOL,#00
02B5 1210C0 LCALL CURCONV
02B8 900B02 MOV DPTR,#MSG_TIME
02BB 752500 MOV POS_H,#00H
02BE 752400 MOV POS_L,#00H ;SHOW AT LINE 0 PAGE 0
02C1 1207E5 LCALL DSP_MSG
02C4 1205E3 LCALL SHOWTIME
02C7 752001 MOV CURROW,#01
02CA 752100 MOV CURCOL,#00
02CD 1210C0 LCALL CURCONV
02D0 900B43 MOV DPTR,#MSG_TIMEF ;SHOW FORM TO FILL DATA
02D3 752500 MOV POS_H,#00
02D6 752450 MOV POS_L,#50H
02D9 1207E5 LCALL DSP_MSG

02DC 752001 MOV CURROW,#01
02DF 752108 MOV CURCOL,#08
02E2 1210C0 LCALL CURCONV ;BACK SPACE
02E5 120321 LCALL IN2BCD
02E8 6036 JZ QPROGTIME
02EA F511 MOV HOUR,A ;LOAD HOUR

02EC 752108 MOV CURCOL,#11
02EF 1210C0 LCALL CURCONV
02F2 120321 LCALL IN2BCD
02F5 6029 JZ QPROGTIME
02F7 F510 MOV MINUTE,A ;LOAD MINUTE

02F9 752002 MOV CURROW,#02
02FC 752100 MOV CURCOL,#00
02FF 1210C0 LCALL CURCONV
0302 752500 MOV POS_H,#00
0305 7524A0 MOV POS_L,#0A0H
0308 900B0B MOV DPTR,#MSG_SETTM
030B 1207E5 LCALL DSP_MSG
030E LPPROGTIME1:
030E 1200A1 LCALL WAITKEY
0311 B40A06 CJNE A,#10,PROGTIME1
0314 120DEF LCALL SETTIME
0317 020320 LJMP QPROGTIME
031A PROGTIME1:
031A B40BF1 CJNE A,#11,LPPROGTIME1
031D 0202AC LJMP PROGTIME
0320 QPROGTIME:
0320 22 RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.....
;ROUTINE FOR READ 2 BYTE OF BCD
.....
0321      IN2BCD:
0321 1200A1  LCALL WAITKEY      ;READ HIGH DIGIT
0324 B40C05  CJNE A,#12,NXTIN2BCD ;ESCAPE KEY
0327 7400    MOV A,#00
0329 020350  LJMP QIN2BCD
032C      NXTIN2BCD:
032C C3      CLR C
032D 940A    SUBB A,#10
032F 50F0    JNC IN2BCD
0331 240A    ADD A,#10
0333 F517    MOV TMP_BUF1,A
0335 2430    ADD A,#30H      ;CONVERT TO ASCII
0337 1210FF  LCALL WRCHR
033A      LPIN2BCD:
033A 1200A1  LCALL WAITKEY      ;READ LOWER DIGIT
033D C3      CLR C
033E 940A    SUBB A,#10
0340 50DF    JNC IN2BCD
0342 240A    ADD A,#10
0344 F516    MOV TMP_BUF0,A
0346 2430    ADD A,#30H      ;CONVERT TO ASCII
0348 1210FF  LCALL WRCHR

034B E517    MOV A,TMP_BUF1
034D C4      SWAP A
034E 4516    ORL A,TMP_BUF0
0350      QIN2BCD:
0350 22      RET

.....
;ROUTINE FOR SHOW TIME MESSAGE
.....
0351      SHW_TIMESET:
0351 12059A  LCALL RAMCLR
0354 12100A  LCALL CLRSCRN
0357 1210AC  LCALL CURHOME

035A 752000  MOV CURROW,#00      ;SHOW AT LINE 0
035D 752100  MOV CURCOL,#00
0360 1210C0  LCALL CURCONV
0363 900AF8  MOV DPTR,#MSG_DATE
0366 752500  MOV POS_H,#00H
0369 752400  MOV POS_L,#00H      ;SHOW AT LINE 0 PAGE 0
036C 1207E5  LCALL DSP_MSG
036F 1205C1  LCALL SHOWDATE

0372 752001  MOV CURROW,#01
0375 752100  MOV CURCOL,#00
0378 1210C0  LCALL CURCONV
037B 900B02  MOV DPTR,#MSG_TIME
037E 752500  MOV POS_H,#00H
0381 752450  MOV POS_L,#50H      ;SHOW AT LINE 1
0384 1207E5  LCALL DSP_MSG
0387 1205E3  LCALL SHOWTIME

038A 752002  MOV CURROW,#02
038D 752100  MOV CURCOL,#00
0390 1210C0  LCALL CURCONV
0393 900B0B  MOV DPTR,#MSG_SETTM
0396 752500  MOV POS_H,#00H
0399 7524A0  MOV POS_L,#0A0H      ;SHOW AT LINE 2
039C 1207E5  LCALL DSP_MSG
039F 22      RET

.....
;ROUTINE CHECK SMART CARD IS VALID OR NOT
.....
03A0      CHKCRD:
03A0 22      RET

.....
;ROUTINE PROCESS WITH SMART CARD USAGE
.....
03A1      CARDPRO:
03A1 22      RET

.....
;ROUTINE SPEAK POSITION WITH OK
.....
03A2      SPEAKPOS:
03A2 22      RET

.....
;ROUTINE PRINT TICKET
.....
03A3      PRNTICK:
03A3 22      RET

.....
;ROUTINE FOR SHOW ROUTE
.....
03A4      SHW_ROUTE:
03A4 12059A  LCALL RAMCLR
03A7 12100A  LCALL CLRSCRN
03AA 1210AC  LCALL CURHOME

03AD 752000  MOV CURROW,#00      ;SHOW AT LINE 0
03B0 752100  MOV CURCOL,#00
03B3 1210C0  LCALL CURCONV
03B6 9008F3  MOV DPTR,#MSG_ROUTE1
03B9 752500  MOV POS_H,#00H
03BC 752400  MOV POS_L,#00H      ;SHOW AT LINE 0 PAGE 0
03BF 1207E5  LCALL DSP_MSG

03C2 752001  MOV CURROW,#01
03C5 752100  MOV CURCOL,#00
03C8 1210C0  LCALL CURCONV
03CB 900911  MOV DPTR,#MSG_ROUTE2
03CE 752500  MOV POS_H,#00H
03D1 752450  MOV POS_L,#50H      ;SHOW AT LINE 1
03D4 1207E5  LCALL DSP_MSG

03D7 752002  MOV CURROW,#02
03DA 752100  MOV CURCOL,#00
03DD 1210C0  LCALL CURCONV
03E0 900934  MOV DPTR,#MSG_ROUTE3
03E3 752500  MOV POS_H,#00H
03E6 7524A0  MOV POS_L,#0A0H      ;SHOW AT LINE 2
03E9 1207E5  LCALL DSP_MSG
03EC 22      RET

.....
;ROUTINE FOR ROUTE MENU
.....
03ED      SHW_RMENU:
03ED 12059A  LCALL RAMCLR
03F0 12100A  LCALL CLRSCRN
03F3 1210AC  LCALL CURHOME

03F6 752000  MOV CURROW,#00      ;SHOW AT LINE 0
03F9 752100  MOV CURCOL,#00
03FC 1210C0  LCALL CURCONV
03FF 900953  MOV DPTR,#MSG_RMENU1
0402 752500  MOV POS_H,#00H
0405 752400  MOV POS_L,#00H      ;SHOW AT LINE 0 PAGE 0
0408 1207E5  LCALL DSP_MSG

040B 752001  MOV CURROW,#01
040E 752100  MOV CURCOL,#00
0411 1210C0  LCALL CURCONV
0414 900972  MOV DPTR,#MSG_RMENU2
0417 752500  MOV POS_H,#00H
041A 752450  MOV POS_L,#50H      ;SHOW AT LINE 1
041D 1207E5  LCALL DSP_MSG

0420 752002  MOV CURROW,#02
0423 752100  MOV CURCOL,#00
0426 1210C0  LCALL CURCONV
0429 900995  MOV DPTR,#MSG_RMENU3
042C 752500  MOV POS_H,#00H
042F 7524A0  MOV POS_L,#0A0H      ;SHOW AT LINE 2
0432 1207E5  LCALL DSP_MSG
0435 22      RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.....
:ROUTINE FOR DISPLAY RATE OF CHARGE
.....
0438      DISPRATE:
0438 22      RET
.....
:ROUTINE FOR CONFIRM MESSAGE
.....
D437      SHW_CONFIRM:
0437 12058A  LCALL RAMCLR
043A 12100A  LCALL CLRSCRN
043D 1210AC  LCALL CURHOME

0440 752000      MOV CURROW,#00      ;SHOW AT LINE 0
0443 752100      MOV CURCOL,#00
0446 1210C0      LCALL CURCONV
0449 900B84      MOV DPTR,#MSG_CONFIRM1
044C 752500      MOV POS_H,#00H
044F 752400      MOV POS_L,#00H      ;SHOW AT LINE 0 PAGE 0
0452 1207E5      LCALL DSP_MSG

0455 752001      MOV CURROW,#01
0458 752100      MOV CURCOL,#00
045B 1210C0      LCALL CURCONV
045E 900B89      MOV DPTR,#MSG_CONFIRM2
0461 752500      MOV POS_H,#00H
0464 752450      MOV POS_L,#50H      ;SHOW AT LINE 1
0467 1207E5      LCALL DSP_MSG

046A 752002      MOV CURROW,#02
046D 752100      MOV CURCOL,#00
0470 1210C0      LCALL CURCONV
0473 900BAD      MOV DPTR,#MSG_CONFIRM3
0476 752500      MOV POS_H,#00H
0479 7524A0      MOV POS_L,#0A0H      ;SHOW AT LINE 1
047C 1207E5      LCALL DSP_MSG

047F 22      RET
.....
:ROUTINE FOR PROGRAM SMART CARD
.....
0480      PROGCARD:
0480 120720      LCALL SHOWEDIT      ;SHOW MESSAGE
0483 751800      MOV BUF_VAL1,#00
0486 751900      MOV BUF_VAL2,#00
0489 751A00      MOV BUF_VAL3,#00      ;CLEAR ALL BUFFER
048C 752001      MOV CURROW,#01
048F 752114      MOV CURCOL,#20      ;START AT COLUMN 20
0492 1210C0      LCALL CURCONV
0495 1207D0      LCALL CURON
0498 752001      MOV CURROW,#01
049B 752114      MOV CURCOL,#20      ;START AT COLUMN 20
049E 1210C0      LCALL CURCONV
04A1 751B00      MOV INDEXPOS,#00

04A4      LPPROGCARD:
04A4 1200A1      LCALL WAITKEY
04A7 B40B03      CJNE A,#11,NXTPROGCARD1
04AA 020541      LJMPPROGCARD      ;CANCEL OPERATION

04AD      NXTPROGCARD1:
04AD B40A03      CJNE A,#10,NXTPROGCARD11
04B0 0204E7      LJMPPROGCARD3

04B3      NXTPROGCARD11:
04B3 B40C06      CJNE A,#12,NXTPROGCARD2
04B6 120674      LCALL CLRVAL
04B9 020460      LJMPPROGCARD

04BC      NXTPROGCARD2:
04BC C3          CLR C
04BD 940A      SUBB A,#10      ;CHECK KEY OVER NINE
04BF 50E3      JNC LPPROGCARD ;IF OVER
04C1 240A      ADD A,#10
04C3 4430      ORL A,#30H      ;CONVERT TO ASCII
04C5 F9      MOV R1,A      ;TEMP STORE
04C8 7818      MOV R0,#BUF_VAL1

04C8 E51B      MOV A,INDEXPOS      ;FIND ADDRESS TO KEEP DATA
04CA 28      ADD A,R0
04CB F8      MOV R0,A
04CC E9      MOV A,R1      ;RESTORE DATA
04CD F6      MOV @R0,A      ;SAVE IN BUFFER
04CE 1210FF      LCALL WRCHR      ;WRITE DATA TO LCD MODULE
04D1 051B      INC INDEXPOS
04D3 E51B      MOV A,INDEXPOS
04D5 B403CC      CJNE A,#03,LPPROGCARD
04D8 751B00      MOV INDEXPOS,#00      ;RESTART FIRST POSITION
04DB 752001      MOV CURROW,#01
04DE 752114      MOV CURCOL,#20      ;START AT COLUMN 20
04E1 1210C0      LCALL CURCONV
04E4 0204A4      LJMPPROGCARD
04E7      NXTPROGCARD3:
04E7 1207D9      LCALL CUROFF
04EA 12060C      LCALL SHOWWAIT
04ED 120680      LCALL ADDWITHOLD      ;ADD NEW WITH OLD DATA
04F0 120F2C      LCALL EWENEEP
04F3 7403      MOV A,#UNITVAL
04F5 A91A      MOV R1,BUF_VAL3      ;WRITE RESULT
04F7 120EDC      LCALL WRITEEEP
04FA 7403      MOV A,#UNITVAL
04FC 120ECA      LCALL READEEP
04FF B51A27      CJNE A,BUF_VAL3,PROGERR ;CHECK ERROR
0502 7404      MOV A,#TENVAL
0504 A919      MOV R1,BUF_VAL2
0506 120EDC      LCALL WRITEEEP
0509 7404      MOV A,#TENVAL
050B 120ECA      LCALL READEEP
050E B51918      CJNE A,BUF_VAL2,PROGERR
0511 7405      MOV A,#HUNDVAL
0513 A918      MOV R1,BUF_VAL1
0515 120EDC      LCALL WRITEEEP
0518 7405      MOV A,#HUNDVAL
051A 120ECA      LCALL READEEP
051D B51809      CJNE A,BUF_VAL1,PROGERR
0520 1207DA      LCALL SHOWCOMPL
0523 1207C7      LCALL DELAYP
0526 020480      LJMPPROGCARD

0529      PROGERR:
0529 752002      MOV CURROW,#02
052C 752100      MOV CURCOL,#00
052F 1210C0      LCALL CURCONV
0532 900A70      MOV DPTR,#MSG_ERR      ;SHOW MESSAGE
0535 752500      MOV POS_H,#00H
0538 7524A0      MOV POS_L,#0A0H      ;SHOW AT LINE 1
053B 1207E5      LCALL DSP_MSG
053E 020480      LJMPPROGCARD

0541      QPROGCARD:
0541 22      RET
.....
:ROUTINE FOR CHECKING DATA OF BUS USAGE
.....
0542      CHECKDATA:
0542 12100A      LCALL CLRSCRN
0545 22      RET
.....
:ROUTINE FOR SHOW LOGO
.....
0546      SHW_LOGO:
0546 12058A      LCALL RAMCLR
0549 12100A      LCALL CLRSCRN
054C 1210AC      LCALL CURHOME
054F D270      SETB FGPAGE0      ;DISPLAY PAGE 0 AND STORE PAGE 1

0551 752000      MOV CURROW,#00      ;SHOW AT LINE 0
0554 752100      MOV CURCOL,#00
0557 1210C0      LCALL CURCONV
055A 900B3D      MOV DPTR,#MSG_LOGO1
055D 752500      MOV POS_H,#00H
0560 752400      MOV POS_L,#00H      ;SHOW AT LINE 0 PAGE 0
0563 1207E5      LCALL DSP_MSG

0566 752001      MOV CURROW,#01

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0569 752100    MOV  CURCOL,#00
056C 1210C0    LCALL CURCONV
056F 90085D    MOV  DPTR,#MSG_LOGO2
0572 752500    MOV  POS_H,#00H
0575 752450    MOV  POS_L,#50H    ;SHOW AT LINE 1
0578 1207E5    LCALL DSP_MSG

057B 752002    MOV  CURROW,#02
057E 752100    MOV  CURCOL,#00
0581 1210C0    LCALL CURCONV
0584 90087D    MOV  DPTR,#MSG_LOGO3
0587 752500    MOV  POS_H,#00H
058A 7524A0    MOV  POS_L,#0A0H    ;SHOW AT LINE 2
058D 1207E5    LCALL DSP_MSG
0590 22    RET

;-----
;BEEP PIEZZO WHILE KEYPRESSED
;GENERATE SQUARE WAVE BY TIMER0 = 500HZ
;-----
0591    BEEP:
0591 22    RET

;-----
;ROUTINE SEND BYT OF DATA THROUGH SERIAL PORT
;-----
0592    SENDBYTE:
0592 3099FD    JNB  TI,$
0595 C299    CLR  TI
0597 F599    MOV  SBUF,A
0599 22    RET

;-----
;CLEAR DISPLAY BUFFER USE EXTERNAL RAM ADDRESS 0000H -
01FFH
;WE SEPARATE BUFFER IN 6 BLOCKS FOLLOW.
; BLOCK 1 USE ADDRESS 0000H - 004FH BUFFER FOR LINE 0
; BLOCK 2 USE ADDRESS 0050H - 009FH BUFFER FOR LINE 1
; BLOCK 3 USE ADDRESS 00A0H - 00FFH BUFFER FOR LINE 2
;
059A    RAMCLR:
059A 900000    MOV  DPTR,#0000H    ;DPTR POINT TO START ADDRESS
059D    RAMCLR1:
059D 7400    MOV  A,#00    ;PUT 00 IN THERE
059F F0 MOVX  @DPTR,A    ;NEXT
05A0 A3 INC  DPTR
05A1 E583    MOV  A,DPH
05A3 B401F7  CJNE A,#01H,RAMCLR1 ;TILL END OF BUFFER
05A6 22    RET

;-----
;ROUTINE ADJUST BCD TO ASCII
;INPUT - ACC
;OUTPUT - R3 (HIGH DIGIT)
;        - R2 (LOW DIGIT)
;-----
05A7    BCDASC:
05A7 FB    MOV  R3,A
05A8 540F    ANL  A,#0FH    ;ADJUST LOWER DIGIT
05AA 4430    ORL  A,#30H
05AC FA    MOV  R2,A    ;SAVE LOWER DIGIT
05AD EB    MOV  A,R3
05AE C4    SWAP A    ;ADJUST HIGHER DIGIT
05AF 540F    ANL  A,#0FH
05B1 4430    ORL  A,#30H
05B3 FB    MOV  R3,A    ;SAVE HIGHER DIGIT
05B4 22    RET

;-----
;ROUTINE DISPLAY BCD 2 DIGIT
;-----
05B5    DISPBCD2:
05B5 1205A7  LCALL BCDASC
05B8 EB    MOV  A,R3
05B9 1210FF  LCALL WRCHR
05BC EA    MOV  A,R2
05BD 1210FF  LCALL WRCHR
05C0 22    RET

;-----
;ROUTINE SHOW DATE IN BUDDHA SAKKARAJ
;-----
05C1    SHOWDATE:
05C1 E512    MOV  A,DATE
05C3 1205B5  LCALL DISPBCD2
05C6 742F    MOV  A,#7
05C8 1210FF  LCALL WRCHR
05CB E513    MOV  A,MONTH
05CD 1205B5  LCALL DISPBCD2
05D0 742F    MOV  A,#7
05D2 1210FF  LCALL WRCHR
05D5 7425    MOV  A,#25H
05D7 1205B5  LCALL DISPBCD2
05DA E514    MOV  A,YEAR
05DC 1205F3  LCALL ADJBUD
05DF 1205B5  LCALL DISPBCD2
05E2 22    RET

;-----
;ROUTINE SHOW TIME BOTH HOUR AND MINUTE NOT SECOND
;-----
05E3    SHOWTIME:
05E3 E511    MOV  A,HOUR
05E5 1205B5  LCALL DISPBCD2
05E8 743A    MOV  A,#7
05EA 1210FF  LCALL WRCHR
05ED E510    MOV  A,MINUTE
05EF 1205B5  LCALL DISPBCD2
05F2 22    RET

;-----
;ROUTINE ADJUST CHRIST SAKKARAJ TO BUDDHA SAKKARAJ
; BY ADD WITH 43
;-----
05F3    ADJBUD
05F3 2443    ADD  A,#43H
05F5 D4    DAA
05F6 22    RET

;-----
;ROUTINE READ POSITION
;OUTPUT: DPTR
;REGISTER USE: R0,R1
;-----
05F7    RDPOST:
05F7 75F002  MOV  B,#2
05FA A4    MUL  AB    ;FIND ABSOLUTE POSITION
05FB F8    MOV  R0,A    ;TEMPORARY STORE
05FC 900C49  MOV  DPTR,#ROUTETABF
05FF 93    MOVC A,@A+DPTR
0600 F9    MOV  R1,A    ;SAVE HIGH BYTE ADDRESS
0601 A3    INC  DPTR
0602 E8    MOV  A,R0
0603 93    MOVC A,@A+DPTR
0604 F582    MOV  DPL,A    ;LOAD TO DPTR
0606 B983    MOV  DPH,R1
0608 22    RET

;-----
;ROUTINE SCAN KEYBOARD
;OUTPUT :CARRY SET - ACC BE FILLED BY KEY'S DATA
;        : CARRY CLEAR - NOT KEY PRESS
;-----
0609    SCANKEY:
0609 120633  LCALL CHKPRESS
060C 601A    JZ   NOPRESS
060E F505    MOV  05H,A    ;STORE COLUMN DATA
0610 12062A  LCALL DEBOUNCE ;DELAY TO ELIMINATE BOUNCE
0613 120633  LCALL CHKPRESS
0616 B5050F  CJNE A,05H,NOPRESS ;IF BOUNCING ERROR

0619 12065E  LCALL CHKRELEASE ;CHECK RELEASE KEY PRESS
061C 1206AA  LCALL FINDROW    ;FIND NUMBER OF ROW AND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COLUMN
061F 120654      LCALL  FINDCOL
0622 120669      LCALL  DEFKEY      ;DEFINE INDIVIDUAL KEY
0625 D3          SETB  C
0626 8001        SJMP  QSCANKEY
0628             NOPRESS:
0628 C3          CLR   C
0629             QSCANKEY:
0629 22          RET

;-----
;ROUTINE DELAY FOR DEBOUNCING ERROR
;REGISTER USE : R0,R1
;-----

062A             DEBOUNCE:
062A 7805        MOV   R0,#5      ;DELAY ABOUT 20MS
062C 7900        MOV   R1,#00
062E             LPDEBOUNCE:
062E D9FE        DJNZ  R1,$
0630 D8FC        DJNZ  R0,LPDEBOUNCE
0632 22          RET

;-----
;ROUTINE CHECK KEY PRESSING
;OUTPUT : ZERO FLAG SET - NOT PRESS
;          ZERO FLAG CLEAR - ACC (RAW DATA) R4 (ROW DATA)
;-----

0633             CHKPRESS:
0633 908002      MOV   DPTR,#PPI2C
0636 7CFE        MOV   R4,#11111110B ;ROW DATA
0638             LPCHKPRESS:
0638 EC          MOV   A,R4
0639 F0          MOVX  @DPTRA ;SEND 0 TO ONE ROW
063A E0          MOVX  A,@DPTR ;READ COLUMN DATA
063B C4          SWAP  A ;CONVERT HIGH BETWEEN LOW NIBBLE
063C 44F0        ORL   A,#0F0H ;SET HIGH NIBBLE TO HIGH
063E F4          CPL   A ;COMPLEMENT ACCUMULATOR
063F 7008        JNZ   QCHKPRESS ;HAVE ANY KEY PRESSED
0641 EC          MOV   A,R4 ;RELOAD
0642 23          RL   A
0643 FC          MOV   R4,A ;STORE NEW VALUE
0644 20E4F1      JB    A OR 4,LPCHKPRESS
0647 5400        ANL   A,#00 ;SET ZERO FLAG
0649             QCHKPRESS:
0649 22          RET

;-----
;ROUTINE FIND ROW THAT KEY BE PRESSED
;INPUT : R4 (ROW DATA)
;OUTPUT : R4 (NUMBER OF ROW)
;REGISTER USE : R2,R4
;-----

064A             FINDROW:
064A 7A00        MOV   R2,#00 ;START ROW 00
064C EC          MOV   A,R4
064D             LPFINDROW:
064D 13          RRC   A
064E 5003        JNC   QFINDROW ;IF CARRY CLEAR
0650 0A          INC   R2
0651 80FA        SJMP  LPFINDROW
0653             QFINDROW:
0653 22          RET

;-----
;ROUTINE FIND COLUMN THAT KEY BE PRESSED
;INPUT : R5 (COLUMN DATA)
;OUTPUT : R5 (NUMBER OF COLUMN)
;REGISTER USE: R3,R5
;-----

0654             FINDCOL:
0654 7800        MOV   R3,#00 ;START COLUMN 00
0658 ED          MOV   A,R5
0657             LPFINDCOL:
0657 13          RRC   A
0658 4003        JC    QFINDCOL ;IF CARRY CLEAR
065A 08          INC   R3
065B 80FA        SJMP  LPFINDCOL
065D             QFINDCOL:
065D 7800        MOV   R3,#00 ;START COLUMN 00
0658 ED          MOV   A,R5
0657             LPFINDCOL:
0657 13          RRC   A
0658 4003        JC    QFINDCOL ;IF CARRY CLEAR
065A 08          INC   R3
065B 80FA        SJMP  LPFINDCOL
065D             QFINDCOL:

065E             CHKRELEASE:
065E 909002      MOV   DPTR,#PPI2C
0661 E0          MOVX  A,@DPTR ;READ STATUS KEY
0662 C4          SWAP  A
0663 44F0        ORL   A,#0F0H
0665 F4          CPL   A ;COMPLEMENT ACCUMULATOR
0666 70F8        JNZ   CHKRELEASE ;IF KEYPRESS
0668 22          RET

;-----
;ROUTINE DEFINE MEANING OF KEY POSITION
;INPUT : R2,R3
;OUTPUT : ACC (DATA)
;-----

0669             DEFKEY:
0669 90862D      MOV   DPTR,#KEYTAB
066C EA          MOV   A,R2 ;FIND ABSOLUTE POSITION OF THAT
ROW
066D 75F004      MOV   B,#4
0670 A4          MUL   AB ;FIRST POSITION
0671 28          ADD   A,R3 ;ADD COLUMN
0672 93          MOVC  A,@A+DPTR ;READ CODE
0673 22          RET

;-----
;ROUTINE CLEAR VALUE IN CARD
;-----

0674             CLRVAL:
0674 120F2C      LCALL  EWENEEP
0677 7403        MOV   A,#UNITVAL
0679 7930        MOV   R1,#0
067B 120EDC      LCALL  WRITEEP
067E 7404        MOV   A,#TENVAL
0680 7930        MOV   R1,#0
0682 120EDC      LCALL  WRITEEP
0685 7405        MOV   A,#HUNDVAL
0687 7930        MOV   R1,#0
0689 120EDC      LCALL  WRITEEP
068C 22          RET

;-----
;ROUTINE ADD NEW VALUE WITH OLD VALUE
;-----

068D             ADDWITHOLD:
068D 7403        MOV   A,#UNITVAL ;LOAD OLD IN UNIT VALUE
068F 120ECA      LCALL  READEEP
0692 540F        ANL   A,#0FH
0694 FB          MOV   R3,A
0695 E51A        MOV   A,BUF_VAL3 ;ADD UNIT DIGIT
0697 540F        ANL   A,#0FH
0699 C3          CLR   C
069A 1206CC      LCALL  ADDDGT
069D 9271        MOV   FGCAR,C ;STORE CARRY FLAG
069F F51A        MOV   BUF_VAL3,A
;ADD TENTH DIGIT

06A1 7404        MOV   A,#TENVAL ;LOAD OLD IN TEN VALUE
06A3 120ECA      LCALL  READEEP
06A8 540F        ANL   A,#0FH
06AB FB          MOV   R3,A
06AD E519        MOV   A,BUF_VAL2
06AB 540F        ANL   A,#0FH
06AD A271        MOV   C,FGCAR ;RESTORE
06AF 1206CC      LCALL  ADDDGT
06B2 9271        MOV   FGCAR,C
06B4 F519        MOV   BUF_VAL2,A
;ADD HUNDRETH DIGIT

06B6 7405        MOV   A,#HUNDVAL ;LOAD OLD IN HUNDRED VALUE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

06B8 120ECA      LCALL READEEP
06BB 540F      ANL A,#0FH
06BD FB        MOV R3,A
06BE E518      MOV A,BUF_VAL1
06C0 540F      ANL A,#0FH
06C2 A271      MOV C,FGCAR
06C4 1206CC     LCALL ADDDGT
06C7 9271      MOV FGCAR,C
06C9 F518      MOV BUF_VAL1,A
06CB 22        RET

;.....
;ROUTINE ADD DIGIT
;INPUT: R3,A
;OUTPUT: A,CARRY
;.....
06CC          ADDDGT:
06CC 3B        ADDC A,R3
06CD D4        DAA
06CE FC        MOV R4,A      ;TEMP STORE
06CF 4430      ORL A,#30H    ;CONVERT TO ASCII
06D1 FD        MOV R5,A
06D2 EC        MOV A,R4
06D3 20E403    JB A OR 4,ADDDGT1 ;CHECK AUXILIARY FLAG SET OR
NOT
06D6 C3        CLR C
06D7 8001      SJMP QADDDGT
06D9          ADDDGT1:
06D9 D3        SETB C
06DA          QADDDGT:
06DA ED        MOV A,R5      ;LOAD TO ACCUMULATOR
06DB 22        RET

;.....
;ROUTINE FOR SHOW MESSAGE WAIT
;.....
06DC          SHOWWAIT:
06DC 12100A    LCALL CLRSCRN
06DF 752000    MOV CURROW,#00
06E2 752100    MOV CURCOL,#00
06E5 1210C0    LCALL CURCONV
06E8 900A07    MOV DPTR,#MSG_WAIT1 ;SHOW MESSAGE
06EB 752500    MOV POS_H,#00H
06EE 752400    MOV POS_L,#00H    ;SHOW AT LINE 0
06F1 1207E5    LCALL DSP_MSG
06F4 752001    MOV CURROW,#01
06F7 752100    MOV CURCOL,#00
06FA 1210C0    LCALL CURCONV
06FD 900A2A    MOV DPTR,#MSG_WAIT2 ;SHOW MESSAGE
0700 752500    MOV POS_H,#00H
0703 752450    MOV POS_L,#50H    ;SHOW AT LINE 1
0706 1207E5    LCALL DSP_MSG
0709 22        RET

;.....
;ROUTINE FOR SHOW MESSAGE COMPLETE
;.....
070A          SHOWCOMPL:
070A 752002    MOV CURROW,#02
070D 752100    MOV CURCOL,#00
0710 1210C0    LCALL CURCONV
0713 900A4D    MOV DPTR,#MSG_COMPL ;SHOW MESSAGE
0716 752500    MOV POS_H,#00H
0719 7524A0    MOV POS_L,#0A0H   ;SHOW AT LINE 2
071C 1207E5    LCALL DSP_MSG
071F 22        RET

;.....
;ROUTINE FOR SHOW MESSAGE AND VALUE OF SMART CARD
;.....
0720          SHOWEDIT:
0720 12100A    LCALL CLRSCRN
0723 752000    MOV CURROW,#00
0726 752100    MOV CURCOL,#00
0729 1210C0    LCALL CURCONV
072C 9009B8    MOV DPTR,#MSG_EDIT1 ;SHOW MESSAGE
072F 752500    MOV POS_H,#00H
0732 752400    MOV POS_L,#0A0H   ;SHOW AT LINE 2
0735 1207E5    LCALL DSP_MSG
0738 752001    MOV CURROW,#01
073B 752100    MOV CURCOL,#00
073E 1210C0    LCALL CURCONV
0741 9009CF    MOV DPTR,#MSG_EDIT2 ;SHOW MESSAGE
0744 752500    MOV POS_H,#00H
0747 752450    MOV POS_L,#50H    ;SHOW AT LINE 1
074A 1207E5    LCALL DSP_MSG
074D 752002    MOV CURROW,#02
0750 752100    MOV CURCOL,#00
0753 1210C0    LCALL CURCONV
0756 9009E8    MOV DPTR,#MSG_EDIT3 ;SHOW MESSAGE
0759 752500    MOV POS_H,#00H
075C 7524A0    MOV POS_L,#0A0H   ;SHOW AT LINE 2
075F 1207E5    LCALL DSP_MSG

0762 7400      MOV A,#00
0764 120ECA    LCALL READEEP
0767 B4AA03    CJNE A,#0AAH,SHOWEDIT1 ;IF NOT EVER BE PRO-
GRAMMED
076A 02078C    LJMPL SHOWEDIT2
076D          SHOWEDIT1:
076D 120F2C    LCALL EWENEEP
0770 7400      MOV A,#CRDVALID ;ENABLE CARD
0772 79AA      MOV R1,#0AAH
0774 120EDC    LCALL WRITEEEP
0777 7403      MOV A,#UNITVAL
0779 7930      MOV R1,#0
077B 120EDC    LCALL WRITEEEP ;WRITE VALUE TO ZERO
077E 7404      MOV A,#TENVAL
0780 7930      MOV R1,#0
0782 120EDC    LCALL WRITEEEP
0785 7405      MOV A,#HUNDVAL
0787 7930      MOV R1,#0
0789 120EDC    LCALL WRITEEEP
;.....
078C          SHOWEDIT2:
078C 1207A5    LCALL RDWRVAL ;READ DATA FROM SMART CARD
078F 752000    MOV CURROW,#00 ;SHOW AT LINE 0
0792 752118    MOV CURCOL,#24
0795 1210C0    LCALL CURCONV
0798 900DEB    MOV DPTR,#MSG_BAHT
079B 752500    MOV POS_H,#00H
079E 752400    MOV POS_L,#00H    ;SHOW AT LINE 0 PAGE 0
07A1 1207E5    LCALL DSP_MSG
07A4 22        RET

;.....
;ROUTINE FOR OLD VALUE IN THE CARD
;.....
07A5          RDWRVAL:
07A5 752000    MOV CURROW,#00
07A8 752114    MOV CURCOL,#20 ;START AT COLUMN 20
07AB 1210C0    LCALL CURCONV
07AE 7405      MOV A,#HUNDVAL
07B0 120ECA    LCALL READEEP
07B3 1210FF    LCALL WRCHR
07B6 7404      MOV A,#TENVAL
07B8 120ECA    LCALL READEEP
07BB 1210FF    LCALL WRCHR
07BE 7403      MOV A,#UNITVAL
07C0 120ECA    LCALL READEEP
07C3 1210FF    LCALL WRCHR
07C6 22        RET

;.....
;ROUTINE DELAY TIME FOR WAIT STRING
;.....
07C7          DELAY:
07C7 7830      MOV R0,#30H
07C9 7900      MOV R1,#00H
07CB          LPDELAY:
07CB D8FE      DJNZ R0,$
07CD D8FC      DJNZ R1,LPDELAY
07CF 22        RET

0770 7400      MOV A,#CRDVALID ;ENABLE CARD
0772 79AA      MOV R1,#0AAH
0774 120EDC    LCALL WRITEEEP
0777 7403      MOV A,#UNITVAL
0779 7930      MOV R1,#0
077B 120EDC    LCALL WRITEEEP ;WRITE VALUE TO ZERO
077E 7404      MOV A,#TENVAL
0780 7930      MOV R1,#0
0782 120EDC    LCALL WRITEEEP
0785 7405      MOV A,#HUNDVAL
0787 7930      MOV R1,#0
0789 120EDC    LCALL WRITEEEP
;.....
078C          SHOWEDIT2:
078C 1207A5    LCALL RDWRVAL ;READ DATA FROM SMART CARD
078F 752000    MOV CURROW,#00 ;SHOW AT LINE 0
0792 752118    MOV CURCOL,#24
0795 1210C0    LCALL CURCONV
0798 900DEB    MOV DPTR,#MSG_BAHT
079B 752500    MOV POS_H,#00H
079E 752400    MOV POS_L,#00H    ;SHOW AT LINE 0 PAGE 0
07A1 1207E5    LCALL DSP_MSG
07A4 22        RET

;.....
;ROUTINE FOR OLD VALUE IN THE CARD
;.....
07A5          RDWRVAL:
07A5 752000    MOV CURROW,#00
07A8 752114    MOV CURCOL,#20 ;START AT COLUMN 20
07AB 1210C0    LCALL CURCONV
07AE 7405      MOV A,#HUNDVAL
07B0 120ECA    LCALL READEEP
07B3 1210FF    LCALL WRCHR
07B6 7404      MOV A,#TENVAL
07B8 120ECA    LCALL READEEP
07BB 1210FF    LCALL WRCHR
07BE 7403      MOV A,#UNITVAL
07C0 120ECA    LCALL READEEP
07C3 1210FF    LCALL WRCHR
07C6 22        RET

;.....
;ROUTINE DELAY TIME FOR WAIT STRING
;.....
07C7          DELAY:
07C7 7830      MOV R0,#30H
07C9 7900      MOV R1,#00H
07CB          LPDELAY:
07CB D8FE      DJNZ R0,$
07CD D8FC      DJNZ R1,LPDELAY
07CF 22        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;.....
;ROUTINE FOR VISIBLE CURSOR
;.....
07D0 1210C0 CURON: LCALL CURCONV
07D3 745F MOV A,#5FH
07D5 121335 LCALL WRFMRBUF
07D8 22RET

;.....
;ROUTINE FOR INVISIBLE CURSOR
;.....
07D9 7821 CUROFF: MOV R0,#CURCOL
07DB 16DEC @R0
07DC 1210C0 LCALL CURCONV
07DF 7420 MOV A,#20H
07E1 121335 LCALL WRFMRBUF
;07E4 22 RET

;.....
;ROUTINE DISPLAY MESSAGE
;INPUT DPTR = POINTER START CHARACTER
; POS_L = START ADDRESS OF EXTERNAL RAM
;.....
07E5 DSP_MSG:
07E5 AA25 MOV R2,POS_H
07E7 A924 MOV R1,POS_L
07E9 7800 MOV R0,#00
;FROM NOW,IT MUST BE DATA TO DISPLAY
07EB LPDSP_MSG.
07EB E8MOV A,R0
07EC 93MOVC A,@A+DPTR ;READ FROM TABLE
07ED C083 PUSH DPH ;SAVE DPTR
07EF C082 PUSH DPL
07F1 8A83 MOV DPH,R2 ;POSITION OF DISPLAY RAM
07F3 8982 MOV DPL,R1
07F5 F0 MOVX @DPTR,A ;SAVE DATA IN BUFFER DISPLAY
07F6 08 INC R0
07F7 09 INC R1
07F8 D082 POP DPL ;RESTORE DPTR
07FA D083 POP DPH
07FC B404EC CJNE A,#04,LPDSP_MSG ;CHECK CODE 04 END OF TEXT
07FF 307009 JNB FGPAE0,QDSP_MSG
0802 852583 MOV DPH,POS_H ;START ADDRESS OF DISPLAY BUFFER
0805 852482 MOV DPL,POS_L
0808 121266 LCALL WRTDAT
080B QDSP_MSG:
080B 22 RET ;IF NULL,GO BACK

;.....
;SERVICE ROUTINE CHECK DISTANCE BY COUNT FROM INTERNAL
;COUNTER
;.....
080C CHKDIST:
080C C2AF CLR EA ;DISABLE INTERRUPT
080E C0E0 PUSH A
0810 C083 PUSH DPH
0812 C082 PUSH DPL ;PUSH SPECIAL REGISTER TO STACK
0814 C0F0 PUSH B
0818 C28C CLR TR0 ;STOP COUNTER
0818 758CFE MOV TH0,#HIGH[KILOP] ;RELOAD
0818 758AB6 MOV TL0,#LOW[KILOP]
081E D28C SETB TR0 ;START COUNTER
0820 051C INC BUF_DIST
;
; MOV A,#55H
; MOV DPTR,#BACKUP
; MOVX @DPTR,A ;WRITE BACK UP CODE
; INC DPTR
; MOV A,BUF_DIST ;LOAD DISTANCE
; MOVX @DPTR,A ;WRITE IN RAM
; MOV B,#2
; DIV AB
; MOV BUF_CURP,A ;CURRENT RANGE
; INC DPTR
; MOVX @DPTR,A ;WRITE IN BACK RAM

0822 D2AF SETB EA
0824 D0F0 POP B
0828 D082 POP DPL

```

```

0828 D083 POP DPH
082A D0E0 POP A
082C 32 RETI

082D 0001020304KEYTAB: DFB 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

083D 202020C3D0MSG_LOGO1: DFB " ระบบจัดการมรภประจำทาง ",04
085D 2020202020MSG_LOGO2: DFB " โดย ",04
087D 202020B9D2MSG_LOGO3: DFB " นาย วรทล ธิลาเกียรติสกุล ",04

089D 203120A1D2MSG_MAIN1: DFB " 1 การทำงานปกติ ",04
08B7 203220E2BBMSG_MAIN2: DFB " 2 ไปรณกรรมของคำขอมัติใหม่ ",04
08D5 203320B5C3MSG_MAIN3: DFB " 3 ตรวจสอบข้อมูลการใช้บริการ ",04

08F3 2020202020MSG_ROUTE1: DFB " รถประจำทางสาย ปล.3 ",04
0911 20202020E0MSG_ROUTE2: DFB " เส้นทางขนส่งสายใต้ - รังสิต ",04
0934 20202020C3MSG_ROUTE3: DFB " ระยะทางประมาณ 34 กิโลเมตร ",04

0953 202020E2BBMSG_RMENU1: DFB " โปรดเลือกเส้นทางการเดินทาง ",04
0972 203120C3D1MSG_RMENU2: DFB " 1 รังสิต ไปบึง ชกานี่ขนส่งสายใต้ ",04
0995 203220CAB6MSG_RMENU3: DFB " 2 สถานีขนส่งสายใต้ ไปบึง รังสิต ",04

09B8 AB03B9C7B9MSG_EDIT1: DFB "จำนวนเงินคงเหลือในบัตร",04
09CF B5E9CDA7A1MSG_EDIT2: DFB "ต้องการเพิ่มจำนวนเงิน ",04
09E6 A1B4BD0E8MSG_EDIT3: DFB "กดปุ่ม 'ตกลง' เมื่อเสร็จสมบูรณ์ ",04

0A07 2020202020MSG_WAIT1: DFB " กำลังไปรณกรรมคำขอมัติใหม่ ",04
0A2A 20202020E2MSG_WAIT2: DFB " .....ไปตรวจสักครู่. ",04
0A4D 2020202020MSG_COMPL: DFB " การทำงานเสร็จสมบูรณ์ ",04
0A70 2020202020MSG_ERR: DFB " การทำงานผิดพลาด ",04

0A93 20E2BBC3B4MSG_CHOICE1: DFB " โปรดเลือกเส้นทางการเดินทางสาย ปล 3 ",04
0AB8 20312920C3MSG_CHOICE2: DFB " 1) รังสิต ไปบึง ชนส่งสายใต้ ",04
0AD8 20322920A2MSG_CHOICE3: DFB " 2) ชนส่งสายใต้ ไปบึง รังสิต ",04

0AF8 20C7D1B920MSG_DATE: DFB " วัน ",04
0B02 20E0C7C5D2MSG_TIME: DFB " เวลา ",04
0B0B 20C2D7B9C2MSG_SETTm: DFB " มีมันกด 'ตกลง' แก้ไขกด 'แก้ไข' ",04

0B2D 20C7D1B920MSG_DATEF: DFB " วัน > 00/00/2500 ",04
0B43 20E0C7C5D2MSG_TIMEF: DFB " เวลา > 00:00 ",04

0B52 2020202020MSG_DISTP: DFB " กิโลเมตรที่ ",04

0B64 C2D7B9C2D1MSG_CONFIRM1: DFB "ยืนยันการเลือกถูกต้อง กดปุ่ม 'ตกลง' ",04
0B89 B5E9CDA7A1MSG_CONFIRM2: DFB "ต้องการแก้ไขทางเลือก กดปุ่ม 'แก้ไข' ",04
0BAD CAD4E9B9CAMS_MSG_CONFIRM3: DFB "สิ้นสุดการทำงาน กดปุ่ม 'จบ' ",04

0BCB 202020B5D3MSG_UP1: DFB " ตำแหน่งระยะที่ขึ้น ",04
0BE3 202020A8D3MSG_UP2: DFB " จำนวนเงินที่เหลือ ",04

0BF9 202020B5D3MSG_DOWN1: DFB " ตำแหน่งระยะที่ลง ",04
0C10 202020A4E8MSG_DOWN2: DFB " ค่าโดยสาร ",04
0C1E 202020A8D3MSG_DOWN3: DFB " จำนวนเงินที่เหลือ ",04

0C34 0102040507INDEXF: DFB 1,2,4,5,7
0C39 0A0F101114 DFB 10,15,16,17,20
0C3E 1516171819 DFB 21,22,23,24,25
0C43 1A1B1C2021 DFB 26,27,28,32,33,34

0C49 0CBF0CC60CROUTETABF: DWM
MSG_RUNG,MSG_BANG,MSG BUMIPOL,MSG SAPAN,MSG_LAKSI
0C53 0CF30CF0D DWM
MSG_KASET,MSG_LADPRAO,MSG_MOHCHIT,MSG_SBUFF,MSG_ANUS
0C5D 0D3E0D46D DWM
MSG_PAYA,MSG_BRIAYUT,MSG_HIGHWAY,MSG_WATBEN,MSG_PARUP
0C67 0D7B0D910D DWM
MSG_KASUANG,MSG_PANFA,MSG_PACHA,MSG_PAPIN,MSG_BAROM,MSG_SAITAI

0C73 0102030708INDEXR: DFB 1,2,3,7,8,9
0C79 0A0B0C0D0E DFB 10,11,12,13,14
0C7E 0F12131419 DFB 15,18,19,20,25
0C83 1C1E1F2122 DFB 28,30,31,33,34

0C88 0DD60DC60DROUTETABR: DWM
MSG_SAITAI,MSG_BAROM,MSG_PAPIN,MSG_PACHA,MSG_PANFA,MSG_KASUANG
0C94 0D6B0D5D0D DWM
MSG_PARUP,MSG_WATBEN,MSG_HIGHWAY,MSG_SRIAYUT,MSG_PAYA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0C9E 0D290D1F0D      DWM
MSG_ANUS,MSG_SBUFF,MSG_MOHCHIT,MSG_LADPRAO,MSG_KASET
0CA8 0CE10C070C      DWM
MSG_LAKSI,MSG_SAPAN,MSG_BUMIPOL,MSG_BANG,MSG_RUNG

```

```

0CB2 0102030405ZONETABF:  DFB 1,2,3,4,5,6,7
0CB9 06080A0C0E0ZERATE:  DFB 6,8,10,12,14,16

```

```

0CBF C3D1A7CAD4MSG_RUNG:  DFB "วังจันทน์",04
0CC8 BAD2A7BED9MSG_BANG:  DFB "บางขุน",04
0CCD C38E2EC0D9MSG_BUMIPOL:  DFB "พ.ท.ภูมิพล",04
0CD7 CAD08ED289MSG_SAPAN:  DFB "สะพานใหม่",04
0CE1 CDB8D8CAD2MSG_LAKSI:  DFB "อนุสาวรีย์ท้าวสุทไธสง",04
0CF3 CAD2C1E1C2MSG_KASET:  DFB "ตามแยกถนนศรี",04
0CFB 8B02A1B7D2MSG_LADPRAO:  DFB "ปากทางภาคท้าว",04
0D0E CAB6D2B9D5MSG_MOHCHIT:  DFB "สถานีขนส่งหมอชิต",04
0D1F CAD08ED289MSG_SBUFF:  DFB "สะพานขาว",04
0D29 CDB9D8CAD2MSG_ANUS:  DFB "อนุสาวรีย์ท้าวสุทไธสง",04
0D3E B62E8EADD2MSG_PAYA:  DFB "ถ.พญาไท",04
0D46 B62EC8C3D5MSG_SRIAYUT:  DFB "ถ.ศรีอยุธยา",04
0D52 A1C3C1B7D2MSG_HIGHWAY:  DFB "กรมทางหลวง",04
0D5D C7D1B4E0BAMSG_WATBEN:  DFB "วัดเบญจมบพิตร",04
0D6B BEC3D0BAC3MSG_PARUP:  DFB "พระบรมรูปทรงม้า",04
0D7B A1C3D0B7C3MSG_KASUANG:  DFB "กระทรวงเกษตรและสหกรณ์",04
0D91 CAD08ED289MSG_PANFA:  DFB "สะพานผ่านฟ้า",04
0D9E CDB9D8CAD2MSG_PACHA:  DFB "อนุสาวรีย์ประชาธิปไตย",04
0DB4 CAD08ED289MSG_PAPIN:  DFB "สะพานพระปิ่นเกล้า",04
0DC6 B62E8EADD2MSG_BAROM:  DFB "ถ.พระบรมราชชนนี",04
0DD6 CAB6D2B9D5MSG_SAITAF:  DFB "สถานีขนส่งสามใต้ใหม่",04

```

```

0DEB BAD2B704 MSG_BAHT: DFB "บาท",04

```

```

0DEF      INCL "RTC.ASM"
;SUBROUTINE FOR PROCESS WITH REAL TIME CLOCK CHIP
;DS1202
;PORT PIN DEFINE
0090 = SCLKRTC: EQU P1.0 ;SERIAL CLOCK DS1202 RTC
0091 = IORTC: EQU P1.1 ;I/O DS1202 RTC
0092 = RSTRTC: EQU P1.2 ;RST DS1202 RTC

```

```

; _____ SETTIME SUB _____
; SETTIME SUBROUTINE
; INPUT = MINUTE, HOUR
; REG = R2, R3

```

```

0DEF      SETTIME:
0DEF C292      CLR RSTRTC
0DF1 C291      CLR IORTC
0DF3 C290      CLR SCLKRTC
0DF5 7B8E      MOV R3,#8EH ;WRITE PROTECT = 0
0DF7 7C00      MOV R4,#00H
0DF9 120E43    LCALL RTCWR
0DFC 7B80      MOV R3,#80H ;SECOND
0DFE 7C00      MOV R4,#00
0E00 120E43    LCALL RTCWR

```

```

0E03 7B82      MOV R3,#82H ;MINUTE = 0
0E05 AC10      MOV R4,MINUTE
0E07 120E43    LCALL RTCWR

```

```

0E0A 7B84      MOV R3,#84H ;HOUR = 0
0E0C AC11      MOV R4,HOUR
0E0E 120E43    LCALL RTCWR

```

```

0E11 7B8E      MOV R3,#8EH ;WRITE PROTECT = 1
0E13 7C80      MOV R4,#80H
0E15 120E43    LCALL RTCWR
0E18 22        RET

```

```

; _____ SETDATE SUB _____
; SETDATE SUBROUTINE
; INPUT = DATE, YEAR, MONTH
; REG = R2, R3

```

```

0E19      SETDATE:
0E19 C292      CLR RSTRTC
0E1B C291      CLR IORTC
0E1D C290      CLR SCLKRTC
0E1F 7B8E      MOV R3,#8EH ;WRITE PROTECT = 0
0E21 7C00      MOV R4,#00H
0E23 120E43    LCALL RTCWR

```

```

0E26 7B86      MOV R3,#86H ;DATE
0E28 AC12      MOV R4,DATE
0E2A 120E43    LCALL RTCWR

```

```

0E2D 7B88      MOV R3,#88H ;MONTH
0E2F AC13      MOV R4,MONTH
0E31 120E43    LCALL RTCWR

```

```

0E34 7B8C      MOV R3,#8CH ;YEAR
0E36 AC14      MOV R4,YEAR
0E38 120E43    LCALL RTCWR

```

```

0E3B 7B8E      MOV R3,#8EH ;WRITE PROTECT = 1
0E3D 7C80      MOV R4,#80H
0E3F 120E43    LCALL RTCWR
0E42 22        RET

```

```

; _____ SUB _____
; WRITE SINGLE BYTE TO DS1202
; IN = R3 COMMAND
; R4 DATA
; REG = A,R2,R3,R4,R5

```

```

0E43      RTCWR:
0E43 C290      CLR SCLKRTC ;CLK=0
0E45 C291      CLR IORTC
0E47 120EC5    LCALL RTCDL
0E4A D292      SETB RSTRTC ;RST=1
0E4C 120EC5    LCALL RTCDL
0E4F EB        MOV A,R3 ;COMMAND
0E50 120E5D    LCALL RTCWRC ;WRITE COMMAND/DATA
0E53 EC        MOV A,R4 ;DATA
0E54 120E5D    LCALL RTCWRC ;WRITE COMMAND/DATA
0E57 C292      CLR RSTRTC ;RST=0
0E59 120EC5    LCALL RTCDL
0E5C 22        RET

```

```

0E5D      RTCWRC:
0E5D 7908      MOV R1,#8 ;WRITE COMMAND/DATA 8 BIT
0E5F 13        RTCWRC1: RRC A
0E60 9291      MOV IORTC,C ;PUT DATA
0E62 D290      SETB SCLKRTC ;RISING EDGE CLOCK
0E64 120EC5    LCALL RTCDL
0E67 C290      CLR SCLKRTC ;FALLING EDGE CLOCK
0E69 120EC5    LCALL RTCDL
0E6C D9F1      DJNZ R1,RTCWRC1
0E6E 22        RET

```

```

; _____ RDTIME SUB _____
; READ TIME SUBROUTINE
; OUTPUT = SECOND, MINUTE, HOUR, DAY, DATE, YEAR, MONTH
; REG = R2, R3, A, DPTR

```

```

0E6F      RDTIME:
0E6F 7B81      MOV R3,#81H
0E71 120E9A    LCALL RTCRD
0E74 8C15      MOV SECOND,R4

```

```

0E76 7B83      MOV R3,#83H ;READ MINUTE TIME
0E78 120E9A    LCALL RTCRD
0E7B 8C10      MOV MINUTE,R4

```

```

0E7D 7B85      MOV R3,#85H ;READ HOUR TIME
0E7F 120E9A    LCALL RTCRD
0E82 8C11      MOV HOUR,R4

```

```

0E84 7B8D      MOV R3,#8DH ;READ YEAR TIME

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
 "ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้"

```

0E86 120E9A      LCALL RTCRD
0E89 8C14        MOV  YEAR,R4

;----- SUB -----
; WRITE DATA
; IN = ACC : ADDRESS
; R1 : DATA
0E8B 7889        MOV  R3,#89H      ;READ MONTH TIME
0E8D 120E9A      LCALL RTCRD
0E90 8C13        MOV  MONTH,R4

0E92 7887        MOV  R3,#87H      ;READ DATE TIME
0E94 120E9A      LCALL RTCRD
0E97 8C12        MOV  DATE,R4

;
; MOV  R3,#8BH      ;READ DAY TIME
; LCALL RTCRD
; MOV  DAY,R4
0E99 22         RET

;----- RTCRD SUB -----
; READ SINGLE BYTE FROM DS1202
; IN = R3 COMMAND
; OUT = R4 DATA
; REG = A,R2,R3,R4

0E9A          RTCRD:
0E9A C290        CLR  SCLKRTC      ;CLK=0
0E9C C291        CLR  IORTC
0E9E 120EC5      LCALL RTCDL
0EA1 D292        SETB RSTRTC      ;RST=1
0EA3 120EC5      LCALL RTCDL
0EA6 EB         MOV  A,R3        ;COMMAND
0EA7 120E5D      LCALL RTCWRC     ;WRITE COMMAND/DATE

0EAA D291        SETB IORTC
0EAC 7908        MOV  R1,#8       ;READ DATA
0EAE E4         CLR  A
0EAF          RTCRD1:
0EAF A291        MOV  C,IORTC
0EB1 13         RRC  A
0EB2 D290        SETB SCLKRTC
0EB4 120EC5      LCALL RTCDL
0EB7 C290        CLR  SCLKRTC
0EB9 120EC5      LCALL RTCDL
0EBC D9F1        DJNZ R1,RTCRD1
0EBE FC         MOV  R4,A
0EBF C292        CLR  RSTRTC     ;RST = 0
0EC1 120EC5      LCALL RTCDL
0EC4 22         RET

;----- SUB -----
; ERASE DATA ALL BYTE
; IN = ACC : ADDRESS
; R1 : DATA
0EF0 120FA3      LCALL EEPDL     ;DELAY ABOUT 15SMS
0EF3 22         RET

0EF4          WRALEEP:
0EF4 7480        MOV  A,#10000000B ;100XXXXXX OPCODE
0EF6 120F4A      LCALL WROP
0EF9 7420        MOV  A,#01000000B ;X01XXXXXX
0EFB 120F68      LCALL WR7BIT   ;WRITE ADDRESS
0F0E E9         MOV  A,R1
0F0F 120F7B      LCALL WR8BIT   ;WRITE DATA

0F12 C293        CLR  CSEEP
0F14 C295        CLR  DIEEP     ;CLEAR DI,CS PIN
0F16 120FA3      LCALL EEPDL     ;DELAY ABOUT 15SMS
0F19 22         RET

;----- SUB -----
; ERASE DATA ALL BYTE
; IN = ACC : ADDRESS
; R1 : DATA
0F1A          ERALEEEP:
0F1A 7480        MOV  A,#10000000B ;100XXXXXX OPCODE
0F1C 120F4A      LCALL WROP
0F1F 7440        MOV  A,#01000000B ;X10XXXXXX
0F21 120F68      LCALL WR7BIT   ;WRITE ADDRESS
0F24 C293        CLR  CSEEP
0F26 C295        CLR  DIEEP     ;CLEAR DI,CS PIN
0F28 120FA3      LCALL EEPDL     ;DELAY ABOUT 15SMS
0F2B 22         RET

;----- SUB -----
; ENABLE WRITE/ERASE DATA
; IN = ACC : ADDRESS
; R1 : DATA
0F2C          EWENEEEP:
0F2C 7480        MOV  A,#10000000B ;100XXXXXX OPCODE
0F2E 120F4A      LCALL WROP
0F31 7460        MOV  A,#01100000B ;X11XXXXXX OPCODE
0F33 120F68      LCALL WR7BIT   ;WRITE ADDRESS
0F36 C293        CLR  CSEEP
0F38 C295        CLR  DIEEP     ;CLEAR DI,CS PIN
0F3A 22         RET

;----- SUB -----
; DISABLE WRITE/ERASE DATA
; IN = ACC : ADDRESS
; R1 : DATA
0F3B          EWDEEEP:
0F3B 7480        MOV  A,#10000000B ;100XXXXXX OPCODE
0F3D 120F4A      LCALL WROP
0F40 7400        MOV  A,#00000000B ;X00XXXXXX OPCODE
0F42 120F68      LCALL WR7BIT   ;WRITE ADDRESS
0F45 C293        CLR  CSEEP
0F47 C295        CLR  DIEEP     ;CLEAR DI,CS PIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

0FE0 7408      MOV A,#08
0FE2 F0 MOVX @DPTR,A
0FE3 1582      DEC DPL
0FE5 7400      MOV A,#00H
0FE7 F0 MOVX @DPTR,A
           ;SET HIGH ADDRESS
0FE8 90A001    MOV DPTR,#CNTLCM
0FE9 7408      MOV A,#09
0FEF F0 MOVX @DPTR,A
0FEE 1582      DEC DPL
0FF0 7400      MOV A,#00H
0FF2 F0 MOVX @DPTR,A
           ;SET CURSOR LOW ADDRESS POINTER
0FF3 90A001    MOV DPTR,#CNTLCM
0FF6 740A      MOV A,#0AH
0FF8 F0 MOVX @DPTR,A
0FF9 1582      DEC DPL
0FFB 7400      MOV A,#00H
0FFD F0 MOVX @DPTR,A
           ;SET CURSOR HIGH ADDRESS POINTER
0FFE 90A001    MOV DPTR,#CNTLCM
1001 740B      MOV A,#0BH
1003 F0 MOVX @DPTR,A
1004 1582      DEC DPL
1006 7400      MOV A,#00H
1008 F0 MOVX @DPTR,A
1009 22 RET
           ;
           ;CLEAR SCREEN OF LCM
           ;SPEC OF LCM. 240 (DOT)ROW X 64(SCAN LINES)COL
           ;
100A 7400      CLRSCRN MOV A,#00
100C 121090    LCALL CURLOW
100F 7400      MOV A,#00
1011 12109E    LCALL CURHI
1014 7D04      MOV R5,#64
1016 7C1E      CLRSCRN2: MOV R4,#30
           ; 64 SCAN LINES
           ; 240 DOTS PER LINE ( OR 30 X 8 )
)
1018 90A001    CLRSCRN1: MOV DPTR,#CNTLCM
101B 740C      MOV A,#0CH
101D F0 MOVX @DPTR,A
101E 1582      DEC DPL
1020 7400      MOV A,#00H
           ; WRITE BLANK ( 8 DOTS/TIME )
1022 F0 MOVX @DPTR,A
1023 00 NOP
1024 00 NOP
1025 DCF1      DJNZ R4,CLRSCRN1
1027 D0ED      DJNZ R5,CLRSCRN2
1029 22 RET
           ;
           ;
           ;CLEAR LINE 0
           ;
102A 7400      CLRLN0: MOV A,#00
102C 121090    LCALL CURLOW
102F 7400      MOV A,#00
1031 12109E    LCALL CURHI
1034 7D14      MOV R5,#20
           ; 20 SCAN LINES
1038 7C1E      CLRLN02: MOV R4,#30
           ; 240 DOTS PER LINE
1038 90A001    CLRLN01: MOV DPTR,#CNTLCM
103B 740C      MOV A,#0CH
103D F0 MOVX @DPTR,A
103E 1582      DEC DPL
1040 7400      MOV A,#00H
           ; WRITE BLANK ( 8 DOTS/TIME )
1042 F0 MOVX @DPTR,A
1043 DCF3      DJNZ R4,CLRLN01
1045 D0EF      DJNZ R5,CLRLN02
1047 22 RET
           ;
           ;
           ;CLEAR LINE 1
           ;
1048 7478      CLRLN1: MOV A,#76H
104A 121090    LCALL CURLOW
104D 7402      MOV A,#02H
104F 12109E    LCALL CURHI
1052 7D14      MOV R5,#20
           ; 20 SCAN LINES
1054 7C1E      CLRLN12: MOV R4,#30
           ; 240 DOTS PER LINE
1056 90A001    CLRLN11: MOV DPTR,#CNTLCM
1059 740C      MOV A,#0CH
105B F0 MOVX @DPTR,A
105C 1582      DEC DPL
105E 7400      MOV A,#00H
           ; WRITE BLANK ( 8 DOTS/TIME )
1060 F0 MOVX @DPTR,A
1061 DCF3      DJNZ R4,CLRLN11
1063 D0EF      DJNZ R5,CLRLN12
1065 22 RET
           ;
           ;
           ;CLEAR LINE 2
           ;
1066 74CE      CLRLN20: MOV A,#0CEH
1068 121090    LCALL CURLOW
106B 7404      MOV A,#04H
106D 12109E    LCALL CURHI
1070 7D14      MOV R5,#20
           ; 20 SCAN LINES
1072 7C1E      CLRLN22: MOV R4,#30
           ; 240 DOTS PER LINE
1074 90A001    CLRLN21: MOV DPTR,#CNTLCM
1077 740C      MOV A,#0CH
1079 F0 MOVX @DPTR,A
107A 1582      DEC DPL
107C 7400      MOV A,#00H
           ; WRITE BLANK ( 8 DOTS/TIME )
107E F0 MOVX @DPTR,A
107F DCF3      DJNZ R4,CLRLN21
1081 D0EF      DJNZ R5,CLRLN22
1083 22 RET
           ;
           ;
           ;WRITE 1 BYTE TO LCM
           ;DATA IN : ACC
           ;LOCATION AT PRESENT CURSOR POSITION NOT EFFECT.
           ;
1084 F9      WR1BYTE: MOV R1,A
1085 90A001    MOV DPTR,#CNTLCM
           ;WRITE TO LCM
1088 740C      MOV A,#0CH
108A F0 MOVX @DPTR,A
108B 1582      DEC DPL
108D E9 MOV A,R1
           ; WRITE 8 DOTS/TIME
108E F0 MOVX @DPTR,A
108F 22 RET
           ;
           ;
           ;SET CURSOR LOW ADDRESS POINTER
           ;DATA IN A
           ;NOTE : CURSOR POSITION = POSITION OF THE DOT ON LCM
           ;SEPARATE IN HIGH BYTE AND LOW BYTE
           ;
1090 C0E0      CURLOW: PUSH A
           ;SET POSITION OF CURSOR
1092 90A001    MOV DPTR,#CNTLCM
1095 740A      MOV A,#0AH
1097 F0 MOVX @DPTR,A
1098 1582      DEC DPL
109A D0E0      POP A
109C F0 MOVX @DPTR,A
109D 22 RET
           ;
           ;
           ;SET CURSOR HIGH ADDRESS POINTER
           ;DATA IN A
           ;
109E C0E0      CURHI: PUSH A
           ;SET POSITION OF CURSOR
10A0 90A001    MOV DPTR,#CNTLCM
10A3 740B      MOV A,#0BH
10A5 F0 MOVX @DPTR,A
10A6 1582      DEC DPL
10A8 D0E0      POP A
10AA F0 MOVX @DPTR,A
10AB 22 RET
           ;
           ;
           ;SET CURSOR TO HOME POSITION
           ;
10AC 7820      CURHOME: MOV R0,#CURROW
           ;INIT CURSOR POSITION
10AE 7600      MOV @R0,#00
           ;ROW 0 , COL 0
10B0 08 INC R0
10B1 7600      MOV @R0,#00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

10B3 1210C0  LCALL CURCONV      ;CONVERT TO HEX POSITION
10B6 22  RET
:
:
:
;SET CURSOR TO BOTTOM POSITION
:
:
10B7 7820  CURBOTM: MOV R0,#CURROW
10B9 761D  MOV @R0,#29
10BB 08  INC R0
10BC 1210C0  LCALL CURCONV
10BF 22  RET
:
:
:
;CAL ROW NUMBER AND COL NUMBER IN HEX POSITION OF LCM
;DATA IN INTERNAL RAM 50H(ROW),51H(COL) STORE ROW,COL
;DATA IN INTERNAL RAM 52H(ROW),53H(COL) STORE HEX POS
:
:
10C0  CURCONV:
10C0 7820  MOV R0,#CURROW      ;GET ROW
10C2 E6  MOV A,@R0
10C3 B40005  CJNE A,#00,CONV1  ;IF ROW 0
10C6 9000F0  MOV DPTR,#240          ;COL 0 ARE 240
10C9 8013  SJMP CONV2
10CB B40105  CONV1: CJNE A,#01,CONV3  ;IF ROW 1
10CE 900348  MOV DPTR,#840          ;COL 0 ARE 1080
10D1 8008  SJMP CONV2
10D3 B40205  CONV3: CJNE A,#02,CONV4  ;IF ROW 2
10D6 9005A0  MOV DPTR,#1440         ;COL 0 ARE 1440
10D9 8003  SJMP CONV2
10DB 900708  CONV4: MOV DPTR,#1800       ;IF NOT ABOVE,SET TO 1800
10DE 08  CONV2: INC R0          ; R0 = 51H
10DF C3  CLR C
10E0 E6  MOV A,@R0          ;GET COL NO.
10E1 3582  ADDC A,DPL          ;ADD WITH COL 0
10E3 F582  MOV DPL,A          ;SAVE BACK
10E5 08  INC R0            ; R0 = 53H = HEX LO
10E6 08  INC R0
10E7 F6  MOV @R0,A
10E8 7400  MOV A,#00          ;IF CARRY
10EA 3583  ADDC A,DPH          ;ADD CARRY
10EC F583  MOV DPH,A          ;SAVE BACK
10EE 18  DEC R0            ; R0 =52H = HEX HI
10EF F6  MOV @R0,A
;THEN DISPLAY CURSOR AT SET CURSOR POSITION
10F0 E583  MOV A,DPH
10F2 C0E0  PUSH A
10F4 E582  MOV A,DPL
10F6 121090  LCALL CURLOW
10F9 D0E0  POP A
10FB 12109E  LCALL CURHI
10FE 22  RET
:
:
:
;WRITE THAI CHARACTER ( NORMAL LEVEL IS PAYANCHANA).
;AFTER DID IT,CURSOR POSITION INCREMENT BY ONE
;ASCII CODE IN ACC.
;CUR POS IN HEX IN 52H & 53H , TEMP USE 54H & 55H
:
:
10FF 9013EC  WRCHR: MOV DPTR,#THAIPAT  ;CAL START ADDRESS OF
PATTERN
1102 75F008  MOV B,#08H          ; 8 BYTE / CHARACTER
1105 A4  MUL AB          ; ASCII CODE * 8
1106 C3  CLR C
1107 3582  ADDC A,DPL          ; DPTR + DATA IN AB
1109 F582  MOV DPL,A          ; DPL = DPL + A
110B E5F0  MOV A,B          ; DPH = DPH + B
110D 3583  ADDC A,DPH
110F F583  MOV DPH,A
1111 C083  PUSH DPH          ;SAVE POINTER WHICH POINT TO PAT-
TERN
1113 C082  PUSH DPL
:
;TRANSFER HEX CUR POS IN TEMP
:
:
1115 C3  CLR C          ;COPY CURRENT CUR POS TO TEMP
1116 7822  MOV R0,#22H      ;BECAUSE IN THIS SUBROUTINE WE
MUST
1118 7924  MOV R1,#24H      ;MODIFY CURRENT POSITION IN TEMP
:
:
:
111A E6  MOV A,@R0          ;;52) = (54) . (53) = (55)
111B F7  MOV @R1,A
111C 08  INC R0
111D 09  INC R1
111E E6  MOV A,@R0
111F F7  MOV @R1,A
;WRITE PATTERN OF CHARACTER TO LCM
1120 7B08  MOV R3,#08          ;WRITE 8 BYTE / CHARACTER
1122 7825  WRCHR1: MOV R0,#25H      ;SET CURSOR ADDRESS
1124 E6  MOV A,@R0
1125 121090  LCALL CURLOW          ;SET LOW ADDRESS
1128 7824  MOV R0,#24H
112A E6  MOV A,@R0
112B 12109E  LCALL CURHI          ;SET HIGH ADDRESS
112E D082  POP DPL          ;GET POINTER BACK
1130 D083  POP DPH
1132 7400  MOV A,#00H
1134 93  MOVC A,@A+DPTR      ;GET PATTERN 1 BYTE
1135 A3  INC DPTR          ;NEXT BYTE OF PATTERN
1136 C083  PUSH DPH          ;SAVE BACK
1138 C082  PUSH DPL
113A 121084  LCALL WR1BYTE        ;WRITE DATA ON LCM 1 BYTE
113D 121169  LCALL ADD1LN        ;SHIFT CURSOR TO NEXT SCAN LINE
1140 DBE0  DJNZ R3,WRCHR1      ;COMPLETE ?
1142 D083  POP DPH          ;YES POP DPTR OUT
1144 D082  POP DPL
;INCREMENT CURSOR POSITION ;INCREMENT CURSOR TO NEXT
POSITION
1146 7821  MOV R0,#CURCOL
1148 06  INC @R0
1149 1210C0  LCALL CURCONV
114C 22  RET
:
:
:
;SHIFT CURSOR BACK 4 SCAN LINES FOR WRITE I,EE
:
:
114D C3  SUB4LN: CLR C          ;SUB 4 LINE SCAN
114E 7825  MOV R0,#25H          ;GET LOW BYTE OF CUR POS
1150 E6  MOV A,@R0
1151 9478  SUBB A,#120          ;4 LINE SCAN
1153 F6  MOV @R0,A
1154 5004  JNC SUB4LN1
1156 18  DEC R0            ; R0 = 54H
1157 E6  MOV A,@R0
1158 14  DEC A
1159 F6  MOV @R0,A
115A 22  SUB4LN1: RET
:
:
:
;SHIFT CURSOR BACK 8 SCAN LINES FOR WRITE EK,TOW
:
:
:
115B C3  SUB8LN: CLR C          ;SUB 8 LINE SCAN
115C 7825  MOV R0,#25H          ;GET LOW BYTE OF CUR POS
115E E6  MOV A,@R0
115F 94F0  SUBB A,#240          ;8 LINE SCAN
1161 F6  MOV @R0,A
1162 5004  JNC SUB8LN1
1164 18  DEC R0            ; R0 = 54H
1165 E6  MOV A,@R0
1166 14  DEC A
1167 F6  MOV @R0,A
1168 22  SUB8LN1: RET
:
:
:
;SHIFT CURSOR TO NEXT SCAN LINE FOR WRITE OTHER DOTS
:
:
:
1169 C3  ADD1LN: CLR C          ;ADD 1 LINE SCAN
116A 7825  MOV R0,#25H          ;GET LOW BYTE OF CUR POS
116C E6  MOV A,@R0
116D 341E  ADDC A,#30          ;1 LINE SCAN
116F F6  MOV @R0,A
1170 5004  JNC ADD1LN1
1172 18  DEC R0
1173 E6  MOV A,@R0
1174 04  INC A
1175 F6  MOV @R0,A
1176 22  ADD1LN1: RET
:
:
:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;SHIFT CURSOR TO NEXT 8 SCAN LINES FOR WRITE U,UU
;
1177 C3 ADD8LN: CLR C ;ADD 8 LINE SCAN
1178 7825 MOV R0,#25H ;GET LOW BYTE OF CUR POS
117A E6 MOV A,@R0
117B 34F0 ADDC A,#240 ;8 LINE SCAN
117D F6 MOV @R0,A
117E 5004 JNC ADD8LN1
1180 18 DEC R0 ; R0 = 54H
1181 E6 MOV A,@R0
1182 04 INC A
1183 F6 MOV @R0,A
1184 22 ADD8LN1: RET
;
;WRITE U,UU AFTER DID IT CURSOR STILL THE SAME
;
1185 9013EC WRU: MOV DPTR,#THAIPAT ;CAL START ADDRESS OF
PATTERN
1188 75F008 MOV B,#08H ; 8 BYTE / CHARACTER
118B A4 MUL AB ; ASCII CODE * 8
118C C3 CLR C
118D 3582 ADDC A,DPL ; DPTR + DATA IN AB
118F F582 MOV DPL,A ; DPL = DPL + A
1191 E5F0 MOV A,B ; DPH = DPH + B
1193 3583 ADDC A,DPH
1195 F583 MOV DPH,A
1197 C083 PUSH DPH ;SAVE POINTER WHICH POINT TO PAT-
TERN
1199 C082 PUSH DPL
;TRANSFER HEX CUR POS IN TEMP
119B C3 CLR C ;COPY CURRENT CUR POS TO TEMP
119C 7822 MOV R0,#22H
119E 7924 MOV R1,#24H
11A0 E8 MOV A,@R0 ;(52) = (54) , (53) = (55)
11A1 F7 MOV @R1,A
11A2 08 INC R0
11A3 09 INC R1
11A4 E8 MOV A,@R0
11A5 F7 MOV @R1,A
11A6 121177 LCALL ADD8LN
11A9 7804 MOV R3,#04 ;WRITE 4 BYTE / CHARACTER
11AB 7825 WRU1: MOV R0,#25H ;SET CURSOR ADDRESS
11AD E6 MOV A,@R0
11AE 121080 LCALL CURLOW
11B1 7824 MOV R0,#24H
11B3 E6 MOV A,@R0
11B4 12109E LCALL CURHI
11B7 D082 POP DPL ;GET POINTER BACK
11B9 D083 POP DPH
11BB 7400 MOV A,#00
11BD 93 MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
11BE A3 INC DPTR ;NEXT BYTE OF PATTERN
11BF C083 PUSH DPH ;SAVE BACK
11C1 C082 PUSH DPL
11C3 121084 LCALL WR1BYTE ;WRITE DATA ON LCM 1 BYTE
11C6 121169 LCALL ADD1LN ;SHIFT CURSOR TO NEXT SCAN LINE
11C9 DBE0 DJNZ R3,WRU1 ;COMPLETE ?
11CB D083 POP DPH ;YES POP DPTR OUT
11CD D082 POP DPL
11CF 22 RET
;
;WRITE I,EE AT FIST LEVEL
;AFTER DID IT CUSOR STILL THE SAME
;
11D0 9013EC WRIE: MOV DPTR,#THAIPAT ;CAL START ADDRESS OF
PATTERN
11D3 75F008 MOV B,#08H ; 8 BYTE / CHARACTER
11D6 A4 MUL AB ; ASCII CODE * 8
11D7 C3 CLR C
11D8 3582 ADDC A,DPL ; DPTR + DATA IN AB
11DA F582 MOV DPL,A ; DPL = DPL + A
11DC E5F0 MOV A,B ; DPH = DPH + B
11DE 3583 ADDC A,DPH
11E0 F583 MOV DPH,A
11E2 C083 PUSH DPH ;SAVE POINTER WHICH POINT TO PAT-
TERN
11E4 C082 PUSH DPL
;TRANSFER HEX CUR POS IN TEMP
11E6 C3 CLR C ;COPY CURRENT CUR POS TO TEMP
11E7 7822 MOV R0,#22H
11E9 7924 MOV R1,#24H
11EB E6 MOV A,@R0 ;(52) = (54) , (53) = (55)
11EC F7 MOV @R1,A
11ED 08 INC R0
11EE 09 INC R1
11EF E6 MOV A,@R0
11F0 F7 MOV @R1,A
;DEC 4 SCAN LINES DUE TO WRITE FROM TOP TO BOTTOM
11F1 12114D LCALL SUB4LN
11F4 7804 MOV R3,#04 ;WRITE 4 BYTE / CHARACTER
11F6 7825 WRU1: MOV R0,#25H ;SET CURSOR ADDRESS
11F8 E6 MOV A,@R0
11F9 121090 LCALL CURLOW
11FC 7824 MOV R0,#24H
11FE E6 MOV A,@R0
11FF 12109E LCALL CURHI
1202 D082 POP DPL ;GET POINTER BACK
1204 D083 POP DPH
1206 7400 MOV A,#00
1208 93 MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
1209 A3 INC DPTR ;NEXT BYTE OF PATTERN
120A C083 PUSH DPH ;SAVE BACK
120C C082 PUSH DPL
120E 121084 LCALL WR1BYTE ;WRITE DATA ON LCM 1 BYTE
1211 121169 LCALL ADD1LN ;SHIFT CURSOR TO NEXT SCAN LINE
1214 DBE0 DJNZ R3,WRU1 ;COMPLETE ?
1216 D083 POP DPH ;YES POP DPTR OUT
1218 D082 POP DPL
121A 22 RET
;
;WRITE EK.TOW AT SECOND LEVEL
;AFTER DID IT CURSOR STILL THE SAME
;
121B 9013EC WREKTOW: MOV DPTR,#THAIPAT ;CAL START ADDRESS OF
PATTERN
121E 75F008 MOV B,#08H ; 8 BYTE / CHARACTER
1221 A4 MUL AB ; ASCII CODE * 8
1222 C3 CLR C
1223 3582 ADDC A,DPL ; DPTR + DATA IN AB
1225 F582 MOV DPL,A ; DPL = DPL + A
1227 E5F0 MOV A,B ; DPH = DPH + B
1229 3583 ADDC A,DPH
122B F583 MOV DPH,A
122D C083 PUSH DPH ;SAVE POINTER WHICH POINT TO PAT-
TERN
122F C082 PUSH DPL
;TRANSFER HEX CUR POS IN TEMP
1231 C3 CLR C ;COPY CURRENT CUR POS TO TEMP
1232 7822 MOV R0,#22H
1234 7924 MOV R1,#24H
1236 E6 MOV A,@R0 ;(52) = (54) , (53) = (55)
1237 F7 MOV @R1,A
1238 08 INC R0
1239 09 INC R1
123A E6 MOV A,@R0
123B F7 MOV @R1,A
;DEC 8 SCAN LINES DUE TO WRITE FROM TOP TO BOTTOM
123C 12115B LCALL SUB8LN
123F 7804 MOV R3,#04 ;WRITE 4 BYTE / CHARACTER
1241 7825 WRU2: MOV R0,#25H ;SET CURSOR ADDRESS
1243 E6 MOV A,@R0
1244 121090 LCALL CURLOW
1247 7824 MOV R0,#24H
1249 E6 MOV A,@R0
124A 12109E LCALL CURHI
124D D082 POP DPL ;GET POINTER BACK
124F D083 POP DPH
1251 7400 MOV A,#00
1253 93 MOVC A,@A+DPTR ;GET PATTERN 1 BYTE
1254 A3 INC DPTR ;NEXT BYTE OF PATTERN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



13AC 7821	MOV R0,#CURCOL	1504 36387F367F	DFB 36H,36H,7FH,36H,7FH,36H,36H,00H	: code-035,23H,#
13AE 06	INC @R0	150C 0C3E031E30	DFB 0CH,3EH,03H,1EH,30H,1FH,0CH,00H	: code-
13AF 1210C0	LCALL CURCONV	036,24H,\$		
13B2 22	RET	1514 006333180C	DFB 00H,63H,33H,18H,0CH,66H,63H,00H	: code-
13B3 C0E0	AZZ22: PUSH A	037,25H,%		
13B5 7821	MOV R0,#CURCOL	151C 0C120C0C52	DFB 0CH,12H,0CH,0CH,52H,22H,5CH,00H	: code-
13B7 16	DEC @R0	038,26H,&		
13B8 1210C0	LCALL CURCONV	1524 0606030000	DFB 06H,06H,03H,00H,00H,00H,00H,00H	: code-039,27H,'
13B8 D0E0	POP A	152C 180C060606	DFB 18H,0CH,06H,06H,06H,0CH,18H,00H	: code-040,28H,(
13BD 12121B	LCALL WREKTOW	1534 060C181818	DFB 06H,0CH,18H,18H,18H,0CH,06H,00H	: code-041,29H.)
13C0 7821	MOV R0,#CURCOL	153C 00683CFF3C	DFB 00H,66H,3CH,0FFH,3CH,66H,00H,00H	: code-
13C2 06	INC @R0	042,2AH,*		
13C3 1210C0	LCALL CURCONV	1544 000C0C3F0C	DFB 00H,0CH,0CH,3FH,0CH,0CH,00H,00H	: code-
13C8 22	RET	043,2BH,+		
13C7 C0E0	AZZ22: PUSH A	154C 0000000000	DFB 00H,00H,00H,00H,00H,0CH,0CH,06H	: code-044,2CH,,
13C9 7821	MOV R0,#CURCOL	1554 000003F000	DFB 00H,00H,00H,3FH,00H,00H,00H,00H	: code-045,2DH,-
13CB 16	DEC @R0	155C 0000000000	DFB 00H,00H,00H,00H,00H,0CH,0CH,00H	: code-046,2EH.,
13CC 1210C0	LCALL CURCONV	1564 6030180C08	DFB 60H,30H,18H,0CH,06H,03H,01H,00H	: code-047,2FHJ
13CF D0E0	POP A			
13D1 121185	LCALL WRU	156C 3E63737B6F	DFB 3EH,63H,73H,7BH,6FH,67H,3EH,00H	: code-
13D4 7821	MOV R0,#CURCOL	048,30H,0		
13D6 06	INC @R0	1574 0C0E0C0C0C	DFB 0CH,0EH,0CH,0CH,0CH,0CH,3FH,00H	: code-
13D7 1210C0	LCALL CURCONV	049,31H,1		
13DA 22	RET	157C 1E33301C08	DFB 1EH,33H,30H,1CH,06H,33H,3FH,00H	: code-
13DB 7820	AZZ23: MOV R0,#CURROW	050,32H,2		
13DD 06	INC @R0	1584 1E33301C30	DFB 1EH,33H,30H,1CH,30H,33H,1EH,00H	: code-
13DE 1210C0	LCALL CURCONV	051,33H,3		
13E1 02139E	LJMP AZZ3	158C 383C38337F	DFB 38H,3CH,36H,33H,7FH,30H,78H,00H	: code-
13E4 7821	AZZ24: MOV R0,#CURCOL	052,34H,4		
13E6 7800	MOV @R0,#00H	1594 3F031F3030	DFB 3FH,03H,1FH,30H,30H,33H,1EH,00H	: code-053,35H,5
13E8 1210C0	LCALL CURCONV	159C 1C08031F33	DFB 1CH,06H,03H,1FH,33H,33H,1EH,00H	: code-
13EB 22	RET	054,36H,6		
13EC	INCL "THAIPT.ASM"	15A4 3F3330180C	DFB 3FH,33H,30H,18H,0CH,0CH,0CH,00H	: code-
	: THAI PATTERN FOR SMO STANDARD	055,37H,7		
13EC	THAIPTAT:	15AC 1E33331E33	DFB 1EH,33H,33H,1EH,33H,33H,1EH,00H	: code-
13EC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	056,38H,8		
13FA 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15B4 1E33333E30	DFB 1EH,33H,33H,3EH,30H,18H,0EH,00H	: code-
13FC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	057,39H,9		
1404 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15BC 000C0C0000	DFB 00H,0CH,0CH,00H,00H,0CH,0CH,00H	: code-
140C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	058,3AH,:		
1414 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15CA 000C0C0000	DFB 00H,0CH,0CH,00H,00H,0CH,0CH,06H	: code-
141C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	059,3BH,;		
1424 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15CC 180C060306	DFB 18H,0CH,06H,03H,06H,0CH,18H,00H	: code-
142C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	060,3CH,<		
1434 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15D4 00003F0000	DFB 00H,00H,3FH,00H,00H,3FH,00H,00H	: code-061,3DH,=
143C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15DC 080C183018	DFB 06H,0CH,18H,30H,18H,0CH,06H,00H	: code-
1444 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	062,3EH,>		
144C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15E4 1E3330180C	DFB 1EH,33H,30H,18H,0CH,00H,0CH,00H	: code-
1454 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	063,3FH,?		
145C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15EC 3E637B7B7B	DFB 3EH,63H,7BH,7BH,7BH,03H,3EH,00H	: code-
1464 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	064,40H,@		
146C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15F4 0C1E33333F	DFB 0CH,1EH,33H,33H,3FH,33H,33H,00H	: code-
1474 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	065,41H,0A		
147C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	15FC 3F66663F66	DFB 3FH,66H,66H,3FH,66H,66H,3FH,00H	: code-
1484 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	066,42H,0B		
148C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	1604 3C66030303	DFB 3CH,66H,03H,03H,03H,66H,3CH,00H	: code-
1494 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	067,43H,0C		
149C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	160C 1F36666666	DFB 1FH,36H,66H,66H,66H,36H,1FH,00H	: code-
14A4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	068,44H,0D		
14AC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	1614 7F46161E16	DFB 7FH,46H,16H,1EH,16H,46H,7FH,00H	: code-
14B4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	069,45H,0E		
14BC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	161C 7F46161E16	DFB 7FH,46H,16H,1EH,16H,06H,0FH,00H	: code-
14CA 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	070,46H,0F		
14CC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	1624 3C8803037B	DFB 3CH,66H,03H,03H,7BH,66H,7CH,00H	: code-
14CD 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	071,47H,G		
14D4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	162C 3333333F33	DFB 33H,33H,33H,3FH,33H,33H,33H,00H	: code-072,48H,H
14DC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	1634 1E0C0C0C0C	DFB 1EH,0CH,0CH,0CH,0CH,0CH,1EH,00H	: code-
14E4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	073,49H,I		
14EC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	163C 7830303033	DFB 78H,30H,30H,30H,33H,33H,1EH,00H	: code-074,4AH,J
14F4 0C1E1E0C0C	DFB 0CH,1EH,1EH,0CH,0CH,0CH,00H,00H	1644 67661E1E36	DFB 67H,66H,1EH,1EH,36H,66H,67H,00H	: code-
033,21H,!		075,4BH,K		
14FC 3636362400	DFB 36H,36H,36H,24H,00H,00H,00H,00H	164C 0F06060648	DFB 0FH,06H,06H,06H,48H,66H,7FH,00H	: code-076,4CH,L
		1654 637777F6B	DFB 63H,77H,7FH,7FH,6BH,63H,63H,00H	: code-
		077,4DH,M		
		165C 636767B7B3	DFB 63H,67H,6FH,7BH,73H,63H,63H,00H	: code-
		078,4EH,N		
		1664 1C36636363	DFB 1CH,36H,63H,63H,63H,36H,1CH,00H	: code-
		079,4FH,O		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

166C 3F66683E06	DFB 3FH,66H,66H,3EH,06H,06H,0FH,00H	: code-080,50H,P	181C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-134,86H
1674 1E33333333	DFB 1EH,33H,33H,33H,33H,1EH,38H,00H	: code-081,51H,Q	1824 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-135,87H
167C 3F66683E38	DFB 3FH,66H,66H,3EH,36H,66H,67H,00H	: code-	182C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-136,88H
082,52H,R			1834 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-137,89H
1684 1E33060C18	DFB 1EH,33H,06H,0CH,18H,33H,1EH,00H	: code-	183C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-138,8AH
083,53H,S			1844 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-139,8BH
168C 3F2D0C0C0C	DFB 3FH,2DH,0CH,0CH,0CH,0CH,1EH,00H	: code-	184C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-140,8CH
084,54H,T			1854 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-141,8DH
1694 3333333333	DFB 33H,33H,33H,33H,33H,33H,3FH,00H	: code-085,55H,U	185C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-142,8EH
169C 3333333333	DFB 33H,33H,33H,33H,33H,1EH,0CH,00H	: code-	1864 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-143,8FH
086,56H,V			186C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-144,90H
16A4 6363636A7F	DFB 63H,63H,63H,6AH,7FH,77H,63H,00H	: code-	1874 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-145,91H
087,57H,W			187C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-146,92H
16AC 6363381C1C	DFB 63H,63H,36H,1CH,1CH,36H,63H,00H	: code-	1884 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-147,93H
088,58H,X			188C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-148,94H
16B4 3333331E0C	DFB 33H,33H,33H,1EH,0CH,0CH,1EH,00H	: code-	1894 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-149,95H
089,59H,Y			189C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-150,96H
16BC 7F6331184C	DFB 7FH,63H,31H,18H,4CH,66H,6FH,00H	: code-	18A4 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-151,97H
090,5AH,Z			18AC 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-152,98H
16C4 1E08080806	DFB 1EH,06H,06H,06H,06H,06H,1EH,00H	: code-091,58H,]	18B4 000000F808	DFB 00H,00H,00H,0F8H,08H,08H,08H,08H	: code-153,99H
16CC 03060C1830	DFB 03H,06H,0CH,18H,30H,60H,40H,00H	: code-	18BC 0000000F08	DFB 00H,00H,00H,0FH,08H,08H,08H,08H	: code-154,9AH
092,5CH,)			18C4 080808F800	DFB 08H,08H,08H,0F8H,08H,08H,08H,08H	: code-155,9BH
16D4 1E18181818	DFB 1EH,18H,18H,18H,18H,18H,1EH,00H	: code-093,5DH,]	18CC 0808080F00	DFB 08H,08H,08H,0FH,08H,08H,08H,08H	: code-156,9CH
16DC 081C386300	DFB 08H,1CH,36H,63H,00H,00H,00H,00H	: code-	18D4 0808080808	DFB 08H,08H,08H,08H,08H,08H,08H,08H	: code-157,9DH
094,5EH,^			18DC 080808F808	DFB 08H,08H,08H,0F8H,08H,08H,08H,08H	: code-158,9EH
16E4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-095,5FH,^	18E4 0808080F08	DFB 08H,08H,08H,0FH,08H,08H,08H,08H	: code-159,9FH
16EC 0C0C180000	DFB 0CH,0CH,18H,00H,00H,00H,00H,00H	: code-	18EC FFFFFFFF	DFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH	: smo-
096,60H,'			160,0A0H,all black		
16F4 00001E303E	DFB 00H,00H,1EH,30H,3EH,33H,6EH,00H	: code-097,61H,a	18F4 003C424444	DFB 00H,3CH,42H,44H,44H,44H,44H,00H	: smo-
16FC 0706063E86	DFB 07H,06H,06H,3EH,66H,66H,38H,00H	: code-098,62H,b	161,0A1H,kokai		
1704 00001E3303	DFB 00H,00H,1EH,33H,03H,33H,1EH,00H	: code-099,63H,c	18FC 0048545850	DFB 00H,48H,54H,58H,50H,50H,30H,00H	: smo-
170C 3830303E33	DFB 38H,30H,30H,3EH,33H,33H,6EH,00H	: code-100,64H,d	162,0A2H,khokhai		
1714 00001E333F	DFB 00H,00H,1EH,33H,3FH,03H,1EH,00H	: code-101,65H,e	1904 005A555250	DFB 00H,5AH,55H,52H,50H,50H,30H,00H	: smo-
171C 1C38060F06	DFB 1CH,36H,06H,0FH,06H,06H,0FH,00H	: code-102,66H,f	163,0A3H,khokhuad		
1724 00006E3131	DFB 00H,00H,6EH,31H,31H,3EH,30H,1FH	: code-103,67H,g	190C 003C42425A	DFB 00H,3CH,42H,42H,5AH,56H,44H,00H	: smo-
172C 0706368E88	DFB 07H,06H,36H,6EH,66H,66H,67H,00H	: code-104,68H,h	164,0A4H,korkhai		
1734 0C000E0C0C	DFB 0CH,00H,0EH,0CH,0CH,0CH,1EH,00H	: code-	1914 00245A425A	DFB 00H,24H,5AH,42H,5AH,56H,44H,00H	: smo-
103,69H,j			165,0A5H,korkon		
173C 3000303030	DFB 30H,00H,30H,30H,30H,33H,33H,1EH	: code-106,6AH,i	191C 004A55504C	DFB 00H,4AH,55H,50H,4CH,4AH,34H,00H	: smo-
1744 070686361E	DFB 07H,06H,66H,36H,1EH,36H,67H,00H	: code-107,6BH,k	166,0A6H,korakang		
174C 0E0C0C0C0C	DFB 0EH,0CH,0CH,0CH,0CH,0CH,1EH,00H	: code-	1924 0020546444	DFB 00H,20H,54H,64H,44H,48H,30H,00H	: smo-
108,6CH,l			167,0A7H,ngoringoo		
1754 0000337F7F	DFB 00H,00H,33H,7FH,7FH,6BH,63H,00H	: code-	192C 003C404854	DFB 00H,3CH,40H,48H,54H,58H,30H,00H	: smo-
109,6DH,m			168,0A8H,jorjarm		
175C 00001F3333	DFB 00H,00H,1FH,33H,33H,33H,33H,00H	: code-110,6EH,n	1934 001E21246A	DFB 00H,1EH,21H,24H,6AH,0ACH,58H,00H	: smo-
1764 00001E3333	DFB 00H,00H,1EH,33H,33H,33H,1EH,00H	: code-111,6FH,o	169,0A9H,chorching		
176C 00003B8688	DFB 00H,00H,3BH,66H,66H,3EH,06H,0FH	: code-112,70H,p	193C 8048345850	DFB 80H,48H,34H,58H,50H,48H,70H,00H	: smo-
1774 00006E3333	DFB 00H,00H,6EH,33H,33H,3EH,30H,78H	: code-113,71H,q	170,0AAH,chorchang		
177C 000038E8E8	DFB 00H,00H,3BH,6EH,66H,06H,0FH,00H	: code-114,72H,r	1944 804A353250	DFB 80H,4AH,35H,32H,50H,48H,70H,00H	: smo-
1784 00003E031E	DFB 00H,00H,3EH,03H,1EH,30H,1FH,00H	: code-115,73H,s	171,0ABH,sorso		
178C 080C3E0C0C	DFB 08H,0CH,3EH,0CH,0CH,2CH,18H,00H	: code-	194C 005EA1A2B3	DFB 00H,5EH,0A1H,0A2H,0B3H,0ABH,0D2H,00H	: smo-
116,74H,i			172,0ACH,choriachur		
1794 0000333333	DFB 00H,00H,33H,33H,33H,33H,6EH,00H	: code-117,75H,u	1954 008E919293	DFB 00H,8EH,91H,92H,93H,0F5H,82H,0F0H	: smo-
179C 0000333333	DFB 00H,00H,33H,33H,33H,1EH,0CH,00H	: code-118,76H,v	173,0ADH,yorying		
17A4 0000636B7F	DFB 00H,00H,63H,6BH,7FH,7FH,36H,00H	: code-119,77H,w	195C 003844484C	DFB 00H,38H,44H,48H,4CH,4AH,0C4H,7EH	: smo-
17AC 000063361C	DFB 00H,00H,63H,36H,1CH,36H,63H,00H	: code-	174,0AEH,chorchadar		
120,78H,x			1964 003844484C	DFB 00H,38H,44H,48H,4CH,4AH,0F4H,2AH	: smo-
17B4 0000333333	DFB 00H,00H,33H,33H,3EH,30H,1FH	: code-121,79H,y	175,0AFH,lorpatak		
17BC 00003F190C	DFB 00H,00H,3FH,19H,0CH,26H,3FH,00H	: code-	196C 403C624854	DFB 40H,3CH,62H,48H,54H,58H,22H,75H	: smo-
122,7AH,z			176,0B0H,lorlam		
17CA 380C0C070C	DFB 38H,0CH,0CH,07H,0CH,0CH,38H,00H	: code-	1974 006A55A4A8	DFB 00H,6AH,55H,4AH,48H,48H,48H,00H	: smo-
123,7BH,{			177,0B1H,lorlangmonto		
17CC 1818180018	DFB 18H,18H,18H,00H,18H,18H,18H,00H	: code-124,7CH,}	197C 009AA5ASBB	DFB 00H,9AH,0ASH,0ASH,0BBH,0ADH,56H,00H	: smo-
17D4 070C0C380C	DFB 07H,0CH,0CH,38H,0CH,0CH,07H,00H	: code-	178,0B2H,lorpooloa		
125,7DH,}			1984 004E515253	DFB 00H,4EH,51H,52H,53H,0B5H,52H,00H	: smo-
17DC CE38000000	DFB 0CEH,3BH,00H,00H,00H,00H,00H,00H	: code-	198C 003C424A58	DFB 00H,3CH,42H,4AH,56H,5AH,4CH,00H	: smo-
126,7EH,~			180,0B4H,dordex		
17E4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-127,7FH,	1994 00344AA4A56	DFB 00H,34H,4AH,4AH,56H,5AH,4CH,00H	: smo-
17EC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-128,80H	181,0B5H,lorloa		
17F4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-129,81H	199C 003C424446	DFB 00H,3CH,42H,44H,46H,4AH,44H,00H	: smo-
17FC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-130,82H	182,0B6H,lorlung		
1804 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-131,83H	19A4 00344AA4C48	DFB 00H,34H,4AH,4CH,48H,48H,48H,00H	: smo-
180C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-132,84H	183,0B7H,lorlaharn		
1814 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-133,85H	19AC 40387C4848	DFB 40H,38H,7CH,48H,48H,1CH,00H	: smo-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 - ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

184,0B8H,lorlong			223,0DFH,bahl		
19B4 00242A2C68	DFB 00H,24H,2AH,2CH,68H,0AH,5CH,00H	: smo-			
185,0B9H,nomoo			1AEC 0008080818	DFB 00H,08H,08H,08H,18H,28H,10H,00H	: smo-
19BC 00444A4C48	DFB 00H,44H,4AH,4CH,48H,48H,38H,00H	: smo-	224,0E0H,SARA-A		
186,0BAH,torbalmal			1AF4 0012121236	DFB 00H,12H,12H,12H,36H,5AH,24H,00H	: smo-
19CA 40444A4C48	DFB 40H,44H,4AH,4CH,48H,48H,38H,00H	: smo-	225,0E1H,SARA-AIR		
187,0BBH,porpla			1AFC FC021C1030	DFB 0FCH,02H,1CH,10H,30H,50H,20H,00H	: smo-
19CC 0048544C54	DFB 00H,48H,54H,4CH,54H,54H,28H,00H	: smo-	226,0E2H,SARA-O		
188,0BCH,porpaung			1B04 1C22241030	DFB 1CH,22H,24H,10H,30H,50H,20H,00H	: smo-
19D4 4048544C54	DFB 40H,48H,54H,4CH,54H,54H,28H,00H	: smo-	227,0E3H,SARA-AJMAJMUAN		
189,0BDH,forta			1B0C 0A14101030	DFB 0AH,14H,10H,10H,30H,50H,20H,00H	: smo-
19DC 0042454654	DFB 00H,42H,45H,48H,54H,54H,28H,00H	: smo-	228,0E4H,SARA-AJMAJMALAI		
190,0BEH,porparn			1B14 0018242020	DFB 00H,18H,24H,20H,20H,20H,20H	: smo-229,0ESH,
19E4 4042454654	DFB 40H,42H,45H,48H,54H,54H,28H,00H	: smo-	1B1C 00142A2430	DFB 00H,14H,2AH,24H,30H,08H,08H,08H	: smo-
191,0BFH,lorhun			230,0E6H,MAJYAMOK		
:			1B24 8058A85800	DFB 80H,58H,0A8H,58H,00H,00H,00H,00H	: smo-
19EC 003844484C	DFB 00H,38H,44H,48H,4CH,4AH,4CH,00H	: smo-	231,0E7H,MAJ-TAIKOO		
192,0C0H,porampoa			1B2C 0040404000	DFB 00H,40H,40H,40H,00H,00H,00H,00H	: smo-
19F4 00444A4C4C	DFB 00H,44H,4AH,4CH,4CH,5AH,24H,00H	: smo-	232,0E8H,MAJ-EK		
183,0C1H,morma			1B34 0098503800	DFB 00H,98H,50H,38H,00H,00H,00H,00H	: smo-
19FC 00242A282C	DFB 00H,24H,2AH,26H,2CH,24H,1CH,00H	: smo-	233,0E9H,MAJ-TOR		
184,0C2H,yonyak			1B3C 80A8544800	DFB 80H,0A8H,54H,48H,00H,00H,00H,00H	: smo-
1A04 007C023C30	DFB 00H,7CH,02H,3CH,30H,28H,10H,00H	: smo-	234,0EAH,MAJ-TREE		
195,0C3H,rorru			1B44 0040E04000	DFB 00H,40H,0E0H,40H,00H,00H,00H,00H	: smo-
1A0C 001C222426	DFB 00H,1CH,22H,24H,26H,2AH,24H,20H	: smo-	235,0EBH,MAJ-JATTAVAR		
196,0C4H,loru			1B4C C030281000	DFB 0C0H,30H,28H,10H,00H,00H,00H,00H	: smo-
1A14 003C425C66	DFB 00H,3CH,42H,5CH,68H,6AH,24H,00H	: smo-	236,0ECH,KARAN		
197,0C5H,lorlong			1B54 0000000000	DFB 00H,00H,00H,00H,00H,40H,0A0H,40H	: smo-237,0EDH,
1A1C 001C222426	DFB 00H,1CH,22H,24H,26H,25H,22H,20H	: smo-	1B5C 0000000000	DFB 00H,00H,00H,00H,00H,80H,60H,20H	: smo-238,0EEH,
198,0C6H,loruu			1B64 00001C222A	DFB 00H,00H,1CH,22H,2AH,22H,1CH,00H	: smo-
1A24 007B444060	DFB 00H,78H,44H,40H,60H,50H,20H,00H	: smo-	239,0EFH,		
199,0C7H,vorvan			1B6C 003C424242	DFB 00H,3CH,42H,42H,42H,42H,3CH,00H	: smo-
1A2C 407C42526A	DFB 40H,7CH,42H,52H,6AH,5AH,4CH,00H	: smo-	240,0F0H,thal 0		
200,0C8H,soraalar			1B74 003C425A5E	DFB 00H,3CH,42H,5AH,5EH,40H,38H,00H	: smo-
1A34 00446AEC48	DFB 00H,44H,6AH,0ECH,48H,44H,78H,00H	: smo-	241,0F1H,thal 1		
201,0C9H,sornusi			1B7C 00546C554A	DFB 00H,54H,6CH,55H,4AH,42H,3CH,00H	: smo-
1A3C 403C425C66	DFB 40H,3CH,42H,5CH,66H,6AH,24H,00H	: smo-	242,0F2H,thal 2		
202,0CAH,soruua			1B84 006C8A5246	DFB 00H,6CH,6AH,52H,46H,4AH,24H,00H	: smo-
1A44 0022552654	DFB 00H,22H,55H,26H,54H,52H,5CH,00H	: smo-	243,0F3H,thal 3		
203,0CBH,horheep			1B9C 807C02122A	DFB 80H,7CH,02H,12H,2AH,1AH,7CH,00H	: smo-
1A4C A052654654	DFB 0A0H,52H,65H,46H,54H,54H,28H,00H	: smo-	244,0F4H,thal 4		
204,0CCH,lorjular			1B94 B07C02122A	DFB 0B0H,7CH,02H,12H,2AH,1AH,7CH,00H	: smo-
1A54 003C42444A	DFB 00H,3CH,42H,44H,4AH,46H,3CH,00H	: smo-	245,0F5H,thal 5		
205,0CDH,orang			1B9C 013E40444A	DFB 01H,3EH,40H,44H,4AH,46H,3CH,00H	: smo-
1A5C 803C5E4442	DFB 60H,3CH,5EH,44H,42H,46H,3CH,00H	: smo-	246,0F6H,thal 6		
206,0CEH,homkhook			1BA4 8094AA8A86	DFB 80H,94H,0AAH,6AH,66H,6AH,64H,00H	: smo-
1A64 002452524C	DFB 00H,24H,52H,52H,4CH,40H,40H,70H	: smo-	247,0F7H,thal 7		
207,0CFH,peiyamnoi			1BAC 807C020A2A	DFB 80H,7CH,02H,0AH,2AH,5AH,34H,00H	: smo-
:			248,0F8H,thal 8		
1A6C 00442A1C44	DFB 00H,44H,2AH,1CH,44H,2AH,1CH,00H	: smo-	1BB4 80543A1226	DFB 80H,54H,3AH,12H,26H,4AH,04H,00H	: smo-
208,0D0H,SARA-A			249,0F9H,thal 9		
1A74 0088543800	DFB 00H,88H,54H,38H,00H,00H,00H,00H	: smo-	1BBC 005A5D5650	DFB 00H,5AH,5DH,56H,50H,28H,14H,00H	: smo-
209,0D1H,MAIHANARGARD			250,0FAH,thal		
1A7C 0018242020	DFB 00H,18H,24H,20H,20H,20H,20H,00H	: smo-	1BC4 0000021556	DFB 00H,00H,02H,15H,56H,0AEH,00H,00H	: smo-
210,0D2H,SARA-AR			251,0FBH,thal		
1A84 031B242020	DFB 03H,18H,24H,20H,20H,20H,20H,00H	: smo-	1BCC 103070FF70	DFB 10H,30H,70H,0FFH,70H,30H,10H,00H	: smo-
211,0D3H,SARA-AM			252,0FCH,thal		
1A8C 0010287C00	DFB 00H,10H,28H,7CH,00H,00H,00H,00H	: smo-	1BD4 010307FF07	DFB 01H,03H,07H,0FFH,07H,03H,01H,00H	: smo-
212,0D4H,SARA-I			253,0FDH,thal		
1A94 4050687C00	DFB 40H,50H,68H,7CH,00H,00H,00H,00H	: smo-	1BDC 011C3E7F01	DFB 01H,1CH,3EH,7FH,01H,01H,01H,01H	: smo-
213,0D5H,SARA-E			254,0FEH,thal		
1A9C 0050A87C00	DFB 00H,50H,0A8H,7CH,00H,00H,00H,00H	: smo-	1BE4 010101017F	DFB 01H,01H,01H,01H,7FH,3EH,1CH,01H	: smo-
214,0D6H,SARA-AU			255,0FFH,thal		
1AA4 00A8B47E00	DFB 00H,0A8H,0B4H,7EH,00H,00H,00H,00H	: smo-			
215,0D7H,SARA-AUU					
1AAc 2050804000	DFB 20H,50H,60H,40H,00H,00H,00H,00H	: smo-			
216,0D8H,SARA-U					
1AB4 4854582000	DFB 48H,54H,58H,20H,00H,00H,00H,00H	: smo-	0000	END	
217,0D9H,SARA-UU			00E0 A	00D8 AC	1169 ADD1LN
1ABC 0000000000	DFB 00H,00H,00H,00H,00H,20H,50H,20H	: smo-218,0DAH,	1176 ADD1LN1	1177 ADD8LN	1184 ADD8LN1
1AC4 000000FF00	DFB 00H,00H,00H,0FFH,00H,00H,00H,00H	: smo-219,0DBH,	06CC ADDDGT	06D9 ADDDGT1	068D ADDWTHOLD
1ACC 080808FF00	DFB 08H,08H,08H,0FFH,00H,00H,00H,00H	: smo-	05F3 ADJUBUD	133B AZZ1	1365 AZZ10
220,0DCH,			136B AZZ11	1371 AZZ12	1377 AZZ13
1AD4 000000FF08	DFB 00H,00H,00H,0FFH,08H,08H,08H,08H	: smo-	137D AZZ14	1383 AZZ15	1389 AZZ16
221,0DDH,			138F AZZ17	1395 AZZ18	139B AZZ19
1ADC 080808FF08	DFB 08H,08H,08H,0FFH,08H,08H,08H,08H	: smo-	139F AZZ2	1383 AZZ22	13C7 AZZ22
222,0DEH,			13DB AZZ23	13E4 AZZ24	139E AZZ3
1AE4 8F66683EE6	DFB 0BFH,66H,68H,3EH,6EH,66H,3EH,01H	: smo-	1341 AZZ4	1347 AZZ5	134D AZZ6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1353 AZZ7	1359 AZZ8	135F AZZ9	008C PS	00D0 PSW	00B9 PTO
00F0 B	1000 BACKUP	05A7 BCDASC	008B PT1	0088 PX0	00BA PX1
0591 BEEP	001D BUF_CURP	001C BUF_DIST	08DA QADDGT	0649 QCHKPRESS	02AB QCHRSTADJ
0029 BUF_INDEXH	0028 BUF_INDEXL	0027 BUF_ROUTEH	01E4 QDECBDC	080B QDSP_MSG	065D QFINDCOL
0026 BUF_ROUTEL	0018 BUF_VAL1	0019 BUF_VAL2	0653 QFINDROW	0350 QIN2BCD	0541 QPROGCARD
001A BUF_VAL3	03A1 CARDPRO	0542 CHECKDATA	02B9 QPROGDATE	0320 QPROGTIME	0629 QSCANKEY
03A0 CHKCRD	080C CHKDIST	0633 CHKPRESS	01FD QTIMESET	0185 QWOERKNORM	058A RAMCLR
065E CHKRELEASE	028A CHRISTADJ	0298 CHRISTADJ1	059D RAMCLR1	009A R88	0FBF RD8BIT
102A CLRLN0	1038 CLRLN01	1036 CLRLN02	0F91 RD8BIT1	05F7 RDOPOST	0E6F RDTIME
1048 CLRLN1	1056 CLRLN11	1054 CLRLN12	07A5 RDWRVAL	0ECA READEEP	009C REN
1066 CLRLN20	1074 CLRLN21	1072 CLRLN22	0098 RI	0C49 ROUTETABF	0C88 ROUTETABR
100A CLRSCRN	1018 CLRSCRN1	1016 CLRSCRN2	00D3 RS0	00D4 RS1	0092 RSTRTC
0674 CLRVAL	A001 CNTLCM	10CB CONV1	0EC5 RTCDL	0E9A RTCRD	0EAF RTCRD1
10DE CONV2	10D3 CONV3	10DB CONV4	0E43 RTCWR	0E5D RTCWRC	0E5F RTCWRC1
0000 CROVALID	0093 CSEEP	10B7 CURBOTM	0099 SBUF	0609 SCANKEY	0094 SCLKEEP
0021 CURCOL	10C0 CURCONV	109E CURHI	0090 SCLKRTC	0098 SCON	0015 SECOND
10AC CURHOME	1090 CURLOW	07D9 CUROFF	0592 SENDBYTE	0E19 SETDATE	0DEF SETTIME
07D0 CURON	1002 CURRENTP	0020 CURROW	070A SHOWCOMPL	05C1 SHOWDATE	0720 SHOWEDIT
00A2 CWORD1	0088 CWORD2	00D7 CY	078D SHOWEDIT1	078C SHOWEDIT2	05E3 SHOWTIME
0012 DATE	A000 DATLCM	082A DEBOUNCE	06DC SHOWWAIT	0437 SHW_CONFIRM	01AF SHW_DISTPRO
01D2 DECBDC	0669 DEFKEY	07C7 DELAYP	0546 SHW_LOGO	00AA SHW_MMENU	0186 SHW_PROCESS
0095 DIEEP	05B5 DISPBCD2	0436 DISPRATE	03ED SHW_RMENU	03AA SHW_ROUTE	018D SHW_TIMEPRO
1001 DISTANCE	0FAC DLCLK	0096 DOEEP	0351 SHW_TIMESET	009F SMO	009E SM1
0083 DPH	0082 DPL	07E5 DSP_MSG	009D SM2	0081 SP	03A2 SPEAKPOS
00AF EA	0FA3 EEPDL	0F1A ERALEEP	0000 START	0040 START1	114D SUB4LN
0E2F ERASEEEP	00AC ES	00A9 ET0	115A SUB4LN1	115B SUB8LN	1168 SUB8LN1
00AB ET1	0F3B EWDSSEEP	0F2C EWENEPP	0098 TB8	0088 TCON	0004 TENVAL
00A8 EX0	00AA EX1	00D5 F0	008D TF0	008F TF1	008C TH0
0071 FGCAR	0072 FGFOR	0070 FGPAE0	008D TH1	13EC THAIPT	0099 TI
0654 FINDCOL	064A FINDROW	0011 HOUR	01E5 TIMESET	01F4 TIMESET_1	008A TLO
0005 HUNDVAL	0089 IE0	008B IE1	008B TL1	0089 TMOD	0016 TMP_BUF0
00A8 IEC	0321 IN2BCD	0C34 INDEXF	0017 TMP_BUF1	008C TR0	008E TR1
001B INDEXPOS	0C73 INDEXR	004A INICNT	0003 UNITVAL	0001 UPCODE	0007 UPHOUR
0FB1 INITLCD	0064 INITPAR	005C INITSER	0006 UPMN	0002 UPPOS	00A1 WAITKEY
0091 IORTC	0088 IPC	0088 ITO	00F3 WORKNORM	00FF WORKNORM0	0119 WORKNORM1
008A IT1	082D KEYTAB	FEB6 KILOP	0133 WORKNORM2	0136 WORKNORM3	014A WORKNORM31
0638 LPCHKPRESS	062E LPDEBOUNCE	07CB LPDELAYP	0150 WORKNORM32	0153 WORKNORM4	018A WORKNORM41
07E8 LPDSP_MSG	0FA7 LPPEPDL	0657 LPFINDCOL	0173 WORKNORM42	017C WORKNORM43	1084 WR18YTE
064D LPFINDROW	033A LPIN2BCD	0080 LPMAIN	0F56 WR3BIT	0F58 WR3BIT1	0F68 WR7BIT
04AA LPPROGCARD	0277 LPPROGDATE1	030E LPPROGTIME1	0F6B WR7BIT1	0F7B WR8BIT	0FD7 WR8BIT1
01E8 LPTIMESET	00F9 LPWORKNORM	007A MAIN	0F04 WRALEEP	10FF WRCHR	1122 WRCHR1
0086 MAIN0	008F MAIN1	0098 MAIN2	121B WREKTOW	1335 WRFRMBUF	11F6 WR81
0010 MMINUTE	0013 MONTH	0D29 MSG_ANUS	1241 WR12	11D0 WR1EE	0E0C WR1EEEP
0DEB MSG_BAHT	0CC6 MSG_BANG	0DC6 MSG_BAROM	0F4A WROP	1266 WR1DAT	126B WR1DAT1
0CCD MSG BUMIPOL	0A93 MSG_CHOICE1	0A88 MSG_CHOICE2	1268 WR1DAT2	1185 WRU	11AB WRU1
0AD8 MSG_CHOICE3	0A4D MSG_COMPL	0B64 MSG_CONFIRM1	0014 YEAR	0C89 ZONERATE	0CB2 ZONETABF
0B89 MSG_CONFIRM2	0BAD MSG_CONFIRM3	0AF8 MSG_DATE	1276 ZZ1	12A0 ZZ10	12A6 ZZ11
0B2D MSG_DATEF	0B52 MSG_DISTP	0BF9 MSG_DOWN1	12AC ZZ12	12B2 ZZ13	12B8 ZZ14
0C10 MSG_DOWN2	0C1E MSG_DOWN3	0988 MSG_EDIT1	12BE ZZ15	12C4 ZZ16	12CA ZZ17
09CF MSG_EDIT2	09E8 MSG_EDIT3	0A70 MSG_ERR	12D0 ZZ18	12D6 ZZ19	12E0 ZZ2
0D52 MSG_HIGHWAY	0CF3 MSG_KASET	0D7B MSG_KASUANG	12F6 ZZ22	130C ZZ222	1322 ZZ223
0CFF MSG_LADPRAO	0CE1 MSG_LAKSI	083D MSG_LOGO1	132B ZZ224	12D9 ZZ3	127C ZZ4
085D MSG_LOGO2	087D MSG_LOGO3	089D MSG_MAIN1	1282 ZZ5	1288 ZZ6	128E ZZ7
08B7 MSG_MAIN2	08D5 MSG_MAIN3	0D0E MSG_MOHCHIT	1294 ZZ8	129A ZZ9	
0D9E MSG_PACHA	0D91 MSG_PANFA	0DB4 MSG_PAPIN			
0D6B MSG_PARUP	0D3E MSG_PAYA	0953 MSG_RMENU1			
0972 MSG_RMENU2	0995 MSG_RMENU3	08F3 MSG_ROUTE1			
0911 MSG_ROUTE2	0934 MSG_ROUTE3	0CBF MSG_RUNG			
0DD6 MSG_SAITAI	0CD7 MSG_SAPAN	0D1F MSG_SBUFF			
0B0B MSG_SETTM	0D46 MSG_SRIAYUT	0B02 MSG_TIME			
0B43 MSG_TIMEF	0BCB MSG_UP1	0BE3 MSG_UP2			
0A07 MSG_WAIT1	0A2A MSG_WAIT2	0D5D MSG_WATBEN			
0628 NOPRESS	01DB NXTDECBDC	01CE NXTDISTPRO			
032C NXTIN2BCD	04AD NXTPROGCARD1	04B3 NXTPROGCARD11			
048C NXTPROGCARD2	04E7 NXTPROGCARD3	015E NXTWORKNORM4			
00D2 OV	00D0 P	0080 P0			
0090 P1	0090 P1.0	0091 P1.1			
0092 P1.2	0093 P1.3	0097 P1.7			
00A0 P2	00B0 P3	00B0 P3.0			
00B1 P3.1	00B2 P3.2	00B3 P3.3			
00B4 P3.4	00B5 P3.5	00B6 P3.6			
0087 PCON	0022 POSHI	0023 POSLO			
0025 POS_H	0024 POS_L	8000 PPI1A			
8001 PPI1B	8002 PPI1C	8003 PPI1CO			
9000 PPI2A	9001 PPI2B	9002 PPI2C			
9003 PPI2CO	03A3 PRNTICK	01FE PROGALLTIME			
0480 PROGCARD	0205 PROGDATE	0283 PROGDATE1			
0529 PROGERR	02AC PROGTIME	031A PROGTIME1			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค.  
รายละเอียดโปรแกรมส่วนแสดงผลจุดเมตริกซ์

```

;DISPLAY DOT MATRIX 120 X 80
0000 CPU "8051.TBL"
0000 HOF "INT8"
0000 INCL "HEADS1.H"

```

;MCS-51 INTERNAL REGISTERS

```

00F0 = B: EQU 0F0H ;B REGISTER
00E0 = A: EQU 0E0H ;ACCUMULATOR
00D0 = PSW: EQU 0D0H ;PROGRAM STATUS WORD
00B8 = IPC: EQU 0B8H ;INTERRUPT PRIORITY
00B0 = P3: EQU 0B0H ;PORT 3
00B0 = P3.0: EQU 0B0H
00B1 = P3.1: EQU 0B1H
00B2 = P3.2: EQU 0B2H
00B3 = P3.3: EQU 0B3H
00B4 = P3.4: EQU 0B4H
00B5 = P3.5: EQU 0B5H
00B6 = P3.6: EQU 0B6H
00A8 = IEC: EQU 0A8H ;INTERRUPT ENABLE
00A0 = P2: EQU 0A0H ;PORT 2
0099 = SBUF: EQU 99H ;SEND BUFFER
0098 = SCON: EQU 98H ;SERIAL CONTROL
0090 = P1: EQU 90H
0090 = P1.0: EQU 90H ;PORT 1
0091 = P1.1: EQU 91H
0092 = P1.2: EQU 92H
0093 = P1.3: EQU 93H
0097 = P1.7: EQU 97H

008D = TH1: EQU 8DH ;TIMER 1 HIGH
008C = TH0: EQU 8CH ;TIMER 0 HIGH
008B = TL1: EQU 8BH ;TIMER 1 LOW
008A = TL0: EQU 8AH ;TIMER 0 LOW
0089 = TMOD: EQU 89H ;TIMER MODE
0088 = TCON: EQU 88H ;TIMER CONTROL
0087 = PCON: EQU 87H ;POWER CONTROL REGISTER
0083 = DPH: EQU 83H ;DATA POINTER HIGH
0082 = DPL: EQU 82H ;DATA POINTER LOW
0081 = SP: EQU 81H ;STACK POINTER
0080 = P0: EQU 80H ;PORT 0

```

;MCS-51 INTERNAL BIT ADDRESSES

```

00D7 = CY: EQU 0D7H ;CARRY FLAG
00D6 = AC: EQU 0D6H ;AUXILIARY-CARRY FLAG
00D5 = F0: EQU 0D5H ;USER FLAG 0
00D4 = RS1: EQU 0D4H ;REGISTER SELECT MSB
00D3 = RS0: EQU 0D3H ;REGISTER SELECT LSB
00D2 = OV: EQU 0D2H ;OVERFLOW FLAG
00D0 = P: EQU 0D0H ;PARITY FLAG
00BC = PS: EQU 0BCH ;PRIORITY SERIAL PORT
00BB = PT1: EQU 0BBH ;PRIORITY TIMER 1
00BA = PX1: EQU 0BAH ;PRIORITY EXTERNAL 1
00B9 = PT0: EQU 0B9H ;PRIORITY TIMER 0
00B8 = PX0: EQU 0B8H ;PRIORITY EXTERNAL 0
00AF = EA: EQU 0AFH ;ENABLE ALL INTERRUPT
00AC = ES: EQU 0ACH ;ENABLE SERIAL INTERRUPT
00AB = ET1: EQU 0ABH ;ENABLE TIMER 1 INTERRUPT
00AA = EX1: EQU 0AAH ;ENABLE EXTERNAL 1 INTERR
00A9 = ET0: EQU 0A9H ;ENABLE TIMER 0 INTERRUPT
00A8 = EX0: EQU 0A8H ;ENABLE EXTERNAL 0 INTERR
009F = SM0: EQU 09FH ;SERIAL MODE 0
009E = SM1: EQU 09EH ;SERIAL MODE 1
009D = SM2: EQU 09DH ;SERIAL MODE 2
009C = REN: EQU 09CH ;SERIAL RECEPTION ENABLE
009B = TB8: EQU 09BH ;TRANSMIT BIT 8
009A = RB8: EQU 09AH ;RECEIVE BIT 8
0099 = TI: EQU 099H ;TRANSMIT INTERRUPT FLAG
0098 = RI: EQU 098H ;RECEIVE INTERRUPT FLAG
009F = TF1: EQU 0BFH ;TIMER 1 OVERFLOW FLAG
009E = TR1: EQU 0BEH ;TIMER 1 RUN CONTROL BIT
009D = TF0: EQU 0BDH ;TIMER 0 OVERFLOW FLAG
009C = TR0: EQU 0BDH ;TIMER 0 RUN CONTROL BIT
009B = IE1: EQU 0BBH ;EXT INTERR. 1 EDGE FLAG
009A = IT1: EQU 0BAH ;EXT INTERR. 1 TYPE FLAG
0099 = IE0: EQU 0B9H ;EXT INTERR. 0 EDGE FLAG
0098 = IT0: EQU 0B8H ;EXT INTERR. 0 TYPE FLAG

```

;HARDWARE DEFINE

```

8003 = PPI1CON: EQU 8003H
8000 = CDATA1: EQU 8000H ;COLUMN DATA
8001 = CDATA2: EQU 8001H
8002 = CDATA3: EQU 8002H

```

```

9003 = PPI2CON: EQU 9003H
9000 = CDATA: EQU 9000H
9001 = RDATA: EQU 9001H

```

;VARIABLE DEFINE

```

0022 = BUF_MAP: EQU 22H ;RESERVED 8 BYTE
002A = BUF_CNT: EQU BUF_MAP+8
002B = BUF_STORE: EQU BUF_CNT+1 ;RESERVE 16 BYTE
0058 = LINE: EQU BUF_STORE+48 ;reserve 32 byte

0010 = CURCOL: EQU 10H
0011 = COUNT1: EQU CURCOL+1
0012 = BYTE: EQU COUNT1+1
0013 = PATTERN: EQU BYTE+1

```

;EXTERNAL DATA BUFFER

```

0000 = BUF_DSP: EQU 0000H
4000 = BUF_DAT: EQU 4000H

```

;BIT ADDRESS AREA

```

0000 = SAVEC: EQU 00H
0001 = FG_SARA: EQU SAVEC+1
0002 = FG_NEW: EQU FG_SARA+1

```

```

0000 ORG 0000H
0000 020050 LJMP START

0023 ORG 0023H
0023 0204A0 LJMP RECVDAT

0050 ORG 0050H
0050 START:
0050 D8FE DJNZ R0,$
0052 D9FC DJNZ R1,START ;DELAY FOR WAKE UP HARDWARE

0054 758160 MOV SP,#60H

;INITIAL 8255 ALL OUTPUT PORT
INIT8255:
0057 909003 MOV DPTR,#PPI2CON
005A 7480 MOV A,#80H
005C F0MOVX @DPTR,A

```

```

005D 908003 MOV DPTR,#PPI1CON
0060 7480 MOV A,#80H
0062 F0MOVX @DPTR,A
0063 C201 CLR FG_SARA
0065 C202 CLR FG_NEW

;INITIAL SERIAL INTERRUPT
0067 758920 MOV TMOD,#20H ;MODE2

```

;INITIAL SERIAL PORT

```

006A INITSER:
006A 759852 MOV SCON,#52H ;ENABLE RECIEVE SERIAL PORT
MODE 1
006D 758DFD MOV TH1,#0FDH ;BUAD RATE 9600
0070 D28E SETB TR1 ;START TIMER1 TO GENERATE
BUAD RATE
0072 7443 MOV A,#FC ;CHECK CONNECTION
0074 120487 LCALL SNOBYTE
0077 12040D LCALL DELAY3
007A C299 CLR TI ;CLEAR FLAG
007C D2AC SETB ES ;ENABLE SERIAL
007E D2AF SETB EA

```

```

0080 751331 MOV PATTERN,#1* ;DISPLAY STATIC AND MOVING
0083 120493 LCALL SHWLOGO
0086 1203A7 LCALL CLRDRSPR ;CLEAR DISPLAY RAM
0089 MAIN:
0089 E513 MOV A,PATTERN
008B B43011 CJNE A,#0,MAIN1 ;DISPLAY STATIC
008E 300208 JNB FG_NEW,NXTMAIN0
0091 C202 CLR FG_NEW
0093 751000 MOV CURCOL,#00
0096 120114 LCALL WRSTR
0099 NXTMAIN0:
0099 120272 LCALL DSPSTAT
009C 020089 LJMP MAIN
009F MAIN1:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

009F B43106 CJNE A,#1,MAIN2 ;DISPLAY MOVING LEFT AND STATIC 012B 22 RET
TEMP
00A2 1200C0 LCALL SHWMSG1
00A5 020089 LJM MAIN ;ROUTINE FOR WRITE ONE CHARACTER
00A8 MAIN2: ;OR ENGLISH IN CURSOR POSITION
00A8 B43206 CJNE A,#2,MAIN3 ;DISPLAY MOVING LEFT ;INPUT :CURCOL(0 - 15) A (ASCII)
00AB 1200DC LCALL SHWMSGM ;OUTPUT: CURCOL +1
00AE 020089 LJM MAIN
00B1 MAIN3:
00B1 B43306 CJNE A,#3,MAIN4
00B4 1200F8 LCALL SHWMSG5
00B7 020089 LJM MAIN
00BA MAIN4:
00BA 751330 MOV PATTERN,#0
00BD 020089 LJM MAIN

;ROUTINE FOR SHOW LONG MESSAGE FROM RIGHT TO LEFT HAND
;ABOUT 48 CHARACTER
00C0 SHWMSG1:
00C0 751000 MOV CURCOL,#0
00C3 120114 LCALL WRSTR
00C6 751120 MOV COUNT1,#20H
00C9 LPSHWMSG1:
00C9 120272 LCALL DSPSTAT
00CC D511FA DJNZ COUNT1,LPSHWMSG1
00CF 751000 MOV CURCOL,#0
00D2 75115A MOV COUNT1,#5AH
00D5 LPSHWMSG2:
00D5 120293 LCALL DSPFLT
00D8 D511FA DJNZ COUNT1,LPSHWMSG2
00DB 22RET

;ROUTINE FOR SHOW MEDIUM MESSAGE FROM RIGHT TO LEFT
HAND
;ABOUT 32 CHARACTER
00DC SHWMSGM:
00DC 751000 MOV CURCOL,#0
00DF 120114 LCALL WRSTR
00E2 751120 MOV COUNT1,#20H
00E5 LPSHWMSGM1:
00E5 120272 LCALL DSPSTAT
00E8 D511FA DJNZ COUNT1,LPSHWMSGM1
00EB 751000 MOV CURCOL,#0
00EE 751140 MOV COUNT1,#40H
00F1 LPSHWMSGM2:
00F1 120293 LCALL DSPFLT
00F4 D511FA DJNZ COUNT1,LPSHWMSGM2
00F7 22 RET

;ROUTINE FOR SHOW SHORT MESSAGE FROM RIGHT TO LEFT HAND
;ABOUT 16 CHARACTER
00F8 SHWMSG5:
00F8 751000 MOV CURCOL,#0
00FB 120114 LCALL WRSTR
00FE 751120 MOV COUNT1,#20H
0101 LPSHWMSG51:
0101 120272 LCALL DSPSTAT
0104 D511FA DJNZ COUNT1,LPSHWMSG51
0107 751000 MOV CURCOL,#0
010A 751120 MOV COUNT1,#20H
010D LPSHWMSG52:
010D 120293 LCALL DSPFLT
0110 D511FA DJNZ COUNT1,LPSHWMSG52
0113 22 RET

;ROUTINE FOR WRITE STRING TO MEMORY AT CURSOR POSITION
0114 WRSTR:
0114 C0E0 PUSH A
0118 1203A7 LCALL CLRDSRP
0119 904000 MOV DPTR,#BUF_DAT
011C LPWRSTR1:
011C E4CLR A
011D E0MOVX A,@DPTR
011E B40402 CJNE A,#04,WRSTR1
0121 8006 SJMP QWRSTR
0123 WRSTR1:
0123 12012C LCALL WRCHR
0128 A3 INC DPTR
0127 80F3 SJMP LPWRSTR1
0129 QWRSTR:
0129 D0E0 POP A

012C WRCHR:
012C C0E0 PUSH A
012E C083 PUSH DPH
0130 C082 PUSH DPL

0132 FA MOV R2,A ;SAVE ASCII
0133 9004E4 MOV DPTR,#THAIPAT
0138 75F008 MOV B,#8
0139 A4 MUL AB
013A 2582 ADD A,DPL
013C F582 MOV DPL,A
013E E5F0 MOV A,B
0140 3583 ADDC A,DPH
0142 F583 MOV DPH,A ;DPTR - START TABLE

0144 E510 MOV A,CURCOL
0148 C3 CLR C
0147 8430 SUBB A,#48
0149 5004 JNC NXTWRCHR1 ;CHECK CURCOL OVER 15 OR NOT
0148 2430 ADD A,#48
014D F510 MOV CURCOL,A
014F NXTWRCHR1:
014F F510 MOV CURCOL,A
0151 7822 MOV R0,#BUF_MAP ;GET CHARACTER DATA MAP
0153 7908 MOV R1,#8
0155 LPWRCHR1:
0155 7400 MOV A,#00
0157 93 MOVX A,@A+DPTR ;SAVE MAP
0158 F8 MOV @R0,A
0159 08 INC R0
015A A3 INC DPTR
0158 D9F8 DJNZ R1,LPWRCHR1
015D EA MOV A,R2 ;RESTORE ASCII
015E B4D803 CJNE A,#*,SARA_UU ;SARA_U
0161 020248 LJMP LEVEL4
0164 SARA_UU:
0164 B4D903 CJNE A,#*,SARA_IE ;SARA_UU
0167 020248 LJMP LEVEL4
016A SARA_IE:
016A B4D405 CJNE A,#*,SARA_IEE ;SARA_IE
016D D201 SETB FG_SARA
016F 0201F9 LJMP LEVEL2
0172 SARA_IEE:
0172 B4D505 CJNE A,#*,SARA_UG
0175 D201 SETB FG_SARA
0177 0201F9 LJMP LEVEL2 ;SARA_IEE
017A SARA_UG:
017A B4D605 CJNE A,#*,SARA_UUH ;SARA_UG
017D D201 SETB FG_SARA
017F 0201F9 LJMP LEVEL2
0182 SARA_UUH:
0182 B4D705 CJNE A,#*,MAIHAN
0185 D201 SETB FG_SARA
0187 0201F9 LJMP LEVEL2 ;SARA_UUH
018A MAIHAN:
018A B4D105 CJNE A,#*,KARAN
018D D201 SETB FG_SARA ;MAIHAN
018F 0201F9 LJMP LEVEL2
0192 KARAN:
0192 B4EC03 CJNE A,#*,MAIEK
0195 0201F9 LJMP LEVEL2 ;KARAN
0198 MAIEK:
0198 B4E809 CJNE A,#*,MAITOE
019B 300103 JNB FG_SARA,MAIEK1 ;MAIEK
019E 0201D1 LJMP LEVEL1
01A1 MAIEK1:
01A1 0201F9 LJMP LEVEL2
01A4 MAITOE:
01A4 B4E909 CJNE A,#*,MAITEE
01A7 300103 JNB FG_SARA,MAITOE1 ;MAITOE
01AA 0201D1 LJMP LEVEL1
01AD MAITOE1:
01AD 0201F9 LJMP LEVEL2
01B0 MAITEE:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

01B0 B4EA09 CJNE A,#*,MAIJATAWA
01B3 300103 JNB FG_SARA,MAITEE1 ;MAITEE
01B6 0201D1 LJMP LEVEL1
01B9 MAITEE1:
01B9 0201F9 LJMP LEVEL2
01BC MAIJATAWA:
01BC B4EB09 CJNE A,#*,MAITAIKU
01BF 300103 JNB FG_SARA,MAIJATAWA1 ;MAIJATAWA
01C2 0201D1 LJMP LEVEL1
01C5 MAIJATAWA1:
01C5 0201F9 LJMP LEVEL2
01C8 MAITAIKU:
01C8 B4E703 CJNE A,#*,PAYANCHANA
01CB 0201F9 LJMP LEVEL2
01CE PAYANCHANA:
01CE 02021F LJMP LEVEL3

;MANAGE SARA BON SOONG
01D1 LEVEL1:
01D1 900000 MOV DPTR,#BUF_DSP ;FIRST ADDRESS OF SARA BON
SOONG
01D4 E510 MOV A,CURCOL
01D6 14 DEC A
01D7 2582 ADD A,DPL
01D9 F582 MOV DPL,A
01DB E4 CLR A
01DC 3583 ADDC A,DPH
01DE F583 MOV DPH,A ;FIND FIRST POSITON OF
CURSOR
01E0 7822 MOV R0,#BUF_MAP
01E2 7904 MOV R1,#4 ;GET 4 BYTE UPPER
01E4 LLEVEL1
01E4 E6 MOV A,@R0
01E5 F0 MOVX @DPTR,A ;SAVE IN DISPLAY RAM
01E8 08 INC R0 ;DPTR + 48
01E7 7430 MOV A,#48
01E9 2582 ADD A,DPL
01EB F582 MOV DPL,A
01ED E4 CLR A
01EE 3583 ADDC A,DPH
01F0 F583 MOV DPH,A
01F2 D9F0 DJNZ R1,LLEVEL1
01F4 C201 CLR FG_SARA
01F6 02026B LJMP QWRCHR

;MANAGE SARA LANG (SARA U ,SARA UU)
0248 LEVEL4:
0248 900300 MOV DPTR,#BUF_DSP+768 ;FIRST ADDRESS OF
SARA LANG
024B E510 MOV A,CURCOL
024D 14 DEC A
024E 2582 ADD A,DPL
0250 F582 MOV DPL,A
0252 E4 CLR A
0253 3583 ADDC A,DPH
0255 F583 MOV DPH,A ;FIND FIRST POSITON OF
CURSOR
0257 7822 MOV R0,#BUF_MAP
0259 7904 MOV R1,#4 ;GET 4 BYTE LOWER
025B LLEVEL4:
025B E6 MOV A,@R0
025C F0 MOVX @DPTR,A ;SAVE IN DISPLAY RAM
025D 08 INC R0 ;DPTR + 48
025E 7430 MOV A,#48
0260 2582 ADD A,DPL
0262 F582 MOV DPL,A
0264 E4 CLR A
0265 3583 ADDC A,DPH
0267 F583 MOV DPH,A
0269 D9F0 DJNZ R1,LLEVEL4
026B QWRCHR:
026B D082 POP DPL
026D D083 POP DPH
026F D0E0 POP A
0271 22 RET

;MANAGE SARA BON
01F9 LEVEL2:
01F9 9000C0 MOV DPTR,#BUF_DSP+192 ;FIRST ADDRESS OF SARA
BON
01FC E510 MOV A,CURCOL
01FE 14 DEC A
01FF 2582 ADD A,DPL
0201 F582 MOV DPL,A
0203 E4 CLR A
0204 3583 ADDC A,DPH
0206 F583 MOV DPH,A ;FIND FIRST POSITON OF
CURSOR
0208 7822 MOV R0,#BUF_MAP
020A 7904 MOV R1,#4 ;GET 4 BYTE UPPER
020C LLEVEL2:
020C E6 MOV A,@R0
020D F0 MOVX @DPTR,A ;SAVE IN DISPLAY RAM
020E 08 INC R0 ;DPTR + 48
020F 7430 MOV A,#48
0211 2582 ADD A,DPL
0213 F582 MOV DPL,A
0215 E4 CLR A
0216 3583 ADDC A,DPH
0218 F583 MOV DPH,A
021A D9F0 DJNZ R1,LLEVEL2
021C 02026B LJMP QWRCHR

;MANAGE PAYANCHANA
021F LEVEL3:
021F 900180 MOV DPTR,#BUF_DSP+384 ;FIRST ADDRESS OF
PAYANCHANA
0222 E510 MOV A,CURCOL
0224 2582 ADD A,DPL
0226 F582 MOV DPL,A
0228 E4 CLR A
0229 3583 ADDC A,DPH
022B F583 MOV DPH,A ;FIND FIRST POSITON OF
CURSOR

022D 7822 MOV R0,#BUF_MAP
022F 7908 MOV R1,#8 ;GET 8 BYTE
0231 LLEVEL3:
0231 E6 MOV A,@R0
0232 F0 MOVX @DPTR,A ;SAVE IN DISPLAY RAM
0233 08 INC R0
0234 7430 MOV A,#48 ;DPTR + 48
0236 2582 ADD A,DPL
0238 F582 MOV DPL,A
023A E4 CLR A
023B 3583 ADDC A,DPH
023D F583 MOV DPH,A
023F D9F0 DJNZ R1,LLEVEL3
0241 0510 INC CURCOL
0243 C201 CLR FG_SARA
0245 02026B LJMP QWRCHR

0248 LEVEL4:
0248 900300 MOV DPTR,#BUF_DSP+768 ;FIRST ADDRESS OF
SARA LANG
024B E510 MOV A,CURCOL
024D 14 DEC A
024E 2582 ADD A,DPL
0250 F582 MOV DPL,A
0252 E4 CLR A
0253 3583 ADDC A,DPH
0255 F583 MOV DPH,A ;FIND FIRST POSITON OF
CURSOR
0257 7822 MOV R0,#BUF_MAP
0259 7904 MOV R1,#4 ;GET 4 BYTE LOWER
025B LLEVEL4:
025B E6 MOV A,@R0
025C F0 MOVX @DPTR,A ;SAVE IN DISPLAY RAM
025D 08 INC R0 ;DPTR + 48
025E 7430 MOV A,#48
0260 2582 ADD A,DPL
0262 F582 MOV DPL,A
0264 E4 CLR A
0265 3583 ADDC A,DPH
0267 F583 MOV DPH,A
0269 D9F0 DJNZ R1,LLEVEL4
026B QWRCHR:
026B D082 POP DPL
026D D083 POP DPH
026F D0E0 POP A
0271 22 RET

;ROUTINE FOR DISPLAY STATIC
0272 DSPSTAT:
0272 C0E0 PUSH A
0274 7B04 MOV R3,#4
0276 LPDSPSTAT0:
0276 755B00 MOV LINE,#0
0279 7C14 MOV R4,#20 ;LINE COUNTER
027B LPDSPSTAT1:
027B 120322 LCALL LOBUF ;LOAD DATA 16 BYTE TO INTERNAL
BUFFER
027E 12034C LCALL DSPLN
0281 1203FF LCALL DELAY1
0284 1202DF LCALL OFFDSP
0287 055B INC LINE
0289 DCF0 DJNZ R4,LPDSPSTAT1
028B 1202DF LCALL OFFDSP
028E DBE6 DJNZ R3,LPDSPSTAT0
0290 D0E0 POP A
0292 22 RET

;ROUTINE FOR DISPLAY SHIIFT LEFT
0293 DSPLFT:
0293 7B04 MOV R3,#4
0295 LPDSPPLFT0:
0295 755B00 MOV LINE,#0
0298 7C14 MOV R4,#20 ;LINE COUNTER
029A LPDSPPLFT1:
029A 120322 LCALL LOBUF ;LOAD DATA 16 BYTE TO INTERNAL
BUFFER
029D 12034C LCALL DSPLN
02A0 1203FF LCALL DELAY1
02A3 1202DF LCALL OFFDSP
02A6 055B INC LINE
02A8 DCF0 DJNZ R4,LPDSPPLFT1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

02AA 909001  MOV  DPTR,#RDATA
02AD 7414    MOV  A,#20      ;OFF LED
02AF F0 MOVX @DPTR,A

02B0 1202FE  LCALL SHFTL
02B3 120404  LCALL DELAY2
02B6 DBDD    DJNZ  R3,LPDSPFLT0
02B8 22 RET

;ROUTINE FOR DISPLAY SHIFT RIGHT
02B9 DSPRGT:
02B9 7B04    MOV  R3,#4
02BB LPDSPRGT0:
02BB 755B00   MOV  LINE,#0
02BE 7C14    MOV  R4,#20      ;LINE COUNTER
02C0 LPDSPRGT1:
02C0 120322  LCALL LDBUF      ;LOAD DATA 16 BYTE TO INTERNAL
BUFFER
02C3 12034C  LCALL DSPLN
02C6 1203FF  LCALL DELAY1
02C9 1202DF  LCALL OFFDSP
02CC 055B    INC  LINE
02CE DCF0    DJNZ  R4,LPDSPRGT1

02D0 909001  MOV  DPTR,#RDATA
02D3 7414    MOV  A,#20      ;OFF LED
02D5 F0 MOVX @DPTR,A

02D6 120404  LCALL DELAY2
02D9 1202EA  LCALL SHFTR
02DC DBDD    DJNZ  R3,LPDSPRGT0
02DE 22 RET

;OFF DISPLAY
02DF OFFDSP:
02DF C0E0    PUSH  A
02E1 909001  MOV  DPTR,#RDATA
02E4 7414    MOV  A,#20      ;OFF LED
02E6 F0 MOVX @DPTR,A
02E7 D0E0    POP  A
02E9 22 RET

;ROUTINE FOR SHIFT DATA TO RIGHT
02EA SHFTR:
02EA 7C30    MOV  R4,#48      ;48 DIGIT
02EC 7D14    MOV  R5,#20      ;20 LINE
02EE 900000  MOV  DPTR,#BUF_DSP
02F1 C3 CLR  C
02F2 LPSHFTR0:
02F2 E0 MOVX A,@DPTR ;READ OUT
02F3 33 RLC  A
02F4 F0 MOVX @DPTR,A ;SAVE BACK

02F5 A3 INC  DPTR
02F6 DCFA   DJNZ  R4,LPSHFTR0
02F8 7C30    MOV  R4,#48
02FA C3 CLR  C
02FB DDF5   DJNZ  R5,LPSHFTR0
02FD 22 RET

;ROUTINE FOR SHIFT DATA TO LEFT
02FE SHFTL:
02FE 7C30    MOV  R4,#48      ;48 DIGIT
0300 7D14    MOV  R5,#20      ;20 LINE
0302 9003BF  MOV  DPTR,#BUF_DSP+959
0305 C3 CLR  C
0306 LPSHFTL0:
0306 E0 MOVX A,@DPTR ;READ OUT
0307 13 RRC  A
0308 F0 MOVX @DPTR,A ;SAVE BACK

0309 9200    MOV  SAVEC,C
030B C3 CLR  C
030C E582    MOV  A,DPL      ;DECREMENT DPTR 1
030E 9401    SUBB A,#1
0310 F582    MOV  DPL,A
0312 E583    MOV  A,DPH
0314 9400    SUBB A,#0
0316 F583    MOV  DPH,A
0318 A200    MOV  C,SAVEC

031A DCEA    DJNZ  R4,LPSHFTL0
031C 7C30    MOV  R4,#48
031E C3 CLR  C
031F DDE5    DJNZ  R5,LPSHFTL0
0321 22 RET

;ROUTINE FOR LOAD DISPLAY DATA FROM EXTERNAL TO INTERNAL
MEMORY
;INPUT : LINE
;REG: R0,R1
0322 LDBUF:
0322 C083    PUSH  DPH
0324 C082    PUSH  DPL
0326 C0E0    PUSH  A

0328 782B    MOV  R0,#BUF_STORE
032A 900000  MOV  DPTR,#BUF_DSP
032D E55B    MOV  A,LINE
032F 75F030  MOV  B,#48
0332 A4 MUL  AB
0333 2582    ADD  A,DPL
0335 F582    MOV  DPL,A
0337 E5F0    MOV  AB
0339 3583    ADDC A,DPH
033B F583    MOV  DPH,A      ;DPTR POINT TO ADDRESS OF DATA
033D 7930    MOV  R1,#48
033F LPDDBUF:
033F E0 MOVX A,@DPTR
0340 F8 MOV  @R0,A
0341 08 INC  R0
0342 A3 INC  DPTR
0343 D9FA    DJNZ  R1,LPDDBUF ;16 LOOP

0345 D0E0    POP  A
0347 D082    POP  DPL
0349 D083    POP  DPH
034B 22 RET

;ROUTINE FOR DISPLAY ONE LINE
;INPUT: BUF_STORE - DATA 16 BYTE TO DISPLAY
; LINE - LINE NUMBER 0 - 19
;REG: R0,R1,R2,A
034C DSPLN:
034C C083    PUSH  DPH
034E C082    PUSH  DPL
0350 C0E0    PUSH  A
0352 LPD SPLN0:
0352 7900    MOV  R1,#00
0354 782B    MOV  R0,#BUF_STORE
0356 LPD SPLN1:
0356 908000  MOV  DPTR,#CDATA1
0359 E8 MOV  A,@R0
035A F0 MOVX @DPTR,A ;DISPLAY DIGIT 1

035B 08 INC  R0
035C 908001  MOV  DPTR,#CDATA2
035F E8 MOV  A,@R0
0360 F0 MOVX @DPTR,A ;DISPLAY DIGIT 2

0361 08 INC  R0
0362 908002  MOV  DPTR,#CDATA3
0365 E8 MOV  A,@R0
0366 F0 MOVX @DPTR,A ;DISPLAY DIGIT 3

0367 08 INC  R0
0368 909000  MOV  DPTR,#CDATA4
036B E8 MOV  A,@R0
036C F0 MOVX @DPTR,A ;DISPLAY DIGIT 4

036D 08 INC  R0
036E B90007  CJNE R1,#0,NXTDSPLN1
0371 1203EB  LCALL SELECT0;LATCH PAGE 0
0374 09 INC  R1
0375 020356  LJMPLPDSPLN1
0378 NXTDSPLN1:
0378 B90107  CJNE R1,#1,NXTDSPLN2
037B 1203F0  LCALL SELECT1;LATCH PAGE 1
037E 09 INC  R1
037F 020356  LJMPLPDSPLN1
0382 NXTDSPLN2:
0382 B90207  CJNE R1,#2,NXTDSPLN3
0385 1203F5  LCALL SELECT2;LATCH PAGE 2
0388 09 INC  R1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0389 020356 L JMP LPDSPLN1
038C NXTDPSPLN3:
038C B90308 CJNE R1,#3,NXTDPSPLN4
038F 1203FA LCALL SELECT3;LATCH PAGE 3
0392 7900 MOV R1,#00
0394 02039A L JMP NXTDPSPLN5
0397 NXTDPSPLN4:
0397 020352 L JMP LPDSPLN0
039A NXTDPSPLN5:
039A 909001 MOV DPTR,#RDATA ;ENABLE LINE
039D E55B MOV A,LINE
039F F0 MOVX @DPTR,A
03A0 D0E0 POP A
03A2 D082 POP DPL
03A4 D083 POP DPH
03A6 22 RET

```

;ROUTINE FOR CLEAR DISPLAY RAM 3200 BYTES

```

03A7 CLRDSPR:
03A7 C083 PUSH DPH
03A9 C082 PUSH DPL

03AB 7C30 MOV R4,#48 ;48 DIGIT
03AD 7D14 MOV R5,#20 ;20 LINE
03AF 900000 MOV DPTR,#BUF_DSP
03B2 LPCLRDSPR0:
03B2 7400 MOV A,#00
03B4 LPCLRDSPR1:
03B4 F0 MOVX @DPTR,A
03B5 A3 INC DPTR
03B6 DCFC DJNZ R4,LPCLRDSPR1
03B8 7C30 MOV R4,#48
03BA DDF6 DJNZ R5,LPCLRDSPR0

03BC D082 POP DPL
03BE D083 POP DPH
03C0 22 RET

```

03C1 SELECTALL:

```

03C1 759000 MOV P1,#00H
03C4 75900F MOV P1,#0FH
03C7 22 RET

```

;ROUTINE FOR SELECTING PAGE DISPLAY  
;INPUT A

```

03C8 SLCTP:
03C8 B40007 CJNE A,#0,SLCTP1
03CB C290 CLR P1 OR 0 ;LATCH PAGE 0
03CD D290 SETB P1 OR 0
03CF 0203EA L JMP QSLCTP
03D2 SLCTP1:
03D2 B40104 CJNE A,#1,SLCTP2
03D5 C291 CLR P1 OR 1 ;LATCH PAGE 1
03D7 D291 SETB P1 OR 1
03D9 SLCTP2:
03D9 B40207 CJNE A,#2,SLCTP3
03DC C292 CLR P1 OR 2 ;LATCH PAGE 2
03DE D292 SETB P1 OR 2
03E0 0203EA L JMP QSLCTP
03E3 SLCTP3:
03E3 B40304 CJNE A,#3,QSLCTP
03E6 C283 CLR P1 OR 3 ;LATCH PAGE 3
03E8 D283 SETB P1 OR 3
03EA QSLCTP:
03EA 22 RET

```

03EB SELECT0:

```

03EB C290 CLR P1 OR 0 ;LATCH PAGE 0
03ED D290 SETB P1 OR 0
03EF 22 RET

```

03F0 SELECT1:

```

03F0 C291 CLR P1 OR 1 ;LATCH PAGE 1
03F2 D291 SETB P1 OR 1
03F4 22 RET

```

03F5 SELECT2:

```

03F5 C292 CLR P1 OR 2 ;LATCH PAGE 2
03F7 D292 SETB P1 OR 2
03F9 22 RET

```

03FA SELECT3:

```

03FA C293 CLR P1 OR 3 ;LATCH PAGE 3
03FC D293 SETB P1 OR 3
03FE 22 RET

```

03FF DELAY1:

```

03FF 7F00 MOV R7,#00H ;DELAY FOR EACH LINE
0401 DFFE DJNZ R7,$
0403 22 RET

```

0404 DELAY2:

```

0404 7F00 MOV R7,#00
0406 7E10 MOV R6,#10H
0408 LPDELAY2:
0408 DFFE DJNZ R7,$
040A DEFC DJNZ R6,LPDELAY2
040C 22 RET

```

040D DELAY3:

```

040D 7F00 MOV R7,#00
040F 7E00 MOV R6,#00H
0411 LPDELAY3:
0411 DFFE DJNZ R7,$
0413 DEFC DJNZ R6,LPDELAY3
0415 22 RET

```

0416 TESTRAM:

```

0416 900000 MOV DPTR,#0000
0419 LPTESTRAM:
0419 7455 MOV A,#55H
041B F0 MOVX @DPTR,A
041C E0 MOVX A,@DPTR
041D B45511 CJNE A,#55H,ERROR
0420 F4 CPL A
0421 F0 MOVX @DPTR,A
0422 E0 MOVX A,@DPTR
0423 B4AA0B CJNE A,#0AAH,ERROR
0426 A3 INC DPTR
0427 E583 MOV A,DPH
0429 B480ED CJNE A,#80H,LPTESTRAM
042C 75900F MOV P1,#0FH
042F 8003 SJMP QTESTRAM
0431 ERROR:
0431 759000 MOV P1,#00H
0434 QTESTRAM:
0434 22 RET

```

;TEST DISPLAY ONE CHARACTER

```

0435 TSTCHR:
0435 751220 MOV BYTE,#20H
0438 LPTSTCHR0:
0438 751005 MOV CURCOL,#5
043B 751108 MOV COUNT1,#8H
043E E512 MOV A,BYTE
0440 12012C LCALL WRCHR
0443 LPTSTCHR1:
0443 120272 LCALL DSPSTAT
0446 D511FA DJNZ COUNT1,LPTSTCHR1
0449 0512 INC BYTE
044B E512 MOV A,BYTE
044D B4FFE8 CJNE A,#0FFH,LPTSTCHR0
0450 751220 MOV BYTE,#20H
0453 80E3 SJMP LPTSTCHR0

```

;WRITE TEST DATA TO DISPLAY RAM

```

0455 WRTST:
0455 7C30 MOV R4,#48 ;48 DIGIT
0457 7D14 MOV R5,#20 ;20 LINE
0459 900000 MOV DPTR,#BUF_DSP
045C LPWRTST0:
045C 74FF MOV A,#0FFH
045E LPWRTST1:
045E F0 MOVX @DPTR,A
045F A3 INC DPTR
0460 DCFC DJNZ R4,LPWRTST1
0462 7C30 MOV R4,#48
0464 DDF6 DJNZ R5,LPWRTST0
0466 22 RET

```

0467 SNDBYTE:

```

0467 3099FD JNB TI,$ ;WAIT FOR READY
046A C299 CLR TI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

046C F599 MOV SBUF,A ;SEND
046E 22 RET

;LOAD DATA FROM CODE MEMORY TO DISPLAY
;INPUT :DPTR (POINT TO START MESSAGE)
;REG: R0 - R3
046F LDFRCD:
046F A882 MOV R0,DPL ;SAVE START ADDRESS OF CODE
0471 A983 MOV R1,DPH
0473 904000 MOV DPTR,#BUF_DAT ;DEFINE START ADDRESS OF DATA
0476 AA82 MOV R2,DPL
0478 AB83 MOV R3,DPH
047A LPLDFRCD:
047A E4 CLR A
047B 8882 MOV DPL,R0
047D 8983 MOV DPH,R1
047F 93 MOVX A,@A+DPTR ;READ CODE
0480 A3 INC DPTR
0481 A882 MOV R0,DPL
0483 A983 MOV R1,DPH ;SAVE ADDRESS OF CODE TABLE

0485 8B83 MOV DPH,R3
0487 8A82 MOV DPL,R2
0489 F0 MOVX @DPTR,A ;WRITE DATA
048A A3 INC DPTR
048B AA82 MOV R2,DPL
048D AB83 MOV R3,DPH
048F B404E8 CJNE A,#04H,LPLDFRCD
0492 QLDFRCD:
0492 22 RET

;ROUTINE FOR SHOW LOGO MESSAGE
0493 SHWLOGO:
0493 751000 MOV CURCOL,#0
0496 900D15 MOV DPTR,#BUS_LOGOT1
0499 12046F LCALL LDFRCD
049C 1200C0 LCALL SHWMSG
049F 22 RET

;ROUTINE FOR RECEIVE DATA FROM SERIAL PORT
04A0 RECVDAT:
04A0 C082 PUSH DPL ;SAVE NEW PROGRAM COUNTER
04A2 C083 PUSH DPH
04A4 C0E0 PUSH A ;SAVE REGISTER
04A8 C0F0 PUSH B
04AA D2D3 SETB PSW OR 3

04AA C2AF CLR EA ;DISABLE ALL INTERRUPT
04AC C298 CLR RI
04AE E599 MOV A,SBUF
04B0 F513 MOV PATTERN,A ;FIRST BYTE IS PATTERN BYTE
04B2 1202DF LCALL OFFDSP
04B5 904000 MOV DPTR,#BUF_DAT
04B8 LPRECVDAT:
04B8 3098FD JNB RI,$ ;WAIT FOR DATA
04BB C298 CLR RI
04BD E599 MOV A,SBUF
04BF F0 MOVX @DPTR,A
04C0 A3 INC DPTR
04C1 B440F4 CJNE A,#0,LPRECVDAT ;CHECK END OF MESSAGE
04C4 7404 MOV A,#04
04C6 1582 DEC DPL
04C8 F0 MOVX @DPTR,A
04C9 7451 MOV A,#0C
04CB D299 SETB TI
04CD 120467 LCALL SNDBYTE
04D0 12040D LCALL DELAY3
04D3 C299 CLR TI
04D5 D202 SETB FG_NEW

04D7 C2D3 CLR PSW OR 3 ;RESTORE REGISTER
04D9 D0F0 POP B
04DB D0E0 POP A
04DD D083 POP DPH
04DF D082 POP DPL
04E1 D2AF SETB EA
04E3 32 RETI

04E4 INCL "THAIPTAT.ASM"
; THAI PATTERN FOR SMO STANDARD

04E4 THAIPTAT:
04E4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=000,00H
04EC 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=001,01H
04F4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=002,02H
04FC 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=003,03H
0504 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=004,04H
050C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=005,05H
0514 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=006,06H
051C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=007,07H
0524 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=008,08H
052C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=009,09H
0534 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=010,0AH
053C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=011,0BH
0544 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=012,0CH
054C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=013,0DH
0554 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=014,0EH
055C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=015,0FH

;
0564 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=016,10H
056C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=017,11H
0574 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=018,12H
057C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=019,13H
0584 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=020,14H
058C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=021,15H
0594 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=022,16H
059C 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=023,17H
05A4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=024,18H
05AC 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=025,19H
05B4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=026,1AH
05BC 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=027,1BH
05C4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=028,1CH
05CC 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=
029,1DH
05D4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=030,1EH
05DC 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=031,1FH

05E4 000000000 DFB 00H,00H,00H,00H,00H,00H,00H,00H ; code=032,20H
05EC 0C1E1E0C0C DFB 0CH,1EH,1EH,0CH,0CH,00H,0CH,00H ; code=
033,21H,1
05F4 3636362400 DFB 36H,36H,36H,24H,00H,00H,00H,00H ; code=
034,22H,*
05FC 36367F367F DFB 36H,36H,7FH,36H,7FH,36H,36H,00H ; code=
035,23H,#
0604 0C3E031E30 DFB 0CH,3EH,03H,1EH,30H,1FH,0CH,00H ; code=
036,24H,$
060C 066333180C DFB 00H,63H,33H,18H,0CH,66H,63H,00H ; code=
037,25H,%
0614 0C120C0C52 DFB 0CH,12H,0CH,0CH,52H,22H,5CH,00H ; code=
038,26H,&
061C 0606030000 DFB 06H,06H,03H,00H,00H,00H,00H,00H ; code=
039,27H,'
0624 180C060606 DFB 18H,0CH,06H,06H,06H,0CH,18H,00H ; code=
040,28H,(
062C 060C181818 DFB 06H,0CH,18H,18H,18H,0CH,06H,00H ; code=
041,29H,)
0634 00663CF3C DFB 00H,66H,3CH,0FFH,3CH,66H,00H,00H ; code=
042,2AH,*
063C 000C0C3F0C DFB 00H,0CH,0CH,3FH,0CH,0CH,00H,00H ; code=
043,2BH,+
0644 000000000 DFB 00H,00H,00H,00H,00H,0CH,0CH,06H ; code=
044,2CH,.
064C 0000003F00 DFB 00H,00H,00H,3FH,00H,00H,00H,00H ; code=
045,2DH,-
0654 000000000 DFB 00H,00H,00H,00H,00H,0CH,0CH,00H ; code=
046,2EH,.
065C 6030180C06 DFB 60H,30H,18H,0CH,06H,03H,01H,00H ; code=
047,2FH/

;
0664 3E63737B6F DFB 3EH,63H,73H,7BH,6FH,67H,3EH,00H ; code=
048,30H,0
066C 0C0E0C0C0C DFB 0CH,0EH,0CH,0CH,0CH,0CH,3FH,00H ; code=
049,31H,1
0674 1E33301C06 DFB 1EH,33H,30H,1CH,06H,33H,3FH,00H ; code=
050,32H,2
067C 1E33301C30 DFB 1EH,33H,30H,1CH,30H,33H,1EH,00H ; code=
051,33H,3
0684 383C38337F DFB 38H,3CH,36H,33H,7FH,30H,78H,00H ; code=
052,34H,4
068C 3F031F3030 DFB 3FH,03H,1FH,30H,30H,33H,1EH,00H ; code=
053,35H,5
0694 1C06031F33 DFB 1CH,06H,03H,1FH,33H,33H,1EH,00H ; code=
054,36H,6
069C 3F3330180C DFB 3FH,33H,30H,18H,0CH,0CH,0CH,00H ; code=
055,37H,7
06A4 1E33331E33 DFB 1EH,33H,33H,1EH,33H,33H,1EH,00H ; code=

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

056,38H,8		07EC 00001E303E	DFB 00H,00H,1EH,30H,3EH,33H,6EH,00H	: code-
06AC 1E33333E30	DFB 1EH,33H,33H,3EH,30H,18H,0EH,00H	097,61H,a		
057,39H,9		07F4 0706063E66	DFB 07H,06H,06H,3EH,66H,66H,3BH,00H	: code-
06B4 000C0C0000	DFB 00H,0CH,0CH,00H,00H,0CH,0CH,00H	098,62H,b		
058,3AH,:		07FC 00001E3303	DFB 00H,00H,1EH,33H,03H,33H,1EH,00H	: code-
06BC 000C0C0000	DFB 00H,0CH,0CH,00H,00H,0CH,0CH,06H	099,63H,c		
059,3BH,:		0804 3830303E33	DFB 38H,30H,30H,3EH,33H,33H,6EH,00H	: code-
06C4 180C060306	DFB 18H,0CH,06H,03H,06H,0CH,18H,00H	100,64H,d		
060,3CH,<		080C 00001E333F	DFB 00H,00H,1EH,33H,3FH,03H,1EH,00H	: code-
06CC 00003F0000	DFB 00H,00H,3FH,00H,00H,3FH,00H,00H	101,65H,e		
061,3DH,=		0814 1C36060F06	DFB 1CH,36H,06H,0FH,06H,06H,0FH,00H	: code-
06D4 060C183018	DFB 06H,0CH,18H,30H,18H,0CH,06H,00H	102,66H,f		
062,3EH,>		081C 0000E3131	DFB 00H,00H,6EH,31H,31H,3EH,30H,1FH	: code-
06DC 1E3330180C	DFB 1EH,33H,30H,18H,0CH,00H,0CH,00H	103,67H,g		
063,3FH,?		0824 0706366E66	DFB 07H,06H,36H,6EH,66H,66H,67H,00H	: code-
06E4 3E637B7B7B	DFB 3EH,63H,7BH,7BH,7BH,03H,3EH,00H	104,68H,h		
064,40H,@		082C 0C00E0C0C	DFB 0CH,00H,0EH,0CH,0CH,0CH,1EH,00H	: code-
06EC 0C1E33333F	DFB 0CH,1EH,33H,33H,3FH,33H,33H,00H	105,69H,i		
065,41H,OA		0834 3000303030	DFB 30H,00H,30H,30H,30H,33H,33H,1EH	: code-
06F4 3F6663F66	DFB 3FH,66H,66H,3FH,66H,66H,3FH,00H	106,6AH,j		
066,42H,OB		083C 070666361E	DFB 07H,06H,66H,36H,1EH,36H,67H,00H	: code-
06FC 3C66030303	DFB 3CH,66H,03H,03H,03H,66H,3CH,00H	107,6BH,k		
067,43H,OC		0844 0E0C0C0C0C	DFB 0EH,0CH,0CH,0CH,0CH,0CH,1EH,00H	: code-
0704 1F36666666	DFB 1FH,36H,66H,66H,66H,36H,1FH,00H	108,6CH,l		
068,44H,OD		084C 0000337F7F	DFB 00H,00H,33H,7FH,7FH,6BH,63H,00H	: code-
070C 7F46161E16	DFB 7FH,46H,16H,1EH,16H,46H,7FH,00H	109,6DH,m		
069,45H,OE		0854 00001F3333	DFB 00H,00H,1FH,33H,33H,33H,00H	: code-
0714 7F46161E16	DFB 7FH,46H,16H,1EH,16H,06H,0FH,00H	110,6EH,n		
070,46H,OF		085C 00001E3333	DFB 00H,00H,1EH,33H,33H,33H,1EH,00H	: code-
071C 3C6603037B	DFB 3CH 66H,03H,03H,7BH,66H,7CH,00H	111,6FH,o		
071,47H,G				
0724 3333333F33	DFB 33H,33H,33H,3FH,33H,33H,33H,00H	0864 00003B6666	DFB 00H,00H,3BH,66H,66H,3EH,06H,0FH	: code-
072,48H,H		112,70H,p		
072C 1E0C0C0C0C	DFB 1EH,0CH,0CH,0CH,0CH,0CH,1EH,00H	086C 0000E33333	DFB 00H,00H,6EH,33H,33H,3EH,30H,7BH	: code-
073,49H,I		113,71H,q		
0734 7830303033	DFB 78H,30H,30H,30H,33H,33H,1EH,00H	0874 00003B6E66	DFB 00H,00H,3BH,6EH,66H,06H,0FH,00H	: code-
074,4AH,J		114,72H,r		
073C 67661E1E36	DFB 67H,66H,1EH,1EH,36H,66H,67H,00H	087C 00003E031E	DFB 00H,00H,3EH,03H,1EH,30H,1FH,00H	: code-
075,4BH,K		115,73H,s		
0744 0F06060648	DFB 0FH,06H,06H,06H,46H,66H,7FH,00H	0884 080C3E0C0C	DFB 08H,0CH,3EH,0CH,0CH,2CH,18H,00H	: code-
076,4CH,L		116,74H,t		
074C 63777F7F6B	DFB 63H,77H,7FH,7FH,6BH,63H,63H,00H	088C 0000333333	DFB 00H,00H,33H,33H,33H,33H,6EH,00H	: code-
077,4DH,M		117,75H,u		
0754 63676F7B73	DFB 63H,67H,6FH,7BH,73H,63H,63H,00H	0894 0000333333	DFB 00H,00H,33H,33H,33H,1EH,0CH,00H	: code-
078,4EH,N		118,76H,v		
075C 1C36636363	DFB 1CH,36H,63H,63H,63H,36H,1CH,00H	089C 0000636B7F	DFB 00H,00H,63H,6BH,7FH,7FH,36H,00H	: code-
079,4FH,O		119,77H,w		
		08A4 000063361C	DFB 00H,00H,63H,36H,1CH,36H,63H,00H	: code-
0764 3F66663E06	DFB 3FH,66H,66H,3EH,06H,06H,0FH,00H	120,78H,x		
080,50H,P		08AC 0000333333	DFB 00H,00H,33H,33H,33H,3EH,30H,1FH	: code-
076C 1E33333333	DFB 1EH,33H,33H,33H,33H,1EH,3BH,00H	121,79H,y		
081,51H,Q		08B4 00003F1B0C	DFB 00H,00H,3FH,19H,0CH,26H,3FH,00H	: code-
0774 3F66663E36	DFB 3FH,66H,66H,3EH,36H,66H,67H,00H	122,7AH,z		
082,52H,R		08BC 380C0C070C	DFB 38H,0CH,0CH,07H,0CH,0CH,38H,00H	: code-
077C 1E33060C18	DFB 1EH,33H,06H,0CH,18H,33H,1EH,00H	123,7BH,{		
083,53H,S		08C4 1818180018	DFB 18H,18H,18H,00H,18H,18H,18H,00H	: code-
0784 3F2D0C0C0C	DFB 3FH,2DH,0CH,0CH,0CH,0CH,1EH,00H	124,7CH,		
084,54H,T		08CC 070C0C380C	DFB 07H,0CH,0CH,38H,0CH,0CH,07H,00H	: code-
078C 3333333333	DFB 33H,33H,33H,33H,33H,33H,3FH,00H	125,7DH,)		
085,55H,U		08D4 CE3B000000	DFB 0CEH,3BH,00H,00H,00H,00H,00H	: code-
0794 3333333333	DFB 33H,33H,33H,33H,33H,1EH,0CH,00H	126,7EH,-		
086,56H,V		08DC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-127,7FH
079C 6363636A7F	DFB 63H,63H,63H,6AH,7FH,77H,63H,00H			
087,57H,W		08E4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-128,80H
07A4 6363361C1C	DFB 63H,63H,36H,1CH,1CH,36H,63H,00H	08EC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-129,81H
088,58H,X		08F4 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-130,82H
07AC 3333331E0C	DFB 33H,33H,33H,1EH,0CH,0CH,1EH,00H	08FC 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-131,83H
089,59H,Y		0904 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-132,84H
07B4 7F6331184C	DFB 7FH,63H,31H,18H,4CH,66H,6FH,00H	090C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-133,85H
090,5AH,Z		0914 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-134,86H
07BC 1E06060606	DFB 1EH,06H,06H,06H,06H,06H,1EH,00H	091C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-135,87H
091,5BH,]		0924 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-136,88H
07C4 0C060C1830	DFB 03H,06H,0CH,18H,30H,60H,40H,00H	092C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-137,89H
092,5CH,^		0934 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-138,8AH
07CC 1E18181818	DFB 1EH,18H,18H,18H,18H,18H,1EH	093C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-139,8BH
093,5DH,^		0944 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-140,8CH
07D4 081C366300	DFB 08H,1CH,36H,63H,00H,00H,00H,00H	094C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-141,8DH
094,5EH,^		0954 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-142,8EH
07DC 0000000000	DFB 00H,00H,00H,00H,00H,00H,0FFH	095C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-143,8FH
095,5FH,^				
		0964 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-144,90H
07E4 0C0C180000	DFB 0CH,0CH,18H,00H,00H,00H,00H	096C 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-145,91H
096,60H,^		0974 0000000000	DFB 00H,00H,00H,00H,00H,00H,00H	: code-146,92H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

097C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-147,93H	:		
0984 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-148,94H	0AE4 0038444B4C	DFB 00H,38H,44H,48H,4CH,4AH,4CH,00H	: smo-
098C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-149,95H	192,0C0H,porsampo		
0994 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-150,96H	0AE4 00444A4C4C	DFB 00H,44H,4AH,4CH,4CH,5AH,24H,00H	: smo-
099C 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-151,97H	193,0C1H,morma		
09A4 000000000	DFB 00H,00H,00H,00H,00H,00H,00H,00H	: code-152,98H	0AF4 00242A262C	DFB 00H,24H,2AH,26H,2CH,24H,1CH,00H	: smo-
09AC 00000F808	DFB 00H,00H,00H,0F8H,08H,08H,08H,08H	: code-	194,0C2H,yoryak		
153,99H			0AFC 007C023C30	DFB 00H,7CH,02H,3CH,30H,28H,10H,00H	: smo-
09B4 000000F08	DFB 00H,00H,00H,0FH,08H,08H,08H,08H	: code-154,9AH	195,0C3H,rorru		
09BC 080808F800	DFB 08H,08H,08H,0F8H,00H,00H,00H,00H	: code-	0804 001C222426	DFB 00H,1CH,22H,24H,26H,2AH,24H,20H	: smo-
155,9BH			196,0C4H,lorru		
09C4 0808080F08	DFB 08H,08H,08H,0FH,00H,00H,00H,00H	: code-156,9CH	080C 003C425C86	DFB 00H,3CH,42H,5CH,66H,6AH,24H,00H	: smo-
09CC 0808080808	DFB 08H,08H,08H,08H,08H,08H,08H,08H	: code-	197,0C5H,lorfing		
157,9DH			0814 001C222426	DFB 00H,1CH,22H,24H,26H,25H,22H,20H	: smo-
09D4 080808F808	DFB 08H,08H,08H,0F8H,08H,08H,08H,08H	: code-	198,0C6H,lorru		
158,9EH			081C 0078444060	DFB 00H,78H,44H,40H,60H,50H,20H,00H	: smo-
09DC 0808080F08	DFB 08H,08H,08H,0FH,08H,08H,08H,08H	: code-159,9FH	199,0C7H,vorvan		
:			0824 407C42526A	DFB 40H,7CH,42H,52H,6AH,5AH,4CH,00H	: smo-
09E4 FFFFFFFF	DFB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH	: smo-	200,0C8H,sorsatar		
160,0A0H,all black			082C 00446AEC48	DFB 00H,44H,6AH,0ECH,48H,44H,78H,00H	: smo-
09EC 003C242444	DFB 00H,3CH,42H,44H,44H,44H,44H,00H	: smo-	201,0C9H,sorsusi		
161,0A1H,kokai			0834 403C425C86	DFB 40H,3CH,42H,5CH,66H,6AH,24H,00H	: smo-
09F4 0048545850	DFB 00H,48H,54H,58H,50H,50H,30H,00H	: smo-	202,0CAH,sorsua		
162,0A2H,khokhai			083C 0022552654	DFB 00H,22H,55H,26H,54H,52H,5CH,00H	: smo-
09FC 005A555250	DFB 00H,5AH,55H,52H,50H,50H,30H,00H	: smo-	203,0CBH,horheep		
163,0A3H,khokhuad			0844 A052654854	DFB 0A0H,52H,65H,46H,54H,54H,28H,00H	: smo-
0A04 003C42425A	DFB 00H,3CH,42H,42H,5AH,56H,44H,00H	: smo-	204,0CCH,lorjular		
164,0A4H,korkwai			084C 003C42444A	DFB 00H,3CH,42H,44H,4AH,46H,3CH,00H	: smo-
0A0C 00245A425A	DFB 00H,24H,5AH,42H,5AH,56H,44H,00H	: smo-	205,0CDH,orang		
165,0A5H,korkon			0854 603C5E4442	DFB 60H,3CH,5EH,44H,42H,46H,3CH,00H	: smo-
0A14 004A55504C	DFB 00H,4AH,55H,50H,4CH,4AH,34H,00H	: smo-	206,0CEH,hornokhook		
166,0A6H,korrakang			085C 002452524C	DFB 00H,24H,52H,52H,4CH,40H,40H,70H	: smo-
0A1C 0020546444	DFB 00H,20H,54H,64H,44H,48H,30H,00H	: smo-	207,0CFH,pariyarnnoi		
167,0A7H,ngomgpo			:		
0A24 003C404854	DFB 00H,3CH,40H,48H,54H,58H,30H,00H	: smo-	0864 00442A1C44	DFB 00H,44H,2AH,1CH,44H,2AH,1CH,00H	: smo-
168,0A8H,jorjarn			208,0D0H,SARA-A		
0A2C 001E21246A	DFB 00H,1EH,21H,24H,6AH,0ACH,58H,00H	: smo-	086C 0088543800	DFB 00H,88H,54H,38H,00H,00H,00H,00H	: smo-
169,0A9H,chorching			209,0D1H,MAIHANARGARD		
0A34 8048345850	DFB 80H,48H,34H,58H,50H,48H,70H,00H	: smo-	0874 0018242020	DFB 00H,18H,24H,20H,20H,20H,20H,00H	: smo-
170,0AAH,chorchang			210,0D2H,SARA-AR		
0A3C 804A353250	DFB 80H,4AH,35H,32H,50H,46H,70H,00H	: smo-	087C 031B242020	DFB 03H,1BH,24H,20H,20H,20H,20H,00H	: smo-
171,0ABH,sorso			211,0D3H,SARA-AM		
0A44 005EA1A2B3	DFB 00H,5EH,0A1H,0A2H,0B3H,0ABH,0D2H,00H	: smo-	0884 0010287C00	DFB 00H,10H,28H,7CH,00H,00H,00H,00H	: smo-
172,0ACH,chorfakachur			212,0D4H,SARA-I		
0A4C 008E919293	DFB 00H,8EH,91H,92H,93H,0F5H,82H,0F0H	: smo-	088C 4050687C00	DFB 40H,50H,68H,7CH,00H,00H,00H,00H	: smo-
173,0ADH,yorying			213,0D5H,SARA-E		
0A54 003844484C	DFB 00H,38H,44H,48H,4CH,4AH,0C4H,7EH	: smo-	0894 0050A87C00	DFB 00H,50H,0A8H,7CH,00H,00H,00H,00H	: smo-
174,0AEH,chorchadar			214,0D6H,SARA-AU		
0A5C 003844484C	DFB 00H,38H,44H,48H,4CH,4AH,0F4H,2AH	: smo-	089C 00A8B47E00	DFB 00H,0A8H,0B4H,7EH,00H,00H,00H,00H	: smo-
175,0AFH,torpatak			215,0D7H,SARA-AUU		
:			08AA 2050604000	DFB 20H,50H,60H,40H,00H,00H,00H,00H	: smo-
0A64 403C624854	DFB 40H,3CH,62H,48H,54H,58H,22H,75H	: smo-	216,0D8H,SARA-U		
176,0B0H,tortam			08AC 4854582000	DFB 48H,54H,58H,20H,00H,00H,00H,00H	: smo-
0A6C 006A554A48	DFB 00H,6AH,55H,4AH,48H,48H,48H,00H	: smo-	217,0D9H,SARA-UU		
177,0B1H,tomangmomo			08BA 0000000000	DFB 00H,00H,00H,00H,00H,20H,50H,20H	: smo-
0A74 009AA5A5B8	DFB 00H,9AH,0A5H,0A5H,0BBH,0ADH,56H,00H	: smo-	218,0DAH,		
178,0B2H,torpooe			08BC 000000FF00	DFB 00H,00H,00H,0FFH,00H,00H,00H,00H	: smo-
0A7C 004E515253	DFB 00H,4EH,51H,52H,53H,0B5H,52H,00H	: smo-	219,0DBH,		
179,0B3H,norned			08CA 080808FF00	DFB 08H,08H,08H,0FFH,00H,00H,00H,00H	: smo-
0A84 003C424A56	DFB 00H,3CH,42H,4AH,58H,5AH,4CH,00H	: smo-	220,0DCH,		
180,0B4H,dordex			08CC 000000FF08	DFB 00H,00H,00H,0FFH,08H,08H,08H,08H	: smo-
0A8C 00344AA4A58	DFB 00H,34H,4AH,4AH,56H,5AH,4CH,00H	: smo-	221,0DDH,		
181,0B5H,kortoa			08DA 080808FF08	DFB 08H,08H,08H,0FFH,08H,08H,08H,08H	: smo-
0A94 003C424446	DFB 00H,3CH,42H,44H,46H,4AH,44H,00H	: smo-	222,0DEH,		
182,0B6H,tortung			08DC BF66663E6E	DFB 08FH,66H,66H,3EH,6EH,66H,3EH,01H	: smo-
0A9C 00344AA4C48	DFB 00H,34H,4AH,4CH,48H,48H,48H,00H	: smo-	223,0DFH,bahl		
183,0B7H,tortaharn			:		
0AA4 40387C4848	DFB 40H,38H,7CH,48H,48H,48H,1CH,00H	: smo-	08E4 0008080818	DFB 00H,08H,08H,08H,18H,28H,10H,00H	: smo-
184,0B8H,kortong			224,0E0H,SARA-A		
0AAC 00242A2C68	DFB 00H,24H,2AH,2CH,68H,0A4H,5CH,00H	: smo-	08EC 0012121236	DFB 00H,12H,12H,12H,36H,5AH,24H,00H	: smo-
185,0B9H,nornoo			225,0E1H,SARA-AIR		
0AB4 00444AA4C48	DFB 00H,44H,4AH,4CH,48H,48H,38H,00H	: smo-	08FA FC021C1030	DFB 0FCH,02H,1CH,10H,30H,50H,20H,00H	: smo-
186,0BAH,borbaimai			226,0E2H,SARA-O		
0ABC 4044AA4C48	DFB 40H,44H,4AH,4CH,48H,48H,38H,00H	: smo-	08FC 1C22241030	DFB 1CH,22H,24H,10H,30H,50H,20H,00H	: smo-
187,0BBH,porpla			227,0E3H,SARA-AIMAIMUAN		
0AC4 0048544CS4	DFB 00H,48H,54H,4CH,54H,54H,28H,00H	: smo-	0C04 0A14101030	DFB 0AH,14H,10H,10H,30H,50H,20H,00H	: smo-
188,0BCH,porpaung			228,0E4H,SARA-AIMAIMALAI		
0ACC 4048544CS4	DFB 40H,48H,54H,4CH,54H,54H,28H,00H	: smo-	0C0C 0018242020	DFB 00H,18H,24H,20H,20H,20H,20H,20H	: smo-
189,0BDH,torta			229,0E5H,		
0AD4 0042454854	DFB 00H,42H,45H,46H,54H,54H,28H,00H	: smo-	0C14 00142A2430	DFB 00H,14H,2AH,24H,30H,08H,08H,08H	: smo-
190,0BEH,porpam			230,0E6H,MAIYAMOK		
0ADC 4042454854	DFB 40H,42H,45H,46H,54H,54H,28H,00H	: smo-	0C1C 8058A85800	DFB 80H,58H,0A8H,58H,00H,00H,00H,00H	: smo-
191,0BFH,lorfun			231,0E7H,MAI-TAIKOO		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0C24 4040400000	DFB 40H,40H,40H,00H,00H,00H,00H,00H	: smo-	00F1 LPSHWMSGM2	0101 LPSHWMSG51	010D LPSHWMSG52
232,0EBH,MAJ-EK			0419 LPTESTRAM	0438 LPTSTCHR0	0443 LPTSTCHR1
*0C2C 9850380000	DFB 98H,50H,38H,00H,00H,00H,00H,00H	: smo-	0155 LPWRCHR1	011C LPWRSTR1	045C LPWRTST0
233,0E9H,MAJ-TOR			045E LPWRTST1	0198 MAIEK	01A1 MAIEK1
0C34 80A8544800	DFB 80H,0A8H,54H,48H,00H,00H,00H,00H	: smo-	018A MAIHAN	01BC MAIJATAWA	01C5 MAIJATAWA1
234,0EAH,MAJ-TREE			0089 MAIN	009F MAIN1	00A8 MAIN2
0C3C 40E0400000	DFB 40H,0E0H,40H,00H,00H,00H,00H,00H	: smo-	00B1 MAIN3	00BA MAIN4	01C8 MAITAIKU
235,0EBH,MAJ-JATTAVAR			0180 MAITEE	01B9 MAITEE1	01A4 MAITOE
0C44 C030281000	DFB 0C0H,30H,28H,10H,00H,00H,00H,00H	: smo-	01AD MAITOE1	0378 NXTDSPLN1	0382 NXTDSPLN2
236,0ECH,KARAN			038C NXTDSPLN3	0397 NXTDSPLN4	039A NXTDSPLN5
0C4C 0000000000	DFB 00H,00H,00H,00H,00H,40H,0A0H,40H	: smo-	0099 NXTMAIN0	014F NXTWRCHR1	02DF OFFDSP
237,0EDH,			00D2 OV	00D0 P	0080 P0
0C54 0000000000	DFB 00H,00H,00H,00H,00H,80H,60H,20H	: smo-	0090 P1	0090 P1.0	0091 P1.1
238,0EEH,			0092 P1.2	0093 P1.3	0097 P1.7
0C5C 00001C222A	DFB 00H,00H,1CH,22H,2AH,22H,1CH,00H	: smo-	00A0 P2	00B0 P3	00B0 P3.0
239,0EFH,			00B1 P3.1	00B2 P3.2	00B3 P3.3
:			00B4 P3.4	00B5 P3.5	00B6 P3.6
0C84 003C424242	DFB 00H,3CH,42H,42H,42H,42H,3CH,00H	: smo-	0013 PATTERN	01CE PAYANCHANA	0087 PCON
240,0F0H,thai 0			8003 PPI1CON	9003 PPI2CON	008C PS
0C8C 003C425A5E	DFB 00H,3CH,42H,5AH,5EH,40H,38H,00H	: smo-	00D0 PSW	00B9 PT0	00B8 PT1
241,0F1H,thai 1			00B8 PX0	00BA PX1	0492 QLDFRCD
0C74 00546C554A	DFB 00H,54H,6CH,55H,4AH,42H,3CH,00H	: smo-	03EA QSLCTP	0434 QTESTRAM	026B QWRCHR
242,0F2H,thai 2			0129 QWRSTR	009A RB8	9001 RDATA
0C7C 006C8A5248	DFB 00H,6CH,8AH,52H,46H,4AH,24H,00H	: smo-	04A0 RECVDAT	009C REN	0098 RI
243,0F3H,thai 3			00D3 RSO	00D4 RS1	016A SARA_IE
0C84 807C02122A	DFB 80H,7CH,02H,12H,2AH,1AH,7CH,00H	: smo-	0172 SARA_IEE	017A SARA_UG	0164 SARA_UU
244,0F4H,thai 4			0182 SARA_UUH	0000 SAVEC	0099 SBUF
0C8C 807C02122A	DFB 0B0H,7CH,02H,12H,2AH,1AH,7CH,00H	: smo-	0098 SCON	03EB SELECT0	03F0 SELECT1
245,0F5H,thai 5			03F5 SELECT2	03FA SELECT3	03C1 SELECTALL
0C94 013E40444A	DFB 01H,3EH,40H,44H,4AH,46H,3CH,00H	: smo-	02FE SHFTL	02EA SHFTR	0493 SHWLOGO
246,0F6H,thai 6			00C0 SHWMSG1	00DC SHWMSGM	00F8 SHWMSG5
0C9C 8094AA6A66	DFB 80H,94H,0AAH,6AH,66H,6AH,64H,00H	: smo-	03C8 SLCTP	03D2 SLCTP1	03D9 SLCTP2
247,0F7H,thai 7			03E3 SLCTP3	009F SMO	009E SM1
0C4A 807C020A2A	DFB 80H,7CH,02H,0AH,2AH,5AH,34H,00H	: smo-	009D SM2	0487 SNDBYTE	0081 SP
248,0F8H,thai 8			0050 START	009B T88	0088 TCON
0C4C 80543A1228	DFB 80H,54H,3AH,12H,26H,4AH,04H,00H	: smo-	0416 TESTRAM	008D TFO	008F TF1
249,0F9H,thai 9			008C TH0	008D TH1	04E4 THAIPAT
0C84 005A5D5650	DFB 00H,5AH,5DH,56H,50H,28H,14H,00H	: smo-	0099 TI	008A TLO	008B TL1
250,0FAH,thai			0089 TMOD	008C TR0	008E TR1
0C8C 0000021556	DFB 00H,00H,02H,15H,56H,0AEH,00H,00H	: smo-	0435 TSTCHR	012C WRCHR	0114 WRSTR
251,0FBH,thai			0123 WRSTR1	0455 WRTST	
0CC4 103070FF70	DFB 10H,30H,70H,0FFH,70H,30H,10H,00H	: smo-			
252,0FCH,thai					
0CCC 010307FF07	DFB 01H,03H,07H,0FFH,07H,03H,01H,00H	: smo-			
253,0FDH,thai					
0CDD 011C3E7F01	DFB 01H,1CH,3EH,7FH,01H,01H,01H,01H	: smo-			
254,0FEH,thai					
0CDC 010101017F	DFB 01H,01H,01H,01H,7FH,3EH,1CH,01H	: smo-			
255,0FFH,thai					

0CE4 425553204DBUS\_LOGOE1: DFB "BUS MANAGEMENT SYSTEM

Mr.WORAPHON LILAKIATSAKUN",04H

0D15 C3D08ABA8BUS\_LOGOT1: DFB "ระบบจัดการบนรถประจำทาง ไทย

นายพรพล สีลาเกียรติสกุล",04h

0000 END

00E0 A

00D6 AC

00F0 B

002A BUF\_CNT

4000 BUF\_DAT

0000 BUF\_DSP

0022 BUF\_MAP

002B BUF\_STORE

0CE4 BUS\_LOGOE1

0D15 BUS\_LOGOT1

0012 BYTE

8000 CDATE1

8001 CDATE2

8002 CDATE3

9000 CDATE4

03A7 CLRDSR

0011 COUNT1

0010 CURCOL

00D7 CY

03FF DELAY1

0404 DELAY2

040D DELAY3

0083 DPH

0082 DPL

0293 DSPFLT

034C DSPLN

02B9 DSPRGT

0272 DSPSTAT

00AF EA

0431 ERROR

00AC ES

00A9 ET0

00AB ET1

00A8 EX0

00AA EX1

00D5 F0

0002 FG\_NEW

0001 FG\_SARA

0089 IE0

008B IE1

00A8 IEC

0057 INIT8255

006A INITSER

00B8 IPC

0088 ITO

008A IT1

0192 KARAN

0322 LDBUF

046F LDFRCD

01D1 LEVEL1

01F9 LEVEL2

021F LEVEL3

0248 LEVEL4

005B LINE

03B2 LPCLRDSR0

03B4 LPCLRDSR1

0408 LPDELAY2

0411 LPDELAY3

0295 LPDSPFLT0

029A LPDSPFLT1

0352 LPDSPLN0

0356 LPDSPLN1

02BB LPDSPRGT0

02C0 LPDSPRGT1

0276 LPDSPSTAT0

027B LPDSPSTAT1

033F LPLDBUF

047A LPLDFRCD

01E4 LPLEVEL1

020C LPLEVEL2

0231 LPLEVEL3

025B LPLEVEL4

04B8 LPRECVDAT

0306 LPSHFTL0

02F2 LPSHFTTR0

00C9 LPSHWMSG1

00D5 LPSHWMSG2

00E5 LPSHWMSGM1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

นายวรพล ลีลาเกียรติสกุล เกิดเมื่อวันที่ 19 มิถุนายน พ.ศ.2512 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2533 และเข้าทำงานเป็นอาจารย์ประจำมหาวิทยาลัยมหานคร ตั้งแต่ พ.ศ.2534 ปัจจุบันดำรงตำแหน่งหัวหน้าห้องปฏิบัติการไมโครโปรเซสเซอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้